



北京希望电脑公司 X Window System 系列丛书 之六

X Toolkit Intrinsic

参考手册

(11.4版)

(共九册)

海洋出版社

北京希望电脑公司 X Window System 系列丛书之六

X Toolkit Intrinsic 参考手册

叶欣
周芳芹 编
薛焱森

海 洋 出 版 社

一九九一年五月·北京

内 容 提 要

X Window System(X 窗口系统)是 80 年代末推出的可移植标准,是软件开发和运行的必备支持系统。

本书是《X Window System 系列丛书》的第六册,它与第五册配套使用。全书包括五个部分和七个附录,分别介绍了内部函数和宏、原型过程、内部类、内部方法、Athena 类的用法、参数、描述信息、结构、应用示例和其它相关信息。

欲购本书的用户可直接与北京 8721 信箱联系,电话 2562329 邮政编码 100080。

X Window System 系列丛书之六
X Toolkit Intrinsics 参考手册
(11.4 版)

叶欣 周芳芹 薛焱森 编

审校:刘莉蕾

责任编辑:阎世尊

海洋出版社出版发行(北京市复兴门外大街 1 号)

双青印刷厂印刷

开本: 787×1092 1/16 印张: 28.4 字数 689013

1991 年 5 月第一版 1991 年 5 月第一次印刷

印数: 1—3000 册

ISBN 7-5027-1909-1/TP·25

定价: 19.00

序

近几年来,国际计算机界出现了一股“窗口”热,从微型机到小型机到大型机,从工作站到网络,窗口系统已成为软件开发和运行的必备支持系统。

由美国 Massachusetts 大学、IBM 公司、DEC 公司、AT&T 公司、Microsoft 公司等十几家团体联合开发的 X 窗口系统(X Window System)更成为窗口中热点,自从它在八十年代末推出以来,得到了国际软件界的高度评价和肯定。

X Window System 运行于工作站上,以其极其方便的用户接口、强大的编程功能和高质量的代码,改变了整个工作站的世界;X Window System 运行于 386、486 等微机上,更能充分发挥这些微机的特点,利用它们开发更好更完美的软件。

X Window System 推出以后,经过不断的更新和完善,已达到 11.4 版,本版系统在原来的基础上,性能方面又有了很大的提高。

X Window System 主要有核心系统(协议)和建立在其上的应用程序环境工具(Xt、XView、Xlib 等)组成,为了使国内广大计算机用户了解并掌握 X Window System 的基础知识和高级程序设计技术,我们在近几年使用它们的基础上,给合国外最新软件和资料,特编辑了 X Window System 系列丛书,包括:

- 第一册:《X 协议参考手册》
- 第二册:《Xlib 编程手册》
- 第三册:《Xlib 参考手册》
- 第四册:《X 教程》
- 第五册:《X Toolkit Intrinsics 编程手册》
- 第六册:《X Toolkit Intrinsics 参考手册》
- 第七册:《X Window 编程指南》
- 第八册:《X View 编程手册》
- 第九册:《X Window 高级编程指南》

《X Toolkit Intrinsics 编程手册》和《X Toolkit Intrinsics 参考手册》配套使用,主要介绍 X Window 软件包中提供的 Xt 工具,其中第六册描述了 Xt Intrinsics 库中每一个函数、宏、原型过程、内部类、方法和 Athena 类的用法、参数、使用说明、结构和其它相关信息。

本丛书编辑过程中,得到了国内许多专家、学者的帮助和支持,特别是石清教授,在百忙之中审阅了全书,提出了许多修改意见,刘京同志、薛梅同志、王丽同志、张阳同志为本丛书的完成作出了许多努力,编者在此向他们表示感谢。

