

# Visual J++ 6.0 中的 DHTML 编程技术

杨少波 (中国科学院计算技术研究所 100080)

**摘要:** 本文通过 Web 网页编程实例, 介绍在微软 Visual J++ 6.0 版集成开发平台中设计 Web 主页时的 DHTML 技术的应用、实现机制和编程要点。

**关键词:** Web 主页 动态 HTML 应用技术

## 1. 引言

Microsoft 公司新近推出的 Visual J++ 6.0 Java 语言程序开发平台, 将 Java 语言的编程高效性和 Windows 操作系统的强大功能结合起来, 是目前最快的基于数据库驱动、客户服务器模式的高性能 Windows 和 Web 解决方案的开发工具; Visual J++ 6.0 集成了对动态 HTML (DHTML) 的支持, 用户借助于其 com.ms.wfc.html 包和 Java 语言可达到控制 HTML 网页中对象属性、动态可交互效果和处理客户的交互事件, 实现 Windows 用户界面和 HTML 的用户界面的统一编程开发。

## 2. 利用 Visual J++ 6.0 生成 DHTML 应用程序

主要内容涉及如下: 生成一个内嵌到 DHTML 网页中的 COM 组件对象, 然后可实现对 DHTML 网页中组件进行编程控制; 编译和链接此程序为一个压缩并数字签名的 \*.CAB (Cabinet) 文件, 最后将此 COM 组件所在的 \*.CAB 文件内嵌到 DHTML 网页文档中。

(1) 创建对 DHTML 编程的 Java 源程序。主要步骤涉及如下:

① 在 Visual J++ 6.0 的 IDE 集成开发平台下, 选择 File 菜单中的 NewProject 子菜单, 在 Visual J++ Projects 类型中选择 Web Pages 类型;

② 选择 Code-behind HTML 项目类型, 并给出项目名称和所在目录路径位置, 最后按打开按钮。

所生成的源程序文件 Class1.java 代表 DHTML 文档, 利用它可控制 HTML Web 网页文档的内容和属性; Module1.java 代表在 HTML Web 网页文档文件 (\*.htm) 中由 <Object> 标签所指示的 COM 组件, 利用它可实现 Class1 类的加载; page1.htm 代表网页文档内容。

(2) 内嵌此 COM 组件到 HTML 文档文件中。利用 <Object> 标签加载项目中的 Module1 类, 此类已被数字签名放在 \*.CAB 文件; <Body> 标签中的 CodeClass 属性指示项目中的文档类的类名 (本例为 Class1 类名)。主

要的 HTML 文档内容如下:

```
<HTML> <HEAD> <META NAME = "GENERATOR" Content = "Microsoft Visual Studio 98">
<META HTTP-EQUIV = "Content-Type" content = "text/html">
<TITLE> Document Title </TITLE>
</HEAD> <BODY --CODECLASS = Class1 >
<OBJECT classid = "java: Module1" height = 0 width = 0 ... VIEWASTEXT>
<PARAM NAME = CABBASE VALUE = Project2.CAB > </OBJECT>
<span id = sampleSpan > </span > </BODY > </HTML>
```

(3) 编程 DHTML 的主要思路及实现机制。首先在 initForm() 内创建一个新组件对象或创建一个代表 HTML 文档中现有组件的组件对象; 其次对所创建出的组件进行事件处理和响应编程; 最后在 initForm() 内调用 setNewComponents() 加入所创建出的新组件, 并调用 bindNewElements() 绑定组件。其主要的实现机制如下:

① 产生一个代表 COM 对象的 DhModule 类 (属于 com.ms.wfc.html 包) 的派生类, 此类的实例即为在 HTML 文档中由 <Object> 标签所指示的 COM 组件对象。

② DhDocument 类 (属于 com.ms.wfc.html 包) 代表用户的 HTML 文档, 用户从此类进行派生并以覆盖方式定义出其中的 initForm() 函数。

