10

Redundancy

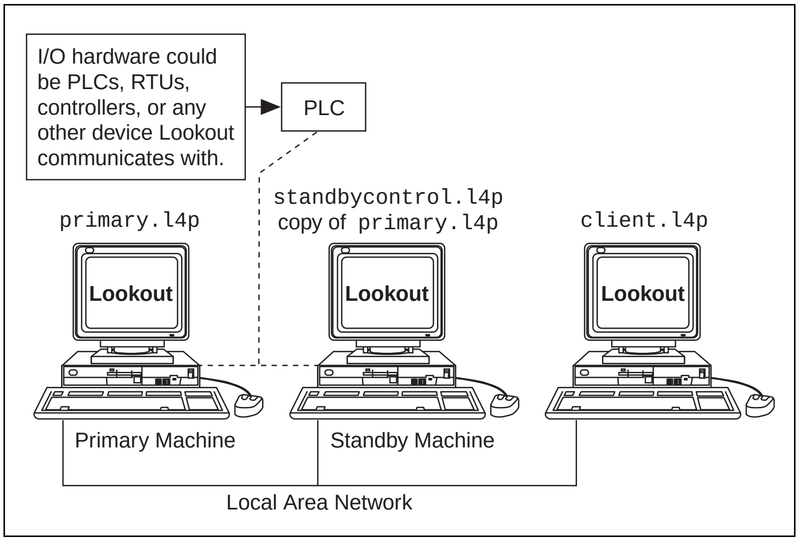
With Lookout*Direct*, you can configure two process control computers for redundancy, providing automatic transfer of control if the primary computer should fail. Redundancy only applies to systems that implement networking of multiple LookoutDirect nodes. It does not apply to stand-alone systems. You should understand process files (`.L4P`) and state files (`.L4T`) before you attempt to use this chapter.

Note In versions of Lookout*Direct* earlier than Lookout*Direct* 4, redundancy was handled through NetDDE. National Instruments recommends you use Lookout*Direct*'s TCP/IP-based networking when possible, but you can still use your old NetDDE-based processes with the newer version of redundancy. However, you must equip your backup computer with a monitor and load process like the one described in the *Setting up the Standby Process and Computer* section of this chapter. You must also modify the process properties of both primary and standby processes to handle state file and data logging paths appropriately, as described in the *Implementing Redundancy* section.

No other alterations should be necessary to upgrade your redundancy processes, though you should test the new arrangement thoroughly. The need remains for following NetDDE naming conventions as well as the other instructions given in Appendix A, *Networking With DDE*. If your client process uses NetDDE in shifting access from one computer to another, it will continue to work in the current version of Lookout*Direct*.

You set up computer redundancy using the Lookout*Direct* Monitor, Loader, and Symbolic Link objects.

Standby refers to a network configuration in which one computer is designated the *primary* computer and another is designated the *standby* computer. The primary computer normally monitors and controls the process while the standby computer monitors the primary computer. If the standby computer quits receiving information from the primary computer, the standby computer automatically takes over and assumes the role of the primary computer. The following illustration depicts a typical Lookout*Direct* network structure with the standby option configured.



Standby Basics

There are three basic principles assumed in the illustration shown above. First, both the primary *and* standby computers must have direct access to your field I/O. If the primary computer fails for any reason, you cannot rely on it anywhere in your backup strategy. Your standby computer must have the same direct access to all your I/O, because it will assume the responsibility of the primary computer. There are several ways to accomplish this, depending on your hardware, network, and communications topology.

The second principle is that you can designate any node on the network as the standby computer. It does not need to be the closest physical computer on the network, though this is typically the case due to the restraints imposed by the first principle. During normal operations (that is, while the primary computer is functioning properly), the standby computer is just another computer on the network.

However, the standby computer should also have a copy of the primary computer process file in its Lookout*Direct* directory, `PRIMARY.L4P` in this example. As soon as the standby computer fails to receive a signal from the primary computer, it automatically loads its local copy of the `PRIMARY.L4P` file. From that point on, the standby node assumes the role of the primary computer, and all other computers running LookoutDirect processes automatically recognize this fact. At the same time, client processes accessing a server running on the primary computer switch from accessing the primary to accessing the standby.

When the primary computer comes back up on the network, the standby computer and any client processes affected return to their previous operation.

The third standby principle takes no additional configuration on your part, but is important to understand for peace of mind. A stand-alone Lookout*Direct* application is responsible for periodically updating its own state file. In a standby system configuration, the primary computer not only updates its own state file, but also the state file of the standby computer. When a failover occurs, the standby computer takes over where the primary computer left off, ensuring bumpless transfer of control (meaning a transfer without discontinuities in control values).

