

附录C Gtk+/Gnome对象总览

本书所介绍的对象只是 Gtk+/Gnome对象中的一部分。还有很多有趣的内容因为篇幅的原因没有涉及。同时，Gtk+/Gnome也是在不断发展的。有一些构件现在使用得很广泛，也许今后版本会推出一个功能更多，更稳定的新构件。有些构件现在还是实验性的，今后也许会变成正式的构件。

本附录是Gtk+和Gnome对象层次的快速教程。它包含 Gtk+和Gnome库中的GtkObject对象和它所有的子类，每个对象有一个简要的描述以及该对象的头文件。这些有助于为特定的任务选用合适的构件。

每个对象都列出了它的头文件。不过在程序中包含 `gtk.h`就可以包含所有的 Gtk+头文件，使用`gnome.h`头文件就可以包含所有的其他头文件。

一些对象被描述为“抽象基类”意指只有该对象子类才能实例化，但是所有的子类都可以由它的基类接口来操纵。

作为通行的规则，应该尽可能使用最有针对性的对象。也就是，可以用一个 `GtkWindow`作为应用程序的主窗口，但是`GnomeApp`才是更好的选择。可以用`GnomeDialog`做一个“关于”对话框，但是最好还是使用`GnomeAbout`。这能够最大程度保证用户界面的一致性，也可以省一些事。

Gtk+和Gnome都带一个“test”程序，分别称为`testgtk`和`test-gnome`。这些程序用于测试每个库中的构件和其他的特性。它们也是优秀的示例代码来源，同时还是一个浏览可用构件并选用合适构件的好方法。

C.1 GtkObject

库：Gtk+

头文件：`gtk/gtkobject.h`

描述：GtkObject是Gtk+的对象层次的基础。它不是一个图形化的组件。它实现了引用数、连接键/值对到对象上，以及对象解构(按C++术语，“虚解构的函数”)等接口。GtkObject本身在Gtk+对象系统中起着很重要的作用。Gtk+的信号/回调函数的基础结构是工作在GtkObject对象上的，也就是，信号是由特定的GtkObject对象引发的，回调函数连接到特定的对象和信号。

C.2 构件

构件也是Gtk+之所以存在的理由。构件是`GtkWidget`的子类，`GtkWidget`是`GtkObject`的子类。一个构件代表了屏幕上的一个矩形区域，它也许是纯粹装饰性的，交互式的控件，或者是一个控制子构件排列的容器。

1. `GtkWidget`

库：Gtk+

头文件：gtk/gtkwidget.h

描述：GtkWidget是所有构件的父类。GtkWidget是一个抽象基类。

2. GtkContainer

库：Gtk+

头文件：gtk/gtkcontainer.h

描述：是能包含其他构件的抽象基类。

3. GtkBin

库：Gtk+

头文件：gtk/gtkbin.h

描述：GtkBin是只能包含一个子构件的容器的抽象基类。它提供了 GtkContainer接口的缺省实现方法，所以为它创建子类是很容易的。

4. GtkWindow

库：Gtk+

头文件：gtk/gtkwindow.h

描述：GtkWindow代表一个顶级对话框和应用程序窗口。作为 Gtk+中的主要顶级构件，它有很多特殊的职责，例如，它维护了当前的键盘焦点并决定它自己的尺寸分配（而不是从它的父构件接收）。典型情况下，Gnome应用程序用GnomeApp构件作为应用程序的主窗口，充分利用它的附加功能。对对话框，在Gnome应用程序中应该使用GnomeDialog，对Gtk+应用程序使用GtkDialog构件。当然还有几个特制的对话框子类可以使用。如果没有合适的GtkWindow子类，可以直接使用GtkWindow窗口。

警告：如果接收到“delete_event”信号，GtkWindow会被自动销毁。要防止这种情形发生，必须设置一个新的要运行的信号处理程序，并且信号处理程序必须返回TRUE。这是很常见的Gtk+程序错误。GnomeDialog构件将有助于处理这种情况。

让GtkWindow作为最后调用gtk_widget_show()显示的构件是一个好主意。大多数构件直到它们的父构件容器映射到屏幕(放到屏幕上)才会实际映射到屏幕上。但是GtkWindow没有父构件，它会立即出现在屏幕上。所以如果在窗口显示之后再显示它的子构件，将会看到屏幕闪烁。

5. GnomeDialog

库：Gnome

头文件：libgnomeui/gnome-dialog.h

描述：在Gnome应用程序中所有的对话框都应该使用GnomeDialog(或其子类)构件。如果没有使用Gnome，GnomeDialog构件还是很有用的，因为它真正实现了一个对话框必须有的所有基本特性。强烈建议研究GnomeDialog的源代码(如果应用程序是基于GPL的，甚至还可以剪切和粘贴其中的代码)。

6. GnomeAbout

库：Gnome

头文件：libgnomeui/gnome-about.h

描述：GnomeAbout是一个“关于...”对话框，与Gnome的about对话框外观是一致的。

7. GnomeMessageBox

库：Gnome

头文件：libgnomeui/gnome-messagebox.h

描述：GnomeMessageBox是一个简单的GnomeDialog，里面预先组装了一个标签和一个小图标。图标对应于一个“信息框类型”，比如一个警告消息、一个错误、一个问题。这个图标让用户快速判定对话框的目的。

8. GnomePropertyBox

库：Gnome

头文件：libgnomeui/gnome-propertybox.h

描述：GnomePropertyBox是一个对话框，可以在程序中用于设置应用程序参数选择值或一些用户可见对象的属性值。它有“apply”、“OK”、“Close”以及“Help”几个按钮，“OK”按钮等价于按“Apply”后再按下“Close”按钮。

9. GnomeScores

库：Gnome

头文件：libgnomeui/gnome-scores.h

描述：GnomeScores用于跟踪和显示所有高分表。大多数Gnome游戏都使用这个构件。

10. GnomeApp

库：Gnome

头文件：libgnomeui/gnome-app.h

描述：GnomeApp是一个专门用作应用程序主窗口的GtkWidget子类。它为可选的工具条、菜单条和状态条留有空间。应用程序的“文档”放在构件中央的特别区域。

11. GtkDialog

库：Gtk+

头文件：gtk/gtkdialog.h

描述：GtkDialog是一个GtkWindow窗口，带有三个预置的构件：一个GtkVBox用于显示对话框内容，一个GtkSeparator，以及一个GtkHBox用于放置对话框的按钮。GtkDialog用处不大，所有的Gnome程序都应该使用GnomeDialog。

