

• MOTIF / X11 TUTORIAL FOR LINUX

- To Compile under Linux (just adapt):

```
gcc -o hello hello.c -lXm -lXt -lX11 -L/usr/X11R6/lib
```

- Tutorial tutorial-X-is-fun.c:

```
/*
    Remember to compile try:
        1) gcc hi.c -o hi -lX11
        2) gcc hi.c -I /usr/include/X11 -L /usr/X11/lib -lX11
        3) gcc hi.c -I /where/ever -L /who/knows/where -l X11

    Brian Hammond 2/9/96.    Feel free to do with this as you will!
*/

/* include the X library headers */
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <X11/Xos.h>

/* include some silly stuff */
#include <stdio.h>
#include <stdlib.h>

/* here are our X variables */
Display *dis;
int screen;
Window win;
GC gc;

/* here are our X routines declared! */
void init_x();
void close_x();
void redraw();

main () {
    XEvent event;          /* the XEvent declaration !!! */
    KeySym key;            /* a dealie-bob to handle KeyPress Events */
    char text[255];        /* a char buffer for KeyPress Events */

    init_x();

    /* look for events forever... */
    while(1) {
        /* get the next event and stuff it into our event variable.
           Note: only events we set the mask for are detected!
        */
        XNextEvent(dis, &event);

        if (event.type==Expose && event.xexpose.count==0) {
            /* the window was exposed redraw it! */
            redraw();
        }
    }
}
```

```

    }
    if (event.type==KeyPress&&
        XLookupString(&event.xkey, text, 255, &key, 0)==1) {
/* use the XLookupString routine to convert the invent
   KeyPress data into regular text.  Weird but necessary...
*/
        if (text[0]=='q') {
            close_x();
        }
        printf("You pressed the %c key!\n", text[0]);
    }
    if (event.type==ButtonPress) {
/* tell where the mouse Button was Pressed */
        int x=event.xbutton.x,
            y=event.xbutton.y;

        strcpy(text, "X is FUN!");
        XSetForeground(dis, gc, rand()%event.xbutton.x%255);
        XDrawString(dis, win, gc, x, y, text, strlen(text));
    }
}

void init_x() {
/* get the colors black and white (see section for details) */
    unsigned long black, white;

    dis=XOpenDisplay((char *)0);
    screen=DefaultScreen(dis);
    black=BlackPixel(dis, screen);
    white=WhitePixel(dis, screen);
    win=XCreateSimpleWindow(dis, DefaultRootWindow(dis), 0, 0,
        300, 300, 5, black, white);
    XSetStandardProperties(dis, win, "Howdy", "Hi", None, NULL, 0, NULL);
    XSelectInput(dis, win, ExposureMask|ButtonPressMask|KeyPressMask);
    gc=XCreateGC(dis, win, 0, 0);
    XSetBackground(dis, gc, white);
    XSetForeground(dis, gc, black);
    XClearWindow(dis, win);
    XMapRaised(dis, win);
};

void close_x() {
    XFreeGC(dis, gc);
    XDestroyWindow(dis, win);
    XCloseDisplay(dis);
    exit(1);
};

void redraw() {
    XClearWindow(dis, win);
};

