Example 1: \texttt{xecute.c}:

Executes any typed \textit{command} after the user confirms the action.

The interface is composed of: A container widget: \texttt{BulletinBoardWidget}, that contains two other widgets:

1. \texttt{LabelWidget}
2. \texttt{PushButtonWidget}

Example Usage:

\% \texttt{xecute "list all files?" "ls -lt"}

This produces the following interface:

\begin{center}
\begin{tikzpicture}
  \node (bb) at (0,0) {Bulletin Board};
  \node (msg) at (bb) {list all files?};
  \node (shell) at (msg) {Shell Window};
  \node (app) at (shell) {Application};
  \node (appstr) at (app) {Application String};
  \node (str) at (appstr) {String};
  \node (height) at (str) {Height};

  \node (yes) at (msg) {Yes};
  \node (no) at (msg) {No};

  \node (push) at (yes) {PushButton};
  \node (label) at (no) {Label};

  \draw (bb) -- (msg);
  \draw (msg) -- (shell);
  \draw (shell) -- (app);
  \draw (app) -- (appstr);
  \draw (appstr) -- (str);
  \draw (str) -- (height);
  \draw (yes) -- (push);
  \draw (no) -- (label);
\end{tikzpicture}
\end{center}

\texttt{xecute.c outline:}

\begin{verbatim}
void main ( int argc, char **argv )
{
    Widget         shell, msg, bb, yes, no;
    XtAppContext   app;
    XmString       xmstr;
    Dimension      height;

    shell = XtAppInitialize( &app, "Xecute", NULL, 0, &argc, argv,
                              NULL, NULL, 0 );

    bb = XtVaCreateManagedWidget
         ("bboard", xmBulletinBoardWidgetClass, shell, NULL );
\end{verbatim}
xmstr = XmStringCreate ( argv[1], XmFONTLIST_DEFAULT_TAG );

// Create msg, yes and no widgets
msg = XtVaCreateManagedWidget ( "message", xmLabelWidgetClass, bb, XmNlabelString, xmstr, XmNx, 0, XmNy, 0, NULL );
yes = XtVaCreateManagedWidget ( "yes", xmPushButtonWidgetClass, bb, XmNx, 0, XmNy, 30, NULL );
no = XtVaCreateManagedWidget ( "no", xmPushButtonWidgetClass, bb, XmNx, 200, XmNy, 30, NULL );

// Add callback functions to yes and no widgets
XtAddCallback ( yes, XmNactivateCallback, YesCallback, ( XtPointer ) argv[2] );
XtAddCallback ( no, XmNactivateCallback, NoCallback, NULL );
XtRealizeWidget ( shell );
XtAppMainLoop ( app );

} void YesCallback ( Widget w, XtPointer clientData, XtPointer callData )
{
    char * cmd = (char *) clientData;
    system ( cmd );
    exit ( 0 );
}

void NoCallback ( Widget w, XtPointer clientData, XtPointer callData)
{
    exit ( 0 );
}
Example 2: A drawing editor: editor.c

The code to create this interface is: editor.c

```c
void main ( int argc, char **argv )
{
    Widget        shell, canvas, panel, commands, options;
    XtAppContext  app;
    shell = XtAppInitialize( &app, "Editor", NULL, 0, &argc, argv, NULL, NULL, 0 );
    panel = XtCreateManagedWidget ( "panel", xmFormWidgetClass, shell, NULL, NULL, 0 );
    commands = XtVaCreateManagedWidget ( "commands", xmRowColumnWidgetClass, panel, XmNnumColumns, 3,
                                          XmNorientation, XmHORIZONTAL,
                                          XmNtopAttachment, XmATTACH_FORM,
                                          XmNrightAttachment, XmATTACH_FORM,
                                          XmNleftAttachment, XmATTACH_FORM,
                                          XmNbottomAttachment, XmATTACH_NONE, NULL );
```
options =
XtVaCreateManagedWidget ( "options", xmRowColumnWidgetClass, panel,
XmNorientation, XmVERTICAL,
XmNtopAttachment, XmATTACH_WIDGET,
XmNtopWidget, commands,
XmNrightAttachment, XmATTACH_NONE,
XmNleftAttachment, XmATTACH_FORM,
XmNbottomAttachment, XmATTACH_FORM, NULL );

canvas =
XtVaCreateManagedWidget ( "canvas", xmDrawingAreaWidgetClass, panel,
XmNtopAttachment, XmATTACH_WIDGET,
XmNtopWidget, commands,
XmNleftAttachment, XmATTACH_WIDGET,
XmNleftWidget, options,
XmNrightAttachment, XmATTACH_FORM,
XmNbottomAttachment, XmATTACH_FORM, NULL );
XtCreateManagedWidget ( "button1", xmPushButtonWidgetClass, commands,
NULL, 0 );
XtCreateManagedWidget ( "button2", xmPushButtonWidgetClass, commands,
NULL, 0 );
XtCreateManagedWidget ( "button3", xmPushButtonWidgetClass, commands,
NULL, 0 );
XtCreateManagedWidget ( "button1", xmPushButtonWidgetClass, options,
NULL, 0 );
XtCreateManagedWidget ( "button2", xmPushButtonWidgetClass, options,
NULL, 0 );
XtCreateManagedWidget ( "button3", xmPushButtonWidgetClass, options,
NULL, 0 );
XtRealizeWidget ( shell );
XtAppMainLoop ( app );

