3

Getting Started with Lookout*Direct*

This chapter contains the following three guided tours. Each tour is designed to provide you with practical hands-on exposure to Lookout*Direct.*

Basic Features Tour—Introduces the basics of the Lookout*Direct* process development cycle. You will use Lookout*Direct's* most common features to create a process that will run locally on your PC without requiring a connection between the PC running Lookout*Direct* and your PLC.

Connected Tour — Provides an immediate demonstration of bi-directional communications between Lookout*Direct* and your PLC using both numeric and logical signals. To complete this tour, the PC on which you are running Lookout*Direct* must be properly connected to a DirectLogic controller. Refer to *Appendix B* for information on establishing a communications link.

Advanced Features Tour—Demonstrates how to create functionality within Lookout*Direct* that you can use to enhance and manage the processes that you create.

To gain full benefit from the tours, we recommend that you complete each tour in the order given. Before starting the tours, ensure that the hardware key supplied with Lookout*Direct* is installed on the parallel port of your PC. Without the hardware key, the functionality of Lookout*Direct* will be limited to one hour of use and ten input/output points. Refer to Chapter 1, *About the Hardware Key,* for additional information on using the hardware key.

After completing the three tours, you may feel you are ready to begin creating your own process. If you would like more in-depth information on Lookout*Direct*, read Chapter 4, *LookoutDirect Basics: Windows, Tools, Files, and Path Names* and Chapter 5, *Using LookoutDirect.*

Basic Features Tour

In this tour, you will create a process that contains some simple functionality associated with a water treatment plant. The completed process file created by this tour can be found in *C:\DirectSoft32\Projects\Examples\LookoutDirect\WaterWorks.l4p.* You may find it helpful to have the example process open while you complete this tour.

Starting a New Process

You will begin the process-development cycle by starting Lookout*Direct,* naming your process, and creating your first control panel.

Note   The first time you start a development version of Lookout*Direct*, the Administrator user account has no password. Any time the Administrator account has no password, Lookout*Direct* opens with Administrator logged in and active, without requiring any log in. This is a convenience for you when creating your Lookout*Direct* processes, but you must be sure to assign a password to the Administrator account before allowing others who should not have Administrator privileges to use your copy of LookoutDirect. Refer to *Creating an Account* later in this section for more information.

1. Click **Start » Programs**. Point to **DirectSOFT32 » LookoutDirect Tools**, and then click **LookoutDirect**.

   The Lookout*Direct* main screen appears. When you start a new process, Lookout*Direct* opens in the edit mode.

2. Select **File » New**.

   The **Create Process** dialog box appears.

3. In the **Process Name** box, type `WaterWorks`. This is the name of the example process.

   Create Process

   Process Name:

   WaterWorks

   State Information:
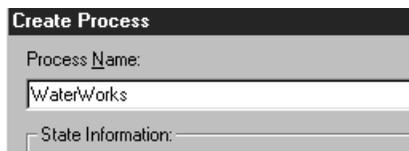
4. Click **OK**.

   The **New Control Panel** dialog appears. Notice that the alarm panel appears at the bottom of the main screen.

5. In the **Name** box, type PNLOverview.

    This is the name of the control panel object. In the **Title** box, type
    Overview. This is the title that will appear at the top of the control panel.
    The control panel you created serves as a pallet on which you will build
    and organize the process.

    New Control Panel

    Name: PNLOverview

    Title: Overview

    Leave the remaining dialog box properties as they are. You will have the
    opportunity to change the way the control looks later in this tour.

Tip   Begin the name of each object that you create with the two or three letter abbreviation
for the type of object. For example, the abbreviation *PNL* is used above to designate the *panel*
object. By consistently using this object naming convention, you will be able to quickly
locate objects in the process directory tree that you will use throughout the three tours.

6. Click **OK**.

    The new *Overview* control panel appears. Maximize the control panel.
    Note that the title of the control panel moves to the window title bar.

7. Select **Alarms » Display Options** and set the window style to *Bottom*
    and the window height to *5* lines.

    This establishes the location and capacity of the alarm window.

Note   If the PC on which you are creating a process does not share the same monitor
resolution as the PC(s) on which the process will be run, set the monitor resolution of the
development PC to match the resolution of the *lowest* resolution monitor on which you
expect the process to be displayed.

Adding an Object

Adding objects is the essence of building a process using Lookout*Direct*. In this part, you will begin by adding a potentiometer object to the new control panel. The potentiometer will be used to control the water level in a simulated tank.

1. Select **Object » Object Explorer**.

    The **Object Explorer** dialog box appears. Notice that the *WaterWorks* process name appears in the tree structure. Drag the Object Explorer to the right side of the control panel to keep it out of the way of the control panel.

Note   Object Explorer provides a single, convenient location from which you can access and modify the object that you add to your Lookout*Direct* processes. Refer to Chapter 4, *Getting Started with LookoutDirect*, *Object Explorer* for information on using Object Explorer.

2. Right-click on **WaterWorks**, and select **New Object**.

    The **Select Object Class** dialog box appears.

3. Double-click on the **Control** folder, and then navigate down the list of objects and highlight the *Pot* object.

    Information on any object that you highlight (before selecting), in this case the *Pot* object, appears on the right side of the dialog box.

Note   Deselecting the **Categorize** check box, located at the top of the *Select Object Class* dialog box, changes the object directory from a folder format to a continuous list format. When displayed in a list format, objects can be found by typing the first letter of the object's name. As you become familiar with the names of an increasing number of objects, you will find it useful to locate them in this manner.

4. Click **OK**.

    The **Select Location for new object** dialog box appears. The dialog box displays the structure of your network in a directory tree format. In this example the name of your computer and the *WaterWorks* process should be visible in the directory tree.
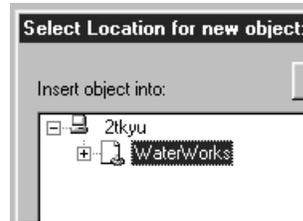
5. Select **WaterWorks**, and then click **OK**.

    The properties dialog box for the new potentiometer object appears.

6. Complete the dialog box properties.

    *   **Tag**: POTSimLevel

        This is the name of the new potentiometer object. Note that we began the name of the new object with the three-letter abbreviation for *Potentiometer*.

    *   **Minimum** and **Maximum**: 0 and 50 respectively.

        For this object, the minimum and maximum values you enter represent the incremental range of the potentiometer.

    *   **Resolution**: .1

7. Leave the remaining properties of the dialog box as they are and click **OK**.

8. Select the new object from the Object Explorer tree, and then drag it onto the panel.

    A dialog box appears that contains the graphic that can be used to represent the new potentiometer object.

