Motif Motif

计算机实用技术系列丛书(二)

Motif 实用编程大全

龚雨 曾田 等编写

学苑出版社

计算机实用技术系列丛书(二

Motif 实用编程大全

龚 雨 曾 田等 编写
刘映杰 李 木 审校

学苑出版社
1994.

(京)新登字 151 号

内 容 提 要

本书全面介绍 Motif 工具库的编程技术。在编写过程中，既照顾初学者内容尽可能简单，同时又为高级程序员的编程技术提供帮助。其中很多地方用到了 X—Window 的特征。每章内容由浅入深，针对特定的 Motif 接口文件，先介绍其功能，然后讲述创建和操作该对象的具体方法，中间的小节介绍该对象的资源及其配置内容等，每章均有小结并配有习题，补充正文中未能涉及的有关内容。

计算机实用技术系列丛书(二)

Motif 实用编程大全

编　　写：龚　雨　曾　田等
审　　校：刘映杰　李　木
责任编辑：甄国宪
出版发行：学苑出版社　　**邮政编码**：100036
社　　址：北京市海淀区万寿路西街 11 号
印　　刷：双青印刷厂
开　　本：787×1092　1/16
印　　张：49.5　　**字 数**：1173 千字
印　　数：1～3000 册
版　　次：1994 年 3 月北京第 1 版第 1 次
ISBN7－5077－0760－1/TP・7
本册定价：45.00 元

前　言

本书讲述如何使用 Open Software Foundtion（开放软件基金会，简称 OSF）的 Motif 工具库编写应用程序。Motif 是建立在 X Toolkit Intrinsics（Xt）的基础之上的。Xt 是一种标准的机制，因此成为许多为 X 窗口系统编写的工具库赖以建立的基础。Xt 提供了一个用于用户接口的对象库。这些对象称为 widget 和 gadget，它们为创建和操作 X 窗口，颜色表（colormap），事件及其它一些显示属性提供了一个方便的接口。简而言之，widget 是程序员用于构造一个完整的应用程序所必需的基本单元。

Xt 提供的 widget 本质上是通用的，而且没有实施任何具体的用户接口方案，因此正好可用类似 Motif 的工具库来实现。Motif 提供了一整套 widget，用来实现满足 Motif 应用环境规范的应用程序。

本书是一本全面的有关 Motif 工具库的编程指南。尽管 OSF / Motif 建立在 Xt 的基础之上，但是本书的重点不在 Xt，而是 Motif 本身。关于 Xt 的详细说明在另一本名为《X Toolkit Intrinsics 编程手册》的书中可以找到。本书中的参考很多是出自于该书。学完本书就可以有效地用 Motif 编程，但为了使应用程序更加完善，最好对 Xt 和 Xlib 也有深入的了解。

本节具体讨论 Motif 是什么、它如何能帮助程序员设计用户接口。简单地说，Motif 是一套准则（guideline），规定了图形计算机用户接口的外观（look）和"感觉"（feel）。所谓外观即规定应用程序在屏幕上的显示方式和布局，而"感觉"则规定用户应该如何与应用程序打交道。

本书的内容一方面考虑到初学者使内容尽可能简单，另一方面也可作为高级程序员的手册，因为其中很多地方用到了 X 窗口系统的很多特征。每章的内容由浅入深，针对某一个特定的 Motif 接口元件，首先简单介绍其功能，然后讲述创建和操作该对象的具体方法，中间的小节讲述该对象的资源及其它可配置的内容，包括便利函数及相关的事项。每章都有小结对本章的内容简要地回顾。有些章还配有习题，补充说明了正文中未能涉及的有关内容。

本书有一个总体的编排，即前面的几章是对 Motif 的一般性介绍，后面的章就逐一介绍每个 widget，先介绍大的，高级的管理器 widget，然后是原始的 widget 和 gadget。书后有若干附录供读者参考。

本书在编写过程中得到了何雪松　刘凯、张祥、魏彬、刘一兵、罗莉、王强、肖帆、钟瑞琪等同志的大力支持，在此表示感谢。

由于时间仓促，编者水平有限，书中难免有不当之处，望读者批评指正。

编者

目　录

上　册

第一章　引言

1.1　基本的用户接口概念

不论读者是软件的设计者还是编写应用程序的程序员，下面的一些基本原则都是适用的:

1. 所有应用程序的接口设计必须是一致的（consistent）.
2. 一些基本的或常用的功能的用法应该简便易记，因为用户往往凭记忆进行某些操作.
3. 用户（特别是初学者）一般不对其应用程序进行定做或修改. 如果确实要定做或修改，则要求方法尽可能简单.

