

ICS Tutorials: Designing for Consistency and Flexibility

In this tutorial, you will create a hierarchy of resource styles and apply them to the objects in a simple interface. You will also learn to use constants and identifiers to create consistent and maintainable interfaces.

Before You Begin

- Make a new directory called `tutorial_5` and change directories to `tutorial_5`.
- If you have not started Builder Xcessory yet, enter the following at the prompt in the `tutorial_5` directory:

Start Builder Xcessory % `bx`

Getting Started

Initialize your application as follows:

- Clearing An Interface**
1. Clear the Builder Xcessory display by selecting **New** from the Main Window **File** menu.
 2. Open the UIL file you saved at the end of tutorial four:


```
<tutorial_path>/tutorial_4.uil
```
 3. Select **Save As...** from the Main Window File menu and save your UIL file as `tutorial_5.uil` in the `tutorial_5` directory you just created.

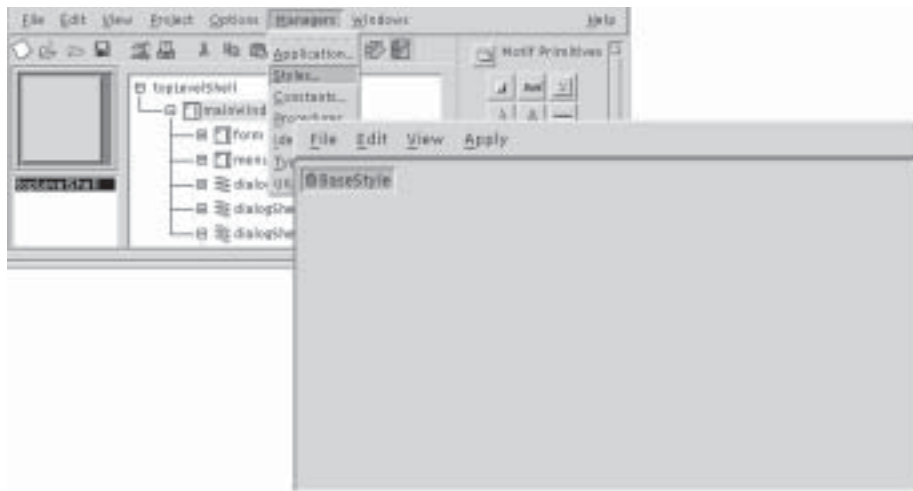
Resource Styles

The Style Manager allows you to create combinations of resources that you can apply to multiple interfaces to maintain a consistent appearance.

Using the Style Manager

Note: This tutorial assumes that you have not already created resource styles. Therefore, `BaseStyle` will be the only style that exists in the Style Manager, and the first style you create will be named `BaseStyle1`.

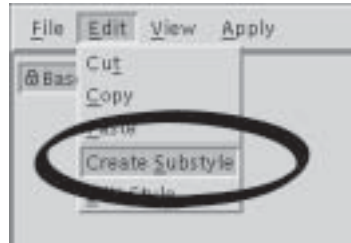
1. Select **Styles** from the Main Window **Managers** menu. The Style Manager appears. Before you create any styles, the style hierarchy is empty except for the `BaseStyle`.



Note: `BaseStyle` is “locked”, that is, although you can bring up the Style Editor for `BaseStyle` in order to examine its contents, you can not add resources to the style.

Creating Substyles

2. Select **Create Substyle** on the Style Manager **Edit** menu. `BaseStyle` remains highlighted while the Style Manager adds `BaseStyle1` as its child. The fact that the parent remains highlighted enables you to create a set of sibling styles without repeatedly reselecting their common parent.



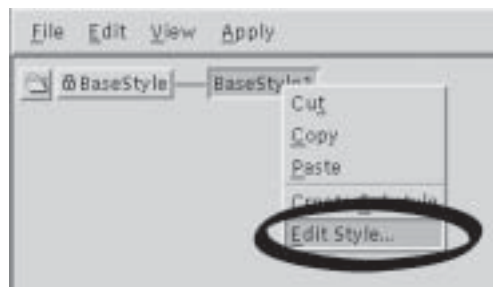
Using the Style Editor

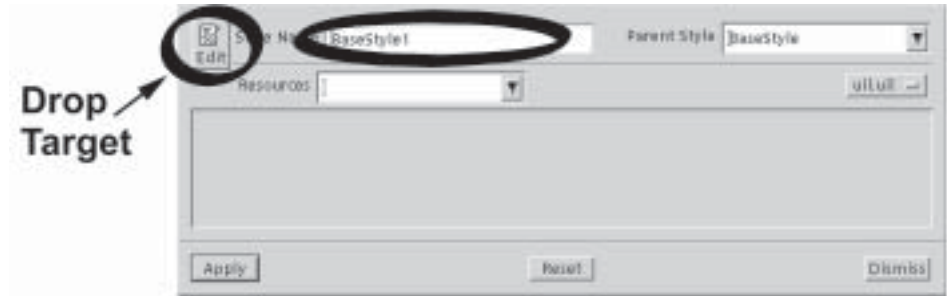
You use the Style Editor to assign lists of resources and their values to a given style.

