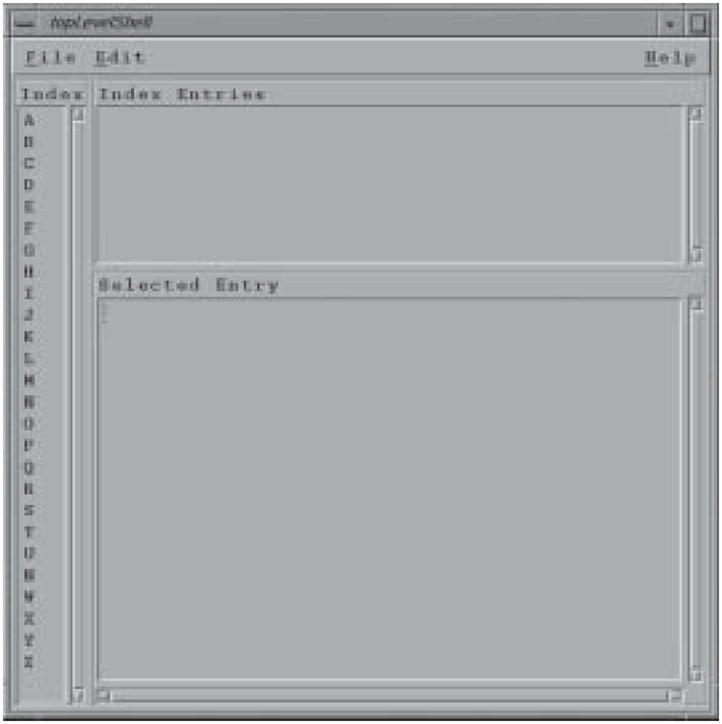
ICS Tutorials: Manipulating Basic Objects

In this tutorial, you will begin to create the sample application - a common address book. The address book consists of three panels:

- Index—letters of the alphabet
- Index Entries—a listing of all items beginning with a particular letter
- Selected Entry—information stored for a single item

Completed Application

The finished application will look something like the following figure:



Before You Begin This Tutorial

Make a new directory called `tutorial_3` and change directories to `tutorial_3`.

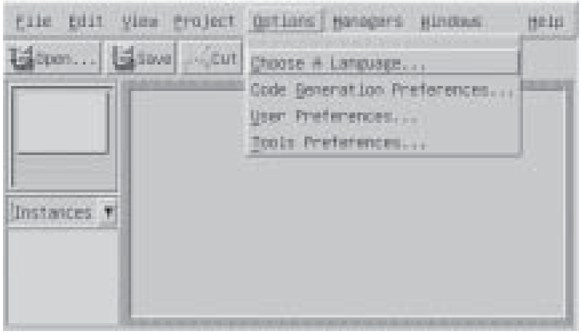
- If you have not started Builder Xcessory yet, enter the following at the prompt in the `tutorial_3` directory:
  % bx
- If you are already running Builder Xcessory, select **New** from the Main Window **File** menu to begin a fresh session. If you have questions about clearing the interface, review *Clearing an Interface* in Tutorial Two.

Choosing a language

Selecting A Code Generation Language

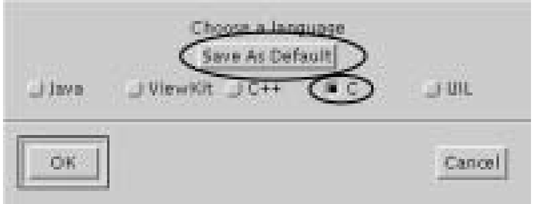
In order to create an interface with Builder Xcessory, you must first select a default code generation language, as follows:

1. Select **Choose a Language** from the **Options** menu of the Browser.



2. Select **C**.
3. If you want to make C the default code generation language for future projects, select **Save As Default**.

Note: Except where explicitly noted, the remainder of these tutorials assume you have chosen C.


Choose a language

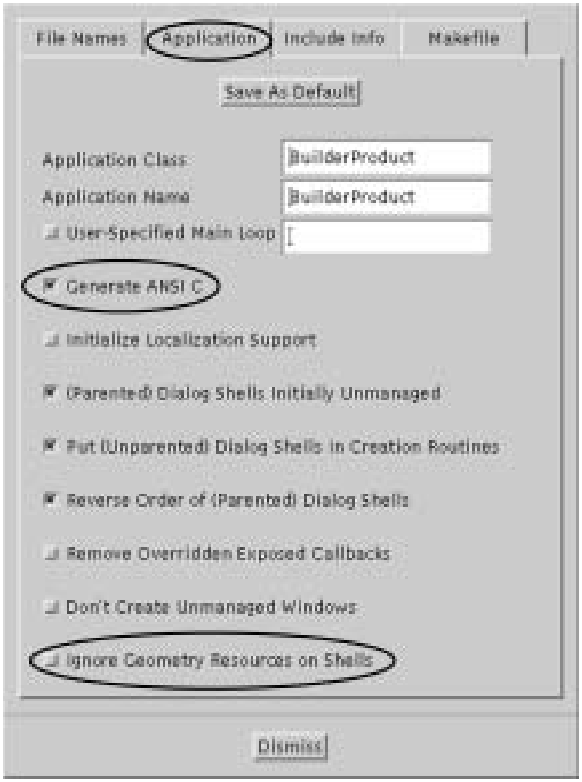
○ Java ○ ViewKit ○ C++ ● C ○ UIL

Code generation preferences

Select **Code Generation Preferences** from the Options menu. On the **Application** tab, you should normally select the following toggles:

Generate ANSI C
This should be selected unless you explicitly need another C standard.

Ignore Geometry Resources on Shells
This toggle should usually be set. It allows a shell widget to use a child’s geometry, rather than forcing the shell geometry on the child.



Application Class: BuilderProduct

Application Name: BuilderProduct

☐ User-Specified Main Loop

☑ Generate ANSI C

☐ Initialize Localization Support

☑ (Parented) Dialog Shells Initially Unmanaged

☑ Put (Unparented) Dialog Shells in Creation Routines

☑ Reverse Order of (Parented) Dialog Shells

☐ Remove Overridden Exposed Callbacks

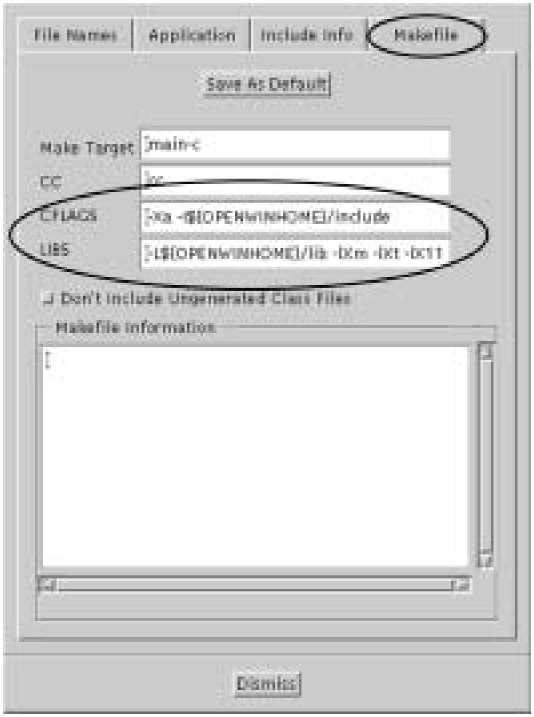
☐ Don't Create Unmanaged Windows

☐ Ignore Geometry Resources on Shells

Some suggested code generation preferences

Makefile preferences

Now select the **Makefile** tab and check that the libraries, library paths, and other link flags listed under **LIBS** are correct for your platform.



Make Target: main-c
CC: cc
CFLAGS: -Xa -I${OPENWINHOME}/include
LIBS: -L${OPENWINHOME}/lib -lXm -lXt -lX11
❑ Don't Include Ungenerated Class Files
Makefile Information

In general, you need the libraries required to link a Motif program. You also need the correct location of Include files and other flags required to compile a C program on your system (since you have chosen C as your code generation language). The location of the Motif libraries and Include files vary from operating system to operating system and your system administrator may have also modified the standard configuration. If you are not sure of the location of the items that are needed, check with your system administrator.

Hint: On Linux systems, the OpenMotif libraries are located in:

```
/usr/X11R6/lib
```

and the Motif Include files are stored in

```
/usr/X11R6/include/Xm
```

Consequently, a common addition to the CFLAGS line is:

```
-I/usr/X11R6/include/
```

and a common addition to the LIBS line is:

```
-L/usr/X11R6/lib
```

Working with widgets

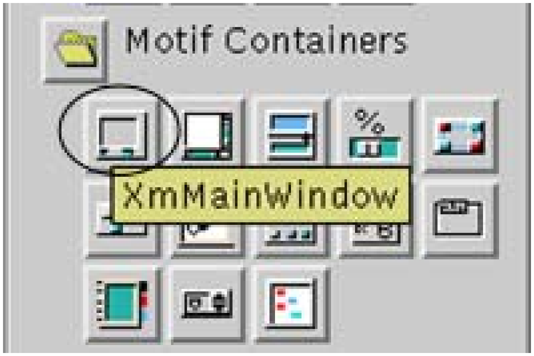
The following sections describe how to create and manipulate widgets.

