

微机接口与应用系列丛书

Motif 程序员指南

赵 生 审校

学苑出版社

1993.

目 录

第一章 OSF / Motif 工具箱简介	(1)
1.1 OSF / Motif 工具箱和 X Window 系统	(1)
1.2 Widget 类和层次结构	(1)
1.3 编译示例程序	(4)
第二章 widget, gadgets 和便利函数	(5)
2.1 widget	(5)
2.2 gadgets	(17)
2.3 便利函数	(19)
第三章 用 Motif Widgets 编程	(21)
3.1 引言	(21)
3.2 xmbutton 程序介绍	(21)
3.3 包含头文件	(25)
3.4 初始化 XtIntrinsics	(25)
3.5 建立 Widgets 的变量表	(27)
3.6 增加回调过程	(28)
3.7 建立 widget	(30)
3.8 使 Widget 可见	(31)
3.9 库链接	(32)
3.10 建立缺省文件	(32)
3.11 使用颜色	(34)
3.12 高级编程技巧	(36)
3.13 一个高级示例程序	(38)
第四章 Shell Widgets	(50)
4.1 Shell Widgets 介绍	(50)
4.2 Shell Widget 的外观	(51)
第五章 对话 widget 及函数	(52)
5.1 对话 widget 及菜单	(52)
5.2 对话 widget 列表	(52)
5.3 便利的对话	(53)
5.4 对话与便利函数的使用	(54)

第六章 菜单	(67)
6.1 Motif 菜单系统综述	(67)
6.2 建立弹出式菜单系统	(72)
6.3 建立一个下拉式菜单系统	(77)
6.4 建立子菜单	(83)
6.5 建立选项菜单系统	(87)
6.6 选中一个菜单光标	(92)
6.7 不使用便利函数来建立菜单	(93)
第七章 专用 Widgets	(100)
7.1 List Widget	(100)
7.2 RowColumn widget	(109)
7.3 正文 widget	(113)
第八章 补充功能.....	(122)
8.1 复合串	(122)
8.2 剪辑和粘贴函数	(134)
8.3 动态资源缺省	(148)
8.4 定位	(148)
8.5 象素图高速缓存函数	(150)
8.6 与分辨率无关的机制	(152)
8.7 与 Motif 窗口管理器的交互操作	(154)
8.8 Motif 版本号	(159)
8.9 Motif 窗口管理器的存在	(159)
第九章 键盘接口.....	(160)
9.1 键盘焦点的模式	(160)
9.2 把 Widgets 组合成制表组	(161)
9.3 制表组内和制表组之间的遍历	(161)
9.4 Widget 的键盘输入处理	(162)
第十章 OSF / Motif 窗口管理器的介绍	(163)
10.1 约定	(163)
10.2 窗口管理程序的注意事项	(164)
第十一章 窗口管理原理.....	(165)
11.1 用户输入的获得	(165)
11.2 焦点策略的使用: 键盘输入和颜色表	(167)

11.3	窗口分类	(168)
11.4	MWM 窗口边框各部分的了解	(170)
11.5	放置窗口的和设置大小	(176)
11.6	标像简介	(178)
第十二章	MWM 和用户之间的通讯: MWM 的特性	(179)
12.1	MWM 接口标准纲要	(179)
12.2	客户之间通讯的约定	(179)
12.3	MWM 专有信息	(185)
第十三章	使用 MWM 管理窗口	(190)
13.1	启动 MWM	(190)
13.2	使用资源配置 MWM	(191)
13.3	管理屏幕	(193)
13.4	活动窗口焦点策略的设置	(193)
13.5	管理客户程序	(195)
13.6	管理窗口框架和衬边的外观和功能	(195)
13.7	设置窗口管理程序的不同字体	(196)
13.8	给窗口框架着色	(196)
13.9	制作窗口框架的像素图	(199)
13.10	指定客户程序窗口的衬边	(200)
13.11	使用无框架或精简窗口	(202)
13.12	控制窗口的大小和位置	(204)
第十四章	管理菜单、鼠标器按钮和键盘束定	(207)
14.1	引言	(207)
14.2	MWM 的资源描述文件	(207)
14.3	修改菜单及隐含的菜单	(208)
14.4	制做新菜单	(209)
14.5	改变与窗口菜单相关的菜单	(215)
14.6	鼠标器按钮的束定	(216)
14.7	键盘的束定	(221)
第十五章	使用标像	(223)
15.1	研究标像的结构	(223)
15.2	操纵标像	(224)
15.3	控制标像的外观和功能	(227)
15.4	给标像着色	(228)
15.5	使用标像框保存标像	(229)

