COMPREHENSIVE SERVICES

We offer competitive repair and calibration services, as well as easily accessible documentation and free downloadable resources.

SELL YOUR SURPLUS

We buy new, used, decommissioned, and surplus parts from every NI series. We work out the best solution to suit your individual needs.

〜 Sell For Cash   〜 Get Credit   〜 Receive a Trade-In Deal

OBSOLETE NI HARDWARE IN STOCK & READY TO SHIP

We stock **New, New Surplus, Refurbished,** and **Reconditioned** NI Hardware.

APEX WAVES

Bridging the gap between the manufacturer and your legacy test system.
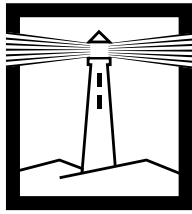
📞 1-800-915-6216

🌐 www.apexwaves.com

✉ sales@apexwaves.com

Request a Quote   ✉ CLICK HERE   *GPIB-VXI*

LOOKOUT™

Getting Started with Lookout

Worldwide Technical Support and Product Information

`www.natinst.com`

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Calgary) 403 274 9391, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521,
China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 30 1 42 96 427, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00, Singapore 2265886,
Spain (Barcelona) 93 582 0251, Spain (Madrid) 91 640 0085, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the
documentation, send e-mail to `techpubs@natinst.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

Lookout™, natinst.com™, and National Instruments™ are trademarks of National Instruments Corporation.

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

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

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

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

Contents

Chapter 3
Lookout Basics: Windows, Tools, and Files

Chapter 4
Using Lookout

Chapter 5
Developer Tour

Appendix A
Lookout Object Descriptions

Appendix B
Technical Support Resources

Glossary

Index

Figures

Tables

About This Manual

This manual provides instructions on installing and setting up Lookout. It also gives an overview of how Lookout works and how you can get started using it. This manual assumes that you are already familiar with your operating system environment.

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.
bold	Bold text denotes items that you must select or click on 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 italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *Lookout Developer's Manual*

- *Lookout Operator's Manual*

- *Lookout Object Reference Manual*

  The *Lookout Object Reference Manual* is available in Portable Document Format (PDF) on the Lookout CD in the documentation directory. The Lookout installation gives you the option of copying the PDF files into the Lookout/documentation folder on your hard drive. To view these files, you must have Adobe Acrobat Reader 3.0 or later installed. If you do not have Adobe Acrobat Reader installed, you can install it from the Lookout/documentation directory or from the Adobe web site at www.adobe.com.

- *Lookout Object Developer's Toolkit Reference*

1

Installing Lookout

Hardware and Software Requirements

Lookout requires the following:

- Pentium class PC running at 90 MHz or faster

- At least 32 MB RAM

- 45 MB free disk space, plus possibly 100 MB or more, depending on how much data you intend to log to the Lookout database, Citadel. You should also have about 50 MB of disk space for file swapping on Windows NT computers.

- Windows 98/95 or Windows NT version 4 or later

- 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.

Because Lookout can run 24 hours a day, your computer should have some form of AC power surge protection. An uninterruptible power supply (UPS) provides the ultimate protection. A UPS 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.

Lookout is Y2K compliant and requires no special considerations for the year 2000.