1. Move the cursor over `BaseStyle1` and press MB3. A pop-up menu identical to the Style Manager **Edit** menu appears.

Note: You can also access the Style Editor by double-clicking on `BaseStyle1`, or by selecting **Edit Style** from the Style Manager **Edit** menu.

2. Select **Edit Style...** from this pop-up menu. The Style Editor is displayed for `BaseStyle1`. You can also load a style into the Style Editor by dropping the style onto the **Edit drop target** on the Style Editor.



**Renaming a Style**

3. Rename BaseStyle1 to DefaultStyle by double-clicking the Style Name field and typing the new name.

Adding Resources to a Style

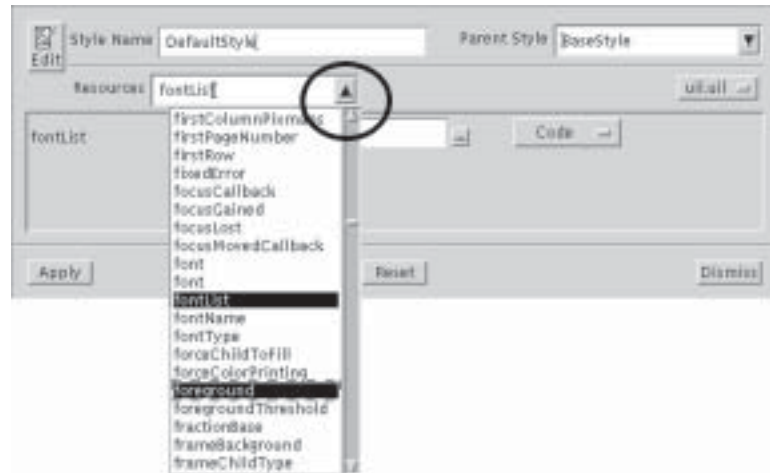
You can add resources to a style in two ways:

- Type the resource names in the Resources text field (as a comma-delimited list).
- Click the arrow button and then select the resource name from the pull-down list. You can select more than one resource from the list by clicking on each resource with MB1. Each selected resource is highlighted in the combination box. You can deselect a resource by clicking on its highlighted name in the combination box.

Defining a Style

To assign resources to DefaultStyle, perform the following steps:

1. Click the arrow button to the right of the Resources label on the Style Editor to display the list of available resources.



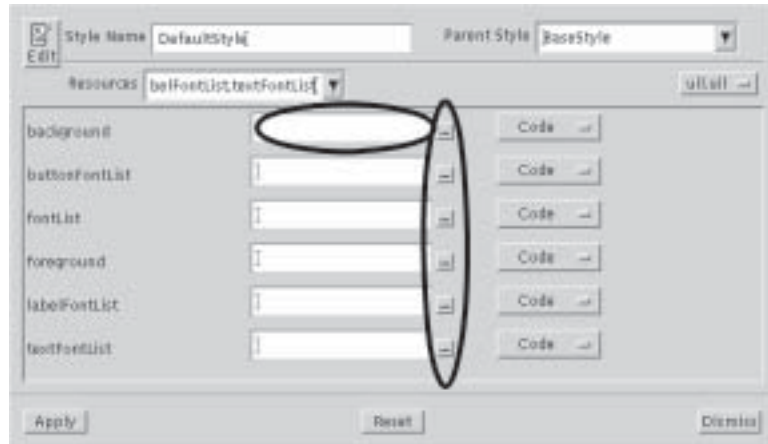
Note: The Resources combination box lists all resources of all objects Builder Xcessory recognizes.

2. Scroll down the resource selection box and select the following resources by clicking on them with MB1: Background, buttonFontList, fontList, foreground, labelFontList, and textFontList. If you accidentally select the wrong resource, unselect it by clicking on it again.
3. Press the arrow button once again to unpost the list.

Editing the Resources Values List

The resources that you have selected for the edited style are displayed in a list in the Style Manager's main display area, directly under the Resources text field. Assign values to these resources either by entering the value directly into the resource text field or by calling up any of the extended editors by clicking the (...) button to the right of the resource text field. You interact with the Style Editor exactly as you would with the Resource Editor.