第十六章 快速参考表	(232)
16.1 资源的语法	(232)
16.2 资源表	(232)
16.3 窗口的装饰	(234)
16.3 MWM 函数	(238)
第十七章 用户接口语言的介绍	(240)
17.1 UIL 和 MRM 概述	(240)
17.2 使用 UIL 和 MRM 的优点	(240)
17.3 UIL 的特征	(242)
第十八章 UIL 语言的语法	(244)
18.1 字符集	(244)
18.2 命名	(245)
18.4 文字	(249)
18.5 值产生函数	(256)
18.6 any 数据类型	(260)
18.7 编译时值的表达式	(268)
第十九章 UIL 模块的结构	(272)
19.1 UIL 模块的结构	(272)
19.2 值和对象的引用范围	(275)
19.3 值段的结构	(275)
19.4 过程段的结构	(277)
19.5 列表部分的结构	(278)
19.6 对象部分的结构	(282)
19.7 标识符部分的结构	(285)
19.8 包含性指令	(287)
19.9 约束参数的定义	(287)
19.10 widget ID 的象征性引用	(288)
第二十章 使用 UIL 编译程序	(289)
20.1 UIL 编译程序的执行	(289)
20.2 字符串文字的缺省字符集	(293)
20.3 解释 UIL 编译程序产生的诊断信息	(294)
20.4 读 UIL 编译程序输出的列表	(294)
第二十一章 Motif 资源管理函数	(297)

21.1	建立存储和数据结构	(297)
21.2	获取 UID 数据文件 ID	(297)
21.3	关闭一个 MRM 搜索层次	(300)
21.4	登记 MRM 信息和回调函数	(300)
21.5	获取 widget	(303)
21.6	获取直接量	(306)
第二十二章 用 UIL 和 MRM 建立		(309)
22.1	用 UIL 来描述一个用户界面	(309)
22.2	用 MRM 实时创建一个用户界面	(325)
22.3	使用 UIL 和 MRM 来定制 MOtIF 界面	(339)
22.4	在大型项目中使用 UIL	(339)
22.5	在 UIL 中使用用户定义的 widget	(340)
第二十三章 widget 元语言		(348)
23.1	运用 WML	(348)
23.2	修改 WML 文件	(351)
附录 A 约束参数		(358)
附录 B UIL 内部表		(359)
B.1	简介	(359)
附录 C UIL 参数		(396)
附录 D UIL 诊断信息		(403)
词汇表		(413)

第一章 OSF / Motif 工具箱简介

1.1 OSF / Motif 工具箱和 X Window 系统

OSF / Motif Widget 集是一个基于 Xt Intrinsics 的, 可以方便快速地访问 X Window 系统低层的函数和过程的集合。由图 1-1 可以看出 Motif Widget 系统位于 Xt Intrinsics 上层, 而 Xt Intrinsics 又位于 X Window 系统上层, 这样, Motif Widget 系统便扩充了抽象的 X Window 系统。

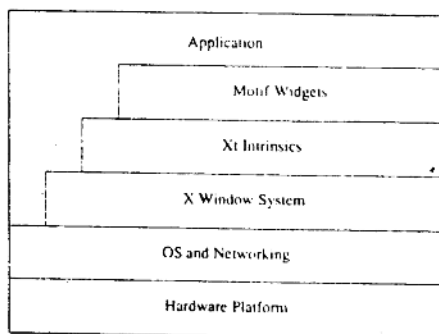


图 1-1 用户界面开发模型

Motif Widget 系统支持独立开发新的或扩充 widget。Motif Widget 系统由许多不同的 widget(工具)组成, 各 widget 的单独或组合使用有助于开发复杂的应用程序。使用 Motif Widget 可以提高编写应用程序的速度并减少源代码行数; 不过, 用 Motif Widget 编写的应用程序所需内存使用它的同样程序大。