Testing TCP/IP

You must be properly set up on a network with TCP/IP protocols installed if you want to use the networking capabilities of Lookout 4. 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, open a DOS window 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=12
```

If TCP/IP is not working properly on your computer, consult your system administrator.

See Chapter 4, *Networking*, in the *Lookout Developer's Manual* for more information on networking and network configuration with Lookout.

Tip   If your computer does *not* have a DOS 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 DOS window you can use for the `ping` command.

Installing Lookout

Installing from the CD

1. Before installing, make sure you have shut down all applications that may currently be using ODBC. Such applications include spreadsheets, word processors, database programs, MS Query, and similar applications.

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, click on the Start icon in the taskbar and select **Run**.

   Enter *N*:\SETUP where *N* represents your CD-ROM drive. Then select **OK**.

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

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

   • When installing Lookout on a Windows NT system, you are presented with a list of options on what you may install optionally. If you want to be able to block authorized users from using <Alt-Tab> to gain access to applications other than Lookout, make sure you install the optional NT keyboard driver.

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

Installing Lookout from Floppy Diskettes

If the computer you want to install Lookout on does not have a CD-ROM drive, follow these instructions for installing the software:

1. Prepare about two dozen blank diskettes: 3.5-inch, 1.4MB. Label each diskette as Disk1, Disk2, and so on.

2. On another computer with a CD-ROM drive and diskette drive, copy the files from the individual Disk*N* subdirectories on the CD onto the appropriately labeled 3.5-inch floppy diskette. Do *not* copy the Disk*N* directory itself onto your diskettes. Copy only the contents of each directory.

3. On the computer where you want to install Lookout, insert the diskette labeled Disk1 and run the setup.exe program from the diskette.

4. Follow the installation instructions on the screen.

Registering Lookout

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 only runs for 30 days.

If you are installing Lookout as an upgrade to an earlier version, you will have already provided registration information, and Lookout will open with a request for you to log in.

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.
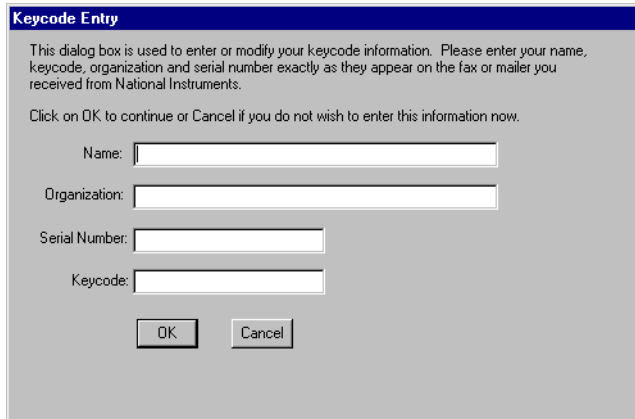
Note   When you register Lookout, you unlock it for permanent use at your appropriate I/O count. If you do not register Lookout by the end of the 30 day period, it lapses to a demo system. You must complete the license agreement and mail or fax a copy of the agreement to National Instruments in order to register Lookout. Upon receipt of the registration form, National Instruments generates a key code to unlock Lookout and faxes or mails it to you.

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»Lookout**.

    A dialog box appears asking you to register Lookout. If you are ready to register Lookout, click on **OK**. The registration dialog box appears.

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

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

    (If you have not yet received your key code 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 key code is not case sensitive and you can leave the hyphens out if desired. Notice that there are no spaces near the hyphens.

6.   After completing the entries, press <Enter> or select **OK**.

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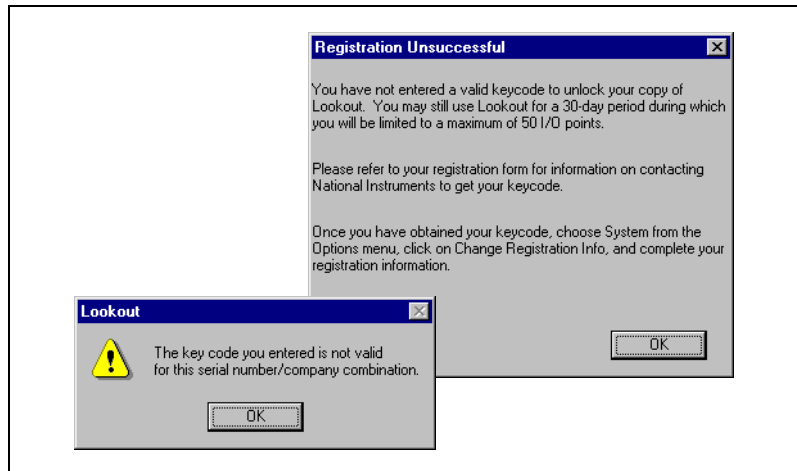
If you are certain that you typed the information correctly and Lookout does not accept it, call the National Instruments technical support line for help.

Changing Registration Information

If you want to change the number of Lookout I/O points you are using, or make other changes in system capabilities, National Instruments must send you a new Lookout keycode.

To change your registration information, select **Options»System»Change Registration Info**. Enter the new keycode in the appropriate field to unlock your additional Lookout functionality.

Adding Client Connections

You are limited in the number of client connections you are allowed to maintain in Lookout. Just as with your I/O license, you must enter a keycode.

Obtain your keycode by faxing your registration form to National Instruments, or by calling.

Unlike the I/O point registration, you can add and remove client licenses from any copy of Lookout. 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 particular keycode in addition to the keycode that authorized your first two client connections.

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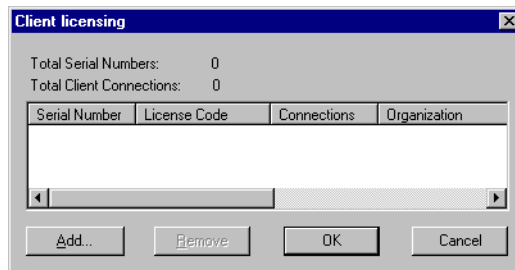
Click on the **Add** button. The following dialog box appears.
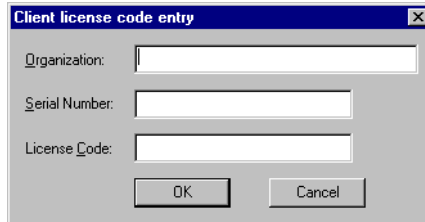
Enter your **Organization** name as you reported it in your registration request for a client keycode. Enter the **Serial Number**, and the **License Code** your received. Click on **OK**.

If your registration attempt fails, the following dialog box appears.
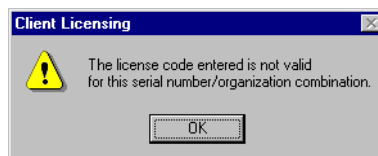
Check to make sure you entered your organization name, serial number, and keycode correctly. Contact National Instruments if correcting these entries does not fix the problem.

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

Setting System Options

The first time you open Lookout is a good time to set some of the Lookout system options.

In Lookout, select **Options»System** from the menu bar. The **System Options** dialog box appears as shown in Figure 1-1. Some default settings may be different for your computer.
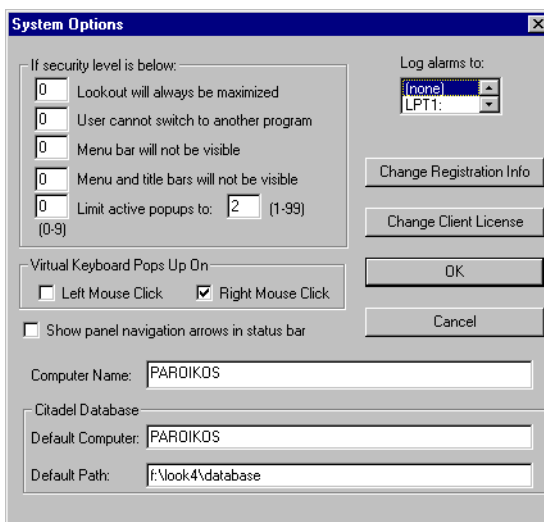
Figure 1-1. System Options Dialog Box

✎ **Note**   Only users logged on with security levels or 9 or greater can access the **System Options** dialog box.

Computer Name Setting

The computer name 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 Settings

The **Citadel Database** fields set the default destination for Citadel 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 pathname for the **Default path** setting, and the fully qualified computer name for the **Default computer** setting.

Virtual Keyboard Settings

Lookout features a virtual keyboard that you can access when in run mode by clicking on a control that accepts a text or numeric input. You can set whether a right-click or a left-click pops up this keyboard in the **Virtual Keyboard Pops Up On** field in the **System Options** dialog box.

Log Alarms Setting

If you have a printer directly connected to your computer, you can direct that all alarms can be printed when they occur. Set the **Log alarms to** box in the **System Options** dialog box to the communications port to which your printer is connected. See Chapter 9, *Alarms*, in the *Lookout Developer's Manual* for information on the alarm and event logging features of Lookout. You can also capture a network printer port on Windows 98/95 machines. Consult your operating system documentation for information on this procedure.

Security Level Settings

You can set local limits on how Lookout runs when users with different security levels are logged on. Use the **If security level is below:** selection boxes in the **System Options** dialog box to set these limits. See Chapter 6, *Security*, in the *Lookout Developer's Manual* for full details on the security features of Lookout.

Panel Navigation Arrows

Select the **Show panel navigation arrows in status bar** checkbox if you want to use 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 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.

Startup Process File Setting

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.

To get startup processes, select **Options»Startup**, and the following dialog box appears.
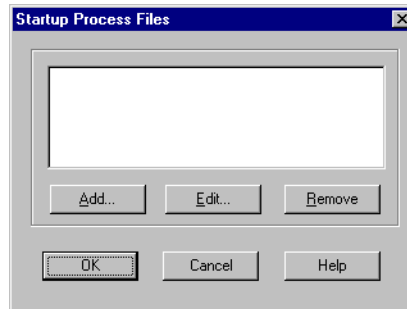
To add a file to your list of startup processes, click on the **Add** button. A dialog box you can use to browse for a file appears. (The contents of the dialog box may not be identical to the illustration.)
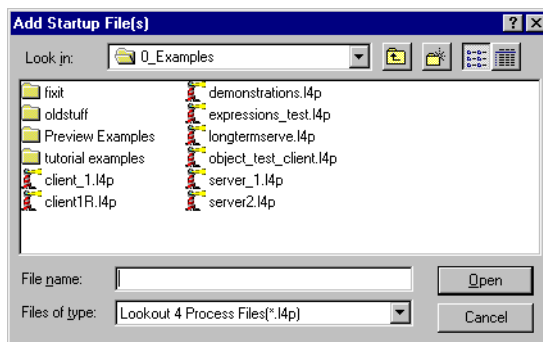
Select the file you want to run when Lookout opens and click on **Open**.

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.

To edit a path name to a file, highlight the file name and click on the **Edit** button.

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.

2

How Lookout Works

This chapter explains the basics of how Lookout works, including descriptions of objects, data members, connections, processes, and services.

Lookout is a powerful yet easy-to-use 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 from Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and other devices. Typical Lookout projects include continuous process monitoring and supervisory control, discrete manufacturing, batch applications, and remote telemetry systems.

Object-oriented and event-driven, Lookout is a configurable package that requires no programming or scripting. Instead, you use Lookout to create graphical representations on a computer screen of real-world devices such as switches, dial gauges, chart recorders, pushbuttons, knobs, sliders, meters, and then link your images to the actual field instruments using PLCs, RTUs, data acquisition boards, or other I/O devices.

Lookout has many diverse capabilities such as Statistical Process Control (SPC), recipe management, Structured Query Language (SQL), built-in security, flexible data logging, running multiple processes on one computer, sophisticated animation, complex alarming, radio and dial-up telemetry support, audit trails of events and setpoint adjustments, multimedia support, touch screen compatibility, networking (including multiple client and server processes running on one or many computers), Dynamic Data Exchange (DDE & NetDDE), and more.

With Lookout you can develop an application completely online, without shutting down. You do not have to recompile or download a database every time you make a modification, nor do you have to switch back and forth between programs. You do not even have to run separate development and configuration programs. Instead, you can add, delete and modify control panels, logic, graphics, PLCs, RTUs, I/O, and other field devices without ever interrupting your process.

Because Lookout is object-oriented and event-driven, you can use Lookout with other programs in the Microsoft Windows multitasking environment. For example, while Lookout monitors and controls your process, you can use a spreadsheet to analyze production figures of hourly average flow rates, then start a word processor to generate a memorandum, paste the spreadsheet into the memo and send it to a laser printer.

The remainder of this chapter describes Lookout architecture—the components and how they work together. This will help you understand fully how to use Lookout for all your continuous process, discreet, or batch applications.

Architecture

Once you understand the basic Lookout components and the fundamentals regarding object-oriented and event-driven structure, using the program becomes much easier.

Lookout consists mostly of objects and their data members, connections, and services. Developing a Lookout application is a matter of creating, configuring, and connecting objects. Objects are software representations of everything from potentiometers and switches to PLCs and RTUs connected physically to the computers you have running Lookout. You then make connections between the software 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.

Add to this the idea of client and server processes and you are well on your way to understanding the basic structure of all Lookout applications.

What is an Object?

A Lookout object is a self-contained software unit designed to do something specific in your HMI/SCADA application. What each object does is generally referred to as its functionality. Each object has a set of data members you can log to a predefined database, and a set of parameters. The following diagram depicts the functionality, data members, and parameters of an object.
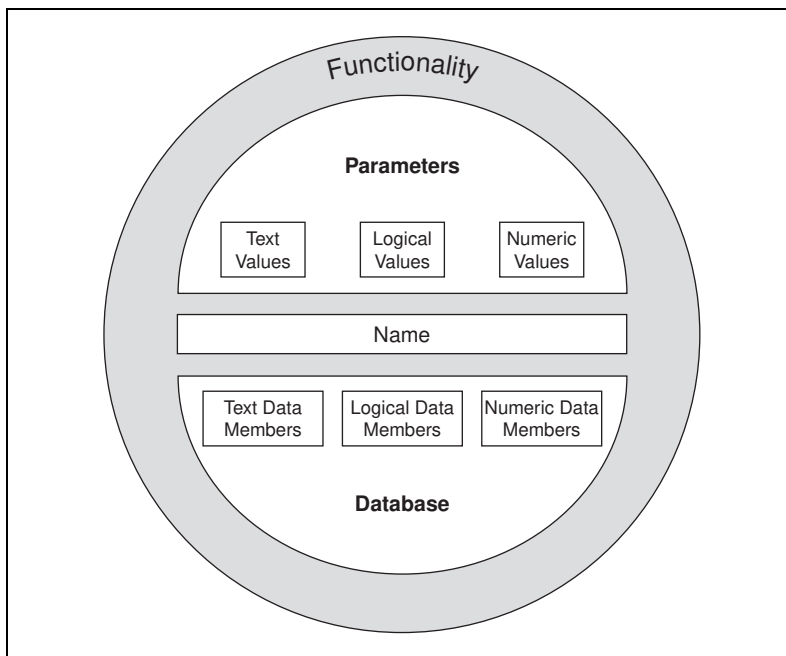
Figure 2-1. Objects Encapsulate Data Members, Parameters, and Functionality

Think of an object as a software model of something physical. For example, a potentiometer is something physical. You can adjust it up and down. In Lookout, a Pot object represents the physical potentiometer. You can adjust it, too.

Parameters define the limits of object functionality. For example, parameters set the minimum and maximum values of the Pot, the size of the smallest interval of adjustment, and other elements of the Pot's functionality.

The 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. The *database* can store data member information depending on what you want recorded, at what level of detail.

Lookout Object Classes and Functionality

A Lookout object is a specific instance of a Lookout object class. You can think of an object class as the generalized form for an object. When you create an object, you are taking the general form described by an object class, defining it with specific parameters, giving it a name, and putting it

to work as a software object. You can make as many objects as you need from each Lookout object class, each specifically configured to perform the task you need that particular object to perform.

For example, Lookout has both Pot and Switch object classes, from which you might create 20 pots and 30 switches. In this case, you would be creating a total of 50 objects using only two object classes.

Different object classes are designed to perform different functions, or tasks. For example, the Pot (potentiometer) object class operates differently from the Switch object class. This is the *functionality* built into every object class.

Global object classes are a special kind of object class. Each contains global system data such as the number of currently active alarms. You cannot create, modify or delete a global object, but you can use its data members just as you would use any other object data members.

When you create or open a Lookout process file, Lookout automatically creates three global objects: $Alarm, $Keyboard, and $System.

Functionality is the way an object class works, operates, or performs a task. Functionality is a general concept that applies in the same way to all objects in a given object class. Parameters, however, can be unique, and define the specific functionality of an individual object.

The object class definitions, found in the *Lookout Object Reference Manual*, outline the functionality of each object class.

Parameters

An object's parameters define its characteristics. Lookout uses object parameters to complete the definition of the object functionality. For example, **Data rate**, **Parity**, and **Stop bits** are a few of the parameters that define how a Modbus object works. Other examples include the **Control security level** of a Switch object; **Minimum**, **Maximum**, and **Resolution** of a Pot object; and **Data** of an Average object.

Every object class supports a set of parameters that you must fill in or select when creating a new object. Many parameters are *expressions*, which means you can change parameters programatically. Others require constant values. Some require you to pick specific settings.

Parameters that accept expressions appear as yellow data entry fields. These parameters can receive signals (that is, they are writable).

See Chapter 1, *Expressions*, of the *Lookout Developer's Manual* for more detailed information on expressions.

All the parameters for any given class are visible in the object definition dialog box. For information on how to create this object, see the *Creating Objects* section of Chapter 4, *Using Lookout*.
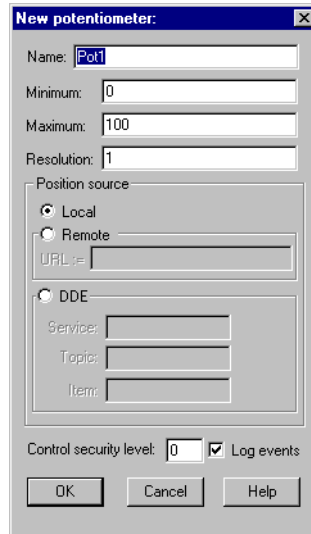
Figure 2-2. Pot Definition Dialog Box

Object Databases and Data Members

Each object has its own built-in database. The individual parts of this self-contained database are called *data members*. Some object classes have a very limited database, while others have extensive databases. The database of an object representing a PLC might have hundreds of data members; but a Switch object database has only six data members. You do not have to build a database—the data members are automatically available when you create the object.

In the case of a switch, the implicit value of the object is a part of the self-contained database. Data members can either generate (*write*) signals, receive (*read*) signals, or both.

Every data member contains a single value that can be one of three types: numeric, logical, or text. Lookout is not strongly typed, however, so you can connect one type data member to another, and Lookout will not generate an error message. See the *Data Polymorphism* section of this chapter for more information on how Lookout interprets data of different types.

The built-in data members for each object are referred to as *native* members, and can be thought of as the default or automatic data members. You can add to and modify the database to suit your specific needs, attaching one or more aliases to a given data member, each with a different set of associated alarm, logging, or scaling properties.

Every object class has an explanation of its database in its class documentation, which you can access by clicking on the Help button in the dialog box used to create or modify an object. The following is an example of the switch object class database explanation.

Table 2-1. Switch Data Members

Data Members	Type	Read	Write	Description
(implicit)	logical	yes	no	Switch Position
enable	logical	no	yes	If TRUE (the default), enables DDE. If FALSE, disables DDE. The default value is ON. This input is ignored for non-DDE Switch objects.
reset	logical	no	yes	While this value equals TRUE, the control will be set to the value in `resetvalue`.
resetvalue	numeric	no	yes	Sets the value a control will take when the reset data member transitions from FALSE to TRUE.
value	numeric	yes	yes	The current value of the control. If you have remoted this control, then `value` is the current value of the position source.
visible	logical	no	yes	When FALSE, the switch object cannot be seen on the display panel. When TRUE, the switch can be seen and controlled.

Logical Data Members

Logical data members contain a value that represents a binary or on/off state. A light switch is a logical device—it is either on or off.

Logical data members are typically used to control equipment that can be turned on and off, to indicate that a piece of equipment is running, or indicate whether a limit switch is open or closed.

The Switch object generates a logical signal that is on when the switch is up and off when the switch is down. In the same way, the Pushbutton object generates a logical value that is on while the pushbutton is depressed. The Pulse object generates a logical signal that turns on and off at prescribed intervals, creating a logical pulse.

The logical signals that some objects generate can be displayed graphically on a control panel. See Chapter 2, *Graphics,* of the *Lookout Developer's Manual,* for more information on graphical displays.

Lookout recognizes the following logical constants as expressions:

*   Logical constants that represent an on state: `yes`, `true`, `on`

*   Logical constants that represent an off state: `no`, `false`, `off`

Lookout is not case sensitive, so case variations such as On, ON, or oN will all be interpreted as ON.

See the *Data Polymorphism* section of this chapter 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 tank level, pressure, flow rate, voltage, and temperature. Numeric data members also represent time, either as a time period (span) or as an absolute time (that is, a particular time of day/week/month/year).

The Pot (potentiometer) object generates a numeric signal compatible with the numeric signals that monitor and control the analog input and output points on a PLC. Numeric signals range from -1.7×10^{275} to 1.7×10^{275}.

See the *Data Polymorphism* section of this chapter 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 as follows:

- 0
- –123.779999
- 1.5E7 (15,000,000)
- –3.7E-3 (–0.0037)
- –.0036
- 23356636.234579

Time or *Time signals* are stored by Lookout as numeric values that represent days and fractions of a day. For example, you enter one hour as 1:00:00. Lookout interprets the number to the right of the rightmost colon (:) as seconds, the number to the right of the second colon from the right as minutes, the next number as hours, and the number to the left of the third colon from the right as days. If there are no colons in the entry, the time period is assumed to be given in days.

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.

Table 2-2. 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

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. The number zero represents Jan. 1, 1900. You may also find it helpful to know that one second = 0.000011574 and one minute = 0.000694444.

Tip If you display a numeric signal on a control panel, Lookout provides a long list of numeric formats to choose from. You can set the format when you first create the display, or by right-clicking on the display and selecting **Display Properties**. The different possible formats are listed in the following tables.

Table 2-3. 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 2-4. 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 2-5. 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 2-6. 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 2-7. Hexadecimal Format Examples

Numeric Format	Value	Display
0x0	123.789	Displayed as 0x7
0x00	123.789	Displayed as 0x7
0x000	123.789	Displayed as 0x07
0x0000	123.789	Displayed as 0x007
0x00000	123.789	Displayed as 0x0007
0x000000	123.789	Displayed as 0x00007
0x0000000	123.789	Displayed as 0x000007
0x00000000	123.789	Displayed as 0x0000007

You can also use numeric signals to represent absolute times and periods of time. 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.

A time *period* represents a span of time or a duration. Time periods are indicated in hours, minutes, seconds, and fractions of seconds. Numeric formats that represent time periods are characterized by capital letters (that is, H rather than h).

Table 2-8. Time Period 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)

Table 2-8. Time Period Display Examples (Continued)

Time Format	Value	Display
HH:MM	0.4789	Displayed as 11:29 (11 hours, 29 minutes)
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 dates and times indicate a specific moment in time. Lookout stores all absolute dates and times as numeric signals. It 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.

Numeric formats that represent absolute times are characterized by lower case letters (for example, hh:mm instead of HH:MM).

Table 2-9. 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

Table 2-9. Absolute Date and Time Display Examples (Continued)

Time Format	Value	Display
dd/mm/yy	34668.7889	Displayed as 30/11/94
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

Text Data Members

Text data members contain text character strings. These character strings consist of all displayable characters. You can use text signals to display alarm descriptions on the alarm panel, to display labels on a control panel, and in parameters or expressions. You can enter text signals as constants, or you can construct them with the many text functions available in expressions. Be sure to enclose text constants within 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"

See the *Data Polymorphism* section in this chapter for information on how Lookout interprets logical or numeric data when it is passed to a text data member.

(implicit) Data Members

 Many object classes have an (implicit) data member. This implicit value is either logical, numeric, or textual, depending on the object class, and 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 class. In many cases, it is the only data member of a class. It saves you time and reduces the amount of typing required to designate a data member.

For example, Lookout could make you specify the numeric signal generated by a Pot object by typing `Pot1.numeric` where `Pot1` is the name and `numeric` is the current value of the Pot. Instead, you enter `Pot1` and Lookout knows you are referring to the implicit value of the Pot. If you

examine the Pot definition in the Lookout online help, you will see that the (implicit) data member is the current value of the Pot.

Data Polymorphism

Polymorphic data is data that is not strongly typed. In Lookout, data of one type is interpreted appropriately when connected to an input of another type. You can also select the data type when inserting expressions, using the selection box below the expression field, as shown in the following picture.
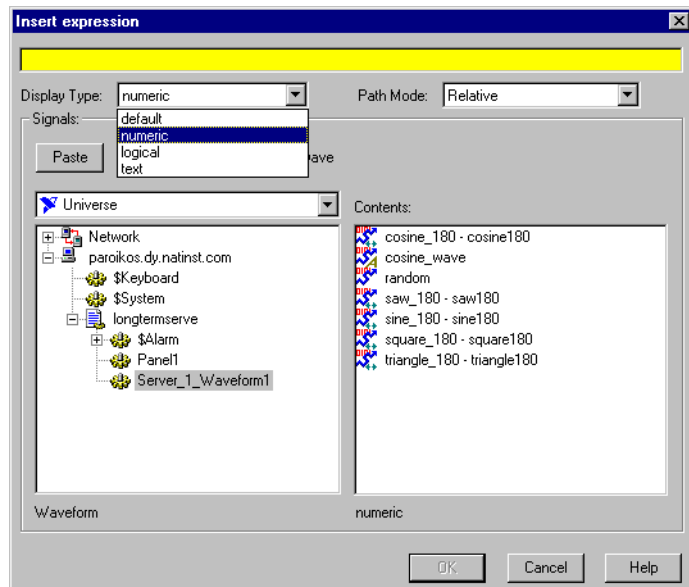
Logical values are displayed as 0 and 1 when interpreted as numeric values and as ON or OFF when interpreted as text values.