9. Select the vertical slider, and then click **OK**.
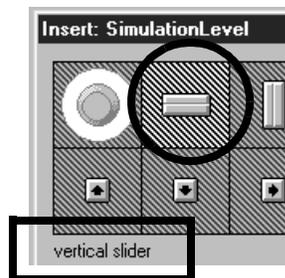
The new potentiometer object appears on the control panel.

Try out the new potentiometer by pressing Ctrl + Space to toggle out of the edit mode, and then click and drag the slider up and down. Return to the edit mode when you are finished.

Creating and Working with Expressions

Because the *POTSimLevel* potentiometer object has an associated numeric range, you can visually represent that range using an expression. For this example, you will create a *graphical* expression in the form of a bar graph and a *numeric* expression in the form of a digital display.

1. Click **Insert » Expression**.

    The **Insert Expression** dialog box appears. The *Insert Expression* dialog box provides a convenient method to locate objects and their data members and *Paste* them into a single completed expression.

2. Expand the network directory tree under the name of your PC until the list of objects under the **WaterWorks** process is revealed, and then select *PotSimLevel.*

Tip   By making the *WaterWorks* process the root of the directory tree, you can avoid having to repeatedly expand the tree each time you want to select an object. To make the *WaterWorks* process the root of the tree, right-click *WaterWorks* and select **Make Root**.

3. Leave the remaining dialog box properties as they are and click **Paste**.

    POTSimLevel appears in the yellow expression field.

4. Click **OK**.

    The **Display numeric signal** dialog box appears. This particular dialog box appears because the *Display type* property, selected by default on the **Insert Expression** dialog box, was set to *numeric*. Lookout*Direct* allows you to select the display properties of a numeric signals.

5. Complete the **Display numeric signal** dialog box.

    *   **Display Style**: Bar (up)
    *   **Bar Color**: deep blue
    *   **Minimum** and **Maximum**: 0 and 50 respectively.

        Note that these values match the range you selected previously for the *POTSimLevel* object.
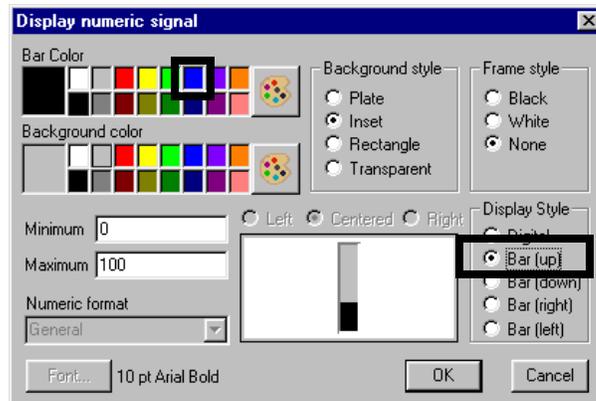
6. Leave the remaining dialog box properties as they are and click **OK**.

    The bar graph appears on the control panel.
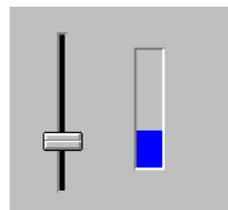
Creating the Digital Display

1. Select the bar graph, hold down the **Shift** key, and then drag the bar graph in any direction and release it.

    A duplicate copy of the bar graph is created. Duplicating objects in this manner is a convenient way to create new objects.

2. Right-click on the duplicate bar graph, and then select **Display Properties**.

3. In the **Display Style** group box, click **Digital**, and then click **OK**.

    The display format of the duplicate expression changes from a bar graph to a digital readout. The readout will initially display as "####". Increase the width of the readout display box, by dragging a handle, to correctly display the numeric value.

4. Move the new digital display to the right of the bar graph.
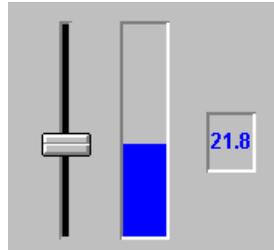
Toggle out of the edit mode, and then move the potentiometer slider up and down to see both graphical expressions function.

Saving Your Work

As with any application, it's a good idea to save your work often.

1. Select **File » Save**.

    The **Select process to save** dialog appear with the *WaterWorks* process—the only one open—highlighted.

2. Click **OK**, and then **Save** to save the process under the name *WaterWorks.l4p*.

Connecting Objects

Earlier in this tour, you created the *POTSimLevel* object and graphically displayed its value at a fixed point in time using a bar graph. In this part of the tour, you will create a hypertrend object that will monitor the state of the *POTSimLevel* object and display, in real-time, its value in the form of a trace line on a simulated chart recorder.

To begin, you will create a hypertrend object, and then assign the signal from the *POTSimLevel* object to the hypertrend. To enable the hypertrend to display the activity of the *POTSimLevel* object, you will activate real-time logging of the *POTSimLevel* signal to a historical database.

Create the Hypertrend Object

1. In the Object Explorer, right-click on **WaterWorks**, and then select **New Object**.

2. Locate and select the **HyperTrend** object from the directory tree, and then click **OK**.

3. Select the **WaterWorks** process and click **OK**.

    The **Create HyperTrend** dialog box appears**.**

4. In the **Name** text box, type `HTTanks`.

5. Click **Add Group**.

    The **Add Group** dialog box appears. The add group function allows you to logically organize and display multiple signals at one time. In this example however, you will only be adding one signal to the group.

6. In the **Name** text box, type `WaterTankTrend`, select deep blue as the **Grid Color**, and then click **OK**.

Add a Signal to the Hypertrend

1. Click **Add Item**.

    The **Add Item** dialog box appears. The properties of this dialog box define the source of the signal you are adding to the *TankLevel* group.

2. Right-click in the green **URL=** field.

    The **URL Editor** dialog box appears. The **URL Editor** dialog box is used to quickly locate the source of a remote signal.

3. Expand the directory tree, select the *POTSimLevel* object, and click **Paste**.

    The URL for the *POTSimLevel* object is pasted into the green URL field.
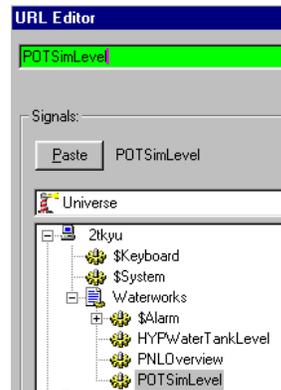
4. Click **OK**.

The source, or URL, of the *POTSimLevel* signal appears in the **URL field** of the **Add Item** dialog box.
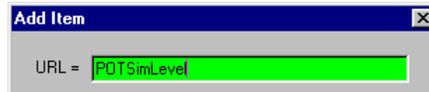
5. Select red as the color for **Line color**, and then click **OK**.

    The new item is added to the *WaterTankTrend* group in the **Create HyperTrend** dialog box.
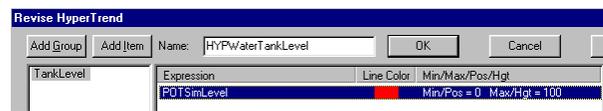
6. Click **OK,** and then select the new object from the Object Explorer tree and drag it onto the panel.