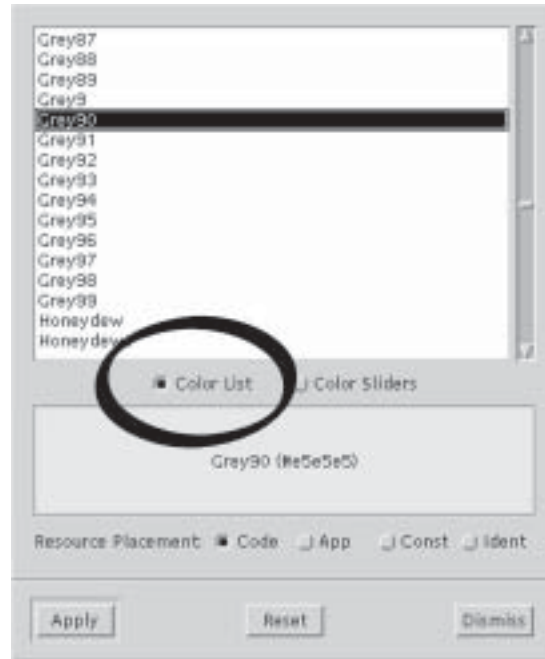
Note: The style is not actually changed until you click the **Apply** button at the bottom of the Style Editor.



Setting Colors

- Set the following color resources either with the appropriate extended editor or by directly entering the value into the resource text field. If you choose to use the extended editor to set these colors, make sure that you select the **Color List** checkbox when the color selector appears. This will change the format of the color selector and allow you to select colors by names (instead of by adjusting the Red-Green-Blue mix).

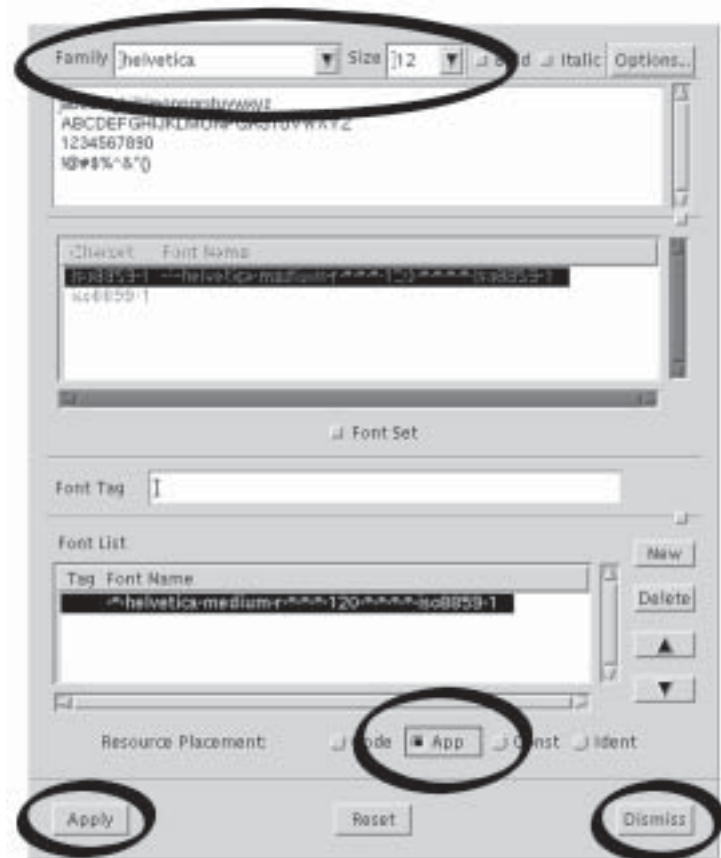
Resource	Value	Location
Background	Gray90	App
Foreground	Black	App



Remember to click the **Apply** button before the **Dismiss** button when done!

Using the FontList Editor

5. Click the (...) besides the ButtonFontList resource. This will bring up the FontList editor.
6. Select the Helvetica Font by clicking on the down arrow to the right of “Family” and clicking on Helvetica. Use the scroll bar to the right of the font names if necessary to find Helvetica.
7. Set the point size to “12” by clicking on the down arrow to the right of “Size”, finding “12” in the list, and clicking on it.
8. Set the Resource Placement to “**App**” by clicking on the appropriate radio button near the bottom of the FontList editor.
9. Click on the **Apply** button to save the changes.
10. Click on the **Dismiss** button to close the FontList editor.



11. Repeat steps 5-10 for the rest of the Font resources in the table below. Remember to click on **Apply** each time to save your changes!

Resource	Family	Point	Loc
FontList	Helvetica	12	/
LabelFontList	Helvetica	12	/
textFontList	Helvetica	12	/

12. **Dismiss** the Style Editor.

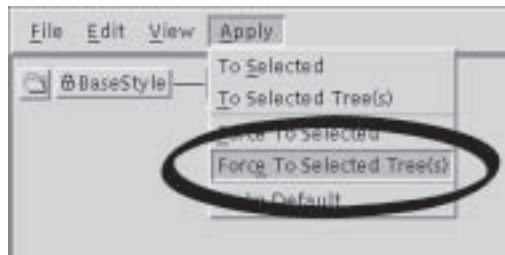
Note: The font resources indicate the font family, point size, and any special attributes. We could have entered the actual X Logical Font Description names in the resource text field. However, many people consider the names too long and difficult to get right. Hence, we recommend that you use the FontList editor. If you are curious, you can examine the actual X Logical Font Description names generated by the FontList editor in the Style Editor after you **Apply** and **Dismiss** the FontList editor.

