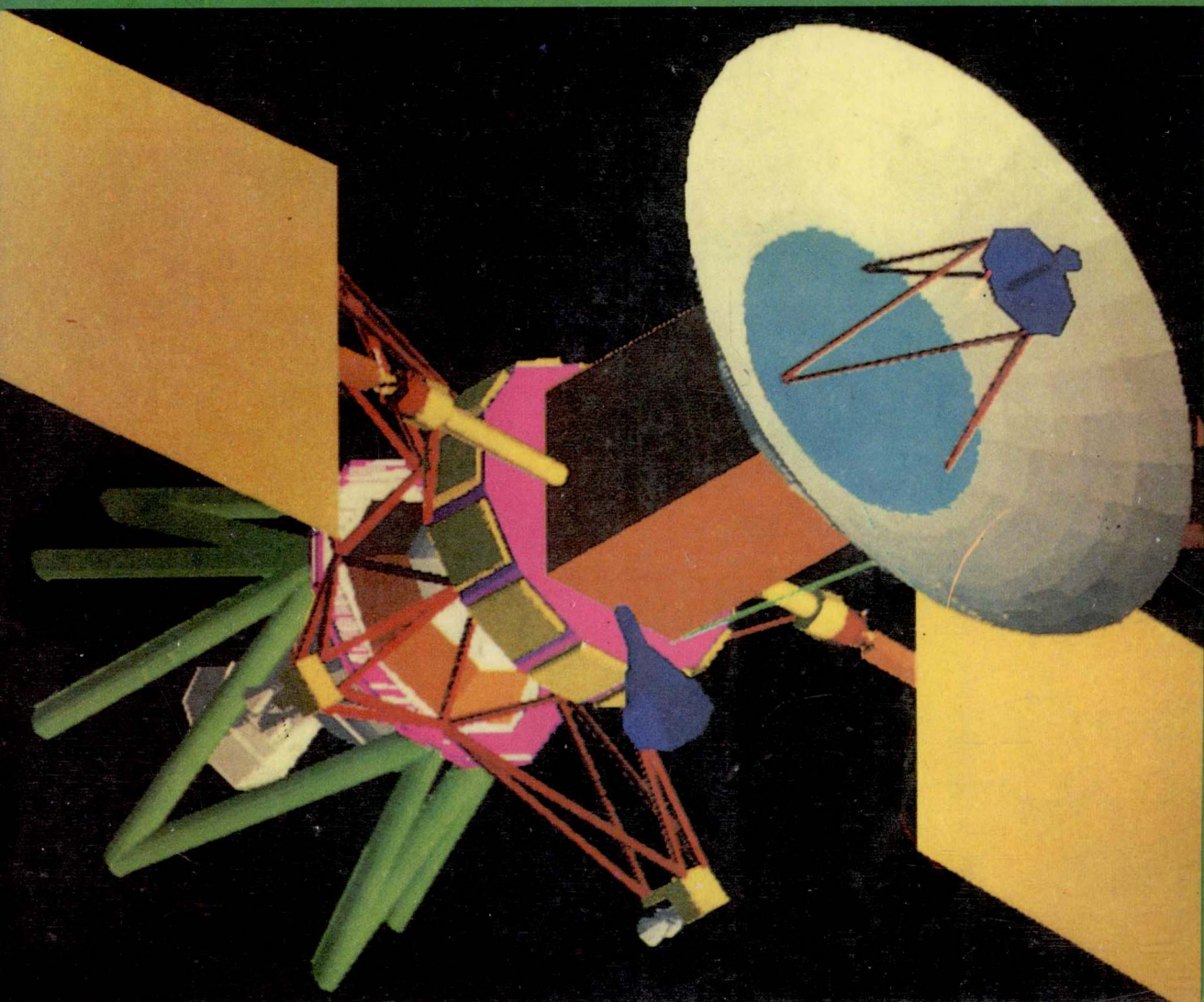




X—Windows 环境下图形 用户界面编程指南 *OPEN LOOK*

王 蕾 文丹岩 编写
覃社庭 占学文

学苑出版社

X—Windows 环境下图形 用户界面编程指南 *OPEN LOOK*

王 蕾 文丹岩
覃社庭 占学文 编写
罗 枫 校

学苑出版社

1993.