由于 X Window System 资料非常新颖,国内又没有可供借鉴的中文参考资料,所以本丛书在编辑过程中,难免会存在一些不足之处,希望各位同仁批评指正。

本丛书出版过程中,得到了中国科学院希望高级电脑技术公司资料部秦人华经理、杨淑欣老师的大力帮助和支持,编者在此表示衷心的感谢。

编者

一九九一年五月

目 录

前言	1
介绍	3
1.1 内部函数及目标基	4
1.2 目标基实例化	4
1.2.1 初始化 X 工具包	5
1.3 目标基管理	6
1.4 外壳目标基	6
1.5 弹出目标基	7
1.5.1 弹出目标基类型	7
1.6 几何管理	8
1.6.1 启动几何变化	9
1.7 事件管理	9
1.7.1 增加及删除附加的事件源	10
1.7.2 把事件约束到目标基串	10
1.7.3 读事件、发送事件	11
1.7.4 设置并检查目标基的灵敏度	11
1.7.5 加进背景工作过程	11
1.7.6 目标基外露	12
1.7.7 X 事件处理程序	12
1.8 反调用	12
1.9 资源管理	13
1.9.1 资源表	13
1.9.2 资源类型	13
1.9.3 子资源	14
1.9.4 资源转换	15
1.10 翻译管理	15
1.10.1 动作表	15
1.10.2 动作分路	16
1.10.3 翻译表	16
1.10.4 翻译表管理	16
1.10.5 使用加速程序	17
1.10.6 键盘解释程序	17
1.10.7 多时间按键(multi-click)	17
1.11 实用函数	17
1.12 非目标基对象	18

第一部分 内部函数及宏指令	19
XtAddActions	10
XtAddCallback	22
XtAddCallbacks	23
XtAddConverter	24
XtAddEventHandler	25
XtAddExposureToRegion	27
XtAddGrab	27
XtAddInput	28
XtAddRawEventHandler	29
XtAddTimeout	30
XtAddWorkProc	31
XtAppAddActionHook	31
XtAppAddActions	32
XtAppAddConverter	33
XtAppAddInput	35
XtAppAddTimeout	36
XtAppAddWorkProc	36
XtAppCreatesShell	37
XtAppError	38
XtAppErrorMsg	39
XtAppGetErrorDatabase	40
XtAppGetErrorDatabaseText	40
XtAppGetSelectionTimeout	42
XtAppInitialize	42
XtAppMainloop	43
XtAppNextEvent	44
XtAppPeekEvent	44
XtAppPending	45
XtAppProcessEvent	46
XtAppReleaseCacheRefs	47
XtAppSetErrorHandler	47
XtAppSetErrorMsgHandler	48
XtAppSetFallbackResources	49
XtAppSetSelectionTimeout	49
XtAppSetTypeConverter	50
XtAppSetWarningHandler	50
XtAppSetWarningMsgHandler	51
XtAppWarning	52
XtAppWarningMsg	52
XtAugmentTranslation	53
XtBuildEventMask	54
XtCallAcceptFocus	55
XtCallActionProc	56
XtCallbackExclusive	56
XtCallbackNone	57
XtCallbackNonexclusive	58
XtCallbackPopdown	58

XtCallbackReleaseCacheRef	59
XtCallbackReleaseCacheRefList	59
XtCallCallbackList	60
XtCallCallbacks	60
XtCallConverter	61
XtCalloc	62
XtCheckSubclass	63
XtClass	63
XtCloseDisplay	64
XtConfigureWidget	64
XtConvert	65
XtConvertAndStore	66
XtConvertCase	68
XtCreateApplicationContext	69
XtCreateApplicationShell	69
XtCreateManagedWidget	70
XtCreatePopupShell	70
XtCreateWidget	72
XtCreateWindow	74
XtDatabase	75
XtDestroyApplicationContext	76
XtDestroyGC	76
XtDestroyWidget	77
XtDirectConvert	78
XtDisOwnSelection	79
XtDispatchEvent	79
XtDisplay	80
XtDisplayInitialize	80
XtDisplayOfObject	82
XtDisplayStringConversionWarning	82
XtDisplayToApplicationContext	82
XtError	83
XtError	83
XtFindFile	84
XtFree	85
XtGetActionKeysym	86
XtGetApplicationNameAndClass	86
XtGetApplicationResource	87
XtGetConstraintResourceList	89
XtGetErrorDatabase	89
XtGetErrorDatabaseText	90
XtGetGC	90
XtGetKeysymTable	91
XtGetMulticlickTime	92
XtGetResourceList	92
XtGetSelectionRequest	93
XtGetSelectionTimeout	94
XtGetSelectionValue	94
XtGetSelectionValueIncremental	95

