Getting Started 2

This chapter introduces Devguide by giving you a hands-on tour. You see how to start Devguide, how to use Devguide's on-line help, and how to put together a simple user interface using Devguide. When you're finished, you should know your way around Devguide well enough to start building your own simple interfaces or to move on to the following chapters to learn more about Devguide's features.

Installing Devguide

Before you run Devguide, make sure Devguide and a companion program (GXV or GOLIT) are installed on your workstation or server. If they aren't installed, you or your system administrator should follow the instructions in the *Software Developer Kit Installation Guide*. The installation procedure uses the CD that comes with your manual set to install Devguide, the two code generators, and related files.

Changing Environment Variables to Run Devguide

Once you've installed Devguide, you must set the appropriate environment variables—`GUIDEHOME`, `PATH`, and `HELPPATH`—before you can run Devguide and the code generators.

- `GUIDEHOME` – Sets Devguide's home directory
- `PATH` – Tells your shell where to find Devguide, GXV, and GOLIT for execution

- HELPPATH – Tells your window system toolkit where to find Devguide- and application-generated help files

If you use the C shell, you'll add GUIDEHOME and HELPPATH to the file named .login in your home directory, and will most likely find PATH in the .login or .cshrc files. If you use the Korn or Bourne shell, you'll work with GUIDEHOME, PATH and HELPPATH in the .profile file.

C Shell Users

If you're a C shell user, use emacs, vi, or another text editor to add a GUIDEHOME line to the contents of the .login file in your home directory. The line starts
setenv GUIDEHOME and follows with a space and the pathname to Devguide's home directory, which is the destination directory you used during installation. For example, if you used the destination directory /usr/devguide during installation, the line you add will read:

```
setenv GUIDEHOME /usr/devguide
```

Now use your text editor to add a HELPPATH line to the contents of the .login file.The line starts setenv HELPPATH and follows with a space and a series of two directories that are searched in order to find help files:

```
setenv HELPPATH ${GUIDEHOME}/lib/locale:${GUIDEHOME}/lib/help
```

Once Devguide's home directory has been set, use your text editor to look through the contents of .login and .cshrc for a line that starts with set path = (and follows with a collection of pathnames, each separated by a space, and ended with a closing parenthesis. This is a typical set path line:

```
set path = (. ~/bin /usr/ucb /bin /usr/bin /usr/local/bin)
```

Once you find the path line, use your text editor to add $GUIDEHOME/bin to the end of the line, just before the closing parenthesis. Put a space before the pathname to separate it from the other pathnames. Add a backslash if you

need more than one line for your path. For example, if you added the Devguide pathname to the sample `set path` line presented above, the line would now read:

```
set path = (. ~/bin /usr/ucb /bin /usr/bin\
           /usr/local/bin $GUIDEHOME/bin)
```

When you're finished editing and saving these files, you must exit the window environment and log in again to put these environment variables into effect.

Korn and Bourne Shell Users

If you're a Korn or Bourne shell user, use `emacs, vi`, or another text editor to add two `GUIDEHOME` lines to the contents of the `.profile` file in your home directory. The first line starts `GUIDEHOME=` and follows with the pathname to Devguide's home directory, which is the destination directory you used during installation. For example, if you used the destination directory `/usr/devguide`, the lines you add will read:

```
GUIDEHOME=/usr/devguide
export GUIDEHOME
```

Note – If `PATH` or `HELPPATH` is not already in your environment, you must export it, as with `GUIDEHOME` above.

Now use your text editor to add a `HELPPATH` line to the contents of the `.profile` file. The line starts `HELPPATH=` and follows with a series of two directories that are searched in order to find help files:

```
HELPPATH=${GUIDEHOME}/lib/locale:${GUIDEHOME}/lib/help
```

Once Devguide's home directory has been set, look through the rest of the .profile file. You're looking for a line in the file that starts PATH= and follows with a collection of pathnames separated by colons. This is a typical PATH line:

```
PATH=.:$HOME/bin:/usr/ucb:/bin:/usr/bin:/usr/local/bin
```

Once you find the path line, use your text editor to add :$GUIDEHOME/bin to the end of the line. For example, if you added the Devguide pathname to the sample PATH line presented above, the line would now read:

```
PATH=.:$HOME/bin:/usr/ucb:/bin:/usr/bin:\
       /usr/local/bin:$GUIDEHOME/bin
```

When you're finished editing and saving these files, you must exit the window environment and log in again to put these environment variables into effect.

Starting Devguide

Before starting Devguide, you must be running the window system environment. You can use the File Manager to start Devguide by double-clicking on the Devguide glyph, but it is probably easiest to start Devguide from a shell tool or shell script.

Execute the devguide command in a shell tool whenever you want to start Devguide:

```
% devguide &
```

Choosing a Toolkit

After you execute the devguide command, Devguide's base window appears. You should make a toolkit choice now if you plan to generate code after you build a user interface. Do this by choosing Devguide... from Devguide's Properties menu. The Devguide Properties window appears.