用过 UNIX 软件的读者都知道 UNIX 应用程序是极其灵活的，它们给用户提供了很多选项. 程序的灵活性固然是一件好事，但如果灵活性太大，要求用户进行一些定做或设置之后才能使用程序，则可能带来相反的效果，使用户觉得程序的用法很繁琐. 当然，目前使用 UNIX 和 X 的人一般都是对软件环境及用法比较熟悉的技术人员，因此完成一些定做的事项并不困难. 但考虑到软件普及后一些非专业人员可能也要用到 UNIX 和 X，此时以上三个基本原则就非常重要了.

1.2　设计用户接口

设计一个有效的用户接口仅依赖于 Motif 或其它 GUI 的技术规范说明是远远不够的. 即使 Motif 工具库说明了如何创建和使用其接口元件，还必须要求程序员有效地利用这些元件合理地设计出用户接口，才能有助于提高用户的使用效率（Productivity）. 在应用程序的设计中，程序员面临的最困难的问题是一方面要考虑到初学者使接口尽可能简单，另一方面要考虑到熟练的用户使接口功能尽可能完善.

设计用户接口并没有捷径，好的用户接口往往是总结长期的使用经验不断改善、不断积累而成的. 这并非意味着接口设计无规律可循. 下面的经验法则就是长期积累的经验之谈:

- 使接口尽可能简单
- 使接口与真实的物体或概念对应
- 如果没有对应的真实物体或概念，可以自行设计
- 接口的简单不应以减少其功能为代价

保持接口尽可能简单依赖于多种因素，其中最基本的一点是直观（intuition）. 因为

用户使用应用程序的目的是为了解决一个现实中遇到的实际问题，如果应用能与现实中的物体或概念直接对应，就会极大地方便用户的使用，这样的接口也是最有效的。例如，一个计算器程序可以使用按钮 PushButton 及文本区 Text 模拟实际计算器表面上的各种数字键和算术运算符等，这样用户就能很快学会如何使用该程序。类似的例子还有模拟单放机，电话及传真机（FAX）的程序。

上面讲述的接口的有效性在于它们有对应的为一般用户所熟悉的现实对象，而且这些对象本身也是很简单的。但在有些性况下，概念本身虽然简单，却没有直接的手段将其表达出来。例如，录像机（VCR）是一个简单的设备，每个人都知道其基本用法，但很少有人知道如何用它报时，复杂一些的功能就更难说了。问题就在于 VCR 的设计者只注重其功能，而忽视了其操作的简单性。在现实中没有与 VCR 的接口类似的东西，此时就应该自行设计。最近有人已设计出了一个操作简单的与 VCR 接口的设备。将该设备与 VCR 相连后，用户只需用鼠标选取要录像的东西即可，其它的操作均由接口自动完成。这样的接口不仅简单，而且充分利用了录像机的全部功能。因此，不论某个对象本身是简单还是复杂，其功能是少还是多，都不意味着其接口就应该复杂，这时需要的是具有创意的设计思想。

下面回到前面的几条原则。第一条是简单性，即在屏幕上使用尽可能少的用户接口元件。但如果接口过于简单，这样提供给用户的信息就相应减少，也会影响其使用。Motif 的优点之一是可由用户方便地配置不同级别的接口。例如，用户可设置颜色，字体及其它多种资源。当然应用程序也应考虑到多数用户的要求，将这些用户的要求设置为缺省的状态。总之，接口设计是与具体的应用密切联系在一起，必须兼顾功能和实用两方面的要求。