XtGetSelectionValues	96
XtGetSelectionValuesIncremental	97
XtGetSubresources	97
XtGetSubvalues	99
XtGetValues	100
XtGrabButton	101
XtGrabKey	102
XtGrabKeyboard	103
XtGrabPointer	103
XtHasCallbacks	104
XtInitialize	105
XtInitializeWidgetClass	109
XtInsertEventHandler	109
XtInsertRawEventHandler	110
XtInstallAccelerators	111
XtInstallAllAccelerators	112
XtIsApplicationShell	113
XtIsComposite	113
XtIsConstraint	113
XtIsManaged	114
XtIsObject	114
XtIsOverrideShell	114
XtIsRealized	115
XtIsRectObj	115
XtIsSensitive	116
XtIsShell	116
XtIsSubclass	116
XtIsTopLevelShell	117
XtIsTransientShell	117
XtIsVendorShell	118
XtIsWidget	118
XtIsWMSHELL	118
XtKeysymToKeycodeList	119
XtLastTimeStampProcessed	119
XtMainLoop	120
XtMakeGeometryRequest	120
XtMakeResizeRequest	122
XtMalloc	123
XtManageChild	123
XtManageChildren	124
XtMapWidget	125
XtMenuPopdown	125
XtMenuPopup	126
XtMergeArgLists	127
XtMoveWidget	127
XtName	128
XtNameToWidget	128
XtNew	129
XtNewString	129

XtNextEvent	130
XtNumber	130
XtOffset	131
XtOffsetOf	131
XtOpenDisplay	132
XtOverrideTranslations	133
XtOwnSelection	134
XtOwnSelectionIncremental	134
XtParent	136
XtParseAcceleratorTable	136
XtParseTranslationTable	136
XtPeekEvent	137
XtPending	138
XtPopdown	138
XtPopup	139
XtPopupSpringLoaded	140
XtProcessEvent	140
XtQueryGeometry	141
XtRealizeWidget	143
XtRealloc	144
XtRegisterCaseConverter	144
XtRegisterGrabAction	145
XtReleaseGC	146
XtRemoveActionHook	147
XtRemoveAllCallbacks	147
XtRemoveCallback	147
XtRemoveCallbacks	148
XtRemoveEventHandler	148
XtRemoveGrab	150
XtRemoveInput	151
XtRemoveRawEventHandler	151
XtRemoveTimeout	152
XtRemoveWorkProc	153
XtResizeWidget	153
XtResizeWindow	154
XtResolvePathname	154
XtScreen	156
XtScreenOfObject	156
XtSetArg	156
XtSetErrorHandler	157
XtSetErrorMsgHandler	158
XtSetKeyboardFocus	159
XtSetKeyTranslator	160
XtSetMappedWhenManaged	161
XtSetMultiClickTime	162
XtSetSelectionTimeout	162
XtSetSensitive	162
XtSetSubvalues	163
XtSetTypeConverter	164

XtSetValues	165
XtSetWarningHandler	166
XtSetWarningMsgHandler	167
XtSetWMColormapWindows	168
XtStringConversionWarning	169
XtSuperclass	169
XtToolkitInitialize	169
XtTranslateCoords	170
XtTranslateKey	170
XtTranslateKeycode	171
XtUngrabButton	172
XtUngrabKey	173
XtUngrabKeyboard	173
XtUngrabPointer	174
XtUninstallTranslations	174
XtUnmanagechild	174
XtUnmanageChildren	175
XtUnmapWidget	176
XtUnrealizeWidget	176
XtVaAppCreateShell	176
XtVaAppInitialize	177
XtVaCreateArgsList	178
XtVaCreateManagedWidget	178
XtVaCreatePopupShell	179
XtVaCreateWidget	179
XtVaGetApplicationResources	180
XtVaGetSubresources	180
XtGetSubValues	181
XtVaGetValues	181
XtVaSetSubValues	182
XtVaSetValues	182
XtWarning	183
XtWarningMsg	183
XtWidgetToApplicationContext	184
XtWindow	184
XtWindowOfObject	184
XtWindowToWidget	185

第二部分 原型过程.....186

XtActionHookProc	186
XtActionProc	187
XtCallbackProc	188
XtCancelConvertSelectionProc	189
XtCaseProc	190
XtConvertArgproc	191
XtConverter	193
XtConvertSelectionIncrProc	196
XtConvertSelectionProc	197