Set your toolkit preference using the Toolkit menu. XView is the default choice. Certain properties of Devguide are targeted to the OPEN LOOK toolkit you choose. It is important to choose the right toolkit for your application before you begin work. You need to set a toolkit choice only once. Thereafter, Devguide uses your selection as the default toolkit whenever you start Devguide.

Note – If you don't choose a toolkit, Devguide assumes you want to build an XView application, and sets the environment accordingly. See "Customizing Devguide's Properties" on page 52 for a discussion of the Devguide property window.

Quitting Devguide

To quit Devguide, choose Quit from the Devguide Window menu. (Read more about the Window menu in "Window Menu" on page 30.) If you have unsaved work, Devguide presents a notice, shown in Figure 2-1, telling you that you have unsaved edits. Choose Cancel to continue running Devguide, or choose Discard Changes to quit Devguide, discarding your unsaved work. When you quit, the Devguide window disappears along with any UI elements you have laid out on the workspace using Devguide.

Figure 2-1 Unsaved-Edits Notice

2

Devguide Window

When you start Devguide, the window shown in Figure 2-2 appears on the screen.

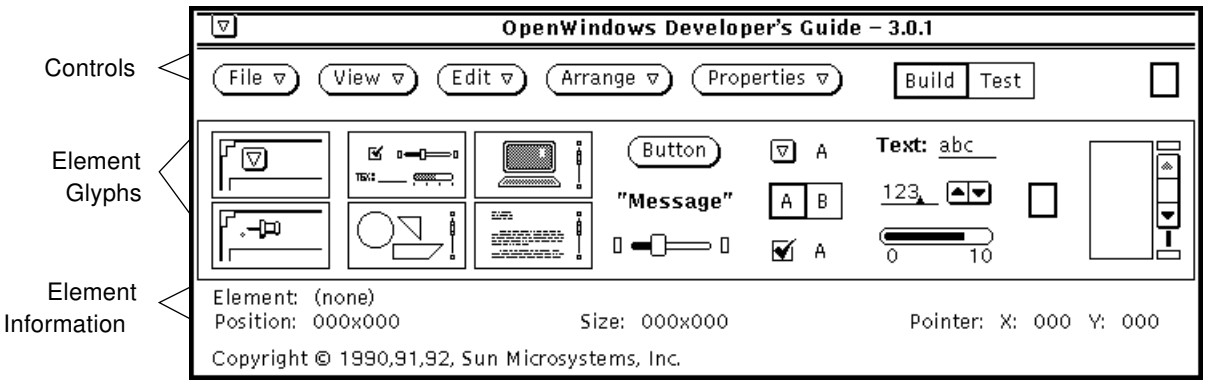


Figure 2-2 Devguide Window

The Devguide window contains the controls and information you need to operate Devguide and the element glyphs you use to add elements to a user interface. Devguide's controls are located in the top of the window, the element glyphs are in the middle of the window, and information about the elements is in the bottom of the window. The middle portion of the window is called Devguide's *palette*. Click SELECT on the abbreviated button in the upper left corner of Devguide's window to iconize the window.

Note – It's important to make a distinction between two types of user interface elements used in Devguide: elements you drag onto the workspace to create an interface, and elements in Devguide's base window (such as menu buttons and menus) that you use to control Devguide. Whenever confusion is possible, this manual uses the term *UI element* (short for User Interface element) for an element you use to build an interface.

Controls

The Devguide controls appear across the top of the Devguide window, as shown in Figure 2-3.

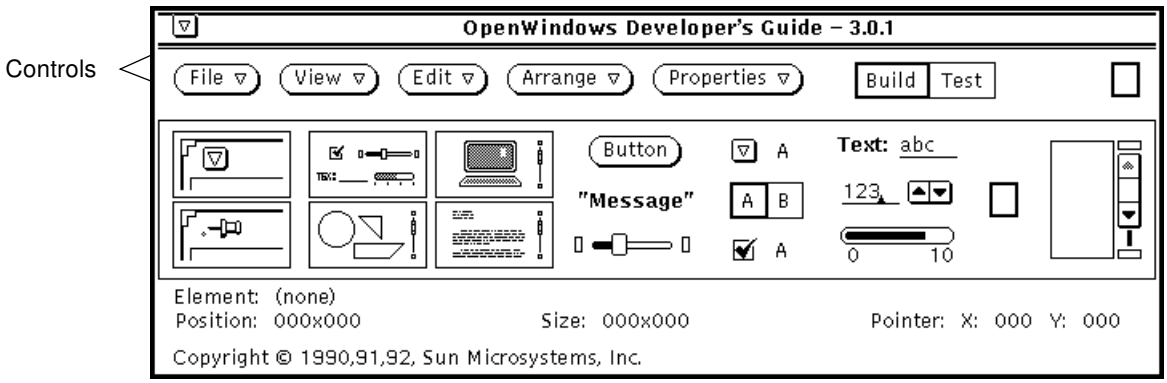


Figure 2-3 Devguide Controls

The Devguide controls include:

- The File menu button, which contains menu items used to load, unload, and save interfaces and projects that you create in Devguide.
- The View menu button, which enables you to scroll through layered panes, open previously dismissed pop-up windows, and bring up the object browser.
- The Edit menu button, which enables you to cut, copy, and paste elements within and between different interfaces.
- The Arrange menu button, which enables you to group, ungroup, and align UI elements, and to use grid alignment options.
- The Properties menu button, which enables you to open windows for editing the properties of individual elements within a user interface.
- The Interface Setting with Build and Test options that set Devguide in build or test mode.