A numeric value of 0 displays as OFF when viewed as a logical value. Any value other than 0 is displayed as ON. A numeric value displayed as text shows the digits of the number.

Text strings displayed as numeric values 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. Numeric interpretation of text in a time format, such as 10:05:30 is interpreted as number (the fraction of one day represented by the time quantity) in scientific notation.

Any text string that does not consist of digits in a valid Lookout format is interpreted as a 0, with the exception of `ON` or `TRUE`, which display as 1. `OFF` or `FALSE` are interpreted as 0, by default. The terms `On`, `Off`, `True`, and `False` are not case sensitive.

Text strings displayed as logical values are interpreted as a 0 or as OFF, except when the text string consists of a `1`, `On`, or `True`, all of which are interpreted as ON. Again, a text string consisting of `0`, `Off`, or `False` is interpreted as OFF, by default. Again, the terms `On`, `Off`, `True`, and `False` are not case sensitive.

Once 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.

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. You can, however, 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.

Polymorphic data types affect the DataTable object. In versions of Lookout earlier than Lookout 4.0, the DataTable used data members such as `A1.logical` to set the type of data in a particular cell. Some Lookout developers used the DataTable to create, in effect, polymorphic data.

The Lookout 3.*xx* DataTable data members are preserved in Lookout 4 for the sake of compatibility. But it is not necessary to use the data-typed data members to simulate polymorphic data. To achieve this compatibility, however, Lookout DataTable data is still strongly typed. You must input numeric data to DataTable numeric data members, logical data to DataTable logical data members, and so on.

Data Quality Attributes

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.

Because the alarm may only be visible on the local computer, and not on a computer accessing that data from another location, Lookout places a red **X** over the affected control or expression when there is a data quality problem.

Check the Alarms panel of the process reporting the data quality problem for details on the problem. You can also use the data quality functions to monitor data quality and report problems. See the *Data Quality Functions* section of Chapter 1, *Expressions*, of the *Lookout Developer's Manual*.

You can also set your computer to monitor alarms from processes running on other computers. See Chapter 9, *Alarms*, in the *Lookout Developer's Manual* for information on how.

Connections

All the Lookout objects you create either take the place of physical objects such as switches or potentiometers, or serve as your interface to some physical object connected to your computer, such as a PLC, and RTU, or another computer on the network that is connected to such objects.

You can connect all these objects in Lookout, allowing signals to pass between them—much the same way you would wire a time delay relay to a motor starter relay, for instance. You can do this by connecting database members to each other, or by connecting database members to parameters.

In the broadest sense, one way to understand Lookout is in terms of these connections. The following figure shows a data member to parameter connection, and a data member to data member connection.
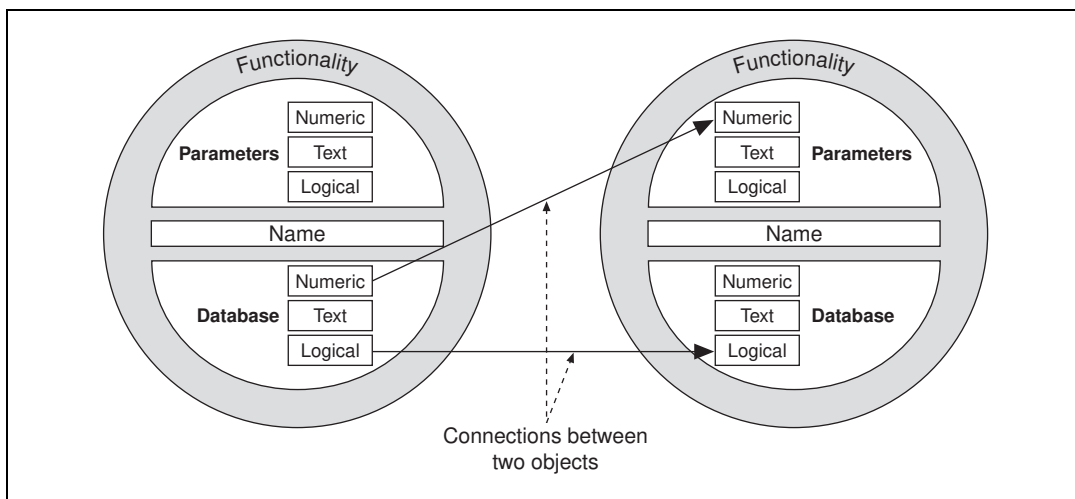

Figure 2-3. Example Connections Between Two Objects

For example, you might make the numeric data member of a Pot object the source for the **High Limit** parameter of an Alarm object. When you adjust the Pot, the Alarm **High Limit** changes.

In this way Lookout handles 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.

Alarming, report generation, trending, data analysis, and a fast and flexible database are all built in; all you do is configure objects and data points, then create the connections, using the Lookout dialog boxes.

Lookout connections vary. Some control hardware directly by determining settings, some control or report on processes interactively, some analyze and present data. To all this capability, Lookout adds a powerful yet simple way of making Lookout connections across a network.

Connections and objects taken together are called a *process*. Because Lookout runs multiple processes, and because networking has been made so simple, you can create as many or as few processes as you need, small or large, to increase the efficiency of your control systems.

Fortunately, you need not worry about creating drivers or databases or displays in Lookout—all these things are either Lookout objects, or contained in Lookout objects. All you have to do is place objects where you want them, and make the necessary connections to and from them.

You do not have to worry about elaborate timing systems and polling loops, either. Because Lookout is event driven, your connections report on and react to changes and other events as they happen, not in some arbitrary order built into your programs.

You can make the following kinds of connections in Lookout:

• Direct—Made with the **Edit Connections** dialog box, these connections are fixed and work in one direction only. This sort of connection should only be made from a server process to hardware installed on the server's computer, or between objects contained within the same process.

• URL remote—A URL remote connection is a flexible type of fully reciprocal connection between a Lookout control and a Lookout object read/write data member. Changing the control changes the data member, as you would expect, but if the value of that data member changes from some other cause, the control value changes as well.

This sort of connection is what you use to connect a client process to a server process, or to link two client processes through a server. (See the next section for an explanation of client and server processes.)

- Complex—You can make complex connections using variables and logic tests to respond dynamically to changing needs and circumstances. You can make direct complex connections, and with the Lookout Symbolic Link object, make remote complex connections as well.

Client and Server Processes

Lookout processes are programs you create to perform some specific function. 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.

- Computers running Lookout servers should be properly connected to external industrial automation hardware such as FieldPoint from National Instruments. Computers running clients only should not be connected to external hardware. In other words, servers connect directly to hardware; clients connect remotely, through the server.

- Servers control and monitor; clients watch, report, and make adjustments to server settings. Servers can do all things clients do; clients act through servers.

- Client applications can move from computer to computer, and so must have no direct dependencies in order to run. In other words, nothing in a client application should refer directly to or depend on the computer running it. You should always use a remote source connection from a client control to a server rather than making a direct connection, so that the control reports settings as well as changing them.

As you might expect, you can have more than one client process attempting to alter a value in a single server process. When this happens, Lookout accepts input as it arrives, so the first client input to arrive is executed before the next.

Supervisory Control

As you create and connect objects, you form a web, or system, containing many objects, all linked to perform a supervisory control strategy.

It is how you design your supervisory system, behind the control panel, that makes your process run. Your system routes signals from field components to bar graphs and visual indicators on control panels. It activates and deactivates alarms. You can design it to make complex decisions based on the values of field control signals and setpoints adjusted through pots and switches on control panels. You can include complex spreadsheet-style formulas as a part of your supervisory design. See Chapter 5, *Developer Tour*, for more detailed information on how to create objects and connect them together.

Event-Driven Processing

An important concept to understand is that Lookout is entirely *event-driven*, not *loop-driven*. To understand the significance of this design requires a digression, to explain how standard loop-driven programs work.

Loop-driven applications execute code sequentially from top to bottom, and loop back to the top to execute the same code over and over, as shown in Figure 2-4.
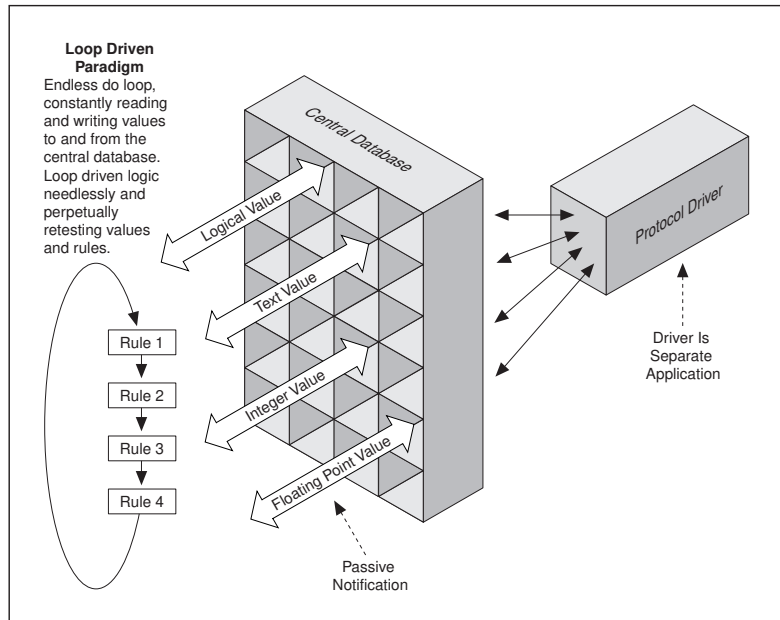

Figure 2-4. Example of Conventional, Loop-Driven Software Program

In this programming model, a given section of code or rule may execute millions of times before the result changes. This wastes computer processor time and slows down responses to frequent events.

The more rules you add to loop-based systems, the slower the response time. Also, as you add more names to the central database, speed and overall performance degrades. This is because many central databases use a *passive notification* system in which the rules of the loop-driven logic must scan an ever larger database for their appropriate values. The larger the database, the longer it takes the rule to find the data it needs to resolve its function.

In sharp contrast to this programming model, Lookout is entirely
event-driven. Each object remains quiescent, doing nothing until an event
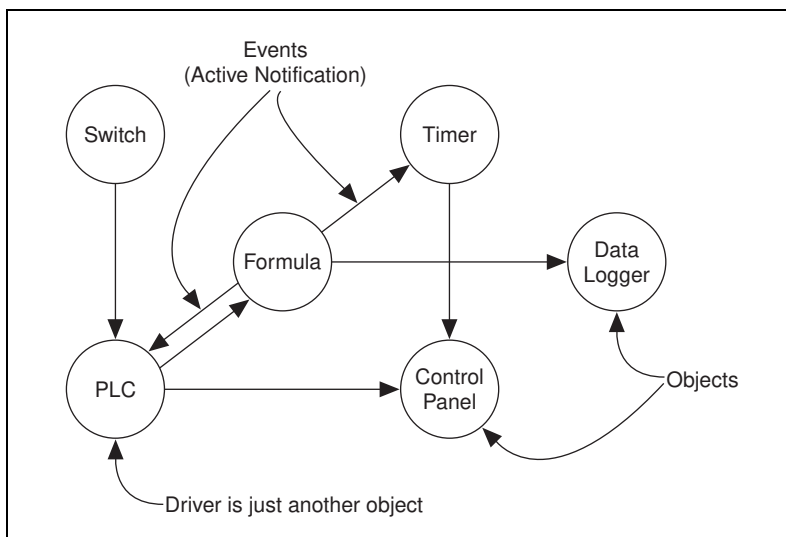occurs, as shown in Figure 2-5.

Figure 2-5. Lookout Object-Oriented and Event-Driven Architecture

An *event* is a change in a data value. When an incoming signal changes, the
object activates, processing the value according to its functionality. Objects
only send out signals when the result of their processing changes. This is
how an event propagates throughout your system, creating a chain-reaction
that affects only the objects in the chain. This is called *active notification*.

Individual objects activate only when notified of an event. This
event-driven paradigm closely simulates the responsiveness of physical
switches, pushbuttons, and relays, and is the reason Lookout is so fast.

Advantages of Active Notification

Consider the advantages of active notification over passive notification.
Active notification is inherently event-driven while passive notification
requires a constant do-loop to check for a change in a data value.

A good example would be two people trying to exchange information. One
person could ask the other for information, and that person would respond.
This would require one person to ask the other repeatedly if anything has
changed. This is what loop-driven rules do when they constantly query the
central database.

Alternatively, one person could just inform the other when something happens. This is what happens when you connect two objects in Lookout.

Adding objects in Lookout adds significantly less demand for processor time than adding new rules or enlarging the central database of a traditional loop-driven system.

Because Lookout is entirely event-driven, the order in which you create the objects and connect them does not affect how your supervisory strategy works.

Lookout Environment Services

While the cornerstone of object technology is the object itself, objects need an environment in which to function. Objects require the use of system resources like the serial port, hard disk, multimedia functions, and more. For example, multiple PLC objects may need to use the same communication port on your computer. In such a situation, Lookout must provide an *environment service*—that is, a mechanism the objects can use to gain access to the communication port in an orderly and timely fashion.

An environment service, then, is a tool that Lookout makes available to objects, or a function Lookout performs outside of its object-oriented structure. Each environment service provides a special function. The following sections describe Lookout environment services.

Serial Port Communication Environment Service

You can configure certain protocol object classes to represent and communicate with PLCs and RTUs through the serial ports of your computer. This environment service arbitrates serial port usage between objects representing PLCs and RTUs. For example, a single two-way radio connected to a serial port on the computer can communicate with several different brands of RTUs out in the field, each one using a different protocol. See Chapter 4, *Serial Communications*, in the *Lookout Developer's Manual* for more information on configuring communications.

Database Environment Service

With the database environment service you can define or modify native data member parameters.

For example, the Modbus object class includes a native data member called 40001. You can give this native data member an alias such as `PumpSpeed`, and define associated unit scaling, alarming, deadband, and other parameters.

With Lookout you can also directly import database information from external packages like Siemens APT. See Chapter 7, *Logging Data and Events*, and Chapter 8, *Structured Query Language*, of the *Lookout Developer's Manual* for more information on database services provided by Lookout.

Graphics Environment Service

Lookout has an extensive library of standard graphics. These include various switches, potentiometers, pushbuttons, bar graphs, valves, tanks, pumps, and so on. You can also create your own custom graphic and add it to your Lookout library. See Chapter 2, *Graphics*, of the *Lookout Developer's Manual* for more information on graphics.

Alarm Environment Service

The alarm subsystem is a powerful and flexible mechanism for generating, displaying, logging, and printing alarms. This subsystem has several distinct parts including the alarm window, object parameters like **Alarm Group** and **Alarm Priority**, alarm filters, display parameters, and print settings.

Lookout permanently archives alarms to disk. You can easily print this alarm history. See Chapter 9, *Alarms*, of the *Lookout Developer's Manual* for additional information.

Multimedia Environment Service

Lookout also provides a multimedia environment service you can use to play sound wave files. See the online help description of the Playwave object for more information on using this feature.

Security Environment Service

Lookout has a highly sophisticated and comprehensive security system for both local and network security. You can configure control security, viewing security, and action verification. You can selectively determine which operators have control of what objects, which operators can view what control panels, and which objects prompt operators for verification of commands. See Chapter 6, *Security*, of the *Lookout Developer's Manual* for more information on security.

Historical Logging Environment Service

With the logging environment service you can store real-time system information to disk in comma-delimited ASCII files, or in a special Lookout database called Citadel.

The Lookout Event Logger keeps track of who did what, and when they did it. Lookout logs operator commands, from closing a process file to flipping a switch or adjusting a Pot. 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. See Chapter 7, *Logging Data and Events*, of the *Lookout Developer's Manual* for additional information on logging environment services.

ODBC Environment Service

Because of the Lookout Open Database Connectivity (ODBC) environment service, you can use other applications, such as Microsoft Access, to query the Lookout historical database. See Chapter 8, *Structured Query Language*, of the *Lookout Developer's Manual* for additional information on ODBC environment services.

DDE Environment Service

Lookout can send its live process values to other applications, and it can receive real-time values from other applications. The Lookout system acts as both a DDE client and a DDE server. See Chapter 5, *Dynamic Data Exchange*, of the *Lookout Developer's Manual* for further information on DDE.

Networking Environment Service

Lookout provides a full client-server networking environment service through the use of TCP/IP. With this environment service, you can monitor and control your processes from multiple workstations on a network. See Chapter 4, *Networking*, of the *Lookout Developer's Manual* for information on networking Lookout computers.

Redundancy Environment Service

Use the Lookout redundancy environment service to configure two computers for redundancy, providing automatic transfer of monitoring and control should one of the computers fail. See Chapter 10, *Redundancy*, of the *Lookout Developer's Manual* for information on configuring computer redundancy.

Lookout 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 Citadel Server, Lookout Classified Ads, and Lookout Time Synchronization. In your Windows NT task manager, these services appear as *Classifieds*, *TimeService*, and *CitadelService.* Under Windows NT, these services run automatically as NT services. If you need to interact with these services, use the NT Services utility, found in **Start»Settings»Control Panel»Services**.

In Windows 98/95, Lookout installs a *services manager* during installation, denoted by a small lighthouse icon at one end of your Windows task bar, as shown in the following illustration.

When you right-click on this icon you will see the following menu.
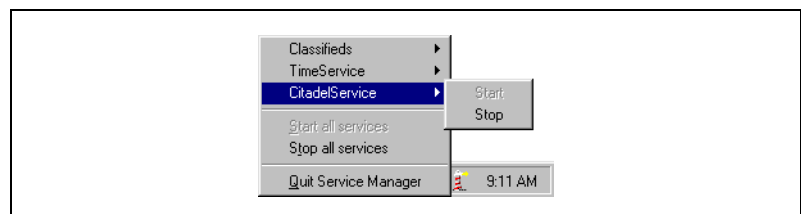
You can start or stop any of the Lookout Windows services using this tool.

3

Lookout Basics: Windows, Tools, and Files

This chapter explains how to start and get around within Lookout. It describes the Lookout screen and introduces some important mouse and keyboard shortcuts.

Starting Lookout

Logging on to Lookout

Note If you are using Lookout outside of the United States, you may be required to use a hardware key. Be sure to plug the key into the parallel port on your computer before activating Lookout. If you do not, the program will not run.

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

When you first install Lookout, the built-in Administrator account has no password. When the Administrator account has no password, Lookout opens without requiring any log-in. The Administrator is logged in automatically.

Note Server and Client run-time versions of Lookout open with the (nobody) user account logged in, no matter what the password setting for the Administrator account.

Once you provide an Administrator password, Lookout requests a log-in before opening with the following dialog box.

Welcome to Lookout

User name:	Administrator
Password:	
Domain:	Local
Idle time:	0 minutes

Log Off   OK   Cancel

Enter the appropriate password in the **Password** field and click on **OK**. Lookout is not case sensitive in this instance.

Note You set passwords with the Lookout User Manager. As long as the administrator password is empty (that is, if you do not insert one of your own), Lookout will always log on with the administrator as the user, and will not require anyone opening Lookout to log on. This can be a great convenience to you as a developer of Lookout processes, but be sure to set and administrator password before allowing others access to your development copy.

See Chapter 6, *Security*, of the *Lookout Developer's Manual* for more information about the Lookout User Manger, passwords, and Lookout security in general.

The **Idle time** field determines how long a period of inactivity Lookout will wait before logging off the current user. When set to 0, Lookout leaves the current user logged in until that user logs off, a new user logs on, or Lookout is shut down.

You can alter passwords and create or edit user accounts and groups with the user manager, accessed by selecting **Options»User Manager**. See Chapter 6, *Security*, in the *Lookout Developer's Manual* for more detailed information on the user manager and other security considerations.

When you first begin to develop your Lookout application, after closing any processes, you will see the following screen if no processes are running. In this figure Lookout does not have a process file open. No monitoring or control is taking place.
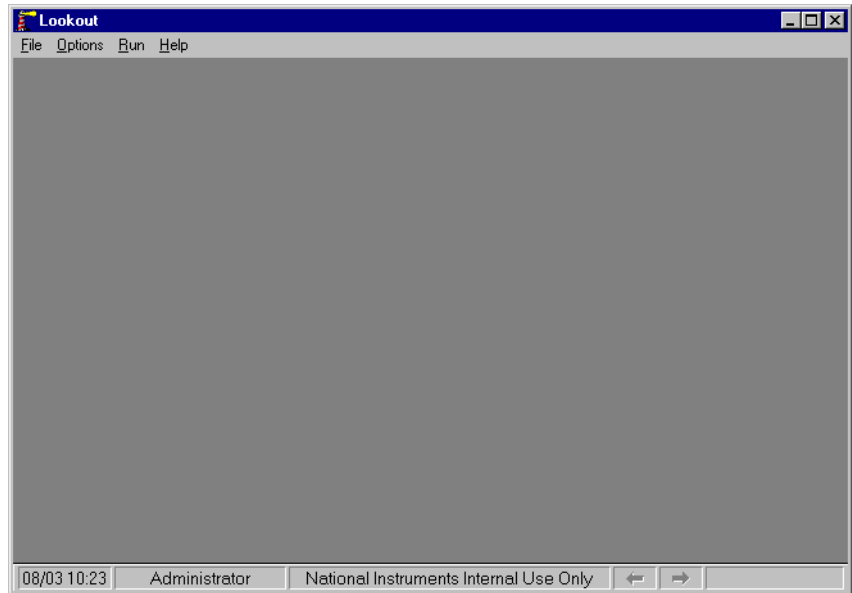
Opening a Process File

1. If you have a Lookout process file available, select the **File»Open** menu item from the Lookout menu bar. You can look in the Examples folder for sample processes to start with.

2. In the **File name** data field, enter the name of a Lookout process file (they have a .l4p file extension), or choose a process file from the list and select **OK**.

If you select a valid process file, Lookout opens the file and immediately begins executing the process.

When a process file opens, Lookout makes additional menu selections available and displays control panels and the alarm window.

You can open and run as many process files at one time as you want in Lookout, and close them individually or all together. While there is no theoretical upper limit to how many process files you can run on one computer, in practical terms you may begin to see degraded performance with a very large number of processes, depending on your computer's hardware resources and other variables. Running multiple instances of Lookout on one computer, or distributing Lookout processes on networked computers can increase performance.

The Lookout Screen

Lookout first appears on your screen maximized, taking up the entire screen.

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. The following illustration shows a Lookout screen in edit mode. (You can create your own version of this display in Chapter 5, *Developer Tour*.)

1	Menu Bar	4	Alarm Window	5	Status Bar	9	Lookout Workspace
2	Root Selection Box		(appearance depends	6	Object Location	10	Selected Display
3	Lookout Object		on Alarms»Display	7	Object Size		Element
	Explorer		Options setting)	8	Object Name	11	Process Control Panel
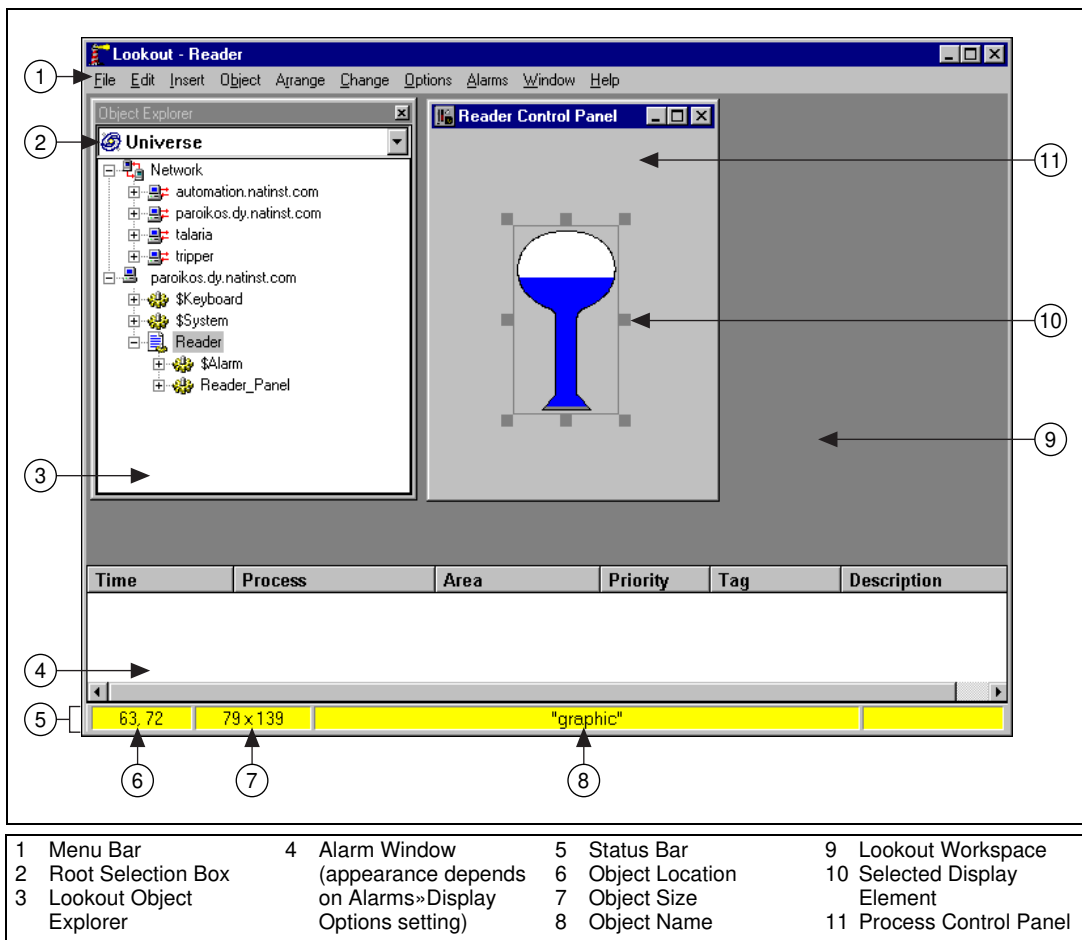
Figure 3-1. The Lookout Screen

Title Bar

The title bar at the top of the Lookout window displays the program name and the name of the currently active process. If a control panel is maximized, the control panel name appears in the title bar.

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

The status bar is at the bottom of the Lookout window, as shown in Figure 3-1. It changes when toggled into and out of edit mode. When not in edit mode, the status bar is gray and 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 may or may not be visible, depending on your system options settings.

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

When in edit mode the status bar turns 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, filename, 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
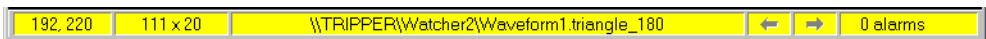
To toggle in and out of edit mode, select **Edit»Edit Mode**, or press <CTRL-Space>.

Lookout Workspace

The Lookout workspace is the area between the menu bar and the status bar. The workspace is the area in which you view and operate control panels. The alarm window also appears within the workspace. You organize and arrange control panels in this workspace area.

In Figure 3-1 the Lookout workspace contains the Lookout Object Explorer (visible only when you are in edit mode) and the Reader process control panel.

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 along the right side and bottom of the visible workspace. If scroll bars are visible, you can scroll around in the virtual workspace to see all of your control panels.

Object Explorer

The Lookout Object Explorer is a useful tool for creating and editing your Lookout processes. You can access the Lookout editing functions from this window by right-clicking on objects displayed there (options change depending on what item you click on). 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 Lookout object browser, make sure Lookout has a process running and is in edit mode, then select **Object»Object Explorer**. The initial window along with a somewhat expanded Object Explorer are shown in the following illustration.
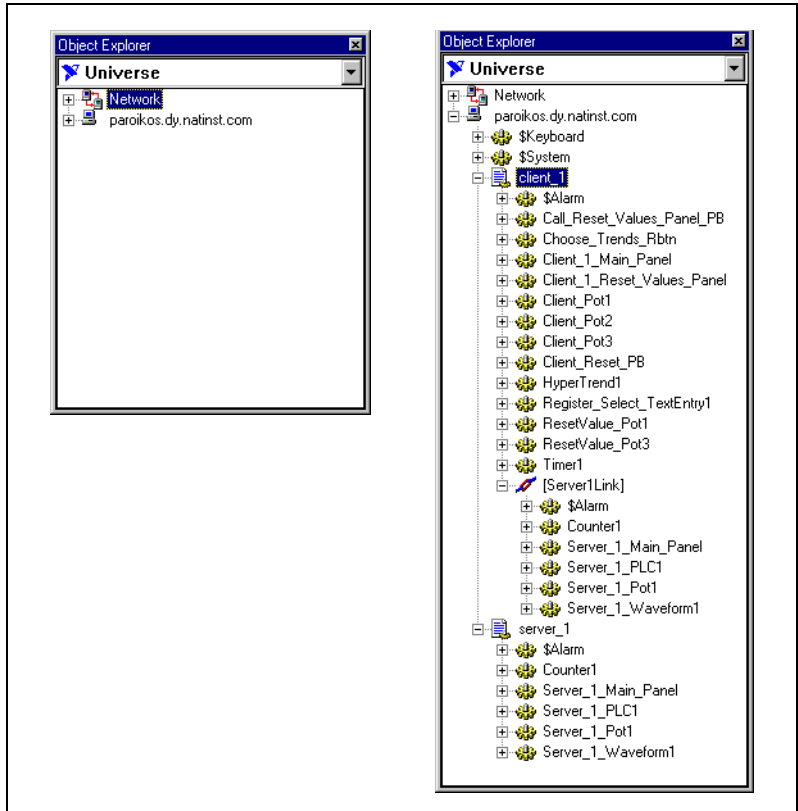
Right-clicking inside the explorer displays some variation of the following menu.
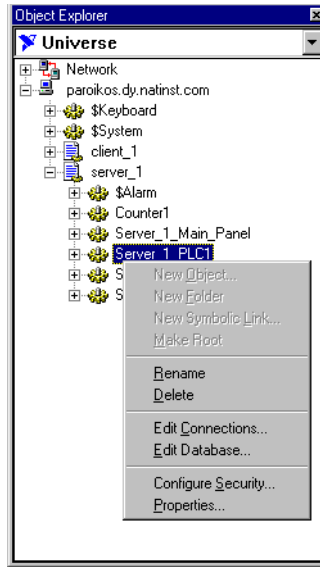
As you can see, depending on what you selected in the explorer before right-clicking, you can:

• create an object

• create a folder

• create a symbolic link

• change the root computer or process.

• modify an object or process by selecting **Properties**

• rename an object or process by selecting **Properties**

• delete an object or a folder

• edit object connections or an object database

• configure security permissions for a computer, process, object or collection of objects

When you create a displayable object in the Object Explorer, it appears in the explorer window but not on a control panel. To place the displayable part of an object on a control panel, you can drag and drop it. If there is no displayable component to the object, Lookout creates an expression (if possible).

If you drag and drop an object or data member from one process to a control panel in another process, Lookout inserts an expression to display the value represented. You cannot place the displayable part of an object on a control panel in a process other than the one in which the object was created.