Applying a Style to Interface

To apply the `DefaultStyle` to your interface, perform the following steps:

1. Select `mainWindow` on the Browser tree.
 2. Select `DefaultStyle` in the Style Manager.
 3. Select **Force To Selected Tree** in the Style Manager **Apply** menu.
- This overrides any previous settings on `mainWindow` and all its children.

Note: If the interface includes dialogs that are not part of the main tree, you must also set the styles for those dialogs separately.



Creating Additional Substyles

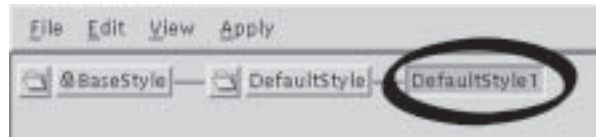
You will now create a substyle of `DefaultStyle`. `InputStyle` is the same as `DefaultStyle` except for the background, and is used to identify fields that accept input.

Creating `InputStyle` To create `InputStyle`, perform the following steps:

1. Select `DefaultStyle`.
2. Select **Create Substyle** on the Style Manager **Edit** menu. This creates a style named `DefaultStyle1`.

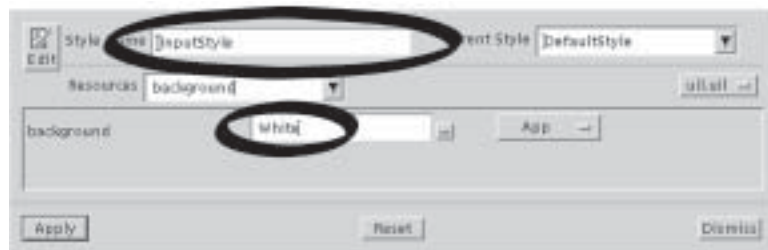
Updating the Style Editor

3. Move the cursor over `DefaultStyle1` and double-click MB1. The Style Editor appears and is now updated for `DefaultStyle1`. `DefaultStyle1` inherits all of the resources specified in its parent styles.



Setting `InputStyle` Resources

4. Change the Style Name to “`InputStyle`”.
5. Add the background resource to the style and set its value to “White” with **App** resource placement.



6. Click on the **Apply** button.

Note: `InputStyle` now contains all of the resource values from `DefaultStyle`, but with a different value for the background resource.

7. Do not dismiss the Style Manager. We will use it in the next section to apply the style we just created to various widgets.

Applying *InputStyle* to Interface Objects

To apply `InputStyle` to input objects, perform the following steps:

1. Select the following text entry widgets in the Browser:
 - `indexList`
 - `entryList`
 - `text`
 - `textField`
 - `textField1`
 - `textField2`

To select multiple objects, hold down `Ctrl` and click `MB1` on the desired objects.

2. In the Style Manager, select `InputStyle`.
3. In the Style Manager **Apply** menu, select **Force To Selected**.

Status Check!

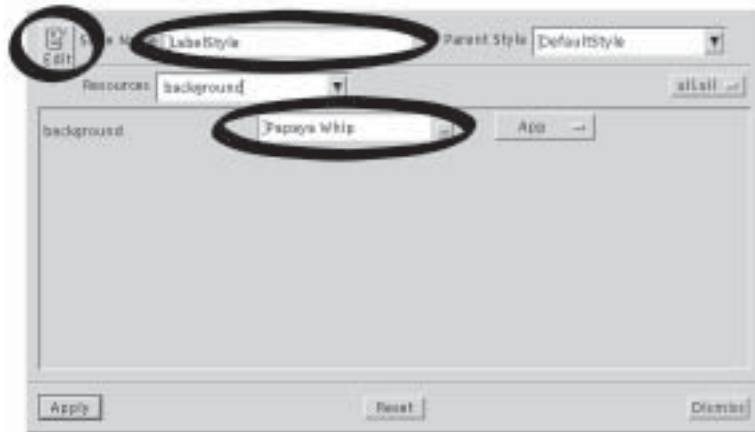
At this point, your user interface should look like the example below. If it does, it would be a good time to save your work before proceeding!



Creating Additional Substyles

Now you will create two additional substyles to add a little variety to your interface. To create a style for labels, perform the following steps:

1. In the Style Manager, select `DefaultStyle`.
2. Select **Create Substyle** under the **Edit** menu. This creates a new style `DefaultStyle1`.
3. Select `DefaultStyle1`.
4. Under the **Edit** menu, select **Edit Style...**
5. Move the cursor over `DefaultStyle1` on the Style Manager and press MB2.
6. Drag the icon to the **Edit drop target** on the Style Editor and release the mouse button. The Style Editor is now updated for `DefaultStyle1`.

