Example 1  xecute.c:

Executes a command after the user confirms the action.

The interface is composed of:

A container widget: **BulletinBoard** Widget, that contains:

- one **Label** Widget and
- two **PushButton** Widget

E.g.: `% xecute "list all files?"  "ls -lt"

This produces the following GUI:
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);

    xmstr = XmStringCreate( argv[1], XmFONTLIST_DEFAULT_TAG );
    msg = XtVaCreateManagedWidget( "message", xmLabelWidgetClass, bb,
                                    XmNlabelString, xmstr, XmNx, 0, XmNy, 0, NULL );

    XtVaGetValues( msg, XmNheight, &height, NULL );

    //Retrieve the height of label widget, so we know where to place the buttons

    yes = XtVaCreateManagedWidget( "yes", xmPushButtonWidgetClass, bb,
                                    XmNx, 0, XmNy, height + 20, NULL );

    no = XtVaCreateManagedWidget( "no", xmPushButtonWidgetClass, bb,
                                   XmNx, 200, XmNy, height + 20, NULL );

    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;
    if ( cmd )
        system ( cmd );
    exit ( 0 );
}

void NoCallback ( Widget w, XtPointer clientData, XtPointer callData )
{
    exit ( 0 );
}

Example 2: A drawing editor:
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, 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, 
XmNnumColumns, 1, 
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, 
XmNrightAttachment, XmATTACH_FORM, 
XmNleftWidget, options, 
XmNleftAttachment, XmATTACH_WIDGET, 
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
*options*background: Blue
*commands*background: Green
To associate the resource file with the program before execution:

Use:

```bash
% setenv XENVIRONMENT Editor
% editor
```

Try color:

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

**Example 3:** editor2.c  Adding life to editor: **Quit & Draw Squares**

```
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 the same as editor.c
    ...... 

    change:

    XtCreateManagedWidget("button1",xmPushButtonWidget
   (getClass,commands,NULL, 0 );
    XtCreateManagedWidget("button2",xmPushButtonWidget
   Class,commands,NULL, 0 );
    XtCreateManagedWidget("button3",xmPushButton
   WidgetClass,commands,NULL, 0 );

    to:

    quit  = XtCreateManagedWidget
          ("Quit",xmPushButtonWidgetClass,commands,NUL
LL, 0 );
    no    = XtCreateManagedWidget
          ("no",xmPushButtonWidgetClass,commands,NUL
LL, 0 );
    yes   = XtCreateManagedWidget
          ("yes",xmPushButtonWidgetClass,commands,NUL
LL, 0 );

    Add the call back:
    XtAddCallback ( quit,
    XmNactivateCallback, quitCallback,
    NULL );
    XtAddCallback ( no,
    XmNactivateCallback, noCallback,
    NULL );
XtAddCallback (yes, XmNactivateCallback, yesCallback, NULL);

Add the following code to draw lines on the canvas:

display = XtDisplay(shell);
screen = DefaultScreen(display);

XtVaGetValues (canvas, XmNforeground, &gcval.foreground, XmNbackground, &gcval.background, NULL);
gcval.foreground = gcval.foreground ^ gcval.background;
draw = XtGetGC (canvas, GCForeground | GCBbackground, &gcval);

XtAddEventHandler(canvas, ButtonPressMask, FALSE, HandleBoardEvents, NULL);

.....

}

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);
```c
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);
}

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

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

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

Use:

% setenv XENVIRONMENT Editor.color

to associate the resource file with the program before execution.

% editor2