12. GnomeFontSelector

库：Gnome

头文件：libgnomeui/gnome-font-selector.h

描述：GnomeFontSelector是一个已经过时的字体选择对话框，现已被GtkFontSelectionDialog构件取代。GtkFontSelectionDialog包含一个GtkFontSelection。Gnome程序应该在GnomeDialog中放一个GtkFontSelection构件作为字体选择对话框，因为GtkFontSelectionDialog并不使用GnomeDialog，因而它的外观与Gnome风格有所不同。不应该使用GnomeFontSelector。

13. GtkInputDialog

库：Gtk+

头文件：gtk/gtkinputdialog.h

描述：GtkInputDialog是一个对话框，用于选择和设置使用X输入扩展的设备（例如绘图板）。它先于Gnome写成，没有使用GnomeDialog，所以在Gnome应用程序中显得有点滑稽，但是如果不重写该构件，又不用这个构件，就没有合适的工具。

14. GtkColorSelectionDialog

库：Gtk+

头文件：gtk/gtkcolorsel.h

描述：GtkColorSelectionDialog是一个对话框，包含一个GtkColorSelection。Gnome应用程序应该在GnomeDialog对话框中手工放置一个GtkColorSelection，或使用GnomeColorPicker。

15. GtkFileSelection

库：Gtk+

头文件：gtk/gtkfilesel.h

描述：GtkFileSelection是一个文件选择对话框（与大多数其他以“selection”结尾的构件不一样，其他组合构件意味着放在一个对话框内）。不幸的是，对这个构件还没有Gnome替代品，所以Gnome应用程序使用它，尽管它的外观有点不太连贯。

16. GtkFontSelectionDialog

库：Gtk+

头文件：gtk/gtkfontsel.h

描述：GtkFontSelectionDialog是一个对话框，包含一个GtkFontSelection。Gnome应用程序应该用包含一个GtkFontSelection的GnomeDialog对话框构件，而不是用这个构件。

17. GtkPlug

库：Gtk+

头文件：gtk/gtkplug.h

描述：GtkPlug是一个顶级窗口，可以嵌入到一个在其他应用程序中运行的GtkSocket构件中。换句话说，GtkSocket构件是进程中的一个“洞”，能包含一个可以运行其他程序的GtkPlug构件。

18. GtkButton

库：Gtk+

头文件：gtk/gtkbutton.h

描述：GtkButton是一个简单的矩形按钮。它是能包含一个子构件的容器；通常它包含一段文本或像素映射图片，但是它能包含所有构件。

19. GnomeColorPicker

库：Gnome

头文件：libgnomeui/gnome-color-picker.h

描述：GnomeColorPicker是一个按钮，包含一个小彩色方块指示当前选中的颜色。当点击按钮时，它创建一个颜色选择对话框，用来改变颜色。

20. GnomeFontPicker

库：Gnome

头文件：libgnomeui/gnome-font-picker.h

描述：GnomeFontPicker与GnomeColorPicker类似，它是一个按钮，上面显示当前选中的字体，当点击它时，将弹出一个字体对话框。

21. GnomeHRef

库：Gnome

头文件：libgnomeui/gnome-href.h

描述：GnomeHRef是一个无边框的按钮，在上面显示一个超级链接。点击用户时，Gnome指向用户的浏览器，连接到超级链接的目标 URL，或者启动一个新的浏览器实例。用于连接到指定URL的命令可以在Gnome控制中心配置。

22. GtkToggleButton

库：Gtk+

头文件：gtk/gtktogglebutton.h

描述：GtkToggleButton看起来更像一个正常的GtkButton按钮。然而，它倾向于表现一个可切换的状态。当按钮是“活动”的时，按钮是按下去的。通常应该使用GtkCheckButton而不是GtkToggleButton，GtkCheckButton经常比GtkToggleButton看起来更好，并且能给用户一个更好的概念：按钮代表一种可切换状态。

23. GtkCheckButton

库：Gtk+

头文件：gtk/gtkcheckbutton.h

描述：GtkCheckButton和GtkToggleButton差不多，但是外观不一样(它看起来像一个左边有小按钮的标签)。在多数情况，GtkCheckButton是比GtkToggleButton更好的选择，因为切换按钮不能给用户一个视觉上的提示表明它是可切换状态。

24. GtkRadioButton

库：Gtk+

头文件：gtk/gtkradiobutton.h

描述：GtkRadioButton代表几个互斥选项中的一个。GtkRadioButton是放在“组”里的。在一个组里，只有一个按钮可以是活动的。GtkOptionMenu也可以用于代表互斥的选项。如果选项较多，选项菜单可能是更好的选择。如果有大量的选项，GtkList或GtkCList也许更合适。

25. GtkOptionMenu

库：Gtk+

头文件：gtk/gtkoptionmenu.h

描述：GtkOptionMenu从许多选项中显示当前的活动选项，点击它时，将弹出一个菜单，允许用户设置新的活动选项。选项菜单构件还有点问题；当请求尺寸时，它不会考虑菜单项的尺寸。如果在选项菜单中包含标签，标签经常会被截短。最好的方法就是通过对它的父构件容器设置合适的选项，给它指定比实际需要更多的空间。

26. GnomeDockItem

库：Gnome

头文件：libgnomeui/gnome-dock-item.h

描述：GnomeDockItem是一个容器，让它的子构件显示在一个GnomeDock上。Dock items能够与它的父窗口分开，并放在桌面的任何地方。它们还可以在dock内移动。GnomeDock可以让用户重新排列工具条。GnomeDockItem提供了一个“手柄”，用于拖动它的子构件。GnomeApp构件在内部使用GnomeDock构件，这样，Gnome工具条可以重新定位。