本指南逐个介绍各 widget, 并说明如何在应用程序中使用它们。

1.2 Widget 类和层次结构

每一个 widget 都是动态分配的, 并包含状态信息。每一个 widget 都属于一个类。类具有静态分配及初始化的结构, 并包含类。图 1-2 展示了基本 widget 类。

基本类是核心(Core)类, 它包含所有其它类要继承的资源。核心类之下有两个类, 组合类和基元类。基元类底下不再有元类, 而组合类有两个基元类: Constraint 类和 Shell 类。低层类部分或全部继承高层类的资源。例如, Manager 类的 widget 可以部分或全部继承 Constraint、Composite 和核心(Core)类的资源。要了解各 widget 具有的资源, 可查阅《OSF / Motif 程序员参考手册》中的相应手册页。

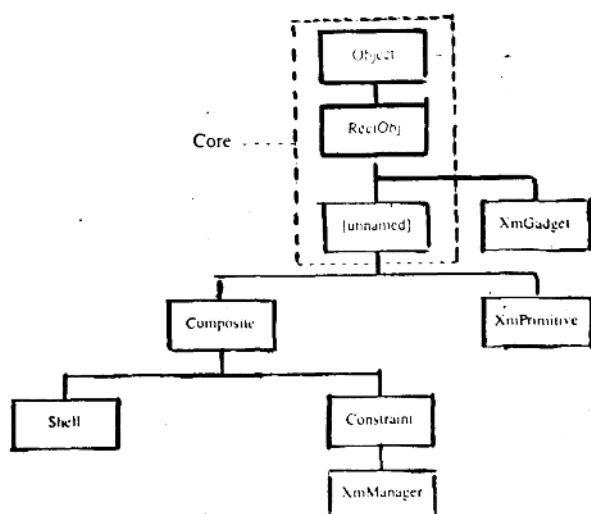


图 1-2 基本 Widget 类层次结构

本节给出了一些层次结构图，旨在帮助读者了解各 widget 之间的关系。图 1-2 所示为最高层 widget 类。从中可以看出核心类由 Object、RectObj 和一无名类组成。核心类是其它所有类的基本类。

