

# 利用 Delphi 流机制自动 保存 / 恢复窗体及组件状态信息

## Using Delphi Stream Mechanism to Save/Restore Status of Forms and Components

秦天保 (同济大学经济管理学院 200135)

**摘要:** 专业应用程序通常要求窗体能记住自己的状态, 在下次打开时能自动恢复状态。本文给出了利用 Delphi 的流 (Stream) 操作机制简化窗体及其组件状态信息的保存恢复操作。

**关键词:** 流 窗体状态 Delphi

### 1 引言

开发专业 Delphi 应用程序时, 经常要求窗体及其包含的组件能记住自己的状态 (如位置、大小等)。在下次打开窗体时能自动恢复状态, 常用的方法是利用 INI 文件或 Windows 注册表来保存状态信息。在窗体的 OnDestroy 事件中将窗体的属性值如位置、大小等写入到一个 INI 文件 (或注册表) 中, 然后在 OnCreate 事件中, 再从此 INI 文件 (或注册表) 读出属性值并据此设置窗体属性, 这种方法要编码遍历窗体和内含组件所有需要保存的属性, 由于属性非常多实现起来比较烦琐, 通用性也不高。本文将介绍另一种简单的自动保存 / 恢复窗体及其组件状态信息的方法, 即利用 Delphi 提供的流 (stream) 机制, 通过流操作可以将整个窗体及其组件的属性保存到一个二进制资源文件中, 并在其后从中恢复, 这种

方法实现简单, 通用性好。

### 2 原理

在 Delphi 中, “流”是一种读写数据的方法, 它提供了向各种存储介质 (如磁盘文件、内存、字符串等) 读写数据的通用的接口, 使你能以统一的方式读写不同的存储介质, 屏蔽了底层的复杂性, 可以将流理解为一个对象, 它代表了对应的存储介质, 或者说它是对应存储介质的抽象表示, 对流的读写就是对相应存储介质的读写, Delphi 针对不同的存储介质定义了相应的流类: TFileStream 用于文件读写, TStringStream 用于操纵内存中的字符串, TMemoryStream 用于内存缓冲区操作, TBlobStream 用于 BLOB 字段操作, TWinSocketStream 用于在套接字连接上读写数据, ToleStream 通过 COM 接口读写数据, 这些流类都继承自 TStream 类。实际上, Delphi 本身就是利用流技术 (使用 TFileStream) 从 DFM 资源文件中读写窗体及组件属性信息的, 本文介绍的技术也就是模拟 Delphi 读写属性的方法, 不同的是属性信息保存在自定义的资源文件中, 使用的流类为 TFileStream。

### 3 实现

以下将举例说明实现技术, 主要思路是首先在窗体的 OnDestroy 事件中, 将窗体及其组件属

性保存到指定的资源文件中 (文件名由用户在 OnCreate 事件中赋值到自定义属性 ResFile 中), 然后在 OnCreate 事件中从该资源文件恢复属性, 具体步骤如下:

(1) 建立一个新项目, 在窗体 Form1 上放一个 CheckBox 组件 (做试验用)。

(2) 定义公共属性 ResFile, 用于存放自定义资源文件名称 (代码行 13, 21)。

(3) 定义并实现私有过程 PutToFile (代码行 15, 56), 用于保存窗体属性到资源文件; 定义并实现私有过程 GetFromFile, 用于从资源文件恢复窗体属性 (代码行 16, 36)。