Creating and Resizing a widget instance

Create and resize an instance of the XmMainWindow widget using the following procedure:

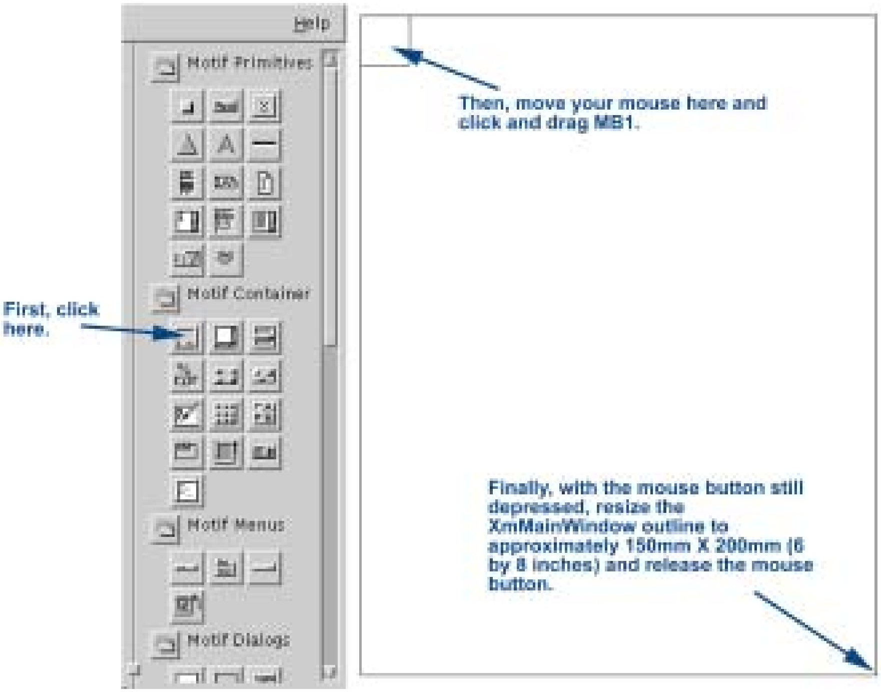
1. Select XmMainWindow by clicking MB1 on its icon in the Containers group on the Palette. An outline of an XmMainWindow appears with the resize cursor in its upper-left corner.

Note: You can display widget names on the Palette by selecting **Both** in Main:Options:User Preferences:Pallete.



2. Using the pointing device, move the cursor to the position on the desktop display where you wish to place the upper-left corner of the XmMainWindow. (See next page for an illustration of steps 2 - 4.)
3. Press MB1, and move the cursor slightly to the right. Notice that the resizing cursor has moved to the lower-right corner of the XmMainWindow widget.
4. With MB1 still depressed, resize the XmMainWindow outline to approximately 6" wide by 8" high (150mm x 200mm). Release MB1.

Your interface now contains a single XmMainWindow, within a `topLevelShell`. Notice that Builder Xcessory automatically created the shell widget for you.



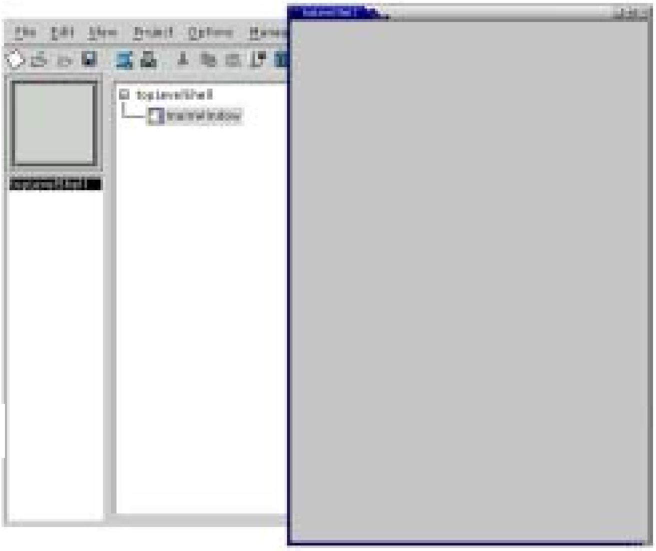
First, click here.

Then, move your mouse here and click and drag MB1.

Finally, with the mouse button still depressed, resize the XmMainWindow outline to approximately 150mm X 200mm (6 by 8 inches) and release the mouse button.

Status check!

The next screen shot below shows an example of XmMainWindow and the BX Browser. Yours should look similar. If it does, now would be a good time to save your work. If you don’t remember how, review Tutorial Two: *Basic Operations*

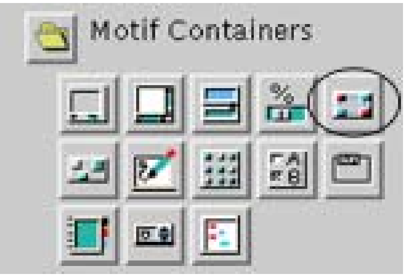


BX Main Window
Note automatic creation
of `topLevelShell`

Resized XmMainWindow

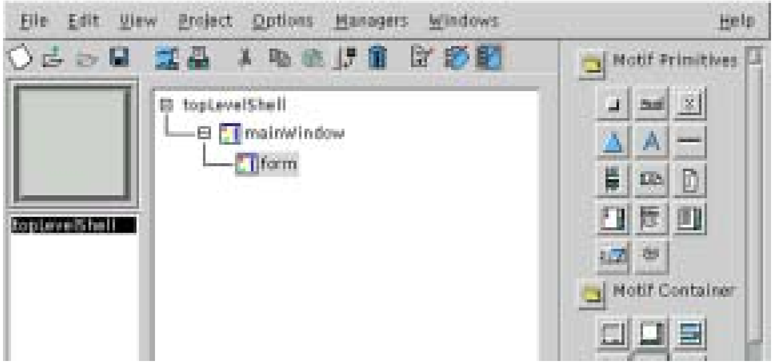
Adding child widgets Insert an XmForm widget as a child of the XmMainWindow by performing the following steps:

1. Select XmForm from the Containers group on the Palette.



2. Place the XmForm inside the XmMainWindow, resized to fill the XmMainWindow. The XmForm is not visible within the XmMainWindow. But if you look at the Browser window of BX, you can confirm that the XmForm was added to your user interface.

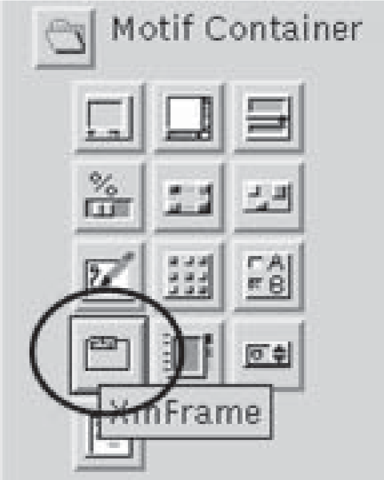
Note: The XmMainWindow resizes itself to fit the XmForm. Just dropping the XmForm in the XmMainWindow will result in XmMainWindow resizing itself to the size of the XmForm icon!



Creating additional widgets

Now add three XmFrames to the XmForm. First add an XmFrame for the Index panel:

1. Select XmFrame from the Containers group on the Palette.

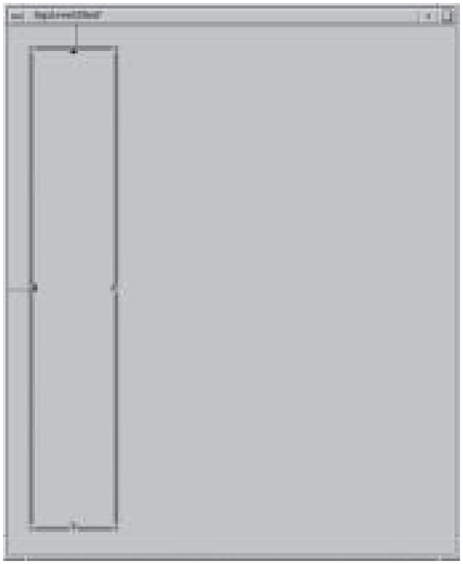


2. Place the XmFrame at the left side of the XmForm, resized to be about 1 inch (25mm) wide and as high as the XmForm working area.

Note: The XmFrame is labeled `frame` in the Browser. The XmFrame appears with a highlighted border and a handle at the midpoint of each side. The left and top handles are automatically connected to the corresponding edges of the XmForm.

To reposition the XmFrame, drag it with MB1. To resize the XmFrame, grab one of the handles at the sides of the widget and drag it with MB1.

Your interface should now look like the figure below.



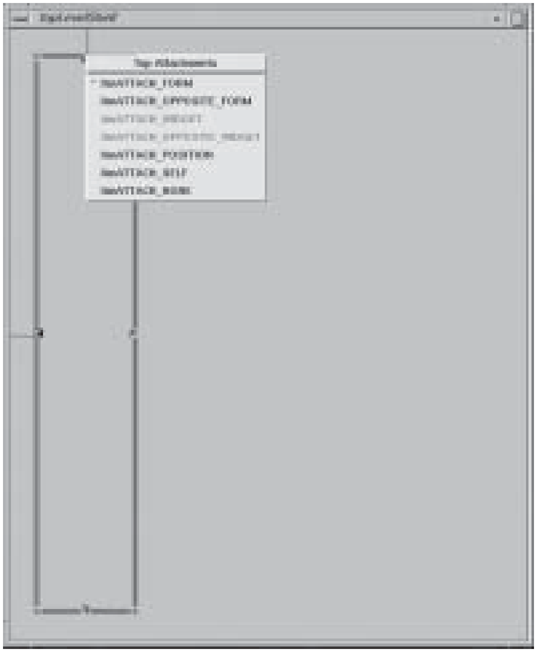
User interface after adding the first XmFrame

Using the Form Editor

The Form Editor allows you to adjust the layout of the widgets in your interface. To set the connection between the XmFrame and the XmForm, and the offset around the XmFrame, use the following procedure:

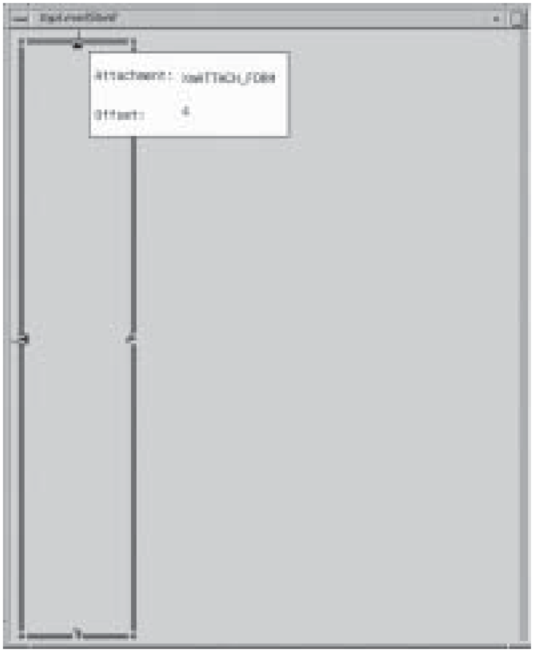
Setting attachment

1. Click and hold MB3 on the top handle.
   A **Top Attachments** popup menu appears. The attachment shows the current value (XmATTACH_FORM in this case).