**Setting LabelStyle Resources**

7. Change the Style Name to “LabelStyle”.
8. Set the `background` resource to “Papaya Whip” (or your favorite color name!).
9. Set the resource storage to **App**.
10. Click on the **Apply** button.
11. **Dismiss** the Style Editor.

Applying LabelStyle to Interface Objects

To apply LabelStyle to input objects, perform the following steps:

1. Select the following labels in the Browser:
 - indexLabel
 - entryLabel
 - displayLabel
 - label
 - label1
 - label2
 - label3

To select multiple objects, hold down Ctrl and click on the desired objects.

2. In the Style Manager, select LabelStyle.
3. In the Style Manager **Apply** menu, select **Force To Selected**.

Expanding Background Color

Notice that only the area around the label text has changed its background color. (It is a little hard to see this in the diagrams below because of the limitations of gray scale.) To change the background color for the entire label area, perform the following steps:



1. Select the following frames in the Browser:
 - indexFrame
 - entryFrame
 - displayFrame
 - frame
 - frame1
 - frame2
 - frame3
2. In the Style Manager, select LabelStyle.
3. In the Style Manager **Apply** menu, select **Force To Selected**.
4. Notice that the entire frame area has now changed its background color.

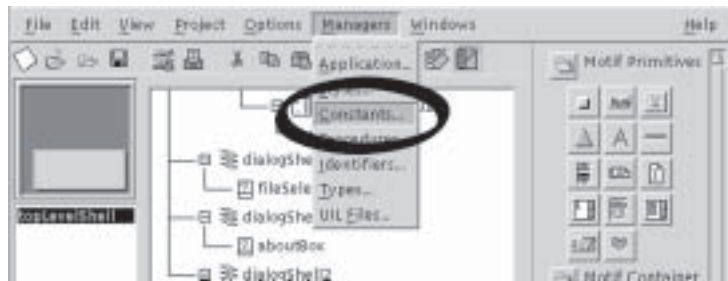


Using Constants

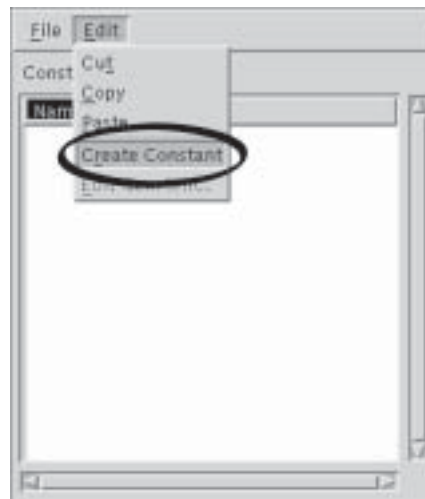
Now you will create a style for menus, which specifies a different font and margin spacing. The margin spacing will be defined using a constant.

Creating Constants First, use the Constant Manager to create the constant MENU_MARGIN, as follows:

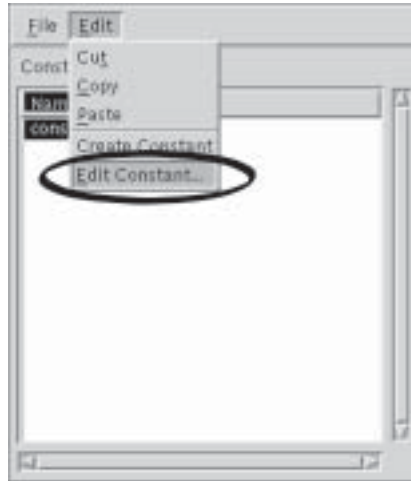
1. Select **Constants...** under **Managers** in the Main Window.



2. In the Constant Manager **Edit** menu, select **Create Constant**.



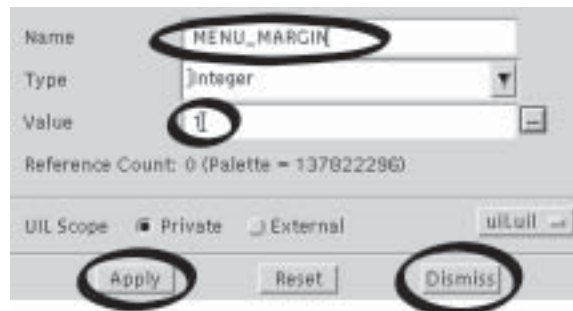
- In the Constant Manager **Edit** menu, select **Edit Constant**. The Constant Editor appears.