(京)新登字 151 号

内 容 提 要

OPEN LOOK 是建立在 X Windows 系统基础上的图形用户界面,它提供了对 X Windows 系统的透明接口,用户可以不熟悉 X Windows 系统就能有效地利用 OPEN LOOK 工具包,因此,可利用它来开发采用“窗口化”先进技术的应用程序。本书是提供给使用 OPEN LOOK 图形用户界面的程序员使用的,它可以帮助程序员有效地利用该工具包。本书的内容涉及到 OPEN LOOK 工具包的总体说明、OPEN LOOK 工具包的总体说明、OPEN LOOK 工具包编程技巧、OPEN LOOK 协议、无鼠标操作和国际化等。本书还提供了一个例子程序,可用来充当最初开发的模板。另外,书中还包含有供程序员编程使用的完整的参考信息。对使用 OPEN LOOK 工具包的程序员而言,本书无疑是一本极佳的参考书籍。它是集体劳动的结晶,凝聚着大家的心血。参加本书编写工作的王蕾、文丹岩、覃社庭、占学文、占卫兵、胡宏建、黄晓宇、张纪录、孙青、刘军、夏春和、丁力等等。

欲购本书的用户,请直接与北京 8721 信箱联系,电话 2562329,邮编 100080。

微机接口与应用系列丛书

X—Windows 环境下

图形用户界面编程指南 OPEN LOOK

编 写: 王 蕾 文丹岩
覃社庭 占学文

审 校: 罗 枫

责任编辑: 甄国宪

出版发行: 学苑出版社 邮政编码: 100032

社 址: 北京市西城区成方街 33 号

印 刷: 兰空印刷厂

开 本: 787×1092 1/16

印 张: 44.125 字 数: 1036 千字

印 数: 1~4000 册

版 次: 1993 年 12 月北京第 1 版第 1 次

ISBN7-5077-0803-9/TP·14

本册定价: 39.00 元

学苑版图书印、装错误可随时退换

前 言

OPEN LOOK 是建立在 X Window 系统基础上的图形用户界面，它提供了对 X Window 系统的透明接口。用户可以不熟悉 X Window 系统就能有效地利用 OPEN LOOK 工具包，因此，可利用它来开发采用“窗口化”先进技术的应用程序。本书是提供给使用 OPEN LOOK 图形用户界面的程序员使用的，它可以帮助程序员有效地使用该工具包。本书的内容涉及到 OPEN LOOK 工具包的总体说明、OPEN LOOK 工具包编程技巧、OPEN LOOK 协议、无鼠标操作和国际化等。本书还提供了一个例子程序，用来充当最初开发的模板。另外，书中还包含有供程序员编程使用的完整的参考信息。对使用 OPEN LOOK 工具包的程序员而言，本书无疑是一本极佳的参考书籍。它是集体劳动的结晶，凝聚着大家的心血。参加本书编写工作的有王蕾、、文丹岩、覃社庭、占学文、占卫兵、胡宏建、黄晓宇、张纪录、孙青、刘军、夏春和、丁力等等。

编 者

目 录

第一章 简介	1
1.1 前言	1
第二章 OPEN LOOK 工具包	2
2.1 X window System 和 Xt Intrinsics	2
2.2 OPEN LOOK 功能选择框	2
2.3 功能选择框函数和应用程序	11
2.4 正文选择操作	31
2.5 设计原理	32
2.6 概述	34
2.7 OPEN LOOK 用户界面的使用	42
第三章 OPEN LOOK 工具包: 编程	57
3.1 简介	57
3.2 如何书写 OPEN LOOK 程序	57
3.3 遍历系统	58
3.4 样本程序	59
3.5 使用平台功能选择框	75
3.6 编程注意事项	79
3.7 元素的设计	80
第四章 X Window 系统 V11 OPEN LOOK 协议	103
4.1 简介	103
4.2 一般因素	103
4.3 窗口特性	105
4.4 与内部用户协议的关系	110
4.5 工作空间和文件管理协议	114
4.6 其它实现问题	117
第五章 无鼠标操作	118
5.1 概述	118
5.2 键盘遍历	118
5.3 键盘累加器	120
5.4 键盘助记符	121
第六章 国际化	124
6.1 国际化	124
6.2 OPEN LOOK 工具包的国际化	126
第七章 扩展功能选择框样本程序	143