    A dialog box appears that contains the graphics that can be used to represent the new object.

7. Select the desired graphic and click **OK.**

    The new virtual chart recorder appears on the control panel.

8. Drag the new recorder off to the left, and then resize it by dragging one of the corner grips.

    The objects you have added to the control panel so far may seem out of proportion or poorly arranged. You will clean up the control panel later in this tour.

Enable the Historical Database

If you toggle out of the edit mode (Ctrl + Shift) and adjust the level of the *POTSimLevel* object, you will notice that a trace line does not appear on the hypertrend chart. In order for the trace line to appear, you must first activate real-time data logging of the *POTSimLevel* object to a historical database.

1. Toggle back to the edit mode, and then right click on the *PotSimLevel* object in Object Explorer.

    The **Edit Object Database** dialog box appears.

2. Double-click the *POTSimLevel* object.

    The database dialog box appears. The title of the dialog box reflects the name of the object you are editing.

3. In the **Filters** group box, type `.1` for the deviation.
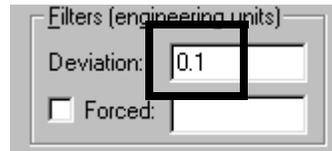
4. Click **Log to historical database**.
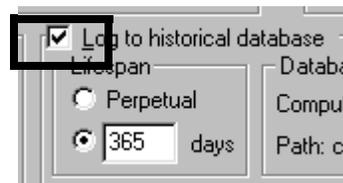
5. Click **Save**, and then **Quit**.

To test the new hypertrend object, toggle out of the edit mode and move the the potentiometer slider up and down. As you move the slider, a red trace line representing the signal from the *POTSimLevel* object appear as a trace line on the recorder.

Creating Useful Interactions Between Objects

To add a degree of automation to the process, you will now create a second potentiometer object, this time in the form of a hand-off-auto (*HOA*) switch; a *neutralzone* object that will monitor a pre-determined deadband of the *POTSimLevel* object; and an logical expression that will take the form of a red or green pump graphic and which will have as its variables, data members from both the *HOA* and *neutralzone* objects. Together, these objects will create functionality that will automatically start the pump (as simulated by our logical expression) only when the *HOA* switch is in the automatic position and the *POTSimLevel* (our simulated water tank level) is between 40 and 46.

Create the Hand-Off-Auto Switch

1. Use Object Explorer to add a new *Pot* object to the *WaterWorks* process.

    After you select the *WaterWorks* process as the location for the new object, the **New Potentiometer** dialog box appears.

2. Complete the **New Potentiometer** dialog box.

    • **Name:** POTHOA

- **Minimum**: 1

  This is the low end of the potentiometer's range.

- **Maximum**: 3

  This is the high end of the potentiometer's range.

- **Resolution**: 1

  We use one as the potentiometer's resolution since we want the potentiometer's range to be segmented into three parts: hand, off, and auto.

- Leave the remaining dialog box properties as they are.
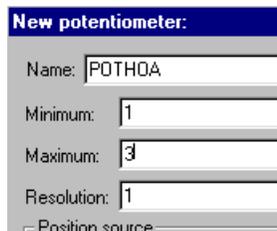
3. Click **OK**.

4. Select the new object from the Object Explorer tree, and then drag it onto the panel.

   The **Insert:** (POTHOA) dialog box appears.

5. Select the *knob* graphic, and then click **OK**.
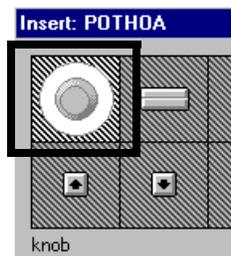
6. Drag the knob to an open area on the control panel.

Toggle out of the edit mode and try out the new three-position switch. For now, we will leave the positions of the switch unmarked. Later in the tour we will add descriptive nameplates to aid in identifying each switch position.

Create the Neutralzone Object

1. Use Object Explorer to add a new *Neutralzone* object to the *WaterWorks* process.

   After you select the *WaterWorks* process as the location for the new object, the **Create Neutralzone** dialog box appears.

2. Complete the **Create Neutralzone** dialog box.

   • **Name**: NZPumpAuto

   This is the name of the new object.

   • **Signal**: POTSimLevel

   This designates that it is the value of the *POTSimLevel* object which is being assigned a neutral zone (deadband).

   Tip   If our example process was complex, and you were unsure of the location or exact name of the *POTSimLevel* object, you could right-click in the yellow expression text box to display the *Expression Editor*. Using the *Expression Editor*, you could locate the *POTSimLevel* object (the signal) and then use the *Paste* function.

   • **High limit**: 46
   • **Low limit**: 40
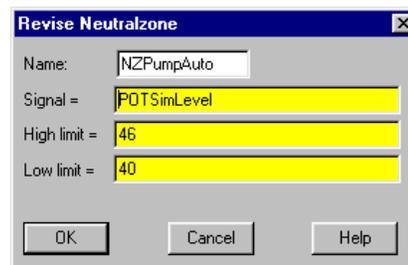
3. Click **OK**.

Create the Logical Expression

1. Select **Insert » Expression**.

   The **Expression Editor** dialog box appears.

2. In the yellow expression text box, type

    `POTHOA = 1 or (POTHOA = 3 and NZPumpAuto.below).`

    This expression sets up the condition where the pump will only run if the HOA switch is in the position 1 or if the *HOA* switch is in position three (Auto) *and* the tank level is within the deadband (the *Neutralzone*).

    Insert expression

    `POTHOA = 1 or (POTHOA = 3 and NZPumpAuto.below)`

    Display Type: logical ▼   Path M
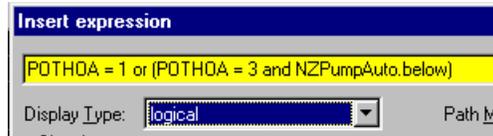
3. Click **OK**.

    The **Display logical signal** dialog appears because the default display type for an expression is a logical signal. From this dialog box you can select how you want the logical outcome of the expression to be displayed.

4. Click **Custom**, and then click the **On** graphic box.

    ◉ Custom

    On   Off
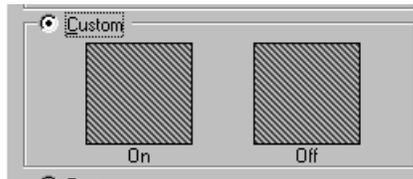
    The **Select Graphic** dialog appears.

5. Navigate to the **Pumps** folder, and then double-click the green **Pmp_3gn.wmf** pump graphic.

    Pumps\Pmp3_gn.wmf

    Transparent pixel
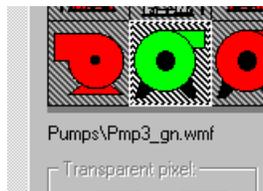
    The pump graphic appears in the **Display logical signal** dialog box as the *On* graphic.