Setting offsets

2. Click Ctrl/MB3 on the top handle.
   A popup dialog window displays the type of attachment for that handle and the offset (in pixels) from the attached object.
3. Hold Ctrl key and drag the handle until the offset value is 4.



Attachment: XmATTACH_FORM

Offset: 4

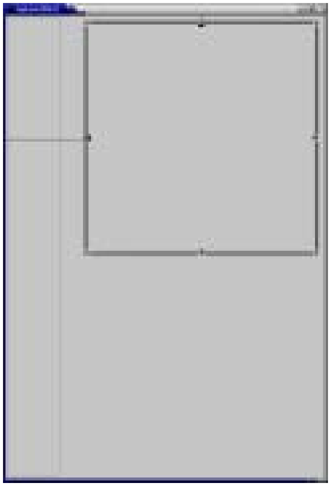
Add index entries panel

4. Release MB3.
5. Repeat steps 2–4 for the left and bottom handles, verifying that in each case the attachment is set to XmATTACH_FORM. Note that the bottom handle is set to XmATTACH_NONE by default. To change the setting, move the pointer into the popup menu and drag it to the new setting.
6. Release MB3.

Add a second XmFrame for the Index Entry panel:

1. Select XmFrame from the Containers group on the Palette.
2. Place the XmFrame to the right side of the XmMainWindow, stretch it to extend about half way down.

Note: The XmFrame is labeled `frame1` in the Browser. The left and top handles are automatically connected to the XmForm.



Setting attachements

To set the connection for `frame1` and the various offsets, use the following procedure:

1. Click on the left attachment handle with MB1 and do not release.
2. Drag the pointer to the edge of the Index panel, and release MB1. The attachment moves from the edge of the form to the edge of the neighboring XmFrame.

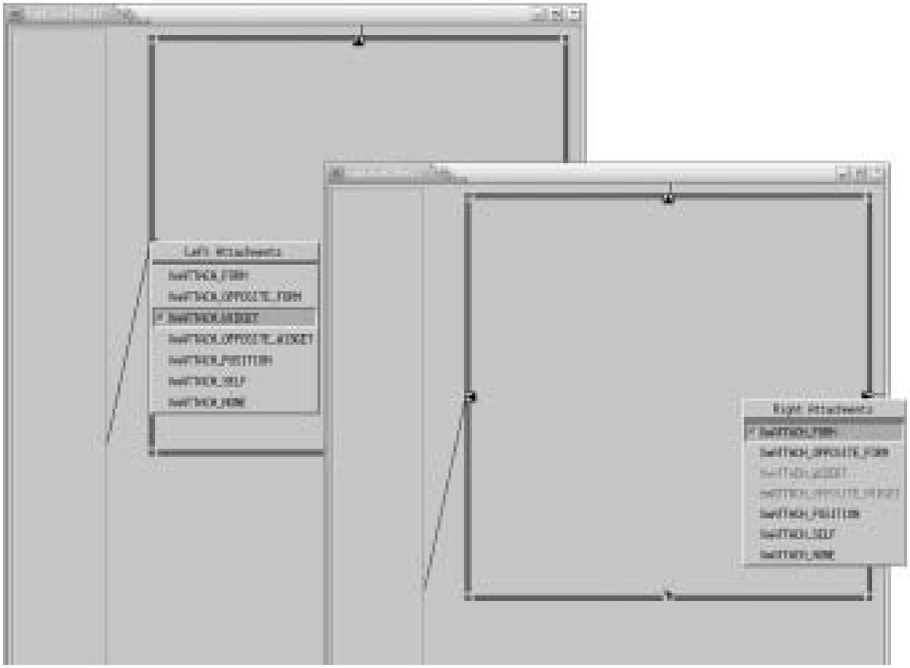
Note: The attachment line goes to the midpoint of the near side of the widget.

Note: If you release MB1 beyond the center line of the target widget, the attachment is made as XmATTACH_OPPOSITE_WIDGET. Simply repeat steps 1 and 2 without releasing beyond the center line of the Index panel.

3. Click MB3 on the left handle. Note that the attachment is now set to XmATTACH_WIDGET.

Setting offsets

4. Use MB3 to set the right handle to XmATTACH_FORM.
5. Set the offsets for the left, top, and right sides to 4 by pressing the Ctrl key and MB3 on each handle and dragging it.

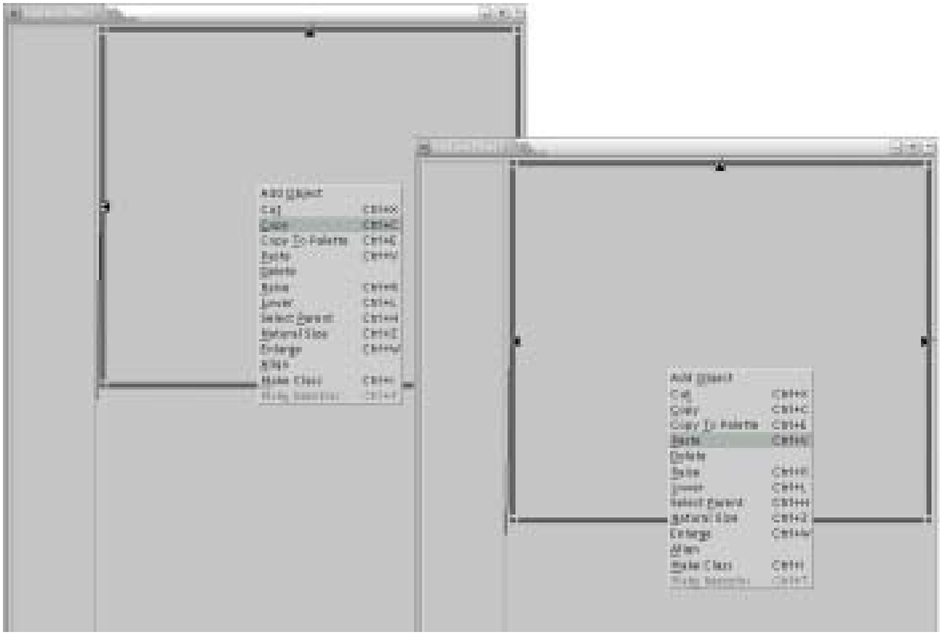


Copying widgets

Add a third XmFrame for the Selected Entry panel by copying `frame1` and resizing it:

Add selected entry panel

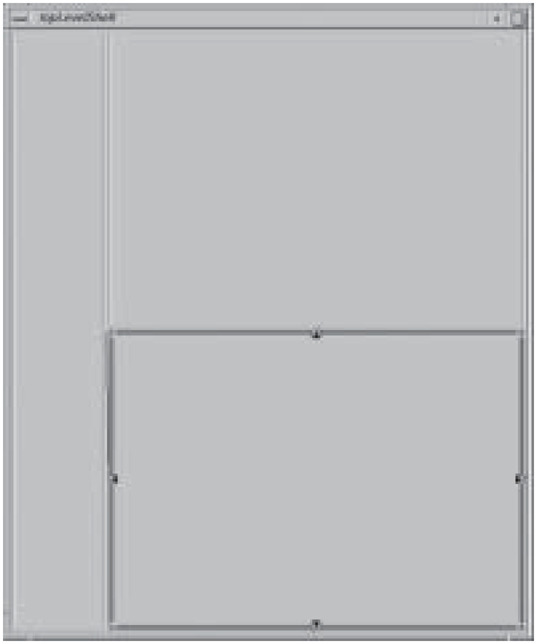
1. Select `frame1` (Index Entries) using MB1.
2. Click on `frame1` with MB3 and slide the pointer to **Copy** the widget.
3. Click MB3 on the form; select **Paste** to create a duplicate of `frame1` (called `frame2`).



4. Use MB1 to place and resize the new `frame2`.
5. Change the right and bottom attachments to XmATTACH_FORM using MB3.
6. Attach the top side of `frame2` to `frame1`.
7. Attach the left side of `frame2` to `frame` (Index) using MB1.
8. Set all four offsets to 4 by pressing the Ctrl key and then pressing MB3 on each handle and dragging it.

Status Check!

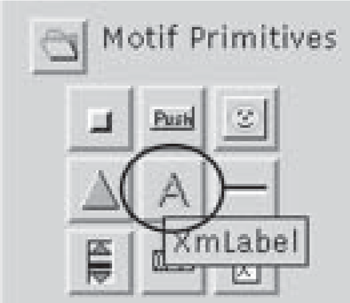
At this point, your user interface should look like the screen below. If it does, now would be a good time to save your user interface.