7.1 设计目标	143
7.2 程序说明	144
附录 A 手册页简介	191
A.1 手册页简介	191
A.2 一般资料简介	192
附录 B 手册页：便利例程	216
B.1 便利例程简介	216
B.2 便利例程	220
附录 C 手册页：功能选择框	313
C.1 功能选择框介绍	313
C.2 功能选择框	316
附录 D 手册页：废弃的例程	474
D.1 废弃例程介绍	474
D.2 废弃的例程和功能选择框列表	474
附录 E 支持的应用程序	493
附录 F 不支持的应用程序	547
附录 G 词汇表	684

第一章 简介

1.1 前言

1.1.1 目的

本手册的目的是提供使用 OPEN LOOK 图形用户界面 (GUI) 的程序员指南。本书是供有经验的 C 语言程序员用的, 可用它来开发使用“窗口化”前沿技术的应用程序。而 OPEN LOOK GUI 工具包是建立在 X Window SystemTM 之上的, 可不熟悉该系统便能有效地使用 OPEN LOOK 工具包。

本手册的目的是指导程序员使用本工具包。本手册中也包括参考资料和手册页。

1.1.2 概述

本手册的第二章提供了 OPEN LOOK 工具包的全面说明。这种说明包括称作 Xt Intrinsics 的 X Window System 工具包背景信息。Intrinsics 提供了对 X Window System 的透明接口, 有些 Intrinsic 函数是使用 OPEN LOOK 工具包所必需的。特别重要的是要注意到功能选择框 (widget) 概念。

该章提供了说明特定 OPEN LOOK 抽象的基础及对每个 OPEN LOOK 功能选择框和重要功能选择框属性 (称作“资源”) 的简述。

第二章也说明了 OPEN LOOK 函数及其用法。

第三章提供了如何使用 OPEN LOOK 工具包的简述, 其中包括如何布局屏幕; 如何创建和管理功能选择框和“回调”。每个 OPEN LOOK 工具包中带有的样本程序都将给出并预以说明。所提供的应用程序增加了复杂性。

第四章提供了使用 OPEN LOOK 工具包的约定, 以使应用程序能共存和互操作。

第五章说明了无鼠标操作。

第六章提供了大量例子程序来说明大多数 OPEN LOOK 功能选择框的用法。

第二章 OPEN LOOK 工具包

2.1 X Window System 和 Xt Intrinsics

X Window System 提供了创建和管理窗口环境的复杂机制。对 X Window System 的访问通常是通过诸如 OPEN LOOK 工具包的应用程序接口 (API) 工具包来完成的。X Window System 的最底层, Xlib, 是完成基本屏幕和窗口管理操作的 C 例程的集合。应用程序可直接使用 Xlib。如画线、弧、矩形等。

API 工具包可分为两个单独层:

■ Xt Intrinsics

■ Widgets

API 工具包的最底层是 Xt Intrinsics (Xt 是 X Toolkit 的缩写)。Xt Intrinsics 是用来监控有关最终用户交互事件和分配正确的代码来处理这些事件的 C 例程集合。

Xt Intrinsics 的主要功能是提供“功能选择框”的创建和管理。功能选择框是一组提供某种对最终用户“观感”的代码和数据。功能选择框定义了屏幕的矩形区域, 来与诸如 OPEN LOOK GUI 的应用程序接口策略保持一致。功能选择框是一种用户界面部件, 它把 X 窗口与必要的语义组合起来以形成对象。对象提供诸如 button 及 scrollbar 的直觉用户界面抽象。

Xt Intrinsics 含有创建、使用、组织和撤消功能选择框的设施。它们也把来自窗口服务器的事件序列转换成过程调用, 来使应用程序能解释。Intrinsics 记录特定功能选择框的状态并在功能选择框改变长度及位置时商定屏幕的实际大小。