27. GtkAlignment

库：Gtk+

头文件：gtk/gtkalignment.h

描述：GtkAlignment是一种不可见容器，用于在空间内对齐子构件。要设置两个因子，每个都要X和Y方向的值；对齐值0.0是左对齐(或顶部对齐)并且1.0是右对齐(或底部对齐)。对齐值为0.5时，子构件在对齐方向居中。比例因子决定子构件应该怎样填充它并不需要的额外空间。如果是0.0，子构件只占据它请求的值。如果是1.0，构件将占据所有的可用空间。(实际上，比例因子1.0 scale使alignment因子不起作用。)

28. GtkFrame

库：Gtk+

头文件：gtk/gtkframe.h

描述：框架构件在它的子构件周围画一个装饰性的框。它还可以有一个标题描述这个框中的内容。要关闭标题，将它设置为 NULL。

29. GtkAspectFrame

库：Gtk+

头文件：gtk/gtkaspectframe.h

描述：GtkAspectFrame用于控制它的子构件的纵横比。与 GtkAlignment相似，它也允许将子构件在两个方向对齐。可以指定一个纵横比，或要求保留子构件的尺寸请求的比率。视觉外观上GtkAspectFrame与GtkFrame完全一样。

30. GtkItem

库：Gtk+

头文件：gtk/gtkitem.h

描述：GtkItem是一个用于列表项、树数据项、菜单项的抽象基类。这种数据项是可以“选中”、“取消选择”和“切换”的构件。

31. GtkMenuItem

库：Gtk+

头文件：gtk/gtkmenuitem.h

描述：菜单项是一个不可见的容器，它是唯一一种能够显示在 GtkMenu上的构件。典型情况下，可以在菜单项上放置一个标签，或者一个标签加一个像素映射图片以指明它的功能。如果菜单项没有子构件，它将绘出一个分隔线。它节省了在菜单项上添加 GtkSeparator 构件的额外开销。

32. GtkCheckMenuItem

库：Gtk+

头文件：gtk/gtkcheckmenuitem.h

描述：GtkCheckMenuItem是一种菜单项，功能与 GtkCheckButton差不多，在它的子构件的左边有一个小按钮，它可以是“活动”的或者是“不活动”的。因为它是 GtkMenuItem的子类，它也可以出现在菜单中。

33. GtkRadioMenuItem

库：Gtk+

头文件：gtk/gtkradiomenuitem.h

描述：GtkRadioMenuItem与GtkRadioButton类似，它允许用户从一系列互斥选项中做出选择。因为它是GtkMenuItem的子类，它可以出现在菜单上。

34. GtkPixmapMenuItem

库：Gnome

头文件：libgnomeui/gtkpixmapmenuitem.h

描述：GtkPixmapMenuItem实际上是一个Gnome构件，尽管它的名字带“Gtk”。这个构件有一个非常特别的问题：当菜单包含检查菜单或无线菜单项时，Gtk会将所有菜单项的子构件缩进，为检查菜单和无线菜单项留出空间。Gnome在菜单旁边放一个像素映射图片；GtkPixmapMenuItem将图片缩进，与Gtk+缩进检查菜单和无线菜单一样。如果只添加一个图片和一个标签到GtkMenuItem中，与其他菜单项相比，图片与普通标签的对齐方式会不正确。这个构件不是专用于图片的；任何构件都可以放在未缩进的“图片”位置。通常，GtkPixmapMenuItem是用Gnome菜单创建函数隐含创建的。

35. GtkTearoffMenuItem

库：Gtk+

头文件：gtk/gtktearoffmenuitem.h

描述：GtkTearoffMenuItem是一个“穿孔”，代表一个点 - 在这里菜单可以被“拖开”（在一个顶级窗口中保持可见，便于用户访问）。缺省时，Gnome菜单都包含一个tearoff菜单，但是用户可以用Gnome控制中心完全禁用它们。

36. GtkListItem

库：Gtk+

头文件：gtk/gtklistitem.h

描述：GtkListItem是一个不可见的容器，它允许它的子构件显示在一个GtkList构件上。也就是，只有列表项能出现在列表中。

37. GtkTreeItem

库：Gtk+

头文件：gtk/gtktreeitem.h

描述：GtkTreeItem是一个不可见的容器，它允许子构件能显示在一个GtkTree上。只有树数据项能出现在树构件上。

38. GtkEventBox

库：Gtk+

头文件：gtk/gtkeventbox.h

描述：GtkEventBox构件也许是Gtk+中最简单的容器构件。它的唯一效果就是有一个GdkWindow窗口。某些操作只对有窗口的构件起作用（比如设置背景颜色或捕获事件）。如果想在无窗口的构件上执行这些操作，可以将构件放在一个事件盒中，然后在事件盒上执行需要的操作，能起到同样的效果。

39. GtkHandleBox

库：Gtk+

头文件：gtk/gtkhandlebox.h

描述：GtkHandlebox构件能够给它的子构件添加一个手柄。拖动手柄，子构件可以从窗口上移开，并放在用户桌面的其他地方。手柄盒一般用于工具条。 GnomeDock和GnomeDockItem提供了一个更灵活的(但是也更精巧)选择。

40. GtkScrolledWindow

库：Gtk+

头文件：gtk/gtkscrolledwindow.h

描述：GtkScrolledWindow为它的子构件提供了一个水平和垂直的滚动条。可选状态，当整个子构件都可见时，滚动条可以隐藏起来。如果子构件在它的 GtkWidgetClas中有 set_scroll_adjustments_signal，滚动窗口把它们当作滚动条的调整值。否则，滚动窗口用一个 GtkViewport 构件滚动整个构件。(例如GtkCList；列标题不会滚动，只有列表内容滚动。如此GtkCList提供了一个滚动调整信号。)

41. GtkViewport

库：Gtk+

头文件：gtk/gtkviewport.h

描述：GtkViewport或多或少是GtkScrolledWindow构件的一个实现细节。它包含了一个不能提供set_scroll_adjustments_signal 方法的构件，为子构件提供这样一个信号。要了解更多信息，参看GtkScrolledWindow。