1.3 Motif是什么?

本节具体讨论 Motif 是什么、它如何能帮助程序员设计用户接口。简单地说，Motif 是一套准则（guideline），规定了图形计算机用户接口的外观（look）和"感觉"（feel）。所谓外观即规定应用程序在屏幕上的显示方式和布局，而"感觉"则规定用户应该如何与应用程序打交道。图 1-1 示出了一个 Motif 应用程序的例子。

用户可通过键盘与应用打交道，也可通过鼠标点中，选取和拖动应用中的各种图形元件。例如，如果将鼠标指针移入应用窗口的标题区（title bar），然后按住鼠标按钮就可将窗口移到另一位置。类似地，在窗口的改变大小的角区(resize corner)中按住并拖动鼠标可以改变窗口的大小。大多数应用还提供了按钮用于启动程序中的某些动作(action)。Motif 还巧妙地用加亮显示和阴影显示取得立体显示效果以及表示按钮的状态(按下还是释放)。大多数应用窗口的顶部有一个菜单拉杆（menu bar），其中的菜单项也可有各自的子菜单。当按下其中某个按钮时，应用可以立即执行一个动作，也可弹出一个对话框要求输入更多的信息。

大多数用户对上述交互作用方式并不陌生，几年前 Apple Macintosh 已广泛使用。但 Motif 的独特之处在于其图形用户接口是与运行应用程序的机器无关的。

Motif 是由 OSF 设计的。OSF 是开放软件其金会的简称，它是由很多公司，例如

HP、Digtal、IBM 等联合创建的一个非营利性的国际组织。OSF 致力于一种通用的软件技术的开发，这种技术将加强不同厂商生产的计算机之间的操作互换性(interoperability)，包括从用户接口到操作系统的各种技术。

