

# Kylix 程序设计

杨宗志 编著

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

Kylix 是 Borland 公司新近推出的 Linux 系统上的可视化开发工具,它比 Linux 系统上的其他开发工具都更具有亲和力,而且能快速、轻松地建立 Linux 应用程序。本书从 Kylix 的安装、主界面开始介绍,由浅入深、循序渐进地介绍了如何用 Kylix 开发应用程序。最后,综合所介绍的基础知识,用 Kylix 创建了“我的记事本”和“我的小画家”两个应用程序,将理论与实践相结合。

本书适合所有 Kylix 程序设计的入门者及希望在 Linux 系统上快速开发应用程序的读者阅读。

本书繁体字版名为《Kylix 学习教本》,由文魁资讯股份有限公司出版,版权属杨宗志所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部。

北京市版权局著作权合同登记号:图字 01-2001-3793 号

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:Kylix 程序设计

作 者:杨宗志

责任编辑:张秋香

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者:

发 行 者:新华书店总店北京发行所

开 本:787×960 1/16 印张:22.25 字数:450 千字

版 次:2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

书 号:ISBN 7-302-04847-9/TP·2874

印 数:0001~5000

定 价:33.00 元

# 序 言

当作者得知 Borland 公司将推出 Linux 系统上的 Delphi——Kylix 时，非常兴奋。经实际操作使用后，发现 Kylix 同样拥有可视化的设计环境，比其他 Linux 系统上的任何开发工具都具有亲和力，而且能快速、简单地建立 Linux 应用程序。

毕竟 Kylix 还是个新软件(目前才正式发行第 1 版)，一定会有一些功能不够完善，再加上 Linux 操作系统的版本很多，使得 Kylix 的安装过程中发生很多问题。作者曾在 RedHat 6.2+CLE0.9pl、RedHat7.0+CLE1.0 等中文系统中安装过，都有很多问题。但尽管如此，Kylix 的开发环境、交互式除错和强大的编译功能，仍使得 Kylix 成为 Linux 系统中最受欢迎的开发工具。

作者站在非程序员的角度来编写本书，即使是程序设计的入门者，也可以轻易地通过阅读本书学会如何使用 Kylix 来开发 Linux 系统上的应用程序。

现将本书主要内容分列如下：

- 第 1 章 介绍如何安装 Kylix 及安装过程中容易出现的问题，并介绍了 Kylix 的开发环境。
- 第 2 章 介绍程序的界面设计，包括窗体的设计、组件的布置等。
- 第 3 章 介绍程序设计的规则，包括程序单元的基本内容、变量及 Kylix 的工作模式等。
- 第 4 章 介绍触发事件及组件属性设置语句。
- 第 5 章 介绍利用 If、Case Of、For 语句编写具有判断能力的程序，并介绍如何跟踪程序的流程。
- 第 6 章 介绍 Kylix 中的各种事件程序。
- 第 7 章 介绍基本组件的应用，包括多样化按钮、选择类组件等。
- 第 8 章 介绍列表框、状态栏、计时器等基本组件的应用。
- 第 9 章 介绍菜单及工具栏的制作，包括下拉菜单、快捷菜单、工具栏组件等。
- 第 10 章 介绍如何利用各种组件来完成一个记事本程序。
- 第 11 章 综合本书各章的主要内容，制作一个简易的画图软件。

# 第 1 章 初识 Kylix

以前，要开发一个 Linux 操作系统上的应用程序是非常困难的。首先，必须要懂得高级的 C 语言；其次，传统的 Linux 应用程序开发工具非常难以学习和使用，而且没有我们在 Windows 操作系统上习惯使用的图形界面。但是当您用 Kylix 在 Linux 操作系统上开发应用程序时，将会发现编写程序是那样的简单。Kylix 就像 Windows 操作系统上的 Delphi，同样提供许多操作简单而功能强大的组件，用这些组件就可编写出属于您自己的程序。

- ➡ 安装 Kylix
- ➡ 集成开发环境
- ➡ 方便易用的工具

在安装 Kylix 之前，必须先检查一下系统的硬件环境。以下是 Borland 公司对于安装 Kylix 的硬件配置建议：

- Intel Pentium 200 以上兼容微处理器(建议采用与 Intel Pentium II 400 以上兼容微处理器)
- 64 MB 内存(建议使用 128 MB 以上)
- 最小安装需要 80 MB 硬盘空间，完全安装需要 230 MB 硬盘空间
- Linux 兼容的 VGA 或更高级的显示卡
- Linux 兼容的鼠标
- Linux 兼容的光驱

