

乔林 编著

您想站在 3 年后软件开发的风口浪尖上吗？

您想编写既可以在 Microsoft Windows 下运行

也可以在 Linux 下运行的跨平台软件吗？

从现在起就开始学习 Linux 编程吧！

Kylix 是您精通 Linux 编程的敲门砖。



Kylix 程序设计

实 战 教 程

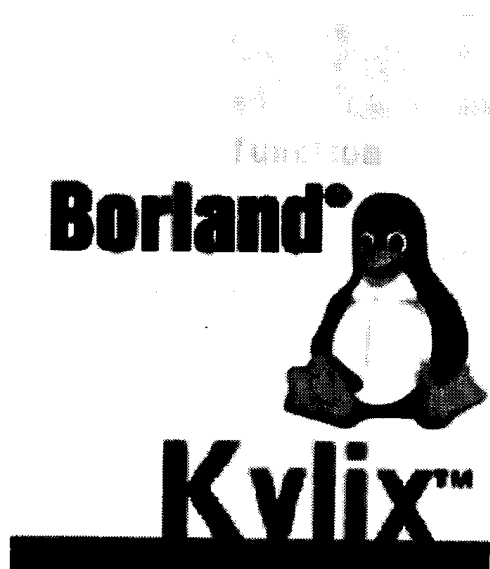


光盘内附 Kylix 试用版

Kylix 程序设计

——实战教程

乔 林 编著



中 国 铁 道 出 版 社

2002 年 · 北京

(京)新登字 063 号

内 容 简 介

本书以多个应用程序实例为基础,介绍了 Kylix 程序设计的基本方法。内容涉及多页面界面、多文档界面、文件、异常处理与程序调试、属性、方法指针与事件、高级面向主题、部件开发、数据库开发基础、网络编程初探、多线程编程及跨平台程序开发等。

书中详细剖析了各个实例,使读者学会正确的思考方法,以及如何正确地将思考方法转化为准确的程序代码。附带光盘中给出了书中所涉及的实例和练习的全部源代码。

本书是 Kylix 程序设计的高级读物,适合计算机软件开发人员和一般计算机人员,尤其是 Linux 爱好者使用。如果与本系列的其他图书配合使用效果更佳。

图书在版编目(CIP)数据

Kylix 程序设计实战教程/乔林编著. —北京:中国铁道出版社, 2001. 12

ISBN 7-113-04473-5

I. K… II. 乔… III. Linux 操作系统—软件工具, Kylix—教材 IV. TP311.56

中国版本图书馆 CIP 数据核字(2001)第 093886 号

书 名: Kylix 程序设计——实战教程

作 者: 乔 林

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街8号)

策划编辑: 严晓舟

特邀编辑: 王占清

封面设计: 王 晖

印 刷: 北京市彩桥印刷厂

开 本: 787×1092 1/16 印张: 28.75 字数: 696 千

版 本: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数: 1~4000 册

书 号: ISBN 7-113-04473-5/TP·657

定 价: 50.00 元

版权所有 盗印必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前言

可以毫不夸张地说，Kylix 是 Linux 发展的里程碑——它的出现是革命性的。

关于 Kylix

近几年来，Linux 无疑是业界上升最快的操作系统。作为最具竞争力的企业环境之一，由于无比低廉的价格，以及与价格相比实在太好的可靠性，Linux 广泛应用于 Web 应用服务器，并迅速成为 Windows 操作系统的主要竞争对手。

然而现实情况时，在 Linux 环境下开发应用程序不仅繁琐困难而且效率太低，程序员不得不进行大量的重复性劳动以编写 X Window 图形用户界面的程序代码，这种局面十分类似 Visual Basic 问世前 Windows 操作系统所面临的难局。随着 Linux 的发展，业界迫切需要一种快速开发工具，以弥补 Linux 下应用程序不足以及编程复杂的欠缺。

1999 年的 9 月 28 日，Borland/Inprise 公司正式宣布开发 Linux 环境下的快速开发工具，时隔仅一年半，Borland/Inprise 公司就适时推出了 Kylix。

Kylix 实际上是 Delphi 的 Linux 版本，它与 Delphi 一脉相承。Kylix 的出现，彻底终结了 Linux 过于艰深、曲高和寡的历史，每个普通的程序员都可以使用 Kylix 快速开发 Linux 下的应用程序。Kylix 是 Linux 发展的里程碑，它使得 Linux 第一次可以在桌面操作系统上与 Windows 一争高低。