- The Drag and Drop Target, which provides a location for users to drop GIL (Guide Interface Language) files for loading into Devguide. (Do not confuse this with the drag and drop target glyph in Devguide's palette that you can use when you build an application.)

Element Glyphs

The element glyphs appear in Devguide's palette, as shown in Figure 2-4. To create an interface with Devguide, you simply drag element glyphs from Devguide's palette onto the workspace, where each glyph turns into an interface element. You can lay elements down on top of one another or next to one another to assemble the interface you want.

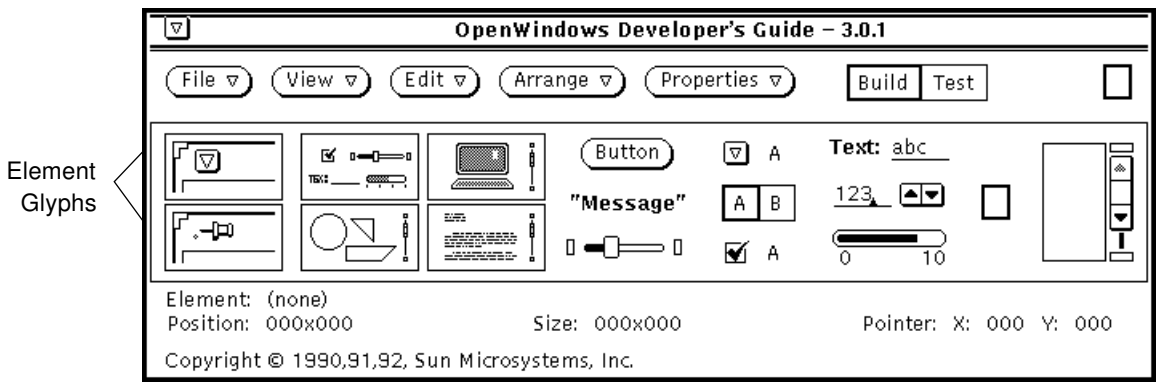
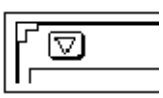


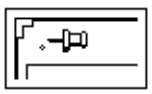
Figure 2-4 Devguide Element Glyphs

These element glyphs include:

Windows

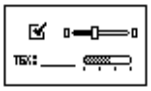


The Base Window glyph places a base window on the workspace when you drag it there.



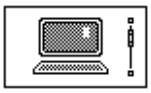
The Pop-up Window glyph places a pop-up window on the workspace when you drag it there.

Panes



The Control Area glyph places a control area in an interface when you drag it onto a base window or pop-up window.

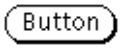
The Canvas Pane glyph places a canvas pane in an interface when you drag it onto a base window or pop-up window.



The Term Pane glyph places a term pane in an interface when you drag it onto a base window or pop-up window.

The Text Pane glyph places a text pane in an interface when you drag it onto a base window or pop-up window.

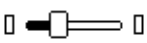
Controls



The Button glyph places a button in an interface when you drag it onto a control area. You can place an *Abbreviated Menu Button* in an interface by first dragging the Button glyph onto the workspace, bringing up the Buttons property window, and altering the button's properties. A button becomes a *menu button* when you attach a menu to it.

"Message"

The Message glyph places a message in an interface when you drag it onto a control area.

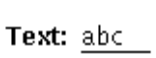


The Slider glyph places a slider in an interface when you drag it onto a control area.

The Stack Setting glyph places a stack setting in an interface when you drag it onto a control area.

The Exclusive Setting glyph places an exclusive setting in an interface when you drag it onto a control area.

The Check Box Setting glyph places a check box setting in an interface when you drag it onto a control area. You can place a *Nonexclusive Setting* in an interface by first dragging any other Setting glyph onto the workspace, bringing up the Settings property window, and altering the setting's properties.

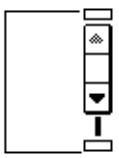


The Alphanumeric Text Field glyph places an alphanumeric text field in an interface when you drag it onto a control area.

The Numeric Text Field glyph places a numeric text field in an interface when you drag it onto a control area. You can place a *Multiline Text Field* in an interface by first dragging any other Text Field glyph onto the workspace, bringing up the Text Fields property window, and altering the text field's properties.

The Gauge glyph places a gauge in an interface when you drag it onto a control area.

The Drop Target glyph places a drag and drop target in an interface when you drag it onto a control area.