③ DhModule 的派生类对象自动调用 DhDocument 派生类中的 initForm();

④ 用户可在 initForm() 内规定 DHTML 文档的各种动态初始特性和设置该组件的事件响应函数。

主要代码如下:

```
import com.ms.wfc.html.*; import com.ms.wfc.core.
```

```

* ;
import com.ms.wfc.ui.* ;
public class Class1 extends DhDocument
{ DhText sampleText; DhButton aNewElement;
  DhText anExistingElement; DhButton myButton;
  protected void initForm()
  { sampleText = new DhText("Class1");
    setNewElements(new DhElement[] { sampleText
});
    aNewElement = new DhButton("button");
    aNewElement.addOnClick(new DhEventHandler
(this.onClickButton));
    aNewElement.addOnDoubleClick(new DhEvent-
tHandler(this.OnDoubleClick));
    anExistingElement = new DhText();
    setNewComponents(new Component[] { aNewEle-
ment });
    bindNewElements(new DhElement[] anExistin-
gElement.persistID("sampleSpan") });
    //...其他代码定义
  }
  //...其他成员函数代码定义
  }
}
3. 编程创建组件及响应事件交互

```

(1) com.ms.wfc.html 包及有关类。此包封装了编程访问 DHTML 动态对象模型的各个类, 程序员利用此包中的类可以直接从 Java 的类代码中使用 DHTML 的有关技术, 其编程方式类同于 Windows 图形用户界面组件编程, 从而实现了 Windows 用户界面和 HTML 网页文档用户界面的统一编程方式。

其中的 DhElement 类是各个用户界面组件的父类, HTML 文档中的各个用户界面组件都从此类派生。其主要编程特点如下: DhElement 类有一个不带参数的构造函数, 从而可利用 new 直接创建生成组件对象; 调用组件的 setLocation() 成员函数可以设置组件的绝对位置, 而调用 setLocationDefault() 成员函数则可以让 HTML 布局管理器自动定位该组件。

(2) 编程创建组件。首先创建出容器组件对象, 并定位容器, 最后加入到文档中;

```

DhForm myForm = new DhForm(); //DhForm 为容
器组件类

```

```

myForm.setToolTip(" This is Container Compo-

```

```

nent");
myForm.setFont("Arial", 10);
myForm.setBackColor(Color.RED);
myForm.setBounds(5, 5, 100, 100);
add(myForm);
其次创建出容器组件中的各个被包容的组件, 并定
义各个组件的前后位置;

```

```

DhText myText = new DhText("I'm adding this in
2D space");

```

```

DhText myOtherText = new DhText("This, too!");
myText.setBounds(10, 10, 50, 50);
myText.setZIndex(2);
myOtherText.setBounds(10, 10, 50, 50);
myOtherText.setZIndex(1); //决定 Z 方向的相对
位置

```

```

myForm.add(myText);
myForm.add(myOtherText);
myButton = new DhButton();
myButton.setText("OK");
add(myButton);
myButton.addOnClick(new DhEventHandler(this.my-
ButtonClicked));

```

```

myButton.addOnMouseMove(new DhEventHandler
(this.myButtonMouseOver));

```

最后决定容器中各个组件在 Z 方向的相对位置, 从而达到一定的层次(Layout)布局。

目前 Visual J++ 6.0 中提供了两种层的编程方式:

(1) 缺省层位置(位于已加入的其他组件之上, 依创建的次序顺序排列), 这可通过调用容器类中的 add() 成员函数来实现, 如 myForm.add(myText);

(2) 强行指定某一个组件处于另一个特定的组件层之上。这可通过调用各组件本身的 setIndex(int num) 成员函数, 给定层次数 num 较大, 则处于较上层。

(3) 组件的事件响应处理。在组件创建时, 可以加入其事件响应代码(设置该组件的事件响应函数); 组件可以响应不同的事件, 这可随应用需要编程它; 利用 DhEvent 事件类的成员数据可以进一步获知更明确的事件信息。

```

void myButtonClick(Object sender, DhEvent e)
{ int MouseButton = e.getMouseButton();
  switch(MouseButton) //事件信息进一步识别。
  { case e.EVT-RIGHTBUTTON:
    myButton.setText(" You Clicke Right But-

