突破语言障碍进行多语言程序的开发

金 辉 石 敏 (装备指挥技术学院软件中心 101416)

摘要：本文以Borland公司Delphi5为开发工具，详细介绍了一种面向世界全球化解决方案的软件开发技术--多语言程序开发技术。

关键词：多语言程序 国际化 本地化 集成翻译环境

1 引言

随着全球信息水平的不断提高、交流日益紧密，软件的开发和应用不再局限于一个国家、一种语言，而是要求一种能够迅速面向世界各国的全球化解决方案。在此背景下，能够有效突破语言障碍，面向世界各国的软件开发技术-多语言程序的软件开发，已成为当前软件开发工具和软件开发的新趋势。本文以Borland公司Delphi 5为开发工具，详细介绍和说明了这种独立于用户国家、语言进行开发，然后可快速、有效进行其他国家和地区软件本地化移植的开发技术。

2 Delphi 5下多语言程序开发

2.1 开发特点

Delphi 5是Borland公司推出的新一代面向对象、可视化快速RAD (Rapid Application Development) 程序开发工具，在它的集成开发环境中融入了许多简化应用开发、缩短开发周期、提高软件开发生产力、友善的编程思想和技术，其中ITE(Integrated Translation Environment)为开发多语言程序提供了直接支持。利用ITE进行多语言程序的开发过程，具有如下特点：

(1) 在程序的开发阶段，按照软件国际化的基本要求进行程序编码，即将所有除Windows控件属性以外的显示资源（如各种提示信息）集中在一个程序单元内进行定义；

(2) 完成程序编码工作后，进入程序的本地化开发阶段，根据locale (语言和地区的组合，如中文(中国)、中文(香港)、英文(美国)等)的要求，将所有显示资源(提示信息+Windows控件属性）根据各种语言的特点（语言、字体、字型等）生成对应的资源动态库；

(3) 分发软件由程序的执行代码和支持其他语言的资源动态库组成，应用程序将根据操作系统的locale特性，自动显示与之对应的语言文字，从而最终自动实现软件多语言的效果。

2.2 开发步骤

步骤1：程序的国际化(Internationalize)

为了使程序能够快速适应多语言、本地化的要求，程序的编码阶段必须要做到国际化，即将所有与语言相关的提示信息集中在一个程序单元内进行定义：

```
unit uResourceConst;
interface
resourcestring
StatusBarHints = `English Version: Hello World!';
...
implementation
end.
```

在上面的示例中，我们将程序状态条显示的提示信息和其他提示信息使用字符串常量统一定义在resourcestring关键字单元中，这些提示信息将在不同语言背景下用不同语言进行显示，它们的集中定义具有如下好处：

· 消除字符串的重复定义，节省一定空间开销；

· 可以有效避免资源名字冲突碰撞问题；

· 易于程序的维护和扩充；

其他程序单元可以通过引用该单元(use uResourceConst)来使用这些字符串常量。按照正常的Delphi程序进行开发完成后，我们将进入多语言程序开发的第二阶段 - 程序的本地化开发，即翻译和生成不同语言的资源动态库。

步骤2：程序的本地化(Localizing)

程序的本地化过程，即翻译和生成与所选择locale一致的资源动态库，该

资源动态库具有与EXE相同的名字，其文件后缀与语言的locale一致，如开发语言所生成的执行文件为test.exe，locale设为“中文(中国)”，其所生成的支持中文(中国)的资源库为

test.CHS，locale设为“英文(美国)”，其所生成的支持英文(美国)的资源库为test.ENU。具体操作过程如下：

(1) 创建一个示意性的工程Test，在Form上放一个Label、一个Button和状态条，Caption赋值为"English Label"和"English Button"，字体分别为"Times New Roman-粗斜体-10"和"MS Sans Serif-粗体-10"，当按下Button时，状态条上显示ResourceString中所定义的StatusBarHints的信息：

(2) 选择菜单Project|languages|Add ...，Delphi显示Add Languages对话框，选择Next：(图略)

(3) 从语言列表中选择程序所需支持的locale，如"英语（美国）"和"中文（中国）"，选择Next：(图略)

(4) ITE会根据所选的语言创建子目录，目录名称是语言locale的缩写，选择Next：(图略)