```

•

```

/* text_box.c — demonstrate simple use of XmNactivateCallback
** for TextField widgets. Create a rowcolumn that has rows of Form
** widgets, each containing a Label and a Text widget. When
** the user presses Return, print the value of the text widget
** and move the focus to the next text widget.
*/

#include <Xm/TextF.h>
#include <Xm/LabelG.h>
#include <Xm/Form.h>
#include <Xm/RowColumn.h>

char *labels [] = { "Name:", "Address:", "City:", "State:", "Zip:" };

main (int argc, char *argv [])
{
    Widget          toplevel, text_w, form, rowcol, label_w;
    XtAppContext    app;
    int             i;
    void            print_result (Widget, XtPointer, XtPointer);
    Arg             args [8];
    int             n;

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaAppInitialize (&app, "Demos", NULL, 0, &argc, argv, NULL,
                                sessionShellWidgetClass, NULL);

    rowcol = XmCreateRowColumn (toplevel, "rowcol", NULL, 0);

    for (i = 0; i < XtNumber (labels); i++) {
        n = 0;
        XtSetArg (args [n], XmNfractionBase, 10);          n++;
        XtSetArg (args [n], XmNnavigationType, XmNONE);    n++;
        form = XmCreateForm (rowcol, "form", args, n);

        n = 0;
        XtSetArg (args [n], XmNtopAttachment, XmATTACHFORM);      n++;
        XtSetArg (args [n], XmNbottomAttachment, XmATTACHFORM);  n++;
        XtSetArg (args [n], XmNleftAttachment, XmATTACHFORM);    n++;
        XtSetArg (args [n], XmNrightAttachment, XmATTACH_POSITION); n++;
        XtSetArg (args [n], XmNrightPosition, 3);                n++;
        XtSetArg (args [n], XmNalignment, XmALIGNMENT_END);      n++;
        XtSetArg (args [n], XmNnavigationType, XmNONE);          n++;
        label_w = XmCreateLabelGadget (form, labels [i], args, n);
        XtManageChild (label_w);

        n = 0;
        XtSetArg (args [n], XmNtraversalOn, True);              n++;
        XtSetArg (args [n], XmNleftAttachment, XmATTACH_POSITION); n++;
        XtSetArg (args [n], XmNleftPosition, 4);                n++;
        XtSetArg (args [n], XmNrightAttachment, XmATTACHFORM);  n++;
        XtSetArg (args [n], XmNnavigationType, XmTAB_GROUP);    n++;
        text_w = XmCreateTextField (form, "text_w", args, n);
        XtManageChild (text_w);

        /* When user hits return, print the label+value of text_w */
        XtAddCallback (text_w, XmNactivateCallback, print_result,

```

```

                                (XtPointer) labels[i]);
        XtManageChild (form);
    }

    XtManageChild (rowcol);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}

/* print_result() — callback for when the user hits return in the
** TextField widget.
*/

void print_result (Widget text_w, XtPointer client_data,
                  XtPointer call_data)
{
    char *value = XmTextFieldGetString (text_w);
    char *label = (char *) client_data;

    printf ("%s %s\n", label, value);
    XtFree (value);

    XmProcessTraversal (text_w, XmTRAVERSENEXT_TAB_GROUP);
}

```

—

*** MOTIF/X11 Tutorial tutorial-buttons-4.c:**

```
#include <Xm/Xm.h>
#include <Xm/PushB.h>
#include <Xm/Form.h>

main (int argc, char **argv)

{ XtAppContext app;
  Widget      top_wid, form,
              button1, button2,
              button3, button4;
  int n=0;

  top_wid = XtVaAppInitialize(&app, "Form1",
                              NULL, 0, &argc, argv, NULL, NULL);

  /* create form and child buttons */

  form = XtVaCreateManagedWidget("form",
                                   xmFormWidgetClass, top_wid, NULL);

  button1 = XtVaCreateManagedWidget("Button 1",
                                     xmPushButtonWidgetClass, form,
                                     /* attach to top, left of form */
                                     XmNtopAttachment, XmATTACHFORM,
                                     XmNleftAttachment, XmATTACHFORM,
                                     NULL);

  button2 = XtVaCreateManagedWidget("Button 2",
                                     xmPushButtonWidgetClass, form,
                                     XmNtopAttachment, XmATTACHWIDGET,
                                     XmNtopWidget, button1, /* top to button 1 */
                                     XmNleftAttachment, XmATTACHFORM, /* left, bottom to form */
                                     XmNbottomAttachment, XmATTACHFORM,
                                     NULL);

  button3 = XtVaCreateManagedWidget("Button 3",
                                     xmPushButtonWidgetClass, form,
                                     XmNtopAttachment, XmATTACHFORM, /* top, right to form */
                                     XmNrightAttachment, XmATTACHFORM,
                                     XmNleftAttachment, XmATTACHWIDGET, /* left to button 1 */
                                     XmNleftWidget, button1,
                                     NULL);

  button4 = XtVaCreateManagedWidget("Button 4",
                                     xmPushButtonWidgetClass, form,
                                     XmNbottomAttachment, XmATTACHFORM, /* bottom right to form */
                                     XmNrightAttachment, XmATTACHFORM,
                                     XmNtopAttachment, XmATTACHWIDGET,
                                     XmNtopWidget, button3, /* top to button 3 */
                                     XmNleftAttachment, XmATTACHWIDGET,
                                     XmNleftWidget, button2, /* left to button 2 */
                                     NULL);
  XtRealizeWidget (top_wid);
  XtAppMainLoop (app);
```