Adding child widgets

Now that the general structure of the address book is set, you need to fill in the operative widgets. For the Index panel, add a Label and a ScrolledList, using the following steps:

1. Select `frame` (Index panel) in the Browser window.
2. Add an XmLabel to `frame`.

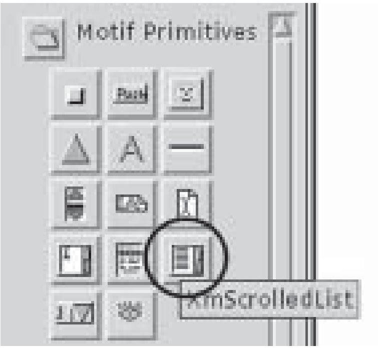


Click on XmLabel on the Palette with MB2, drag it to the `frame` node in the Browser tree. The XmLabel will automatically expand to fill the `frame`. This will be corrected later when we modify the appropriate resources.

Note: If you are working on a complicated interface, it will often be easier to make changes by placing objects in the Browser tree.

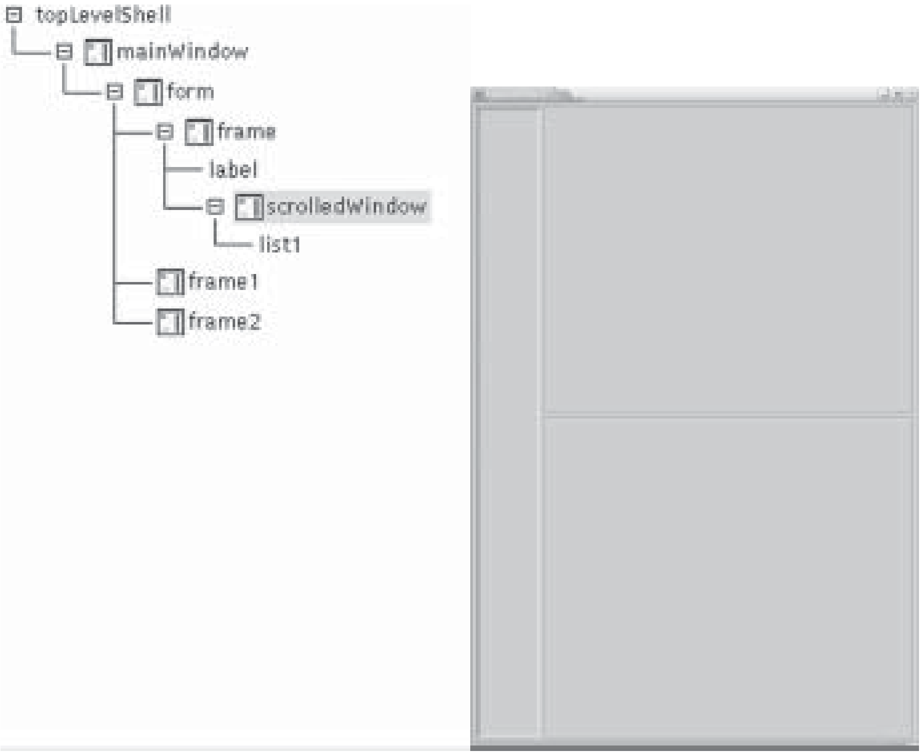
---

3. Add an XmScrolledList to the `frame`. The XmScrolledList text will overlay the XmLabel in the user interface that you are developing.



Status Check!
Save your work!

At this point it is a good idea to confirm that your user interface looks like the screen below. You should also confirm that you have correctly established your widget hierarchy by comparing your Browser with the example provided.



Working with the Resource Editor

The following sections describe how to use the Resource Editor for the objects you have created. The next steps modify the Label and ScrolledList you have added to `frame.`

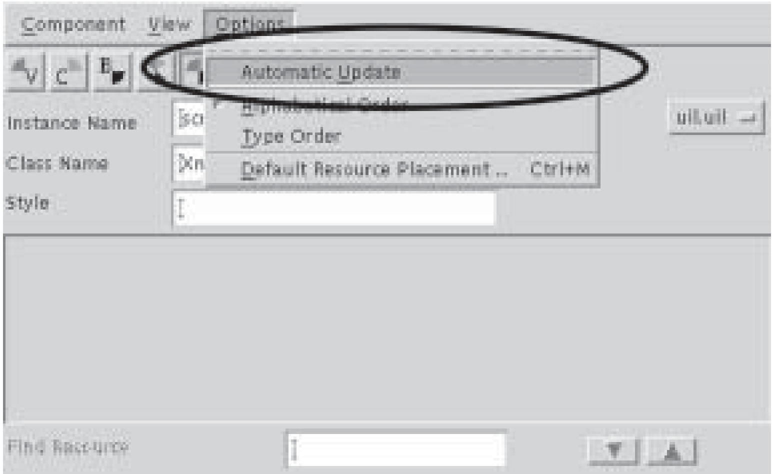
Updating the Resource Editor

You can update the Resource Editor in two ways:

- Click MB1 on the object.

You can select the object either in the interface itself, or on the Browser display. This accomplishes two things: it selects the object, and it updates the Resource Editor so that it displays the resources associated with the object.

- Select **Automatic Update** from the Resource Editor **Options** menu, forcing the Resource Editor to update whenever an object is selected (default).



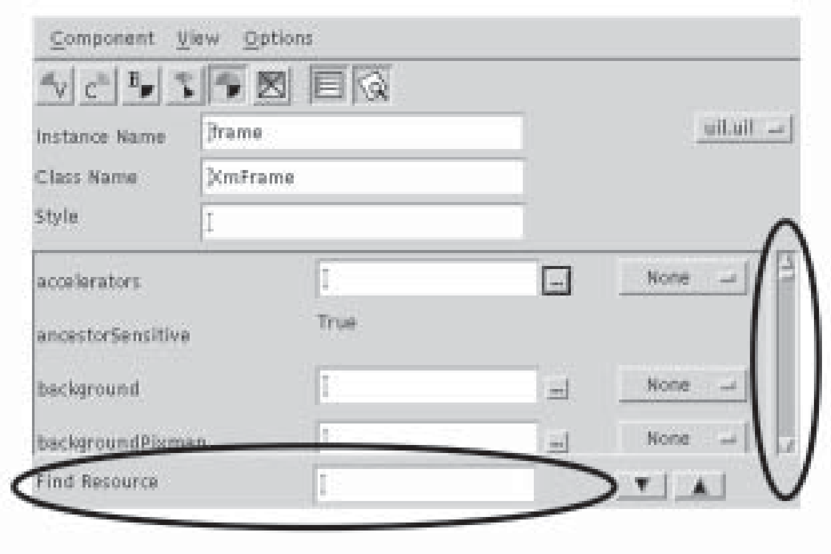
Searching the Resource Editor

To find a resource for a widget in the Resource Editor, first confirm that the widget is currently selected and that the Resource Editor has been updated. The selected object's name should be displayed in the Resource Editor Instance Name field.

Locating resources

To locate a resource, do either of the following:

- Enter the name of the resource you are seeking into the **Find Resource** field. Type the first several letters of the resource in the **Find Resource** search field at the bottom of the Resource Editor until this resource is located. If the **Find Resource** field is not already displayed at the bottom of the Resource Editor, position the cursor within the Resource Editor and press Ctrl/F to toggle it on.
- Use the scrollbar at the right side of the Resource Editor.



Instance Name frame

Class Name XmFrame

Style

accelerators None

ancestorSensitive True

background None

backgroundPixmap None

Find Resource

Widget Size and Position

Working with resources

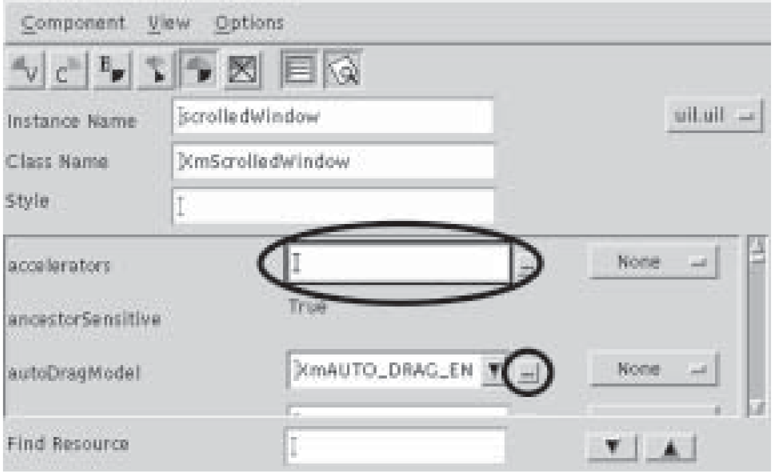
The following sections describe how to modify the value of a resource.

Setting a resource

Now that you have assembled the elements of the Index panel, the next step is to refine the appearance of these widgets.

You can set a widget resource with one of the following procedures:

- Enter the value directly into the resource value entry field to the right of the resource name in the Resource Editor.
- Apply the value to the resource using the appropriate extended editor, by clicking the "`...`" button.