图 1-3 展示了基元类的子类。

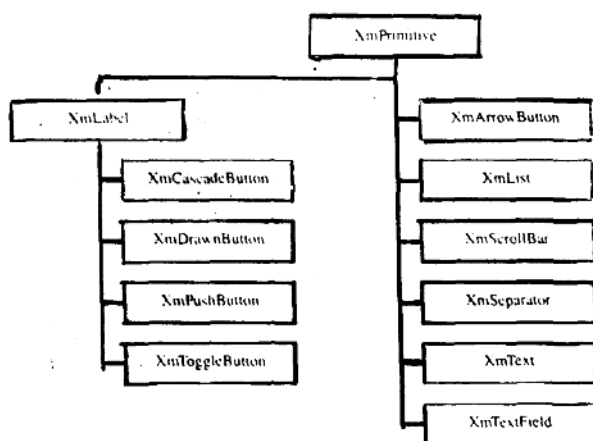


图 1-3 基元类 Widget

图 1-4 展示了 Shell 类的子类。

图 1-5 展示了各 Manager 类 widget。注意，由图 1-2 可知 Manager 是组合类的约束类的子类。

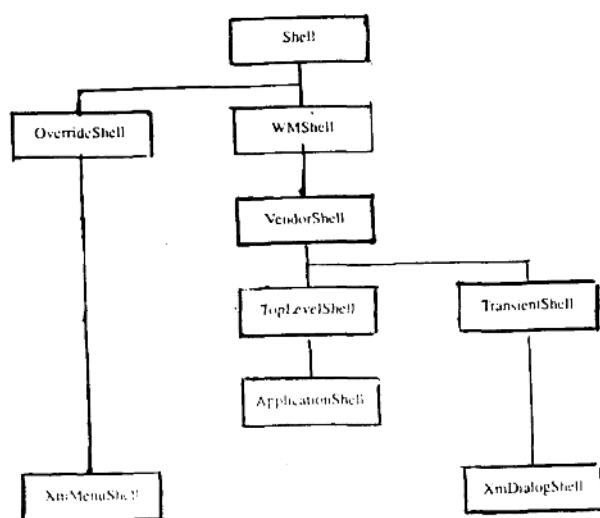


图 1-4 Shell Widget

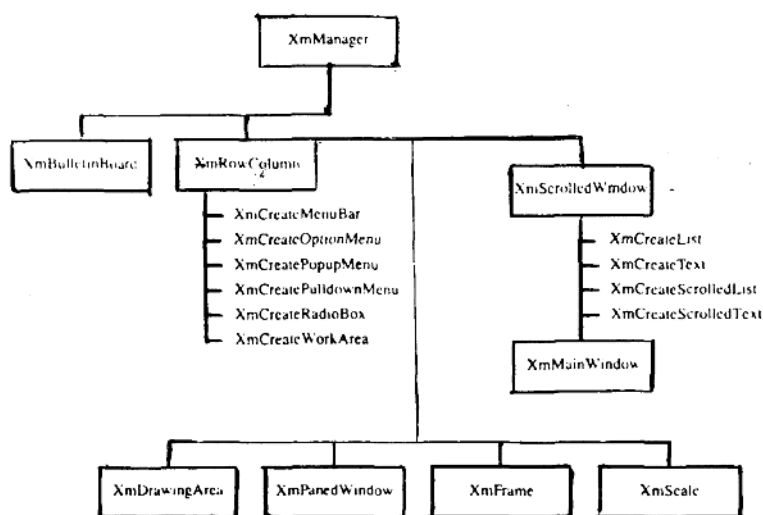


图 1-5 Manager Widget

图 1-6 展示了各对话(Dialog)widget，它们是 Manager 的子类，注意，所有对话 widget 都是布告牌(Bulletin Board)的子类。另外，上图还列出了便利类数。第五章“对话 Widget 和函数”将详细解释这些函数。

图 1-7 展示了作为 Motif 工具箱的一部分的 gadget。

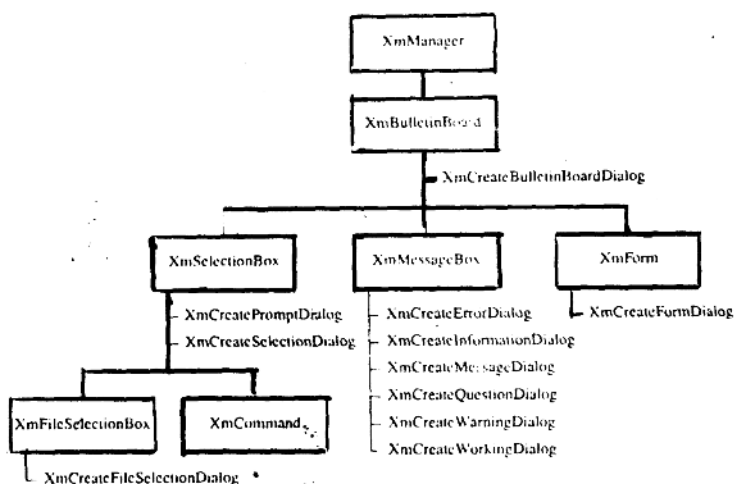


图 1-6 对话(Dialog) Widget

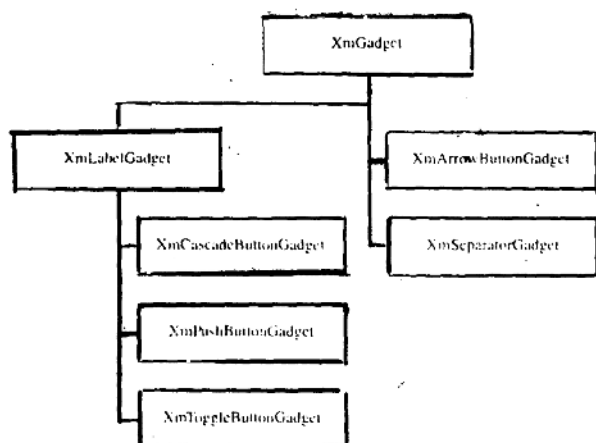


图 1-7 Gadget

1.3 编译示例程序

本指南给出了大量示例程序，多数程序的原代码可在 `/usr/lib/X11/examples` 目录中找到。该目录中有一个 `Makefile`，可用来编译和链接这些程序。可按以下步骤来编译和链接程序。

1. 将 `/usr/lib/X11/example` 目录中的源程序和 `Makefile` 拷入用户的工作目录。不要要在 `/usr/lib/X11/example` 目录中编译程序。
2. 运行下面命令编译程序：

make programname

3. 如果源代码中注释有缺省文件，则在运行程序前将缺省文件移入 `/usr/lib/X11/app_defaults` 目录。

第二章 widget,gadgets 和便利函数

Motif 有各种各样的 widget 和 gadgets. widget 可以单独地或与其它 widget 的一起来完成某项特定的任务. 便利函数为某一任务建立特定的 widget 或一类 widget. 本章解释 widget, gadgets 和便利函数.