XtDestructor	200
XtErrorHandler	200
XtErrorMsgHandler	201
XtEventHandler	202
XtFilePredicate	203
XtInputCallbackProc	204
XtKeyProc	205
XtLoseSelectionIncrProc	207
XtLoseSelectionProc	208
XtOrderProc	208
XtResourceDefaultProc	209
XtSelectionCallbackProc	210
XtSelectionDoneIncrProc	211
XtSelectionDoneProc	211
XtTimerCallbackProc	212
XtTypeConverter	212
XtWorkProc	216

第三部分 内部函数类.....218

Composite	223
Constraint	225
Core	228
Object	232
OverrideShell	235
Rectobj	235
Shell	239
TopLevelShell	245
TransientShell	247
VendorShell	247
WMShell	248

第四部分 内部函数方法.....253

accept_focus	253
change_managed	253
class_initialize	254
class_part_initialize	255
delete_child	256
destroy	257
constraint_destroy	258
display_accelerator	258
expose	258
geometry_manager	261
get_values_hook	264
initialize	265
constraint_initialize	267
initialize_hook	268
insert_child	269
query_geometry	270

realize.....	273
resize.....	275
set_values.....	276
constraint set_values.....	278
set_values_almost.....	279
set_values_hook.....	281

第五部分 Athena 类.....283

Box.....	283
Command.....	285
Dialog.....	288
Form.....	290
Grip.....	293
Label.....	295
List.....	297
MenuButton.....	301
Paned.....	304
Scrollbar.....	309
SimpleMenu.....	313
Sme.....	318
SmeBSB.....	319
Smeline.....	321
Stripchart.....	322
Template.....	325
Text.....	331
Toggle.....	346
ViewPort.....	350

第六部分 附录.....353

附录 A 函数摘要.....	354
附录 B X 工具包数据类型.....	367
附录 C 事件参考.....	380
附录 D 标准错误和警告信息.....	413
附录 E 资源文件格式.....	418
附录 F 翻译表语法.....	419
附录 G StringDef.h 头文件.....	426
附录 H 发行说明.....	431

前言

X 工具包的一些说明

X 工具包是两个 C 语言子例程库(Xt 及 Xaw)的集合名字,使用称之为目标基的可重用部件。它可以简化 X 窗口系统应用程序的开发工作。典型的目标基有:滚动棒,菜单,对话框,文本编辑区,作图区等等。每一个目标基都由它自己的 X 窗口构成,但在这个窗口中进行的绝大多数工作都已被作好了,应用程序员所必须作的工作就是装配这些目标基,写一些在响应目标基的事件中要调用的应用专用码。

Xt 库(内部函数)由一些使用及构造目标基的例程组成,目标基是用面向对象的类机构来定义的。Xaw 目标基库建立在 Xt 的基础上,提供数量较小的一些目标基,写一些简单的应用程序要用到它。

Xt 内部函数是用 Xlib(X 库)写成的,它是到 X 窗口系统最低一级的 C 语言接口。在任何允许编写 X 应用程序的系统上,X 标准(由 X 集团建立)都要求 Xt 内部函数及 Xlib。

Xaw 是 MIT 工程 Athena 开发研究的,首字母缩字"Xaw"表示 Athena Widgets(雅典娜目标基)。最初,Xaw 被设计成一种简单的内部函数示范及对内部函数的测试工具,而并不作为写所要求的应用程序的一个完整的目标基。还有一些系统卖主所提供的目标基可以完成一些特殊的用户接口。两类主要的目标基是 Qsf/Motif,其用户接口类似于 IBM 的表示管理程序,AT&T 及 Sun 的 OPENLOOK 目标基集。

X 工具包内部函数的工作方式与任何 Xt 兼容的目标基的工作方式相同。尽管有不同的审美要求或经济要求,但事实上,可以把不同的目标基集组合在同一应用之中。

注:还有一些其它的 X 工具包具有类似的对象——即容易编写标准的 X 应用程序。这些工具包包括:Andrew(来自 Carnegie-Mellon 大学),InterVnew(来自 Stanford)及 XView(来自 Sun)。这些不仅仅是一些不同的目标基集,而且也完全是不同的工具包。这些工具包与 Xt 是不兼容的。

有关本书的说明

本书是 X Window System 系列丛书的第六册。每一个 Xt 内部函数功能及宏指令,有用的原型过程,内部函数级及方法,Athen 目标基都有参考版面。为了便于查阅,每一节中的参考部分按字母顺序进行组织。