6. Click on the **Off** graphic group box, and then double-click the red **Pmp_3rd.wmf** pump graphic.

Both the **On** and **Off** graphic group boxes should now be selected.
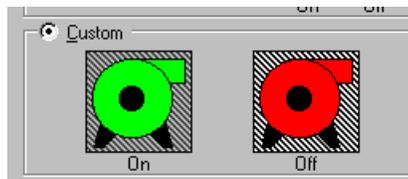
7. Click **OK**.

    The green pump graphic appears on the control panel.

8. Drag the pump to the right side of the control panel.

To test the new functionality, toggle out of the edit mode and drag the indicator mark on the *POTHOA* switch from the left-hand position (position 1 or *Hand)* around to the twelve-o'clock position (position 2 or *Off*) to see the pump graphic change from green to red. Now drag the indicator mark around to the far right-hand position (position 3 or *Auto*), and then slide the *POTSimLevel* potentiometer up and down. The pump graphic should turn green when the *POTSimLevel* is between 40 and 46.

Dressing Up the Control Panel

Now that you have created the functionality for the *WaterWorks* process, your control panel should look similar to the following illustration.
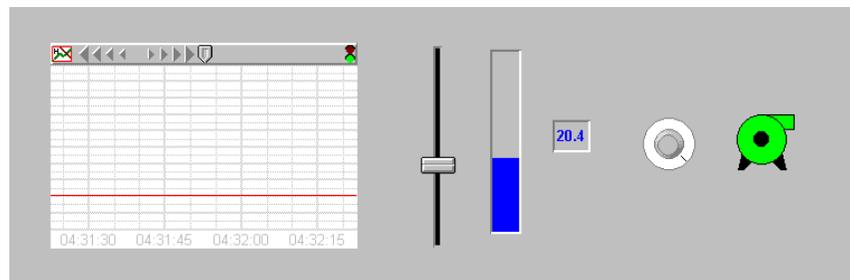
To make the control panel intuitive and easy to use, you will now resize and arrange objects and add various graphics, text name plates, and scales.

Resizing and Arranging Objects on the Control Panel

1. Ensure that you are in the edit mode, and then select the *HyperTrend* chart and resize it so that it takes up about one-half of the control panel.

2. Drag the chart to the left side of the control panel.

3. Select, resize, and arrange each of the remaining objects until your control panel looks similar to the illustration that follows.
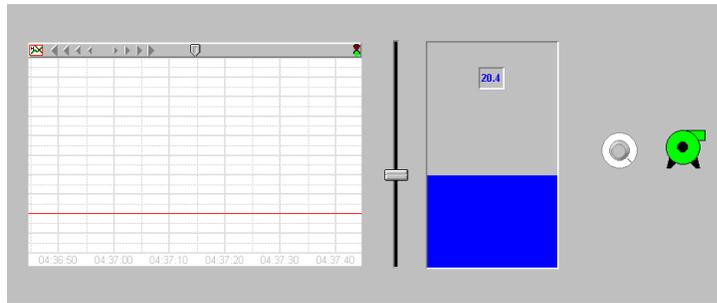
Adding Graphics

1. Select **Insert » Graphic**, open the **Tanks** folder, and double-click on the **Tank1_A.wmf** graphic.

   The tank graphic appears on the control panel. The tank graphic will serve as a stencil over the bar graph giving the appearance of the water level in the tank raising and lowering.

2. Resize both the bar graph and tank until the blue portion of the bar graph fills the tank at its widest point. You may need to toggle out of the edit mode and adjust the *POTSimLevel* potentiometer to check the fit of the tank over the bar graph.

3. Move the digital display of the *POTSimLevel* to the middle of the tank.

When you are finished, your control panel should look similar to the following illustration.
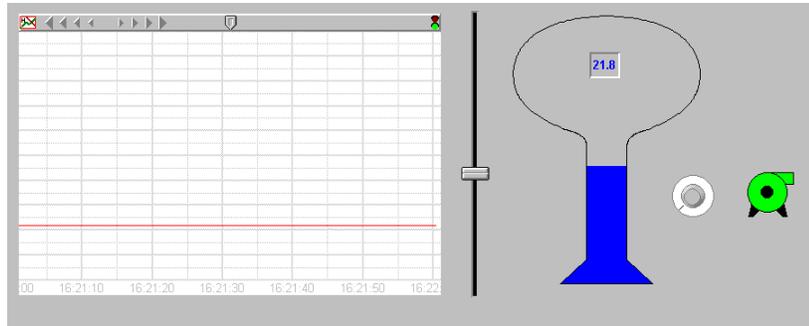
Adding Labels

In this procedure you will create text plates to label the three positions of the HOA switch.

1. Select **Insert » Text/Plate/Inset**.

    The **Insert Text/Plate/Inset** dialog box appears. Use this dialog box to design text labels and add background panels to the control panel. Available design options include the ability to change the style of the background plate; font type, color, and size; and text justification.

2. In the **Text** box, type Hand, and then select black for the text color.

    As you type, the text appears in the preview window in the default font (10 pt. Arial, bold).

    You can experiment with other fonts, background colors, and frame styles or you can accept the default settings.

3. Click **OK**.

4. Drag the new text plate to the left side of the HOA switch.

5. Using the *Shift* + drag method, make two duplicate copies of the *Hand* text plate, and then move one of the text plates to the top of the HOA switch and the other to the right side of the HOA switch.

6. Change the text of the middle text plate to *Off* and the text of the right-hand plate to *Auto*, by right-clicking each text plate, selecting **Display Properties**, and then editing the contents of the **Text** box.
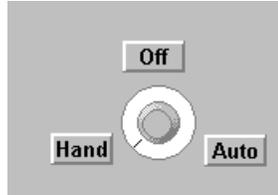
7. To make it easier to reposition the HOA switch and text plates, drag a box around all four objects, and then select **Arrange » Group**.

Adding Scales

Enhance the usability of hypertrend chart and *POTSimLevel* potentiometer object by adding scale bars.

1. Select **Insert » Scale**.

    The **Insert Scale** dialog box appears.

2. Complete the dialog box.

    • **Orientation** group box: *Right side, increasing up* scale style.
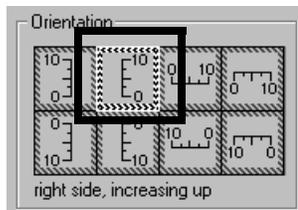
    • **Minimum**: 0
    • **Maximum**: 50
    • **Major Unit**: 10
    • **Minor Unit**: 2.5

3. Leave the other properties as they are and click **OK**.

4. Drag the new scale to the right side of the hypertrend chart and resize it to align with the top and bottom of the chart.

To make more room on the right side of the chart, you will now change the display properties of the *POTSimLevel* potentiometer from a vertical slider to a horizontal slider and add a scale to it.