When you leave edit mode, the Lookout Object Explorer window closes. Switching into edit mode restores the explorer window in the same location and at the same size it was when you left edit mode.

You can rename folders and relocate objects with the Lookout Object Explorer as well.

Lookout also has a complimentary tool called the Connection Browser, not shown above, that you can use to trace the connections you make between objects. See the *Connection Browser* section later in this chapter for more information on the Connection Browser.

Control Panels

Control panels provide the display area for any collection of switches, knobs, bar graphs, digital displays, trend graphs, and other components that you want to use to visually monitor and control your operations. Panels can be full-screen, minimized, or any size in between. 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. Control panels can also "pop up" when an event occurs such as when a pushbutton is pressed or when an alarm is activated. 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 some of the Lookout editing functions by right-clicking on 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. A pop-up control panel in either a pop-up state or minimized. When a pop-up control panel is displayed, it remains on top of all other panels until you minimize it.

Control panels do not have a standard appearance. You can customize each control panel for your particular control strategy with switches, potentiometers, and other graphical objects. Some control panels may only display information, while others combine control and display information. The information displayed always represents the most current values available to Lookout through its communications with your I/O devices.

With Lookout you have considerable latitude in how you make control panels look and operate. For example, you can draw your own switches and substitute them in place of the standard switches. You can easily make a pushbutton or switch work backwards from the way one might ordinarily expect. You might insert time delays and conditions behind the panel that may not be obvious just from viewing the panel.

Control panels are windows into your process you use to control equipment by flipping switches, pressing buttons and turning knobs. Unlike real switches and knobs, you can assign security levels to individual controls, as well as entire control panels within Lookout. You can use this feature to lock out operators that do not have high enough security levels for specific devices. See Chapter 6, *Security*, of the *Lookout Developer's Manual* for more information on security.

Unlike physical switches and knobs, you can display the same control object multiple times in a given process by using multiple displays, both on the same control panel and on different control panels. For example, when you turn a knob in one location, all copies of that knob turn at the same time, in each location of that control. The same control object may appear differently in each location. You might display a potentiometer as a horizontal slide on one panel, a pair of increment/decrement buttons on another panel, and a knob on yet another panel.

You can have the same sort of synchronous operation between controls in different processes running on your computer, or between controls running in processes on different computers in your network. Lookout networking makes this possible, using remote position sources and objects such as the Symbolic Link.

Alarm Window

Below the Lookout Object Explorer and the Reader Control Panel in Figure 3-1 is the alarm window. You can choose how the alarm window is displayed by selecting **Alarms»Display Options**. In Figure 3-1 the alarm window is set to display at the bottom of the workspace.

Operator Input and Navigation

There are numerous methods for plant operators to make setpoint adjustments or switch from one control panel to another. These include mouse, trackball, touchscreen, and keyboard activated commands.

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 on 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.

Virtual Keypad

When you are in operating mode, you can click on a digital Pot control and bring up a virtual numeric keypad to enter numeric values, either with a mouse or a touch-sensitive screen.

Virtual Keyboard

Lookout also has a virtual keyboard you can use with a touch-sensitive screen 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 touch-sensitive screen displays the virtual keyboard, as shown in the following illustration.
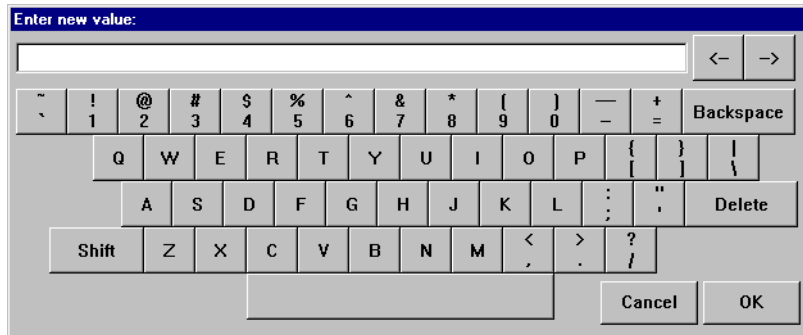
Connection Browser

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

To access the Connection Browser, select **Object»Connection Browser**. The following dialog box appears.
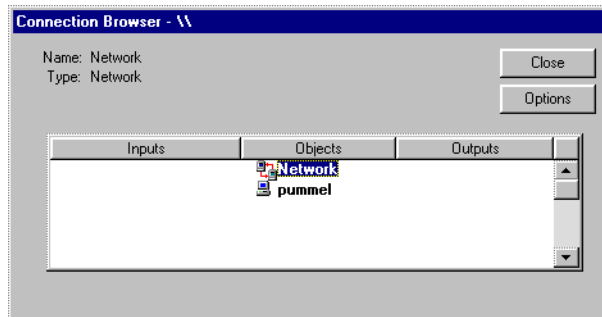
See the *Registering Computers* section in Chapter 4, *Networking*, in the *Lookout Developer's Manual* for information on using the Connection Browser to register networked computers in Lookout.

To create and modify objects as well as edit connections and the object database in the Connection Browser, right-click on the object you want to modify.

You can also create folders and move objects into and out of folders.

Some planning on how you want to arrange the objects in each process, combined with an awareness of how path relativity in Lookout works, can protect you from most inconveniences in this respect. See the *Path Names in Lookout* section in Chapter 4, *Using Lookout*, for more information on how a change in location can affect object connections.

To browse the connections in your Lookout processes, Lookout must be in edit mode. After opening the browser, click on your local computer name, and select the process you want to browse. You will then be presented with a list of objects in that process.

You can use the Connection Browser to create or rename folders inside your process, and you can 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 edit existing objects, connections, or the database through the Connection Browser. Right click in the Connection Browser to get the menu to choose one of these operations.
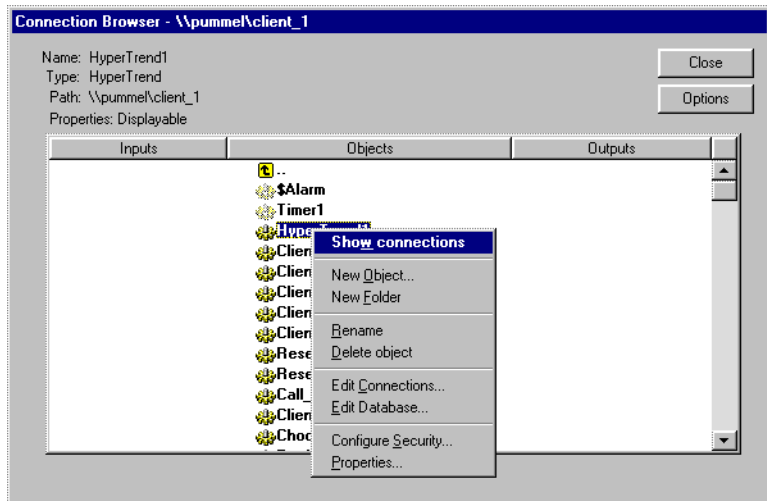
When you select an individual object to show connections, your browser screen looks something like the following illustration.
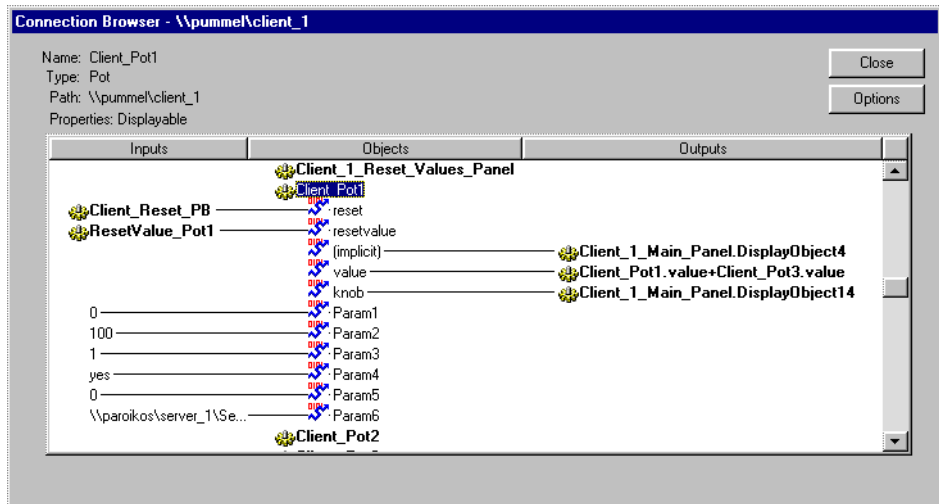
In this example, the Pot object named `Client_Pot1` has been selected. All the various display sites and connections are shown, including the way this Pot is remote connected to a data member in another process.

The `$Alarm` and `Timer1` object icons are faded in appearance. This indicates that this object is an orphan, meaning that its output is not being used by any other object.

By clicking on input or output objects you can trace chains of connections in your processes. By right clicking on an object or parameter, you can edit the object or parameter from the Connection Browser.

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 a network object on the screen, but cannot browse that object.

Lookout also has a complimentary tool called the Object Explorer that you can use to trace the connections you make between objects. See the *Object Explorer* section earlier in this chapter for more information on the Object Explorer.

Connection Browser Options

You can simplify the Connection Browser view by selecting what elements of an object are displayed. The Lookout default is for all settings to be enabled.

Click on the **Options** button to activate the Connection Browser **Options** dialog box, shown in the following illustration.
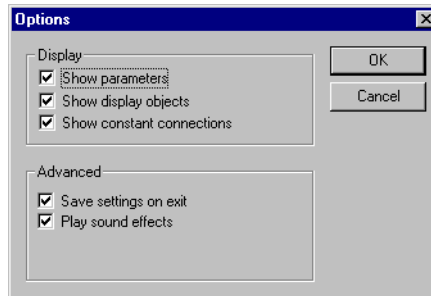
If you want to eliminate object parameters (in order to focus on data member connections) disable the **Show parameters** option.

Disabling the **Show display objects** 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.

Lookout Application Files

Using Lookout you can quickly develop a complete process file. All development is interactive. You do not have to program or compile. Just create objects and connect them as needed. The development cycle creates four unique files for every application: a process file, a source code file, a state file, and a security file.
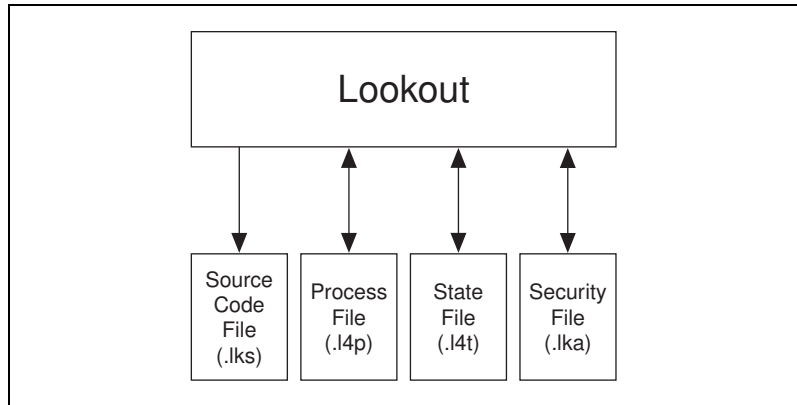
If you develop a Lookout process on one computer and want to run it on another computer, you must copy the process (.l4p) and security (.lka) files to the new computer for your process to run properly.

You can also copy the source and state files, if you want, but Lookout can recreate these files in the new location if necessary.

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 an .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. With this approach you can stay online while creating, updating, or editing your application. There is no need to shut down your process. Lookout does not cease operation while you add or delete controllers, I/O, graphics, or modify your logic. It is recommended that you save your process often while editing, so that you do not lose changes in the event of a power failure.

Source Code File

Lookout also automatically updates a corresponding *source code file*
when you invoke the **File»Save** command. Lookout can compile this file to
create an `.l4p` file, so it includes object definitions, names, I/O
configuration, communications, control logic, control panel layout and
other object parameters.

Source code files have an `.lks` (Lookout Source) file name extension.
These 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.

✎   **Note**   Because the binary format of Lookout process files both past and future is subject to
change for efficiency reasons, you should always keep your `.lks` source files backed up.
This is the file you will recompile to open a process created in an older version of Lookout
with any new version of Lookout released in the future.

You can recompile Lookout source files through the Open command, or
manually, using a DOS window.

To open and compile a source file inside Lookout, select **File»Open**. The
following dialog box opens.
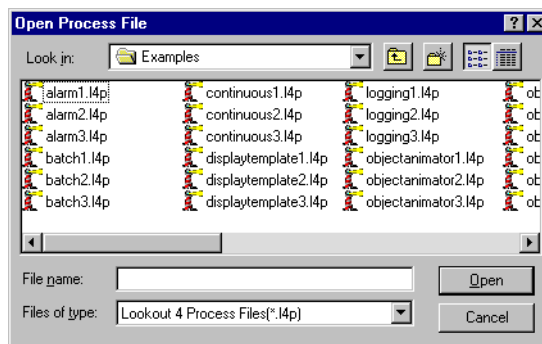
Set the **Files of type** to `Lookout Source Files (*.lks)` and select the
source file you want to compile.

To compile a lookout source file manually, consult the instructions in
Appendix B, *The CBL Compiler*, in the *Lookout Developer's Manual*.

State File

The *state file* 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 Pots, switches, trend lines, and so on should be in upon startup.

The state file updates any time you save, close or exit the Lookout application file. It can also update on a periodic basis as defined in the process properties dialog box or the Loader object. You invoke this dialog box by selecting the **File»Modify Process** menu command. State files have an `.l4t` (Lookout State) file name extension.

Security File

Every Lookout process has an attached security file, identified by the `.lka` extension. All the 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 security settings to work. If you misplace the `.lka` file, all users will have complete access to all parts of the process. To reconstruct the file, you would have to reset all your security permissions.

See Chapter 6, *Security*, of the *Lookout Developer's Manual* for more detailed information on Lookout security.

Configuration Shortcuts

Mouse Shortcuts

This manual references a number of mouse-implemented shortcuts. Becoming familiar with mouse commands is easy and pays significant dividends during application development.

Function	Mouse Action
select object display, graphic, text, expression, or any item on a display panel	**Click**: Select the item by pointing the cursor at the item and clicking the left mouse button. (*See Note*.)
create a new object from an existing object	**CTRL-drag**: Select the item(s) you want more of. Hold the CTRL key down and drag the selected item(s). This makes a completely new instance of the control: <CTRL>-dragging Pot1 will produce a new object named Pot2.
copy object display, graphic, text, expression, or any item on a display panel	**Shift-drag**: Select the item(s) to be copied. Hold the <Shift> key down and click the left mouse button while dragging the item(s). This makes a duplicate of the selected object that is linked to the original. Both the original and the duplicate have the same name.
modify display or object parameters of an object, graphic, text, expression, or any item on a display panel	**Right-click**: Select the item(s) to modify. Position the cursor over the item(s) and click the right mouse button.

Note   You can select multiple items on a display panel by clicking in a panel and dragging the box outline around the objects you want to work with. Subsequent commands affect the entire group at once or toggle from one item to the next, prompting you for new instructions. You can stop this process at any time by holding the <Shift> key down and selecting either the **OK** or **Cancel** buttons in the current dialog box.

Remembering Object Names

Many dialog boxes contain *expression fields*. Expression fields are yellow to indicate that you can get help identifying or remembering an object name and associated data members by pressing the right mouse button (right-clicking on the selected expression field).

Assume, for example, that you want to average some value. You start to create an expression for this value, but you cannot remember the name of the object or data member you want to have an average value for. Right-click on the expression field (the field in which you normally type the a data member you want to connect or display).

When you right-click any expression field, Lookout presents the **Revise expression** dialog box, as shown in the following diagram.
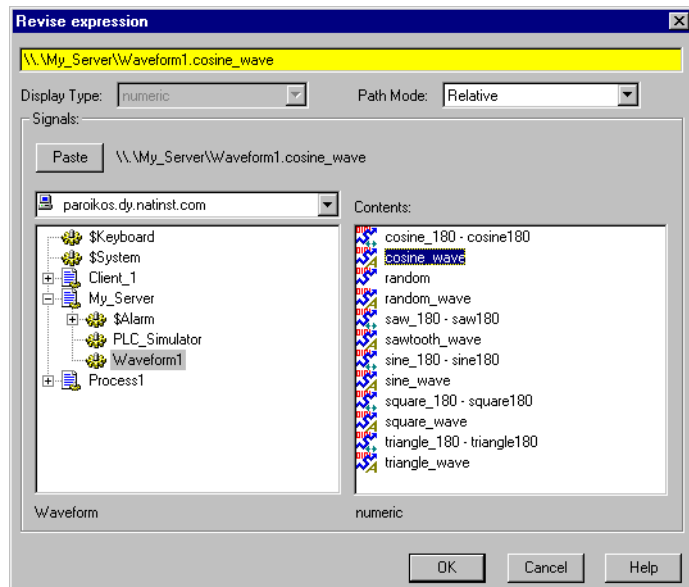
Figure 3-2. Expression Editor Activated by Right-Clicking on a Yellow Data Field

Using the expression editor, you can easily paste object names and their data members directly into the expression field. Navigate through processes and objects until you have selected the object and data member you want. Paste them into the expression editor data field, either by clicking on the **Paste** button or double-clicking on the data member. Click on **OK**, and Lookout writes your expression into the expression data field of the object, such as the **Data** field in the **Create Average Object** dialog box shown in the following illustration.

4
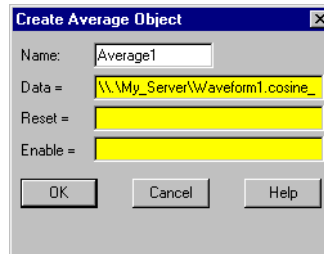
Using Lookout

This chapter discusses the basics of using Lookout, including selecting objects, creating objects, editing object databases, and other operations. In Chapter 5, *Developer Tour*, you can work though an exercise that uses the features discussed in principle in this chapter.

If you have not read the discussion of Lookout architecture in Chapter 2, *How Lookout Works*, and the discussion of basics in Chapter 3, *Lookout Basics: Windows, Tools, and Files*, you should go back and read those chapters. This chapter builds on information presented in the earlier chapters, and assumes that you have absorbed that information.

The keys to using Lookout are objects—their parameters and data members. The technique for using Lookout involves solving problems by selecting objects and making connections.

Development Process

The first step in developing a process file is creating a process. Selecting **File»New** opens a new Lookout process. After you name your new process, a dialog box opens for you to use to create a control panel for the process.

Control panels are windows you use to place other objects, such as switches, Pots, and trends to be displayed. You can make them look like physical control panels, complete with switches, pushbuttons, bezels, insets, lamps, gauges, and so on. Your imagination is the only limit to how the control panels look and function.

After you create at least one control panel, you can then create and display any number of other objects. Remember that objects can be displayed on a single control panel or on multiple control panels. For example, you can insert the same switch on several panels. If you flip the switch on one panel it immediately flips on all the other panels.

The first objects you might want to create could be those that represent your PLCs and RTUs. These type of object classes often use your computer communication ports, so you may have to configure your ports for radio, dial-up modem, or hard-wired connection, as appropriate. See Chapter 3,

Serial Communications, in the *Lookout Developer's Manual* for information on configuring communications.

When you create an object that represents a PLC, RTU, or other I/O device, its associated functionality and database are part of the object. The functionality built into this type of object class includes a protocol driver, enabling the object to talk with the physical device. The native (default) database includes all registers, bits, I/O signals, parameters and other values that the physical device can transmit to or receive from Lookout.

At this point, you may want to add data members to the native databases for your objects. Such new data members, called aliases, can include descriptive names, signal scaling parameters, alarming parameters, and so on. You can create more than one alias for a given native data member, using different scaling or alarm parameters for each.

As you create objects, you can connect them to other objects. The type of objects you create (such as Pot, Switch, Modbus, Trend, and so on) and the manner in which you connect them determine how your system interacts with your process.

Many object classes you use take advantage of Lookout services. For example, you can assign a security level to a Pot object, and the Alarm object class uses the Alarm processing subsystem. As you create objects that take advantage of these services, you should configure the services to custom fit your application requirements.

Invoke the **File»Save** command to save your process to disk. Every time you save, Lookout updates the `.l4p`, `.lka`, `.lks` and `.l4t` files.

Tip   You can save a process file with any name you choose. Your process file name does not have to be identical to your process name. You can copy (or Save As) Lookout files to retitle them and run the process just as you would run the original files. You cannot, however, run two processes with the same process name in a single instance of Lookout. If you have a basic process that you need to make several variations on, you can rename the process in each file by right-clicking on the process name in the Lookout Object Explorer and selecting **Rename**.

You should save often while working and backing up your work to a separate disk, regularly, to avoid inadvertent loss of work. Remember that the `.lks` file is the primary source file for your Lookout processes. This file in particular needs to be backed up for safety's sake.

Building Client and Server Processes

As explained in the *Client and Server Processes* section of Chapter 2, *How Lookout Works*, Lookout server processes are run on computers with physical connections to your hardware. Lookout clients run on computers that can communicate with the computers running Lookout servers.

You build these clients and servers by choosing objects and making connections.

Choosing Objects

How do you decide what objects to use and what connections to make?

Some decisions are easy—you will certainly use driver objects for the different PLCs, RTUs and other devices you need to monitor and control. Other decisions are driven by what you want to do.

To solve problems with Lookout, ask yourself which object does what you need done.

Lookout has many object classes with obvious uses, such as pushbuttons, switches, and trend displays. Other Lookout object classes may not be as immediately obvious as to how they might be used. To familiarize yourself with Lookout object classes, you can browse through the online help *Lookout Object Reference* topic, or the PDF *Lookout Object Reference Manual* that was installed along with Lookout.

The **Select Object Class** dialog box that appears when you create a new object has short descriptions of each object class. Browsing through these descriptions is another good way to acquaint yourself with Lookout functionality. You can find a list of these descriptions in Appendix A, *Lookout Object Descriptions*.

After you become familiar with the Lookout object classes, you have a better idea of which object to create—or which object classes to investigate—to solve the problem you are facing at the moment.

Lookout Navigation

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 dialog box. This dialog box is used to locate any object in any Lookout process running on your network.

Each computer registered as running Lookout appears in this dialog box. Clicking on the computer to open the full display reveals the processes currently running on that computer. Clicking on a particular process reveals all the objects in that process that you can access.

The pull-down combo box at the top of such displays indicates the root location. To display all the computers registered as running Lookout as well as displaying your own computer as local, you 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 on that node and selecting **Make Root**.

✒ **Note** Lookout cannot edit processes over the network, so what you can do with computers and processes running under the Network node is limited. 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.

See Chapter 4, *Networking*, in the *Lookout Developer's Manual* for more information on registering computers and navigating through Lookout processes on your network. Registering a computer is demonstrated in the exercises in Chapter 5, *Developer Tour*, as are practical elements of navigating in Lookout.

Creating A Lookout Process

To create a Lookout process, select **File»New**. The following dialog box appears.

Enter a Lookout process name in the **Process Name** field. This name must follow the same conventions as a Lookout object name. See the *Object Names* section for detailed information on Lookout naming conventions.

As you may recall from the *Lookout Application Files* section of Chapter 3, *Lookout Basics: Windows, Tools, and Files*, the Lookout state file is what Lookout uses to initialize certain values when it opens a process. Lookout state files use a .l4t file extension.

The **State Information** section is where you specify the locations you want Lookout to save state files for this process. If you are going to have a backup computer in your network that will come online and run this process in the event that your primary computer crashes, you should include a location on that computer as one place to save the process

state file. See Chapter 10, *Redundancy*, of the *Lookout Developer's Manual* for more information about setting up backup computers and processes.

Select **Save State File with Process File** to save the state file in the location where the process file was opened. This is the default behavior.

Select **Save State File in Lookout Folder** to save 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.

Select the **Save Standby State File** checkbox to save 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.

To set the frequency that Lookout saves the state file, check the **Save state files** checkbox and fill a value for the number of minutes you want to pass between each save. 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.

The **Citadel Database** section of the **Create Process** dialog box sets the location of the Citadel database Lookout logs data to for this process. If you check the **Use default values** checkbox, 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.

To designate a specific computer and path for your process to log data to, enter the fully qualified network name for the target computer in the **Citadel database computername** field, and the complete path to the target database directory in the **Citadel database folder** field.

State Information and **Citadel Database** settings you make when creating a process override any settings you may have made in the **System Options** dialog box.

Creating Objects

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 on the process you want to create an object in and select **New Object**. For this example, select the **Pot** class of objects, found in the **Control** category.

   • Select the **Object»Create** command from the Lookout menu bar. For this example, select the **Pot** class of objects, found in the **Control** category.

3. The **Select object class** dialog box appears, as shown in the following illustration.
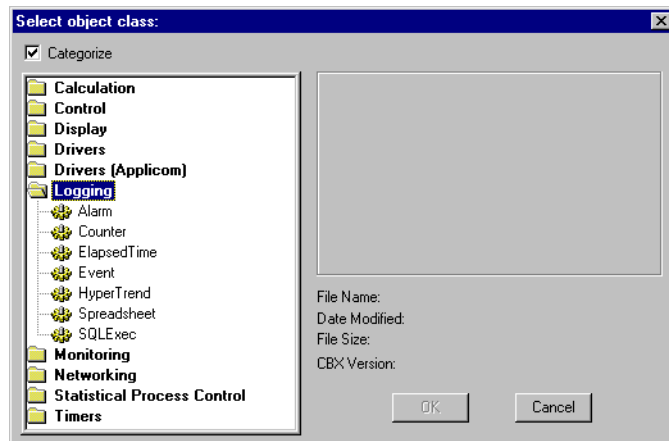
Figure 4-1. Select Object Class Dialog Box

The first time you create a Lookout object, the **Select object class** dialog box appears with the **Categorize** option checked. Some object classes are contained in more than one category, depending on their versatility.

If a category is open, you can jump to the object class you want by type in the first few letters. To have access to all object classes with this

technique, disable the **Categorize** option by clicking the **Categorize** checkbox.