对操作系统的要求，Borland 公司提出以下的建议：

- 支持英文的 Linux 版本
  - ◆ RedHat 6.2 以上版本
  - ◆ Mandrake 7.2 以上版本
  - ◆ SuSE(r)7.0 以上版本
- 支持中文的 Linux 版本
  - ◆ RedHat 6.2 附加 CLE(GNUChineseLinuxExtension)0.9p1



### 注 意

本书内容是按照 Borland 公司所建议的中文系统环境，将 Kylix 安装在 RedHat 6.2 附加 CLE(GNUChineseLinuxExtension)0.9p1(Yami)的运行环境中，也可以安装在 RedHat 7.0 附加 CLE1.0(Atayal)的运行环境中。本书所使用的桌面环境是 KDE 的 X Window 图形界面。

## 1.1 安装 Kylix

在 Windows 系统中安装软件，一般都是直接将安装光盘放入光驱中就会自动开始安装软件了。但是，在 Linux 操作系统中，要使用光驱则必须经过安装才能使用。如何安装光



更新 glibc 后,将文件管理器的显示路径切换回光盘的根目录,也就是 file:/mnt/cdrom,在目录中单击 setup.sh 文件开始安装 Kylix。如果以终端机的字符模式进行安装,将会见到 Kylix 安装前的环境系统检查信息,不过必须先切换工作路径到光盘目录,然后运行 sh setup.sh 命令,如图 1.3 所示。



图 1.3 在终端机模式下运行安装程序

检查过安装环境没有问题之后,第 1 个出现的安装界面就是版权声明画面,如图 1.4 所示。



图 1.4 License Agreement 对话框

单击 I Agree 按钮后进入安装选项设置对话框。作者建议:如果您没有特殊的需求,就直接单击 Begin Install 按钮,按照默认的条件安装就行,如图 1.5 所示。



图 1.5 安装选项设置对话框

单击 Begin Install 按钮后，接着出现安装进度对话框，如图 1.6 所示。



图 1.6 安装进度对话框

所有的程序组件安装结束后，出现安装结果的提示对话框，告诉用户 Kylix 的安装路径在哪里，要如何启动 Kylix 等，如图 1.7 所示。最重要的是我们所使用的桌面环境是 KDE 的 X Window 图形界面，所以，安装完毕之后要重新启动 X Window 图形界面。



图 1.7 安装结果提示对话框

## 1.2 集成开发环境

安装完 Kylix 后,选择 K 菜单中的 Borland Kylix | Kylix 命令,就可以直接启动 Kylix 了,如图 1.8 所示。



图 1.8 启动 Kylix

进入 Kylix 的开发环境后，您将发现 Kylix 的图形化开发环境很友好。如果您使用过 Windows 系统上的 Delphi，看到这样的开发环境一定感觉很亲切，两者的界面几乎一模一样。Kylix 开发环境的整体外观如图 1.9 所示。

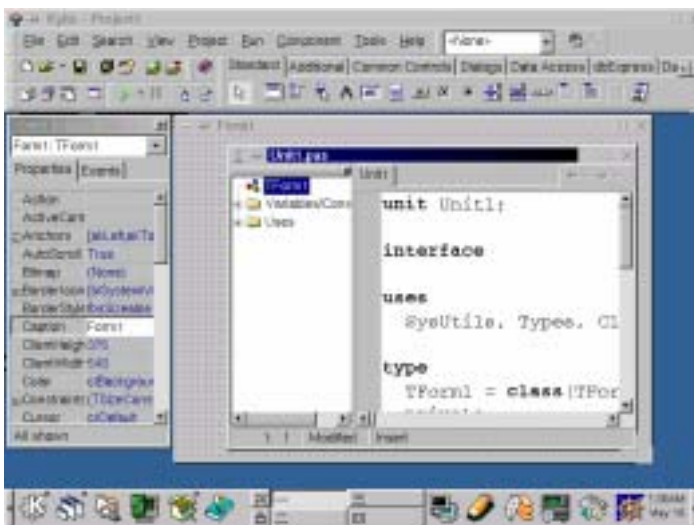


图 1.9 Kylix 的开发环境

进入 Kylix 后，共有 4 个工作窗口，分别是主窗口(标题栏上显示 Kylix - Project1 的窗口)、对象查看器(标题栏上显示 Object Inspector 的窗口)、窗体窗口(标题栏上显示 Form1 的窗口)、程序代码单元窗口(标题栏上显示 Unit1.pas 的窗口)。请注意：单击主窗口上的关闭按钮将关闭整个 Kylix 程序，而其他的窗口单元都只是 Kylix 的子窗口，用户可以根据自己的需要决定子窗口是否显示。