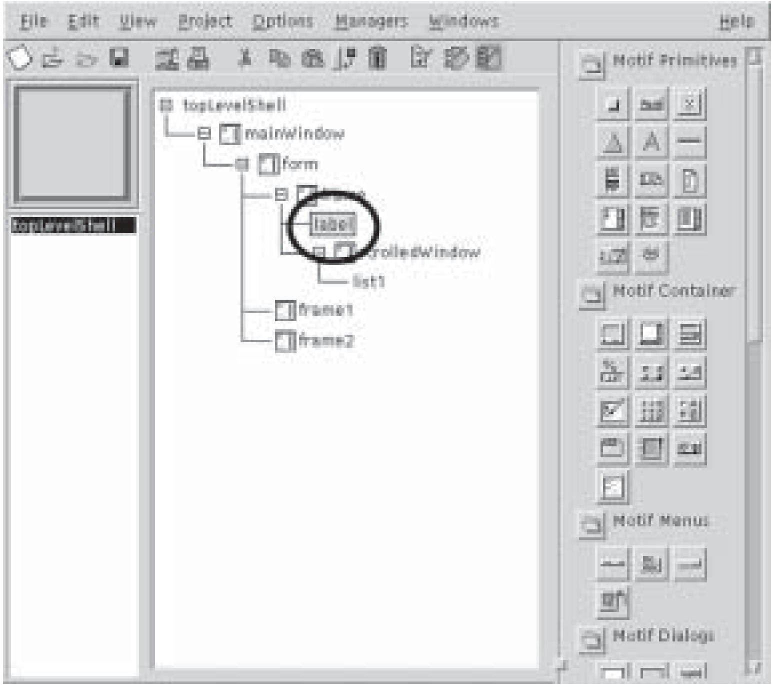
Instance Name ScrolledWindow
Class Name XmScrolledWindow
Style
accelerators
ancestorSensitive True
autoDragModel XmAUTO_DRAG_EN
Find Resource

Modifying Resources

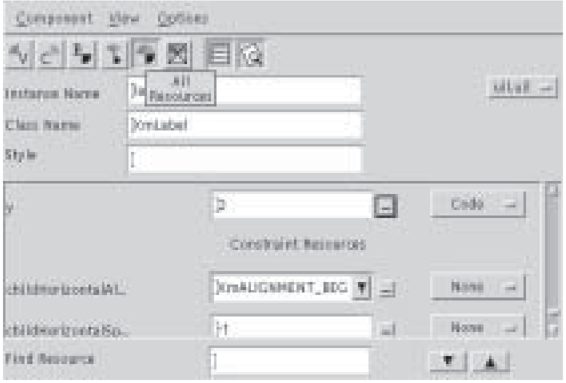
To refine the appearance of the Index panel (`frame`), you will set resources for the two child components—label and `scrolledWindow`—and set frame and all its children to **Natural Size**.

To set the correct values for the XmLabel (`label`), perform the following steps:

1. Click on `label` in the Browser to update the Resource Editor.

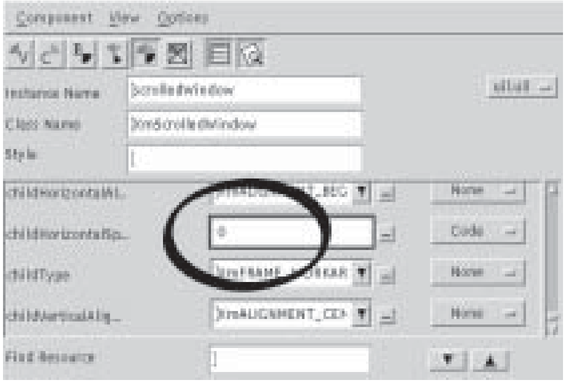


2. Select All Resources



Instance Name
Class Name: XmLabel
Style:
y: 2
childHorizontalAl...: XmALIGNMENT_BEG
childHorizontalSp...: -1
Find Resource:

2. Set the constraint resource `childHorizontalSpacing` to `0`. Click on the current value, and type the new value. Enter Return, or click on the **OK** (check mark) button to the right of the text entry field.



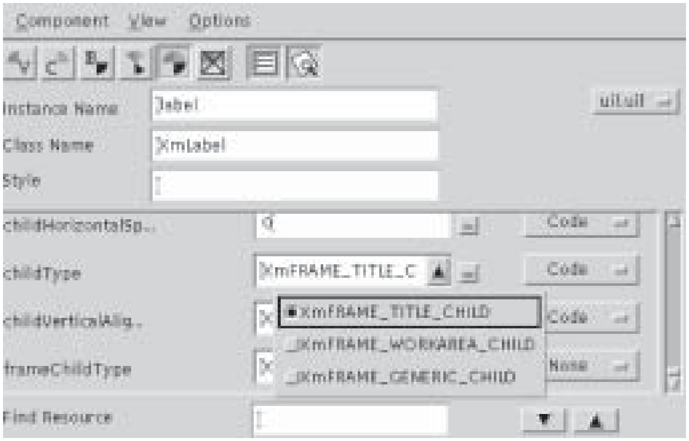
Instance Name: scrolledWindow
Class Name: XmScrolledWindow
Style:
childHorizontalAl...
childHorizontalSp...: 0
childType
childVerticalAlig...: XmALIGNMENT_CEN
Find Resource:

Note: Constraint resources always appear at the end of the resource listing in the Resource Editor.

3. Set the remaining resources as follows:

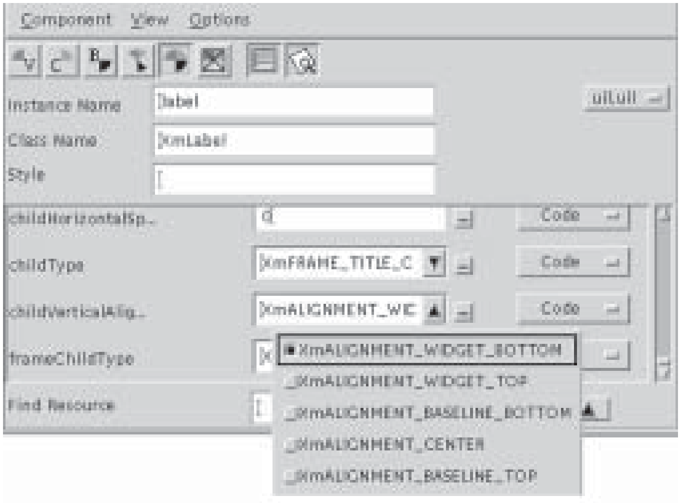
Resource	Value
childType	XmFRAME_TITLE_CHILD
childVerticalAlignment	XmALIGNMENT_WIDGET_BOTTOM

Click on the arrow associated with the text field. A list of possible values appears. Select the correct value according to the table above, and as seen in the following images.



Instance Name: label
Class Name: Xmlabel
Style:
childHorizontalSp..: 0
childType: XmFRAME_TITLE_C
childVerticalAlig..: X
frameChildType: X

■XmFRAME_TITLE_CHILD
XmFRAME_WORKAREA_CHILD
XmFRAME_GENERIC_CHILD



Instance Name: label
Class Name: XmLabel
Style:
childHorizontalSp... 0
childType: XmFRAME_TITLE_C
childVerticalAlig... XmALIGNMENT_WIE
frameChildType: X
Find Resource:

Setting resources for `scrolledWindow`

To set the correct values for the XmScrolledList, perform the following steps:

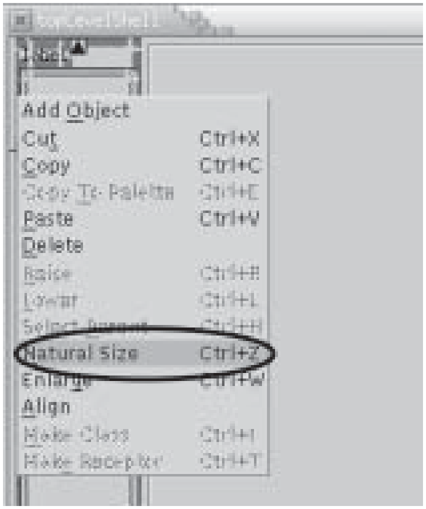
1. Select `scrolledWindow` in the Browser.
2. Set the following constraint resources:

Resource	Value
childType	XmFRAME_WORKAREA_CHILD
childHorizontalSpacing	0

Natural size frame

Set the components of frame to **Natural Size** using the following steps:

1. Select all four components: `frame`, `label`, `scrolledWindow`, `list1.` Press the Ctrl key, then, using MB1, click on each item in the Browser.
2. Click on any of the selected items in the Browser with MB3, and select **Natural Size**. This removes any hard-coded width and height values on the objects, allowing the widgets to adjust appropriately when resources such as label strings are changed and require a different arrangement.



Adding widgets to the Index Entries panel

To fill in the Index Entries panel, add an XmLabel and an XmScrolledWindow to `frame1`, using the same procedure you did for `frame`.

1. Select `frame1` (Index Entries) in the Browser window.
2. Add an XmLabel to `frame1`. Click on XmLabel on the Palette with MB2, drag it to the `frame1` node in the Browser tree. The XmLabel will automatically expand to fill `frame1`. This will be corrected later when we modify the appropriate resources.
3. Add an XmScrolledList to `frame1`. The XmScrolledList text will overlay the XmLabel in the user interface that you are developing.

Setting resources for `frame1`

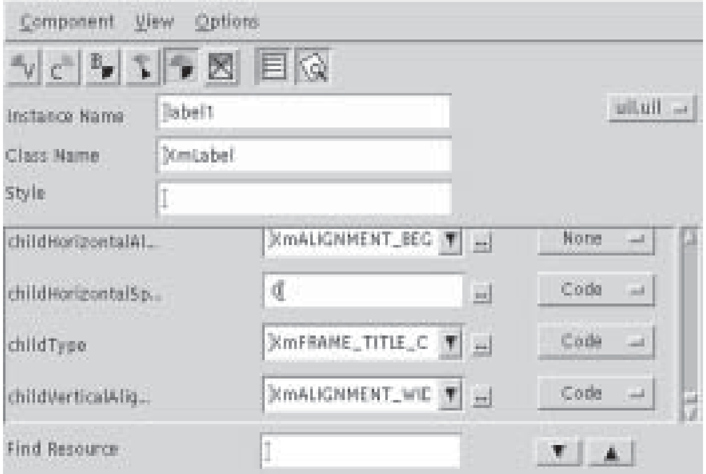
To refine the appearance of the Index Entries panel (`frame1`), set resources for the four components—`frame1`, `label1`, `scrolledWindow1`, and `list2`—and then set them all to **Natural Size**.