}

*

MOTIF/X11 Tutorial tutorial-list-alpha.c:

```
/* alpha_list.c — insert items into a list in alphabetical order.
*/

#include <Xm/List.h>
#include <Xm/RowColumn.h>
#include <Xm/TextF.h>

main (int argc, char *argv[])
{
    Widget          toplevel, rowcol, list_w, text_w;
    XtAppContext     app;
    Arg              args[5];
    int              n = 0;
    void             add_item(Widget, XtPointer, XtPointer);

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Demos", NULL, 0, &argc, argv, NULL,
                                     sessionShellWidgetClass, NULL);

    rowcol = XmCreateRowColumn (toplevel, "rowcol", NULL, 0);
    XtSetArg (args[n], XmNvisibleItemCount, 5); n++;
    list_w = XmCreateScrolledList (rowcol, "scrolled_list", args, n);
    XtManageChild (list_w);

    n = 0;
    XtSetArg (args[n], XmNcolumns, 25); n++;
    text_w = XmCreateTextField (rowcol, "text", args, n);
    XtAddCallback (text_w, XmNactivateCallback, add_item,
                   (XtPointer) list_w);
    XtManageChild (text_w);

    XtManageChild (rowcol);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}

/* Add item to the list in alphabetical order. Perform binary
** search to find the correct location for the new item position.
** This is the callback routine for the TextField widget.
*/
void add_item (Widget text_w, XtPointer client_data, XtPointer call_data)
{
    Widget          list_w = (Widget) client_data;
    char            *text, *newtext = XmTextFieldGetString (text_w);
    XmString        str, *strlist;
    int             u_bound, l_bound = 0;

    /* newtext is the text typed in the TextField widget */
    if (!newtext || !*newtext) {
        /* non-null strings must be entered */
        XtFree (newtext); /* XtFree() checks for NULL */
        return;
    }

    /* get the current entries (and number of entries) from the List */
    XtVaGetValues (list_w, XmNitemCount, &u_bound,
                   XmNitems, &strlist, NULL);
}
```

```

u_bound--;

/* perform binary search */
while (u_bound >= l_bound) {
    int i = l_bound + (u_bound - l_bound) / 2;

    /* convert the compound string into a regular C string */
    if (!(text = (char *) XmStringUnparse (strlist[i],
                                           XmFONTLIST_DEFAULT_TAG,
                                           XmCHARSET_TEXT,
                                           XmCHARSET_TEXT,
                                           NULL, 0,
                                           XmOUTPUT_ALL)))

        break;

    if (strcmp (text, newtext) > 0)
        u_bound = i - 1; /* newtext comes before item */
    else
        l_bound = i + 1; /* newtext comes after item */
    XtFree (text); /* XmStringUnparse() allocates memory */
}

str = XmStringCreateLocalized (newtext);
XtFree (newtext);

/* positions indexes start at 1, so increment accordingly */
XmListAddItemUnselected (list_w, str, l_bound+1);
XmStringFree (str);
XmTextFieldSetString (text_w, "");
}