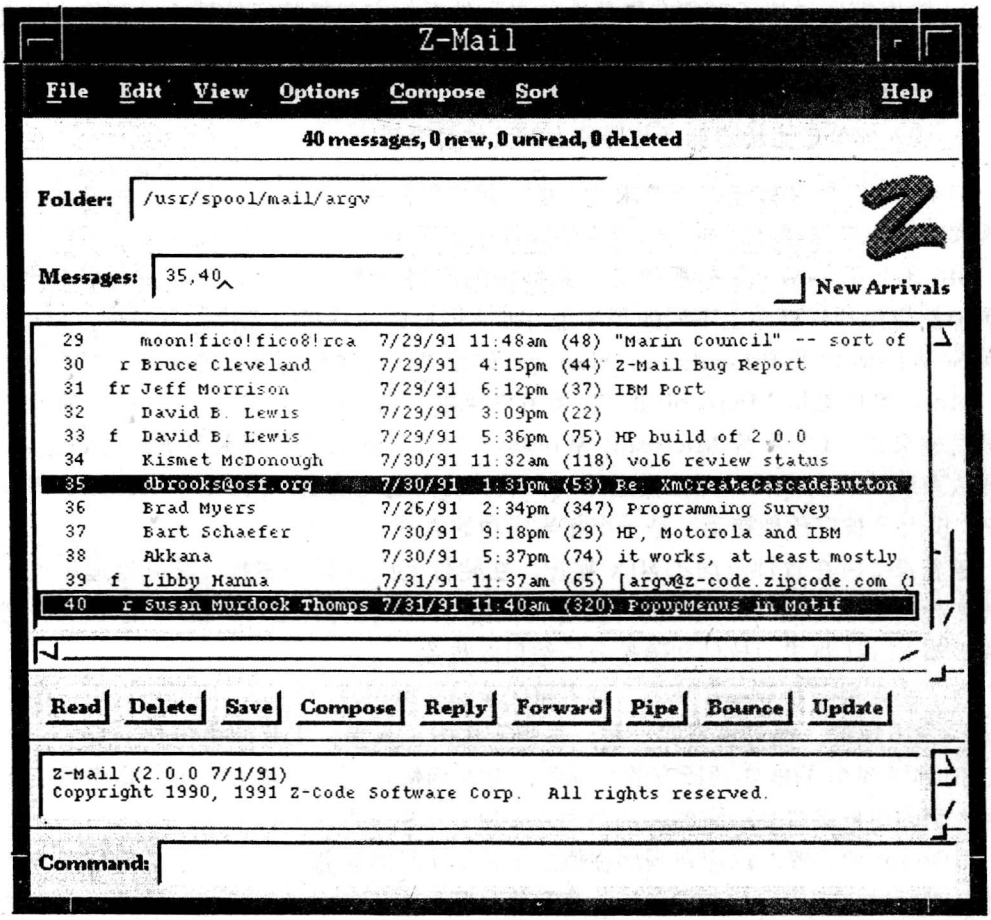

图 1-1 一个 Motif 应用程序

OSF 首先要选取一种适用机型广的窗口系统环境。由于 X 窗口系统是一种基于网络的系统，而且可适用于 UNIX、DOS、Macintosh 及其它操作系统，因此最后决定 OSF／Motif 工具库以 X 窗口系统为基础。

如果能合理地利用 Motif 工具库，就可在相对短的时间内建立与 Motif 规范完全符合的应用程序。Motif 的核心是一种规范（specification），而不是一种实现(implemenatation)。尽管大多数 Motif 应用程序是用 OSF／Motif 工具库实现的，但也可以用其它完全不同的方法建立一个遵循 Motif GUI 的应用程序。Motif 规范的细则参考下面两本书《Motif 风格指南》，其中定义了应用的外部显示和接口操作；《Motif 应用环境规范》，其中定义了应用程序员的接口（API）。

Motif 规范没有对应用的总体布局加以过多的限制，而主要集中于组成用户接口的对象的设计，这些对象包括菜单，按钮，对话框，文本输入区，显示区等。虽然有一些一般

性的规则（例如规范要求每个应用程序应该提供在线范围，帮助菜单应位于菜单拉杆的最右边），但总体而言，用户接口的一致性是依赖于构成它的对象的一致性的。

Motif 规范可分成两个基本的部分：

1. 输出模型描述了对象在屏幕上的显示外观。它包括按钮的形状，三维立体效果的运用，光标和位图的用法，窗口及子窗口的定位。尽管 Motif 给出了有关字体及台面上（desktop）其它的视觉特征的用法的建议，但其使用是相当灵活的。
2. 输入模型指定用户如何与屏幕上的元件打交道。

Motif 规范的关键点在于要求所有的应用之间保持一致性，即类似的用户接口元件的外观和功能也应该类似，与它们所在的具体应用无关。

Motif 几乎可用于任何要求交互式操作的应用程序。即使象 CAD／CAM 软件包或电子邮件应用这样概念上不同的程序也可使用同样的用户接口元件。进一步，Motif 可视为 MS-Windows 和 Presentation Manager 的超集，因为它们建立的准则是相同的。

Motif 接口是以 Microsoft 的“公用用户存取”（Common User Access，简称 CUA）为原型建立的，CUA 为 Microsoft Windows 定义了接口。其原因可规纳为四点。首先，从开放系统的观点来看，上述模型是一个成功的商业化模型；其次，Microsoft Windows 在 PC 机上已被广泛地接受，是一个相当成功的软件。

随着 PC 机在软件上向 UNIX 靠拢，在硬件上向大型机靠拢，其应用程序也作出相应的变化。可以预言 PC 应用的开发者将采用 Motif 作为其 UNIX 版本的 GUI。因此今后若干年 Motif 将成为软件和硬件公司注目的焦点。

为了使应用符合 Motif 规范，读者可完全自行编写用户接口，但要保证用户接口符合 Motif GUI 规范。也可更现实一些，采用现成的工具库。工具库中包含了为某一个特定的 GUI 预先编好的能实现所有的特征和规范的函数，例如使用 Motif 工具库保证不同的应用之间的用户接口是一致的，从而大大方便了程序员的编程。