42. GtkBox

库：Gtk+

头文件：gtk/gtkbox.h

描述：GtkBox是GtkVBox、GtkHBox和GtkButtonBox的抽象基类。它是不可见的布局容器。

43. GtkHBox

库：Gtk+

头文件：gtk/gtkhbox.h

描述：GtkHBox是一个GtkBox，将子构件从左往右组装。左边看作组装盒的开始。

44. GnomeAppBar

库：Gnome

头文件：libgnomeui/gnome-appbar.h

描述：GnomeAppBar是一个简单的状态条，带一个可选的进度条。它不像 GtkStatusbar有一个“上下文”。

45. GnomeDateEdit

库：Gnome

头文件：libgnomeui/gnome-dateedit.h

描述：GnomeDateEdit允许用户编辑日期和时间。如果仅仅对日期感兴趣，时间编辑部分可以关掉。

46. GtkCombo

库：Gtk+

头文件：gtk/gtkcombo.h

描述：GtkCombo是一个带下拉菜单“快速选择”的文本输入框。如果想限制用户只能选择一些固定的选项，GtkOptionMenu更合适。GtkCombo允许用户输入一些东西，但是还提供了一些建议值。GnomeEntry是一个类似于GtkCombo的组合框。它将用户输入的字符串添加到下拉列表中，并且在会话与会话之间能自动记住列表。

47. GnomeEntry

库：Gnome

头文件：libgnomeui/gnome-entry.h

描述：GnomeEntry是GtkCombo构件的扩展，将下拉菜单作为历史选项。如果用户输入一些在历史列表中还没有的东西，GnomeEntry将它添加进去，并保存在一个配置文件中，在应用程序下次启动时，能够再次加载。

48. GnomeFileEntry

库：Gnome

头文件：libgnomeui/gnome-file-entry.h

描述：GnomeFileEntry是一个GnomeEntry，将文件名保存在历史记录中。它有一个“浏览”按钮，点击时会弹出一个GtkFileSelection。它还有一个“只浏览目录”的模式。

49. GnomeNumberEntry

库：Gnome

头文件：libgnomeui/gnome-number-entry.h

描述：GnomeNumberEntry允许用户输入一个数字，它在它的下拉菜单中保存了一个曾经输入数字的历史记录。它还有一个“计算器”按钮，点击时会弹出一个GnomeCalculator。

50. GnomeProcBar

库：Gnome

头文件：libgnomeui/gnome-procbar.h

描述：GnomeProcBar用在Gnome面板小程序上，用来显示CPU和内存负载，还用在Gtop程序中(Gtop是top程序的图形化克隆)。它显示一个可变长度的着色长条，它还能用于显示任何类型的经常变化的值。

51. GtkStatusbar

库：Gtk+

头文件：gtk/gtk.h

描述：GtkStatusbar是一个状态条构件，它在窗口的底部显示一行文本。

52. GtkVBox

库：Gtk+

头文件：gtk/gtkvbox.h

描述：GtkVBox是一个GtkBox，将构件从顶部向底部组装。顶部被看作组装盒的“开始”。

53. GnomeCalculator

库：Gnome

头文件：libgnomeui/gnome-calculator.h

描述：GnomeCalculator是作为一个GtkWidget实现的简单计算器。

54. GnomeGuru

库：Gnome

头文件：libgnomeui/gnome-guru.h

描述：GnomeGuru试图实现一个“向导”构件（一系列的页面，代表用户能一步步完成的任务中某些步骤）。它还不成熟，不应该使用它。现在有一个新的构件，称为 GnomeDruide，可能会在 Gnome 的下一个版本中取代 GnomeGuru。在这之前，GnomeDruide 极有可能作为一个外接程序模块提供给用户，所以如果想要一个向导构件，可以找一下 GnomeDruide 构件。

55. GnomeIconEntry

库：Gnome

头文件：libgnomeui/gnome-icon-entry.h

描述：GnomeIconEntry 与 GnomeColorPicker 和 GnomeFontPicker 很相似。它是一个显示当前选中图标的按钮，还有一个图标浏览按钮，点击它可以设置新图标。它过去有一个文本输入框，用来输入图标文件名，所以它被称为 GnomeIconEntry 而不是 GnomeIconPicker。

56. GnomeIconSelection

库：Gnome

头文件：libgnomeui/gnome-icon-sel.h

描述：GnomeIconSelection 浏览图标文件，它用在 GnomeIconEntry 中，但是也可以直接使用。

57. GnomeLess

库：Gnome

头文件：libgnomeui/gnome-less.h

描述：GnomeLess 是 GtkText 构件的简单扩展，从一个文件或文件描述符加载一个文件并显示它。要尽量避免使用这个构件，因为它不是特别有用，所以它可能会从今后的 Gnome 版本中消失。

58. GnomePaperSelector

库：Gnome

头文件：libgnomeui/gnome-paper-selector.h

描述：GnomePaperSelector 是另一个要避免使用的构件，在 Gnome 1.0 发布版本中，它是严格的实验性构件。它允许用户选择打印纸张尺寸。

59. GnomePixmapEntry

库：Gnome

头文件：libgnomeui/gnome-pixmap-entry.h

描述：GnomePixmapEntry 与 GnomeIconEntry 构件的目的完全一样，它允许用户选择一个图片。喜欢某一个或另一个构件的唯一理由是 GnomeIconEntry 将图象按比例缩小到标准的 Gnome 图标大小。

60. GnomeSpell

库：Gnome

头文件：libgnomeui/gnome-spell.h

描述：GnomeSpell 是一个拼写检查接口，它在内部使用 ispell 程序。这个构件应该看作实

验性的，要避免在程序中使用。

61. GtkColorSelection

库：Gtk+

头文件：gtk/gtkcolorsel.h

描述：GtkColorSelection允许用户指定颜色，用一个轮形或滑块调色板来指定一种颜色。

它构成了GtkColorSelectionDialog的内容。

62. GtkGammaCurve

库：Gtk+

头文件：gtk/gtkgamma.h

描述：GtkGammaCurve 允许用户编辑一个曲线，它是专门用于 GIMP的构件。很少有应用程序会用到它。

63. GtkButtonBox

库：Gtk+

头文件：gtk/gtkbbox.h

描述：GtkButtonBox是一种特殊类型的GtkBox，用于放置对话框的按钮。它有水平和垂直两种变体。GnomeDialog会创建一个按钮盒，所以在Gnome下编程时，不需要直接使用这个构件。