***Values for** label1*

To set the correct values for the `label1`, perform the following steps:

1. Click on `label1` in the Browser to update the Resource Editor.
2. Set the constraint resource `childHorizontalSpacing` to `0`. Click on the current value, and type the new value.
3. Set the remaining resources as follows:

Resource	Value
childType	XmFRAME_TITLE_CHILD
childVerticalAlignment	XmALIGNMENT_WIDGET_BOTTOM



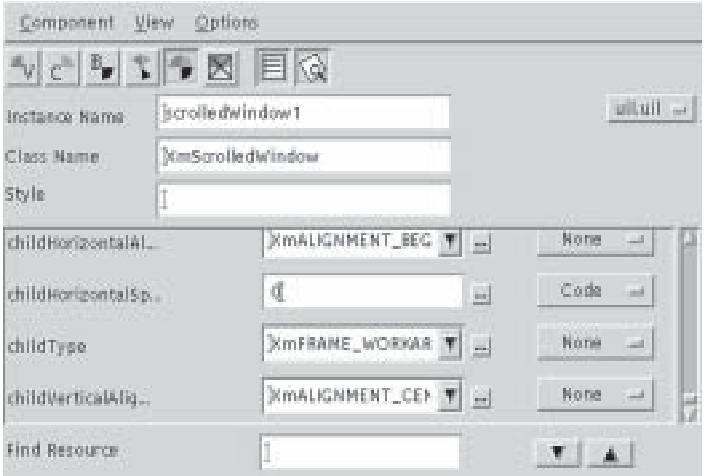
Instance Name label1
Class Name XmLabel
Style
childHorizontalAl... XmALIGNMENT_BEG
childHorizontalSp... 0
childType XmFRAME_TITLE_C
childVerticalAlig... XmALIGNMENT_WIE

***Values for** scrolledWindow*

To set the correct values for the `scrolledWindow1`, perform the following steps:

1. Select `scrolledWindow1` in the Browser.
2. Set the following constraint resources:

Resource	Value
childType	XmFRAME_WORKAREA_CHILD
childHorizontalSpacing	0



Instance Name: scrolledWindow1

Class Name: XmScrolledWindow

Style:

childHorizontalAl... XmALIGNMENT_BEG

childHorizontalSp... 0

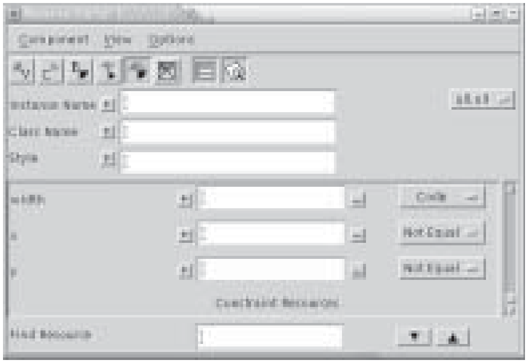
childType XmFRAME_WORKAR

childVerticalAlig... XmALIGNMENT_CEN

Natural size frame1

Set the components of `frame1` to **Natural Size** using the following steps:

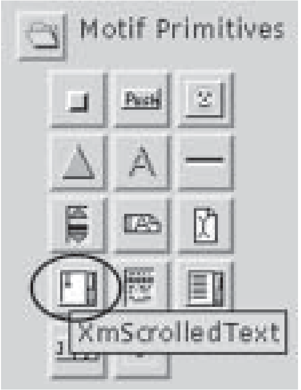
1. Select all four components: `frame1`, `label1`, `scrolledWindow1`, `list2`. Press Ctrl, then click on each item with MB1.
2. Click on any of the selected items in the Browser using MB3, select **Natural Size**.



Adding widgets to the Selected Entry panel

The process for refining `frame2`—the Selected Entry panel—is similar to that for `frame` and `frame1` except that you will use a XmScrolledText instead of a XmScrolledList widget.

1. Select `frame2` (Selected Entry) in the Browser window.
2. Add an XmLabel to `frame2`. Click on XmLabel on the Palette with MB2, drag it to the `frame2` node in the Browser tree. The XmLabel will automatically expand to fill `frame2`. This will be corrected later when we modify the appropriate resources.
3. Add an XmScrolledText to `frame2`. The XmScrolledText will overlay the XmLabel.



Setting frame2 resources

To refine the appearance of `frame2`, you will set resources for the four components—`frame2`, `label2`, `scrolledWindow2`, and `text`—and set them all to **Natural Size**.

Values for label2

To set the correct values for `label2`, perform the following steps:

1. Click on `label2` in the Browser to update the Resource Editor.
2. Set the constraint resource `childHorizontalSpacing` to `0`. Click on the current value, and type the new value.
3. Set the remaining resources as follows:

Resource	Value
childType	XmFRAME_TITLE_CHILD
childVerticalAlignment	XmALIGNMENT_WIDGET_BOTTOM

Values for scrolledWindow2

To set the correct values for `scrolledWindow2`, perform the following steps:

1. Select `scrolledWindow2` in the Browser.
2. Set the following constraint resources:

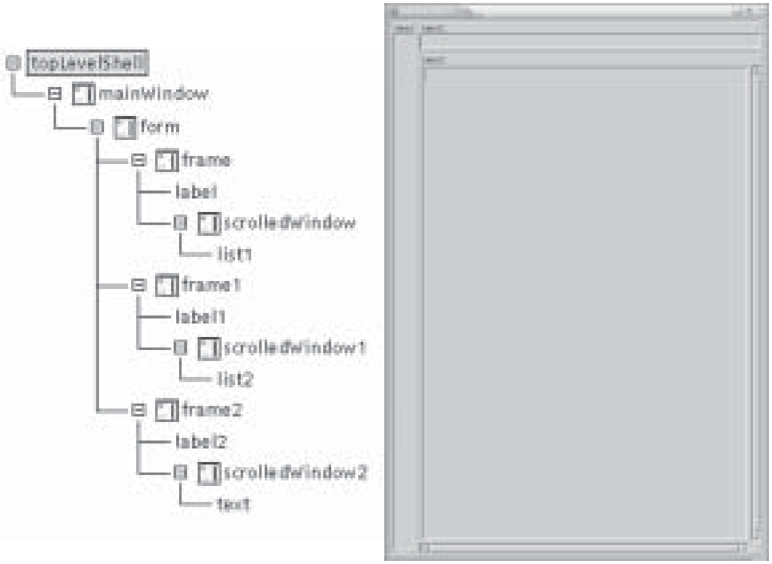
Resource	Value
childType	XmFRAME_WORKAREA_CHILD
childHorizontalSpacing	0

Natural size for frame2

Set the components of `frame2` to **Natural Size** using the following steps:

1. Select all four components: `frame2`, `label2`, `scrolledWindow2,text`. Hold down the Ctrl key and click on each item with MB1.
2. Click on any of the selected items with MB3, select **Natural Size**.

Status Check! Save Your Work!



Now Check your user interface and browser against the examples above

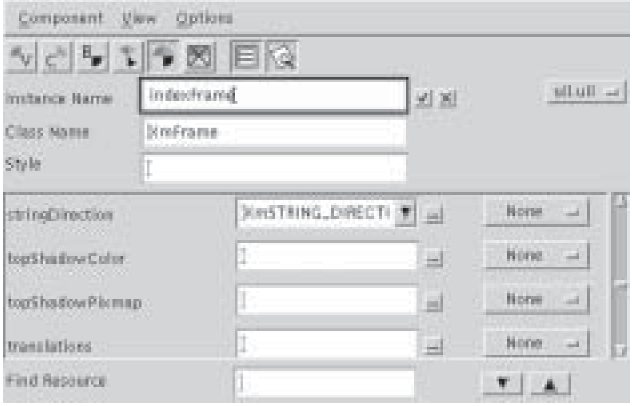
Setting instance names

Using your own instance names when working with Builder Xcessory is good practice. There's no guarantee that the default names will be maintained from session to session. (Besides, imagine sitting down to maintain your application months later and trying to figure out what `pushButton73` is supposed to do.)

Renaming index objects

Let's rename the index objects so that they make more sense. Select each object in `frame` and enter the new name in the Instance Name field at the top of the Resource Editor. To change the Instance Name, perform the following steps:

1. For example, click on `frame` in the Browser window to select it (if it is not already selected). Then in the Resource Editor window, click on the text “`frame`” to the right of the `label` “Instance Name”



Instance Name indexFrame
Class Name XmFrame
Style
stringDirection XmSTRING_DIRECTI
topShadowColor
topShadowPixmap
translations

2. Type the new name.
3. Click on the check mark at the end of the text field, or press Return. Use the following names for the widgets that are part of the `frame` hierarchy:

Object	New Name
frame	indexFrame
label	indexLabel
scrolledWindow	indexSW
list1	indexList

Note: When generating C++ or ViewKit code, Builder Xcessory automatically prepends an underscore character to all instance names to construct the class data member used to store the object ID (following accepted convention). So the object named `indexFrame` is referenced as `_indexFrame` in all generated code.