When you select an object class, a dialog box you use to define the parameters for a specific object appears, as shown in the following example for creating a Pot object.
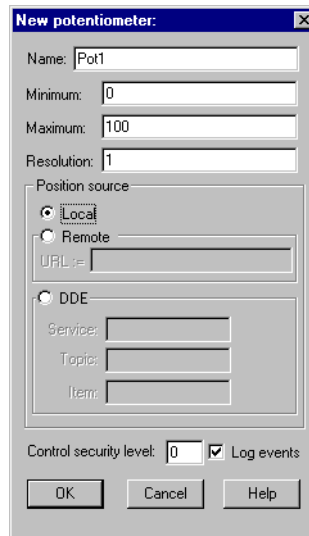
Figure 4-2. Object Definition Dialog Box

4. Assign a unique name to the object. Valid names are described in the next section, *Object Names*.

5. Define the object parameters. Each object class has unique parameters, described in the Lookout help file or the *Lookout Object Reference Manual*.

    Ordinary parameters that can only be set in the object definition dialog box are indicated by white fields.

    Some parameters, however, are *expressions*, 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, which are roughly the equivalent of spreadsheet formulas. See the *Connecting Objects* section of this chapter for a basic explanation of using expressions, and Chapter 2, *Expressions*, of the *Lookout Developer's Manual* for detailed information on expressions and expression functions. The tutorial exercise in Chapter 5, *Developer Tour*, also demonstrates using expressions.

6. Leave **Position source** set to **Local**.

The **Position source** determines what value your control takes when your process first opens, and also affects the way the control operates.

Selecting **Local** configures your control to take its value from the state file when the process opens. The value of the control will not change unless you adjust it manually or programmatically using a direct connection to the control's `value` data member.

Selecting **Remote** configures your control to take its initial value from a read/write data member of some other Lookout object. To accomplish this you enter a URL to the data member, wherever its host process may be running. This connection is reciprocal; adjusting the control changes the value of the data member, and any change in the data member from some other cause adjusts the value of the control. See the *Remote Position Source Connections* section of this chapter for more information on remoting controls.

Note   The remote position source has changed in Lookout 4.0. If you are accustomed to developing processes in earlier versions of Lookout, make sure to read the *Remote Position Source Connections* section of this chapter in order to understand the changes and make the best use of this feature.

You can select **DDE** as your position source if you want to use DDE networking or if you need to connect your control to an application other than Lookout, such as Microsoft Excel. See Chapter 5, *Dynamic Data Exchange*, of the *Lookout Developer's Manual* for detailed information on using this feature.

DDE networking was used in versions of Lookout earlier than Lookout 4.0. In general, the TCP/IP networking built into Lookout 4.0 and greater is more efficient and easier than DDE networking. If you are maintaining an old Lookout process, or have some special need to use DDE networking, you can find detailed information on this function in Appendix A, *Networking with DDE*, of the *Lookout Developer's Manual*.

7. Select **OK** to create the new object.

Many object classes do not have components (other than a data member value) that you can display on a control panel, such as a protocol driver for a typical PLC, or an object class such as Neutralzone or Pulse (both Lookout object classes with special functionality).

Some objects, like Pots, can be displayed on a control panel. When you create such an object, Lookout presents you with a dialog box you use to define how the object is displayed on the currently active panel (if there is one).

8. If you created your object using the Lookout Object Explorer, it will appear in the Object Explorer window but not on any control panel. To display the pot, you will have to drag and drop it to the control panel you want it to appear on. If you created your object using the menu, the display properties dialog box appears as soon as you click on **OK** to finish defining the object.
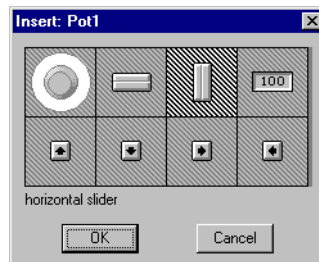
Figure 4-3. Pot Display Dialog Box

9. In the display dialog box, choose how you want the object to be represented and select **OK**.

10. Position the object on the control panel and adjust the size.

Note   You can copy objects to paste onto another Lookout panel in the same process, but if the panels are of different sizes, the pasted objects may not appear in the proper location on the new panel. In general it is easier just to drag the object to the new panel from the Lookout Object Explorer.

Object Names

A name is how you identify a specific object. You must assign each object a unique name. 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 thirty two (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 class 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.

Some example of *valid* names are as follows:

- Pump3_HOA

- HighServicePump3_Run

- ClearWellLvl_at_Hiway289_and_I35

- ShaftTempF

- Pushbutton1

- runtime

- Sw1

- Sw2

Some examples of *invalid* names are as follows:

- 3Pump_HOA, which begins with a number

- _HighServicePump3_Run, which begins with an underscore

- ClearWellLvl at Hiway289 and I35, which contains spaces

- #ShaftTempF, which begins with #

- Pushbutton, which is a reserved word

- ThisnameIsWellOverThirtyTwoCharacters, which is too long

- and, which is a reserved word

Editing Object Databases

Lookout creates the *native* database of an object automatically when you create the object. The native database is documented at the end of each object class definition in the online help and in the *Lookout Object Reference Manual*.

You can create new data members through aliasing, or modify the parameters of any existing native data member. These parameters include such things as alarm setpoints, deviation filters, scaling factors, historical logging, and alias names.

✐ **Note**   Any object in Lookout can have its native database modified. However, this is most practical for objects with large native databases, such as driver objects and DataTable objects.

Editing Database Parameters

1. For the purposes of this illustration, create a Modbus object, the same way you created the a Pot object. Use the name PLC_1 for your object, and accept the default parameters.

    The Modbus object can be added to your system without a physical connection while you are learning to use Lookout. Let the default settings stand when you create the object.

2. There are two ways to access the dialog box you use to edit a database.

    •   From the Lookout menu bar, select **Object»Edit Database**. In the **Edit object database** dialog box, navigate to your demo process running on your local computer and select PLC_1.
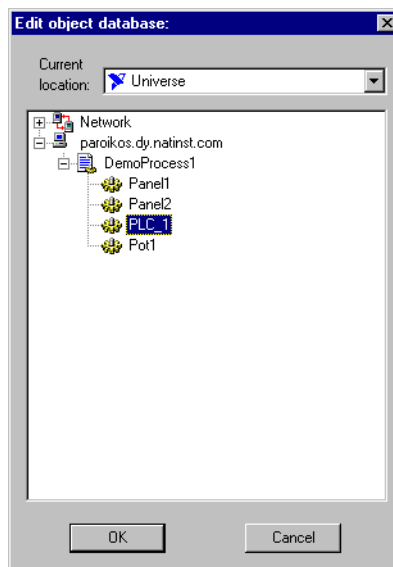
    •   In the Lookout Object Explorer, right-click on the object you want to edit the database for, and select **Edit Database**.

The following dialog box appears.
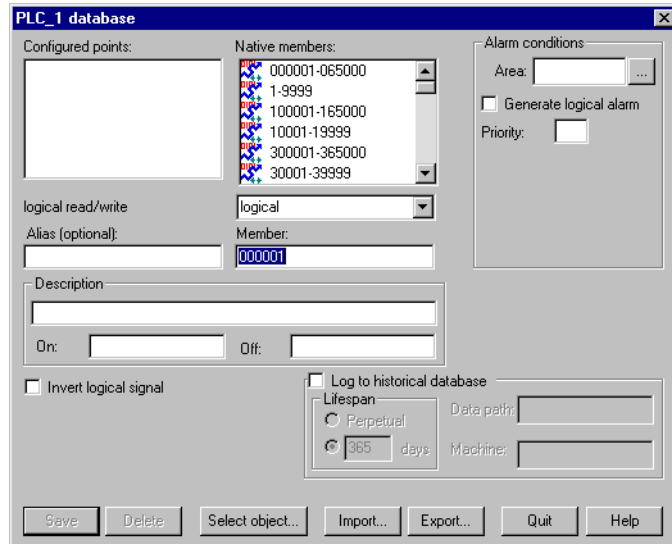
3. Follow these steps for each data member to be configured:

   a. Identify the desired data member by entering it into the **Member** data field. If you are modifying a data member that has been previously configured, you can select it from the **Configured points** list box.

      For this example, select 40011 as the modbus data member. Enter FlowRate as the alias.

✏️ **Note** Parameter fields automatically change depending on the data member you select. Lookout automatically determines whether the data member is logical or numeric, and presents you with the appropriate parameter attributes.

   b. Configure appropriate parameters shown in the following illustration. See the individual parameter sections in this chapter for details on setting each parameter.

c. Select **Save** or **Update**. (If you are modifying a data member that was previously defined, the **Save** button changes to an **Update** button.)

Lookout stores all the new parameter settings for the specified data member when you select **Save** or **Update**. In addition, Lookout adds the modified data member to the **Configured points** list box for future reference. Lookout immediately reflects these changes throughout your configuration.

4. Select **Quit** to exit the dialog box.

Numeric Member Parameters

The following diagram and paragraphs describe numeric data member parameters. Logical data members, covered in the following section, share a number of the same parameters.
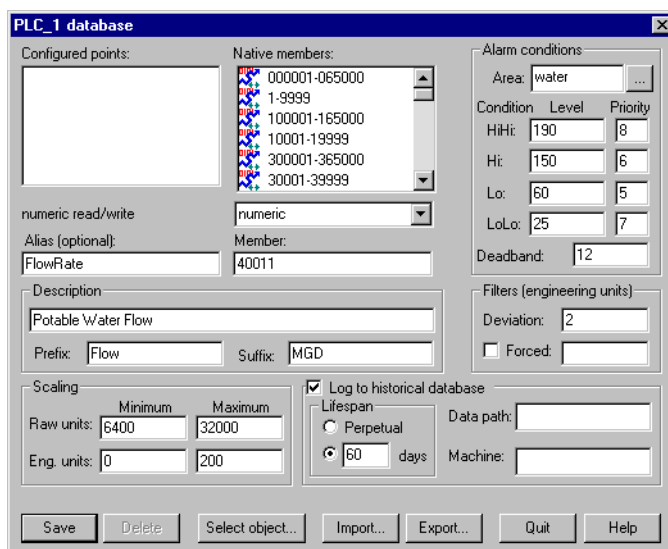
Alias renames any native data member. You can think of an alias as a sort of nickname. For example, the Modbus driver object includes the native data member `40011` that represents an analog input. You can give this native data member an alias like `FlowRate`. From then on, you can reference the alias `FlowRate` instead of its native name `40011`. All associated parameters (such as **Scaling**) are also applied to the alias value.

An alias is a good way to insulate your Lookout configuration from changes in your PLC, RTU, or I/O configuration. For example, consider a flow transmitter wired to an analog input at `40011`. You can give `40011` the alias name `FlowRate`, just as you did in the example. Multiple control panels can then display the `FlowRate` data member and numerous other objects in Lookout can use it. If you later rewire the transmitter to the analog input at `40012`, you need only modify the alias `FlowRate` to reflect the new I/O address. Lookout instantly reflects this change everywhere `FlowRate` is used.

You can modify all associated parameters of an existing alias *except* the alias name itself. 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.

Most developers implement aliases on objects with large native databases, such as driver objects (like Modbus and Tiway) and DataTable objects.

📝 **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.

The **Description** appears as the message text in the alarm window. It can have spaces, and it can be lengthy. You do not have to enter quotes in this field.

Prefix and **Suffix** are part of the description, but do not appear in the alarm window. They are just additional descriptive text.

Define **Scaling** by entering **Raw units** and **Eng. units**. The raw numeric data member is converted (scaled) to an engineering unit value. The PLC in this example generates a raw value ranging from 6,400 to 32,000. Lookout converts that raw signal to range from 0 mgd to 200 mgd. The conversion is linear. See 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 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.
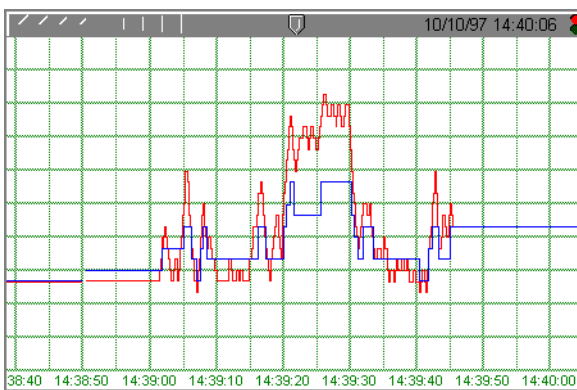
The Citadel database also uses **Deviation** as the criteria which triggers logging of new historical data to disk. See Chapter 7, *Logging Data and Events*, in the *Lookout Developer's Manual* for more information on logging data.

Use the **Forced** data 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.

Use the **Alarm condition** 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.

You can assign an alias or native data member to any existing alarm **Area** or you can create a new **Area**. To create a new **Area**, enter the new area name in the field. See Chapter 9, *Alarms*, in the *Lookout Developer's Manual* for more information on alarms.

Use the alarm **Deadband** parameter to prevent fluttering between alarm and normal states when the signal value hovers near an alarm limit. The following figure shows a value plotted against its **Hi** and **Lo** alarm setpoints.
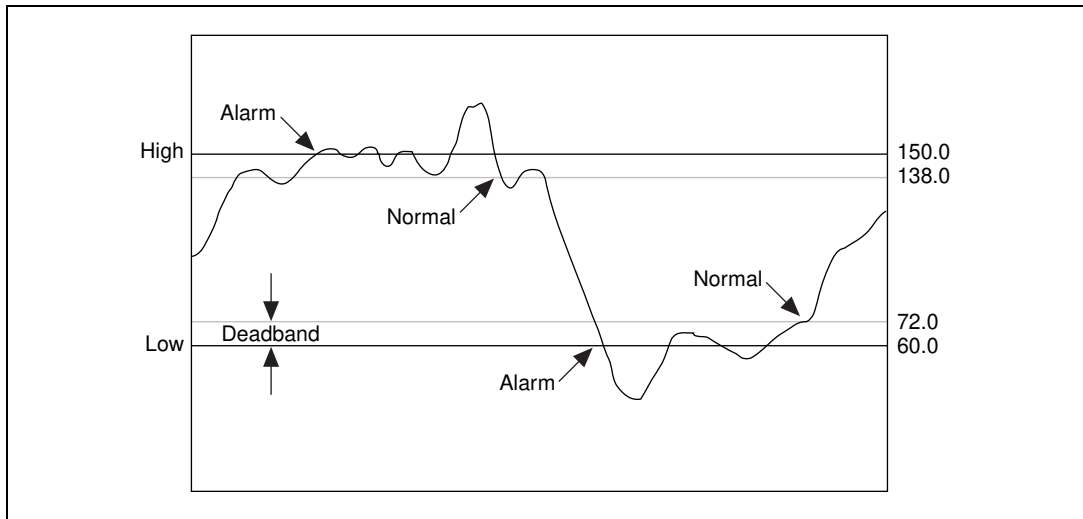
Figure 4-4. Alarm Processing of Data Member Whose Alarm Setpoints are
Shown and Whose Deadband is 12

Lookout generates an alarm the moment the 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, or goes above the low alarm
setpoint plus the Deadband. The Deadband also applies to all **HiHi** and
LoLo alarm limit setpoints.

The **Log to historical database** parameters define how long to store a
value in the Citadel database on your hard drive. If you do not select this
option, Lookout does not log the value to your disk. If any scaling or
filtering parameters are defined, Lookout logs the scaled, filtered value
(that is, the engineering unit value).

💡 **Tip**   If you intend to display data with a Lookout Hypertrend object, you must log the data
to the historical database.

When you log data, it will go to the database directory you set up in the
Create Process dialog box.

See Chapter 7, *Logging Data and Events*, in the *Lookout Developer's
Manual* for more information on logging data.

Logical Member Parameters

Some of the parameters of logical data members are different from those of 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 database is shown being configured in the following figure.
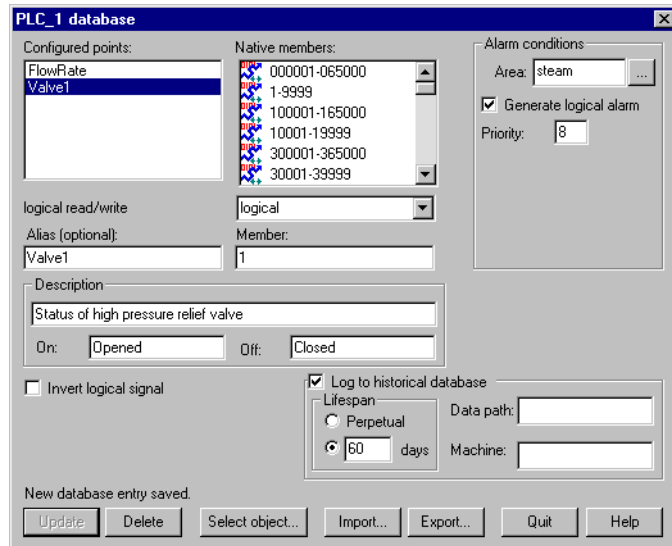
Alarm parameters of a logical signal include the alarm **Area** assignment field and the **Generate Logical Alarm** checkbox. When selected, the data member generates an alarm whenever the value is ON; the alarm condition clears whenever the value is OFF. Notice that if the **Invert Logical Signal** checkbox is selected, the value used here is the inverted value. See Chapter 9, *Alarms*, in the *Lookout Developer's Manual* for more information about alarms.

Text Member Parameters

The text data member database contains only **Alias**, **Member**, and **Description** fields. You can log text data to the database, but the only way to query for the logged text string is though the Lookout SQLExec object.

Importing and Exporting Object Databases

Use the import database service to copy database member parameters from an Excel spreadsheet file directly into an object. Use the export database service to copy an object database into an Excel spreadsheet file. This is what you can do using these services:

- Export object database parameter definitions to Excel for the purpose of documentation.

- Export an object database to Excel, perform global replacements on data member parameters, 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 a single driver object database parameters in Excel. Import that database into multiple driver objects.

Although you can import and export any object database, you may find that these services are most useful for objects with large native databases, such as driver objects and Data Table objects.
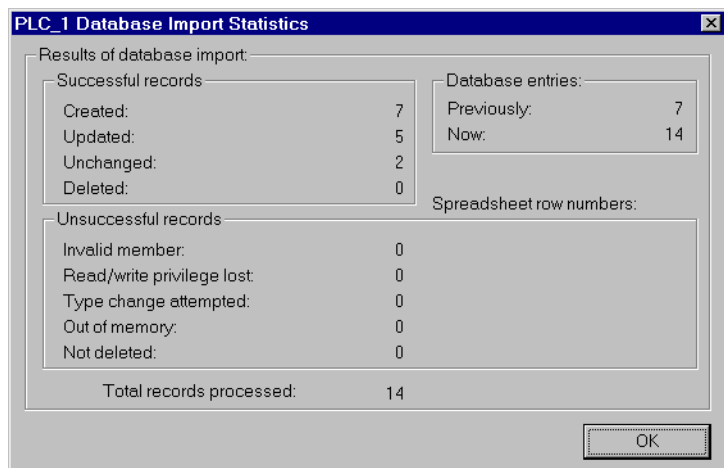
Exporting an Object Database

Follow these steps to export an object database:

1. Open the **Edit Database** dialog box, either through the menu or the Object Explorer, choosing the object you want to export from.

   The following diagram shows a Modbus object that already has a number of logical and numeric data members defined.
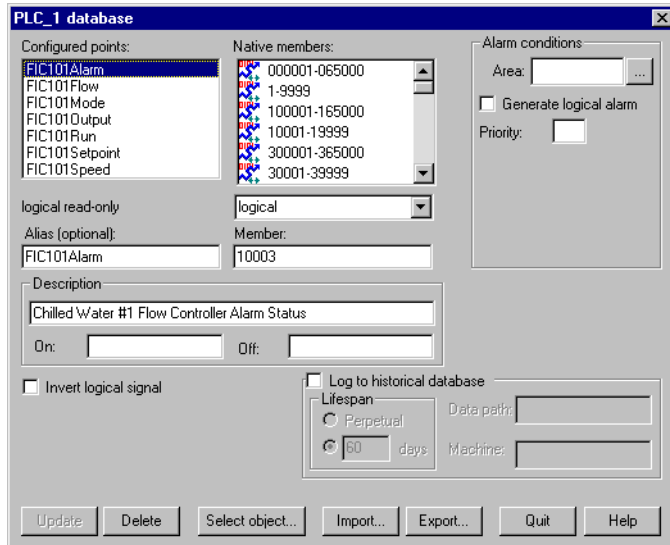
2. In the database dialog box, click on the **Export** button.
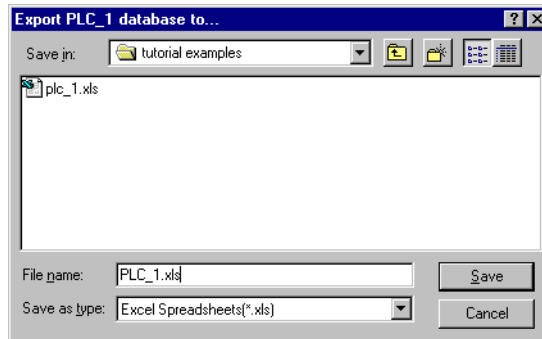
3. In the **Export Object Database to** dialog box, choose a directory path, enter a filename and click on **OK**.

When you export a database, Lookout does not export every possible data member (many driver objects have a capacity for thousands of members). Instead, Lookout exports configured points; that is, data members that have at least one parameter already defined. Lookout also exports native members that are in use (that is, connected to other objects).

The spreadsheet file that the Lookout **Export** command creates is in Excel Version 2.0 format. An example of the `.xls` file is shown in the following illustration.

Figure 4-5. Example Spreadsheet File Created with Lookout Export Command

Creating a Database Spreadsheet

The easiest way to create a database spreadsheet for an object is to create the object in Lookout, define the parameters for one logical data member, one numeric data member, and one text member (if necessary). You then export the database to a spreadsheet file.

This creates a basic spreadsheet file with all the necessary column labels for Lookout to read. The three data members you edited in Lookout furnish examples you can follow to rapidly create or edit more data members of the same type, using the convenient tools of your spreadsheet program.

Notice that row 1 in Figure 4-5 contains column labels. These include the names of all possible data member parameters. Lookout references the labels of a spreadsheet file, not the column letters, so if you were to create a spreadsheet to import into Lookout manually, you would have 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
- Member
- Alias
- 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.

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

The rows below the column labels (below row 1) each represent a database data member. For example, row 5 in the spreadsheet in Figure 4-5 represents the data member whose alias name is `FIC101Mode`.

You can easily add rows to define new data members. Copy the rows associated with FIC101, for example, and then modify the new rows slightly by identifying different native members and giving them new aliases and descriptions.

After you have edited and expanded your spreadsheet file, you can save it and import the new values back into Lookout, adding all the new data members to those your configured earlier as template samples.
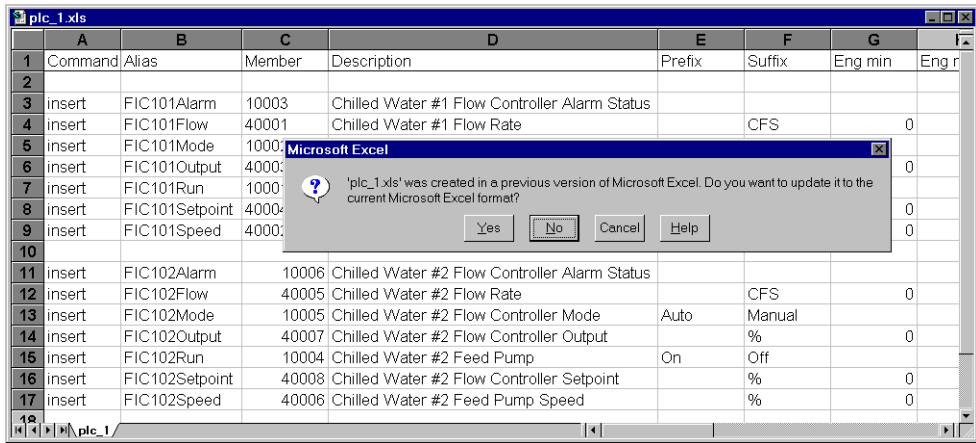
Figure 4-6. Example of Importing Spreadsheet into Lookout

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 when you select **File»Save.** This dialog box is shown in Figure 4-6. *Be sure to select* **No**. Lookout does not currently accept Excel spreadsheet files from versions greater than 4.0.

Importing an Object Database

Editing a spreadsheet file for several similar data members, and then importing it as an object database, can be much faster than working in the Lookout dialog boxes one at a time. When you import a database, Lookout reads the first 30 columns (A–AD) It ignores columns that do not have labels and columns beyond AD.

Each row in the Command column (column A) contains either the keyword `insert` or the keyword `delete`. When you import a database, Lookout ignores rows that do not have the `insert` or `delete` keyword. It adds those records whose command keyword is `insert`. It removes those records whose command keyword is `delete`. To determine exactly which records to delete, Lookout uses the record alias name; or if the record does not have an alias name, it matches the record member name.

Follow these steps to import an object database:

1. Open the **Edit Database** dialog box, either through the menu or the Object Explorer, choosing the object you want to import from.

2. In the object database dialog box, click on the **Import** button.
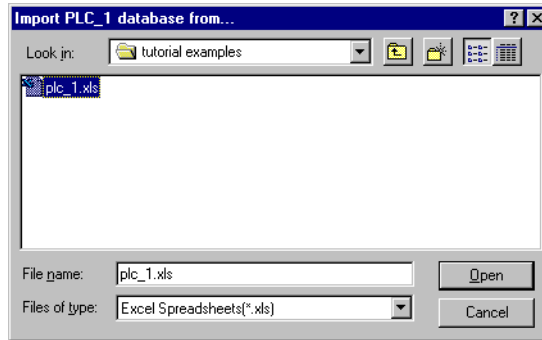
3. In the **Import Object Database from** dialog box, choose a directory path, select the filename and click on **OK**.

   When finished, Lookout presents you with a set of database import statistics, as shown in the following illustration.

As you can see in the following figure, the data members added to the database spreadsheet were successfully added to the **Configured points** list.
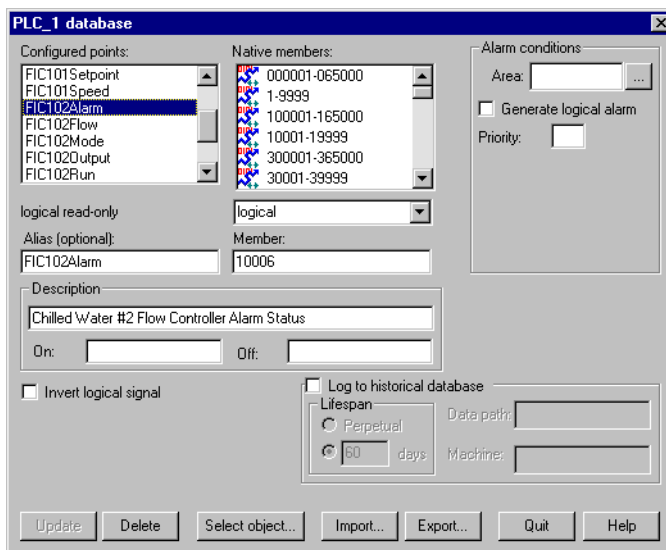
Copying an Object Database

The import and export features make it easy to copy the database of an object. This is especially useful when creating large SCADA applications, such as gas pipelines with multiple compressor stations.

The key to defining multiple driver objects that require duplicate databases is to first create an object in Lookout for each RTU or PLC. Then create a single database in Excel. Next, import that database into each driver object.

Connecting Objects

In Lookout, you connect objects to pass control signals between objects. Once connected, objects pass signals between their various data members and parameters.

There are a number of different kinds of connections in Lookout, and different ways to make those connections. The method you use depends on what you are trying to accomplish.

You can connect one data member to another. You can connect a data member to an object parameter, or an object parameter to a data member.

Finally, you can make a URL connection as a remote position source for a Lookout control or symbolic link, or as a trace item for a HyperTrend object.

You can make these Lookout connections in the following ways:

- Direct—Made with the **Edit Connections** dialog box, these connections are fixed and work in one direction only. This sort of connection should only be made from a server process to hardware installed on the server's computer, or between objects contained within the same process.

- URL remote—A URL remote connection is a flexible type of fully reciprocal connection between a Lookout control and a Lookout object read/write data member. Changing the control changes the data member, as you would expect, but if the value of that data member changes from some other cause, the control value changes as well. This sort of connection is what you use to connect a client process to a server process, or to link two client processes through a server.

- Complex—You can make complex connections using variables and logic tests to respond dynamically to changing needs and circumstances. Complex connections can apply mathematical operations to data, processing it for use by another object. You can make direct complex connections, and with the Lookout Symbolic Link object, make remote complex connections as well.

Tip   Remember, when you choose an object in the **Edit Connections** dialog box, always select the object you want to connect *to*.

Connecting Data Members to Parameters

The following example uses the (implicit) data member of a Pot object as the **Data** source of an Average object to illustrate the simplest type of Lookout connection. You can tell that **Data** is a parameter because it appears in the Average object dialog box.

Suppose you create a Pot object as shown in the following illustration.

Next you might create an Average object. Connect the potentiometer numeric signal to the **Data** parameter by entering Pot1 as the Average object's **Data** parameter. The Average object calculates the average level of Pot1 over time, as described in the Average object definition in Lookout help or the *Lookout Object Reference Manual*.

This is the simplest case for making a direct connection in Lookout; connecting the input for one object to the output of an object in the same process. In many cases you will have to connect objects in one process to objects in another process on the same computer, or on another computer on the network. For these connections you may find it more convenient to use the expression editor, which you access by right-clicking in the yellow expression field. The tutorial examples in Chapter 5, *Developer Tour*, guide you through making connections in detail.

Identifying Object Data Members

Because many objects generate multiple signals you can use as inputs to other objects, the object name is not always sufficient to identify the desired signal. You must identify both the object name and the appropriate data member. For example, a Modbus object generates thousands of signals, some logical and some numeric. To specify exactly which signal you want from a particular Modbus object, enter the object name followed by a period (.) and the data member you want.

If you have a Modbus object with a name `PLC1`, and you want to display the value residing in holding register 40001 and the status of coil 100, reference those particular data members by typing `PLC1.40001` in one expression and `PLC1.100` in another expression.

Connecting Data Members to Data Members

The following information expands on the previous example, in which you created a potentiometer named `Pot1`. In this example, you create a Modbus object and connect the numeric signal generated by the pot to a 16-bit holding register at address 40001 on the PLC.

1. Create a Modbus object. You do not have to have a real PLC connected to your system. Just accept the default parameter settings in the Modbus dialog box and click **OK**.

2. If your Object Explorer is open, right-click on the Modbus object you just created and select **Edit Connections** from the pop-up menu. Otherwise select **Object»Edit Connections** from the Lookout menu bar. You use this the **Edit Connections** dialog box any time you are connecting to the database of another object.

3. If you use the **Object»Edit Connections** menu command, the **Edit Object Connections** list box appears. Choose the object that contains the data member you are *connecting to*.
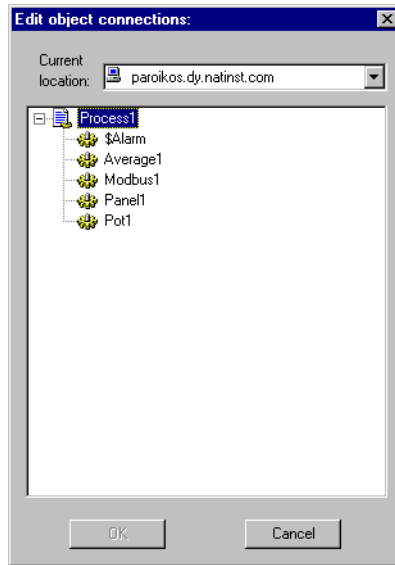
The **Current location** combo box at the top of this dialog box should show your computer as root. See the *Lookout Navigation* section for more information on how to set the root in a Lookout dialog box.

Because you are connecting the Pot1 signal *to* data member 40001 in the Modbus object, select the Modbus1 object and click on **OK**. The **Edit Connections** dialog box appears.
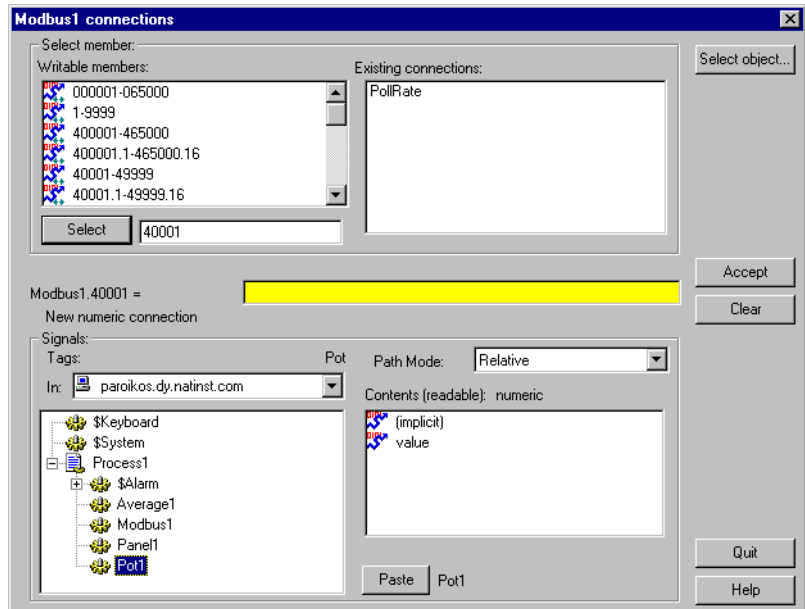
The top part of this dialog box contains information on the object you are connecting to. You select a writable data member from those listed in the **Writable members** list, and the selected member appears to the left of the yellow expression field in the center of the dialog box.

The lower part of the dialog box lets you navigate to the data member you want to connect to the data member you select in the upper portion of the box. You can also enter a data member directly in the yellow expression field.