### 1.2.1 主窗口的菜单

Kylix 主窗口的菜单如图 1.10 所示。

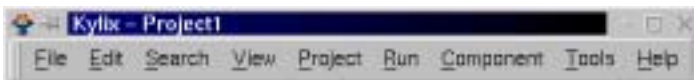


图 1.10 Kylix 主窗口的菜单

## 1. File(文件)

File 菜单命令包括打开、保存、打印、关闭新的和旧的工程或文件。也就是说，所有与文件相关的操作都在这个菜单内，如图 1.11 所示。

## 2. Edit(编辑)

我们常用的剪切、复制、粘贴的操作命令都在编辑菜单里。这个菜单内还提供在程序设计阶段时，可以用来编辑程序及组件属性的命令，如图 1.12 所示。

## 3. Search(搜索)

在我们编写程序代码时将会经常用到搜索菜单中的命令，其中包括查找文字、错误、对象、变量、符号、程序单元，以及快速将光标移到指定的程序代码行等命令，如图 1.13 所示。

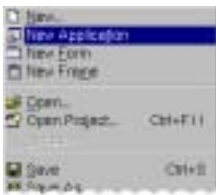


图 1.11 File 菜单



图 1.12 Edit 菜单



图 1.13 Search 菜单

## 4. View(查看)

查看菜单用来显示或隐藏 Kylix 的各种工具栏及相关的工具窗口，例如工程管理器 (Project Manager) 窗口等，如图 1.14 所示。

## 5. Project(工程)

必须先打开一个工程才能使用这个菜单内的各命令，工程菜单内的命令主要用来编译工程和管理工程，如图 1.15 所示。

## 6. Run(运行)

运行菜单包括各种不同的运行程序的方法，尤其是 Program Reset 命令，当程序在运行

中发生重大错误时，可利用这个命令强制中断程序的运行，如图 1.16 所示。



图 1.14 View 菜单

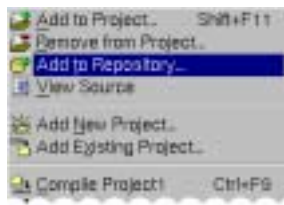


图 1.15 Project 菜单



图 1.16 Run 菜单

## 7. Component(组件)

组件管理菜单包含组件制作、单独组件安装和组件包安装等命令，如图 1.17 所示。

## 8. Tools(工具)

工具菜单主要包括一些用来管理工具栏及设置各个工具栏的命令，并可利用其中的 Configure Tools 命令来添加常用的程序，如图 1.18 所示。

## 9. Help(帮助)

联机帮助菜单如图 1.19 所示。

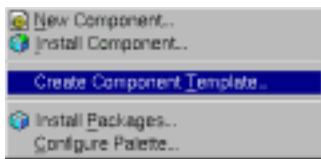


图 1.17 Component 菜单

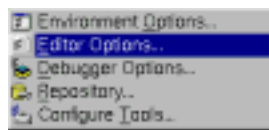


图 1.18 Tools 菜单

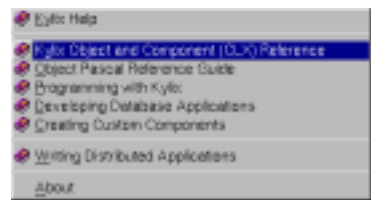


图 1.19 Help 菜单

### 1.2.2 主窗口的工具栏

在 Kylix 的主窗口中，共有 6 个工具栏，如图 1.20 所示。分别是 View(查看)工具栏、Standard(标准)工具栏、Debug(调试)工具栏、Custom(定制)工具栏、Component Palette(组件选项板)工具栏。每个工具栏都可脱离主窗口而成为单独的浮动窗口，并可按照个人的爱好

进行工具栏的排列组合。



图 1.20 Kylix 主窗口中的工具栏

## 1. Standard 工具栏

Standard 工具栏中的工具按钮就是由 File 菜单中的 New、Open、Reopen、Save、Save All、Open Project 等 6 个命令和 Project 菜单中的 Add to Project、Remove from Project 两个命令所组合而成，用来管理与工程文件相关的文件，如图 1.21 所示。