用 Motif 建立简单的应用程序在概念上是简单而且直接的。大多数应用程序的用户接口部分不超过全部代码的 30–40%。其它部分则是专门针对具体应用问题的，因此要求读者能熟练地使用 C 语言编程。由于很多机器上仍然使用旧的 K&R 版本，因此本书也沿用了这一版本。

1.4 本书的编排

本书的内容一方面考虑到初学者使内容尽可能简单，另一方面也可作为高级程序员的手册，因为其中很多地方用到了 X 窗口系统的很多特征。

每章的内容由浅入深，针对某一个特定的 Motif 接口元件，首先简单介绍其功能，然后讲述创建和操作该对象的具体方法，中间的小节讲述该对象的资源及其它可配置的内容，包括便利函数及相关的事项。每章都有小结对本章的内容简要地回顾。有些章还配有习题，补充说明了正文中未能涉及的有关内容。

读者可按章节的顺序阅读，也可直接阅读感兴趣的章节。事实上，读者阅读过程中不难发现各章之间的内容是互相联系的，并无严格的次序，有时前面的章节用到后面章节的

内容，有时后面章节的内容要求回过头参考前面的内容。当然，本书仍然有一个总体的编排，即前面的几章是对 Motif 的一般性介绍，后面的章就逐一介绍每个 widget，先介绍大的，高级的管理器 widget，然后是原始的 widget 和 gadget。书后有若干附录供读者参考。

全书共分二十章及四个附录，内容概括如下：

第一章　引言。讲述了用户接口的一般性原则及Motif工具库的特点。

第二章　Motif编程模型。用实例讲述了Motif的基本概念。通过对"Hello, World"程序的剖析，说明了所有 Motif 程序通用的程序结构和风格。其中大部分内容已在《X Toolkit Intrinsics 编程手册》(Motif 版本) 一书中讲过。它要求读者对窗口，widget，事件，回调，资源及翻译等基本概念有初步了解。

第三章　Motif widget概述。讲述了建立一个实际的应用要解决的问题：对位于某种主窗口内的原始 widget 进行几何管理的细节；何时在主窗口中显示内容及何时使用弹出式窗口 (对话框和菜单)：应用程序与窗口管理器的关系。学完本章内容之后，程序员应该对 Motif 编程有一个完整的概念，就可以阅读后续任一章节的内容。

第四章　主窗口。讲述了Motif的MainWindow widget。它是很多种应用的框架。MainWindow 是一个管理器 widget，它提供了菜单拉杆，可变动工作区及其它任选的显示和控制区。

第五章　Motif对话简介。讲述了Motif对话的基本概念。它为理解后续章节提供了基础。本章还讲述了 Motif 预定义的 MessageDialog (信息对话框) 类。

第六章　选择对话框。讲述了Motif提供的更复杂的选择对话框，用于给用户提供一个可选择的列表。

第七章　用户定做对话。讲述如何创建新的对话类型，方法是定做Motif对话或创建全新的对话。

第八章　管理器Widget。讲述了不同类的Motif管理器widget。本章提供了几个使用 Form 和 RowColumn 定位的有用的例子。

第九章　ScrolledWindow和Scrollbar。讲述了与卷动有关的内容，特别是由应用定义的卷动，因为 ScrolledWindow widget 提供的简单卷动功能往往不够用。

第十章　DrawingArea Widget。讲述了Motif的DrawingArea (画图区) Widget，它为交互式绘画提供了一个画布。本章没有讲述 Xlib 绘图，而重点在于用实例说明了使用此 Widget 可能遇到的问题。其中要用到一些 Xlib 的知识，可参考《Xlib 编程手册》。

第十一章　标号和按钮。详细讲述了最常用的原始widget：标号Label和按钮Button。还讲到了解释多次接动按钮的方法。PushButton widget 的缺省翻译不支持多次接动按钮。