64. GtkHButtonBox

库：Gtk+

头文件：gtk/gtkhbbox.h

描述：GtkHButtonBox是GtkButtonBox的水平变体。

65. GtkVButtonBox

库：Gtk+

头文件：gtk/gtk.h

描述：GtkVButtonBox是GtkButtonBox的垂直变体。

66. GtkLayout

库：Gtk+

头文件：gtk/gtklayout.h

描述：GtkLayout是一个容器，给人的感觉好象有无限的尺寸似的。因为X窗口最大可以是32 768像素(子构件必须放在父构件的X窗口上)。其他的构件都用简单移动它们的GdkWindow窗口来滚动。GtkLayout比这些构件更聪明些。

67. GnomeCanvas

库：Gnome

头文件：libgnomeui/gnome-canvas.h

描述：它绘制不闪烁的结构化图形，对定制的显示也是非常理想的。

68. GnomeIconList

库：Gnome

头文件：libgnomeui/gnome-icon-list.h

描述：GnomeIconList用在Gnome文件管理器。它显示图标和它们的名称，用户能够通过

拖动鼠标选择图标组。

69. GnomeDockBand

库：Gnome

头文件：libgnomeui/gnome-dock-band.h

描述：GnomeDockBand包含一行或一列GnomeDockItem。GnomeDock也包含了一个或多个GnomeDockBand。

70. GnomeDock

库：Gnome

头文件：libgnomeui/gnome-dock.h

描述：GnomeDock里面包含GnomeDockBand，而GnomeDockBand里面包含GnomeDockItem。GnomeDock允许用户将工具条和应用程序的其他部件重新定位。

71. GtkCList

库：Gtk+

头文件：gtk/gtkclist.h

描述：GtkCList是一个多列的列表构件；它也是GtkCTree的基类。GtkCList在每个单元格中显示文本和/或图片；单元格里不能容纳子构件。GtkCList仅仅是一个容器，因为它使用构件作为它的栏标题。你可能更愿意使用GtkList，因为在它的列表项中能够使用子构件，但是它效率较低，列表项最大只能有32 768个像素。

72. GtkCTree

库：Gtk+

头文件：gtk/gtkctree.h

描述：GtkCTree与GtkCList类似，但是显示一个可展开的节点而不是显示一个简单的列表。GtkTree是更灵活的树构件（在树单元里可以是任意类型的构件），但是它效率较低，只能有32 768个像素大小。

73. GtkFixed

库：Gtk+

头文件：gtk/gtkfixed.h

描述：GtkFixed容器允许用户将子构件放在绝对坐标上，并且给子构件所需要的尺寸大小。最好还是用其他的布局容器构件。

74. GtkNotebook

库：Gtk+

头文件：gtk/gtknotebook.h

描述：笔记本构件为用户显示“多页”内容。用户可以通过选择它的“页标签”在“页”间移动。每个添加到GtkNotebook上的子构件都成为一页，还可以在页标签上使用其他的构件。通常具有一行以上的页标签是一种很糟糕的做法，但是它确实允许这么做。还可以将标签放在构件的左边、右边或下边，但是为界面连贯性起见，最好还是将它放在构件的顶部。

75. GtkFontSelection

库：Gtk+

头文件：gtk/gtkfontsel.h

描述：GtkFontSelection是一个组合构件，允许用户从系统可用的字体中选择一种字体。GtkFontSelectionDialog里面包含了GtkFontSelection的一个实例。Gnome应用程序应该在一个GnomeDialog中放一个GtkFontSelection构件，而不是使用GtkFontSelectionDialog构件。

76. GtkPaned

库：Gtk+

头文件：gtk/gtkpaned.h

描述：GtkPaned构件将一个区域划分为两个用户可调整大小的部分。它有水平和垂直两个变体。

77. GtkHPaned

库：Gtk+

头文件：gtk/gtkhpaned.h

描述：GtkHPaned是GtkPaned的水平变体，它将一个区域划分为左右两个区域。

78. GtkVPaned

库：Gtk+

头文件：gtk/gtkvpaned.h

描述：GtkVPaned是GtkPaned构件的垂直变体，它将一个区域划分为上下两个部分。

79. GtkList

库：Gtk+

头文件：gtk/gtklist.h

描述：GtkList显示一个项目列表。每个项目称为一个GtkListItem，GtkListItem是一个可以包含任何种类构件的容器。GtkList的大小是有限的，因为它的列表项是放在列表的GdkWindow中的固定坐标上的，滚动是通过移动GdkWindow做到的。GdkWindow最大可以是32 768像素，任何超过这个范围的列表项都是不可见的。GtkCList构件克服了这个限制，但是不能包含任意种类的构件。

80. GtkMenuShell

库：Gtk+

头文件：gtk/gtkmenushell.h

描述：GtkMenuShell是一个内部包含GtkMenuItem的构件的抽象基类。它的两个子类是GtkMenu和GtkMenuBar。

81. GtkMenuBar

库：Gtk+

头文件：gtk/gtkmenubar.h

描述：GtkMenuBar是一个菜单条。它包含一个或多个菜单项，通常，每个菜单项都包含一个子菜单（也就是，一个GtkMenu带多个菜单项）。例如，菜单条也许有一个菜单项叫“文件”，该菜单项包含子菜单项“打开”和“退出”。

82. GtkMenu

库：Gtk+

头文件：gtk/gtkmenu.h

描述：GtkMenu包含菜单项。因为它不会由程序员显示出来（用Ggtk_widget_show()），所

以它是独一无二的。菜单会响应用户的动作弹出来。

83. GtkPacker

库：Gtk+

头文件：gtk/gtkpacker.h

描述：GtkPacker是一个布局容器构件，是受到 Tk工具包启发而创建的。如果你对 Tk工具包很熟悉，就会发现它比标准的 Gtk+布局容器更容易使用。

84. GtkSocket

库：Gtk+

头文件：gtk/gtksocket.h

描述：GtkSocket是在一个应用程序中的“洞”，它允许来自与另一个应用程序的 GtkPlug构件嵌入到里面。

85. GtkTable

库：Gtk+

头文件：gtk/gtktable.h

描述：GtkTable是最重要的Gtk+布局构件之一。

86. GtkTed

库：Gnome

头文件：libgnomeui/gtk-ted.h

描述：GtkTed是Gnome早期遗留物。“ted”意为“表格编辑器”。它是一种原始的GUI生成器。现在至少有两个GUI生成器项目(Glade和GLE)正在开发，所以可以忽略这个构件，它也会在libgnomeui库今后的版本中消失。