为什么要使用 Kylix/Delphi

Kylix，或者说 Delphi for Linux，是运行于 Linux 环境下的 Delphi。无论是窗体设计器、代码编辑器还是对象检查器、部件面板，Kylix 与 Delphi 的运行界面都几乎不存在什么差别。此外，Kylix 下的编程习惯与 Delphi 几乎完全相同：键盘响应模式与鼠标操作完全一样，菜单和命令也几乎完全一样。

从集成开发环境上看，Kylix 与 Delphi 最大的不同是使用 CLX 部件库代替了 VCL 部件库。CLX 与 VCL 在众多方面都是相似的，CLX 的最大优势是基于 Qt 工具包，支持跨平台开发。

使用过 Delphi 的读者将会发现向 Kylix 中移植 Delphi 的代码是最容易不过的事情。除了 CLX 部件开发，大部分 Delphi 程序都只需要按照 Kylix 指出的方法做一些小改动就可以在 Linux 下直接编译运行。

您想站在 3 年后软件开发的风口浪尖上吗？您想编写既可以在 Microsoft Windows 下运行也可以在 Linux 下运行的跨平台软件吗？从现在起就开始学习 Linux 编程吧！Kylix 是您精通 Linux 编程的敲门砖。

关于本书

本书的宗旨是通过大量或深或浅的实例讲解 Kylx 的内容。整套丛书使用多个有趣的例子讨论 Object Pascal 语言的基本要素、Kylx 编程的基本与高级方法。本书引出的概念不是按照传统的教学顺序组织的，而是按照对读者是否有用的顺序组织的。读者会发现，这样的组织方式将使学习过程更有效。

贯穿整本丛书的是读者应该遵循什么样的步骤，应该采取什么样的思考方法，以及如何将自己的思考转化为正确的程序代码。因此，丛书中包含大量实例和练习，这些实例和练习有的难、有的易，而且在很多情况下还有一定的关联。我们希望读者在阅读本丛书的同时也上机实践。

每一章后面都包含一些问题与练习，它们的答案通常可以在书中找到线索，但也有一些练习必须通过上机实践才能完成。笔者深信，只有实践才是出真知的最佳途径。当然我们也要指出，有一些练习是稍难的。笔者同样深信，只有独立完成稍难一点点的问题才是迅速提高的唯一途径，太易的练习没有任何意义，而太难的练习则失去了实践的价值。对于较难的练习我们将给出一定的提示，请读者务必完成它们。

本书是如何组织的

本书包括四个部分：

- 《Kylx 程序设计——基础教程》：基础教程着重解决安装和使用 Kylx 面临的基本问题。考虑到 Linux 下的程序员大多对 C/C++ 很熟悉，而对 Object Pascal 语言不太了解，该书详细介绍了 Object Pascal 语言，对于初学者有极大的帮助。
- 《Kylx 程序设计——进阶教程》：进阶教程着重讨论《基础教程》遗留下来的 Object Pascal 语言的高级特性以及如何设计漂亮的应用程序界面。本书同时研究了如何使用 Kylx 开发数据库应用程序。
- 《Kylx 程序设计——实战教程》：详细讨论使用 Kylx 开发高级应用程序、多线程应用程序、Internet 应用程序、数据库应用程序、跨平台应用程序设计的基本技巧。
- 《Kylx 程序设计——类库参考手册》：提供完整的 CLX 类库参考手册，免去读者在帮助系统中跳来转去的困扰。

本书的内容由浅入深，没有阅读前面的内容就直接学习后面的章节可能会给读者带来一定的麻烦，当然对相应内容有相当程度了解的读者又另当别论了。

本书是集体劳动的结晶，参加编写的有乔林、林杜、费广正、王程铭、王志海、何隽、计莉琴、朱辰麒、乔木、蒋培、蔡宏、汪巨涛、陈爱华、王冰、乔森、徐斌、马友兰、胡一敏、李享、刘丰收、朱琨、张喜等。由于编者学识水平有限，书中难免有缺点、错误和不当之处，恳请读者批评指正。

本书的读者对象

本书包含了使用 Kylx 编程的方方面面。即使读者没有任何计算机编程经验，也可以阅读本书。为理解本书的内容，读者应该熟悉 X Window 操作系统的基本知识。对读者