2.1 widget

应用 widget 可使复杂应用程序的建立变得更加容易,更加迅速. 有些 widget 显示信息,而其它一些仅仅是其它 widget 的容器. 某些 widget 仅限于显示信息而对键盘或鼠标器 (mouse) 输入不作任何处理,其它的 widget 则按照输入改变它们的显示或按照要求激活函数. 用户可以把一个 widget 的某些方面用户化,如字型,前景色和背景色,边界的宽度、颜色和大小.

widget 按照功能被分成若干类. 从逻辑上讲,一个 widget 类由属于这个类的所有 widget 的过程或数据构成,这些过程和数据可以被子类所继承. 从物理上讲,一个 widget 类是一个指向结构的指针,这个结构的内容对于这个 widget 类的所有 widget 是不变的.

某个 widget 类的示例是包含属于这个类的某个 widget 的值及过程的数据结构. 也有包含适用于这个类的所有 widget 的值和过程的类结构. 一个 widget 示例可用 XmCreatewidgetname, XmCreatewidget 或 XtCreateMangeWidget 来分配空间和初始化. 请参见第三章《使用 Motif widget 编程》,这一章有建立 widget 的实例.

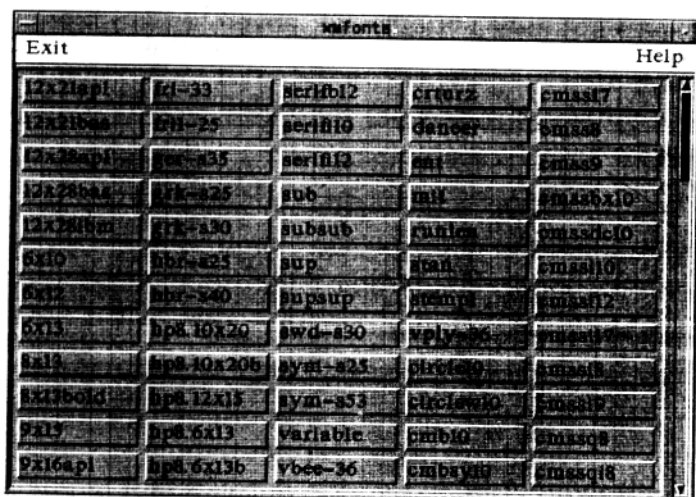


图 2-1 Widget 实例

这一节是所有 widget 的概述。在“OSF/Motif 程序员参考手册”里有每个 widget 的详细描述。图 2-1 表明如何在一个应用中使用 widget。

在图 2-1 中列出了几种类型的 widget。大窗口是 Main Window widget,其中有菜单条和按钮菜单,带有许多按钮 gadgets 的 RowColumn widget 和垂直滚动条。产生这个窗口的程序调用 Xmfonts。这个程序在第三章予以说明。

这一节把 widget 分成五类,如下表所示:

表 2-1 widget 的分类

Class Name	Widget Class
Shell Widgets	
XmDialogShell	xmDialogShellWidgetClass
XmMenuShell	xmMenuShellWidgetClass
VendorShell	vendorShellWidgetClass
Display Widgets	
Core	widgetClass
XmPrimitive	xmPrimitiveWidgetClass
XmArrowButton	xmArrowButtonWidgetClass
XmDrawnButton	xmDrawnButtonWidgetClass
XmLabel	xmLabelWidgetClass
XmList	xmListWidgetClass
XmPushButton	xmPushButtonWidgetClass
XmScrollBar	xmScrollBarWidgetClass
XmSeparator	xmSeparatorWidgetClass
XmText	xmTextWidgetClass
XmTextField	xmTextFieldWidgetClass
XmToggleButton	xmToggleButtonWidgetClass
Container Widgets	
XmManager	xmManagerWidgetClass
XmDrawingArea	xmDrawingAreaWidgetClass
XmFrame	xmFrameWidgetClass
XmMainWindow	xmMainWindowWidgetClass
XmPanedWindow	xmPanedWindowWidgetClass
XmRowColumn	xmRowColumnWidgetClass
XmScale	xmScaleWidgetClass
XmScrolledWindow	xmScrolledWindowWidgetClass
Dialog Widgets	
XmBulletinBoard	xmBulletinBoardWidgetClass
XmCommand	xmCommandWidgetClass
XmFileSelectionBox	xmFileSelectionBoxWidgetClass
XmForm	xmFormWidgetClass
XmMessageBox	xmMessageBoxWidgetClass
XmSelectionBox	xmSelectionBoxWidgetClass
Menu Widgets	
XmCascadeButton	xmCascadeButtonWidgetClass