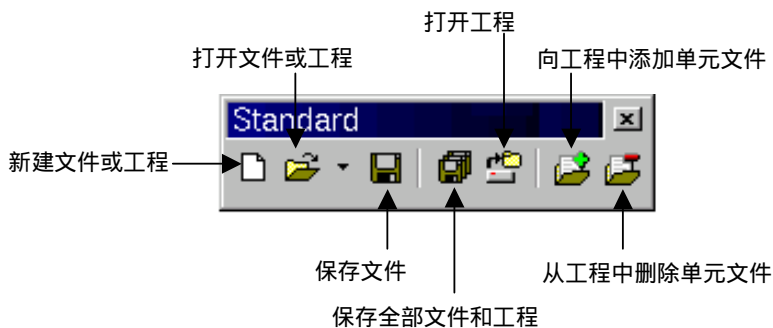


图 1.21 Standard 工具栏

单击新建文件或工程按钮将会弹出如图 1.22 所示的 New Items 对话框。



图 1.22 New Items 对话框

## 2. View 工具栏

View 工具栏中的工具按钮是由 File 菜单中的 New Form 命令和 View 菜单中的 Toggle Form/Unit、Units、Forms 等 4 个命令所组合而成，如图 1.23 所示。

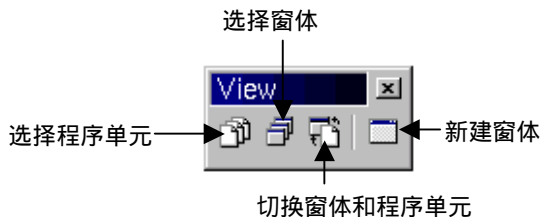


图 1.23 View 工具栏

在互相对应的窗体与程序单元间可使用 Toggle Form/Unit 按钮来做快速切换，特别是操作环境中有很多不同单元窗体时。如果要在工程中添加新的窗体则可单击 New Form 按钮。

如果我们设计的程序(工程)中有很多的窗体及程序单元，那么使用 Units 与 Forms 按钮是很好的选择。单击 Units 或 Forms 工具按钮将弹出 View Unit 或 View Form 对话框，如图 1.24 所示，我们可从列表中直接选择所需的窗体或单元。



图 1.24 View Form 和 View Unit 对话框

## 3. Component Palette 工具栏

Component Palette 工具栏是最重要的一个工具栏，其中放着各式各样的组件。组件的分类比较明确，各种不同性质的组件摆放在不同的选项卡中。要选择不同的选项卡只要使

用鼠标单击该选项卡的标签, Kylix 就会显示该选项卡的组件供我们选择。如果选项卡中的组件过多, 无法一次全部显示出来, 组件滚动指针就发挥作用了, 我们可以利用工具栏右侧的三角按钮来显示更多的组件, 如图 1.25 所示。



图 1.25 Component Palette 工具栏

标准选择指针按钮有两种状态。当其为被按下的状态时, 我们可利用鼠标拖动的操作方法对放置于窗体中的组件进行位置及大小的调整。相反, 当我们选中组件选项板中的某个组件后, 标准选择指针按钮就不会是被按下的状态, 这时表示我们要将选中的组件插入到窗体中。当我们将组件放置到窗体中后, 标准选择指针按钮就会再度成为被按下的状态。

如果要更改组件类别标签的名称, 请选择 Component | Configure Palette 命令, 如图 1.26 所示。

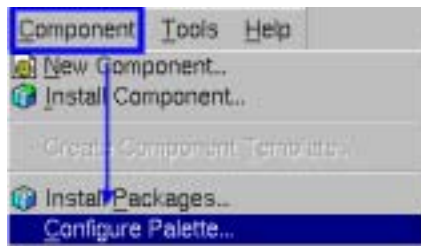


图 1.26 选择 Component | Configure Palette 命令

在打开的 Palette Properties 对话框中单击 Rename 按钮, 就可以更改标签名称, 更改的步骤如图 1.27 所示。

要更改标签的名称, 我们也可以在组件选项板上直接单击鼠标右键, 在弹出的快捷菜单中选择 Properties 命令, 这样也会打开 Palette Properties 对话框。