第五册和第六册是合在一起使用的。第五册中有 X 工具包的解释,其中包括编程指导材料及许多编程实例。按照任务或者标题排列,每一章都引入一组 X 工具包函数,描述从概念上所遵照的基本原则,说明他们怎样在写应用程序时被使用的。这册被组织得就象一个指导教师那样有用,且也是一本面向应用的参考手册。为了完全弄明白第五册中的例子,就要用到第六册中每一个函数准确的调用程序。为了完全明白如何使用第六册中的每一个函数,即使最有经验的工具包"计算机窃贼"也需要第五册的解释及例子。

尽管该工具包打算掩盖 Xlib 所提供的低级 X 接口,但由于没有 Xt 功能部件做同样的工作,所以需要 Xlib 等目标基时,在几个地方还有低级 X 接口。第五册描述了使用 Xlib 最常用的情况,但并未提供对特殊函数调用的参考。有关特殊的函数调用,请参阅第二册

《Xlib 编程手册》、第三册《Xlib 参考手册》。

本书是如何组织的?

第六册由工具函数的几部分参考信息组成的,还有很多有用的附录。

本书的组成情况如下:

“前言”:描述本书的组织情况。

“介绍”:概述参考信息所分成的函数区。

第一部分“内部函数及宏指令”:其中包括内部函数及宏指令参考信息,在每一个参考的标题中说明该函数是否适用于使用目标基还是建造目标基,全部按字母顺序编写。

第二部分“原型过程”:在这里列出说明应用程序反调用例程、动作、目标基方法、及其它用户所提供的函数所使用的原型程序。

第三部分“内部函数分类”:其中包含所要求的目标基分类的参考——核心、复合、约束及外壳。

第四部分“内部函数方法”:其中有内部函数类所使用的方法。

第五部分“Athena 类”:描述了 Athena 类的参考资料。

附录 A “函数摘要”:这里有快速参考表,按字母顺序及逻辑顺序列出每一个函数。

附录 B “X 工具包数据类型”:按字母顺序列出并解释, Xt 函数及宏指令变量所使用的结构, enums 及其它类型定义。

附录 C “事件参考”:描述以参考格式存在的每一个事件的类型。每一个参考版面中都有当事件被产生时如何选择事件,事件结构的内容并注释了如何使用这些事件。

附录 D “标准错误及警告信息”:X 工具包除了返回由于出错而引起的其它信息外,还返回可能存在的出错信息或警告信息。

附录 E “资源文件格式”:它解释了该资料文件所使用的 EBNF 语法。

附录 F “翻译表语法”:它解释了翻译表语法。这里讨论了修饰及事件类型。

附录 G “StringsDefs.h 头文件”:它把在 StringsDefs.h 中所找到的标识符分组。

附录 H “发行说明”:它摘要地解释了第三次发行及第四次发行的一些细微的差别。

使用本书的前提

使用这本书时对读者面向对象编程或者 X 窗口系统的知识没有任何要求。然而,对于那些先进的题目,读者则需要查阅其系列书中的前边几册——第二册《Xlib 编程手册》、第三册《Xlib 参考手册》。

尽管所提供的这些例子偶然用到语言工具(在用 X 工具包编程时却需要语言工具或者说,它是很有用的),但读者应熟悉 C 语言编程。另外,熟悉光栅图形也很有好处。

介 绍

在埋头攻读参考信息的第五部分之前,浏览一下内部函数大有用处。在本节中就可以浏览内部函数,另外,还要描述如何在参考部分找到要找的信息资料。

在编写软件手册时典型的问题通常是如何解决好指导性信息及参考信息的矛盾。长期以来,我们就发现 UNIX 参考手册中的字母"man page(手控版面)"方法使该书成为我们所见到的最容易理解的参考手册,然而 UNIX 的反对者却指出这种方法对使用该手册初学者来说太难了。

我们相信,解决问题的办法就是既有指导手册又有参考手册,在指导手册中让读者从整体上去理解,而字母参考手册则给出全部的附加细节说明。第六册就是一本参考手册而第五册是指导手册,在使用时,这两册最好不要分开。