Failover Scenarios

There are four reasons for the standby computer to take control of the process.

- The primary computer is down (that is, it is not turned on, Windows is not running, or an application fault has locked the computer).
- The primary computer is running, but it is not running Lookout*Direct*.
- The primary computer is running Lookout*Direct*, but it is not running the correct process file.
- The primary computer is running the correct process file within Lookout*Direct*, but the two computers cannot communicate over the network.

The purpose of the standby feature is to provide automatic failover to a second computer, restoring your Lookout*Direct* network to its original setup (that is, with the primary computer running its `PRIMARY.L4P` file and the standby computer running its monitor process).

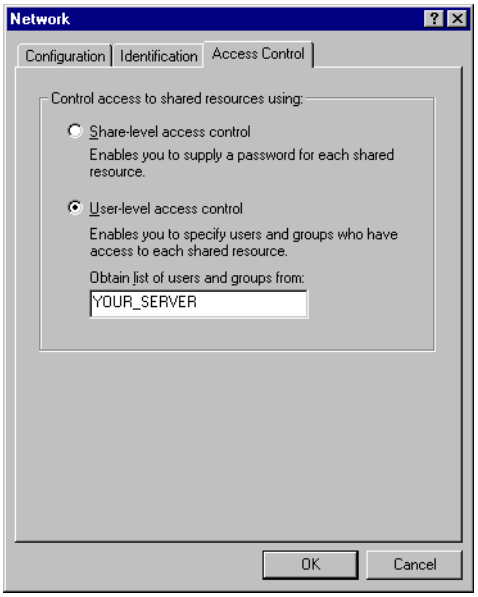
Configuring Standby

For Lookout*Direct* to provide standby operation, all the computers involved must be properly set up on your network. If your computers passed the ping test described in the *Testing TCP/IP* section of Chapter 1, *Installing LookoutDirect*, in the *Getting Started with LookoutDirect* manual, you should be able to implement redundancy with no difficulties. If you cannot ping each computer necessary for your backup system to work, consult your network administrator.

Enabling File Sharing in Windows 98/95 and Windows NT

One part of proper redundancy is making sure that Lookout*Direct* has the most recent state file for a process available when it loads that process. This maximizes your chance of a bumpless transfer of control. For this to occur, your computers must have file sharing permission with each other.

1. Configure the computers you need to share files between using the sharing options in the Network control panel, as shown in the following illustration.



Control access to shared resources using:

○ Share-level access control

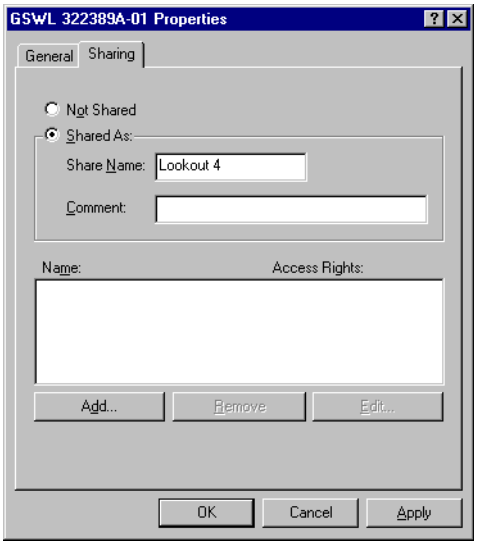
Enables you to supply a password for each shared resource.

● User-level access control

Enables you to specify users and groups who have access to each shared resource.

Obtain list of users and groups from: YOUR_SERVER

2. Select the Lookout*Direct* directory on the standby computer and configure it to be shared, as shown in the following illustration.



○ Not Shared

● Shared As:

Share Name: Lookout 4

Comment:

3. Using the **Add** option, add the authorized computers you want to be able to share to. Make sure full access privileges are granted.

Time Synchronization with Standby Computers

If timing and time-stamped data are crucial to your backup system, consider synchronizing both your primary and secondary computers to a third computer.

In any case, you should make certain that your primary, backup, and client computers are all appropriately synchronized. See the *Time Synchronization* section of Chapter 4, *Networking*, for detailed information about time synchronization.

Implementing Redundancy

To set up your redundancy system, you must configure your primary and standby processes correctly and incorporate certain Lookout*Direct* objects in primary, standby, and client processes.

Redundancy Overview

A Lookout*Direct* Monitor object watches the process running on the primary computer. When that process disappears from the network, the Monitor signals the other elements of the Lookout*Direct* redundancy system into action. When the primary computer and process reappear on the network, the Monitor object signals the standby process and clients to return to their previous mode of operation.