4. Enter the data member you want to connect *to* in the list box to the right of the **Select** button. You can type the data member you want directly, or you can choose it by scrolling through the list of writable members and clicking on the data member (or range of data members) you want. You can edit the number that appears in the selection box.

    Click on the **Select** button to identify the data member to be written to. Notice the dialog box now reflects the specific member; **Modbus1.40001 =**. Also notice the equal (=) sign. This prompts you to connect an expression to **Modbus1.40001**. In this case, use the expression Pot1.

5. Inside the **Signals** section of the dialog box, set the your local computer as the root in the **In** field. Navigate to Pot1 and select it by clicking on it. Click on the (implicit) data member in the **Contents (readable)** list and then click on the **Paste** button.
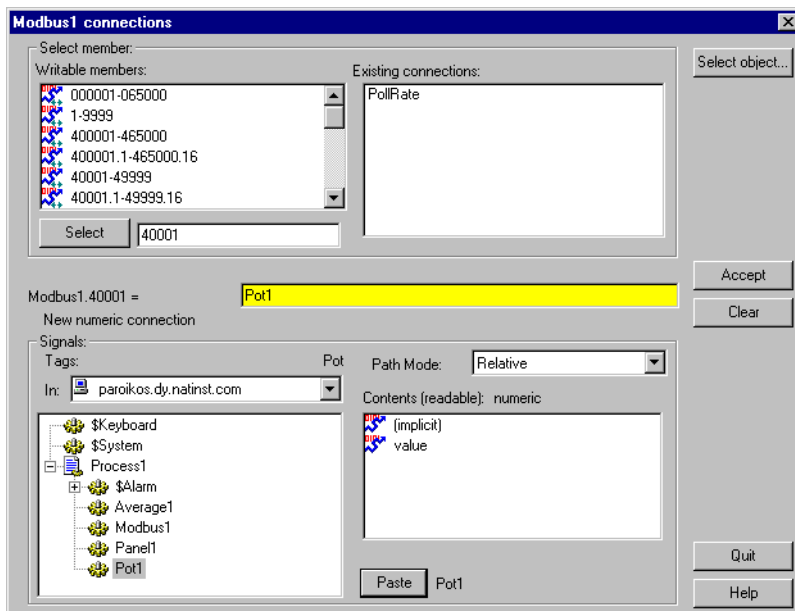
Notice that the data type of the data members you have selected are displayed to help you make sure you are connecting like type to like type. Notice too that you can easily edit existing connections by clicking on the list in the **Existing connections** field.

✏️ **Note**   Just above the Contents (readable) list in the Signals section, you will notice a listbox named Path Mode. This is an important setting for developing networked processes in Lookout. See the *Path Names in Lookout* section for more detailed information on how to use this feature. Using different path settings is demonstrated in the exercises contained in Chapter 5, *Developer Tour*.

6. Click on the **Accept** button. This confirms the connection. You should see a message confirming the action. You will also see data member 40001 added to the **Existing connections** list box.

You have now connected a data member of one object to a data member of another object. Any change in the (implicit) value of Pot1 is, by definition, an event, sending a signal to all the other objects Pot1 is connected to.

Because it is now connected to `Modbus1.40001`, adjusting `Pot1` changes the value of data member `40001` in object `Modbus1`.

URL Connections

A URL (Universal Resource Locator) connection is distinct from a connection made through an expression because it cannot contain any condition checking or mathematical operations. It is also designed to permit reciprocal flow of data through the connection. A control connected to a PLC register using a URL connection changes the register value when it is adjusted. But if the register value changes from some other cause, the control will also change to match the current register value.

You can only make a URL connection to data members that are writable, or both readable and writable. You cannot make a URL connection to a read-only data member. Remote position source connections, Symbolic Links, and the HyperTrend object require URL connections, as indicated by a green field.

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.
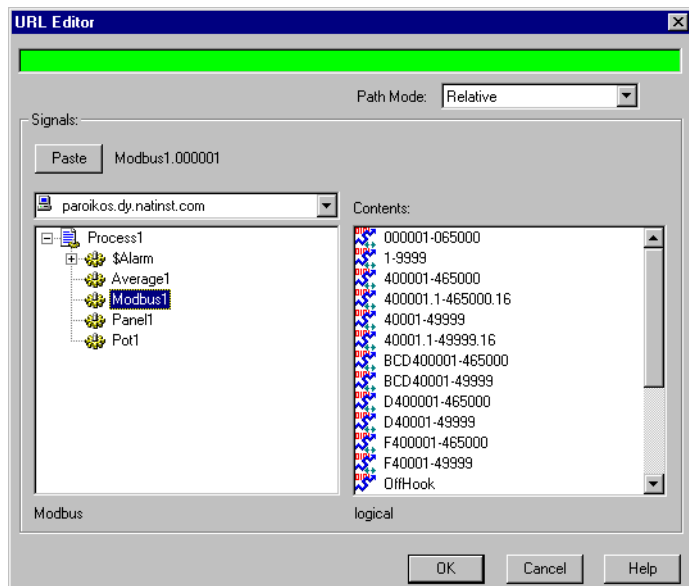
You can use the tree view in the left window to locate the object you want, and select the data member in the right window. Click on the **Paste** button to insert the complete string into the URL field.

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

Remote Position Source Connections

Remote position source connections are the preferred method for connecting a control in a client process to a server process in Lookout.

Because you ordinarily want to be able to have your client processes work from any computer on your network without having to edit them, it is best to make connections from client processes to server processes, and not from servers to clients. Because server processes usually need to be run from the same computer all the time, you can use absolute paths to make sure your client processes will connect to your server processes no matter where the clients are run.

Symbolic Link Connections

Lookout Symbolic Link objects have several important uses in Lookout. The following figure shows the **Create Symbolic Link** dialog box.
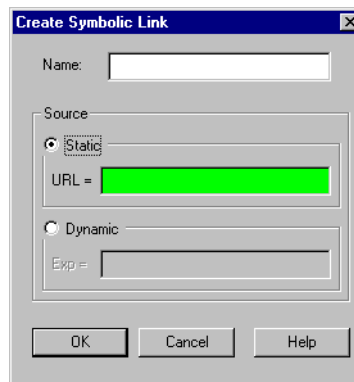
You can make a URL connection in the **Static** source field to an entire process with a Symbolic Link and then access any object in that process through the link. This is a crucial part of Lookout redundancy, which is described in Chapter 10, *Redundancy*, of the *Lookout Developer's Manual*.

You can enter a complex expression in the symbolic link **Dynamic** source field, and then make a URL connection to the symbolic link itself, circumventing the restriction on complex expressions in remote position source connections for controls.

Using a Symbolic Link is demonstrated in Chapter 5, *Developer Tour*.

HyperTrend Item Connections

Because the HyperTrend trends data from the Citadel database, every item you display on a HyperTrend object is accessed through a URL connection.

If you need to trend a complex expression, you can create a named expression by creating an (expression) object. You can enter the complex expression you want to trend, log it to the historical database, and then display it as a HyperTrend item.

Note   If you convert a Lookout process created in a version earlier than Lookout 4.0, Lookout will automatically create and log a named expression for your HyperTrend object to display when it recompiles the .lks source file.

Path Names in Lookout

In versions of Lookout prior to Lookout 4.0, you could only have one process running in any instance of Lookout, and there were no folders in a process to organize objects. Networking was done through DDE, which used specific paths.

In Lookout 4 and greater, the relationship between objects running in different processes on different computers is far more flexible and potentially complicated. When you insert an expression on a Lookout panel or make a connection to or between data members, you might be displaying a value that comes from the process containing that panel. You might also need to display a value that comes from another process running on the same computer or from a process running on some other computer in your network.

A path can be *absolute*, starting with a computer's fully qualified network name and leading through nested directories and subdirectories to a single object; or *relative*, instructing the computer to search for an object in some directory defined not from the computer down, but relative to the currently active folder.

In order to provide maximum programming flexibility, you can use a number of different levels of relativity, or *path relativity modes*, in setting the path to the data when you create an expression. These modes operate very much like other relative paths used in DOS operations, spreadsheet cells, and Web page addressing.

You can set a path for each individual object or connection, using a different level of relativity. These path relativity modes include:

- Relative (the Lookout default)
- Process Relative
- Computer Relative
- Absolute

The level of path relativity you choose makes a difference in how your process operates when copied or moved to another computer. For instance, suppose you have a client process running on computer Alfred that refers to objects in a server process also running on computer Alfred. If you run that client process on computer Bert, will it refer to objects on computer Alfred or computer Bert?

Setting the level of path relativity is how you can control which objects your processes refer to. The key point to remember about path relativity modes in Lookout is that the modes are distinguished by the path prefix.

In *Relative* mode, the form of the path is:

object

or

folder\object

without prefix.

Relative mode is the default mode in Lookout 4. All paths are relative to the folder or process that contains the expression you are creating or the object you are connecting to. Use Relative mode when connecting objects that should be grouped together and that should be copied or moved as a group.

Connections between objects in a folder should use Relative mode so that they can be copied and moved elsewhere while maintaining the same relationship with each other.

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.

Computer Relative mode in 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
running on the same computer.

For instance, if you had a process called `Station_Control` that called a
number of other processes by name on computer Alfred, you could use that
same `Station_Control` process on computer Bert, to refer to the
identically named processes running on computer Bert.

Absolute 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.

Note   Though Relative mode is the Lookout default, when you use an object outside the
immediate location of your expression, the path becomes more specific. The addition of
two periods (..) at the beginning of a path, for instance, can be used to specify the parent of
the first folder listed.

The reason for manually changing a path mode to a level beyond that strictly necessary
under your current circumstances is to ensure that your process will still work properly if
you relocate it, completely or partially.

Displaying Data Members on Control Panels

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

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.
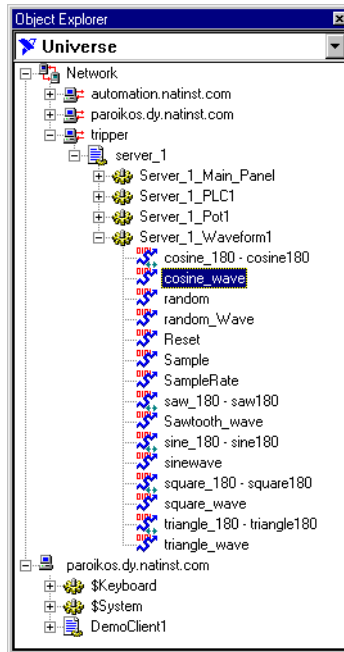
3. Drag and drop the data member onto the panel you want the data to display in. Lookout inserts an expression to display the data. You may have to adjust the size of the display for the data to appear properly.

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

You can also insert an expression to display data by choosing
Insert»Expression from the Lookout menu bar. The **Insert Expression**
dialog box appears as shown in the following illustration.
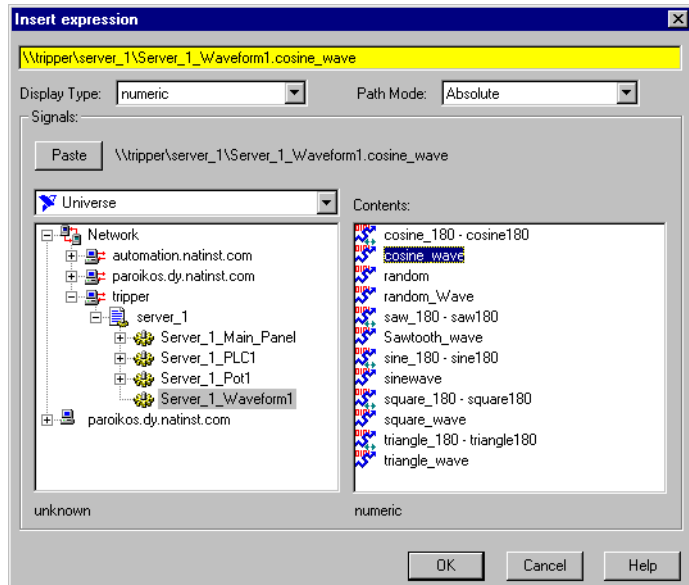
5. Navigate to the data member you want to display, just as you did with
    the Object Explorer.

6. Set the **Path Mode**. In the preceding illustration it is set to absolute
    because the data is being displayed on a client process panel in one
    computer (paroikos) and originating from a server process on a
    different computer (tripper).

7. Select the data member you want to display in the **Contents** list.

8. Click on **Paste** and then **OK**.

    Lookout responds by presenting the **Display numeric signal** or
    Display logical signal dialog box. 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.
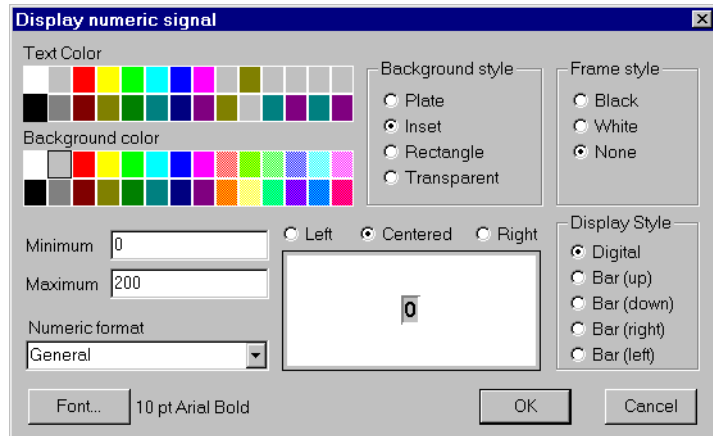
Select the display parameters you want, and click on **OK**.

You can also drag an object from the Object Explorer to a panel. When you do, the **Insert Expression** dialog box appears, and you can proceed as you did when you invoked this dialog box from the menu.

5

Developer Tour

The following tour guides you through the development of both client and server process files. Any time you want to learn more about a particular object class, function, or feature, refer to the appropriate section of your manuals or online help for a more detailed explanation.

First you will start Lookout and access data from a server process running on a computer at National Instruments headquarters in Austin, Texas.

After that you can build your own server process to run on your own computer. You can follow that with a client process that you can run on the same computer as your server, or on any other computer on your network.

Finally, you will have a chance to experiment with the Web Client capabilities of Lookout.

Before Starting the Examples

If you skipped Chapter 3, *Lookout Basics: Windows, Tools, and Files*, you might want to take some time and look through that chapter to familiarize yourself with the Lookout environment.

Chapter 4, *Using Lookout*, introduces you to the fundamental Lookout operations, such as making connections and editing databases, that you will be using in the following exercises. If you find yourself having difficulties performing some of the operations described, you might find a review of Chapter 4 useful.

Make a Quick and Easy Network Connection to a Lookout Server

A computer located at National Instruments in Austin, Texas is running a Lookout server process. In this brief exercise, you are going to connect to that process and read data from that server with your own Lookout client.

1. Launch Lookout 4.

2. Create a Lookout process by selecting **File»New**. A dialog box appears prompting you to name your new process. If this is your first time to launch a new version of Lookout, it may take a few moments for this first dialog box to appear. If you have never run Lookout at all on the computer you are using, you have to register the program. See Chapter 1, *Installing Lookout*, for information concerning Lookout registration and configuration.
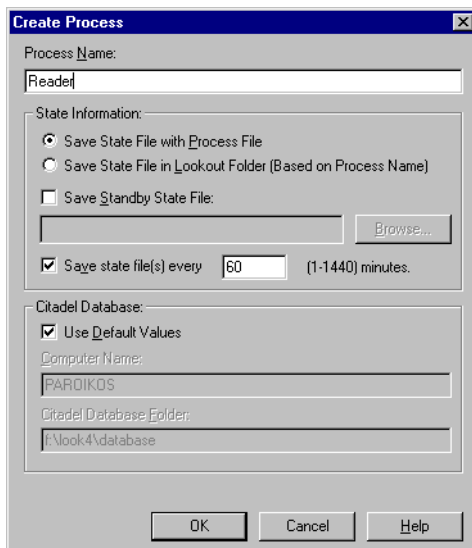
3. Name your process Reader and accept the default settings for **State Information** and the **Citadel Database**.

    For more information on configuring the **State Information** and **Citadel Database** options, see the *Creating A Lookout Process* section of Chapter 4, *Using Lookout*.

4. Click on **OK**. The **New Control Panel** dialog box appears.
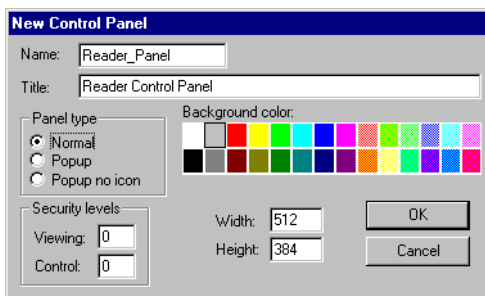
5. Use `Reader_Panel` for the **Name** of the panel object. When you
   make a connection in Lookout, you use the object name as part of the
   connection. There can be no space in the string, which is the reason for
   the underscore. Enter `Reader Control Panel` for the **Title** of the
   panel. This is the phrase that appears at the top of your panel.

6. Click on **OK**.

7. If it is not open already, open the Lookout Object Explorer by selecting
   Object»Object Explorer. The explorer window appears, with your
   computer as the root. In the following illustration, the computer
   `paroikos` is set as the root.
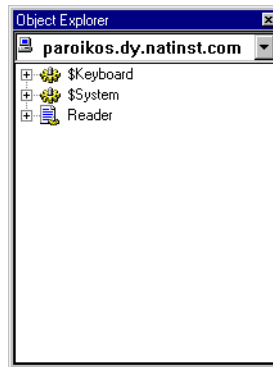
You can use the Lookout Object Explorer to create and organize
objects, as well as a shortcut for making connections and performing
other Lookout tasks. The figrst task to accomplish is to use the Object
Explorer to register the server at National Instruments that is
generating the data you are about to access.

8. Click on the down arrow in the upper right section of the Object
   Explorer window and select `Universe` as the root.

9. Right-click on `Network` and select **Register Computer**.
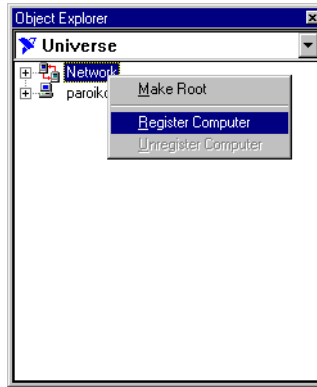
The **Register Network Computer** dialog box appears.
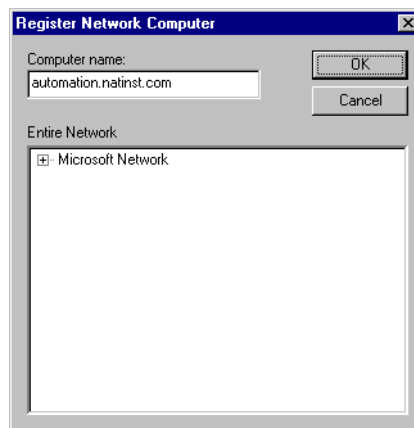
10. Enter `automation.natinst.com` in the **Computer name** field, and click on **OK**.

11. Click on the + sign to the left of `Network` in the Object Explorer window. It expands to reveal your computer name along with `automation`.

✎ **Note**   If for some reason you cannot connect to a computer outside your company firewall, or if for some other reason the `automation.natinst.com` computer is not available when you attempt this exercise, you can use a process called `demoserver`, located in the examples subdirectory of your main Lookout directory. Load this process into Lookout using the **File»Open** command, and connect to this instead of the process running on the network computer.

12. Click on the + sign next to `automation.natinst.com`. It opens to reveal the names of the Lookout processes running on `automation`.

13. Click on the `demoserver` process to examine the objects used in that process. It should have several folders. Open the `Waveforms` folder, and find the `Waveform1` object.

    Almost all Lookout objects have data members, each containing numeric, logical, or text string data. Double-click on `Waveform1` to see the data members contained in that object.

14. Right-click and drag `sine_180-sine180` over to your control panel. Lookout data is *polymorphic* (meaning that data is passed without being strongly typed). When you right-click to drag and drop a data member, a dialog box gives you the choice of determining what type of data to display. (This only works if you right-click and drag. A left-click and drag inserts the expression in its default type.) Choose **Insert Expression as Default**. Click and drag the sizing handles to enlarge the display.
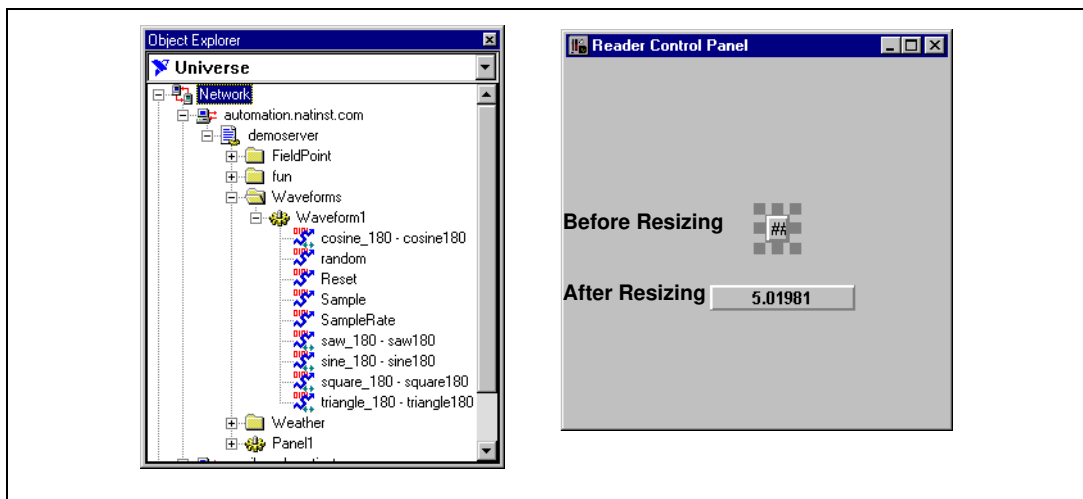
The changing numbers you see come from National Instruments headquarters in Austin, Texas. You have completed your first network connection in Lookout.

Modifying the Display

1. A number is not always the best way to display a value. Right-click on the value display you just created, and choose **Display Properties**. The following dialog box appears.
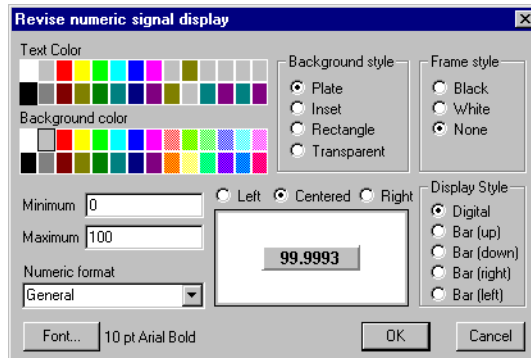
2. Choose **Bar (up)** as the **Display Style**. Notice that the color option labels change to reflect the new display appearance.

3. Select blue as the **Bar Color**, white as the **Background** color, and click on **OK**.
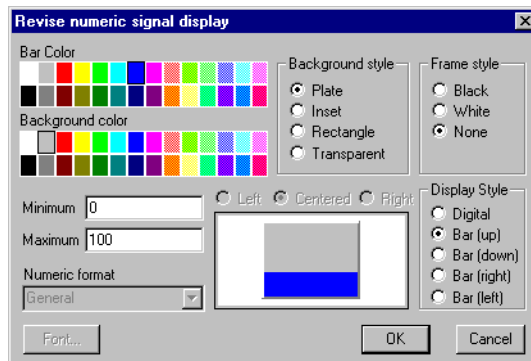
4. Adjust the bar so that it is roughly four or five times as high as it is wide.

5. Select **Insert»Graphic**. The **Select Graphic** dialog box appears, initially on a collection of arrows.

6. Click on the **Tanks** category.
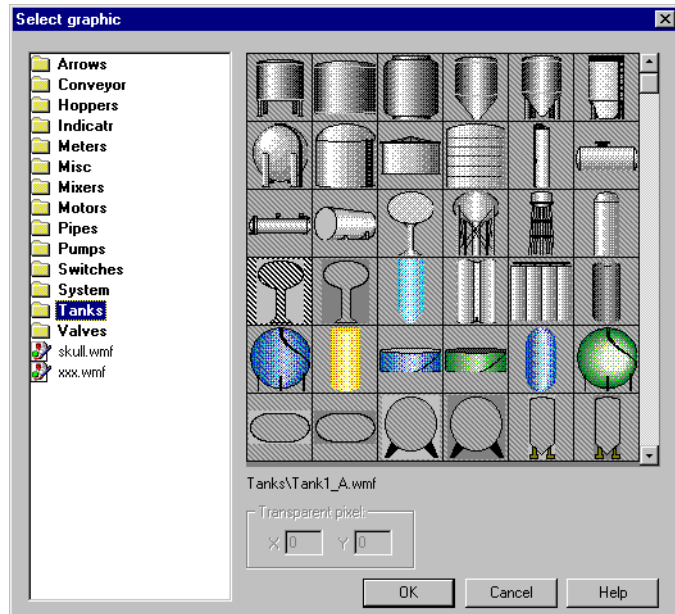
7. Select the water tower graphic Tank1_A.WMF, the left-most item in the fourth row, and click on **OK**.

8. Drag the water tower graphic over your bar display, and adjust its size with the sizing handles until the transparent section is entirely filled by the bar display, without any of the bar showing around the edges of the transparent section.
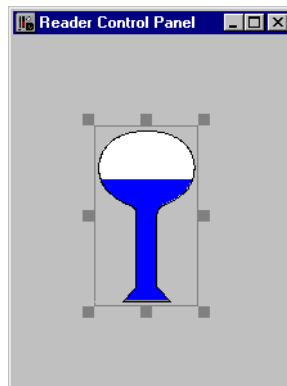
While a real water tower is unlikely to fill or drain as quickly as this representation, this example shows that you can display data in Lookout in any number of informative and dramatic ways.

Make a Lookout Server

In this exercise you will make a simple Lookout server to furnish data for the client you will build in the exercise following this one. This example focuses on using the Lookout Object Explorer to help in building a process, using the data table as a data repository, and creating and scaling data members.

Lookout servers are what you use in a computer connected to your field hardware. A server is one element of your Lookout control system that should run on one computer all the time, and is not intended to be mobile, (that is, capable of being run from any computer on your network). The server depends on the direct connections from the computer running it to the hardware it is controlling and monitoring.

Clients operate through servers, and should be mobile, capable of being run on any computer in your network, including the computer running a Lookout server.

Creating the Process

In the first section of this exercise you will create a process and two objects.

1. Begin by closing your Reader process. Select **File»Close** to close a Lookout process. The **Select process to close** dialog box appears. Select your reader process and close it by clicking on **OK**. Save the process if you want to, and have not saved it already.

   Clicking on the Windows close box in the upper right corner of a control panel does not halt a Lookout process.

   Notice that when you close your Reader process, Lookout leaves edit mode, which automatically closes the Object Explorer. When you create a new process, Lookout automatically switches back to edit mode, and the Object Explorer reappears. If you were to open an existing process, Lookout would stay in run mode.

2. Select **File»New** to create a new process. Name the process `My_Server` and click on **OK**.

3. When the **New Control Panel** dialog box appears, click on **Cancel**. While there are many reasons to use a control panel in a server process, it is not always necessary, and in this instance you can omit it.

4. If the Lookout Object Explorer did not appear when the **New Control Panel** dialog box did, check to make sure you are in edit mode, and to see if the dialog box is minimized at the bottom of the Lookout workspace. If the status bar at the bottom of the screen is yellow, you are in edit mode. If the Object Explorer is neither visible nor minimized, select **Object»Object Explorer** to open the explorer window.

    The Lookout Object Explorer lets you explore Lookout processes running on your computer in two ways: as the local computer running Lookout, and as one of all the computers on the network registered as running Lookout.

5. Make sure the local representation your computer is selected as the Object Explorer root. 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; the local representation has no I/O arrows. Your computer should appear in the root selection box of the Object Explorer window, with no red arrows, as shown in the following illustration.
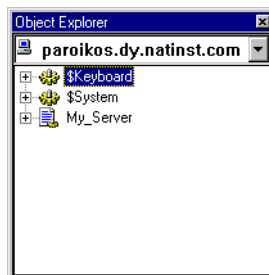
If your root is not set correctly, right-click on your computer name *outside the network node*, and select **Make Root**, as shown in the following illustration.
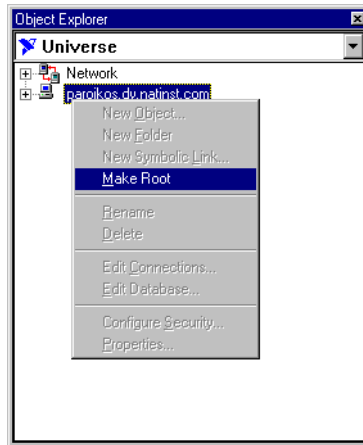
You can only work through your local computer node when you are creating new objects and performing similar editing tasks. This is why it is more convenient to make your local computer the root while creating or editing processes.

6. Right-click on your `My_Server` process and choose **New Object**.
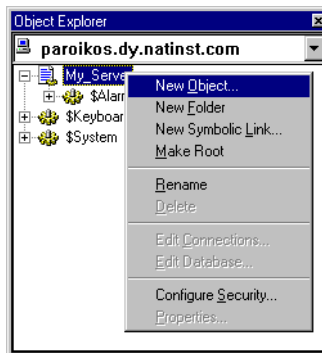
The **Select object class** dialog box appears. When it first appears, this dialog box may have the **Categorize** option selected. You may find this a convenience, allowing you to browse through all the Lookout object classes by category. Some prefer to have all the object classes available listed alphabetically, which happens when you deselect the **Categorize** option. Experienced Lookout users who know what object they want to create can then type the first two or three letters of the object class they want in order to select it quickly. This fast selection technique does not work when you are at the folder level, however.

In either configuration, when an object class is selected, a brief description of that object class appears in the outlined section of the box to the right of the selection list.
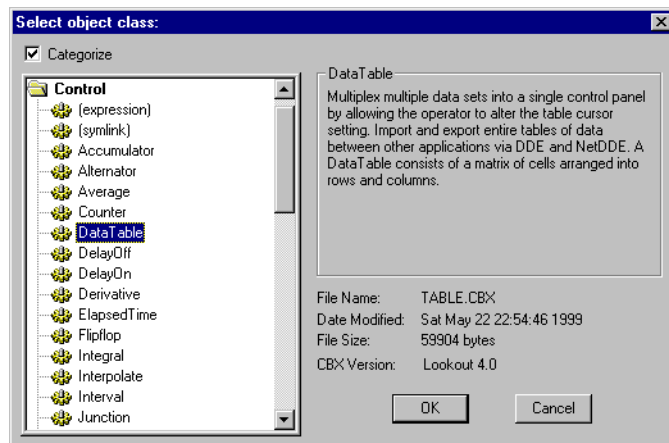
7. Open the **Control** category folder and select the DataTable object class. You can use a DataTable object to simulate the registers of a PLC, or to hold or transfer values. Click on **OK**.

    The DataTable is a Lookout object you can use for everything from multiplexing data to storing, updating, importing, and exporting data. Click on the **Help** button for detailed information, including examples, on how to use this versatile Lookout object, or print the DataTable pages from your *Lookout Object Reference Manual*.

8. Name your DataTable PLC_Simulator, and click on **OK**.

9. Right-click on My_Server in the Object Explorer and choose **New Object** again. From the **Calculation** category select the Waveform object class.

10. You can leave the name of the new object as Waveform1. Set **Period** to 0:10 (10 seconds), **Amplitude** to 45, and **Offset** to 50. Leave the rest of the parameters at their default settings, and click on **OK**.

    At this point you have built a simple working server process. Your PLC simulator can hold numeric, logical, and text data, simulating various kinds of PLC registers and control inputs or outputs. The waveform

object is generating streams of numeric data ranging from 5.0000 to 95.0000.

You may find it convenient to save your processes in your own folder. Consider creating a subfolder in your Lookout directory to hold your processes files.

11. Select **File»Save** and click on **OK** in the **Select process to save** dialog box to save your server. You will need it in the next exercise.

You can drag an expression to display your server data into a control panel, just the way you did in the first exercise with your Reader process. You might find it slightly more complicated with your server in its current state, however, because none of the data points have been configured or set up to log into the Lookout Citadel database.

You must configure data in order to log it to Lookout's Citadel database, and there are other things you can do to make your server data easier to work with. The next part of the tour guides you through these processes.

Editing the Database

1. Right-click on PLC_Simulator in the Object Explorer. Select **Edit Database**.
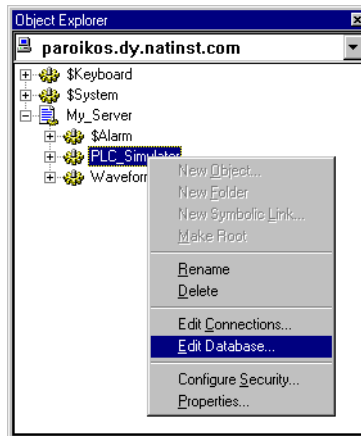
The dialog box for editing the object database appears.
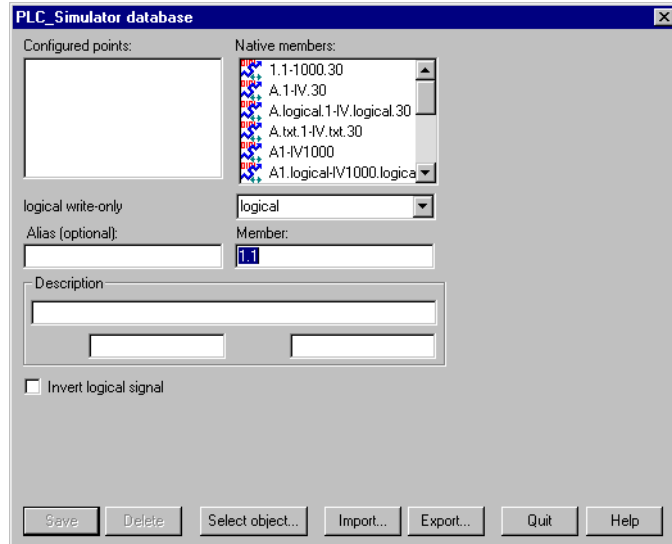
2. To prepare your server for use in the client exercises that follow this one, enter A1 in the **Member** field. Enter amplitude in the **Alias (optional)** field. Click in the **Log to historical database** checkbox, and enter .5 in the **days** field. Enter 0.1 in the **Deviation** field. Leave the **Alarm conditions** fields blank, and click on **Save**.
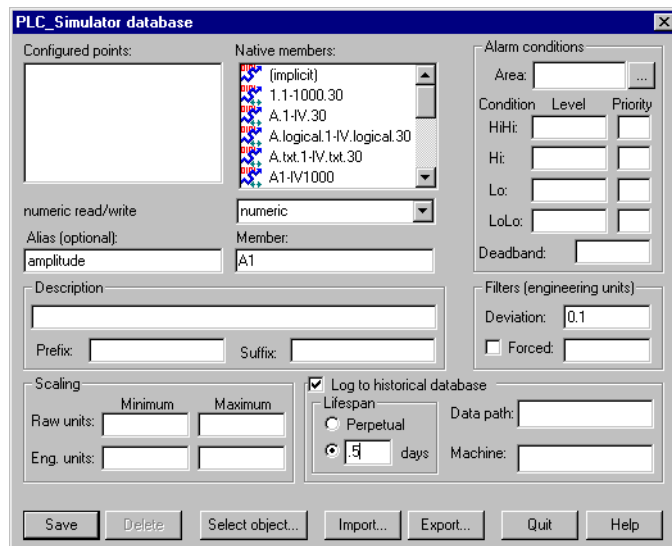
When you create an alias for a data member, you are, in effect creating a new data member. That is because you can access the same data in two ways, both through your alias and through the name of the native data member.

3. Repeat Step 2 for data members A2 and A3, using the same data logging parameters, but changing the alias to offset for A2 and phase for A3.