The Scrolling List glyph places a scrolling list in an interface when you drag it onto a control area.

Element Information

The Devguide element information appears along the bottom of the Devguide window, as shown in Figure 2-5.

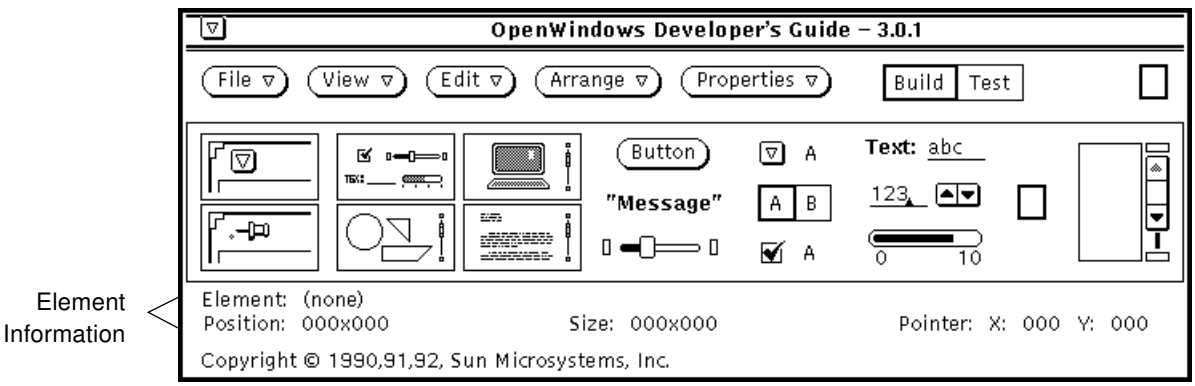


Figure 2-5 Devguide Element Information

The element information area gives you precise information on the position, size, and shape of the current element as well as the precise pointer location.

- The Element field displays the name of the interface element under the pointer.
- The Position field displays the position of the interface element under the pointer. (If the element is in a layer of panes, the layered panes' name and size and relative position of the pane in the layer is also displayed.)
- The Size field displays the size of the interface element under the pointer.
- The Pointer field displays the location of the pointer within an interface window or pane.

On-line Help

If you want information about any of the elements in the Devguide window or in any of the Devguide property windows, you can read the help text about an element simply by moving the pointer over the element in question and pressing the Help key on the keyboard. (The Help key is the F1 key on keyboards that do not have a special Help key.) A Help window like that shown in Figure 2-6 appears.

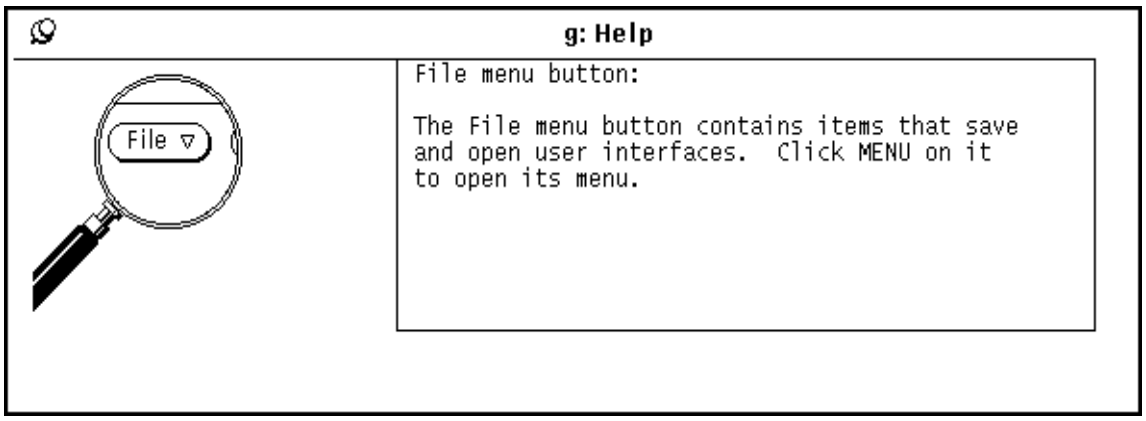


Figure 2-6 Help Window for the File Menu Button

Once you are finished reading the help text about an element in the help window, you can view the help text about another element by moving the pointer over the new element and pressing the Help key once again. The text in the Help window changes to describe the new element. When you are finished reading help text, dismiss the Help window by clicking SELECT on the pushpin in the upper-left corner of the Help window.

Creating Interfaces: A Quick Example

The easiest way to get to know Devguide is to use it to create a very simple user interface. In this quick example, you will:

1. Create an interface by placing a base window in the workspace, adding a control area, and then populating the control area with a button and a setting.
2. Edit the interface by changing the base window's label, changing the button's label, and adding some items to the setting.
3. Test the interface by trying the controls.
4. Save the interface and quit Devguide.

Creating the Interface

First you will place the basic user interface elements in the approriate positions: a base window, a control area, a button, and a setting.