最低限度的假设是能够知道 X Window 的基本操作，能够知道如何使用 X Window 环境就足够了。

我们再一次地重申，本丛书的对象包含了 Kylix 的初级到高级的所有读者：

- 初学者：本丛书的《Kylix 程序设计——基础教程》和《Kylix 程序设计——进阶教程》是极好的学习材料，初学者可以从中学到几乎全部的 Object Pascal 语言细节和基本使用技巧。
- 中级读者：可以通过学习本套丛书澄清很多以前没有完全了解的概念。
- 高级读者：我们敢保证，您肯定会在本套丛书的字里行间发现很多在其他书中看不到的内容——那是作者使用 Kylix 时的独特体会，也许是本丛书最具价值的部分——这部分内容对于您理解 Borland 公司在实现 Kylix 时的设计原则有莫大的裨益。要知道，Kylix 本身是如何实现的与我们使用她时采用什么样的编程方法和编程习惯息息相关。

“边做边学”的方法是最有效的。每学完一个例子，尝试着改变一点点，或者添加一点东西，并改变一些代码将帮助读者体验进步和成功的乐趣。

编 者

2001.10

目 录

第 1 章 多页面界面	1
1.1 多页面界面的基本概念	1
1.2 使用 TTabControl 部件	2
1.3 使用 TPageControl 部件	8
1.3.1 静态 MPI 界面	8
1.3.2 动态 MPI 界面	12
1.3.3 动态 MPI 界面的管理	15
1.4 MPI 文本编辑器实例	24
1.4.1 程序基本框架	24
1.4.2 程序代码分析: 动态创建工作区	30
1.4.3 程序代码分析: 编辑与选项操作	33
1.4.4 程序代码分析: 查找与替换操作	34
1.4.5 程序代码分析: 文件的新建、打开与保存操作	36
1.4.6 程序代码分析: 文件的关闭操作	38
1.4.7 程序代码分析: 窗口标题的更新操作	41
1.5 小 结	41
1.6 问题与练习	42
第 2 章 多文档界面	43
2.1 多文档界面的基本概念	43
2.2 窗体的继承与多态	44
2.2.1 窗体的继承	44
2.2.2 窗体的多态	46
2.3 TActionList 部件与 TAction 类	55
2.3.1 为什么要引入行为抽象	55
2.3.2 行为抽象的机制	56
2.3.3 使用 TActionList 部件	58
2.3.4 活动的触发规则	60
2.4 MDI 文本编辑器实例	62
2.4.1 使用 MDI 应用程序向导	62
2.4.2 程序代码分析: 修改主窗体	69
2.4.3 程序代码分析: 修改子窗体 clxchildwin	72
2.4.4 程序代码分析: 设计窗体 clxtextwin	74
2.4.5 程序代码分析: 设计窗体 clximagewin	76

2.4.6	程序代码分析: 设计窗体 clxfiletypewin	78
2.4.7	程序代码分析: 修改主窗体 clxmain	80
2.4.8	程序代码分析: 文件的打印	88
2.5	小 结	90
2.6	问题与练习	90
第 3 章	文 件	91
3.1	文件类型及标准文件过程与函数	91
3.1.1	文件类型	91
3.1.2	文件操作标准子程序	92
3.2	文本文件的处理	98
3.2.1	文本文件的基本操作	98
3.2.2	实例程序 TextIO	98
3.3	有型文件的处理	103
3.3.1	有型文件的基本操作	103
3.3.2	实例程序 TypedIO	104
3.4	无型文件的处理	108
3.5	使用文件流类	110
3.5.1	创建 TFileStream 流式对象	110
3.5.2	使用文件流进行文件复制	113
3.5.3	使用文件流读写数据文件	113
3.6	彩票分析系统实例	116
3.6.1	基本数据结构	116
3.6.2	TUntypedFileStream 类的设计	118
3.6.3	程序界面设计	122
3.6.4	TMainForm 类的设计	130
3.6.5	“Stakes” 页面程序代码分析	133
3.6.6	“Frequency” 页面程序代码分析	136
3.6.7	“Distribution” 页面程序代码分析	140
3.7	小 结	149
3.8	问题与练习	149
第 4 章	异常处理与程序调试	151
4.1	异常处理机制	151
4.1.1	异常处理的基本理论	151
4.1.2	异常引发的基本理论	153
4.2	CLX 异常类	154
4.2.1	异常类 Exception	154
4.2.2	异常类层次	155