2.1.1 shell widget

shell widget 是顶层的 widget。它提供具有窗口管理器的必要接口,不同的 shell widget 类用于不同的顶层 widget。XtIntrinsics 提供一些基本的 shell widget,其它的

shell widget 由 Motif 工具箱提供。XtIntrinsics 提供下列 shell 类:

Shell

它是所有 shell widget 的基类。它是 Composite 的子类并为其它的外壳提供资源。

OverrideShell

该类用于 shell 窗口, shell 窗口“旁路”掉窗口管理器。它是 Shell 的子类。

WMShell

该类包含公共窗口管理器协议所必须的资源。它是 Shell 的子类。

VendorShell

该类包含特定于售主的窗口管理器所用的资源,它是 WMShell 的子类。

TransientShell

该类用于能被窗口管理器所操纵但不能标像化的 Shell。它是 VendorShell 的子类。

TopLevelShell

这个类用于正常的顶层窗口,它是 VendorShell 的子类。

ApplicationShell

该类用于应用程序的顶层窗口,它是 TopLevelShell 的子类。

类 Shell, WMShell 和 VendorShell 是内部的,不能被示例化。

Motif 工具箱提供下列 widget :

XmDialogShell (XmDialogShellWidgetClass)

DialogShell widget 类是 TransientShell 的子类。这个类的示例用作与其它顶层窗口相联系形式或的非型式包对话框 widget。DialogShell 为第二类顶层窗口(如 Dialogs)提供与 ICCM(Inter-client Communications Conventions Manual) 一致的与 Motif 窗口管理器的通讯。关于这个 widget 是如何用的更多信息,请参见第五章《对话 widget》。

XmMenuShell (XmMenuShellWidgetClass)

MenuShell Widget 类是 OverrideShell 的子类。关于菜单 widget 的具体规定和菜单界面的内容,参看第六章《菜单》。

VendorShell (VendorShellWidgetClass)

VendorShell widget 类是 WMShell 的子类。它提供由窗口管理器可见 shell 所需要的公共状态信息和服务设施。更多的信息请看第四章“shell widget”。

2.1.2 显示 widget

注:

每个类的资源清单可在《OSF / Motif 程序员参考手册》里找到。

Motif 提供下列显示 widget:

Core (widgetClass)

核心 (Core) 类是支持其它 widget 类的支持高层类。它提供所有 widget 需要的公共资源,包括 x 和 y 位置,高度,宽度,窗口边界宽度等等。

XmPrimitive (xmPrimitiveWidgetClass)

XmPrimitive 类也是其它 widget 类的支持高层类。它提供诸如边框的绘制和高亮度显示,遍历激活及关闭等资源。

XmArrowButton (xmArrowButtonWidgetClass)

ArrowButtonWidget 由方向箭头组成。当选取箭头按钮时,在被压下的箭头按钮处显示出阴影。当退出选择时,阴影将消失以给出箭头键被释放的外观。箭头按钮和推压式按钮有同样的功能。下图列出了在一个 RowColumn Widget 里的四个箭头按钮。

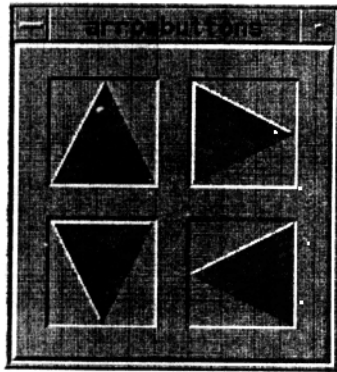


图 2-2 箭头按钮

箭头方向由设定 XmNarrowDirection 资源的值来确定,图 2-2 中 ArrowButtons 之间的间隔由把 RowColumn 资源 XmNmarginWidth, XmNmarginHeight 和 XmNspacing 设为 20 得到。