Placing a Base Window

Start your interface by placing a base window on the workspace.

1. **Move the pointer over the Base Window glyph (shown in Figure 2-7) in Devguide's palette.**

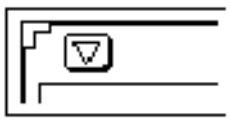


Figure 2-7 Base Window Glyph

2. **Drag the glyph from the palette onto the workspace.**
   To drag, hold down SELECT while you move the pointer, then release SELECT when the pointer is located where you want it. When you drop the glyph, it disappears and a base window appears.

3. **Drag the base window so it appears just to the right of the Devguide window.**
   To drag a window, place the pointer on its header—the strip along the top of the window—then move the pointer while holding down SELECT. Release SELECT when the window is in the location you want.

Adding a Control Area to a Base Window

A base window is the foundation for any user interface. To add controls to a base window, you must first add a control area to the window.

1. **Drag the Control Area glyph from Devguide's palette to the upper left corner of the base window you just created and drop the glyph there.**
   A small control area appears where you dropped the glyph. Because you just placed the control area on the workspace, it is selected and ready for you to resize. Eight small black squares—its resize handles—appear around the control area's borders to show that it's selected. Figure 2-8 shows the control area's resize handles, along with what the control area looks like as it is being stretched to fill the base window. (The oversized control area snaps to fit the window.)

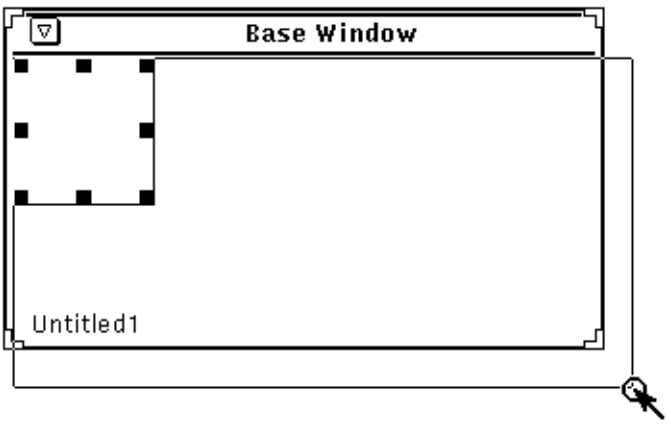


Figure 2-8 Resize Handles; Stretching Out the Control Area

2. **Drag the control area's lower right resize handle down and to the right until you've stretched the control area past the lower right corner of the base window (as shown in Figure 2-8), then release the resize corner.**
   The control area resizes to fill the entire base window with a small strip of space along the bottom of the window—the window's footer.

Adding Controls to the Control Area

Now that you have a base window with a control area, you can add controls to the control area. Add a setting and a button.

1. **Drag the Exclusive Setting glyph from Devguide's palette to the upper half of the control area and drop it there.**
   An exclusive setting appears in the control area.

2. **Drag the Button glyph from Devguide's palette to the bottom half of the control area and drop it there.**
   A button appears in the control area.

Editing the Interface

Once the elements of a user interface are in place, you can edit each individual element to your specifications. To edit an element, you change element attributes in a property window that lists the properties of that element.

To see the property window, you can select the element on the workspace and then use its pop-up menu to open its property window. Or you can use the Properties menu button to open the appropriate property window and then choose the name of the specific element you want to edit. You'll see how to use both methods of editing in the examples that follow. (You can also double-click SELECT on the element to open its property window.)

Labeling the Base Window

The label of the base window you created appears in the window's header; it is labeled "Base Window". To change the label, first open the Base Windows property window and select the base window in the property window's scrolling list.

1. **Move the pointer over the Properties menu button and click MENU.**
   The Properties menu (shown in Figure 2-9) opens.

Figure 2-9 Properties Menu

2. **Click MENU on the Base Windows item in the Windows submenu.** The Base Windows property window (shown in Figure 2-10) appears. The Base Windows scrolling list at the top of the property window lists all the base windows in your interface. It should show only "window1" right now. Click SELECT on the name "untitled1::window1" in the Elements list to show the properties for your base window.