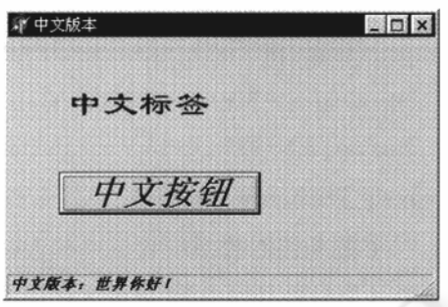
(5) 第一次增加语言时，Update Mode固定为Create new，选择Next：(图略)

(6) 选择Finish (图略)

(7) Delphi会自动创建支持不同语言的资源动态库(如Test.enu，Test.chs)，同时会自动创建一个工程组(Project Group)，工程组由Test.EXE、Test.enu、Test.chs组成：(图略)

(8) 启动Translation Manager，左边是需要翻译的语言，右边是已完成百分比信息：(图略)

(9)下面，我们将对程序的VCL Windows控件进行汉化。选择“中文(中国)”|Forms|uMain，修改label1和Button1的Caption属性为“中文标签”和“中文按钮”，Font.Charset属性为“GB2312_CHARSET”，Font.Name属性为“隶书”和“宋体”，Font.Style属性为“[]”和“[fsBold、fsItalic]”：(图略)



(10) 对程序的resourceString单元内定义的字符串进行汉化。选择“中文（中国）”|ResourceScripts|test_DRC，将uResourceConst_StatusBarHints汉化为“中文版本：世界您好！”：(图略)

(11) 关闭Translation Manager。打开Project Manager，编译Test.enu和Test.chs，Delphi会在相应的子目录生成资源动态库，但后缀不是DLL而是ENU和CHS：(图略)

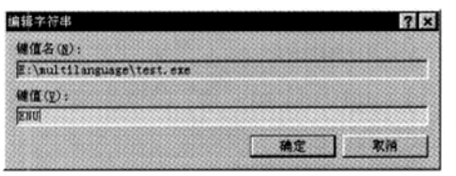
(12) 对于程序VCL Windows控件的汉化移植工作，我们也可以在Project Manager下激活test.chs，直接对Form1进行中文界面的修改，这使得本地化移植工作更加直观、灵活和简单；

(13) 以下分别是中文和英文版本的执行效果：

(14) 可以在test.exe project被激活时，选择Project|languages|Set Active ...，来设置当前多语言程序所使用的资源动态库。当应用程序界面发生变化，我们也可以通过Project|languages|Update Resource DLLs，来更新已有的资源动态库。值得注意的是，当且仅当原有开发语言所建立的project (test.exe)被激活时，与多语言相关的菜单才被使能、才能被选中。

步骤3：多语言程序的测试

由执行文件和语言资源动态库组成的多语言程序启动时，首先寻找执行目录下是否有与当前系统locale特性一致的资源库,如果有，程序将使用该资源库，否则，程序将使用与locale



特性中与语言相一致的资源库，如果上述资源库均不存在，程序将缺省使用包含在EXE文件内部开发语言所使用的各种资源。例如，我们在中文(中国)平台上运行test.exe，程序启动时，将首先寻找test.chs是否存在，

如果存在，那么将显示中文的各种界面，如果test.chs未被找到，程序将进一步寻找是否有其他中文的资源库存在，如果也不存在，程序将缺省使用程序开发时的各种资源。由于大家的操作系统绝大多数为中文操作系统，那么，我们如何进行其他语种的程序测试呢？

方法1：不断修改操作系统的locale属性（控制面板-）区域设置）；

方法2：修改注册表。在系统注册表HKEY_CURRENT_USER＼Software＼Borland＼Locales下，增加一条以应用程序文件名（包含路径和EXE后缀）为key的串值，其键值设为资源库的locale后缀，例如：

那么程序启动时，注册表中所注册的资源库如果存在，将首先被使用，在Delphi 5开发环境中指明当前活跃的语言，其内部实现也是修改了注册表。我们可以单独开发一个小程序，对开发的多语言程序进行测试：

```
procedure SetLocaleOverrides
(const FileName, LocaleOverride :
string);
    var
    Reg: TRegistry;
    begin
    Reg := TRegistry.Create;
    try
    if Reg.OpenKey(`Software ＼
Borland ＼ Locales',True) then
    Reg.WriteString(FileName,
LocaleOverride);
    finally
    Reg.Free;
    end;
    end;
    ...
    //test resources for English
    SetLocaleOverrides(ParamStr(0),
`ENU');
```