XmDrawnButton (xmDrawnWidgetClass)

DrawButtonWidget 由阴影边框包围的空 widget 窗口组成。它为应用程序开发人员提供控制按钮输入语义的图形区域。

widget 暴露和重定大小的调用类型允许应用程序重新确定图形的位置和重画图形。如果 DrawButton Widget 用醒目显示并且阴影加厚,应用程序要注意不要在这个区域内画图。这可通过在 widget 里使用剪取矩形建立图形的图形上下文来实现。矩形裁剪应当考虑到 widget 的醒目厚度和阴影的大小。

XmLabel (xmLabelWidgetClass)

Label Widget 由正文或图形组成。它可以被初始化,但它也可用做按钮 widget 的高层类。一个 Label 正文是一复合字符串,可以多方向、多行、多字型或它们的任意组合。Label 被认为是静态的,因为它不象 widget 上的求助按钮那样接受按钮或键盘输入。求助回调 (help callback) 是为 Label 定义的唯一回调。

XmList (xmListWidgetClass)

List Widget允许用户从项目表中选取某一项。应用程序定义一组复合字符串，每一个字符串作为表中的一项。用户可以设定表中可见项的数目。用户也可以设定表有滚动条，以便用户能够滚动地看到整个表。用户可以把光标移到要选取的项上然后按下鼠标器按钮或按键来选择某一项。被选取的项用反转色显示。下图展示出了 List Widget。

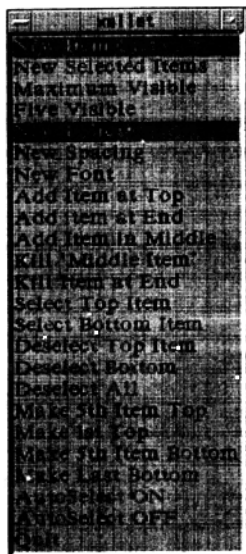


图 2-3 List Widget

XmPushButton (xmPushButtonWidgetClass)

PushButton Widget由边框阴影色围的正文标志或象素图构成。用户可以把鼠标器光标移到要选取的按钮上并按下鼠标器的 1 按钮来选取它。当鼠标器的按钮按下时，Widget 和阴影的颜色将反转，显示用户刚按下的控制按钮。当松开鼠标器按钮时，颜色回到原来的模式，显示被释放的控制按钮。控制按钮用于执行动作，如运行、撤销、停止等诸如此类的操作。

XmScrollBar (xmScrollBarWidgetClass)

ScrollBarWidget允许用户查看一次不能全部看到的数据。滚动条和含有要查看的数据的 widget 相结合。当用户和滚动条交互时，数据上下滚动。可见的部分数据叫做工作区域。一个滚动条由在小矩形的每一端里的两个相反方向的箭头构成。矩形叫做滚动域 (scroll region)。的一个更小的矩形在滚动域里，这个小矩形叫做滑块 (slider)。滑块的颜色通常和滚动域的颜色相反。滑块的大小和滚动域大小的比值与可见数据和全部数据的比值关系相对应。例如，如果在工作区域里 10%的数据可见，滑块则占滚动域的 10%。

用户可以将滚动条水平，垂直或二者同时放置。水平滚动条放在工作区域的底边位置。而垂直滚动条则放在右边。ScrollBarWidget 如下图所示。

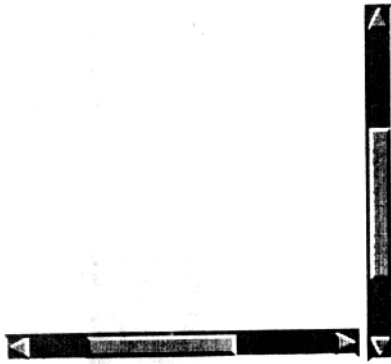


图 2-4 ScrollBars

XmSeparator (xmSeparatorWidgetClass)