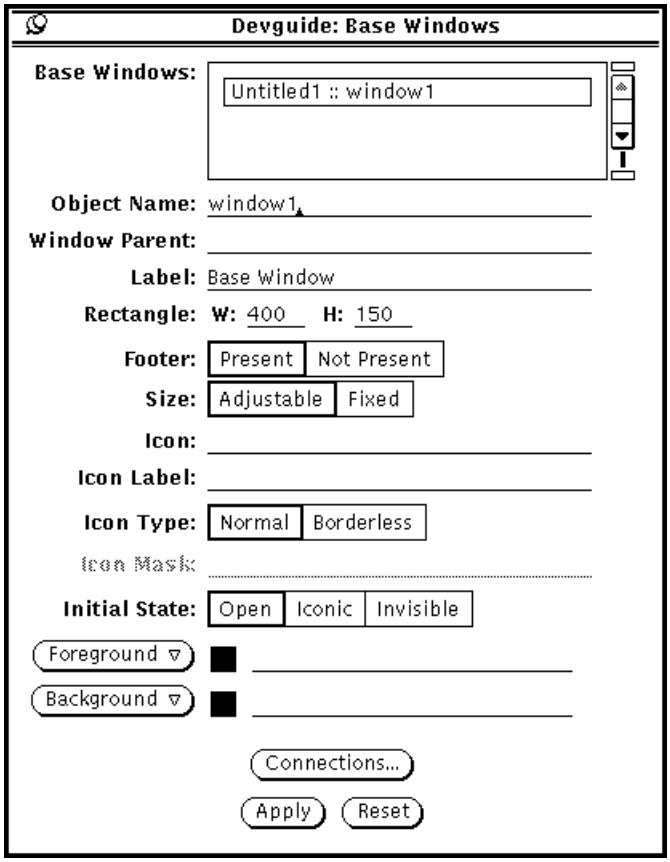


Figure 2-10 Base Windows Property Window

3. **Move the pointer to the end of the characters following the text field Label (the characters should read Base Window) and click SELECT.**
   A small triangular caret appears after the letter "w".

4. **Press the Delete key until all the characters in the text field are deleted.**

5. **Type the label** `Coffee Machine`**.**

6. **Choose the Apply button at the bottom of the property window to apply the name change you just made.**
   The header of the base window now reads `Coffee Machine`.

7. **Click SELECT on the pushpin in the upper left corner of the property window to dismiss the window.**

Adding Items to the Exclusive Setting

Now add some items to the Exclusive setting you placed in your interface. Select the setting and use its property window:

1. **Move the pointer over the setting and click SELECT. A selection box appears around the setting.**
2. **Click MENU to open the setting's pop-up menu.**
3. **Choose Properties from the menu.**
   The Settings property window (shown in Figure 2-11) appears. The setting's name is selected in the Setting scrolling list in the top of the window, and the rest of the window shows the setting's current properties. The Items scrolling list in the bottom half of the window shows the items currently available in the setting. There are two, each named Item. The top one is already selected for editing, as you can see by the selection box around it.

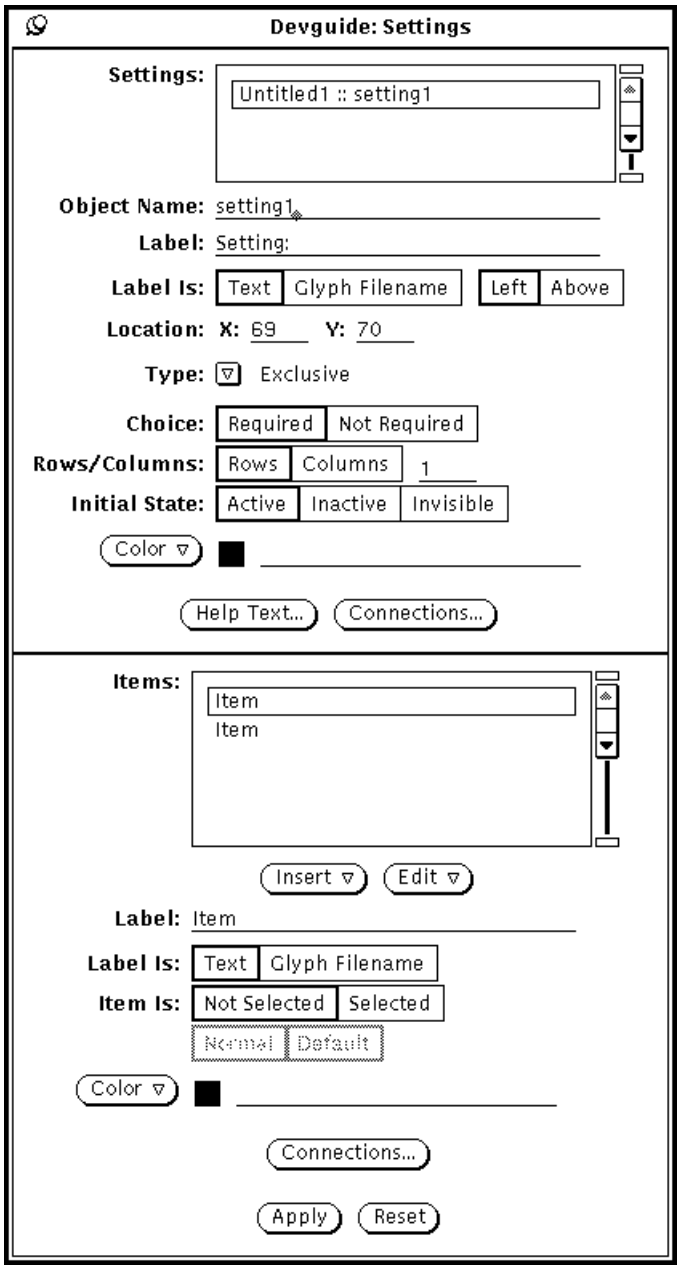


Figure 2-11 Settings Property Window

4. **Click SELECT on the Insert menu button below the Items scrolling list.**
   This chooses the default menu item Bottom in the button menu, which adds a third item—also labeled Item—to the Items scrolling list.