A Lookout*Direct* Loader object handles the job of loading and unloading the standby process on command from a Lookout*Direct* Monitor object.

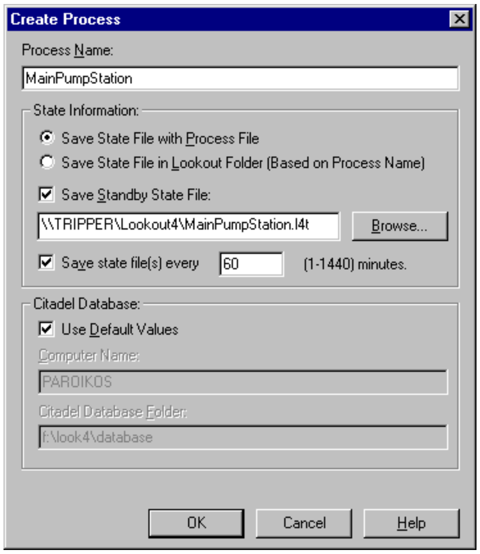
A Lookout*Direct* Symbolic Link redirects a client process from interaction with the primary process to interaction with the secondary process when signaled by a Monitor object. When the Monitor object signals that the primary process is back on line, the client switches back to interacting with the primary process.

For detailed information on the Monitor, Loader, and Symbolic Link objects used in implementing redundancy, see your LookoutDirect online help or the online PDF Lookout*Direct Object Reference Manual.*

Setting up the Primary Process and Computer

If your primary process should go down, you want the transfer to be as bumpless as possible. For this to happen, your standby process needs to open with the most current Lookout*Direct* state file (`.l4t`) available. To do this, configure the **State Information** options of your process.

1. When you create your primary process, you do so using the **Create Process** dialog box. If you are setting up redundancy on an existing process, you can access these same options by right-clicking on the process name in the Lookout*Direct* Object Explorer and selecting **Properties**. In either case, the following dialog box appears.



Create Process

Process Name: MainPumpStation

State Information:

● Save State File with Process File

○ Save State File in Lookout Folder (Based on Process Name)

☑ Save Standby State File: \\TRIPPER\Lookout4\MainPumpStation.l4t

☑ Save state file(s) every 60 (1-1440) minutes.

Citadel Database:

☑ Use Default Values

Computer Name: PAROIKOS

Citadel Database Folder: f:\look4\database

2. Select **Save State File with Process File** to save the state file in the location where the process file was opened from. This is your local state file for your process.

3. Select the **Save Standby State File** checkbox to save one (or more) copies of the state file for your standby process. Enter a complete path, including state file name, to each location to which 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 (|) operator symbol.

   In the preceding illustration, Lookout*Direct* saves the state file to the local directory containing your process file and to the LookoutDirect4 directory of the standby computer, named TRIPPER in this case. Remember that you must make sure your primary computer has write permission to the Lookout*Direct*4 directory of TRIPPER for this backup state file to work.

   When your primary computer goes off line, your process begins to run on your standby computer. The state file the standby computer uses to initialize its version of your process file must be in the directory containing the standby copy of the process file.

   When your process opens, Lookout*Direct* checks the dates of the state files you have selected and uses the most recent file as the state file to initialize values in your process.

4. Set how often Lookout*Direct* saves the state file in the **Save state file(s) every** *[xxxx]* **minutes** field. The Lookout*Direct* default is 60 minutes. Your state file save rate should be the same in your primary and backup processes.

Any time Lookout*Direct* opens this process, it will open with the most recent state file available.

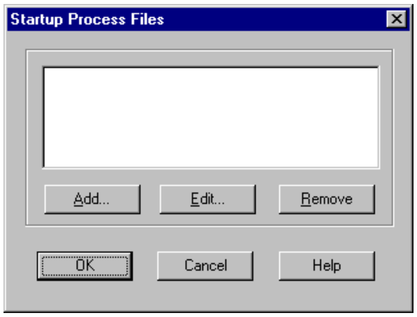
Note The process file you use on your standby computer must be identical to the process file on your primary computer it is backing up except in one respect. The standby version of the process file need only save its state file to the location specified by your primary computer's process file. In this way, when your primary computer comes back online, it can locate the most recent version of the state file to use in initializing.

You might think that you would need to alter your process file before moving a copy of it to your standby computer. The Lookout*Direct* Loader object, however, overrides a number of settings for any process it loads, including the state file information. Therefore, you do not need to alter your process file in any way when you move a copy of it over to your standby computer to be used for redundancy.