所有 X 联合标准,例如 Xt 内部函数都由一些文件定义,而不是由原型码来定义。内部函数规范全部是一些细节,主要用于帮助专家程序员端口或者完成各种系统的内部函数,该规范采用了一些熟悉的概念,其中许多显然都不在其开发研究的概念之中(由于该内部函数在许多领域都开辟了新的天地,其中包括用户的配置能力,可移植性,共用化(使程序可用多个语言工作)),尽管如此,所有手册都应建立在该规范的基础之上,就是因为它就是内部函数的定义。为了准确保持并忠实于 Xt 规范,实际上我们包含了其全部文本。尽管这样,我们还是做了尽可能多的工作以使得新手(非专家)更容易阅读该书,更容易使用。

首先,我们相当仔细地编写了这册材料使得包括进了更多的概念和文本材料。书中我们试图提供足够的文本以使读者可以明白每个函数如何使用,而不用去埋头钻研过多的细节延误阅读它们的时间。例如,在讨论原型过程时,我们对所提供的规范做了补充性的描述以把用来标明那个过程的功能说明包含进去。在许多地方我们还在许多版面中加进了一些例子和结构定义。

第二还加进了一级不按字母排序的结构。这本手册共分五个部分(加上这个介绍及一些附录)。

1. 内部函数及宏指令——这是一些应用程序编写者及目标基编写者所使用的函数。
2. 原型过程——这是一些用户或者目标基定义的函数,是由其中一个内部函数用内部功能注册的。
3. 内部函数类——这是一些内部函数定义的对象,其建立在目标基集定义的所有对象的基础上。
4. 内部函数方法——这是一些由内部函数编写者定义的函数以完成一个目标基。目标基把这些方法做为内部函数类来定义。
5. Athena 类——这是在 Athen 目标基集中对每一个目标基的描述。Athena 集用于示范和测试目标基,而不是用于商务应用程序开发,但由于它的应用范围广,所以在写一些简单的应用程序时很有用。

介绍的其余部分由概述材料组成,它出自 Xt 规范,遵守 Xt 原来的组织情况。每一部分都由一些文本开头,以介绍一组特殊的函数或过程类型,其后有那个函数组的列表信

息。

1.1 内部函数及目标基

内部函数(也称之为 Xt)及目标基就构成了 X 工具包,内部函数提供建立各种目标基集及应用环境所要求的各种基本机构。由于该内部函数屏蔽了目标基程序员及应用程序员编程所要做的细节,所以建立目标基及应用环境工作可以延伸至支持单独开发,也支持新的或者扩充的部件,遵照一些约定,目标基程序员就可以用新的方法扩充原有的目标基集,已有的程序也可以很好地使用这些扩充功能。

内部函数是一个程序库,处于 Xlib 的上一层。由于这种原因,内部函数提供扩充 X 窗口系统所提供的基本编程的抽象概念的机构(函数及结构)。

典型的 X 工具包应用很象一个给定目标基集的客户,内部函数功能的一个子集,一小组 Xlib 函数的客户。在同一时间内,目标基集既是内部函数的客户,又是 Xlib 的客户,而内部函数只是 Xlib 的客户。

X 工具包为应用程序员提供以下函数:

- 写应用程序的任一接口(目标基集)
- 在写应用程序时也可以使用的一些小的内部函数机构。

X 工具包为目标基程序员提供以下函数

- 建立目标基的一组内部函数机构
- 构造及复合目标基的构造模型
- 统一的编程接口(目标基集)

为了可以扩充,X 工具包的策略很自由。应用环境——而不是 X 工具包——定义、实现及强行完成以下的内容:

- 策略
- 统一性
- 风格

每一个目标基的执行程序都定义它自己的策略。X 工具包设计允许开发完全不同的目标基执行程序。

对于有关的讨论,请参阅以下 Xt 例程及方法:

<i>XtCheckSubclass(1)</i>	<i>XtIsOverrideShell(1)</i>	<i>XtIsVendorShell(1)</i>
<i>XtClass(1)</i>	<i>XtIsRectObj(1)</i>	<i>XtIsWidget(1)</i>
<i>XtInitializeWidgetClass(1)</i>	<i>XtIsShell(1)</i>	<i>XtIsWMShell(1)</i>
<i>XtIsApplicationShell(1)</i>	<i>XtIsSubclass(1)</i>	<i>XtSuperclass(1)</i>
<i>XtIsComposite(1)</i>	<i>XtIsTopLevelShell</i>	<i>class_initialize(4)</i>
<i>XtIsConstraint(1)</i>	<i>XtIsTransientShell(1)</i>	<i>class_part_initialize(4)</i>
<i>XtIsObject(1)</i>		