```

```

ton");
    break;
    case e.EVT-LEFTBUTTON:
        myButton.setText(" You Clicke Left But-
ton");
    break;
    case e.EVT-MIDDLEBUTTON:
        myButton.setText(" You Clicke Left But-
ton");
    break;
}
void myButtonMouseOver(Object sender, DhEvent e)
{ myButton.setText("I can feel that mouse");
}

```

#### 4. 动态样式和动态表格

(1) 动态样式(Style)。利用动态样式,可以编程交互地动态改变 DHTML 文档中的各个组件外观,其主要的编程方法:首先创建一个 DhStyle 类的对象并设置样式的各个属性项目;其次对于采用此样式的组件调用 setStyle() 使用该样式;响应用户的交互事件,重新改变样式的属性项目,从而达到动态样式效果。

```

DhStyle myStyle = new DhStyle();
myStyle.setForeground(Color. RED);
myStyle.setFont("arial", 50 );
anExistingElement.setStyle( myStyle );
myButton.setStyle( myStyle );
//...其他成员函数代码定义
myStyle.setBackgroundColor(Color. BLUE); //重新改变样式

```

(2) 动态表格(Table)。利用 Visual J++ 6.0 不仅可以创建动态样式,也可以创建出动态表格,并根据用户的事件响应动态地改变表格中各个表格单元的内容,主要的编程方法如下。

首先创建一个 DhTable 对象(代表表格)、若干个 DhRow 对象(代表表格中的各个行)和若干个 DhCell 对象(代表表格中的各个单元格);再调用 DhTable 类中的 add() 加入各个 DhRow 对象到表格中,调用 DhRow 类中的 add() 加入各个 DhCell 对象到相应的 DhRow 对象中。

编程的主要规则:

- (1) 只能加 DhRow 对象到一个 DhTable 对象中;
- (2) 只能加 DhCell 对象到一个 DhRow 对象中;

(3) 可以加任何其他 DHTML 用户界面组件到一个 DhCell 对象中。

```

DhText Text11 = new DhText(" Row one Column
One");
DhCell Cell11 = new DhCell();
Cell11.add(Text11);
DhText Text12 = new DhText(" Row one Column
Two");
DhCell Cell12 = new DhCell();
Cell12.add(Text12);
DhButton Button21 = new DhButton(" Row Two Col-
umn One");
DhCell Cell21 = new DhCell();
Cell21.add(Button21);
DhButton Button22 = new DhButton(" Row Two Col-
umn One");
DhCell Cell22 = new DhCell();
Cell22.add(Button22);
DhRow oneRow = new DhRow();
DhRow twoRow = new DhRow();
oneRow.add(Cell11);
oneRow.add(Cell12);
twoRow.add(Cell21);
twoRow.add(Cell22);
DhTable myTable = new DhTable();
myTable.add(oneRow);
myTable.add(twoRow);
add(myTable);

```

#### 5. 结束语

Visual J++ 6.0 对 DHTML 强有力的支持,使网页设计者不仅可以利用脚本描述语言来编程 DHTML 文档,也允许直接在 Java 程序中编程控制 Web 网页中的组件,并且编程风格类同于 VisualJ++ 6.0 中的 Windows 应用程序的用户界面设计,采用快速应用程序开发 RAD 工具来设计 Web 应用。

本文所给出的程序代码在 Internet Explorer 4.0 浏览器中调试通过。

#### 参考文献

- [1] 杨少波,Java Applet 调用 ActiveX 控件的技术,计算机系统应用,1998 年第 5 期
- [2] 杨少波,Java 程序设计中的数据交换和通信技术,计算机系统应用,1998 年第 12 期

(来稿时间:1998 年 11 月)