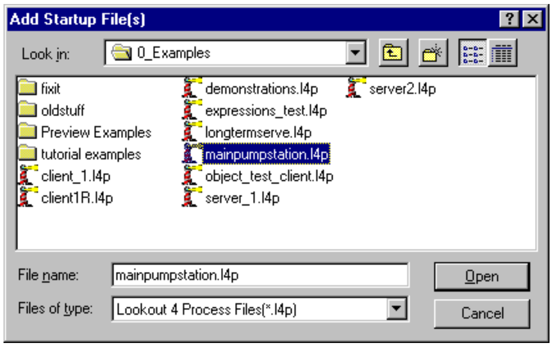
Make sure that when your primary computer comes back on line after being down that Lookout*Direct* loads as soon as the computer boots, and that Lookout*Direct* then loads the proper process.

To make sure your computer launches Lookout*Direct* when it boots, consult your operating system documentation and follow the instructions for your computer. After you have made sure Lookout*Direct* will launch when your computer boots, follow these steps to make sure Lookout*Direct* runs your primary process.

1. Select **Options»Startup**. The following dialog box appears.



2. To add a file to your list of startup processes, click on the **Add** button. A dialog box you use to browse for a file appears. You can autoload as many processes as you want on startup.



Look in: 0_Examples

File name: mainpumpstation.l4p

Files of type: Lookout 4 Process Files(*.l4p)

3. Select the file you want to run when LookoutDirect opens, then click on **Open**.
4. Click on **OK**

Repeat these steps for each process running on your primary computer that you need to run on your standby computer.

You can also reboot a computer, load Lookout*Direct*, bring a state file up to date, and load a process manually if you choose or if it becomes necessary.

Setting up the Standby Process and Computer

Your standby computer must be connected to your Lookout*Direct* network and running a copy of Lookout*Direct* for Lookout*Direct* redundancy to work.

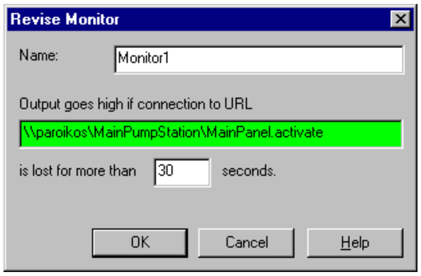
The standby Lookout*Direct* must run a simple process to monitor the process being run on your primary computer. When data ceases to be available from the primary computer, this process loads the standby copy of the primary process. When the primary computer comes back on line and data begins to flow from the primary process again, this same process unloads the standby version of the primary process.

Create this standby control process using the following steps.

1. Create a new process. In this example, it is named `StandbyControl`.

   You do not need to have a front panel for this process, at least in this simple case. You can cancel the **Create Panel** dialog box if you choose.

2. Create a Monitor object. The following dialog box appears.



Revise Monitor

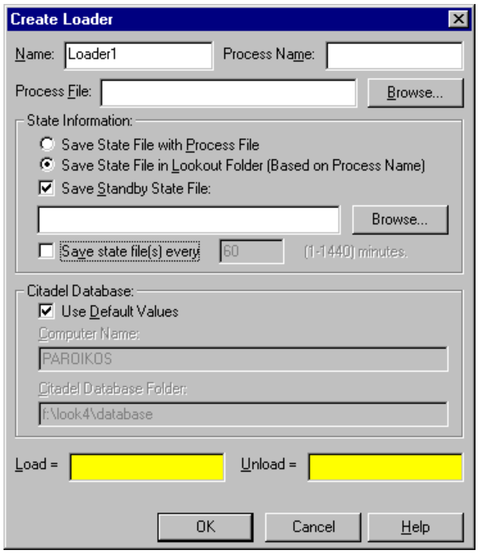
Name: Monitor1

Output goes high if connection to URL

\\paroikos\MainPumpStation\MainPanel.activate

is lost for more than 30 seconds.

3. Enter the URL to some data member in the process you want to back up from your primary computer. The value of this data member does not matter, nor does it matter if that data member is used by any other Lookout*Direct* object. If the Monitor object does not detect the presence of that data member for the amount of time you enter in the **seconds** field, the Monitor will go high (TRUE).
4. Create a Loader object. The following dialog box appears.



Create Loader

Name: Loader1 Process Name:

Process File:

State Information:

○ Save State File with Process File

● Save State File in Lookout Folder (Based on Process Name)

☑ Save Standby State File:

□ Save state file(s) every 60 (1-1440) minutes.