4.2.3	运行库异常.....	156
4.2.4	对象异常.....	161
4.2.5	部件异常.....	163
4.3	异常的响应与处理.....	164
4.3.1	异常保护机制.....	164
4.3.2	使用异常实例.....	167
4.3.3	缺省异常响应.....	167
4.3.4	多个异常的响应.....	168
4.3.5	异常的重引发.....	169
4.3.6	异常层次嵌套.....	170
4.3.7	异常编程.....	171
4.4	自定义异常类的应用.....	172
4.4.1	定义异常类.....	172
4.4.2	引发自定义异常类的异常.....	172
4.5	程序调试.....	175
4.5.1	设置调试工具选项.....	176
4.5.2	控制程序执行.....	177
4.5.3	使用断点.....	178
4.5.4	检查变量和表达式的值.....	181
4.5.5	检查函数或过程调用.....	183
4.5.6	检查线程状态.....	184
4.5.7	检查 CPU 状态.....	184
4.5.8	检查局部变量.....	186
4.5.9	检查浮点处理器状态.....	186
4.5.10	检查模块装载与卸载.....	188
4.5.11	检查事件日志.....	189
4.6	小 结.....	190
4.7	问题与练习.....	190
第 5 章	属性、方法指针与事件.....	191
5.1	属 性.....	191
5.1.1	引入属性的意义.....	191
5.1.2	属性的声明.....	193
5.1.3	属性的访问说明.....	193
5.1.4	属性的存储说明.....	195
5.1.5	属性的索引说明.....	196
5.2	高级属性.....	197
5.2.1	数组属性.....	197
5.2.2	属性的覆盖与重定义.....	199

5.3	TMyDate 类实例	200
5.4	方法指针与事件	208
5.4.1	方法指针类型	208
5.4.2	事 件	210
5.4.3	为 TMyDate 类添加事件	212
5.5	小 结	216
5.6	问题与练习	216
第 6 章	高级面向对象主题	217
6.1	类过程与类函数	217
6.2	类引用类型	219
6.2.1	类引用类型的声明	220
6.2.2	类引用类型与动态创建部件	220
6.2.3	类引用类型与类方法	224
6.2.4	类引用类型与类定义	225
6.3	对象的内部存储格式	227
6.3.1	域的存储格式	227
6.3.2	静态方法的存储格式	229
6.3.3	动态与虚拟方法的存储格式	230
6.3.4	属性的存储格式	240
6.4	类操作符	240
6.4.1	is 操作符	240
6.4.2	as 操作符	241
6.5	运行时类型信息	243
6.5.1	运行时类型信息的定义	243
6.5.2	运行时类型信息的意义	243
6.5.3	运行时类型信息的数据结构	245
6.5.4	获取类型信息	247
6.5.5	获取方法指针信息	255
6.6	小 结	258
6.7	问题与练习	258
第 7 章	部件开发	261
7.1	部件开发的基本方法	261
7.1.1	部件的基本概念	261
7.1.2	部件开发的基本途径	262
7.1.3	部件开发的基本流程	263
7.2	由现有部件派生新部件	265
7.2.1	创建和注册简单部件	265

7.2.2	修改部件类.....	268
7.2.3	测试新部件.....	268
7.3	包.....	270
7.3.1	包的基本概念.....	270
7.3.2	包与标准共享对象文件.....	270
7.3.3	创建包.....	271
7.3.4	包的组织与管理.....	273
7.4	由 TCustomControl 类派生新部件.....	276
7.4.1	使用部件专家创建新部件.....	276
7.4.2	创建属性.....	276
7.4.3	实现构造方法.....	278
7.4.4	实现字体调整方法.....	279
7.5	创建不可见部件.....	280
7.5.1	创建非可视化部件.....	280
7.5.2	使用非可视化部件.....	282
7.7	小 结.....	283
7.8	问题与练习.....	283
第 8 章	数据库开发基础.....	285
8.1	数据库系统概述.....	285
8.1.1	数据库的基本概念.....	285
8.1.2	数据库的基本类型.....	286
8.1.3	数据库的安全性.....	287
8.1.4	事务处理.....	287
8.1.5	引用一致性、存储过程与触发器.....	288
8.2	数据库系统的体系结构.....	288
8.2.1	数据库系统的一般结构.....	288
8.2.2	基于文件的客户数据集.....	289
8.2.3	单向数据集.....	290
8.2.4	使用客户数据集缓冲数据.....	291
8.2.5	多层体系结构.....	292
8.3	使用数据控制部件.....	293
8.3.1	数据控制部件的基本特性.....	293
8.3.2	数据控制部件的公共任务.....	294
8.3.3	数据的组织.....	298
8.3.4	显示与编辑多行记录.....	310
8.3.5	浏览与操作记录.....	320
8.4	使用数据集部件.....	323
8.4.1	数据集的基本概念.....	323