5. Right-click the *POTSimLevel* potentiometer and select **Display Properties**.

6. Select the *Horizontal slider*, and then click **OK**.

7. Move the horizontal slider to beneath the chart and resize to the width of the chart.

8. Duplicate the scale used on the right side of the chart, change the scale's display properties to a *Bottom, increasing right* scale, move it beneath the horizontal slider, and then resize the scale to fit the slider.
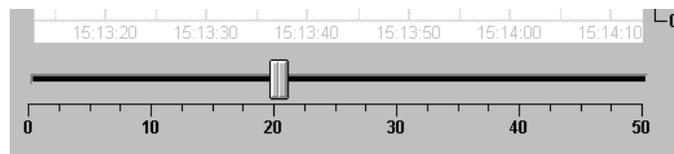
For practice, finish dressing up the control panel by:

- adding text plates, without any text, behind the HOA switch and horizontal slider,

- labeling the chart,

- resizing and arranging the tank above the HOA switch group,

- removing the plates from behind the HOA switch position labels,

- moving the digital display of the *POTSimLevel* inside the top of the tank graphic, and

- creating and placing a large text plate, without text, behind all of the objects on the control panel using the commands provided in the **Arrange** menu.

When finished, your control panel should look similar to the following illustration.
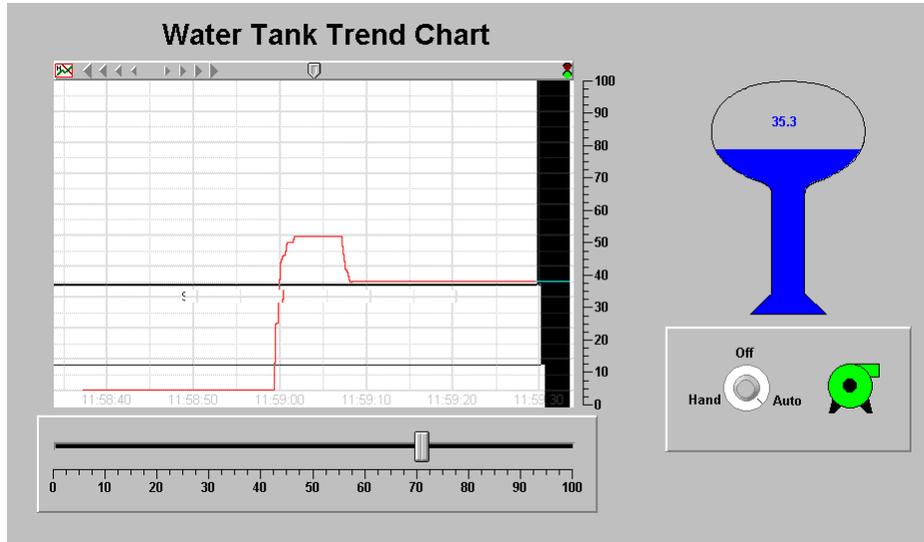
Adding Alarms

To complete the basic features tour, you will now add alarm functionality to the *WaterWorks* process. You will create both a simple alarm and a database generated alarm. A simple alarm is an alarm object that activates only when an alarm-specific parameter is met. A database generated alarm differ from a simple alarm in that it can use the value or condition of one or more data members of an existing object for its activation parameters.

Creating a Simple Alarm

1. Use Object Explorer to add a new *Alarm* object to the *WaterWorks* process.

   The **Create Alarm** dialog box appears.

2. Complete the dialog box.

   - **Name**: `AlarmManMode`

   - **Alarm Area**: `Main`

   - **Message**: `"Pump in Manual Mode"`

     This is the message that will appear in the alarm panel. Note that the message text must be enclosed in quotation marks.

   - **Condition**: `POTHOA=1`

     This simple expression specifies that the alarm should activate when the *POTHOA* switch is in the hand position.

3. Leave the other properties as they are and click **Ok**.

Toggle out of the edit mode and set the *POTHOA* switch to **Hand**. The alarm you just created should activate. To silence the alarm, right-click in the alarm panel and select **Acknowledge All**. You can enter a comment for the alarm or just click **Ok**.

Creating a Database Generated Alarm

1. In Object Explorer, under *WaterWorks* process, right-click the *POTSimLevel* object, and then select **Edit Database**.

    The database dialog for the *POTSimLevel* object appears.

2. Complete the properties of the **Alarm conditions** group box.

    *   **Area**: Main
    *   **HiHi**: 48 , priority 8
    *   **Hi**: 42 , priority 6
    *   **Lo**: 22 , priority 5
    *   **LoLo**: 10 , priority 7
    *   **Deadband**: 3

    These properties specify that an alarm should activate when the numeric value of the *POTSimLevel* object is 48, 42, 22, or 10.
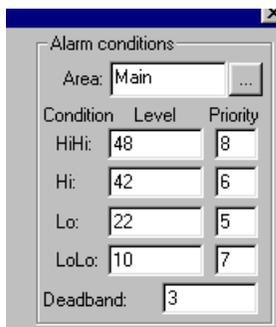
3. Click **Update**, and then **Quit**.

    Toggle out of the edit mode and move the potentiometer up and down. As the numeric value of the *POTSimLevel* object changes, each alarm activates in-turn.

Connected Tour

If you have already set up your PLC and established communications between the PLC and the PC running Lookout*Direct*, this tour will demonstrate Lookout*Directs's* ability to read and write numeric and logical data to and from your PLC. If you have not already setup your PLC or established communications between the PLC and the PC that is running Lookout*Direct*, refer to *Appendix B.*

The examples in this tour demonstrate the following types of connectivity between Lookout*Direct* and your PLC.

- using a virtual switch to toggle a discrete PLC output,
- writing a numeric value to a PLC register, and then reading the value back and displaying it in the form of a virtual trend line,
- reading a discrete signal back from the PLC and use a graphic expression to display its status in the form of a light, and
- Using a digital counter to display the number of scans since the last power-up or program-to-run transition.

The example process file used for this tour can be found in *C:\DirectSoft32\Projects\Examples\LookoutDirect\ConnectedTour.l4p.*

Refer to the *Basic Features Tour* for information on how to create and manage objects using Lookout*Direct*.

Toggling a Discrete PLC Output

In this example, you will create a driver object, and then link the object to a discrete output on your PLC using a new or an existing connection. To test the connection between the driver object and your PLC, you will create a switch object to toggle the discrete output. To indicate the state of the output on the control panel, you will create a logical expression in the form of an indicator light.

Creating the Driver Object

1. Create a new process. Use *ConnectedTour* as the tag name and *PNLConnectedTour* as the title of the control panel.

2. Select **Object » Object Explorer**.

   Move Object Explorer to the right side of the display.

3. Use Object Explorer to add a new *DirectLogic* driver object to the *ConnectedTour* process.

The **Create DirectLogic** dialog appears.