2.2 OPEN LOOK 功能选择框

OPEN LOOK 工具包给应用程序员提供了一组功能选择框和其它用户界面抽象。应用程序员并不主要地涉及定义功能选择框。而与定义特定应用程序屏幕布局和设计用功能选择框来管理特定最终用户交互代码有关。OPEN LOOK API 也提供通过定义 OPEN LOOK 功能选择框子类型来定制应用程序的能力。这是通过提供对定义了功能选择框的基本 C 结构的直接访问来完成的。

功能选择框具有称作资源的某些属性。在编程界面层, 资源是命名的数据项, 是结构定义的命名成份。例如, 与功能选择框有关的一些资源是背景色、父功能选择框 (所有功能选择框都有一个父功能选择框, 其中“最上面”的父功能选择框是应用程序的基窗口)、定义区域的高度和宽度等等。

有些功能选择框只存在于定义其它功能选择框的区域, 即它们为组合功能选择框, 只能作为子功能选择框的父亲而存在。例如, 公告板功能选择框只是提供空间来与其它功能选择框连接。没有孩子的功能选择框是主功能选择框。主功能选择框直接与动作相关联:

它们完成一种功能，输入数据或输出数据。它们不含有其它功能选择框。

每当指定了功能选择框，也可登记例程的名字，这些例程是已编写来处理该功能选择框的；即把这些例程名传给 Xt Intrinsics。这些登记了例程的应用程序称作“回调”。回调用来管理最终用户交互的语义。Xt Intrinsics 也监控登记的应用程序、非图形事件并调度应用程序例程来处理它们。这些特性允许应用程序员在数据库管理、网络管理、进程控制和其它需要响应外部事件的应用程序中使用这种 OPEN LOOK 工具包的实现。

注意：功能选择框如本指南所概述，当在 XWIN Release 4i 中用 X Toolkit Intrinsics 实现时，对 OPEN LOOK 规范和某些二级特性（如带有多种类型控制的颜色和菜单的使用）满足一级一致性。

2.2.1 平台功能选择框和工具框

OPEN LOOK 工具提供了两种其它的抽象界面成份：

■ Flat Widgets

■ Gadgets

这些抽象很类似功能选择框。它们是设计用来通过节省时间和空间来提高 OPEN LOOK 应用程序的性能。它们通过使用冗余窗口的定义来使空间最小。

在大多数部分，术语功能选择框是用来指任意的功能选择框、平台功能选择框及工具框的。

2.2.1.1 平台功能选择框

平台功能选择框是一种维护类似用户界面成份（子对象）集合的功能选择框。平台功能选择框提供许多功能选择框的外观和行为。可把子对象定义成平台功能选择框，因为它们共享同样的基本特性，定义它们会导致性能的提高。例如，平台功能选择框只占用功能选择框等价系统所需的内存的一部分。

在本 OPEN LOOK 版本中，有三种平台功能选择框：

1. Flat Exclusives, 它含有等价的 RectButton 功能选择框。
2. Flat NonExclusives, 也含有等价的 RectButton 功能选择框。
3. Flat checkboxes, 等价于流行在 checkbox 功能选择框中的非独占功能选择框。

一般而言，平台功能选择框具有如下属性：

- 它们为窗口集成器型对象，负责管理一个或多个子对象的观感。
- 在集成器流行之后，就在子对象上完成最小化或无操纵任务。
- 每个集成器只是一个区域，它含有某种类型的一个或多个子对象。
- 集成器内的子对象不具有相关联的窗口及功能选择框结构。

从最终用户的角度来看，平台功能选择框界面和利子功能选择框定义成组合功能选择框的同样界面并无本质区别。

从应用程序员的角度来看，平台功能选择框与传统的功能选择框及工具框具有不同的界面。平台功能选择框更有效且易于处理，特别是在子对象为延期变化的且只有类似的属性时，单个工具包请求指定任意数目的子对象成平台功能选择框，这样就实际上减少了生成复杂图形界面成份所需的代码行数。此外，对平台功能选择框的所有子对象，有一个回调例程。

第三章包含了使用平台功能选择框进行编程的详细说明。

2.2.1.2 工具框

工具框是一种无窗口对象，即它是使用其父功能选择框窗口的功能选择框。在大多数情况下，应用程序的工具框处理和定义成功能选择框的相同对象的处理并没有差别。唯一的差别是在功能选择框类上（例如，Oblong Button Widget 属于 OblongButton WidgetClass，而 OblongButton Gadget 则属于 OblongButtonGadgetClass）。