Citadel Database:

☑ Use Default Values

Computer Name: PAROIKOS

Citadel Database Folder: f:\look4\database

Load = Unload =

5. When using the Loader for redundancy, set the **Process Name** the same as the name of the process you are standing by for. The **Process File** should be identical to the process file you intend it to stand in for.
6. The **State Information** settings are important for achieving bumpless transfer back to your primary computer and process when they come back on line. The process file Lookout*Direct* loads using this Loader object will use the state file you configured your primary process to save

on your standby computer. It is important that you update that same state file while the standby process is running, because if it is more recent than the local state file, Lookout*Direct* on the primary computer will load the updated state file from your standby computer.

Because you earlier set the state file to be located in the Lookout*Direct* directory of your standby computer, you should select the **Save State File in Lookout*Direct* Folder** option. If you have reason to save copies of the state file in other locations as well, you can select the **Save Standby State File** option.

Note Settings in the Loader object override any settings you might have made when you created or set the properties for your process.

7. Set how often you want the state file saved in the **minutes** field.
8. Set the location for your Citadel database logging to take place, just as you would when creating a new process.
9. In this example, the Monitor object is named `Monitor1`. Enter `Monitor1=TRUE` in the **Load** field, and `Monitor1=FALSE` in the **Unload** field.
10. Click on **OK**.

When the primary computer goes off line and the Monitor object goes high, the Loader object loads the standby process. When the primary computer comes back on line, the Monitor object reverts to low and Lookout*Direct* unloads your standby process.

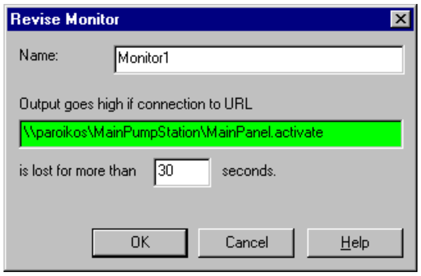
Configuring Clients for Standby Operation

It is most likely that you will use redundancy for server processes on Lookout*Direct*. You may have any number of Lookout*Direct* client processes running on other computers in your LookoutDirect network. These clients also must respond to a change from primary to standby computers.

When you create a Lookout*Direct* client process, you always make connections to the server process and not directly to field hardware, frequently by using the remote position source type of connection.

When you create a Lookout*Direct* client for use with a redundancy-equipped system, you use the Lookout*Direct* Symbolic Link to switch your client process access from one process to another. Instead of connecting your client objects directly to objects in another process, you connect them to the symbolic representation of those objects. The Symbolic Link can shift from primary to standby process when necessary and your client will continue to function, having made its connections through the Symbolic Link.

1. To create this kind of client process, you begin by making a Monitor object, just as you did with the standby control process.



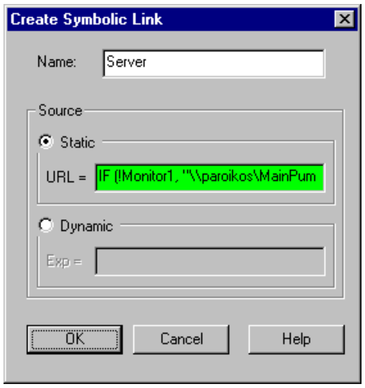
Name: Monitor1

Output goes high if connection to URL \\paroikos\MainPumpStation\MainPanel.activate

is lost for more than 30 seconds.

You should set your Monitor object to the same setting you used in the standby control process on your standby computer.

2. You next create a Symbolic Link. The following dialog box appears.



Name: Server

● Static

URL = IF (!Monitor1, "\\paroikos\MainPum

○ Dynamic

Exp =

3. In this example, the Symbolic Link is named `Server`. The symbolic link `[Server]` refers to which server (primary or standby) is currently active. You use the **Dynamic** field as the source. The following expression is an example of how to set your source.

```
IF(!Monitor1,"\\paroikos\MainPumpStation","\\TRIPPER\MainPumpStation")
```

The `IF` operator lets Lookout*Direct* choose between two processes, depending on the state of the Monitor object. Notice that `Monitor1` (which signifies the `(implicit)` value of the Monitor object) is preceeded by an explanation point. This inverts the value of the Monitor object. In this statement, as long as the Monitor object is *not* `TRUE`, the primary process running on the computer paroikos is available and the objects in your client link to objects in that process through the Symbolic Link.

> If the primary computer process goes off line for some reason, the Monitor object goes `TRUE` and your client process will connect to objects in the standby server process running on TRIPPER.

After you have made this preparation, develop your client process as you ordinarily would, except that you make your connections to the `MainPumpStation` process through your symbolic link instead of with the process itself. Use your Symbolic Link to insert expressions, create remoted controls, and display historical data in HyperTrend.