8.4.2	数据集的打开与关闭.....	324
8.4.3	数据集的状态设置.....	324
8.4.4	数据集的失效.....	327
8.4.5	数据集的记录定位.....	327
8.4.6	数据集的编辑.....	328
8.4.7	数据集的书签.....	331
8.4.8	数据集的查找.....	332
8.4.9	数据集的过滤.....	334
8.4.10	数据集的事件.....	335
8.5	使用客户数据集部件.....	336
8.5.1	创建主—明细结构数据库应用程序.....	336
8.5.2	记录的浏览与搜索.....	339
8.5.3	数据编辑.....	340
8.5.4	排序与索引.....	341
8.6	使用字段部件.....	342
8.6.1	TField 类及其派生类.....	343
8.6.2	动态字段部件与持久字段部件.....	343
8.6.3	定义新的持久字段部件.....	345
8.6.4	设置持久字段部件的显示格式.....	347
8.6.5	运行时的持久字段部件操作.....	348
8.6.6	字段值的显示、转换与访问.....	350
8.7	小 结.....	351
8.8	问题与练习.....	352
第 9 章	网络编程初探.....	353
9.1	HTML 语言.....	353
9.1.1	SGML 的组成.....	353
9.1.2	HTML 元素.....	353
9.1.3	HTML 注释.....	354
9.1.4	HTML 文档的显示.....	354
9.1.5	HTML 文档字符集.....	354
9.1.6	HTML 字符实体.....	355
9.1.7	HTML 数据类型.....	355
9.1.8	HTML 文档结构.....	355
9.1.9	HTML 文档浏览器.....	357
9.2	HTTP 编程.....	360
9.2.1	HTTP 协议概要.....	360
9.2.2	HTTP 协议的基本特点.....	360
9.2.3	HTTP 的工作原理.....	362

9.2.4	HTTP 的请求与响应	363
9.2.5	HTTP 头标	365
9.2.6	HTTP 编程实例	367
9.3	FTP 编程	379
9.3.1	FTP 会话	379
9.3.2	FTP 模型	380
9.3.3	FTP 文件类型	380
9.3.4	FTP 传输模式	381
9.3.5	FTP 应答	381
9.3.6	FTP 编程实例	383
9.4	小 结	398
9.5	问题与练习	398
第 10 章	多线程编程	399
10.1	线程的基本概念	399
10.1.1	进程与线程	399
10.1.2	宜使用多线程的场合	400
10.1.3	不宜使用多线程的场合	400
10.1.4	线程的调度与优先级	401
10.2	定义线程对象	402
10.2.1	TThread 类	402
10.2.2	定义线程类	404
10.2.3	线程对象的初始化	405
10.2.4	线程对象的释放	406
10.3	实现线程函数	407
10.3.1	使用 CLX 主线程	407
10.3.2	使用线程局部变量	408
10.3.3	检查其他线程是否终止	408
10.3.4	处理线程异常	408
10.3.5	线程清除代码	409
10.4	线程同步	409
10.4.1	使用 Synchronize 方法	409
10.4.2	使用 Lock 方法与 Unlock 方法	418
10.4.3	使用 TCriticalSection	422
10.4.4	使用 TMultiReadExclusiveWriteSynchronizer	426
10.4.5	使用 WaitFor 方法	427
10.5	小 结	433
10.6	问题与练习	433

第 11 章 跨平台程序开发.....	435
11.1 从 Windows 到 Linux 的标准移植步骤.....	435
11.2 CLX 与 VCL 的差异	439
11.3 Windows 与 Linux 的差异	441
11.3.1 环境差异	441
11.3.2 目录差异	442
11.4 编写可移植代码的一般原则.....	443
11.4.1 一般原则	443
11.4.2 使用条件编译指令	443
11.4.3 位置无关代码与位置相关代码	445
11.4.4 消息与系统事件	445
11.5 小 结.....	446