分隔符 (Separator) 是基元Widget。它用作显示的各项之间的分隔标志。除了水平方位和竖直方位以外, 还提供了其它几种画线方式。

在分隔符里的画线在水平时自动画在Widget高度的中心位置, 在竖直时则画在Widget 宽度方向的中心位置。

XmNO-LINE的XmNseparatorType可使应用程序员需要其它的线画类型时从前一个类型退出。可用 XmNbackgroundPixmap 参数类型建立参数表的办法建立一个 widget 高度的位图作为背景位图, 其中 XmNbackgroundPixmap 在 Core 中定义。只要 widget 被重画, 包含适当分隔符的背景也显示出来。

XmText (xmTextWidgetClass)

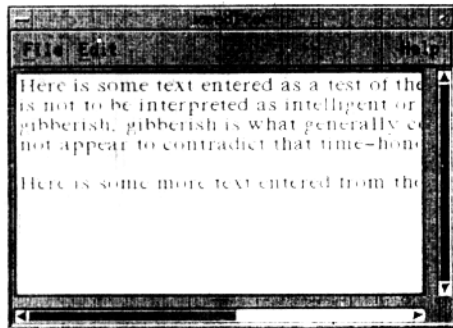


图 2-5 Text Widget

正文 (text) widget提供一个具有可定制的用户或程序员接口的、单行或多行的正文编辑器。它可作为单行字符串项、带验证过程的表格项, 可用于多页文档显示和全屏幕编辑。图 2-5 展示了 Text Widget。

XmTextField (xmTextWidgetClass)

TextField Widget和TextWidget相似, 但仅限于单行正文编辑, 并作了优化。和 Text Widget 一起, 用户可以定制用户和程序员接口。

XmToggleButton (xmToggleButtonWidgetClass)

这个Widget由正文或图形按钮构成。在正文或图形按钮的表面的左边有一指示

器（正方形或钻石形的方块）。选择切换按钮的方法是：把鼠标器的光标移到矩形内并按下按钮 1。然后指示器用选择的颜色进行填充，表明那个切换按钮被选取。切换按钮用于在一个应用中设定永久不变的数据。下图展示了 Toggle Buttons。

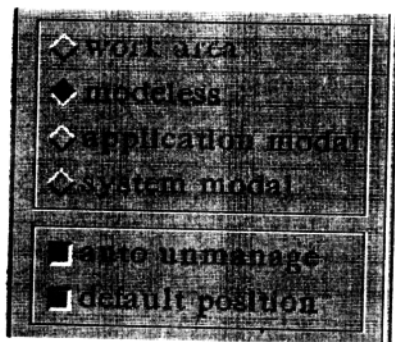


图 2-6 Toggle Buttons

2.1.3 Container Widget

Container Widget 是 Composite widget，为应用程序提供一般的布局功能。因为它们是 Compositewidget，所以它们可以有孩子。所有的容器 Widget (Container Widget) 都由类 Core, Composite, Constraint 和 XmManager 建立。

Motif 提供下列 Container widget:

XmManager (xmManagerWidgetClass)

XmManager 类是一个 Motif Widget 基元类，没有示例。它的唯一用途是作为支持其它 Widget 的高层类。它支持可见资源，图形上下文和图形所需要的遍历资源的遍历机制。XmManager 由类 Core, Composite 和 Constraint 建立。

XmDrawingArea (xmDrawingAreaWidgetClass)

DrawingAreaWidget 是一个空 Widget，容易适于各种用途。除了执行调用返回 (callback) 外不能绘图或定义其它功能。当需要在图区绘图或当 Widget 从键盘或鼠标器接收到输入时调用返回通知应用程序。应用程序负责定义需要响应 DrawingArea 调用返回的响应。DrawingArea 是一个复合 Widget 并且是 XmManager 的子类。它支持多个 widget 或 gadget 孩子的最小几何管理。

XmFrame (xmFrameWidgetClass)

XmFrame Widget 是一个使用 XmFrame Widget 所画的边框包围单个子 widget 的管理器。当它和其它的 XmManager 和 XmPrimitive widget 有相同的边框外观时，它常用来包含其它管理器。

XmMainWindow (xmMainWindowWidgetClass)

XmMainWindow Widget 为应用的主窗口提供标准的布局。这个布局包括菜单条，命令窗口，工作区域和滚动条。它们中的一部分或全部是任选的。在主窗口中的工作区域和滚动条和它们在 ScrolledsWindow Widget 中的作用是完全一样。