Resource files: Editor & Editor.color

*options*button1*labelString: Option 1
*options*button2*labelString: Option 2
*options*button3*labelString: Option 3
*commands*button1*labelString: Command One
*commands*button2*labelString: Command Two
*commands*button3*labelString: Command Three
! Color resources.

*foreground: white
*XmDrawingArea*background: khaki
*XmDrawingArea*height: 500
*XmDrawingArea*width: 400
*options*background: Blue
*commands*background: Green
*options*XmPushButton*background: Red
*commands*XmPushButton*background: grey50
To associate the resource file with the program before execution, use:

```plaintext
% setenv XENVIRONMENT Editor
% editor
```

To get colors:

```plaintext
% setenv XENVIRONMENT Editor.color
% editor
```

To use no resource file:

```plaintext
% setenv XENVIRONMENT
```

---

**Adding life to editor:**

**Example 3:** `editor2.c`  
**Quit & Draw Squares**

```c
Widget quit;
Display *display;
int screen;
long fgcolor, bgcolor;
XGCValues gcval;
GC draw;
Window window;
int pointx, pointy;
int FirstPt = TRUE;

void main ( int argc, char **argv )
{
    .......
    code here is almost the same as
ever.c with the changes below.

    .......
    change:
    XtCreateManagedWidget
        ("button1", xmPushButtonWidgetClass, commands, NULL, 0);
    XtCreateManagedWidget
        ("button2", xmPushButtonWidgetClass, commands, NULL, 0);
    XtCreateManagedWidget
        ("button3", xmPushButtonWidgetClass, commands, NULL, 0);

to:
    quit  = XtCreateManagedWidget
        ("Quit", xmPushButtonWidgetClass, commands, NULL, 0 );
    no    = XtCreateManagedWidget
        ("no", xmPushButtonWidgetClass, commands, NULL, 0 );
    yes   = XtCreateManagedWidget
        ("yes", xmPushButtonWidgetClass, commands, NULL, 0 );
```
Add the callback function:

```c
void quitCallback ( Widget w, XtPointer clientData, XtPointer callData)
{
    Colormap cmap;
    XColor color, ignore;
    char *colorname = "red";

    XtSetArg(wargs[0], XmNlabelString, XmStringCreateLocalized("Are you sure?"));
    XtSetValues(quit, wargs, 1);
    XtVaSetValues( quit, XmNsensitive, False, NULL);
    XtMapWidget(no);
    XtMapWidget(yes);
    cmap = DefaultColormap (display, screen);
    XAllocNamedColor(display, cmap, colorname, &color, &ignore);
    XtSetArg(wargs[0], XmNbackground, color.pixel);
    XtSetValues(quit, wargs, 1);
}
```

```c
void yesCallback ( Widget w, XtPointer clientData, XtPointer callData)
{
    exit(0);
}
```

```c
void noCallback (Widget w, XtPointer clientData, XtPointer callData)
{
    XtSetArg(wargs[0], XmNlabelString, XmStringCreateLocalized("QUIT "));

    XtSetValues(quit, wargs, 1);
}
```

Add the following code to draw lines on the canvas:

```c
display = XtDisplay(shell);
screen = DefaultScreen(display);
XtVaGetValues ( canvas, XmNforeground, &gcval.foreground, XmNbackground, &gcval.background, NULL );
draw = XtGetGC ( canvas, GCForeground | GCBbackground , &gcval );
XtAddEventHandler(canvas,ButtonPressMask,FALSE, HandleBoardEvents, NULL);
```

```c
....
```
XtVaSetValues(quit, XmNsensitive, True, NULL);
XtMapWidget(quit);
XtUnmapWidget(no);
XtUnmapWidget(yes);

} void HandleBoardEvents( Widget w, XtPointer clientData, XEvent *event,
Boolean *flag )
{
    XDrawRectangle(XtDisplay(w), XtWindow(w), draw,
        event->xbutton.x, event->xbutton.y, 10, 10);
}

To associate the resource file with the program before execution.

% setenv XENVIRONMENT Editor.color

Then type:
% editor2