特别需要注意的是，工具框为 RectObj 类的子类，而功能选择框为 Core 的子类，而工具框只拥有与 Core 类相关联的资源子集。因此应注意在编写工具框代码或在已有应用程序中把功能选择框代码转换成工具框代码时，避免引用存在于功能选择框但不存在于工具框的域。

注意：MenuButton 工具框不能为父辈（即，在创建功能选择框及其工具框时不能用作父参数）。工具框共享某些核心域。但由于它们不是 Core 的子类，它们不能拥有全部 Core 域。特别地，它们不能拥有一个名字域及转换域（因此，转换不能被指定/重载）。

2.2.2 功能选择框的命名约定

OPEN LOOK 工具包程序员总共要处理数百种编程成份：功能选择框、功能选择框资源、Xt Intrinsic 例程，OPEN LOOK 例程等。已采纳了命名约定的完备集来简化读写应用程序的过程。命名约定是最重要的，对有效工具包编程和调试是必需的。

结构成员的名字是小写的；用下划线来连接复合词。这些名字的例子如 tag、display 和 id_type 等。

类型和过程名以大写字母开头。大写用来隔开复合词的成份。这种风格的例子如：

- XtGetValues
- XtSetArg
- XtSetValues
- ArgList
- olInitialize

由于工具包的面向对象本质，所有数据结构和大多数数据类型都是类型定义的（“typedef-ed”）。进一步的风格通常是使用这些私用数据类型派生的 typedef 来完成的。

要强调的附加约定如下：

- 资源名得有前缀 XtN。使用本约定，资源名 XtNbackgroundpixmap 就与结构成员 background_pixmap 相关联。
 - 资源类得有前缀 XtC。例如，资源类 Background 就定义成 XtCBackground。
- 因此，基本的功能选择框命名约定可按下表考虑：
- Intrinsics 过程名以“Xt”起始。
 - OPEN LOOK 例程以“ol”起始。
 - 资源名以“XtN”起始。
 - 资源类名以“XtC”起始。

这些约定类似于 X Window 的 Xlib 所用的约定。

2.2.3 功能选择框资源

资源是一种命名的数据项。本节说明资源的缺省值是如何建立的。下节说明在创建功能选择框之后如何设置和获得这些值。在功能选择框的本质或其“Widget Class”（例如，OblongButton）和专用功能选择框或功能选择框的实例（这里的兰色 OblongButton）间有着基本的区别。功能选择框可定义成其它已定义的功能选择框的子类，然后可把它作是其“super-class”功能选择框。

资源值可由程序设置，由用户指定或指定成缺省值。例如，对 OblongButton 功能选择框，有些专用于给定功能选择框实例的资源项包括前景颜色、正文字体、标号串和标号对齐。其中一些资源成份可来自于功能选择框的超类，如背景颜色、边界颜色和边界宽度。

在 OPEN LOOK 工具包初始化期间（通过 olInitialize 例程的调用），资源按下面所示的次序从几个源地合并而来。在合并期间，无关的资源只是添加给完备集，而覆盖资源则以其最新值来替代。例如，如果资源 XtNforeground 指定于几个源地中，则来自最后源地的 XtNforeground 值就覆盖掉以前的值。

资源值可按给定次序从下列源地提供给应用程序：

1. 内部缺省值——工具包中的每个对象都拥有所用资源的缺省值。这些缺省值一直起作用，除非被来自其它源地的值覆盖值。
2. 应用程序缺省值——应用程序专用的资源文件是从来自应用程序的类名中构造的，并指向场所专用资源文件，它通常是在安装应用程序时由场所管理器安装的。在基于 UNIX 的系统上，应用程序资源文件为：

/usr/X/lib/app-defaults/class

其中：class 为应用程序类名。

3. 服务器资源及.Xdefaults文件——下一个源地为X服务器的RESOURCE MANAGER 特性，它是由 XOpenDisplay 例程返回的。如果这种特性存在，则用户主目录的.Xdefaults 文件就加载，以代替服务器特性。