87. GtkToolbar

库：Gtk+

头文件：gtk/gtktoolbar.h

描述：GtkToolbar是一个工具条构件。通常Gnome应用程序应该使用Gnome的helper函数而不是直接创建工具条。

88. GtkTree

库：Gtk+

头文件：gtk/gtktree.h

描述：GtkTree与GtkCTree的关系和GtkList与GtkCList的关系一样。也就是，GtkTree比GtkCTree更灵活(tree items可以包含任何构件)但是也比GtkCTree更慢，只能容纳有限数量的项。Item的最大数目依赖于行高，但总的行高必须能容纳在一个GdkWindow内，也就是32 768个像素。

89. GnomeAnimator

库：Gnome

头文件：libgnomeui/gnome-animator.h

描述：GnomeAnimator显示一系列图像，生成一个动画。它有一个“循环模式”和一个“播放一次”模式。GnomeAnimator API在Gnome 1.0中标志为“不成熟”，在未来的版本中可能会变得不兼容；它是一个实验性的构件。

90. GnomePixmap

库：Gnome

头文件：libgnomeui/gnome-pixmap.h

描述：GnomePixmap应该是比GtkPixmap更好的构件，虽然它与后者有同样的作用。

GnomePixmap在视觉上更灵活。它还使用Imlib可以加载多种不同格式的图像；所以使用它更方便。

91. GnomeStock

库：Gnome

头文件：libgnomeui/gnome-stock.h

描述：GnomeStock是一个封装了GnomePixmap的容器。它自动地创建pixmap的“不敏感”和“具有焦点”拷贝，以反映构件的状态。它可以设置为一个Gnome内置的pixmap宏，比如GNOME_STOCK_PIXMAP_CUT(小剪刀图片)，GNOME_STOCK_PIXMAP_PRINT(一个小打印机)等。还可以在运行时注册一个新的内置的与应用程序图片。Gnome在用GnomeUIInfo创建菜单和工具条时，在内部使用这个构件。

92. GtkMisc

库：Gtk+

头文件：gtk/gtkmisc.h

描述：GtkMisc抽象基类用来设置它的子类的“对齐值”和“填充量”。对齐值是介于0.0和1.0之间的浮点数，0.0是左对齐，1.0是右对齐，0.5是居中对齐。只有在GtkMisc构件接收到比它所请求更多的尺寸分配时才管用。对齐值将构件的自然约束框在它实际约束框中定位。填充量是增加到构件尺寸请求中的像素值。构件让这些像素值的地方空着。

不是从GtkMisc中派生的构件也可以变成“可对齐的”，只需将构件放在一个GtkAlignment容器中就可以了。

93. GtkLabel

库：Gtk+

头文件：gtk/gtklabel.h

描述：GtkLabel简单地显示一个文本串。如果文本串包含新行，GtkLabel显示多行。GtkLabel标签从它的父类(GtkMisc)中取得对齐参数，可以用这个参数让文本居中，或将它向左或向右移动。对齐与版面调整不一样。版面对齐根据彼此的内容调整多行文本的位置。版面左对齐意味着每一行从同样的位置开始，版面右对齐意味着每一行都在同样的地方终止，版面居中对齐意味着每一行都以一条假象的线居中对齐。版面对齐对只有一行的标签没有任何意义。对齐用gtk_misc_set_alignment()函数设置整块文本在所分配的空间中的位置。对齐只在标签接收到比它请求更多的空间时才起作用(它请求足够的空间以容纳文本块)。GtkLabel标签是GTK_NO_WINDOW类型的构件(这意味着它们不会接收到事件，直接绘制在父构件的背景上)。

94. GtkAccelLabel

库：Gtk+

头文件：gtk/gtkaccellabel.h

描述：GtkAccelLabel与其他构件相关联，作为标签的一部分，为该构件显示加速键。

95. GtkClock

库：Gnome

头文件：libgnomeui/gtk-clock.h

描述：GtkClock是一个标签，用来显示时间。它可以周期性地更新时间（就像时钟一样！）。

96. GtkTipsQuery

库：Gtk+

头文件：gtk/gtktipsquery.h

描述：GtkTipsQuery是一个显示工具提示的标签。它还有一个“ What 's This? ”功能。调用gtk_tips_query_start()函数切换到“ 查询模式 ”。在查询模式中，光标变成一个问号；当鼠标在应用程序的构件上面移动时， GtkTipsQuery显示这些构件的工具提示。如果用户点击构件，GtkTipsQuery引发一个“ widget_selected ”信号，可以用它来为构件显示更广泛的帮助信息。可以在tkTooltip的“ 私有 ”组件中存储更广泛的帮助信息，或用 gtk_object_set_data()函数来存储一些应用程序相关的信息。

97. GtkArrow

库：Gtk+

头文件：gtk/gtkarrow.h

描述：GtkArrow显示一个箭头。它是一种简单的 GTK_NO_WINDOW构件不能接受事件。在缺省主题中，箭头仅仅是一个三角形。

98. GtkImage

库：Gtk+

头文件：gtk/gtkimage.h

描述：GtkImage在一个构件中显示一个GdkImage。它只在已经有了一个GdkImage时才有用；要显示一个确定的图像，应该使用 GnomePixmap。

99. GtkPixmap

库：Gtk+

头文件：gtk/gtkpixmap.h

描述：GtkPixmap显示一个GdkPixmap。如果正在使用Gnome，建议使用GnomePixmap构件。理由在GnomePixmap的介绍中已有说明。

100. GtkCalendar

库：Gtk+

头文件：gtk/gtkcalendar.h

描述：GtkCalendar显示一个日历页(一个月)。它允许用户选择一个日期。 GnomeDateEdit构件在弹出菜单中使用GtkCalendar构件。

101. GtkDrawingArea

库：Gtk+

头文件：gtk/gtkdrawingarea.h

描述：GtkDrawingArea是GdkWindow的一个“ 瘦 ”封装。它提供一个空白区域，可用于绘画。正常情况，应该连接到它的“ configure_event ”信号，以捕获区域的大小变化，在一个

