Appendix G - New Features in Motif 2.3

This appendix provides a summary of the new features in Motif 2.3. It lists the new toolkit functions and widget classes in Motif 2.3, as well as any new resources added to existing widget classes. The last section describes the four major new features in Motif 2.3 and gives sample programs on how to use the features, as applicable. For more information on the functions and widgets, see the appropriate reference pages in Section 1, *Motif Functions and Macros,* and Section 2, *Motif and Xt Widget Classes*. For more information on the changes and new features in this version of Motif, see the *OpenMotif 2.3.0 Beta Release Notes*.

G.1 New Toolkit Functions

XmDropDownGetArrow ( )	Returns the “arrow” child from a DropDown.
XmDropDownGetChild ( )	Returns the specified child of a DropDown widget.
XmDropDownGetLabel ( )	Returns the “label” child from a DropDown.
XmDropDownGetList ( )	Returns the “list” child from a DropDown.
XmDropDownGetText ( )	Returns the “text” child from a DropDown.
XmDropDownGetValue ( )	Retrieve the value from a DropDown.
XmMultiListDeselectItems ( )	Clear the selection state of rows of MultiList by matching column entries to XmString.
XmMultiListDeselectRow ( )	Clears the selection state on a specific row of MultiList.
XmMultiListGetSelectedRowArray ( )	Returns NULL_terminated array of pointers to selected rows of a MultiList.
XmMultiListGetSelectedRows ( )	Returns a NULL-terminated array of pointers to data of selected rows of MultiList.
XmMultiListMakeRowVisible ( )	Scrolls the MultiList to make the specified row visible.
XmMultiListSelectAllItems ( )	Sets the selection state on all rows of MultiList.
XmMultiListSelectItems ( )	Sets the selection state by matching column entries to XmString.
XmMultiListSelectRow ( )	Sets the selection state on a specific row of MultiList.
XmMultiListToggleRow ( )	Toggles the selection state of a specified row.
XmMultiListUnselectAllItems ( )	Unselects all selected rows of a MultiList.
XmMultiListUnselectItem ( )	Unselects the specified row of a MultiList.

G.2 Obsolete Toolkit Functions

XmComboBox2GetArrow ( )
XmComboBox2GetChild ( )
XmComboBox2GetLabel ( )
XmComboBox2GetList ( )
XmComboBox2GetText ( )
XmComboBox2GetValue ( )
XmMultiListDeselectItems ( )
XmMultiListDeselectRow ( )
XmMultiListGetSelectedRowArray ( )
XmMultiListGetSelectedRows ( )
XmMultiListMakeRowVisible ( )
XmMultiListSelectAllItems ( )
XmMultiListSelectItems ( )
XmMultiListSelectRow ( )
XmMultiListToggleRow ( )
XmMultiListUnselectAllItems ( )
XmMultiListUnselectItem ( )

G.3 New Resources in Existing Widget Classes

XmColumn	XmNdefaultEntryLabelRenderTable
	XmNentryLabelRenderTable
XmDataField	XmNrenderTable
XmMultiList	XmNrenderTable
XmLabel	XmNpixmapPlacement
	XmNpixmapTextPadding
XmTabStack	XmNrenderTable

G.4 New Features in Motif 2.3

XmPIXMAP_AND_STRING

Motif 2.3 gives programmers the ability to show pixmaps and label strings simultaneously in Labels, Label gadgets, and inherited widgets. Labels of Unified Transfer Models can export all targets exported by labels of type XmPIXMAP and XmSTRING.

The following sample shows ways of using this new feature:

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

String fallback[] = {
    "*label.labelType:          XmPIXMAP_AND_STRING",
    "*label.labelPixmap:        toucan.png",
    "*label.labelString:        specified from resources",
    NULL
};

int
main(int argc, char *argv[]) {
  Widget            toplevel, rowcolumn;
  Arg               al[10];
  Cardinal          ac;
  XtAppContext      app;
  XmString          str;
  Pixmap            pixmap;
  XColor            color, exact;

  XtSetLanguageProc(NULL, NULL, NULL);
  toplevel = XtAppInitialize(&app, "test", NULL, 0, &argc, argv, fallback,
      NULL, 0);

  XtSetArg(al[ac], XmNorientation, XmVERTICAL); ac++;
  rowcolumn = XmCreateRowColumn(toplevel, "rowcolumn", al, ac);

  XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, NULL, 0);

  XtCreateManagedWidget("sep", xmSeparatorWidgetClass, rowcolumn, NULL, 0);

  /* Look for value of cyan color */
  XAllocNamedColor(XtDisplay(toplevel),
    DefaultColormap(XtDisplay(toplevel), DefaultScreen(XtDisplay(toplevel))),
    "cyan",
    &color,
    &exact);

  pixmap = XmGetPixmap(XtScreen(toplevel), "toucan.png",
        XmUNSPECIFIED_PIXEL, XmUNSPECIFIED_PIXEL);
  str = XmStringCreateLocalized("specified from code");