4. Rename the driver object to *DRConnectedTour*, and then click the ellipses button to the right of the **Comm Link** text box.

Tip   When you create your own driver objects, use the *DR* prefix to identify the object as a driver and the name of your PLC's CPU as the tag's suffix.

The **Select Link** dialog box appears.
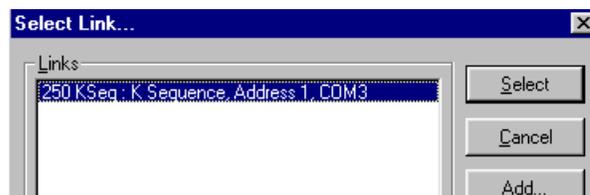
5. Do *one* of the following.
    *   If the **Links** list box contains active communications links found by the links wizard during the installation process, highlight the link you want to use.
    *   If no active link is available, or, if you do not want to use one of the active links, click **Add**, and then follow the directions provided in the **Links Wizard**. Click **Finish** to close the wizard.

6. On the **Select Links** dialog box, click **Select**.

    The name of the new link appears in the **Comm Link** text box.
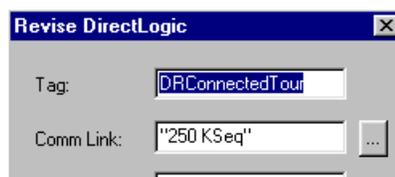
7. Click **OK**, and then go to *Creating the Switch Object*.

Creating the Switch Object

1. Use Object Explorer to add a new *Switch* object to the *ConnectedTour* process.

    The **New Switch** dialog box appears.

2. Rename the switch *SWDiscreteOutput*, and then click **OK**.

    The **Display Switch** dialog box appears.

3. Select one of the standard switch options, and then click **OK**.

    The switch appears on the *ConnectedTour* control panel.

4. Go to *Linking the Switch Object to a PLC Output*

Linking the Switch Object to a PLC Output

1. In Object Explorer, right-click on the *DRConnectedTour* object, and then select **Edit Connection**.

    The connections dialog box for the driver object appears.

2. In the **Writable members** scroll list, double-click the *Y0* member. If the Y0 member is in use, select any other available and writable data member.

3. In the **Signals** group box, navigate through the directory tree to the *ConnectedTour* process, and then double-click *SWDiscreteOutput*.

    Look in the middle of the connections dialog box. Notice that the connection between the driver object and the switch is defined as
    DRConnectedTour.Y0 = SWDiscreteOutput.

4. Click **Accept**, and then **Quit**.

Creating the Indicator Light

1. Select **Insert » Expression**, select *SWDiscreteOutput* from the directory tree, double-click *value* in the **Contents** pane, and then click **OK**.

    The logical signal display dialog box appears.

2. Click **Custom**, and then click in the **On** group box.

3. Open the **Indicators** folder, and then double-click on the green light named *Light0G.wmf*.

4. Click in the **Off** group box, and then double-click on the red light named *Light0R.wmf*.

5. Click **OK**.

    The light representing the logical expression appears.

Toggle out of the edit mode and test the connection between the Lookout*Direct* switch object and the PLC. When the switch is turned on, both the output LED on the PLC and virtual indicator on the control panel should light. If the connection did not work and an alarm error occurred, check the PC-to-PLC connection. If the connection did not work and *no* alarm error occurred, there is a problem with the connection setup. Review this procedure and confirm that the connection was set-up correctly.

Writing and Reading a Numeric Value

In this example, you will use a potentiometer object, in the form of a digital input, to write a numeric value to the PLC. To confirm that data is being written to the PLC, you will create a hypertrend object to read back and display the numeric values sent to the PLC.

Creating the Potentiometer Object

1. Create a new *digital* potentiometer object. Use the following values to complete the **New Potentiometer** dialog box.

   • **Name**: PotNumRWTest

   • **Min**: 0

   • **Max**: 9999

   • **Resolution**: 1

2. Click **OK**.

Connecting the Potentiometer Object to the PLC

1. In Object Explorer, right-click on the *DRConnectedTour* object, and then select **Edit Connection**.

2. In the **Writable Members** scroll list, double-click *V1400-7377*.

   V1400 appears in the selection text box.

3. Replace the *V1400* selection with V2000.
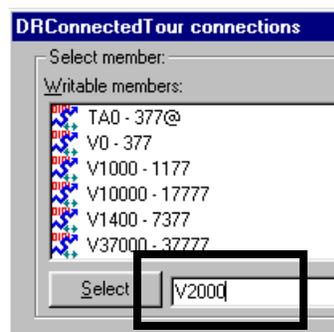
Note   The *V2000* register is used since it is common to all DirectLogic PLCs.

4. In the **Signals** group box, double-click the *PotNumRWTest* object.
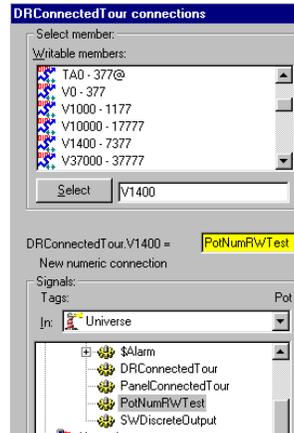
5. Click **Accept**, and then **Quit**.

    The new digital potentiometer appears on the control panel.

Creating the Hypertrend Object

1. Create a new hypertrend object named *HyperRWTest*.

2. Add a group to the hypertrend. Use the following values to complete the **Add Group** dialog box.

    *   **Name**: RWTest
    *   **Major increments**: 10
    *   **Minor increments**: 1

3. Add an item to the *RWTest* group. Use the following values to complete the **Add Item** dialog box.

    *   **Numeric Minimum**: 0
    *   **Numeric Maximum**: 9999

4. Right-click in the **URL=** field.

5. In the **URL Editor** dialog box, highlight *DRConnectedTour*, and then double-click *V1400-7377* in the **Contents** pane.

6. In the **URL** text box, replace *V1400* with *V2000*, and then click **OK**.

7. Click **OK** in both the **Add Item** and **Create HyperTrend** dialog boxes.

    The new hypertrend object appears on the control panel.

8. Enable data logging for the new hypertrend object. Select **Object » Edit Database** and select the *DRConnectedTour* object.

9. In the **Native members** scroll list, double-click *V1400-737*, replace *V1400* with *V2000*, and then update the database dialog box.

    • **Deviation**: `0.1`

    • **Log to historical database**: selected

10. Click **Update**, and then **Quit**.

Test the connection between the digital potentiometer and PLC by toggling out of the edit mode, clicking on the potentiometer object, and entering a new value. The hypertrend object displays the change to the value entered.

Reading a Discrete Signal

In the previous example, you were able to toggle a discrete PLC output using a switch object. In this example, you will read a discrete signal from the PLC and display its status using an indicator object.