```



```

/* simple_popup.c — demonstrate how to use a simple popup menu.
** Create a main window that contains a DrawingArea widget, which
** displays a popup menu when the user presses the third mouse button.
*/

#include <Xm/RowColumn.h>
#include <Xm/MainW.h>
#include <Xm/DrawingA.h>

main (int argc, char *argv[])
{
    XmString      line, square, circle, exit_b, exit_acc;
    Widget        toplevel, main_w, drawing_a, popup_menu;
    void          popup_cb(Widget, XtPointer, XtPointer);
    XtAppContext  app;
    Arg           args[4];
    int           n;

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Demos", NULL, 0, &argc, argv, N
                                     sessionShellWidgetClass, NULL);

    /* Create a MainWindow widget that contains a DrawingArea in
    ** its work window.
    */
    n = 0;
    XtSetArg (args[n], XmNscrollingPolicy, XmAUTOMATIC); n++;
    main_w = XmCreateMainWindow (toplevel, "main_w", args, n);

    /* Create a DrawingArea — no actual drawing will be done. */
    n = 0;
    XtSetArg (args[n], XmNwidth, 500); n++;
    XtSetArg (args[n], XmNheight, 500); n++;
    drawing_a = XmCreateDrawingArea (main_w, "drawing_a", args, n);
    XtManageChild (drawing_a);

    line = XmStringCreateLocalized ("Line");
    square = XmStringCreateLocalized ("Square");
    circle = XmStringCreateLocalized ("Circle");
    exit_b = XmStringCreateLocalized ("Exit");
    exit_acc = XmStringCreateLocalized ("Ctrl+C");
    popup_menu = XmVaCreateSimplePopupMenu (drawing_a, "popup", popup_cb,
                                             XmNpopupEnabled, XmPOPUP.AUTOMATIC,
                                             XmVaPUSHBUTTON, line, 'L', NULL, NULL,
                                             XmVaPUSHBUTTON, square, 'S', NULL, NULL,
                                             XmVaPUSHBUTTON, circle, 'C', NULL, NULL,
                                             XmVaSEPARATOR,
                                             XmVaPUSHBUTTON, exit_b, 'x', "Ctrl<Key>c", exit_acc,
                                             NULL);

    XmStringFree (line);
    XmStringFree (square);
    XmStringFree (circle);
    XmStringFree (exit_b);
    XmStringFree (exit_acc);

    XtManageChild (main_w);
    XtRealizeWidget (toplevel);

```

```

        XtAppMainLoop (app);
    }

    /* popup_cb() — invoked when the user selects an item in the popup menu */
    void popup_cb (Widget menu_item, XtPointer client_data, XtPointer call_data)
    {
        int item_no = (int) client_data;

        if (item_no == 3) /* Exit was selected — exit */
            exit (0);

        /* Otherwise, just print the selection */
        puts (XtName (menu_item));
    }
}

```

MOTIF/X11 Tutorial tutorial-list.c:

```
/* simple_list.c — introduce the List widget. Lists present
** a number of compound strings as choices. Therefore, strings
** must be converted before set in lists. Also, the number of
** visible items must be set or the List defaults to 1 item.
*/
#include <Xm/List.h>

char *months[] = {"January", "February", "March", "April", "May", "June",
                  "July", "August", "September", "October", "November", "December"};

main (int argc, char *argv[])
{
    Widget          toplevel, list;
    XtAppContext    app;
    int             i, n = XtNumber (months);
    XmStringTable   str_list;
    Arg             args[4];

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Demos", NULL, 0, &argc, argv, NULL,
                                    sessionShellWidgetClass, NULL);

    str_list = (XmStringTable) XtMalloc (n * sizeof (XmString));

    for (i = 0; i < n; i++)
        str_list[i] = XmStringCreateLocalized (months[i]);

    i = 0;
    XtSetArg (args[i], XmNvisibleItemCount, n); i++;
    XtSetArg (args[i], XmNitemCount, n); i++;
    XtSetArg (args[i], XmNitems, str_list); i++;
    list = XmCreateList (toplevel, "Hello", args, i);

    for (i = 0; i < n; i++)
        XmStringFree (str_list[i]);
    XtFree ((char *) str_list);

    XtManageChild (list);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}
```