但当 OPEN LOOK 工作区管理器 (olwsm) 在运行时，RESOURCE MANAGER 特性总存在，可从.Xdefaults 文件中获得其值。这就要注意这些替换源地的差别。

注意：用户在olwsm程序未运行时改变.Xdefaults文件。在工作区管理器运行时做的改变将会丢失。

应用程序可使用 olwsm 程序来存储资源值的改变。这些改变立即存储于.Xdefaults 文件和 RESOURCE MANAGER 特性中。

4. XENVIRONMENT 及 .Xdefault-host 文件——用户的环境变量 XENVIRONMENT 提供下一个源地的文件名。如果该环境变量未设置，则使用用户主目录中的 .Xdefaults-host，其中 host 为用户宿主机的名字。
5. 命令行任选项——下一个源地为命令行，用户可从中提供几种标准和应用程序专用任选项。olInitialize 例程拥有一个标准命令行任选项表来添加资源到资源数据库，它采用参数附加的应用程序专用资源缩写。见 X Toolkit Intrinsics 第四章中“命令行语法分析”中的本表格式。

6. 应用程序覆盖值——在应用程序内设置的资源值为最后的源地。这些赋值会覆盖掉以前的赋值。见下一节“获得和设置功能选择框资源”中有关应用程序如何设置资源值的详细内容。

2.2.4 获得和设置功能选择框资源

Interinsics 提供了获得和设置功能选择框当前资源值和赋值给功能选择框资源的过程。这些函数可在创建功能选择框之后使用。

这些值可在初始化期间设置，可以用对 XtSetValues 的调用来设置，可以用对 XtGetValues 的调用来读取，及用其它方式设置(详见本指南的参考部分)。

下表说明了 XtSetValues 和 XtGetValues 的参数。

	XtSetValues
widget	指定功能选择框。
args	指定要名字 / 值部分的变量长度参数表。
num_args	指定参数表中表目的数目。
	XtGetValues
widget	指定功能选择框。
args	指定名字 / 地址对的变量长度参数表。地址部分是约定成名字的类型对象的地址。
num_args	指定参数表中的表目数。

这两个函数需要把参数的数目 (num_args) 作为参数来传递。函数 XtNumber(array_name) 可用来返回约定长度数组中的表目数。

2.2.5 基本的功能选择框资源

本节说明属于 Core 功能选择框的那些功能选择框资源。Core 功能选择框含有对所有功能选择框为公共的资源定义。所有功能选择框都是 Core 的子类。

这里简述这些资源的说明，详见本指南后面的参考手册部分。这些资源对所有功能选择框是公用的，在实际使用它们之前应参阅一下参考手册。

XtNancestorSensitive

值的范围:

TRUE

FALSE

本变元指定功能选择框的直接父辈是否接收输入事件。

XtNbackground

值的范围:

任何对当前显示器有效的像素值
本资源指定了功能选择框的背景色。

XtNbackgroundpixmap

应用程序可指定用来调整背景的像素图。

其优先级高于 XtNbackground。

表 资源集表

Name	Class	Type	Access
XtNancestorSensitive	XtCSensitive	Boolean	G*
XtNbackground	XtCBackground	Pixel	SGI
XtNbackgroundPixmap	XtCPixmap	Pixmap	SGI
XtNborderColor	XtCBorderColor	Pixel	SGI
XtNborderPixmap	XtCPixmap	Pixmap	SGI
XtNborderWidth	XtCBorderWidth	Dimension	SGI
XtNdepth	XtCDepth	Cardinal	SG
XtNdestroyCallback	XtCCallback	XtCallbackList	SI
XtNheight	XtCHeight	Dimension	SGI
XtNmappedWhenManaged	XtCMappedWhenManaged	Boolean	SGI
XtNsensitive	XtCSensitive	Boolean	GI*
XtNtranslations	XtCTranslations	XtTranslations	G
XtNwidth	XtCWidth	Dimension	SGI
XtNx	XtCPosition	Position	SGI
XtNy	XtCPosition	Position	SGI

存取栏解释如下:

S 值可由 XtSctValues 设置

G 值可由 XtGetValues 读取

I 值可在初始化时设置

* 值以其它方式设置

XtNborderColor

值的范围:

任意对当前显示器有效的像素

本资源指定边界的颜色。

XtNborderPixmap

应用程序可指定用来调整边界的像素图。其优先级高于 XtNborderColor。

XtNborderWidth

值的范围:

$0 \leq \text{XtNborderWidth} \leq \min(\text{XtNwidth}, \text{XtNheight})/2$

本资源设置为功能选择框的边界宽度。0 宽度意味着无边界。

XtNdepth

值的范围:

0 或(当前显示器支持的任意值)

确定功能选择框窗口中的每外像素应该用多少位。本资源值由 Xt Intrinsic 用来设置功能选择框窗口的创建时深度。

XtNdestroyCallback

这是指向含有撤消功能选择框时要调用例程的回调表。

XtNheight

值的范围:

$0 \leq \text{XtNheight}$

本资源含有功能选择框窗口的高度(以像素为单位), 不考虑边界区域。

XtNmappedWhenManaged

值的范围:

TRUE

FALSE

如果设成 TRUE, 则在其实现和管理时映射功能选择框(使之可见)。如果设置成 FALSE, 应用程序负责映射和解除映射功能选择框。如果在功能选择框实现和管理之后值由 TRUE 变成 FALSE, 则功能选择框解除映射。

XtNsensitive

值的范围:

TRUE

FALSE

本资源确定功能选择框是否接收输入事件。注意一下, 表中本资源标记成*。为设置本资源, 应使用函数 XtSetSensitive。这是因为本资源影响到任意子功能选择框的状态, 且 XtSetSensitive 函数产生新值给全部的子辈。

XtNwidth

值的范围:

$0 \leq \text{XtNwidth}$

本资源含有以像素为单位的的功能选择框窗口的宽度, 并不包括边界区域。

XtNtranslations

本资源不应由应用程序设置。

XtNx, XtNy

这两种资源含有功能选择框左上角的 x 坐标和 y 坐标, 不包括边界, 这是相对其父功能选择框的。

2.2.6 为平台功能选择框指定资源

平台功能选择框拥有三种资源:

- 影响集成器对象的资源, 如 `XtNcontainerType` 及 `XtNgravity`, 指定了本容器中的子对象是如何在另一集成器内定位的。
- 必须为每个子对象指定的资源, 如 `XtNlabel`, 指定了进入到按钮及对号框内的标号。
- 可适用于各个子对象或个别适用于各个子对象的资源, 如高度、宽度、字体、背景色等。

这意味着资源指定必须允许指定资源数目的变化(即指定的数目是子对象数的函数), 也要允许指定为集成器定义资源及为子对象定义资源的变化。这种变化是由使用指定的资源表来确定的。

2.2.6.1 子对象资源表

用于平台集成器子对象的资源在 `list format` 中指定。这就允许程序员来确定哪种集成器资源由子对象继承, 哪种资源要单独指定。

每个表是一个应用程序定义的记录的数组(通常是以“c”结构格式或作为数组), 其中每个记录说明了一个特定的子对象。要在表上输入的项是由应用程序员确定的。在同一个应用程序中, 不同的平台集成器可拥有不同的表。对给定的平台集成器, 表上的所有项以同样的格式指定。即, 由于效率原因, 数组中的每个记录都必须具有和数组中其它记录同样的格式, 表中的每个结构具有同样的域。

各个不同的平台集成器功能选择框在其子对象资源中可对每个记录拥有一组不同的记录。例如, 如果应用程序要指“unselect”回调过程给一组独占资源而不给另一组, 应用程序应指定 `XtNunselectProc` 域作为头一个表的元素域而不是第二个表的元素域。由于数据赋值和语法分析的原因, 每个记录的字段必须使用 `XtArgVal` 类型(见第三章中的代码例子)。

2.2.6.2 资源继承

由于所有子对象都是集成器(父辈)的一部分, 因此子对象从父集成器继承任何非专用资源。例如, 如果应用程序需要用于所有子对象的特定字体颜色, 就不必对每个子对象指定 `XtNfontColor`, 只需在父集成器上设置字体颜色资源, 所有的子对象便可使用该字体颜色。