Renaming index entries objects

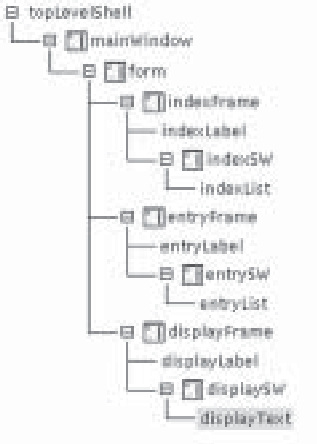
4. Select each object in `frame1` and enter the new name in the Instance Name field at the top of the Resource Editor. Use the following names:

Object	New Name
frame1	entryFrame
label1	entryLabel
scrolledWindow 1	entrySW
list2	entryList

Renaming selected entry objects

5. Select each object in `frame2` and enter the new name in the Instance Name field at the top of the Resource Editor. Use the following names:

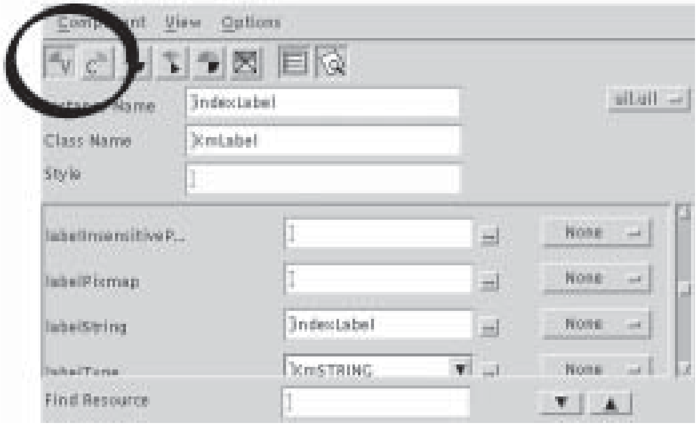
Object	New Name
frame2	displayFrame
label2	displayLabel
scrolledWindow 2	displaySW
text	displayText



Setting label string text

To set the label of each of the three panels in your address book, change the value of the labelString resource, using the following steps:

1. Select `indexLabel` on the Browser.
2. Select Visual Resources

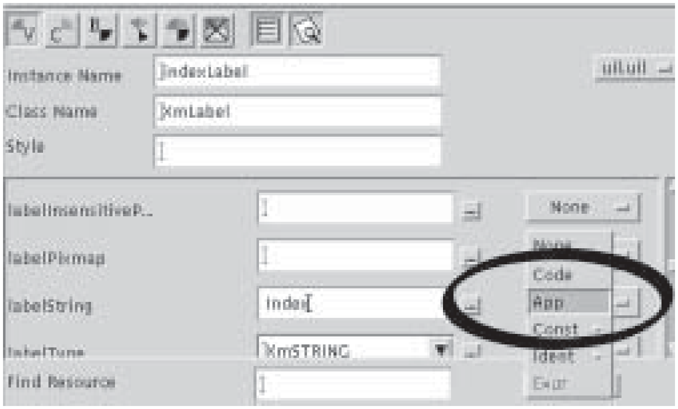


Name indexLabel
Class Name XmLabel
Style
labelInsensitiveP... None
labelPixmap None
labelString indexLabel None
labelType XmSTRING None
Find Resource

3. Search for the `labelString` resource.
4. Click on the resource value text to select it.

Note: The default value for `labelString` is the instance name for the object. In this case, the default `labelString` value is “`indexLabel`”.

5. Type `Index`.
6. Click on the check mark at the end of the text field, or press Return.
7. Click on the option menu to the right of the text field.
8. Select **App** from the popup list.


Instance Name IndexLabel
Class Name XmLabel
Style
labelInsensitiveP...
labelPixmap
labelString index
labelType XmSTRING
Find Resource

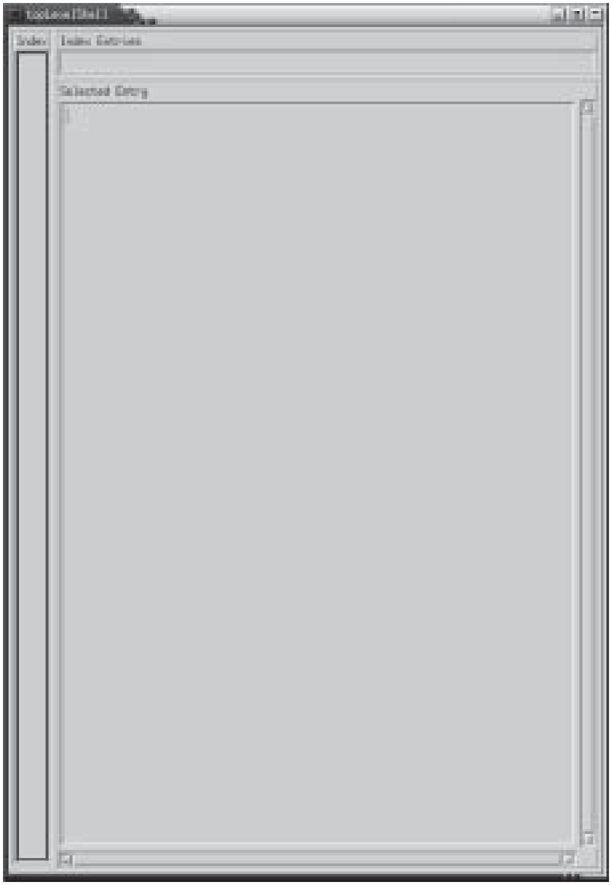
The default setting is **Code**, which means that the value of the resource is hard-coded. Text strings should be kept in an `app-defaults` file, to allow for easily changing language-specific items for internationalization.

Repeat steps 1–8 for `entryLabel` and `displayLabel`. Use the following names for those objects:

Object	labelString	Location
entryLabel	IndexEntries	App
displayLabel	SelectedEntry	App

Status Check!

At this point, your user interface should match the sample provided below. If it does, now would be a good time to save your work!



Your User Interface Should Look Like This After Changing the labelStrings

Fine tuning sizes

When you originally designed your user interface, you were asked to estimate sizes for various forms. At this point, we’re going to adjust the sizes precisely. To do so, follow these instructions:

1. Select each widget listed in the table below and change the specified resources:

Widget	Resource	Value
form	width	576
entryList	visibleItemCount	7
displayText	rows	20
indexList	visibleItemCount	26

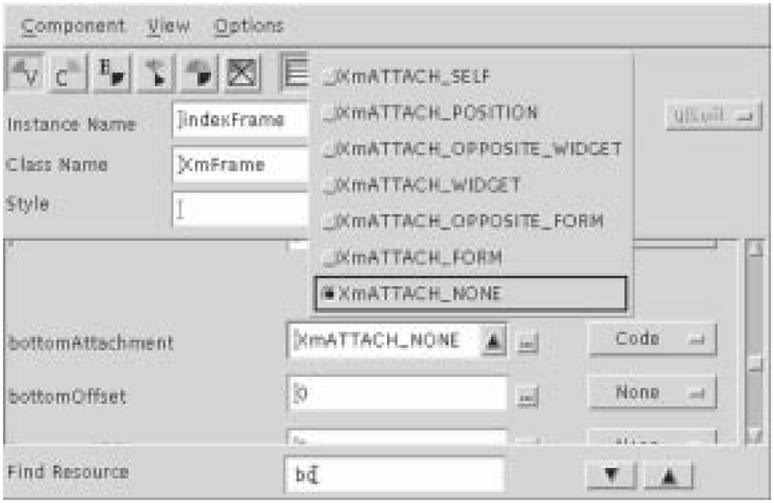
Fixing the Size of Index Entries

You may have noticed that the two panes on the right, `entryFrame` and `displayFrame` are not sized correctly. This occurred because of the `Form` attachments we set earlier.

Note: Try to directly set the number of rows for the `displayText`. You should get the error message “Widget does not allow the resource “rows” to be modified to this value at this time.” This error message is generated because we set the `bottomAttachment` of `displayFrame` to the bottom of the form in an earlier step. The geometry of the parent overrides the geometry setting of the child. Hence you cannot set rows directly. This geometry hierarchy is one of the most powerful aspects of Motif, and it is also one that at times can be the most frustrating.

To resize the three frames, perform the following steps:

1. Select `indexFrame` in the Browser.
2. Find the `bottomAttachment` resource and set it XmATTACH_NONE.

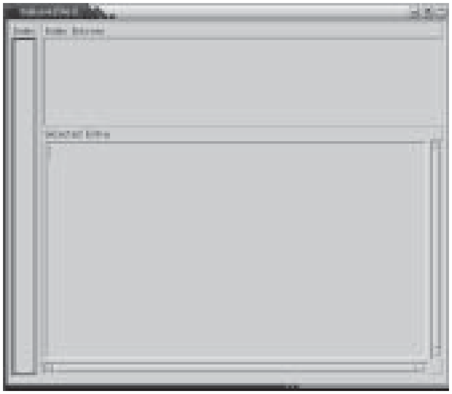


Instance Name indexFrame
Class Name XmFrame
Style
bottomAttachment XmATTACH_NONE
bottomOffset 0
Find Resource bd

3. Select `displayFrame` in the Browser.
4. Find the `bottomAttachment` resource and set it to XmATTACH_NONE too.
5. Set the following resources:

Widget	Resource	Value
displayText	row s	20
entryList	visibleItemCount	7
indexList	visibleItemCount	26

6. Select `mainWindow` in the Browser, and find the resource height in the Resource Editor.
7. Using your Window Manager, set the height to approximately 508 pixals.
8. Find the Width resource for mainWindow and set the width to 576.
9. Reset both to XmATTACH_FORM for the `bottomAttachment` of both `indexFrame` and `displayFrame`.



After you have fixed the sizing, the application should look like this.