- In the Constant Editor, enter the following values:

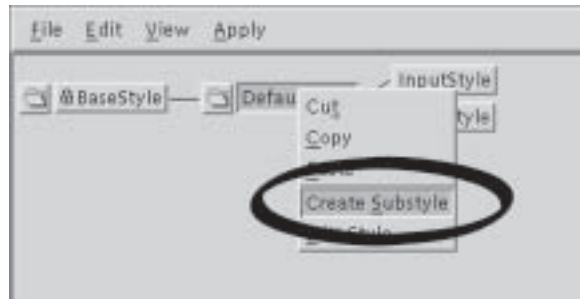
Field	Value
Name	MENU_MARGIN
Value	1

- Click on **Apply**.
- Dismiss** the Constant Editor and Constant Manager.



Creating a Menu Style To create the MenuStyle, perform the following steps:

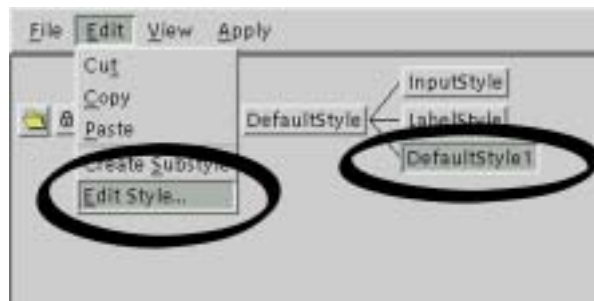
1. Open the Style Manager.
2. Move the cursor over DefaultStyle and press MB3.
3. Select **Create Substyle** on the popup menu. This creates a style named DefaultStyle1.



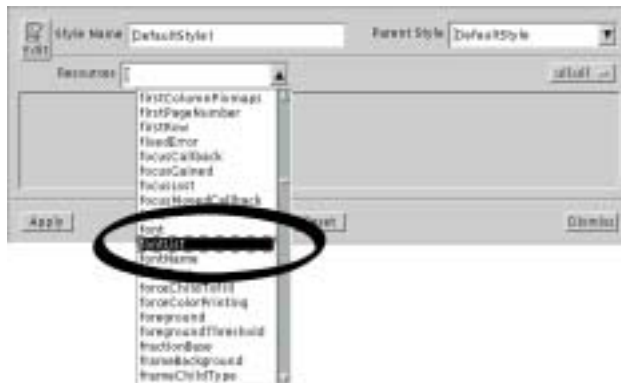
Setting Style Resources

Set resources for the new style by performing the following steps:

1. Select DefaultStyle1.
2. Select **Edit Style...** from the the Constant Manager **Edit** menu. The Style Editor appears and is loaded for DefaultStyle1.



3. For DefaultStyle1, change the Style Name to MenuStyle.
4. In the Style Editor, click the arrow button to the right of the Resources label to display the list of available resources.
5. By clicking MB1 three times, add the following resources to MenuStyle:
 - fontlist
 - marginHeight
 - marginWidth



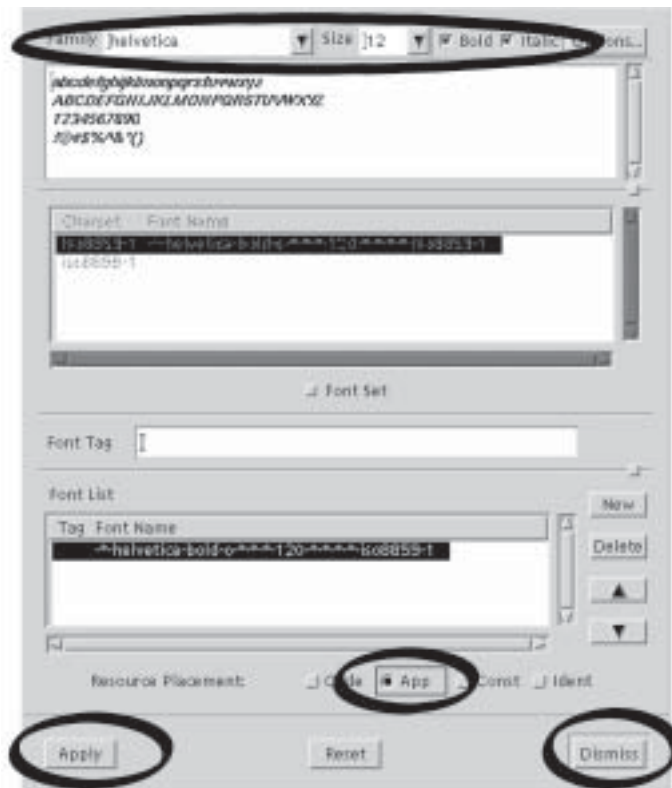
6. Click on the arrow button to the right of the Resources label to collapse the list of resources. The resources selected on the previous page will appear in the Style Editor.