尽管子对象资源为其集成器资源集的一部分, 子对象资源对集成器却没有直接影响。

2.2.6.3 排序中的资源

对于子对象的“格式”是在应用程序域内定义的, 必须给集成器提供一个关于记录格式的提示, 以使它能分析所提供的表。资源名表是解开应用程序子对象表的钥匙。而每个记录中域的次序并不重要, 应用程序必须按其相关值出现在记录中的相同序列给出资源名。

例如, 如果指定平台独占集成器子对象的记录拥有后跟以“`XtNselectProc`”回调字段的“`XtNlabel`”字段, 则应用程序必须给集成器后跟以 `XtNselectProc` 资源名的 `XtNlabel` 资源名。字段的不一致次序会导致子对象实例化时的未定义行为。

2.2.6.4 用于指定平台功能选择框子对象的资源

五个公用资源由每个集成器类用来说明必要的子对象信息:

资源集表

Name	Class	Type	Access
XtNItems	XtCItems	XtPointer	SGI
XtNnumItems	XtCNumItems	Cardinal	SGI
XtNitemFields	XtCItemFields	String *	SGI
XtNnumItemFields	XtCNumItemFields	Cardinal	SGI
XtNitemsTouched	XtCitemsTouched	Boolean	SG

XtNItems	这是子对象项表。
XtNnumItems	指定表中子对象的数目。
XtNitemFields	含有用来分析 XtNItems 表中记录的资源名表。
XtNnumItemFields	指定 XtNitemFields 中资源名的数目。
XtNitemsTouched	值为 TRUE 或 FALSE(缺省值)。每当直接修改项表时,都必须置之成对平台功能选择框集成器为 TRUE,以使它能更新视觉。

2.2.7 屏幕分辨率和颜色

2.2.7.1 各种最终颜色

OPEN LOOK 工具包的这种实现支持各种最终颜色表示,即基本颜色及好看的阴影。

如果想要调整自己的颜色选择,可在自己的.Xdefaults 文件中编辑颜色资源。注意,只能在 OPEN LOOK 当前未运行时做一工作。

资源	效果
* Background	窗口和功能选择框背景
* TextBackground	TextEdit, Xterm
* TextFontColor	TextEdit, TextField, ScrollingList, xterm
* inputFocuscolor	光标、横过、高亮
* inputWindowHeader	带有输入焦点的窗口标头
* olwsm. workspace	工作区颜色
* foreground	窗口标题, 窗口菜单标号, 按钮, 滚动条滑动器
* borderColor	功能选择框和窗口边界
* FontColor	TextEdit, TextField 和 ScrollingList 功能选择框及 xterm

注意: 参见本指南附录中“手册页: 功能选择框”里的功能选择框资源集。

2.2.7.2 设备分辨率

下表给出了本工具包提供的屏幕格式。假定使用 11 英寸对角线和 13 英寸对象线（正常）监视器，本表也表明了相应的像素密度。数字按水平每英寸像素数和垂直每英寸像素数给出。

适配器类型	格式	11"监视器	13"监视器
EGA	640×350	76×55	66×48
EGA (AT&T 扩展)	640×480	76×76	66×55
VGA	640×480	76×76	66×66
增强 VGA	800×600	95×95	83×83

在满足这些分辨率的屏幕上，由本工具包提供的视频遵守 12 点长度视频，这是 OPEN LOOK 规范所需的。但本工具包可以任意分辨率工作。在低分辨率屏幕上，如 CGA 格式，视频比 12 点大，在高分辨率屏幕上，视频比 12 点小。

2.2.8 支持的字体

2.2.8.1 标准字体

所定义的 OPEN LOOK 字体“Lucida”用于作为所存标号化控制的标号中及作为正文功能选择框中的缺省字体。

2.2.8.2 用于分辨率的字体自动选择

本工具包自动地选择适当的缺省字体来匹配设备的分辨率。应用程序可重载字体选择，但本工具包并不自动调整字体以适应屏幕分辨率的改变。

注意：字体必须按每屏的基础来调整。

2.3 功能选择框函数和应用程序

本节说明了 OPEN LOOK 功能选择框和处理例程，为 OPEN LOOK GUI 定义的功能选择框列表如下：

Action Widgets

OblongButton

RectButton

CheckBox

MenuButton (was: ButtonStack)

AbbrevMenuButton (was: AbbrevStack)

Slider

Scrollbar

Stub