第十二章　Toggle widget。讲述了ToggleButton及对 RowColumn widget进行设

置，以便对多个 Toggle 进行管理使它们按 CheckBox 或 RadioBox 方式工作。

第十三章　List Widget．讲述了另一个可让用户控制应用的方法．List widget提供了一列文本选项，用户可从中交互式地选择。

第十四章　Scale Widget．讲述了如何使用标尺（scale）显示数据的范围。

第十五章　Text Widget．讲述了如何用Text及TextField widget提供一个用于单个数据项输人的区域或一个完整的文本编辑器．重点讲述了如何屏蔽或转换用户输人的数据以便控制其格式。

第十六章　菜单．详细讲述了Motif的菜单系统。

第十七章　窗口管理器的用法．讲述了Shell widget与Motif窗口管理器之间的关系，内容包括 Shell widget 的资源、如何用 Motif 函数添加或修改窗口管理器协议。

第十八章　剪切板．讲述了在一个应用中与其它的应用交互作用的方法．一个应用将数据放在剪切板上，桌面上的其它应用就可以通过剪切板访问其中的数据。

第十九章　复合串．讲述了Motif的用于记录字符串的字体变化和字符方向的技术．复合串几乎可由所有的 Motif widget 使用。

第二十章　高级对话编程．讲述了前面的章节中没有涉及到的Motif特征，包括创建多级帮助系统，创建 WorkingDialog 允许用户中断一个正在执行的任务，此外还讲述了一种动态改变对话中的象素图的方法。

附录A　Motif函数和宏．列出了Motif库函数的语法。

附录B　Xt和Motif Widget类．讲述了每个Motif用户接口对象的资源，回调和翻译。

附录C　数据类型．讲述了Motif函数用到的特殊数据类型，包括必要的Xt或Xlib数据类型。

附录D　几个示范程序．提供了几个示例程序，说明了正文中没有涉及到的一些技术。

第二章　Motif 编程模型

本章简要介绍 Motif 赖以建立的基础—X Toolkit Intrinsics。这是出于以下考虑。首先，考虑到有些读者不熟悉 Xt，本章定义了其中的一些术语。其次，很多 X Toolkit Intrinsics 的重要内容不可能在本书中详细介绍，因此本章专门给出了与之相关的参考资料，以便读者进一步了解有关细节。第三，本章指出了 Motif 与 Xt 的一些重要区别。最后，由于 Xt 和 Motif 往往提供了多种方法用于实现同一目标，不同的程序员可能会有不同的处理方法，因此本章说明了作者所采用的编程风格和技巧。

2.1　基本的 *X Toolkit* 术语和概念

第一章中已讲到 Motif 的用户接口规范是完全与具体实现无关的，也就是说，不必使用 X 窗口系统实现 Motif 风格的图形用户接口（GUI）。但为了加强移植性和稳健性(robustness)，OSF 选择了 X 作为实现 Motif GUI 的窗口系统以及 X Toolkit Intrinsics (简称 Xt) 作为应用程序员接口（API）的平台。

Xt 提供了一个面向对象的框架，可用来创建可重复使用的、可配置的用户接口部件 Widget。Motif 为一些常用的用户接口元素，例如标号（label）、按钮（pushbutton）、菜单、对话框、卷动杆（scrollbar）及文本输入或显示区等提供了 Widget。此外，还提供了可称为"管理员"（manager）的 Widget，其功能是控制其它 Widget 的布局，这样当应用中要移位或改变尺寸时就不必关心 Widget 放置的细节。

如果没有采用特殊的手段，一般情况下 Widget 的操作是与具体应用独立的。例如，一个按钮 Widget 只知道如何画出自身,当用鼠标选中时如何加亮显示及如何通过调用应用函数响应鼠标按钮操作或其它用户定义的动作。

Widget 的一般行为定义为 Motif 库的一部分。Xt 定义了一些 Widget 的基类，其行为可通过其它的 Widget 类（即子类）加以继承、增强或修改。Widget 基类是所有基于 Xt 的 Widget 集的基础。所谓一个 Widget 集（Widget set），例如 Motif 的 Xm 库，定义了一套完整的 Widget 类，足以满足大多数用户接口的需要。Xt 也提供了用于创建新的 Widget 或修改现有 Widget 的机制。