(4) 编写窗体 OnCreate 事件代码, 指定资源文件名, 并调用 GetFromFile 从资源文件恢复窗体设置 (代码行 27)。

(5) 编写窗体 OnDestroy 事件代码, 调用 PutToFile 将窗体属性保存到指定资源文件 (代码行 32)。

(6) 运行窗体后, 选中 CheckBox, 改变窗体大小和位置, 关闭窗口再运行, 你会发现窗体及其组件都记住了它们以前的状态。

以下列出了示例窗体的单元文件, 重要代码用黑体标出并给出了注释, 本程序在 Delphi6 中调试通过。

```
1 unit Unit1;
```

```
2 interface
```

```

3 uses
4 Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms,
5 Dialogs, StdCtrls;
6 type
7 TForm1 = class(TForm)
8   CheckBox1: TCheckBox;
9   procedure FormCreate(Sender: TObject);
10  procedure FormDestroy(Sender: TObject);
11 private
12 { Private declarations }
13 FResFile: TFileName; // 保存资源文件名的
  自定义内部私有变量
14 // 保存窗体属性到资源文件的过程声明, 参
  见步骤 3
15 procedure PutToFile(AResFile: TFileName);
16 // 从资源文件恢复窗体属性的过程声明, 参
  见步骤 3
17 procedure GetFromFile(AResFile:
  TFileName);
18 public
19 { Public declarations }
20 // 存放资源文件名的属性, 参见步骤 2
21 property ResFile: TFileName read FResFile
  write FResFile;
22 end;
23 var
24 Form1: TForm1;
25 implementation
26 {$R *.dfm}
27 procedure TForm1.FormCreate(Sender:
  TObject); // 窗体的 OnCreate 事件代码, 参
  见步骤 4
28 begin

```

```

29 ResFile := 'UNIT1.FRC'; // 设置资源文件
30 if ResFile <> "" then GetFromFile(ResFile);
  // 从资源文件恢复窗体设置
31 end;
32 procedure TForm1.FormDestroy(Sender:
  TObject); // 窗体 OnDestroy 事件代码, 参
  见步骤 5
33 begin
34 if ResFile <> "" then PutToFile(ResFile); //
  保存窗体属性到资源文件
35 end;
36 procedure TForm1.GetFromFile(AResFile:
  TFileName); // 从资源文件恢复窗体属性的
  过程代码
37 var
38 Stream: TFileStream;
39 I: integer;
40 begin
41 try
42 Stream := TFileStream.Create(AResFile,
  fmOpenRead);
43 try
44 // 先删除 Delphi 从 DFM 文件建立的所有
  组件
45 for I := ComponentCount - 1 downto 0 do
46 Components [I].Free;
47 // 利用流读取属性信息
48 Stream.ReadComponentRes(Self);
49 finally
50 Stream.Free;
51 end;
52 except
53 on EOpenError do {nothing};
54 end;

```

```

55 end;
56 procedure TForm1.PutToFile(AResFile:
  TFileName); // 保存窗体属性到资源文件的
  过程代码
57 begin
58 // 保存窗体属性到资源文件, 该 Delphi 内置
  函数利用了流操作方法写入属性信息
59 WriteComponentResFile(AResFile, Self);
60 end;
61 end.

```

几点说明:

① 可以将该窗体做为通用窗体模板, 对新建窗体, 只需修改 OnCreate 事件中的 ResFile := 'UNIT1.FRC' 语句, 为每个窗体设置不同的资源文件名, 资源文件存放位置为应用程序所在目录。

② 不要用 DFM 作资源文件后缀, 以免与 Delphi 创建的窗体资源文件冲突。

③ 第一次运行窗体时, 由于资源文件尚未创建, 会出现 "Cannot Open file UNIT1.FRC" 的异常对话框(实际是被 Delphi 开发环境捕获, 若关闭开发环境的错误捕获功能, 则不会出现异常), 按 "确定" 按钮后, 再按 F9 继续执行, 程序就会创建资源文件, 以后就不会出现异常了。

④ 在设计时, 每修改一次窗体或其组件的属性, 或者增加、删除组件后, 都要手工将资源文件删除, 再运行程序, 重新创建资源文件, 否则你的修改不会在运行时反映出来(因为程序还是从老资源文件读出属性), 更好的方法是在窗体设计没有最终完成时, 将 OnCreate 事件代码中的 ResFile := 'UNIT1.FRC' 注释掉(如果已创建了 UNIT1.FRC, 也要将此文件删除), 当最终完成窗体设计后, 再取消注释。 ■