   As you may have guessed, you are configuring the database to use these data members to control waveform parameters, one of which controls the waveform period. You will have to scale that data member in order to convert ordinary numeric data into data that Lookout can interpret as time values.

4. As shown in the following illustration, enter A4 in the **Member** field and period in the **Alias (optional)** field. You do not need to log this data, so if the **Log to historical database** checkbox has a check in it, unselect it.

You will use the `period` data member to control the period of a waveform in seconds. Lookout uses fractions of a day as the basis for time measurements. Rather than set up your control to work with the fractions involved, you can scale the values for this data member when you configure it in the database.

a. Enter `0` as the **Minimum** and `100` as the **Maximum** in the **Raw Units** fields of the **Scaling** section of the dialog box. The control you will create later in these exercises to adjust the period of the waveform will have a value range of 0 to 100.

b. Enter `0` as the **Minimum** and `0:00:30` (30 seconds) as the **Maximum** in the **Eng. Units** fields of the **Scaling** section of the dialog box. Lookout now automatically converts the value of the control to the appropriate number of seconds between 0 and 30.

c. You do not need to log this data. If you do, make sure the **Deviation** field of the **Filters** section is left blank. The small magnitude of the scaled data member will not log well, otherwise.

d. Click on **Save** (or **update**).

Tip   When you edit the database and add an alias, you are, in effect, creating a new data member. To use any scaled data you must use the aliased data member. The native data member reads and returns the raw data, not the scaled data.

5. Click on the **Select object** button at the bottom of the dialog box and navigate through the tree to your My_Server process. Double-click on Waveform1. You can now edit the Waveform1 object database, as shown in the following illustration.
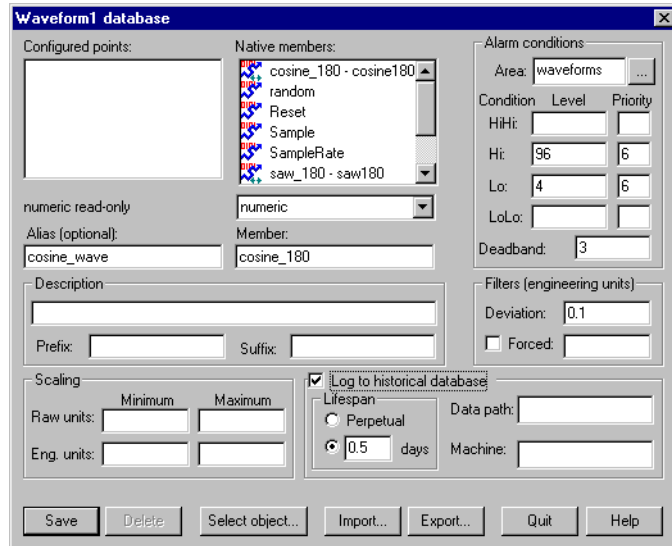
6. Select cosine_180 as the data member to configure (it appears as the default on opening the database editing dialog box). Assign the alias cosine_wave.

7. Check the **Log to historical database** option, and enter .5 in the days field. Set **Deviation** to 0.1. Create an alarm area by entering waveforms in the **Area** listbox.

8. Set the **Hi** alarm **Level** to 96 and assign a **Priority** of 6. Set **Lo** to **Level** 4 with a **Priority** of 6, and enter 3 in the **Deadband** field.

9. Click on **Save**.

    If you want, you can also configure random, sawtooth, sine, square, and triangle waves using the same filter, and logging settings. *Leave the* **Alarm conditions** *fields blank for all these waveforms except the cosine wave.* Assigning alarm levels to all waveforms in this object will create a flood of alarms later in this exercise that are not necessary to the demonstration.

10. Click on **Save** after you configure each data source, before moving on to configure the next.

11. Click on **Quit** when you have finished editing the database. If you attempt to quit without saving your last change, Lookout prompts you to save or cancel the last change.

12. Expand the tree view of My_server in the Lookout Object Explorer. You now have a collection of configured and raw data points visible, as shown in the following illustration.
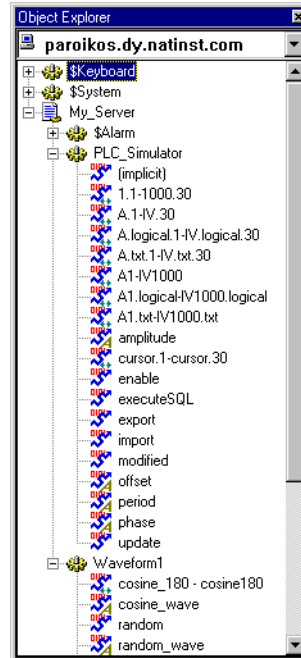
Notice the A in the icons denoting an alias.

Your server application is now ready to feed data to Lookout clients and accept control changes.

13. Select **File»Save** and then click on **OK** to save the changes you have made in your server. You will need the server in the next exercise.

Make a Lookout Client

A Lookout client process can be run on the same computer as a Lookout server, or it can be moved and run on any computer on the same network as a Lookout server. In order to make sure your Lookout client can be moved with no difficulties, you must follow some simple principles in designing and building clients.

- Controls in client processes should have no direct connections to another process. Client process controls should connect using remote position source and Symbolic Links.

- You can connect data from a server process to a client process, but a server process should not be dependent on a client process being present in order to operate properly.

- Client processes should not connect directly to your field hardware. Client processes should interact with hardware through a server process.

Most of these principles are illustrated in the following example.

1. Open Lookout if it is not already running. If `My_Server` is not already running, select **File»Open** and open the server you created in the previous exercise. If for some reason you did not build the server in the previous example, or you failed to save it, you can use copies of the `Server1` files in the `Examples` subfolder of your Lookout directory. There are four `Server1` files, each with a different extension (`.l4p`, `.lks`, `.l4t`, and `.lka`). Make copies of each and rename the copies `My_Server`, leaving the extension names intact.

   The `.l4p` file is the compiled process file that Lookout runs. The `.lks` file is the Lookout source file. If a problem develops with the binary version of the process, or if you are using a different version of Lookout than the version which created the `.l4p` file, Lookout recompiles this source file. This is the file you need to make sure you have back-up copies for. The `.l4t` file is the Lookout state file, which keeps a record of the values of different controls when Lookout closes. The `.lka` file is the Lookout security file for your process. This file must be in the same directory as your process files for your security configurations to work properly.

2. Make sure Lookout is in edit mode, with the yellow field status bar at the bottom of the workspace. If Lookout is not in edit mode, press <Ctrl-Space> to shift into the proper mode.

3. Select **File»New** and name the new process `Client_1`. You can let the control panel keep the default name of `Panel1`. Enter `Client 1 Control Panel` in the **Title** field, and click **OK**.

4. If the Lookout Object Explorer is not already open, select **Object»Object Explorer** to open it.

5. Begin with displaying some of the data you configured in your server in the earlier exercise as a trace on a chart display using the Lookout HyperTrend object class. Create a HyperTrend object by right-clicking on `Client_1` in the Object Explorer and selecting **New Object**. Choose HyperTrend from the **Display** category. The **Create HyperTrend** dialog box appears.
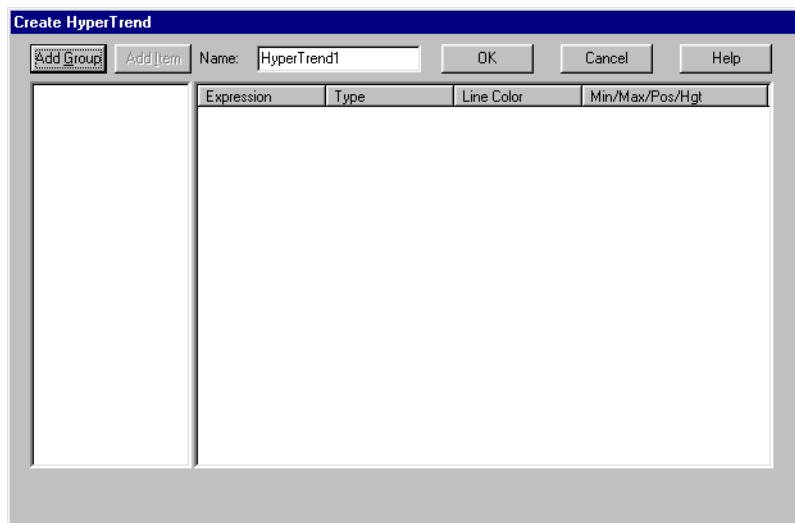
6. Create a new group by clicking on the **AddGroup** button.
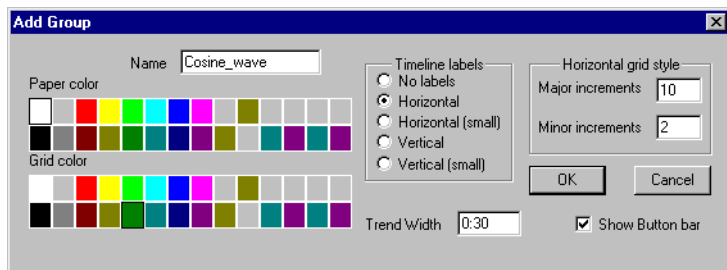
7. Name the group Cosine_wave. Set the Trend Width to 0:30
    (30 seconds), and click on **OK**.
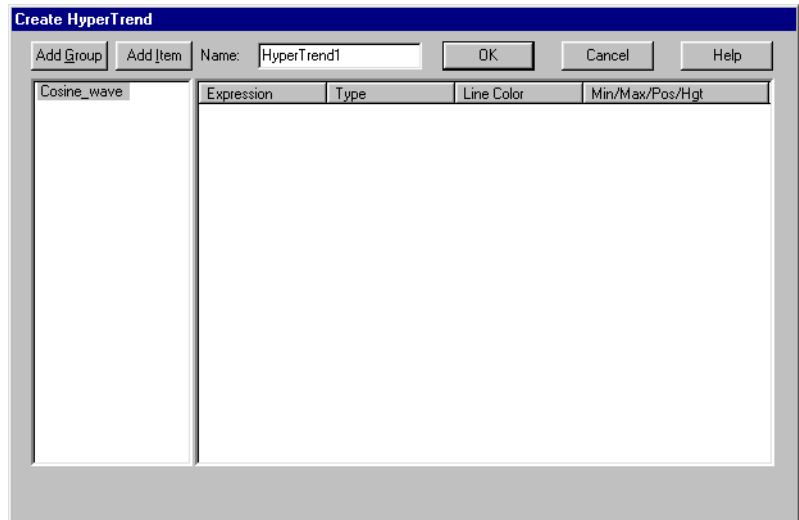
8. Add a trace by right-clicking on the Cosine_wave group and selecting
    Add Item, or by clicking on the **Add Item** button. Set the **Minimum**
    to 0 and the **Maximum** to 100.
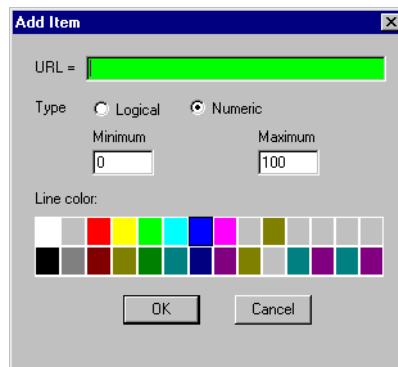
9. Right-click in the green **URL** field. The **URL Editor** dialog box appears.

    Because you want to be able to move a client process to any computer on your network and still have the client work properly, make sure you set **Path Mode** to Absolute and that your local computer is selected as the root. See the *Connecting a Control* section of this chapter for a more detailed discussion of path modes.
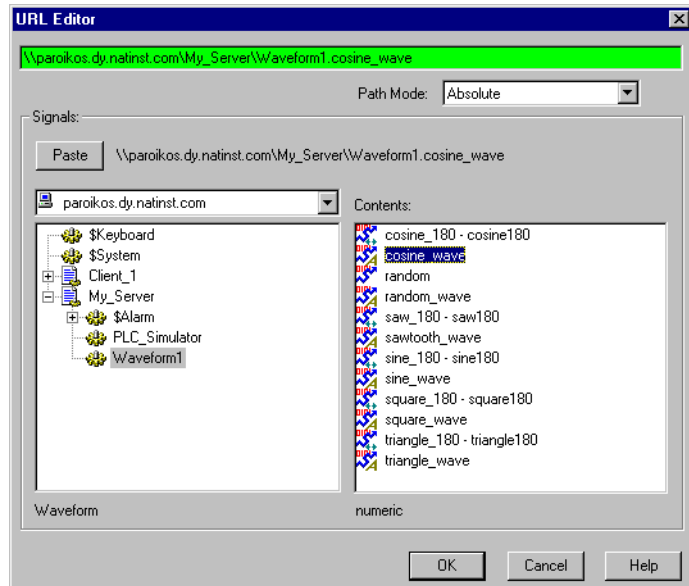
10. Navigate to Waveform1, the waveform object you created in your server process, and select the cosine_wave data member you configured earlier in this exercise. Click on **Paste** to insert the complete expression.

11. Click on **OK** until you have closed all the dialog boxes to finish creating your HyperTrend.

12. Drag the HyperTrend from the Object Explorer to your client control panel. Drag on the sizing handles to enlarge it. Your display should resemble the following illustration.
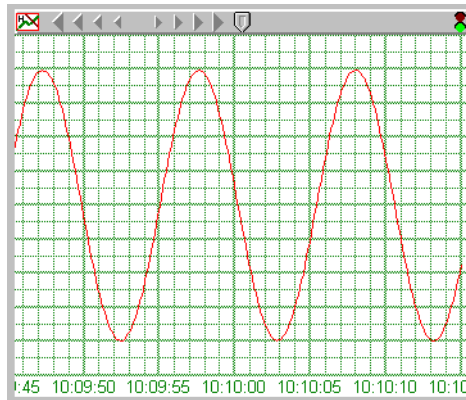
13. Press <Ctrl-Space> on your keyboard to exit edit mode and experiment with the capabilities of the HyperTrend object. Clicking on the icon in the upper left corner of the HyperTrend calls a menu you can use to start or stop the scroll, change trace groups (when you have more than one), or activate the HyperTrend cursor.

    When you stop scrolling, which you can also do by clicking on the stoplight in the upper right corner, you can run the chart back and forth to look at historical data using the scroll arrows.

    When the HyperTrend cursor is activated you can use the cursor dialog box to search for maximum values, minimum values, or breaks in your data.

    You can also activate the HyperTrend chart by clicking on the pointer at the top center of the chart. When the cursor is active, this pointer disappears and is replaced by a vertical line cursor bisecting the chart. You can drag this cursor with your mouse. A readout denotes the time the data at the cursor location was taken.

14. Save your client process by selecting **File»Save**. Choose the `Client1` process to save. You should also save your server process if you have made any changes in it.

Leave both processes running for the next set of exercises.

Connecting a Control

In this section of the tour, you will create a control to manage the amplitude parameters of the waveform you are graphing. This section also introduces you to some of the basic concepts behind the way Lookout operates.

Because the waveform object is located in the server process, changing it so that the various parameters reference controls in your client process would involve making connections of the type that would break when you move the client process to another computer. To avoid having to edit your client, you must make your connection in a different way.

1. Launch Lookout if it is not already running, and open your server and client processes.

2. Make sure you are in edit mode.

3. If it is not already open, select **Objects»Object Explorer** to open the Lookout Object Explorer. Make Universe the root.
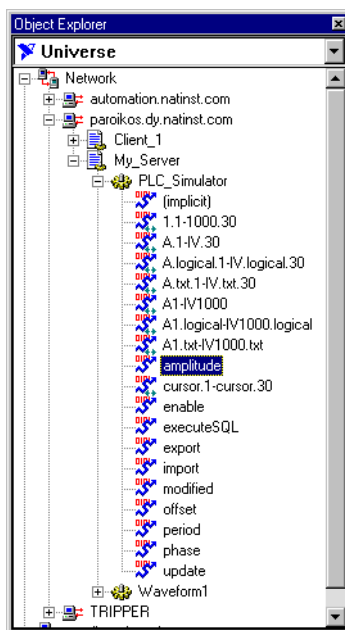
4. Open the Network node, and then open the branch featuring your computer name. Open the PLC_Simulator object in the My_Server process. Right-click on amplitude and drag it to the control panel for

Client 1. When you release the drag and drop icon, the following list of options appears.

Create Display
Insert Expression as Default
Insert Expression as Logical
Insert Expression as Numeric
Insert Expression as Text

Cancel
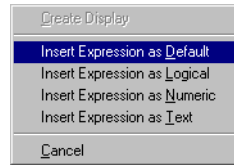
5. Select the **Insert Expression as Default** type. In this case the default type is numeric. Adjust the size so that the expression is large enough to hold a 3-digit number.

    By dragging this value from the network branch of the Object Explorer, you have created it with an absolute path to the amplitude value. Had you dragged the value from the local computer part of the Object Explorer tree, Lookout would have created a different path. The next series of steps briefly explores the difference.

    a. Right-click on the amplitude display and select **Object Properties**. The following dialog box appears.

Revise expression

\\paroikos.dy.natinst.com\My_Server\PLC_Simulator.amplitude

Display Type: default   Path Mode: Absolute
                                          Relative
Signals: Process Relative
                                          Computer Relative
Paste   \\paroikos.dy.natinst.com\Client_1   Absolute

Universe   Contents:

Network   Client_1
    automation.natinst.com   My_Server
    paroikos.dy.natinst.com
    TRIPPER
paroikos.dy.natinst.com

OK   Cancel   Help
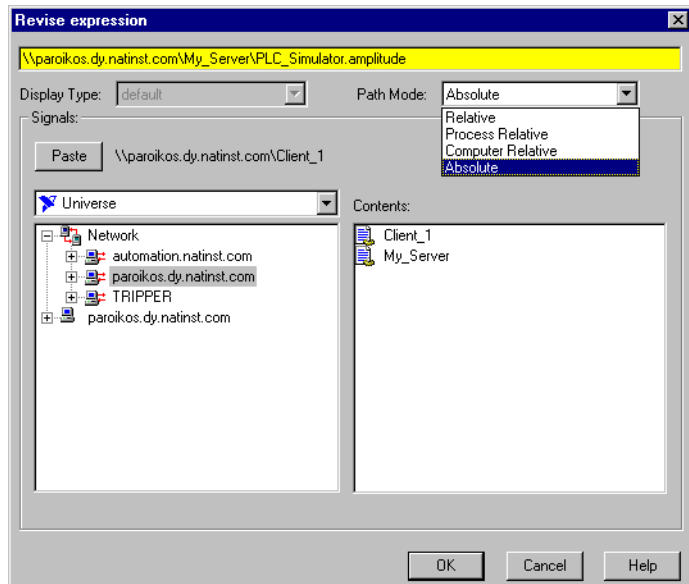
b.   Notice the path is

`\\Your_computername\My_Server\PLC_Simulator.amplitude`

If your path contains `\\.\` instead of `\\Your_computername\`, you may have dragged from the local expression of your computer in the Object Explorer instead of from the network expression. Before you can move your client to another computer successfully, you would have to edit or replace this expression.

c.   Click in the **Path Mode** listbox. Notice the four levels of path relativity listed. For detailed information on how paths work in Lookout, select **Help»Contents**. See the *Path Names in Lookout* section of Chapter 4, *Using Lookout*, for more detailed information on path modes.

A complete discussion of the power and flexibility of expressions is beyond the scope of this brief exercise, but there are at least two important things to know.

Any time you encounter a yellow field in any Lookout dialog box you have found an expression, and you can call the expression editor by right-clicking in that field. This makes it easy to find the particular data member you need for some connection or display purpose without having to remember every alias or data member name in all your processes.

The way you construct a path in Lookout makes a difference in how easily you can move your processes between computers. The path for the `amplitude` expression will work fine for your client whether it is running in the same instance of Lookout on the same computer as your server process, or any other instance of Lookout on any computer in your network. Other path modes look for data only in processes running on your computer, or in the process containing the expression, or in the folder holding the expression.

d.   Click on **Cancel** to close the expression editor.

6. Choose **Insert»Text/plate/insert** from the menu bar. The following dialog box appears.
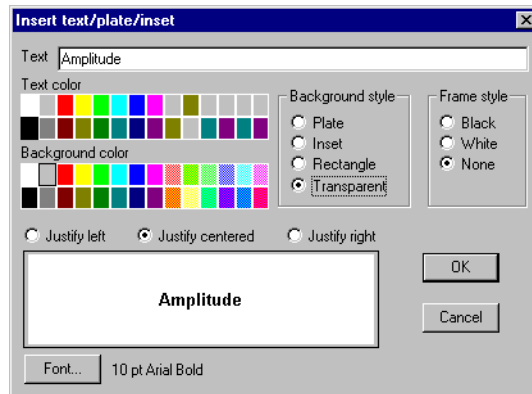
7. Enter Amplitude in the **Text** field, select **Transparent** in the **Background style** section, and click on **OK**.

8. Place your label next to the readout you created. Select both by shift-clicking or by lassoing both objects with your mouse cursor. Group them into one object by selecting **Arrange»Group** or pressing <Ctrl-G> on the keyboard.

    Labeling display objects as you go is generally a good idea. You can use options in the **Arrange** menu to line up, space, and group objects as you construct panels. Inserting graphics, labels, colored plates, and other decorative touches make your control panels more pleasing and easier to use.

    To help keep your panel design symmetrical, you can activate a snap to grid feature in Lookout by selecting **Edit»Snap to grid**. This feature aligns objects to a grid on the Lookout panel.

    You can adjust this grid by selecting **Edit»Options**.
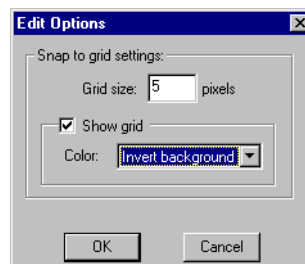
Use the dialog box that appears to set the distance (in pixels) between grid points, show or hide the grid, and select different color options for the grid display.

There will be no more explicit cosmetic steps included in this exercise, but you can create labels and experiment with other decorative objects as much as you like.

9. Drag and drop `phase`, `period`, and `offset` data members from your server process in the Object Explorer to your client control panel, and label each of them. You will need these readouts in the next exercise if you choose to do the extended tour.

   You are about to create a connection from a client to a server. There are at least two things you want to accomplish when you make such connections.

   One thing is that you want your client to be portable, so that you can run it on any computer in your Lookout network without modifying it.

   Another is that you want to make sure that your control will reflect the value of the simulated PLC register as well as control it. In other words, your control should connect to the register reciprocally—changing the control changes the value in the register, but if the register value changes from some other cause, the control will also change to reflect the new value. The control, instead of holding its local value, initializes its value from a remote source when the client is first opened, and responds to any changes in that remote source that occur while the client is running.

10. Reset your Object Explorer to have your computer as the root as shown in the following illustration. Make sure you select the Lookout Object Explorer's local representation of your computer, and not the network representation.

11. Right-click on your client process and select **New Object**. Select a Pot from the **Control** category. The **New Potentiometer** dialog box appears, as shown in the following illustration.
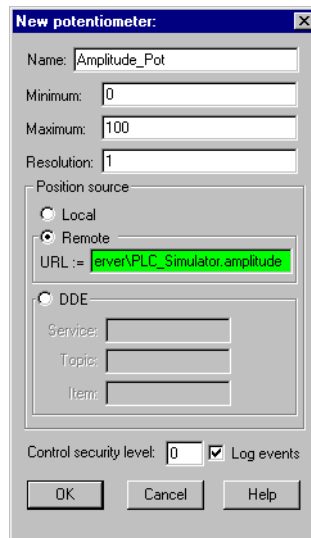
12. Name your new pot `Amplitude_Pot`. Set **Minimum** to `0`, **Maximum** to `100`, and **Resolution** to `1`.

13. Because you want this to be a reciprocal connection, select **Remote** in the **Position source** section. Right-click in the green **URL** field to open the URL editor.
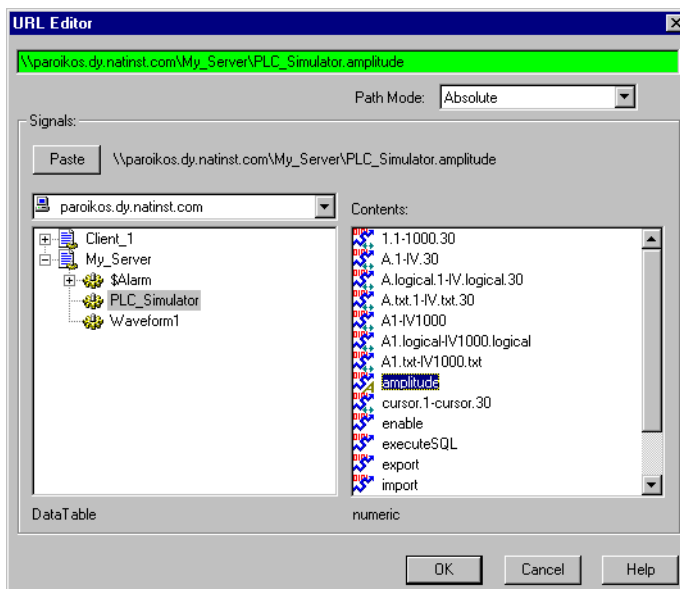
14. If you want, you can type the URL directly into the field, or you can construct the string.

    • To construct the string, set the **Path Mode** to `Absolute` and navigate to the PLC simulator your created in your server process. Select it, and then select `Amplitude` in the **Contents** list on the right. Click on **Paste** to put the complete path in the green field at the top.

    • Alternatively, type the full path in the green URL field as follows:

      `\\`*computername*`\My_Server\PLC_Simulator.amplitude`

      where *computername* is the fully qualified network name of your computer.

15. Click on **OK** to insert the URL into the **New Potentiometer** dialog box.

16. Click on **OK** in the **New Potentiometer** dialog box to finish creating the Pot. The new object appears in your Object Explorer under the `Client_1` branch of the tree.

To display the Pot, drag it to your Client 1 control panel. The dialog box that appears show you your display options.
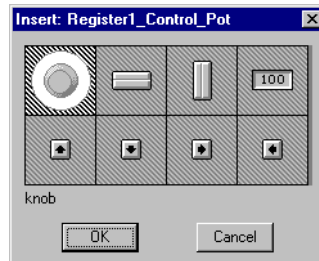
You can display this control as a knob, a slider, a numeric control you set with the keyboard, or increment/decrement buttons that change the value according to the setting you entered in the **Resolution** field when you created the control. Choose the knob for this example.

17. Press <Ctrl-Space> on the keyboard to leave edit mode, and operate the knob with your mouse. Notice that it changes the value of amplitude in your server, as displayed on your client control panel. Any control in any client that is connected to this same register will also reflect the change in value.

    This demonstrates the remote source type of Lookout connection. Direct connections, made using the **Edit Connections** dialog box, operate in one direction only, and should only be used to connect objects inside the same process. See the *Connecting Objects* section of Chapter 4, *Using Lookout*, for a more detailed discussion of remote source connections.

    You can also make complex connections using the Lookout Symbolic Link. These connections can change depending on the variables you incorporate in their construction.

    While you are changing the simulated PLC register value, your register is not connected to your waveform object. This is why the graph did not change when you operated the Pot. Making this connection is the next step.

18. Operate your control knob to set the amplitude to 0.

19. Make sure your HyperTrend object is actively tracing the waveform. The traffic light in the upper right corner should be green. If it is not, click on the stoplight icon to make sure the HyperTrend is plotting the current data.

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

21. Right-click on `Waveform1` in your server process and select **Properties**. The **Revise Waveform** dialog box appears.

22. In the **Amplitude** field, replace `45` with `PLC_Simulator.amplitude`.

23. Click on **OK**.

24. Notice that your graph immediately goes to a flat line. Press <Ctrl-Space> to return to edit mode.

25. If your alarm window is not visible, select **Alarms»Display Options** from the menu bar. Select the **Bottom** option from the **Alarm Display Options** dialog box. The alarm window appears at the bottom of your Lookout screen.

26. Set the Amplitude knob to `50` or greater. Your alarm system should notify you that you are now in an alarm range. If the alarm is making sounds, you can silence it by pressing <Ctrl-S> on the keyboard.

27. Reset the Amplitude knob to `40` or less to halt the alarms. Press <Ctrl-Space> to return to edit mode.

28. To acknowledge the alarms, right-click on one of the alarm messages in the alarm window, and select **Acknowledge All**. You can leave the comments field blank in the **Acknowledge** dialog box. Click on **OK** to clear the alarm window.

    For more information on the Lookout alarm system, consult the online documentation.

29. Save your client and server processes.

30. Select **Object»Connection Browser**. The following dialog box appears.
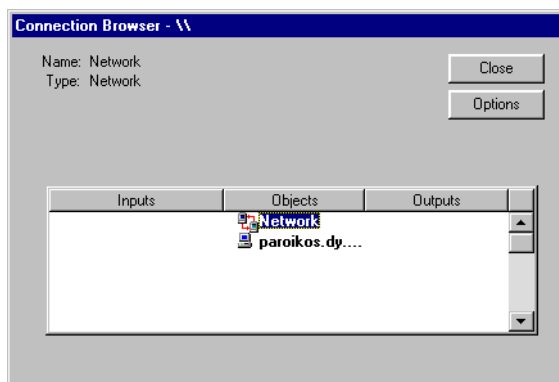
31. Use the Connection Browser to trace connections through your Lookout processes. Click on the object you are interested in, and then click on the element it is connected to in order to trace the connections throughout the process.

    You can also edit objects or create new objects by right-clicking on any element showing in the Connection Browser. The particular editing functions you can perform depend on which object you click on.

    For more information on the Connection Browser, consult the *Connection Browser* section of Chapter 3, *Lookout Basics: Windows, Tools, and Files*.
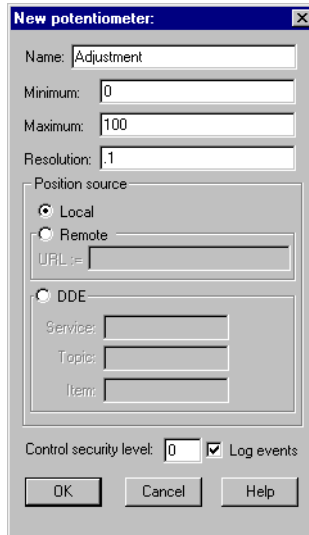
Note Lookout object parameters appear in the Connection Browser by parameter number, not parameter name. Unconnected data members do not appear at all. You cannot browse connections across the network, but you can see network connections and URLs.

32. (OPTIONAL) Experiment with your processes. Transfer a copy of the client process (all four files) to a second computer on your network that is running Lookout and observe how two copies of the same client interact, both in controlling your server process and responding to changes made from another location.

Create a Multiple-Use Control and Symbolic Link

The following instructions guide you through the process of using a Symbolic Link to create a multiple-use control. You can use this to control the rest of your waveform parameters. Make sure the server and client applications you created in the main exercise are running, and that you are in edit mode.
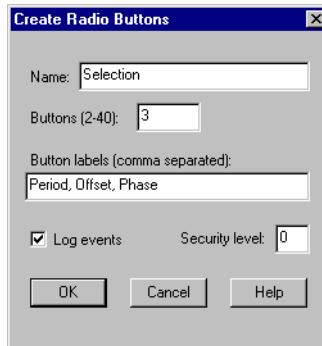
1. Right-click on Client_1 in the Object Explorer and create a new Pot object. Name it Adjustment. Enter parameter values as shown in the following illustration, and drag the new Pot to the control panel. This time, select a horizontal slider as the display.

2. Next, create a radio button, found in the **Control** category. Configure it as shown in the following illustration.

3. Click on **OK**. Drag the Selection radio button from the Object Explorer to your control panel. The **Create Display** dialog box appears.
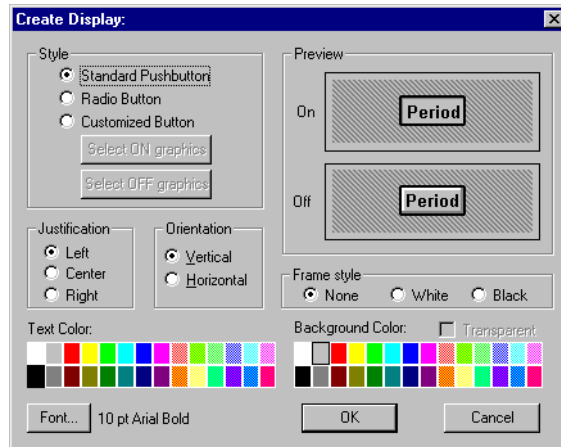
4. Choose **Standard Pushbutton**, and click on **OK**.

5. Earlier you created readouts for the values Period, Offset, and Phase. You can now move those readouts next to the appropriate button of the radio button control, and delete the labels (ungroup if necessary). One possible arrangement of the components you have created is shown in the following illustration, which also includes some plates (rectangles

of solid color) created for cosmetic purpose, using the
Insert»Text/plate/inset command from the menu bar.
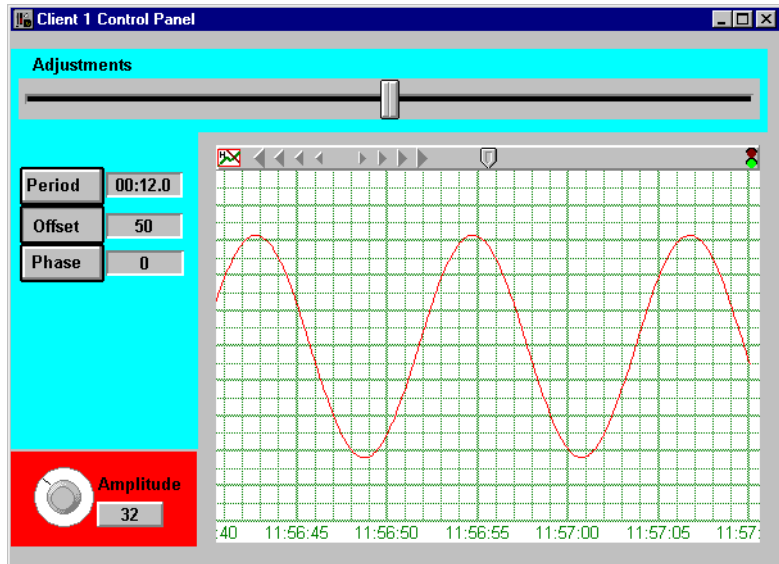
Use the radio button object you made to select which waveform
parameter you want to control, with the adjustment Pot supplying the
input. In the following steps you will modify the waveform object to
accept changes from your PLC simulator, and connect the controls.

6. Right-click on `Client_1` in the Object Explorer, and choose **New
Symbolic Link**. The **Create Symbolic Link** dialog box appears.
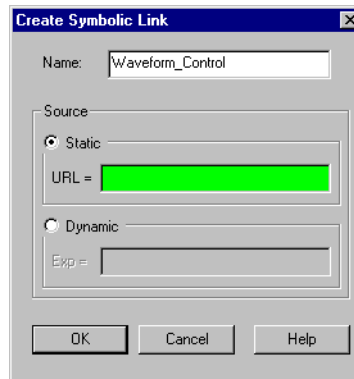
In this exercise you will use the Symbolic Link to create a multiple use
control. You might use a similar technique to control multiple PLCs

with one control panel, switching from PLC to PLC the way the control you are about to create switches from function to function.

You use a symbolic link any time you need to use a complex expression with a remote source connection. Other uses for the symbolic links include making clients flexible enough to switch the server they are using when a primary server goes down and a backup computer is brought online, or creating a shortcut connection to objects buried deeply in a hierarchy of folders and subfolders.

7. Name your new Symbolic Link `Waveform_Control`. Select **Dynamic** as the **Source**. Right-click in the **Dynamic** field to enter the expression editor.
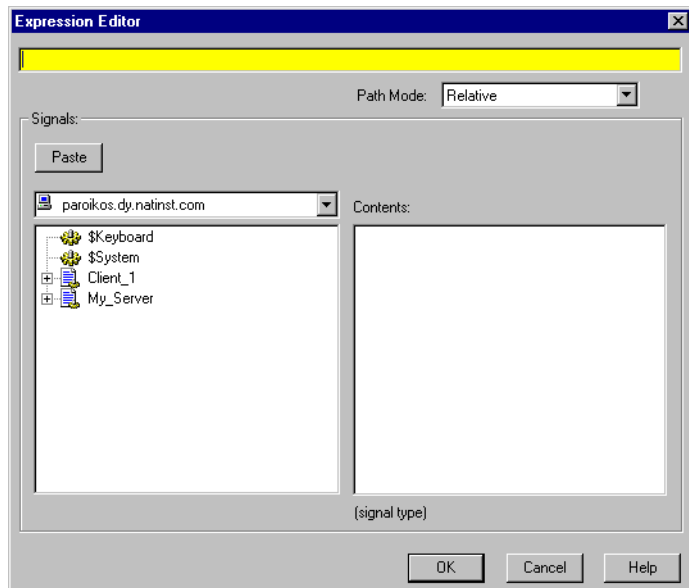
8. There are two different ways to create the expression you need to insert in the **Dynamic** field. You can manually enter the following string (line returns added for clarity), or you can also create this expression in the expression editor, following the directions in step 9.