5. **Click SELECT on the top item shown in the Items scrolling list.**
   This selects it for editing, and its label appears in the Label text field below the scrolling list.

6. **Click SELECT just after the "m" in "Item" as it's shown in the Label text field.**
   A caret appears there.

7. **Use the Delete key to delete all the characters in the field.**

8. **Type `W/Cream` in the text field, then click on the second item in the Items scrolling list to select it for editing.**
   The first item's label now reads `W/Cream`. (When you select a new item in the Items scrolling list, this automatically applies edits you just made.)

9. **The second item's label appears in the Label text field; change it to `W/Sugar`, then click SELECT on the third item to select it for editing.**
   The second item's label now reads `W/Sugar`.

10. **The third item's label appears in the Label text field; change it to `Dispense Cup After Coffee`, then click SELECT on any of the other items in the Items scrolling list.**
    The third item's label now reads `Dispense Cup After Coffee`.

11. **Look for the abbreviated menu button in the Settings property window labeled Type; it should be followed by the word Exclusive. Click MENU on the abbreviated menu button (it's labeled with a small, down-pointing wedge).**
    It opens to reveal four different setting types.

12. **Click SELECT on the Nonexclusive item.**
    The menu closes and the Type line now reads Nonexclusive. (This makes the item nonexclusive so the user can choose more than one of the items at a time.)

13. **Click SELECT at the end of the Label text field at the top of the window, then press Delete repeatedly to remove all text from the text field.**
    This removes the setting label so no setting label appears in the interface.

14. **Choose the Apply button at the bottom of the property window to apply the property changes you just made.**
    The setting now offers three different items. If the setting is now off-center in the window, drag it to a new centered location.

15. **Click SELECT on the pushpin in the upper left corner of the property window to dismiss the window.**

Changing the Button Label

Change the label of the button in the bottom half of the window by using its property window.

1. **Move the pointer over the button and double-click SELECT.**
   The button's property window (shown in Figure 2-12) appears, showing the button's current properties.

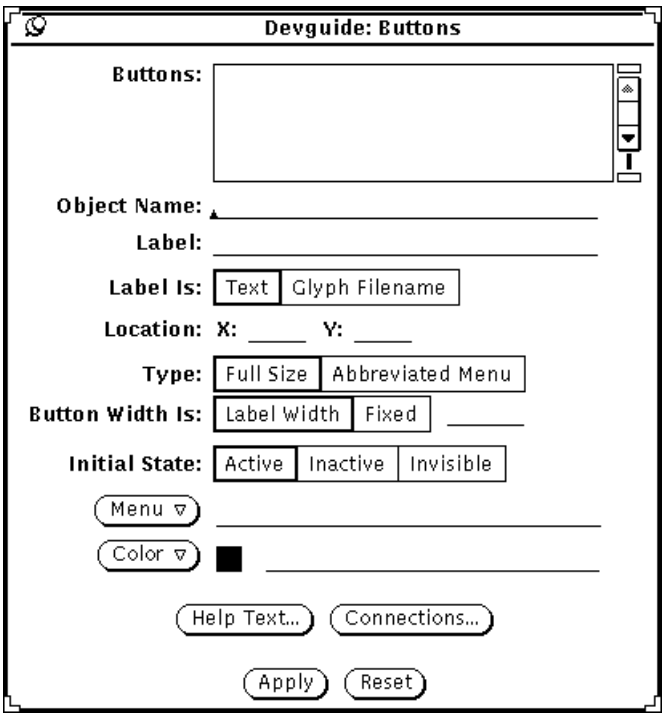


Figure 2-12 Buttons Property Window

2. **Change the contents of the Label text field from Button to `Dispense`. (Change it the same way you changed the base window's label in its property window.)**
3. **Choose the Apply button at the bottom of the property window to apply the name change you just made.**
4. **Click SELECT on the pushpin in the upper left corner of the property window to dismiss the window.**

You've now created the interface shown in Figure 2-13. If your interface looks different, you can resize your base window by dragging its resize corners, and reposition the controls by dragging them to new locations. If you've made a typing error or some other mistake, you can edit properties using the element property windows.

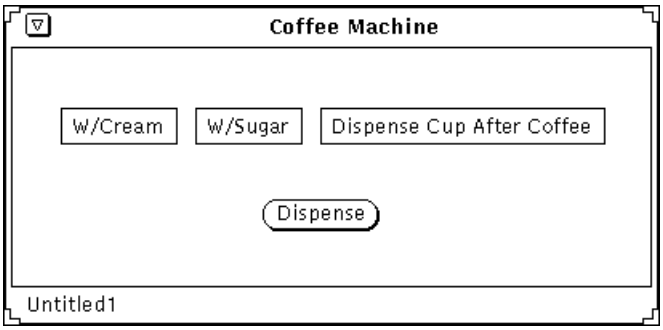


Figure 2-13 Coffee Machine Interface

Testing the Interface

After creating an interface, you may want to see how its elements behave when you use them. One way to do that is to generate the source code for the interface, compile and link the code, and then run the object code file.