Xt 也支持"重量较轻"（即较简单）的对象，称为 gadget。gadget 在外观和作用上与 Widget 基本相同，但其行为实际上是由包含 gadget 的相应的管理器 Widget 提供的。例如，一个下拉菜单框（pane）是由按钮 gadget 而不是按钮 Widget 组成的，菜单框完成了很多通常是由按钮 Widget 完成的工作。

在类分级结构中，大多数 Widget 和 gadget 继承了上一级对象的特性。例如，Motif PushButton 类继承了 Label Widget 类的显示一个标号的特性，而 Label Widget 类又继承其"超类"（superclass）的更基本的 Widget 行为。这部分内容可参考

《X Toolkit Intrinsics 编程手册》中有关 Xt 的分类机制的介绍，本书第三章详细讨论了 Motif Widget 的类分级结构。

Xt 的面向对象的方法意味着程序员完全不必关心 Widget 内部的代码。程序员只要调用函数便可建立、管理或取消 Widget。程序员还可以使用一些公共的 Widget 变量，称为"资源"（resource）。简而言之，改变一个 Widget 的内部实现时并不要求对 API 进行修改。面向对象方法的另一个好处是强制程序员以更抽象和更一般化的方式考虑具体的应用，这样从长远的观点看可以形成更好的设计方法，从眼前的利益出发，可以减少程序的错误。

创建一个 Widget 也称为"具体实现"（instantiate）一个 Widget，即请求 Toolkit 按照用户设定的资源建立指定的 Widget 类的一个具体实现（instance）。例如，所有的 Motif PushButton Widget 都能显示一个标号，相应地 PushButton Widget 类的一个具体实现就包含一个实际的标号。

Widget 的很多资源可由用户在运行时刻加以修改。在运行应用程序时，Xt 从一些系统文件和与应用相关的特定文件中自动调入数据，这些文件统称为资源数据库。Xt 使用资源数据库对应用中用到的 Widget 进行配置。

如果程序员不让用户改变资源，可在创建 Widget 时就设置好 Widget 的资源的值，称为对资源进行"硬编码"（hard-coding）。

一般情况下只对一些程序运行必需的资源进行硬编码，其它的资源可让用户重新配置。可配置资源的缺省值存放在应用缺省文件中。本书约定应用缺省文件与应用名字相同，但第一个字母大写，存放在 /usr/lib/X11/app-defaults 目录中。应用缺省文件与其它包含由系统管理员或用户设定的不同值的文件一起调入资源数据库。如果不同的设置之间发生冲突，就要遵循一套复杂的优先级规则来决定资源的取值。参见《X Toolkit Intrinsics 编程手册》中有关如何在应用缺省文件或其它文件中设置资源的内容。

不论资源值是从资源数据库中读取的还是在应用中硬编码设置的，都可在应用中调用 Xt 函数 XtVaSetValues( )随时加以修改。

Motif Widget 拥有丰富的资源可供利用。对于每个 Widget 类，有很多资源从不需要应用程序或用户加以修改。这些不变资源中一部分提供了对 Motif Widget 屏幕三维显示的细致控制，另一部分供 Motif 内部使用，以便使同一个大而复杂的 Widget 以不同的外观显示出来。

回调表（callback list）是一类非常重要的资源，必须在应用中进行设置。每一个与应用相关的 Widget 都公布一个或多个回调资源，在应用程序中必须提供一个与该资源相关的指向应用函数的指针。当 Widget 中发生某件事时，就要调用这些应用函数。例如，当用户在按钮上按动鼠标时，PushButton Widget 就要调用一个应用函数。但是，并非 Widget 中发生的每个事件都会回调应用函数。Widget 本身不用与应用程序相互作用就可以处理很多事件。例如，所有 Widget 都能画出自身。Widget 甚至可以提供类似应用程序的功能。例如，Text Widget 通过称为 actions（动作）的内部 Widget 函数提供了一套完整的编辑命令。动作（action）是通过翻译表（translation table）映射成事件的。可以扩大、有选择性地修改，甚至完全替换原来的翻译表，这些都是通过在 Widget 类、应用或