7. Click on the “. . .” to the right of the fontlist entry field to start the fontlist editor. Select the following values:

Resource	Value
Family	Helvetica
Size	12
Bold	checked
Italic	checked
Resource Placement	App

8. After setting these values, click **Apply** and then click **Dismiss**.



9. Next, we'll set `marginHeight` and `marginWidth` to the `MENU_MARGIN` constant you defined earlier in this section. You can do this in either of two ways. We'll illustrate both in the next two steps.
10. Set `marginHeight` to `MENU_MARGIN` by typing it directly into the Style Editor.



Note: When you click on the check mark to enter the resource definition, the source placement automatically changes to **Const**.

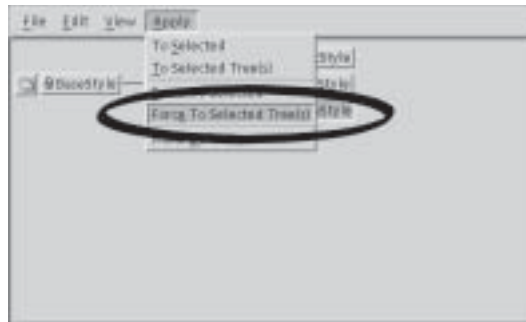
11. Click on the resource placement option menu to right of the `marginWidth` resource. Click on the **Const** option and select `MENU_MARGIN` by clicking on it with MB1.



Note: In this case, you can't use the extended editor, the Dimension Editor, because it does not accept a text string as input.

Applying the MenuStyle

17. Click on the **Apply** button in the Style Editor.
18. **Dismiss** the Style Editor.
19. Select menuBar on the Browser.
20. Select MenuStyle in the Style Manager.
21. From the **Apply** menu in the Style Manager, select **Force to Selected Tree**. From the **Apply** menu in the Style Manager, select **Force to Selected Tree**.

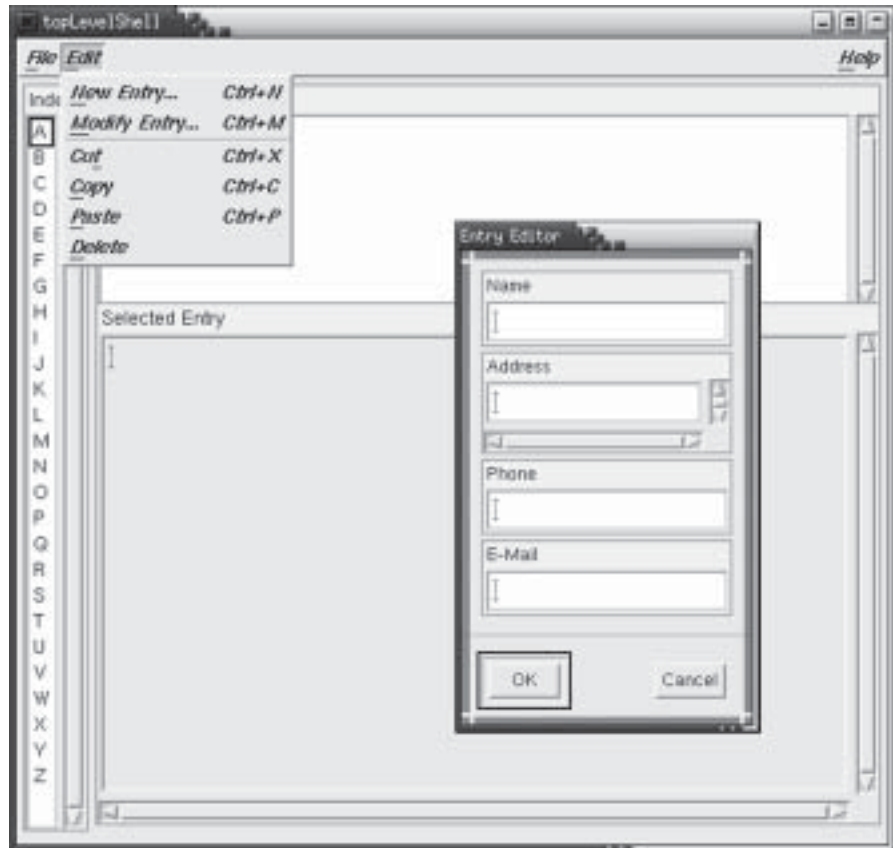


All of the menus in the address book now have the MenuStyle. Notice that the margins have changed and that the menu text is **bold italic**.

You can now use the Constant Manager to change the value of the MENU_MARGIN constant, which automatically updates the interface. Large values of MENU_MARGIN create larger pulldown menus.

Status Check!