A quicker way is to turn on the test mode. In test mode, the elements of your user interface work, for the most part, as they will when a user works with them; that is, the buttons flash, menus appear, and selections set choices. Nothing is connected to the user interface in this example, so the elements don't initiate any actions—coffee won't pour, no cream or sugar is added—but

you can get a feel for the way the interface works in use. Read "Building and Testing" on page 49 and "Handling Events Using Connections" on page 81 to learn about actions that can work in test mode.

1. **Click SELECT on the Test option of the Build/Test setting in the upper right corner of the Devguide window.**
   Devguide enters test mode. Try testing the coffee machine interface now.

2. **Click SELECT on the W/Cream item.**
   If you have a monochrome monitor, a black border appears around the item to show it was chosen. If you have a color monitor, it appears "depressed."

3. **Click SELECT on the Dispense button.**
   It flashes to show it was chosen.

You can't edit your interface in test mode. To edit, you must return to build mode, Devguide's default mode.

4. **Click SELECT on the Build option in the Interface setting. Devguide enters build mode.**

Saving the Interface

When you're finished working on an interface and want to save it for later editing or source code generation, use the Save As... command in the File button menu.

1. **Click MENU on File at the top of Devguide's base window.**
   The File button menu appears.

2. **Choose Save As... from the menu.**
   The menu disappears and the Save As file chooser window (shown in Figure 2-14) appears.

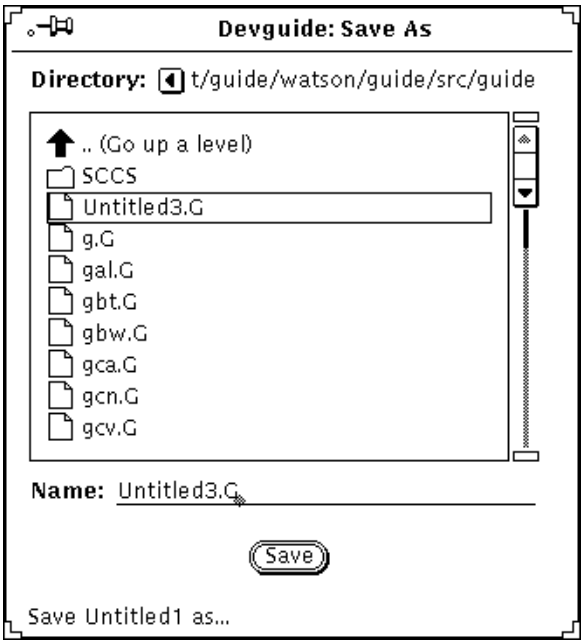


Figure 2-14 Save As File Chooser Window

3. **Move the pointer over the Save As file chooser window, double-click SELECT on the line labeled Name, then type `coffee` to name this interface.**

4. **Choose the Save button to save the interface to Devguide's resident directory using the file name `coffee`.**
   Note that the Directory text field shows the current directory where Devguide will save the interface. The window closes after the save is complete, unless you have pinned it to the workspace.

You have now saved the interface as a GIL file. The interface is stored under the file name `coffee.G`. Devguide appends `.G` to any GIL file it creates, unless you include this extension in the file name. (Devguide appends `.P` to the name of any project file you create, if you have not done so. See "Assembling Interfaces into a Project" on page 87 to learn about creating projects in Devguide.)

Once you're finished working on an interface and have saved it, you can quit Devguide.

Generating Source Code

Now that you've created and saved a user interface, you can use one of Devguide's companion toolkit code generators (GXV, or GOLIT) to generate source code for the user interface. To do this, you can first open a shell tool in the workspace if it's not already open, and then use the appropriate code generator command. As an example, here is how to generate source code for your interface using the `gxv` command:

```
% gxv coffee
```

Alternatively, you can use the Code Generator Tool. See "Using the Code Generator Tool" on page 101 for details.

Compiling Object Code and Running the Results

You can invoke the `make` command yourself, by entering `make` in the shell:

```
% make
```

The executable file for our example is called `coffee`.

Alternatively, you can use the Code Generator Tool to compile and run this application. See "Using the Code Generator Tool" on page 101 for more information.

See the manual in this documentation set for your toolkit code generator for details on generating and compiling source code, and for setting code generator options in the Code Generator Tool.

That's it for this quick and easy introduction to Devguide. Before you leave the scene of your activities, take a few seconds to clean out the files you generated: you should remove all of the `coffee` files, including the generated `Makefile`, from the current directory. If you want to keep these files, create a separate subdirectory in which to store them. If you're ready for more user interface creation, you'll enjoy the next chapter, where you'll learn to use many of Devguide's features.

2