第 1 章 多页面界面

经过前两册书的学习，读者应已掌握了 Kylix 程序设计的基本方法与技巧，本书将深入研究使用 Kylix 开发应用程序时所需要处理的各种实际问题。

本章首先通过多个实例研究 Kylix 多页面界面部件以及多页面程序界面的设计方法。Kylix 的多页面部件主要有 TTabControl 与 TPageControl 两类，其中后者的每个页面为单独的 TTabSheet 部件。

本章内容包括：

- 多页面界面的基本概念
- 使用 TTabControl 部件
- 使用 TPageControl 部件
- MPI 文本编辑器实例

1.1 多页面界面的基本概念

多页面界面（multiple-page interface, MPI）是一种非常友好的用户界面。一般地，MPI 由一个窗体和多个页面组成，每个页面的信息列在窗体顶部或底部的页标签（Tabs）上，用户可通过选择页标签的方式进行页面切换。例如如图 1.1 所示，Kylix 的代码编辑器就是典型的 MPI 界面。

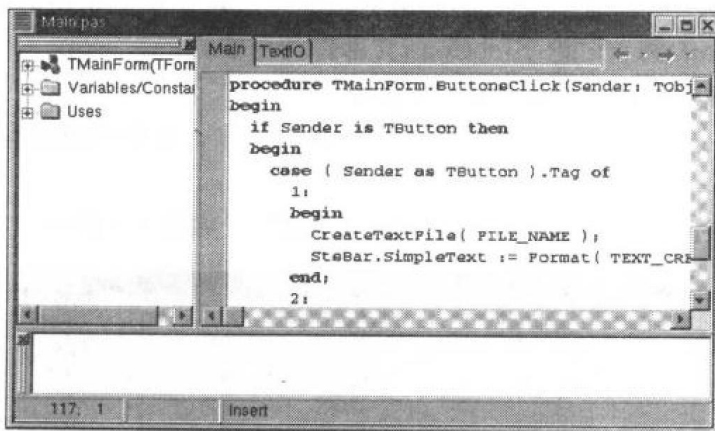


图 1.1 Kylix 的代码编辑器

MPI 界面一般分为静态和动态两类。静态 MPI 的页标签数量固定，用户在事先设计好的多个页面上进行切换；而动态 MPI 的页标签数量在程序运行过程中并不固定，由程序根据需要动态地生成或释放。图 1.1 所示的代码编辑窗体就是动态 MPI，程序可根据用户的需要产生多个文本页面，也可以动态地关闭某个页面。

闲言碎语

在多页面界面中，各页的排列方式比较固定，因而它们之间的切换速度较快。但在同一时刻，包含多页面的窗体只能显示其中的一个页面。

利用 Kylux 的多页面界面部件 TTabControl 或 TPageControl 可以很方便地设计静态 MPI，而设计动态 MPI 则需要编写额外的程序代码。下面分别讨论这两个部件。

1.2 使用 TTabControl 部件

TTabControl 部件位于部件面板的“Common Control”页上，它是一个页标签的集合体。TTabControl 部件重要的属性有：

- TabHeight 属性：设置页标签的高度。注意，此属性对所有页标签有效。
- TabWidth 属性：设置页标签的宽度。此属性同样对所有页标签有效。
- Align 属性：确定页标签在其容器对象（由 Owner 属性指定）中的对齐属性。
- MultiLine 属性：确定是否以多行的形式显示 TTabControl 部件的页标签。
- Images 属性：指定要在页标签上绘制的图像；Images 指定正常显示的图像。
- HotImages 属性：指定要在页标签上绘制的图像；HotImages 指定当鼠标移动到某个页标签上时的显示图像。

再学一招

Images、DisabledImages 与 HotImages 都为 TCustomImageList 类类型，各个图像在 TCustomImageList 中的出现顺序与页标签的顺序严格一一对应。

- Tabs 属性：指定各个页标签的标题。
- TabIndex 属性：维护页标签的索引号。注意，索引号由 0 开始计数。

我们以一个实际的例子讨论 TTabControl 部件的用法。

- 步骤 1：首先创建一个新目录“mpitab”。
- 步骤 2：创建一个新工程，并保存到上述目录中。我们这里没有更改文件和单元的名称。
- 步骤 3：在窗体上添加一个 TTabControl 部件，如图 1.2 所示。