1.2 目标基实例化

对象实例的集合构成一个目标基树。XtAppCreateShell 所返回的外壳目标基是目标基树实例的根。有一个或者多个孩子的目标基是该树的中间节点,没有任何孩子的目标基是目标基树的叶子。除了弹出(pop_up)孩子之外,这类目标基树实例定义相关的 X 窗口树。

目标基可以为复合的也可以为原始的。这两种目标基都可以有孩子,但内部函数提供

一组管理机构，以在复合目标基、复合目标基的孩子及其他客户之间进行构造和形成接口。

复合目标基，即复合类的子类，是用于任意的、但由执行程序定义的孩子的集合的容器，可以由复合目标基本身，或由其它客户或由复合目标基及客户两者合起来对它进行实例化。复合目标基也含有管理任何孩子目标基的几何外形的方法。在非常规的环境下，复合目标基可以有零孩子，但它通常至少有一个孩子。相反，含有孩子的原始目标基实例化已知类本身的特殊孩子，但并不希望外部客户这样做。原始目标基也没有一般的几何外形管理方法。

另外，内部函数在复合目标基及所有孩子上递归完成许多操作(例如：认识及破坏)，有孩子的原始目标基必须准备好自己为其孩子而完成递归操作。

目标基树是由几个内部函数操作的。例如，*XtRealizeWidget* 向树的下方遍历，并递归地认识所有弹出目标基及复合目标基的孩子。*XtDestroyWidget* 向树的下方遍历并破坏所有弹出的目标基及复合目标基的孩子。取资源和修改资源函数向树的上方遍历并且从目标基的祖先决定资源的孩子。*XtMakeGeometryRequest* 遍历到该树的上一级并调用负责目标基孩子的几何外形的几何外形管理程序。

为了便于向目标基树的上方遍历，每一个目标基都有一个指向父目标基的指针。然而，*XtAppCreateShell* 返回外壳目标基具有零(NULL)父指针。

为了便于向目标基树的下方遍历，每一个复合目标基有到孩子对象阵列的指针，其中包含所产生的所有常规的孩子，而不是由复合目标基的几何外形管理程序所管理的孩子子集。实例化孩子的原始目标基完全负责自己以下向下遍历所要求的所有操作。另外，每一个目标基都有弹出孩子目标基阵列的指针。

1.2.1 初始化 X 工具包

在应用程序能调用内部函数之前，必须用以下几个方式初始化 X 工具包：

- *XtToolkitInitialize* 初始化 X 工具包。
- *XtCreateApplicationContext* 初始化每个应用的状态。
- *XtDisplayInitialize* 或者 *XtOpenDisplay* 初始化每个显示状态。
- *XtAppCreateShell* 产生初始目标基。

或者调用过程 *XtAppInitialize* 把前边的几个函数组合在一起来完成初始化工作。

在一个地址空间中可以完成 X 工具包应用程序的多个实例。要求每一个实例都能单独读输入及发送其他实例中的事件。另外，一个应用程序可能需要多个显示连接或者在多个屏幕上有目标基。为了适应这两种要求，该内部函数定义一些应用文本，每一种文本都提供把应用实例相互区分开来所要求的信息。应用文本的主要部件是那个应用程序的 X 显示指针表。应用文本类型 *XtAppContext* 不传给客户。

对于有关的讨论，请参见以下 X 例程、原型过程及方法：

<i>XtAddCallback</i> (1)	<i>XtDisplay</i> (1)	<i>XtVaAppCreateShell</i> (1)
<i>XtAppCreateShell</i> (1)	<i>XtIsRealized</i> (1)	<i>XtVaAppInitialize</i> (1)
<i>XtAppInitialize</i> (1)	<i>XtMergeArgLists</i> (1)	<i>XtVaCreateArgsList</i> (1)
<i>XtAppSetFallbackResources</i> (1)	<i>XtName</i> (1)	<i>XtVaCreateWidget</i> (1)
<i>XtCloseDisplay</i> (1)	<i>XtOpenDisplay</i> (1)	<i>XtWidgetToApplicationContext</i> (1)
<i>XtCreateApplicationContext</i> (1)	<i>XtParent</i> (1)	<i>XtWindowOfObject</i> (1)