“expose_event”信号处理程序中实现绘画。要消除闪烁，可以保留一个与绘画区尺寸相等的GdkPixmap，画到GdkPixmap上。在“expose_event”处理程序中，简单将暴露的区域从GdkPixmap复制到绘画区。对高级图形绘画，GnomeCanvas可能是更好的选择。

102. GtkCurve

库：Gtk+

头文件：gtk/gtkcurve.h

描述：GtkCurve用于在GtkGammaCurve构件中显示曲线。它是一种带曲线会话能力的绘图区扩展。不大可能会用到这个构件。

103. GtkDial

库：Gtk+

头文件：gtk/gtkdial.h

描述：GtkDial是一种“里程计”显示。缺省时，用户可以拖动指针旋转，改变标度盘的值。GtkDial还有一种“仅供观察”模式。

104. GtkEditable

库：Gtk+

头文件：gtk/gtkeditable.h

描述：GtkEditable是允许用户编辑文本的构件的抽象基类。基类接口允许光标定位，获取字符等待。它还包含一个“changed”信号，可用于监测用户的输入。

105. GtkEntry

库：Gtk+

头文件：gtk/gtkentry.h

描述：GtkEntry允许用户输入一行文本。它有一种“口令”模式，在“口令”模式中，输入框中的文本被星号“*”替代。

106. GtkSpinButton

库：Gtk+

头文件：gtk/gtkspinbutton.h

描述：GtkSpinButton是一个定制的GtkEntry，允许用户输入一个数字。Spin buttons给输入框增加了向上和向下两个按钮，用户可以在可能取值之间快速调整。

107. GtkText

库：Gtk+

头文件：gtk/gtktext.h

描述：GtkText是一个文本构件。它能显示文本，并且还提供简单的文本编辑实用程序。

108. GtkRuler

库：Gtk+

头文件：gtk/gtkruler.h

描述：GtkRuler是水平标尺和垂直标尺构件的抽象基类。标尺构件在GIMP中用于显示图象的尺寸。

109. GtkHRuler

库：Gtk+

头文件：gtk/gtkhruler.h

描述：GtkRuler的水平变体。

110. GtkVRuler

库：Gtk+

头文件：gtk/gtkvruler.h

描述：GtkRuler的垂直变体。

111. GtkRange

库：Gtk+

头文件：gtk/gtkrange.h

描述：GtkRange是“滑块类”构件的一个抽象基类。这些构件当“滑块”在“滑槽”内移动时修改某些数字值。它的两个子类是 `GtkScale`，用于让用户输入一个数值，和 `GtkScrollbar`，Gtk+的滚动条构件。

112. GtkScale

库：Gtk+

头文件：gtk/gtkscale.h

描述：GtkScale允许用户通过移动一个滑块来输入数字值。它能在滑块的上面显示当前值；如果精确值没有什么关系，可以将这个特性关闭，或者，也可以以其他方法提供反馈。GtkScale是一个抽象基类，必须使用它的垂直或水平变体。

113. GtkHScale

库：Gtk+

头文件：gtk/gtkhscale.h

描述：GtkHScale是GtkScale的水平变体。

114. GtkVScale

库：Gtk+

头文件：gtk/gtkvscale.h

描述：GtkVScale是垂直的GtkScale。

115. GtkScrollbar

库：Gtk+

头文件：gtk/gtkscrollbar.h

描述：GtkScrollbar是一个抽象基类，为垂直和水平滚动条提供公用接口。

116. GtkHScrollbar

库：Gtk+

头文件：gtk/gtkhscrollbar.h

描述：水平滚动条。

117. GtkVScrollbar

库：Gtk+

头文件：gtk/gtkvscrollbar.h

描述：垂直滚动条。

118. GtkSeparator

库：Gtk+

头文件：gtk/gtkseparator.h

描述：GtkSeparator构件是一分隔线，可以让用户界面更有吸引力。例如，用GtkHSeparator将GnomeDialog对话框的内容区与它的按钮分隔开。

119. GtkHSeparator

库：Gtk+

头文件：gtk/gtkhseparator.h

描述：水平的GtkSeparator。

120. GtkVSeparator

库：Gtk+

头文件：gtk/gtkvseparator.h

描述：垂直的tkSeparator。

121. GtkPreview

库：Gtk+

头文件：gtk/gtkpreview.h

描述：GtkPreview显示一个RGB图像，GIMP用它以显示一个图像转换的预览效果。

122. GtkProgress

库：Gtk+

头文件：gtk/gtkprogress.h

描述：GtkProgress是进度显示的抽象基类。在 Gtk+ 1.2 中它只有一个具体的子类(GtkProgressBar)。未来的Gtk+版本也许会增加其他的进度构件。

123. GtkProgressBar

库：Gtk+

头文件：gtk/gtkprogressbar.h

描述：GtkProgressBar是一个灵活的进度条构件。除进度条功能以外，还可以在上面显示文本，还有一个“活动”模式以指示“正在活动，但不知道任务的大小”。在“活动”模式中，一个小滑块会前后移动。GtkProgressBar是可配置的，为了与其他应用程序保持一致，应该试一下它的缺省外观和感觉。

124. ZvtTerm

库：Zvt

头文件：zvt/zvtterm.h

描述：ZvtTerm是随gnome-libs发布的，但是它并不在 libgnomeui中。它是一个单独的libzvt库。ZvtTerm只是一个终端仿真器；可以产生一个子进程，在构件内运行，并与用户交互。ZvtTerm提供与Gnome桌面环境中的gnome-terminal程序具有的全部功能。