1. Select **Insert » Expression**.

2. In the **Insert Expression** dialog box, double-click the *DRConnectedTour* object.

3. Edit the yellow expression field so that the expression reads: *DRConnectedTour.SP4*, and then click **OK**.

Note   The *.SP4* refers to a special relay feature in the DirectLogic controller that provides a one second pulse signal.
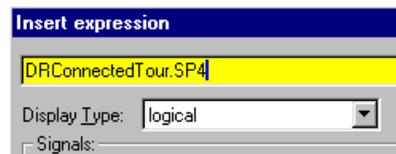
4. In the **Standard** group box, select any indicator, and then click **OK**.

    The logical expression, in the form of the indicator you selected, appears on the control panel and begins to flash at a one second interval.

To confirm that the expression is displaying the signal from the *SP4* relay, toggle out of the edit mode, switch off the PLC, and observe that the indicator stops flashing.

Displaying the Number of Scans

In this connected tour example, you will create a numeric expression that will actively display the instantaneous total for the number of scans performed by your DirectLogic controller.

1. Select **Insert » Expression.**

2. In the **Insert Expression** dialog box, double-click the *DRConnectedTour* object.

3. Edit the yellow expression field so that the expression reads: *DRConnectedTour.V7765:B*, where ":B" is the data modifier that specifies conversion of the data to BCD (16 bit) format. Refer to Appendix B, *Data Modifiers*, for information on using data modifiers.

4. In the **Display Type** drop-down list, select **Numeric**, and then click **OK**.

5. In the **Display numeric signal** dialog box, accept the default settings and click **OK**.

    The numeric readout appears on the control panel and immediately begins displaying the total number of scans.

Advanced Features Tour

Much of Lookout*Direct's* functionality—control panels, alarm functions, security services, etc.—are themselves part of an object class and can, therefore, be customized and to create useful process management functionality. This tour will introduce you to some of Lookout*Direct's* advanced features that you can use to enhance and manage a process.

In this tour you will:

- create buttons that will allow you to quickly navigate between control panels,

- import a custom graphic into a process control panel,

- create a shortcut push-button to acknowledges and silence alarms

- establish security permissions to limit access to all or part of your process, and

- create a data log to record process activities to a spreadsheet file

This tour assumes you are understand the process development concepts introduced in *Basic Features Tour*.

Creating Panel Navigation Buttons

In situations where Lookout*Direct* is displayed on a touch-panel display, where no PC mouse is available, it may be necessary to provide a means of navigating between process control panels. In this example you will create a set of simple push-button that will allow you to navigate between the connected tour control panel (*PNLConnectedTour*) and a new panel you will create in this example. The second portion of this example explains how to create a "Home panel" for your process.

The functionality created in this example can be found in the *ConnectedTour.l4p* process file in the examples folder.

1. Create a new panel object named *PNLNavTest* in the *ConnectedTour* process. Give the panel the title *Navigation Test*.

2. Select the *Connected Tour* control panel.

3. Create a new push-button object named *PBNavTest.* In the **Button Text** box, type `Navigation Test.`

4. In Object Explorer, right-click on the *PNLNavTest* object, and then select **Edit Connection**.

   The edit connection dialog box appears.

5. In the **Writable members** scroll list, double-click *activate*, and then double-click *PBNavTest* in the **Signals** group directory tree.

    The logical connection you just created between the *Navigation Test* panel and the *Navigation Test* push-button appears in the center of the dialog box.

PanelNavTest.activate =   PBNavTest

📝 **Note** Take a moment to review the **Writable member** scroll list to learn about other actions you can generate when editing a connection.

6. Click **Accept**, and then **Quit**.

Test the new inter-control panel push-button by toggling out of the edit mode and clicking the new button.

Continue by creating a second push-button to return to the *Connected Tour* panel.

7. With the *Navigation Test* panel still selected, create a new push-button object named *PBNavTestReturn*. Type Return for the button's text.

8. In Object Explorer, right-click on the *PNLConnectedTour* object, and then select **Edit Connection**

9. In the **Writable members** scroll list, double-click *activate*, and then double-click *PBNavTestReturn* in the **Signals** group directory tree.

    The logical connection you just created between the *Navigation Test* control panel and the new *Return* push-button appears in the center of the dialog box.

PanelConnectedTour.activate =   PBNavTestReturn

Toggle out of the edit mode and test the new *Return* push-button.

Creating a Home Panel

To ensure users do not get lost when switching between panels, you might define one panel as your master control panel, or home panel or computer main menu. You could connect the activate data member of your home panel to $Keyboard.Shift.F1, or perhaps to a pushbutton object. If connected to the

function key, any time the user presses <SHIFT-F1> (no matter what panel he or she is looking at), the home panel is called, returning the operator to a familiar control panel.

You might also want the home panel to maximize upon startup. If you have already created a pushbutton to call the home panel, you can connect it to the maximize data member.

The exclamation point (!) instructs LookoutDirect to use the opposite of the pushbutton value. At startup, the pushbutton is not depressed so its value is FALSE. But because you are using the opposite of the pushbutton value, Panel1.maximize is TRUE at startup. Any time a user depresses CallHomePb after this connection is made, nothing happens until the pushbutton is released—at which time the panel is called.

Importing Custom Graphics

While Lookout*Direct's* graphics library is quite extensive, you may occasionally want to customize a process control panel using your own graphics. You can import graphics saved in either the windows metafile (.wmf) or bitmap (.bmp) file format. To insert your own graphic, first save the graphics in *C:\DirectSOFT32\Bin\Graphics*, and then insert it using the **Insert » Graphic** command.

Creating Alarm Management Buttons

To more efficiently acknowledge process alarms, you can create push-button objects that can acknowledgment or silence an alarm with a single mouse click. Normally, acknowledging and silencing alarms requires a multi-step procedure that must be executed from the **Alarms** menu.

The functionality created in this example can be found in the *ConnectedTour.l4p* process file in the examples folder.

1. Create two new push-button objects, one named *PBAlarmAck* with *Alarm Ack* as its button text and the other named *PBAlarmSilence* with *Silence* as its button text.

2. In Object Explorer, right-click on the *$Alarm* object, and then select **Edit Connection**.

    $Alarm Connections dialog box appears.

3. In the **Writable Members** list box, double-click *Ack*, and then, in the **Signals** directory tree, double-click *PBAlarmAck*.

The logical connection that provides alarm auto-acknowledgement functionality appears in the center of the dialog box.

$Alarm.ack =	PBAlarmAck

4. Click **Accept**.

5. In the **Writable Members** list box. double-click *Silence*, and then, in the **Signals** directory tree, double-click *PBAlarmSilence*.

   The logical connection that provides the alarm silencing functionality appears in the center of the dialog box.