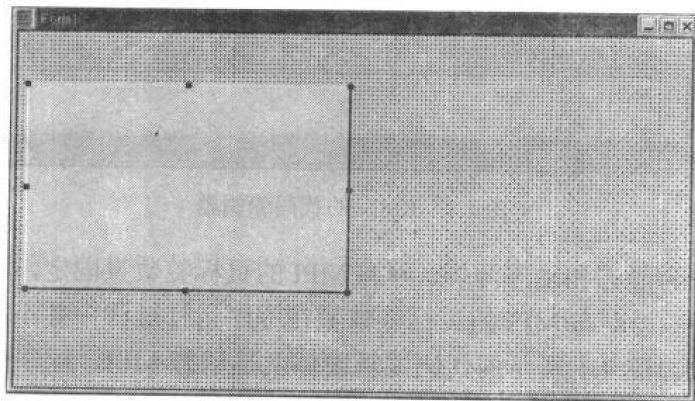


图 1.2 添加 TTabControl 部件

- 步骤 4: 在对象检查器中选择 TabControl1 对象, 单击 Tabs 属性右边的按钮; 或者如图 1.3 所示, 直接在窗体中 TabControl1 对象上单击鼠标右键, 从弹出的菜单中选择“Tabs Editor”菜单项。

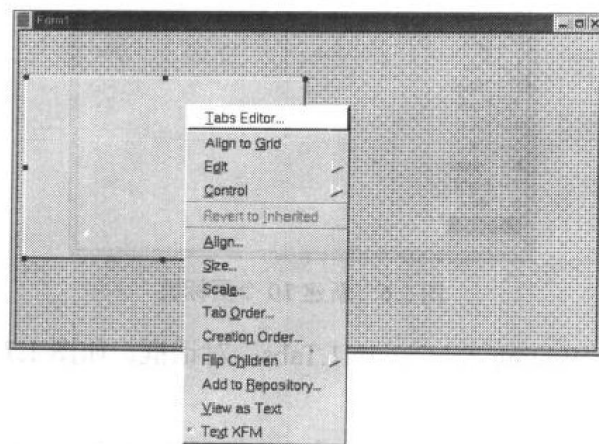


图 1.3 TTabControl 部件的弹出式菜单

- 步骤 5: 在 Kylix 显示的“Editing TabControl1.Tabs”对话框中添加新的页标签。如图 1.4 所示, 整个对话框只有一个按钮处于活动状态, 其即为“New”（新建）按钮。

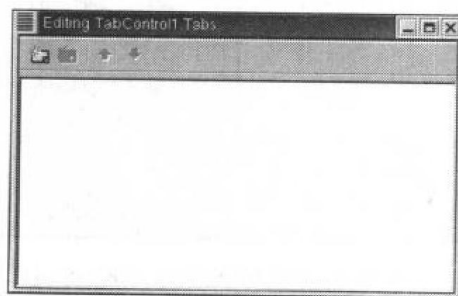


图 1.4 “Editing TabControl1.Tabs”对话框

- 步骤 6: 单击“Editing TabControl1.Tabs”对话框的“New”按钮, Kylix 自动将光标转移到对象检查器中, 如图 1.5 所示, 直接输入页标签的标题“One”即可。

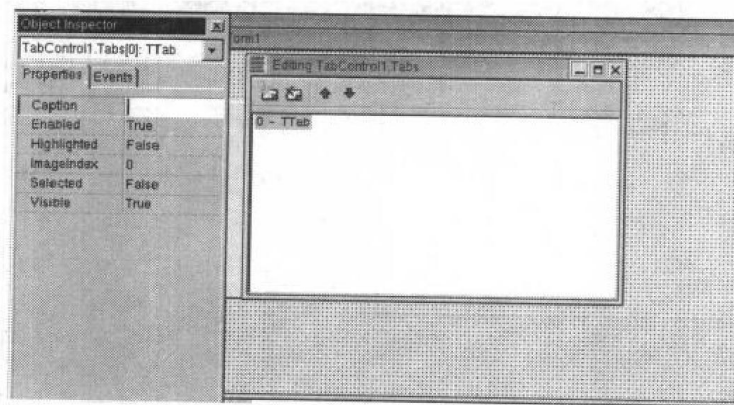


图 1.5 新建一个页标签