C.3 画布项

1. GnomeCanvasItem

库：Gnome

头文件：libgnomeui/gnome-canvas.h

描述：GnomeCanvasItem是画布项的抽象基类。

2. GnomeCanvasRE

库：Gnome

头文件：libgnomeui/gnome-canvas-rect-ellipse.h

描述：GnomeCanvasRE是矩形和椭圆画布项的抽象基类。今后，它还会是GnomeCanvasArc项的基类。

3. GnomeCanvasEllipse

库：Gnome

头文件：libgnomeui/gnome-canvas-rect-ellipse.h

描述：GnomeCanvasEllipse在画布上绘制一个椭圆。

4. GnomeCanvasRect

库：Gnome

头文件：libgnomeui/gnome.h

描述：GnomeCanvasRect在画布上绘制一个矩形。

5. GnomeCanvasGroup

库：Gnome

头文件：libgnomeui/gnome-canvas.h

描述：GnomeCanvasGroup是一个包含其他GnomeCanvasItem的GnomeCanvasItem。它创建一个GnomeCanvas中画布项的分层次树状结构。GnomeCanvas创建一个成为“根”的特殊的GnomeCanvasGroup，所有用户创建的画布项加到“根”组。

6. GnomeCanvasImage

库：Gnome

头文件：libgnomeui/gnome-canvas-image.h

描述：GnomeCanvasImage在画布上显示一幅图片(更明确地说，是一幅GdkImlibImage)。

7. GnomeCanvasLine

库：Gnome

头文件：libgnomeui/gnome.h

描述：GnomeCanvasLine在画布上显示一条线段或一系列的线段；它还能将最后一根线的终点与第一根线的起点连接起来，显示一个未填充的多边形。在线的两端还可以带箭头。

8. GnomeCanvasPolygon

库：Gnome

头文件：libgnomeui/gnome-canvas-polygon.h

描述：GnomeCanvasPolygon显示一个填充的多边形。用GnomeCanvasLine画中空的多边形。

9. GnomeCanvasText

库：Gnome

头文件

libgnomeui/gnome-canvas-text.h

描述：GnomeCanvasText在画布上显示一些文本。

10. GnomeCanvasWidget

库：Gnome

头文件：libgnomeui/gnome-canvas-widget.h

描述：GnomeCanvasWidget模拟一个GtkContainer构件。它容纳一个子构件，并将它显示在画布上。

11. GnomeCanvasTextItem

库：Gnome

头文件：libgnomeui/gnome-icon-item.h

描述：GnomeCanvasTextItem用于GnomeIconList构件内部。不应该直接使用它，它被当作一个实现细节，在今后Gnome库的版本中可能会变化。

C.4 其他对象

1. GnomeClient

库：Gnome

头文件：libgnomeui/gnome-client.h

描述：GnomeClient是一个GtkObject对象，它隐蔽了会话管理的细节，并为Gnome应用程序提供了很好的会话管理API。

2. GnomeDEntryEdit

库：Gnome

头文件：libgnomeui/gnome-dentry-edit.h

描述：GnomeDEntryEdit是一个非常特别的对象。它是一种“构件管理器”，它创建并监视两个子构件：一个“简单”的页面，一个“高级”的页面。这两页结合起来让用户编辑一个Gnome的.desktop文件。GnomeDEntryEdit倾向于与已有的GtkNotebook联合使用。它本身并不是GtkNotebook的子类，因为允许会在GnomePropertyBox中使用GtkBoteBook构件。Gnome面板和Gnome菜单编辑器使用这个构件。

3. GnomeDockLayout

库：Gnome

头文件：libgnomeui/gnome-dock-layout.h

描述：GnomeDockLayout维护在GnomeDock中的构件的当前位置信息。它还能够加载和保存这个信息，GnomeDock使用这个功能来保存和加载工具条的位置。GnomeApp使用GnomeDock构件来排列它的布局。

4. GnomeMDIChild

库：Gnome

头文件：libgnomeui/gnome-mdi-child.h

描述：GnomeMDIChild是一个抽象接口。必须派生新的子类，或者为了利用GnomeMDI对象，使用GnomeMDIGenericChild构件。GnomeMDI是一个“多文档接口”管理器。

5. GnomeMDIGenericChild

库：Gnome

头文件：libgnomeui/gnome-mdi-generic-child.h

描述：GnomeMDIChild构件的一般性的实现。对较复杂的应用程序，也许需要重新写一个定制的GnomeMDIChild实现。

6. GnomeMDI

库：Gnome

头文件：libgnomeui/gnome-mdi.h

描述：GnomeMDI跟踪多个文档。用户能够配置如何在应用程序中排列文档；它们也许被放在同一个GtkWindow中的GtkNotebook上，或者每个文档都有自己的GtkWindow窗口。还有，用户也能将文档（“笔记本页”）拖出GtkWindow，它们会被放置在自己的顶级窗口中。

7. GtkData

库：Gtk+

头文件：gtk/gtkdata.h

描述：GtkData是可为多个对象所共享的数据片断的抽象基类。现在它的接口还是空的，今后，对所有GtkData对象，也许会有一些通用的操作函数。

8. GtkAdjustment

库：Gtk+

头文件：gtk/gtkadjustment.h

描述：GtkAdjustment代表一个数值。它还存储一个最大值和一个最小值，一个“单步增量”，一个“单页增量”和一个“页面值”。一些对象并不使用在GtkAdjustment中的所有的这些值，一些对象对这些值的解释也略有不同。GtkRange构件(包括它的GtkScale和GtkScrollbar子类)允许用户将滑块在最大值和最小值之间移动。点击GtkScrollbar构件端部的箭头让滚动条推动一个“单步增量”，用鼠标点击它们会将滚动条移动一个“单页增量”。“页面值”决定滚动条滑块的尺寸大小(就是在整个范围内的同一单元上可见页的长度)。当GtkAdjustment的值改变时，引发一些信号。

9. GtkTooltips

库：Gtk+

头文件：gtk/gtktooltips.h

描述：GtkTooltips将构件与一些帮助文本联系起来。如果将鼠标指针在构件上方做短暂停留，工具提示会出现，显示帮助文本。GtkTooltips也能存储一些“私有”文本；可以将它与GtkTipsQuery一起使用来显示扩展的帮助信息。

10. GtkItemFactory

库：Gtk+

头文件：gtk/gtkitemfactory.h

描述：GtkItemFactory用于简化菜单创建过程，Gnome应该使用GnomeUIInfo模板来创建菜单。