The Extended Editor

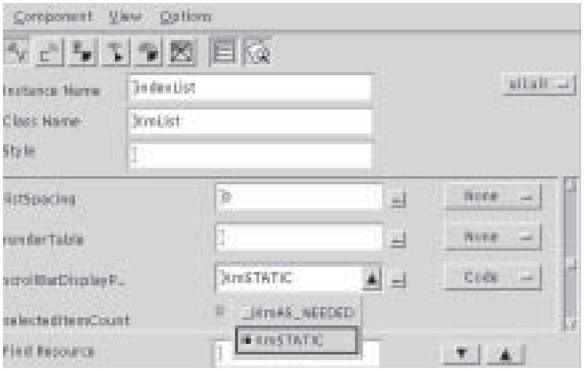
Completing the Index panel

To fill in the Index, you should use an extended editor. Extended editors allow you to repeat actions without exiting from the editor. To fill in the alphabet in the Index, perform the following actions:

1. Select `indexList`.
2. Set the resource `scrollBarDisplayPolicy` to **XmStatic**.

Note: XmStatic means that the scrollbar is always visible.

4. Find the items resource, and use an extended editor to change it, as described in the following section.



Instance Name indexList

Class Name XmList

Style

listSpacing 0

renderTable

scrollBarDisplayP... XmSTATIC

selectedItemCount

XmAS_NEEDED

XmSTATIC

Find Resource

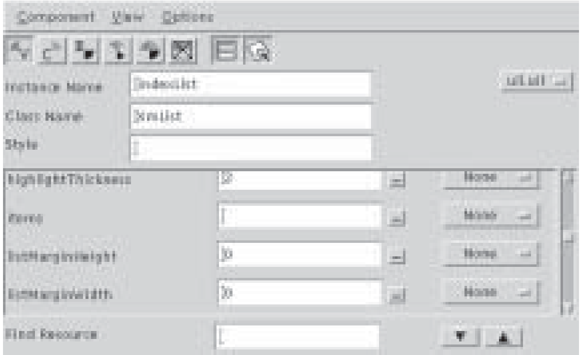
Using an extended editor

Extended editors provide additional flexibility in working with resources. For more information on extended editors, see *Extended Editors* in the *Builder Xcessory Reference Manual*.

To set the value of the items resource of `indexList` to include the letters of the alphabet:

Starting an extended editor

1. Click the (`...`) button to the right of the `items` text field.



Instance Name: indexList

Class Name: XmList

Style:

highlightThickness: 2

items:

listMarginHeight: 0

listMarginWidth: 0

Find Resource:

2. This displays the extended editor appropriate for the items resource, the XmStringTable Editor. (The window title includes the name of the resource, in this case “items Editor.”)
3. Enter the first new resource value, “A” (without quotes), in the text entry area at the top of the editor. Do not press return when you finish typing; that would simply add another (blank) line to the label.

Note: The XmStringTable editor has three main areas: a text entry area at the top, a “preview” area in the middle, and a list of previous entries at the bottom.

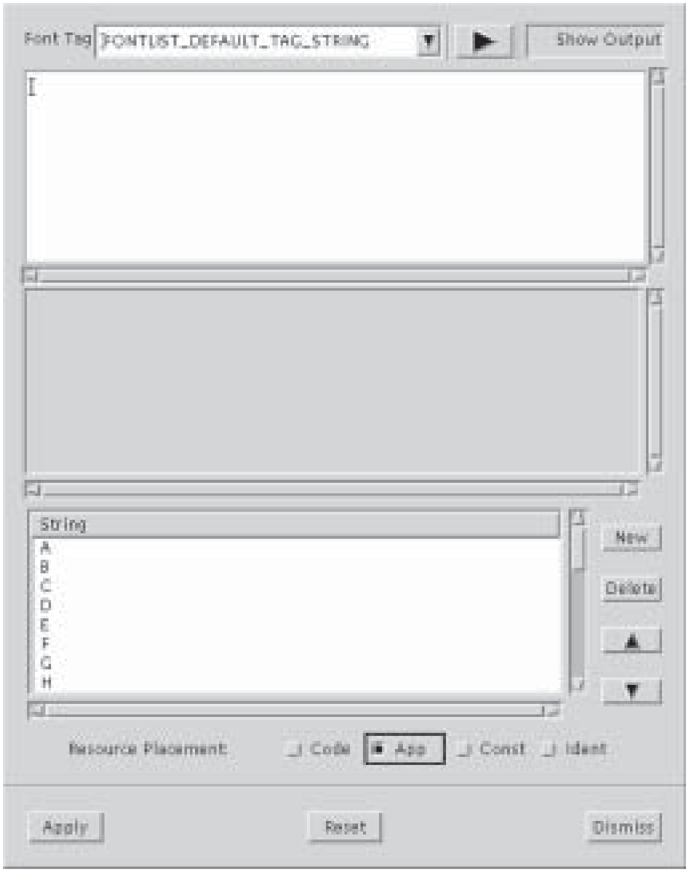
Resource placement

4. Hold the Shift key and press Return. This enters the value for the current list item and resets the editor for the next item.

Hint: You can also enter the succeeding items by pressing the New button after you type each letter.

---

5. Repeat steps 2–3 for letters B through Z.
6. Select **App** for resource placement.
7. Click the **Apply...** button.
8. Click **Dismiss** to hide the XmStringTable Editor.



Font Tag FONTLIST_DEFAULT_TAG_STRING

String
A
B
C
D
E
F
G
H

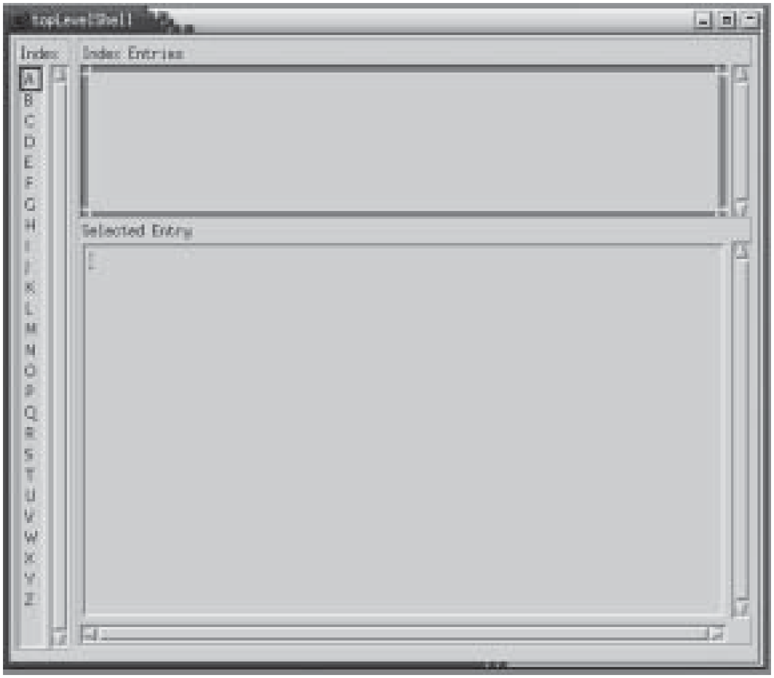
Resource Placement: □ Code ☒ App □ Const □ Ident

Completing the index entries panel

1. Select `entryList` in the Browser window.
2. Update the Resource Editor.
3. Set the resource `scrollBarDisplayPolicy` to **XmStatic**.

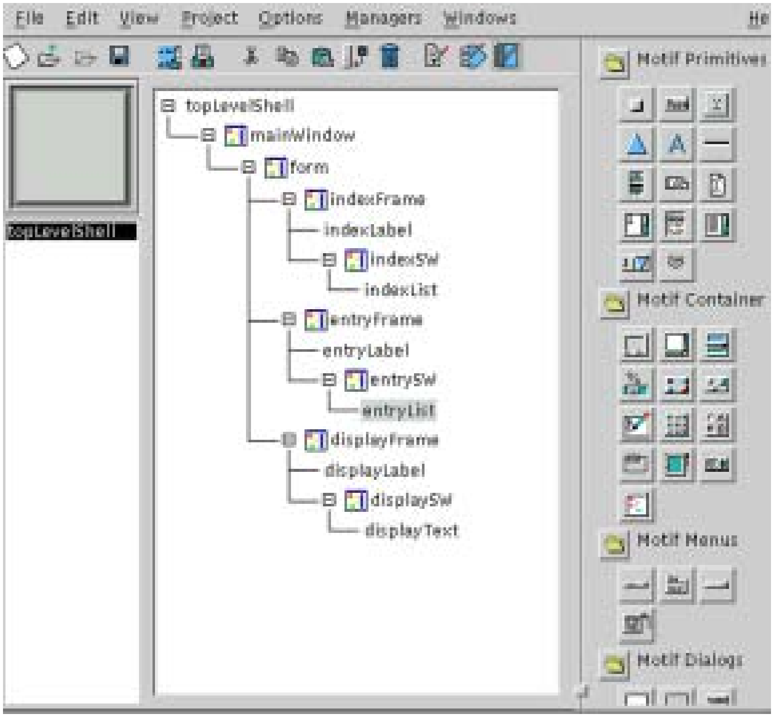
Finished!

Your interface should now look like the following figure:



Your First User Interface with Builder Xcessory!

The instance tree in the Browser should look like the following figure:



Final Interface Displayed in the Browser

Saving the interface Now is the time to save your work. If you are unsure of how to do this, please review *Saving* Your User Interface *in* Tutorial One: Bas*ic Operations*. Make sure you save it in the `tutorial_3` directory you created before starting this tutorial.

Summary

You have now completed Tutorial Three. In this tutorial you have:

- Chosen a code generation language
- Created widget instances
- Added widget children to existing parent widgets
- Positioned and sized widgets
- Modified widget attributes using the Resource Editor
- Changed the storage location for widget resources