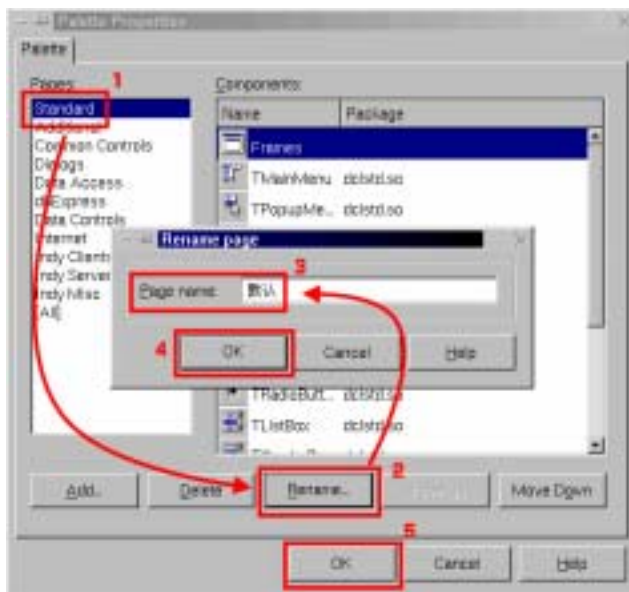


图 1.27 更改组件类别标签名称的步骤

#### 4. Debug 工具栏

当我们的程序设计告一段落(包含窗体设计及程序代码编写)时, 我们就会用到 Debug 工具栏了。在 Debug 工具栏中包含了 Run(运行程序)、Program Pause(暂停运行程序)、Trace Into(跟踪运行程序)、Step Over(单步运行程序)等 4 个按钮, 用来协助我们排除应用程序中的错误。要查看程序的设计结果只需单击 Run 按钮就行了。如果要暂停程序的运行则需要单击 Program Pause 按钮, 如图 1.28 所示。

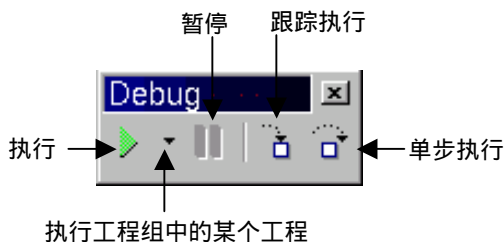


图 1.28 Debug (调试)工具栏

单击 Program Pause 按钮只能暂停程序的运行。要中止程序的运行可以直接单击窗口控

制按钮中的关闭窗口按钮，也可以选择 Run | Program Reset 命令来中止程序的运行。

如果我们设计的程序在运行时发生了错误，可以单击 Trace Into 按钮，此时 Kylix 将会采用每运行一行程序就暂停一次的方式来调试程序。当确认编写的程序或函数不会有错误发生时，则可以采用单步运行程序的方式来调试程序，单击 Step Over 按钮时，Kylix 在调试过程中如果遇到调用程序或函数时将会把该程序或函数当作一程序来执行，进而忽略其中所包含的程序代码。

## 5. Custom 工具栏

如图 1.29 所示的 Custom 工具栏是一个最简单的工具栏，只有一个 Help Contents 工具按钮，也就是所谓的在线帮助。如果您在程序编写过程中或对 Kylix 的操作有疑问时，可直接单击此工具按钮即可打开帮助对话框，进而获得相关的疑难解答，如图 1.30 所示。



图 1.29 Custom 工具栏

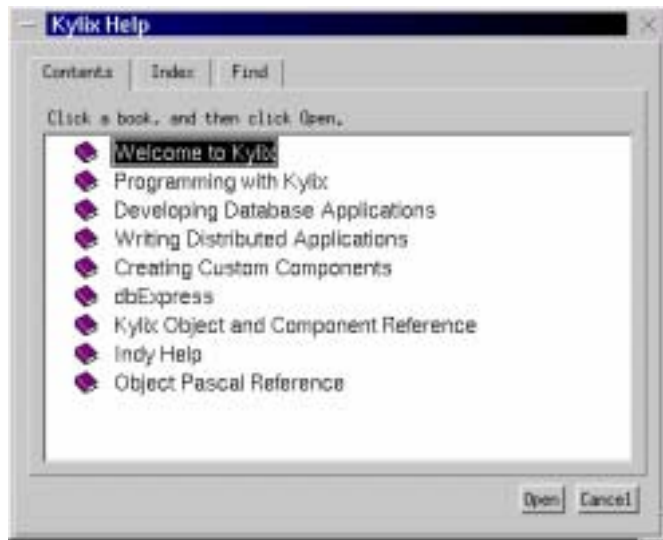


图 1.30 单击 Help Contents 工具按钮所出现的帮助主题对话框

## 6. Desktop 工具栏

Desktop 工具栏可以让用户保存多个桌面样式，并且让用户可以快速地在多个桌面之间进行切换，如图 1.31 所示。