```
TCHOOSE (Selection,
"\\your_computer_name\My_Server\PLC_Simulator.A4",
"\\your_computer_name\My_Server\PLC_Simulator.offset",
"\\your_computer_name\My_Server\PLC_Simulator.phase")
```

As you can see, even though it is possible to type such an expression directly, you may find it difficult to remember all the details of a path,

or even the precise name of objects and data members you want to include in such a complex statement. This is why Lookout includes the expression editing feature, to make keeping track of multiple objects and data members at the end of complex paths easier to do.

Tip Notice that the expression in step 8 includes the data member `PLC_Simulator.A4`. This is the data member you configured earlier for scaling. To use a data table to scale a value, you have to write the raw data stream to the native data member and read it from the scaled, aliased version.

9. To construct the string shown in step 8, using the Expression Editor, right-click in the yellow **Dynamic** field. The **Expression Editor** dialog box appears.

    If you typed the expression in directly, you can skip the following lettered steps.

    a. Enter `TCHOOSE` in the yellow expression field. This is a Lookout function that accepts a numeric input and returns a text string depending on that value. The syntax for this expression is `TCHOOSE (numeric, "text1", "text2", "text3",...)`. For more detailed information on using expressions in Lookout 4, consult the online help *Expressions* topic.
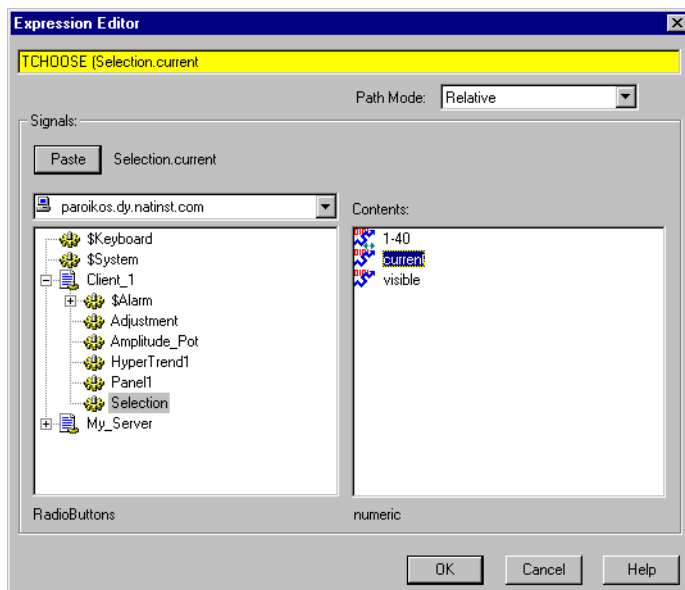
b. Insert an open parenthesis after TCHOOSE. You may use as many spaces for clarity as you like between terms of this expression; Lookout does not read spaces except in a designated text string.

c. Navigate to the Selection Pot and select the current data member. Leave the **Path Mode** set to relative. **Paste** the line into your expression.

d. Insert a comma (,) and then a double quote (") after the Selection term.
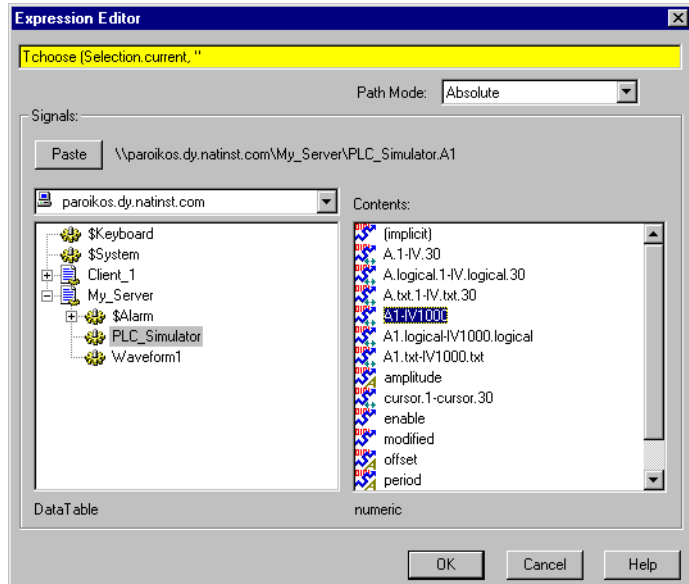
e. Navigate to the PLC simulator in your server process, and set the **Path Mode** to Absolute. Select the data members indicated by the range A1-IV1000 and **Paste** the data member into the expression. It will appear in the yellow field above as
    your_computer_name\My_Server\PLC_Simulator.A1.
    Change A1 to A4.

f.   Add double quotes (**"**) and a comma (**,**) at the end of the string.
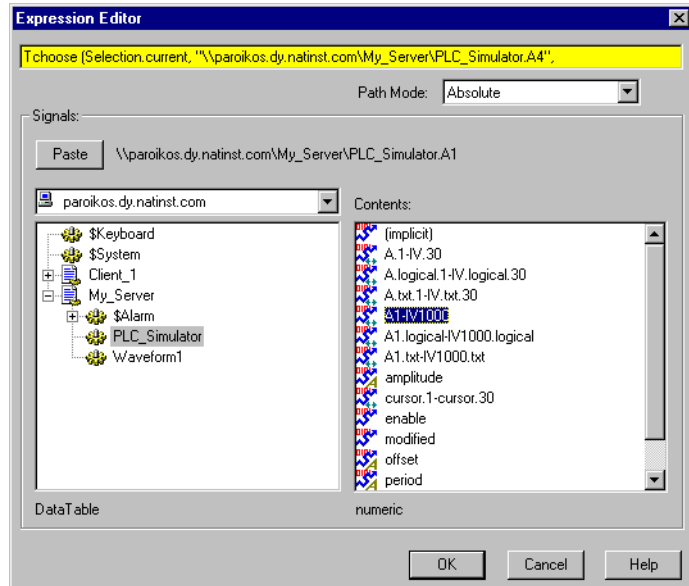
g.   Copy the string you just edited by highlighting it, including quotes and the final comma, and pressing <Ctrl-C> on the keyboard.

h.   Paste the copied string in place by pressing <Ctrl-V> on the keyboard, then edit the pasted string to change A4 to offset. Repeat this operation a second time, this time changing A4 to phase.

i.   Insert a close parenthesis at the end of the string you have constructed.

j.   Check the string to make sure all the elements are correct, as shown in step 8.

k.   Click on **OK** to paste the complete string into the **Dynamic** expression field of your Symbolic Link. If an error message appears, check your string against the full string in step 8 again.

10. Click on **OK** to finish creating your Symbolic Link.

11. Right-click on the `Waveform1` object in your server and select
    Properties. The **Revise Waveform** dialog box appears.
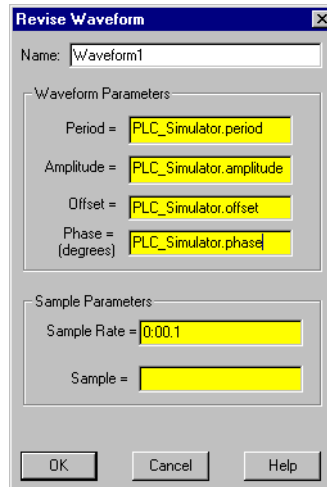
12. Enter `PLC_Simulator.period` for the **Period**,
    `PLC_Simulator.offset` for the **Offset**, and
    `PLC_Simulator.phase` for the **Phase (degrees)**.

    (Remember that you already scaled the `period` data member alias
    earlier in these exercises. This is the appropriate place to use that
    scaled value.)

13. Click on **OK**.

    The first thing you will likely notice is that the HyperTrend graph is
    looking very different. The last step is to connect the Adjustment Pot.

14. Right-click on your Adjustment Pot, and select **Properties**. Select the
    Remote option in **Position source**, and enter [Waveform_Control]
    in the green URL field as shown in the following illustration.
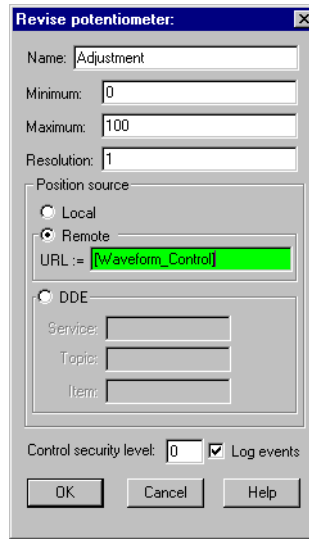
15. Click on **OK**.

16. Operate your controls. Begin by setting **Period** to 12, **Amplitude** to
    45, and **Offset** to 50. This should halt the alarms.

Tip   You can import data from a spreadsheet into a data table object. If you have one or
more spreadsheet files with data for different waveform settings, you can connect a control
to the DataTable import data member and import a complete set of values for the waveform
object parameters with the touch of a button. Using a text entry object or radio button, you
can change which spreadsheet file you import, creating a fast and easy way to reconfigure
the parameters of any Lookout object.

17. Right-click in the alarm window and select **Acknowledge All**. You can
    leave the comments box that appears blank.

18. Save your process files.

Conclusion

There are only a few basic principles to learn if you want to master Lookout—the most important being the paired concepts of objects and connections. Once you understand what objects are, how they behave, and how to make connections, you should be able to quickly learn Lookout and begin building your application.

You do not need to memorize every object class in Lookout, but you should scan their definitions and data members to familiarize yourself with their functionality. You can also consult Appendix A, *Lookout Object Descriptions*, for a capsule description of each object class.

To review, an object bundles functionality with an integrated database. An object has parameters that define how it works. You define these parameters when you create or modify the object.

In the predefined object database, the data members can be numeric, logical, or textual. Data members can generate signals (that is, they are readable), receive signals (that is, they are writable), or both.

The parameters and data members are the connection points for getting objects to work together and accomplish the things you want to have done.

As a final analogy, think of the many object classes in Lookout as a comprehensive set of tools. Each instance of an object is designed to solve a particular problem. However, you must open the toolbox (documentation) to learn what is available, in order to become an efficient developer. In addition to the detailed information on using the features and functions of Lookout in your *Lookout Developer's Manual*, you can also browse through the Lookout object classes using the online help browse buttons (**»** to move to the next topic; **«** to move to the previous topic); along with their capsule descriptions in Appendix A, *Lookout Object Descriptions*.

A

Lookout Object Descriptions

This appendix lists descriptions of the Lookout system objects and Lookout driver objects.

You build Lookout processes by creating Lookout objects and by making select connections. Lookout has two kinds of objects: Lookout system objects, which you use for control, monitoring, data analysis, and data presentation, and Lookout driver objects, which you use to connect to hardware, such as PLCs (Programmable Logic Controllers), RTUs (Remote Terminal Units), computer boards, and other hardware.

Lookout System Objects

The following table lists the Lookout system objects, along with their categories and a description of what each one does.

Table A-1. Lookout System Objects

Object Class	Category	Description
Accumulator	Calculation, Control	Accumulate a running total by adding a sampled value to the previous total each time a sample trigger fires. A reset trigger is also available.
Alarm	Logging	Define specific alarm conditions that can trigger alarms via complex conditions, or as a result of numeric inputs going out of range or based on rate of change. Attach audible (.wav) files to these alarms that play back whenever the alarm conditions is triggered.
Alternator	Control	Control the orderly alternation of a set of pumps or other devices.
Animator	Display	Dynamically animate graphics files through any combination of horizontal and vertical motion, resizing, visibility, and color changes. Play consecutive filmstrip images from a bitmap to simulate complex motions on the screen.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
Average	Calculation, Control	Calculate the average of a variable input over time. Optionally control when averaging is active via the Enable input, and reset the average to the current input via the Reset input. The results of up to 35 previous averages are available, and the current average is updated approximately once per second.
Counter	Control, Logging	Count and record the number of times a logical signal turns on. Optionally reset the count with the Reset input. Maximum count is approximately 2^52.
DataTable	Control, Networking	Multiplex multiple data sets into a single control panel by allowing the operator to alter the table cursor setting. Import and export entire tables of data between other applications via DDE and NetDDE. A DataTable consists of a matrix of cells arranged into rows and columns.
DdeLink	Networking	Import a single live value into Lookout from another application via DDE or NetDDE.
DdeTable	Networking	Import a table of values into Lookout from another application via DDE or NetDDE. DdeTable is a subset of the DataTable object class which allows the bidirectional flow of DDE and NetDDE data.
DelayOff	Control, Timers	Introduce a variable time delay between the time the input signal turns off and this timer turns off.
DelayOn	Control, Timers	Introduce a variable time delay between the time the input signal turns on and this timer turns on.
Derivative	Calculation, Control	Calculate the rate of change of a numeric signal, specifying the time unit.
DialGauge	Display	Display a numeric signal as a sweeping dial. Specify the starting position and angle of sweep.
ElapsedTime	Control, Logging, Timers	Calculate and record the cumulative elapsed time that a particular signal has been on. Optionally use the Reset input to reset the elapsed time at any time.
Event	Logging, Monitoring	Permanently record the occurrence a particular event to the event log.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
Flipflop	Control	Change the state of this object from on to off, or from off to on each time the input signal goes high.
Gauge	Display	Display a numeric value as a digital value or as a bar graph, and change the color of the bar or digits based on any set of conditions.
Histogram	Calculation, Display, Statistical Process Control	Calculate and display the distribution of a sampled signal. Also display the central tendency and variability of the sample set. Calculate the process capability ratio (PCR), confidence interval, mean, variance, and standard deviation.
HyperTrend	Display, Logging	Display and peruse historical data as a moving chart, and zoom in on data of interest or search the Citadel historical database for maxima, minima, and other points of interest. Scroll back in time, or leap to any specific date and time instantly.
Integral	Calculation, Control	Totalize a numeric input signal that represents a rate with respect to time, specifying the time unit. Totalizing the flow based on a measured flow rate is a typical use of the Integral.
Interpolate	Calculation, Control	Perform a linear interpolation between a set of up to 1000 X/Y coordinate pairs. Specify a particular X and get the corresponding Y and vice versa. Simultaneously calculate up to 1000 interpolations with each Interpolate object.
Interval	Control, Timers	Introduce a variable time delay between the time the input signal turns on and this timer turns off. The timer turns on immediately when the input turns on, and remains on for the timer duration or until the input signal goes low.
Junction	Control	Select and output the value of the last changed numeric variable from a set of up to nine variable input signals.
L3OPCClient	Obsolete	Obsolete Lookout OPC client. Use OPC client version 4.0 instead. This class is included for compatibility.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
L3Pot	Obsolete	A Lookout 3.8-style potentiometer to track and/or alter the value of a numeric variable. A Pot can be displayed on a panel as a slider, rotating knob, or digital display/input. This class is included for backward compatibility with Lookout 3.*xx*.
L3Pushbutton	Obsolete	A Lookout 3.8-style pushbutton to navigate between control panels within Lookout, or to initiate any other action, such as executing an external application, or latching a pump on or off. This class is included for backward compatibility with Lookout 3.*xx*.
L3Switch	Obsolete	A Lookout 3.8-style toggle switch to track and/or alter the value of a logical variable. A Switch can be displayed on a panel in a variety of ways. This class is included for backward compatibility with Lookout 3.*xx*.
L3TextEntry	Obsolete	A Lookout 3.8-style text-entry object to track and modify a text value via a control panel, using either the computer keyboard or a keyboard display on the screen. This class is included for backward compatibility with Lookout 3.*xx*.
LatchGate	Control	Latch the output of this object on or off based on whichever of two input signals last went high.
Loader	Control	Load or unload a process file when input goes high.
Maximum	Calculation, Control	Calculate the maximum of a variable input over time. Optionally control when maximum monitoring is active via the Enable input, and reset the maximum to the current value via the Reset input. The results of up to 35 previous maxima are available.
Minimum	Calculation, Control	Calculate the minimum of a variable input over time. Optionally control when minimum monitoring is active via the Enable input, and reset the minimum to the current value via the Reset input. The results of up to 35 previous minima are available.
Monitor	Networking	Monitor a single network I/O point and output true if the data quality is bad.
Multistate	Display	Display a unique graphic for each of up to six different conditions.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
Neutralzone	Control	Control the on/off status of a signal based on an incoming numeric signal, and based on two potentially variable upper and lower control limits. Use a Neutralzone to control whether a pump is running based on the varying level in a water storage tank, for example.
OneShot	Control, Timers	Introduce a variable time delay between the time the input signal turns on and this timer turns off. The timer turns on immediately when the input turns on, and remains on for the timer duration, and then turns off irrespective of the input signal.
Pager	Monitoring	Call a numeric or alphanumeric pager and automatically deliver messages based on specified conditions or alarms in your process.
Pareto	Calculation, Display, Statistical Process Control	Calculate and display a frequency distribution of a set of occurrences sorted by frequency. With a Pareto, one can identify and rank the most commonly occurring anomalies in a system.
PID	Control	Control a closed-loop process using the classical proportional/integral/derivative approach. PID includes bumpless transfer, integral windup protection and both the positional and velocity forms of control.
Pipe	Display	Display rectangles on the screen, and control their colors based on a set of up to six different conditions.
PlayWave	Display	Play a standard Windows audio (`.wav`) clip based upon the satisfaction of a specified condition.
Pot	Control, Display	Create a potentiometer to track and/or alter the value of a numeric variable. A Pot can be displayed on a panel as a slider, rotating knob, or digital display/input.
Pulse	Control, Timers	Create a periodic pulse train of a specified and variable period and duration. The pulse train can be turned on and off based on the state of the On/Off signal.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
Pushbutton	Control, Display	Create a pushbutton to navigate between control panels within Lookout, or to initiate any other action, such as executing an external application, or latching a pump on or off.
RadioButtons	Control, Display	Create a set of buttons only one of which can be active at a time. The radio buttons can be displayed in a variety of ways.
Recipe	Control, Display	Create a set of recipes in a spreadsheet application such as Microsoft Excel and import them into a Recipe to allow the user to select a particular recipe. Recipe supports any number of ingredients—each column in the spreadsheet represents a specific ingredient.
Run	Control	Invoke a Windows application or DOS batch command with a variable set of command line parameters, whenever the **Run When** input signal turns on.
Sample	Calculation, Control	Take a sample of a numeric variable whenever the Reset input goes high. Optionally control when sampling is active via the Enable input, and reset first sample to the current value via the Reset input. The results of up to 35 previous samples are available.
SampleText	Calculation, Control	Take a sample of a text variable whenever the Reset input goes high. Optionally control when sampling is active via the Enable input, and reset first sample to the current value via the Reset input. The results of up to 35 previous samples are available.
Scale	Display	Create a dynamic scale where the range and minimum/maximum can change dynamically based upon variable inputs.
Sequencer	Control	Create a sequencer object that generates a sequence of states with associated outputs. The sequencer will remain in the current state until that state's time limit is reached or until a Goto or Jump input causes it to transition to another state.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
Spinner	Display	Display a small rotating disk that rotates at a speed that varies according to a numeric input value. A Spinner can spin in both directions and can also be controlled by a simple logical input.
Spreadsheet	Logging	Permanently log data to disk in industry-standard comma-separated-value (`.csv`) spreadsheet files. Specify which variable to sample in each column and set up a logical trigger that will log the set of data each time the trigger goes high or log data at specified time intervals.
SQLExec	Drivers, Logging	Perform SQL queries and log data to ODBC-compliant databases. SQLExec accepts complex SQL statements, and automatically buffers logged records if connection is temporarily lost to the database.
Switch	Control, Display	Create a toggle switch to track and/or alter the value of a logical variable. A Switch can be displayed on a panel in a variety of ways.
TextEntry	Control, Display	Allow the operator to track and modify a text value via a control panel, using either the computer keyboard or a keyboard display on the screen.
TimeOfDay	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each day. This time can be enabled and disabled via the On/Off input signal.
TimeOfHour	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each hour. This time can be enabled and disabled via the On/Off input signal.
TimeOfMinute	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each minute. This time can be enabled and disabled via the On/Off input signal.
TimeOfMonth	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each month. This time can be enabled and disabled via the On/Off input signal.
TimeOfWeek	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each week. This time can be enabled and disabled via the On/Off input signal.

Table A-1. Lookout System Objects (Continued)

Object Class	Category	Description
TimeOfYear	Control, Timers	Schedule a periodic pulse of a specified duration that will occur at some offset into each year. This time can be enabled and disabled via the On/Off input signal.
Waveform	Calculation	Create sine, square, saw tooth, triangle, and random waveforms with specified values for the period, amplitude, offset, phase, and sampling rate.
XBarR	Calculation, Display, Statistical Process Control	Calculate and display an X-Bar chart and/or an R chart based upon a specified sampled signal.
XChart	Calculation, Display, Statistical Process Control	Display an X chart based upon a specified sampled signal.
XYChart	Display, Statistical Process Control	Collect X/Y coordinate pairs from two sampled signals and graph the resultant set of points. Specify the maximum number of points to be sampled and graphed.

Lookout Driver Objects

The following table lists the Lookout driver objects, along with their categories and a description of what each one does.

Table A-2. Lookout Driver Objects

Object Class	Category	Description
AB_PLC2	Drivers	Communicate with Allen-Bradley PLC2 line of programmable controllers via serial communications.
AB_PLC5	Drivers	Communicate with Allen-Bradley PLC5 line of programmable controllers via serial port, Data Highway Plus (KT, KTx, PCMK, S-S 5136SD), and TCP/IP Ethernet.

Table A-2. Lookout Driver Objects (Continued)

Object Class	Category	Description
AB_SLC500	Drivers	Communicate with Allen-Bradley SLC5*xx* line of programmable controllers via serial port, Data Highway Plus (KT, KTx, PCMK, S-S 5136SD), and DH485 (AB1747-KE, AB1770-KF3, KTx, PCMK).
AdvantechPCL	Drivers	Communicate with the Advantech PC-Multilab card, Model 711B.
Applicom_Local	Drivers (Applicom)	Communicate with the local mode interface for any device that can communicate with an Applicom interface card.
April1000 (Applicom)	Drivers (Applicom)	Communicate with a Schneider April PLC using an Applicom card.
Aquatrol	Drivers	Communicate with Aquatrol W1500 controllers via the serial port.
ASCII	Drivers	Communicate with any device that transmits and accepts formatted strings of ASCII data. Extract numeric, logical and text values from messages and incorporate them into a Lookout process. For example, one can use the ASCII object class to read the output from a barcode scanner.
CutlerHammer	Drivers	Communicate with Cutler-Hammer PLCS using serial communications.
DeltaTau	Drivers	Communicate with the Delta Tau Data Systems PMAC Motion Controller cards via dual-ported memory.
DL205	Drivers	Communicate with Koyo, PLCDirect and Siemens Series 205 programmable controllers. This driver supports the GE CCM protocol, which is called DirectNet by PLCDirect and HostLink by Siemens, and it implements both the N-sequence and K-sequence versions of this protocol.
DL405	Drivers	Communicate with Koyo, PLCDirect and Siemens Series 405 programmable controllers. This driver supports the GE CCM protocol, which is called DirectNet by PLCDirect and HostLink by Siemens, and it implements both the N-sequence and K-sequence versions of this protocol.

Table A-2. Lookout Driver Objects (Continued)

Object Class	Category	Description
Dynamic	Drivers	Communicate with RTUs and other equipment that supports the Siemens Dynamic RTU protocol. This driver was originally written to communicate with the Texas Instruments 8635, 8640 and 8641 RTU.
FieldPoint	Drivers	Communicate with National Instruments FieldPoint Modular Distributed I/O system via RS-232 or RS-485 serial communications or Ethernet.
FisherROC	Drivers	Communicate with a Fisher Controls ROC364 (Remote Operations Controller) via serial communications.
GE_Series90	Drivers	Communicate with GE Series 90-30 and 90-70 programmable logic controllers via RS-232 using the SNPX serial protocol, or via Ethernet using the GE TCP/IP communication module.
Hitachi	Drivers	Communicate with a Hitachi H-Series Programmable controller via RS-232 and RS-422 serial ports.
IPASCII	Drivers	Communicate with any device that transmits and accepts formatted strings of ASCII data via the TCP/IP UDP or IP protocol. Extract numeric, logical and text values from messages and incorporate them into a Lookout process. For example, one can use the ASCII object class to read the output from an Ethernet barcode scanner.
JBUS (Applicom)	Drivers (Applicom)	Communicate with a JBUS network using an Applicom card.
Klockner-Moeller (Applicom)	Drivers (Applicom)	Communicate with a Klockner-Moeller PLC using an Applicom card.
Mitsubishi	Drivers	Communicate with Mitsubishi A-Series PLCs via the serial port.
MitsubishiFX	Drivers	Communicate with Mitsubishi FX-Series PLCs via the serial port.

Table A-2. Lookout Driver Objects (Continued)

Object Class	Category	Description
Modbus	Drivers	Communicate with any device that supports the Modbus serial protocols, has a Modbus+ port, or has a Quantum Ethernet module. Modbus+ requires the Modicon Modbus Plus network card and support software to be installed in the computer. The serial protocol communicates via both the ASCII and RTU protocols.
ModbusMOSCAD	Drivers	Communicate with Motorola MOSCAD PLCs and RTUs via the Modbus serial protocols. Supports the Motorola data naming convention.
ModbusSlave	Drivers	Allow other devices or computers to read and write variables in your process by polling a serial port on your computer with the Modbus RTU protocol.
NIDAQDevice	Drivers	Communicate with National Instruments data acquisition cards and SCXI hardware connected in parallel mode. Use the NISCXI object class to communicate with SCXI hardware connected in multiplex mode. This object class requires NI-DAQ 5.0 software or better to be installed.
NISCXI	Drivers	Communicate with National Instruments data acquisition devices connected in multiplex mode to SCXI hardware. This object class requires NI-DAQ 5.0 software or better to be installed.
Omron	Drivers	Communicate with Omron PLCs using the Host Link serial protocol.
OPCClient	Drivers	Communicate with OPC (OLE for Process Control) servers. Use the shortcut objects OPCFieldPoint and OPCNIDAQ to make OPC connections to National Instruments FieldPoint modules and data acquisition boards.
OptoHostwords	Drivers	Communicate with any device that uses the Opto22 Hostwords protocol.
OptoMistic	Drivers	Communicate with any device that supports the Opto22 Mistic protocol via a serial link.
Optomux	Drivers	Communicate with any device that supports the Optomux serial communications protocol. This includes, of course, the Opto 22 I/O.

Table A-2. Lookout Driver Objects (Continued)

Object Class	Category	Description
OticFP (Applicom)	Drivers (Applicom)	Communicate with an Otic Fischer & Porter device using an Applicom card.
Philips	Drivers	Communicate with Philips devices using the PPCCOMM serial communication protocol.
ProfibusDP	Drivers	Communicate with Profibus DP slave PLCs and I/O modules with the S-S Technologies 5136-PFB PC card.
ProfibusDP (Applicom)	Drivers (Applicom)	Communicate with a Profibus DP network using an Applicom card.
ProfibusFMS (Applicom)	Drivers (Applicom)	Communicate with a Profibus FMS network using an Applicom card.
ProfibusL2 (Applicom)	Drivers (Applicom)	Communicate with a Siemens S5 PLC via a Profibus L2 network using an Applicom card.
ProfiSinecL2	Drivers	Communicate with PLCs using the Profibus Sinec-L2 protocol with the S-S Technologies 5136-PFB PC card.
Reliance	Drivers	Communicate with Reliance Automate Controllers using a Reliance PC-Link card.
RKC_FSeries	Drivers	Communicate with RKC Instrument Inc. New F-series digital loop controllers.
S5_3964	Drivers	Communicate with Siemens S5 PLCs via a serial port and the 3964 or 3964R protocol.
S5_AS511	Drivers	Communicate with Siemens S5 PLCs through serial programming ports using the AS511 protocol.
S7MPI (Applicom)	Drivers (Applicom)	Communicate with a Siemens S7 PLC via MPI using an Applicom card.
S7PPI (Applicom)	Drivers (Applicom)	Communicate with an S7 PLC via the PPI interface using an Applicom card.
SAIA-SBus (Applicom)	Drivers (Applicom)	Communicate with an SAIA S-Bus device using an Applicom card.
SiemensH1 (Applicom)	Drivers (Applicom)	Communicate with a TI500 PLC via the Siemens H1 NIM using an Applicom card.
SiemensH1S5 (Applicom)	Drivers (Applicom)	Communicate with a Siemens S5 PLC via the Siemens H1 NIM using an Applicom card.

Table A-2. Lookout Driver Objects (Continued)

Object Class	Category	Description
SiemensS5_AS511 (Applicom)	Drivers (Applicom)	Communicate with a Siemens S5 PLC via the AS511 serial protocol using an Applicom card.
SiemensTI505	Drivers	Communicate with a Siemens TI505 PLC via the Siemens H1 Ethernet protocol. This driver requires the Siemens CP1413 Ethernet communications card and associated Siemens software, and the Siemens CP1431 NIM for the PLC.
Sixnet	Drivers	Communicate with Sixnet IOMUX RTUs, Versamux RTUs, and DIN-rail-mounted Sixtrac I/O modules via Sixnet's Control Room software. Includes provisions for importing the Sixtags database.
SquareD	Drivers	Communicate with SquareD PLCs via a SY/MAX card, SY/ENET, or just over the serial port.
Telemecanique (Applicom)	Drivers (Applicom)	Communicate with a Telemecanique PLC using an Applicom card.
Tesco	Drivers	Communicate with the Tesco family of Liquitronic LIQ programmable controllers, including the LIQ4 and LIQ5.
Tiway	Drivers	Communicate with the Siemens TI5x5 line of PLCs and other equipment via the serial port using the TBP or NITP protocol, via the TIWAY card or Unilink Host Adapter, or via the CTI or Siemens TCP/IP Ethernet NIM card. Import APT tag files once a Tiway object is created via the **Options»Import APT database** menu command.
ToshibaMseries	Drivers	Communicate with Toshiba M-Series Ex 100, M20 and M49 controllers using the Host Link serial communication protocol.
ToshibaTseries	Drivers	Communicate with Toshiba T-Series T1, T2 and T3 controllers using the Host Link serial communication protocol.
Wizdom	Drivers	Communicate with the dual-ported RAM on the Wizdom Coprocessor Card.

B

Technical Support Resources

This appendix describes the comprehensive resources available to you in the Technical Support section of the National Instruments Web site and provides technical support telephone numbers for you to use if you have trouble connecting to our Web site or if you do not have internet access.

NI Web Support

To provide you with immediate answers and solutions 24 hours a day, 365 days a year, National Instruments maintains extensive online technical support resources. They are available to you at no cost, are updated daily, and can be found in the Technical Support section of our Web site at `www.natinst.com/support`.

Online Problem-Solving and Diagnostic Resources

- **KnowledgeBase**—A searchable database containing thousands of frequently asked questions (FAQs) and their corresponding answers or solutions, including special sections devoted to our newest products. The database is updated daily in response to new customer experiences and feedback.

- **Troubleshooting Wizards**—Step-by-step guides lead you through common problems and answer questions about our entire product line. Wizards include screen shots that illustrate the steps being described and provide detailed information ranging from simple getting started instructions to advanced topics.

- **Product Manuals**—A comprehensive, searchable library of the latest editions of National Instruments hardware and software product manuals.

- **Hardware Reference Database**—A searchable database containing brief hardware descriptions, mechanical drawings, and helpful images of jumper settings and connector pinouts.

- **Application Notes**—A library with more than 100 short papers addressing specific topics such as creating and calling DLLs, developing your own instrument driver software, and porting applications between platforms and operating systems.

Software-Related Resources

- **Instrument Driver Network**—A library with hundreds of instrument drivers for control of standalone instruments via GPIB, VXI, or serial interfaces. You also can submit a request for a particular instrument driver if it does not already appear in the library.

- **Example Programs Database**—A database with numerous, non-shipping example programs for National Instruments programming environments. You can use them to complement the example programs that are already included with National Instruments products.

- **Software Library**—A library with updates and patches to application software, links to the latest versions of driver software for National Instruments hardware products, and utility routines.

Worldwide Support

National Instruments has offices located around the globe. Many branch offices maintain a Web site to provide information on local services. You can access these Web sites from `www.natinst.com/worldwide`.

If you have trouble connecting to our Web site, please contact your local National Instruments office or the source from which you purchased your National Instruments product(s) to obtain support.

For telephone support in the United States, dial 512 795 8248. For telephone support outside the United States, contact your local branch office:

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011, Canada (Calgary) 403 274 9391, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521, China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24, Germany 089 741 31 30, Greece 30 1 42 96 427 Hong Kong 2645 3186, India 91805275406, Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625, Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00, Singapore 2265886, Spain (Barcelona) 93 582 0251, Spain (Madrid) 91 640 0085, Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

Glossary

Prefix	Meanings	Value
μ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}

A

absolute date
absolute time

Numeric system Lookout uses for keeping track of dates and times, in which midnight (0 hours), January 1, 1900 is represented by 1, midnight of January 2, 1900 is represented by 2, and so on. The absolute date/time number 36234.47222250 represents 11:20 AM, March 15, 1999.

The numeric value for 1 second in Lookout is .000011574, the numeric value for 1 minute is .000694444, and the numeric value for 1 hour is .041666667.

ACK

Acknowledge (an alarm or event).

active notification

A feature of event-driven software systems in which the application is alerted of value changes when they occur instead of through continuous, loop-driven queries.

address space

An OPC term for the area you browse to find what items are available on an OPC server. Part of the standard OPC interface, this space may arrange items hierarchically.

alarm

Software notification of a condition in a process. This alarm may call attention of a value that has exceeded or fallen below certain levels, set in the object database or in an Alarm object.

alias

Name given to a data member using the **Edit Database** dialog box. This name can be descriptive or mnemonic, and may be associated with other data member configurations such as scaling, logging, and alarming. A data member can have more than one alias, each with different associated configurations.

B

baud rate	Measurement of data transmission speed, formally defined as the number of electronic state changes per second. Because most modems transmit four bits of data per change of state, is sometimes misused or misunderstood—a 300 baud modem is moving 1200 bits per second. *See* BPS.
.bmp files	Graphic files in bitmap format. If you are using a .bmp file in Lookout, you cannot resize it on screen. *See* Windows metafile.
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 class. A .cbx (Control Block Extension) file may have one or more object classes in it.
checksum	A method of verifying that the number of bits received is the same as the number of bits transmitted. Used by TCP/IP and serial protocols.
Citadel	The Lookout historical database that stores your data for access later.
classes	*See* object classes.
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. Lookout server processes run on computers actually connected to your control hardware.
comm port	Term sometimes used for a serial port.
connection	Input to a Lookout object's writable data members. For more information, refer to Chapter 4, *Using Lookout*.
control objects	Lookout objects you use to control a process, change a data value, adjust a register, and so on.
controllable objects	Lookout objects you can control with a Lookout control object.

.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. See the *RTS/CTS Handshaking Settings* section of Chapter 3, *Serial Port Communications*, in the *Lookout Developer's Manual* for detailed information.
cursor (data table)	The Lookout data table can activate one row of data at a time using the data table cursor. See the data table reference in the Lookout online help or the *Lookout Object Reference Manual*.

D

DAQ	Short for Data AcQuisition.
data member	Data source or sink associated with a Lookout object. A readable data member, or source, may be used in expressions or as inputs to other objects. A writable data member, or sink, may have at most one connection into it, created using the **Object»Edit Connections** dialog box. A data member may be both readable and writable. *See also* native data member *and* 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 (running Windows 98/95 or Windows NT).
DDE	Dynamic Data Exchange, currently used in Lookout to exchange data with other programs (such as Microsoft Excel) running on your network.

deadband	A value that must be exceeded for an alarm to sound or a change in state to be recorded. For instance, if you have a low-level alarm set at 5 with a deadband of 2, the alarm will not trigger until the value being monitored drops to 5. The alarm will then stay active until the value being monitored moves above 7. A deadband keeps small oscillations of value from triggering an alarm and then canceling it too rapidly.
deviation	Set a deviation to filter out small changes in value when logging data. Before being logged to a database, a value must change by at least the deviation amount of the last logged value.
dialing prefix	Part of the Hayes AT command set for use with modems. See the Dial-Up Modem Settings section of Chapter 3, *Serial Port Communications*, in the *Lookout Developer's Manual* for detailed information.
displayable objects	A Lookout object class that has a displayable component, such as a Pot, a Switch, or a Pushbutton.
DLL	A Dynamic Link Library, which is a collection of small, special purpose programs which can be called by a larger program running on the computer. Sometimes called Dynamically Linked Library.
driver objects	Lookout objects used to communicate with PLCs, RTUs, and other I/O devices.

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. Thermocouple data, for instance, arrives in volts as the raw unit, and must be converted to degrees, an engineering unit.
environment services	Tasks Lookout performs as a part of making your SCADA/HMI work easier. Lookout environment services include serial communications, database and logging, security, networking, alarming, and so on.
Ethernet	A widely used, standardized local area networking technology, specified in the IEEE 802.3 standard.

event	Anything that happens can be an event. In Lookout, events include such things as adjusting a control value, entering or exiting 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 are often paths to a data member. They can also function like variables that, using a spreadsheet cell type formula, become capable of performing flexible, real-time math operations, condition testing, and other complex operations functions. See Chapter 1, *Expressions*, in the *Lookout Developer's Manual* for more information on expressions.

F

failover	A failover is the takeover of a process by a standby computer when the primary computer fails for any reason.
FieldBus	An all digital communication network used to connect process instrumentation and control systems.
FieldPoint	A National Instruments hardware product line for industrial automation, control, monitoring, and reporting.
frame	Sequence of bytes sent from a computer to a device or vice versa. The syntax of the frame depends on the protocol being used. A read frame contains enough information to specify a set of variables whose values the device should return. A write frame specifies a variable in the device and a new value to write into that variable. Some protocols support the writing of multiple variables in a single frame. A response frame is returned from the device to the computer, indicating whether the frame just sent to it was received successfully. If the frame just received was a read frame, the response frame contains a set of requested values.
functionality	The way an object works, operates, or performs a task. Functionality is a general concept that applies in the same way to all objects in a given object class. Parameters define the specific functionality of an individual object.
functions	*See* expression functions.

G

gray proximity | A term used in Lookout color animation. This sets what percentage of gray will be replaced by a given color as conditions change in a monitored value or set of values.

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.

HOA | Hand-Off-Auto control, used to set whether a value must be changed manually, is completely turned off, or functions automatically. You can use a Pot object and a complex expression to create this sort of control in Lookout, or you can use a RadioButton object, depending on the particular requirements of the task you need to accomplish.

I

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 | A Lookout data member containing the fundamental data for certain object classes. When you make a connection to an (implicit) data member, you only use the name of the object, not the name of the object followed by the data member name.

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.
`.lkp` files	File extension for Lookout process files in versions of Lookout earlier than Lookout 4.
`.lks` files	File extension for a Lookout source file, which Lookout compiles to make a Lookout process file that Lookout can run. This is the file you should make sure you keep backed up in case you need to recreate a corrupted process file, or in case some future version of Lookout cannot run a process file compiled in an earlier version of Lookout.
logging	The process of storing data in a computer database file. See Chapter 7, *Logging Data and Events*, in the *Lookout Developer's Manual* for more information on logging data in Lookout.
logical data member	A Lookout data member of the logical data type.
`.lst` files	Extension for the Lookout state file in versions of Lookout earlier than Lookout 4.

M

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

native data member	Data members built into a Lookout object class, as opposed to data members you create by using aliases.
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 of the numeric data type.

O

object	A specific instance created from an object class.
object classes	Software modules you use to create individual objects to perform tasks in Lookout.
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	Input to an object, similar to a writable data member, whose value is specified in the object parameter list in a Lookout source (`.lks`) file. Typically, parameter values are set in the object **Object»Create** or **Object»Modify** dialog box.
ping	A small 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.
pixel	Picture Element, the smallest bit of a picture. Has one color or shade of grey. The number of pixels per inch determine the resolution of an image.
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.
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 popup 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.

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, an 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. See the *Setting Receive Gap* section of Chapter 3, *Serial Communications*, in the *Lookout Developer's Manual* for more information about the receive gap.

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.
remote	In the context of Lookout, remote is a position source location for a control. See the *Remote Position Source Connections* section of Chapter 4, *Using Lookout*, for detailed information on the Lookout remote position source.
resolution	The smallest signal increment that can be detected by a measurement system. Also, the number of pixels per inch on a computer monitor screen or dots per inch in printer output.
RTS	Request to Send, part of a handshaking protocol for certain devices that connect the serial port of a computer. See the *RTS/CTS Handshaking Settings* section of Chapter 3, *Serial Communications*, in the *Lookout Developer's Manual* for detailed information.
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

SCXI	Signal Conditioning eXtensions for Instrumentation, a National Instruments product line for conditioning low-level signals.
security accounts	Also called user and group accounts, Lookout uses security accounts to define what users or group of users have different operation privileges in Lookout. See Chapter 6, *Security*, in the *Lookout Developer's Manual* for detailed information on Lookout security.
server	A process that provides data (services) to client processes. In Lookout, server processes are intended to be run on one computer only, with 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 in the **System Options** dialog box that Lookout will 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.
system objects	Lookout objects used to control other objects or process and analyze data.

T

TCP 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.

W

.wav files	File extension given to sound files. You can play a .wav file in Lookout to add sounds or speech to alarms or events.
Windows metafile	A standard graphics file type for use in the Microsoft Windows operating environment. If you use a metafile graphic in Lookout, you can enlarge or reduce it on the screen, use them as masks without specifying transparent pixels, and use the Lookout Animator to animate the colors of the graphic.

`.wmf` file   File extension given to Windows Metafile graphic files.

X

`.xls` file   File extension given to Microsoft Excel files.

Index