  XtSetArg(al[ac], XmNlabelType, XmPIXMAP_AND_STRING); ac++;
  XtSetArg(al[ac], XmNlabelPixmap, pixmap); ac++;
  XtSetArg(al[ac], XmNlabelString, str); ac++;
```

```
    XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, al, ac);

    XmStringFree(str);

    XtManageChild(rowcolumn);
    XtRealizeWidget(toplevel);

    XtAppMainLoop(app);
}
```

Advanced Image support

Added support of PNG and JPEG images in Motif 2.3 allows users to specify images of these types to pixmap resources, as well as load these images using Pixmap loading functions such as XmGetPixmap().

The following program demonstrates how to use the new feature:

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

String fallback[] = {
        "*label.labelType:          XmPIXMAP",
        "*label.labelPixmap:        toucan.png",
        NULL
};

int
main(int argc, char *argv[]) {
    Widget          toplevel, rowcolumn;
    Arg             al[10];
    Cardinal        ac;
    XtAppContext  app;
    Pixmap          pixmap;
    XColor          color, exact;

    XtSetLanguageProc(NULL, NULL, NULL);
    toplevel = XtAppInitialize(&app, "test", NULL, 0, &argc, argv, fallback,
            NULL, 0);
    XtSetArg(al[ac], XmNorientation, XmVERTICAL); ac++;
    rowcolumn = XmCreateRowColumn(toplevel, "rowcolumn", al, ac);

    XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, NULL, 0);

    XtCreateManagedWidget("sep", xmSeparatorWidgetClass, rowcolumn, NULL, 0);

    pixmap = XmGetPixmap(XtScreen(toplevel), "toucan.jpg",
        XmUNSPECIFIED_PIXEL, XmUNSPECIFIED_PIXEL);

    XtSetArg(al[ac], XmNlabelType, XmPIXMAP); ac++;
    XtSetArg(al[ac], XmNlabelPixmap, pixmap); ac++;
    XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, al, ac);

    XtManageChild(rowcolumn);
    XtRealizeWidget(toplevel);

    XtAppMainLoop(app);
}
```

XFT support

The support of the XFT library introduced in OpenMotif 2.3 offers user's the ability to use client-side anti-aliased fonts from Motif applications. This feature is available via the new Rendition Font Type, named XmFONT_IS_XFT.

The following sample shows how to specify the definition of the default RenderTable with the new Font Type:

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

String fallback[] = {
        "*label.renderTable:                three",
        "*three.fontType:                   FONT_IS_XFT",
        "*three.fontName:                   Times",

        "*label.labelType:         XmPIXMAP_AND_STRING",
        "*label.labelPixmap:       toucan.png",
        "*label.labelString:       specified from resources",
        NULL
};


int
main(int argc, char *argv[]) {
    Widget           toplevel, rowcolumn;
    Arg              al[10];
    Cardinal         ac;
    XtAppContext     app;
    XmString         str;
    Pixmap           pixmap;
    XColor           color, exact;

    XtSetLanguageProc(NULL, NULL, NULL);
    toplevel = XtAppInitialize(&app, "test", NULL, 0, &argc, argv, fallback,
            NULL, 0);

    XtSetArg(al[ac], XmNorientation, XmVERTICAL); ac++;
    rowcolumn = XmCreateRowColumn(toplevel, "rowcolumn", al, ac);

    XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, NULL, 0);

    XtCreateManagedWidget("sep", xmSeparatorWidgetClass, rowcolumn, NULL, 0);

    /* Look for value of cyan color */
    XAllocNamedColor(XtDisplay(toplevel),
      DefaultColormap(XtDisplay(toplevel), DefaultScreen(XtDisplay(toplevel))),
      "cyan",
      &color,
      &exact);

    pixmap = XmGetPixmap(XtScreen(toplevel), "toucan.png",
              XmUNSPECIFIED_PIXEL, XmUNSPECIFIED_PIXEL);
    str = XmStringCreateLocalized("specified from code");

    XtSetArg(al[ac], XmNlabelType, XmPIXMAP_AND_STRING); ac++;
    XtSetArg(al[ac], XmNlabelPixmap, pixmap); ac++;
    XtSetArg(al[ac], XmNlabelString, str); ac++;
    XtCreateManagedWidget("label", xmLabelWidgetClass, rowcolumn, al, ac);

    XmStringFree(str);

    XtManageChild(rowcolumn);
    XtRealizeWidget(toplevel);

    XtAppMainLoop(app);
  }
```

UTF-8 support

Added support of UTF-8 allows for displaying and editing UTF-8 encoded strings in all widgets. Interoperability with these strings is also supported via the UTF8_STRING atom proposed by X.Org.

This feature is active in an appropriate environment (i.e. in the UTF-8 locale) without any effort from the application programmer.