Your interface should now look like the following figure:

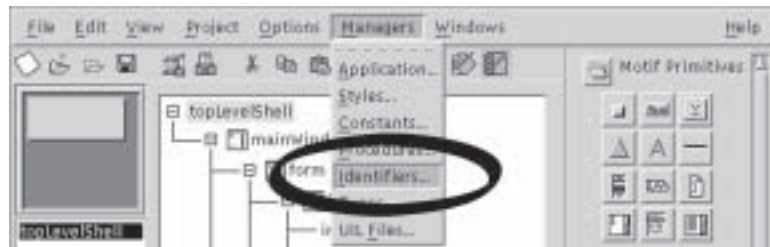


Using Identifiers

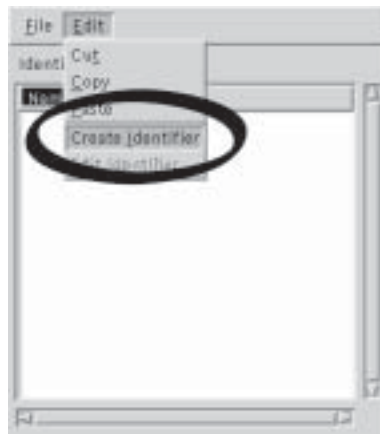
Identifiers allow you to specify actions that will be performed or values that will be set when your application runs. As an example, you can create an identifier that will generate a title to replace `topLevelShell`:

Creating an Identifier

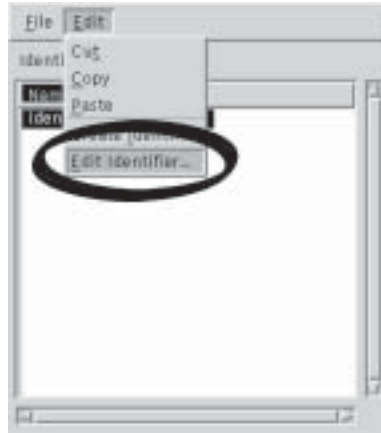
1. Select `topLevelShell` on the Browser tree.
2. Open the Identifier Manager by selecting the **Managers** menu item in the Main Window and then selecting the **Identifiers...** option.



3. Select **Create Identifier** from the **Edit** menu on the Identifier Manager.



4. Select **Edit Identifier** to call up the **Identifier Editor**.



5. On the Identifier Editor, enter the following values:

Identifier	Value
Name	GetTitleString()
Type	String

Note: Remember to click the button to apply the Name!

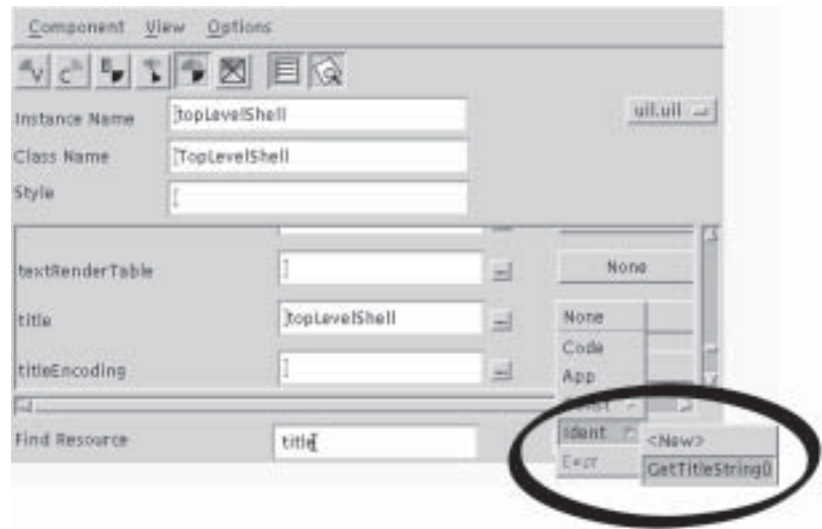


6. **Apply** the changes to the identifier.
7. **Dismiss** the Identifier Editor and Identifier Manager.

Applying an Identifier

To apply your new identifier to your interface, perform the following steps:

1. On the Browser, select `topLevelShell`.
2. In the Resource Editor, on the resource placement option menu associated with the `title` resource, select **Ident**, which displays the available identifiers.
3. Select **GetTitleString()**.



4. The appearance of your interface in Builder Xcessory has not changed. The `GetTitleString()` function will be called at run time to get a value for the `title`.

Important: Builder Xcessory will not generate any source code to define or declare identifiers. You must provide the declarations and definitions in either user code blocks or in code modules not generated by Builder Xcessory.

For more information on identifiers, see *Identifiers* in the *Builder Xcessory Reference Manual*.

Saving the Interface

Select **Save** from the **File** menu in the Main Window, then click OK. (This assumes that you had previously done a **Save As...** and saved your work in the tutorial_5 directory that you created at the beginning of this tutorial.

Summary

You have now completed tutorial five. In this tutorial, you have:

- Created resource styles using the Style Manager
- Modified styles using the Style Editor
- Applied styles to an interface
- Created substyles
- Used constants in defining a style
- Used an identifier to define run-time aspects of an interface