$Alarm.silence =	PBAlarmSilence

6. Click **Quit**.

Establishing Object Security Permissions

To prevent unauthorized or accidental changes to your process, you can restrict access to specific process-level objects using security permissions. Typically, security permissions are assigned to control-type objects such as pots, switches, etc., but they can also be assigned to the process as a whole or to a control panel within the process. When a user, who is logged into the process under a valid account containing their security level, attempts access an object with a security permission, the user's security level is compared to that of the object's, and then access to the object is either granted or denied.

Creating an Account

Complete the following procedure to create accounts for four fictitious users.

1. Select **Options » User Manager**.

   The **User Manager** dialog box appears.

2. Select **User » New User**.

   The **New User** dialog box appears.

3. Create the first of the four users by completing the **New User** dialog box.

   - **Username**: User1

- **Full Name**: UserOne
- **Password**: 1111
- **Confirm Password**: 1111
- **Security Level** = 1

4. Click **OK**.

   User1 appears on the **User Manager** window.

5. Repeat step 2 through 4 to create accounts for users named *User2, User3*, and *User4*. Use 2222 as *User2's* password and assign a security level of 2. Continue this sequence for the other three users.

Setting an Object's Security Level

1. Open the *BasicSecurity.L4p* process file located in the examples folder.

   This process contains a control panel with four simple switch objects.

2. Right-click switch #1 and select **Object Properties**.

3. In the **Control security level** dialog box, type 1.

4. Edit the object properties for each of the remaining three switches, setting their security level to 2, 3, and 4 respectively.

To test the security levels you established for the four switches, you will now log into Lookout*Direct* under each of the user accounts you created.

5. Select **File » Log On**.

   The **Login** dialog appears.

6. Log in as *User1* (password=1111), and then click **OK**.

7. Toggle out of the edit mode and activate switch #1. Now attempt to activate any of the other switches.

    When you point at switch #2, 3, or 4, notice that the cursor changes to a prohibition symbol. The prohibition symbol indicates that the current account does not have permission to use these controls.

8. Log in as *User2* and then *User3* and finally *User4*, each time you log in as a new user, attempt to activate the remaining switches to see which you have access to.

Process Data Logging

An extremely valuable feature of Lookout*Direct* is its ability to log process events to file. Using the *Spreadsheet* logging object you can dynamically save the state or value of any data member to a comma separated value spreadsheet file.

In this example, you will create a logging spreadsheet object to record, at 10 second intervals, the value of two objects that you created during the connected tour. To automate the data logging process, you will create a push-button and an indicator object to start the logging process.

The functionality created in this example can be found in the *WaterWorks.l4p* process file in the examples folder.

Creating the Spreadsheet Object

1. Open the *WaterWorks* process file in the examples folder, and then maximize the *Overview* control panel.

2. Add a new *Spreadsheet* object to the *WaterWorks* process.

    The **Create spreadsheet storage object** dialog appears.

3. Complete the dialog box properties.

    - **Name**: SSDataLogging
    - **File Configuration / File Name**: "ProcessData"

        The complete file name will appears in windows as *ProcessData.csv*.

    - **Interval Logging / Interval**: 00:00:10.00

        This specifies that data will be written to the spreadsheet file every 10 seconds.
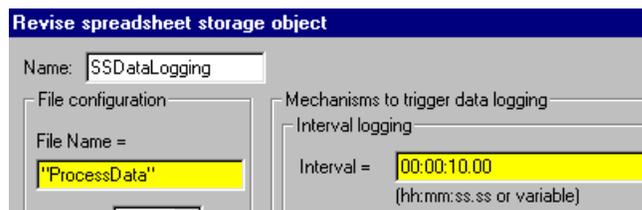
4. In the **Data fields** group, select (**new field**), and then right-click in the yellow expression text box.

    The expression editor appears.

5. In the **Signals** group directory tree, select the *POTSimLevel* object, double-click the **value** data member in the **Contents** list box, and then click **OK**.

6. Click **Save**.

    The value of the simulation level potentiometer is added to the list of data fields that will be included on the *ProcessData* spreadsheet.

7. Repeat step 4 through 6, except this time, add the *value* data member of the *POTHOA* object to the **Data fields** list box.
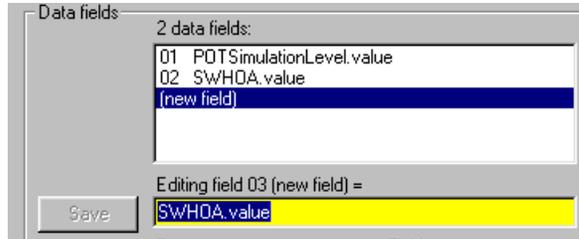
8. Click **OK**.

Creating the Push-button to Start the Data Logging

1. Create a new push-button object named *PBStartLogging* in the
    WaterWorks process. Use *Start Logging* as the button text.

2. Create a new *Flipflop* control object named *FFStartLogging* in the
    WaterWorks process. When the **Create Flipflop** dialog appears,
    right-click in the **Input=** expression text.

    The expression editor appears.

3. Double-click the *PBStartLogging* object, and then click **OK**.

4. Click **OK** to close the **Create Flipflop** dialog box.

    The expression editor appears again.

5. Click **OK**.

6. Select a green (On) and red (Off) custom indicator light, and then click
    OK.

7. Group the *PBStartLogging* push-button and flipflop indicator light in the
    lower right corner of the control panel.
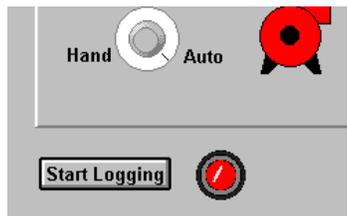
8. Select **Object » Object Explorer**, right-click on the *SSDataLogging* object, and select **Properties**.

    The **Revise spreadsheet storage object** dialog appears**.**

9. Right-click in the **Logging** expression field, double-click *FFStartLogging*, and then click **OK**.

Testing the Data Logging Functionality

1. Toggle out of the edit mode, and then click **Start Logging**.

2. Move the HOA switch to **Auto** (or **Hand** if it's already in **Auto**), and then move the simulation level slider back and forth across its full range of motion.

3. Repeat step 2 approximately every 10 seconds, for one minute.

4. Click **Start Logging** again to stop logging.

5. Use Microsoft *Excel* or another spreadsheet program to open the *ProcessData.csv* file located in: *C:\DirectSOFT32\Projects*[Year]\[Month]. Notice that the file contains a time column plus the columns for the two data members you selected when you created the *SSDataLogging* object.

	A	B	C	
1	Time	POTSimulationLevel.value	SWHOA.value	
2	########	50	1	
3	########	0	3	
4	########	0	3	
5	########	50	1	
6	########	0	3	
7	########	50	1	
8	########	32.8	1	
9				

Note   The Excel spreadsheet file must be closed before logging is started or the data cannot be written to it.