MOTIF/X11 Tutorial tutorial-hello.c:

```
#include <Xm/PushB.h>

main (int argc, char *argv[])
{
    Widget          toplevel, button;
    XtAppContext     app;
    void            button_pushed(Widget, XtPointer, XtPointer);
    XmString         label;
    Arg              args[2];

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Hello", NULL, 0, &argc, argv,
                                     NULL, sessionShellWidgetClass, NULL);
    label = XmStringCreateLocalized ("Push here to say hello");
    XtSetArg(args[0], XmNlabelString, label);
    button = XmCreatePushButton (toplevel, "pushme", args, 1);
    XmStringFree (label);
    XtAddCallback (button, XmNactivateCallback, button_pushed, NULL);
    XtManageChild (button);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}

void button_pushed (Widget widget, XtPointer client_data, XtPointer call_data)
{
    printf ("Hello Yourself!\n");
}
```

MOTIF/X11 Tutorial tutorial-textbox-1-allcaps.c:

```
/* allcaps.c — demonstrate the XmNmodifyVerifyCallback for
** Text widgets by using one to convert all typed input to
** capital letters.
*/

#include <Xm/Text.h>
#include <Xm/LabelG.h>
#include <Xm/RowColumn.h>
#include <ctype.h>

void allcaps(Widget, XtPointer, XtPointer);

main (int argc, char *argv[])
{
    Widget          toplevel, text_w, rowcol, label_w;
    XtAppContext    app;
    Arg              args[2];

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Demos", NULL, 0, &argc, argv, NULL,
                                     sessionShellWidgetClass, NULL);

    XtSetArg (args[0], XmNorientation, XmHORIZONTAL);
    rowcol = XmCreateRowColumn (toplevel, "rowcol", args, 1);

    label_w = XmCreateLabelGadget (rowcol, "Enter Text:", NULL, 0);
    XtManageChild (label_w);
    text_w = XmCreateText (rowcol, "text_w", NULL, 0);
    XtManageChild (text_w);

    XtAddCallback (text_w, XmNmodifyVerifyCallback, allcaps, NULL);

    XtManageChild (rowcol);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}

/* allcaps() — convert inserted text to capital letters. */
void allcaps (Widget text_w, XtPointer client_data, XtPointer call_data)
{
    int len;
    XmTextVerifyCallbackStruct *cbs =
        (XmTextVerifyCallbackStruct *) call_data;

    if (cbs->text->ptr == NULL)
        return;
    /* convert all input to upper-case if necessary */
    for (len = 0; len < cbs->text->length; len++)
        if (islower (cbs->text->ptr[len]))
            cbs->text->ptr[len] = toupper (cbs->text->ptr[len]);
}
```

MOTIF/X11 Tutorial tutorial-bitmap-show.c:

```
/* show_pix.c — A minimal example of a MainWindow. Use a Label as the
 * workWindow to display a bitmap specified on the command line.
 */
#include <Xm/MainW.h>
#include <Xm/Label.h>
main (int argc, char *argv[])
{
    Widget          toplevel, main_w, label_w;
    XtAppContext    app_context;
    Pixmap          pixmap;
    Arg             al[4];
    Cardinal        ac = 0;

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app_context, "Demos", NULL, 0, &argc,
                                   argv, NULL, sessionShellWidgetClass, NULL);

    if (!argv[1]) {
        printf ("usage: %s bitmap-file\n", argv[0]);
        exit (1);
    }

    ac = 0;
    XtSetArg(al[ac], XmNscrollBarDisplayPolicy, XmAS_NEEDED); ac++;
    XtSetArg(al[ac], XmNscrollingPolicy, XmAUTOMATIC); ac++;
    main_w = XmCreateMainWindow (toplevel, "main_window", al, ac);

    /* Load bitmap given in argv[1] */
    pixmap = XmGetPixmap (XtScreen (toplevel), argv[1], BlackPixelOfScreen
                          (XtScreen (toplevel)), WhitePixelOfScreen (XtScreen (toplevel)));
    if (pixmap == XmUNSPECIFIED_PIXMAP) {
        printf ("can't create pixmap from %s\n", argv[1]);
        exit (1);
    }
    /* Now create label using pixmap */
    ac = 0;
    XtSetArg(al[ac], XmNlabelType, XmPIXMAP); ac++;
    XtSetArg(al[ac], XmNlabelPixmap, pixmap); ac++;
    label_w = XmCreateLabel (main_w, "label", al, ac);

    /* set the label as the "work area" of the main window */
    XtVaSetValues (main_w, XmNworkWindow, label_w, NULL);
    XtManageChild (label_w);
    XtManageChild (main_w);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app_context);
}
```

MOTIF/X11 Tutorial tutorial-menu-opensavequit.c:

```
/* file_menu.c — demonstrate how to create a menu bar and pulldown
** menu using the Motif creation routines.
*/

#include <Xm/RowColumn.h>
#include <Xm/MainW.h>
#include <Xm/CascadeB.h>
#include <Xm/SeparatorG.h>
#include <Xm/PushButtonG.h>

main (int argc, char *argv[])
{
    Widget          toplevel, main_w, menu_w, file_w, cascade_w, push_b, sep_w;
    XmString        label_str;
    XtAppContext    app;
    Arg             args[4];
    int             n;

    XtSetLanguageProc (NULL, NULL, NULL);
    toplevel = XtVaOpenApplication (&app, "Demos", NULL, 0, &argc, argv, NULL,
                                   sessionShellWidgetClass, NULL);

    n = 0;
    XtSetArg (args[n], XmNscrollingPolicy, XmAUTOMATIC); n++;
    main_w = XmCreateMainWindow (toplevel, "main_w", args, n);
    menu_w = XmCreateMenuBar (main_w, "MenuBar", NULL, 0);

    /* create the "File" Menu */
    file_w = XmCreatePulldownMenu (menu_w, "FilePullDown", NULL, 0);

    /* create the "File" button (attach Menu via XmNsubMenuId) */
    label_str = XmStringCreateLocalized ("File");

    n = 0;
    XtSetArg (args[n], XmNmnemonic, 'F'); n++;
    XtSetArg (args[n], XmNlabelString, label_str); n++;
    XtSetArg (args[n], XmNsubMenuId, file_w); n++;
    cascade_w = XmCreateCascadeButton (menu_w, "File", args, n);
    XtManageChild (cascade_w);
    XmStringFree (label_str);

    /* Now add the menu items */
    push_b = XmCreatePushButtonGadget (file_w, "Open", NULL, 0);
    XtManageChild (push_b);
    push_b = XmCreatePushButtonGadget (file_w, "Save", NULL, 0);
    XtManageChild (push_b);
    sep_w = XmCreateSeparatorGadget (file_w, "separator", NULL, 0);
    XtManageChild (sep_w);
    push_b = XmCreatePushButtonGadget (file_w, "Exit", NULL, 0);
    XtManageChild (push_b);

    XtManageChild (menu_w);
    XtManageChild (main_w);
    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}
```

MOTIF/X11 Tutorial tutorial-vroot-root-screensaver.c:

```
#include<stdlib.h>
#include<X11/Xlib.h>

#include "vroot.h"

main ()
{
    Display *dpy;
    Window root;
    GC g;

    /* open the display (connect to the X server) */
    dpy = XOpenDisplay (getenv ("DISPLAY"));

    /* get the root window */
    root = DefaultRootWindow (dpy);

    /* create a GC for drawing in the window */
    g = XCreateGC (dpy, root, 0, NULL);

    /* set foreground color */
    XSetForeground(dpy, g, WhitePixelOfScreen(DefaultScreenOfDisplay(dpy)) );

    /* draw something */
    while (1)
    {
        /* draw a square */
        XFillRectangle (dpy, root, g, random()%500, random()%500, 50, 40);

        /* once in a while, clear all */
        if( random()%500<1 )
            XClearWindow(dpy, root);

        /* flush changes and sleep */
        XFlush(dpy);
        usleep (10);
    }

    XCloseDisplay (dpy);
}
```


· **MOTIF/X11 Tutorial tutorial-run.c:**

```
gcc -o hello hello.c -lXm -lXt -lX11 -L/usr/X11R6/lib -lXm -lXt -lX11 -lm
```

·