

# 用 VB、VC 实现 MapInfo 的回调

## Using Callback Of MapInfo With VB & VC

祝晓鹰 余 锋 惠玉洁 (河南洛阳解放军 63880 部队试验技术部 471003)

**摘要:**本文介绍了如何在用户程序中通过回调机制响应 MapInfo 的事件,并提供了两个例程。

**关键词:**MapInfo 回调

### 1 概述

MapInfo 是一个被广泛应用于各行业的桌面级地理信息系统开发软件,它支持以 OLE 方式集成到用户程序中,从而实现地图的跨平台显示和编辑。当 MapInfo 集成到用户程序中时,地图窗口的消息由 MapInfo 负责处理。也就是说,发生在地图窗口的鼠标移动或点击等事件,其处理程序均在 MapInfo 内部完成,用户程序接收不到相应的消息。

但有时用户需要定制一些事件处理:比如对自定义菜单和工具按钮的响应;或者实时显示鼠标指针的地理座标等等。为此 MapInfo 提供了一种机制,当发生特定事件时,它会执行用户指定的代码,这种机制被称作回调(Callback)。

虽然也叫回调,但其实现原理和 Win32 中基于函数指针的回调截然不同,事实上,MapInfo 的回调是通过 OLE 自动化实现的:用户程序先要创建一个自动化对象作为事件的接收对象,并为它定义一个或多个 OLE 方法,然后把方法名和相关的事件传递给 MapInfo;发生特定事件后,MapInfo 将调用自动化对象中对应的方法。

### 2 MapInfo 的事件

通过回调,用户程序可以处理下列事件:

选中自定义工具按钮后,用鼠标点击地图窗口;

在地图窗口选择了自定义的快捷菜单项;

地图窗口内容发生改变,比如添加或删除图层;

状态栏内容发生改变。独立的 MapInfo 软件是一个多文档(MDI)程序,其主窗口有一个状态栏,用于显示视野、地图比例、鼠标指针的地理座标等信息。但是集成 MapInfo 没有状态栏,如果想在用户程序中实现状态信息的显示(最常见的就是显示鼠标指针的地理座标),就需要处理此事件。

### 3 实现回调的一般步骤

① 创建一个支持 OLE 自动化的类用于事件接收;

如果想接收 MapInfo 的状态信息,则为该类定义一个名为 SetStatusText 的方法,它带一个字符串参数。当状态

栏内容改变时,MapInfo 将调用此方法,把原本显示在状态栏的信息以字符串的形式传递给用户程序,如“视野: 6,000 km 编辑: 无 选中: 无”;

② 如果想在地图窗口内容发生改变时得到通知,则定义一个名为 WindowContentsChanged 的方法,它带一个长整型参数,用来标识产生变化的窗口(该标识号只能用于 MapBasic 语句,和窗口句柄无关);

③ 如果想使用自定义工具按钮或快捷菜单,则分别为它们定义一个方法,方法名自定,带一个字符串参数以接收 MapInfo 传回的数据,其格式稍后说明;

④ 用前面定义的类创建一个对象,以该对象为参数调用 MapInfo 的 SetCallback 方法;

⑤ 用 MapBasic 语句 Alter ButtonPad ID 1 Add Tool-Button ... 添加自定义工具按钮,通过子句 Calling OLE methodname 把工具按钮和回调方法关联在一起,其中 methodname 是用户自定的方法名。集成 MapInfo 的工具按钮是不可见的,需要通过用户程序界面(如菜单、按钮或工具条等),调用 MapBasic 语句 Run Menu Command ID 激活自定义工具;

⑥ 用 MapBasic 语句 Alter Menu ID 17 Add ... 添加自定义快捷菜单,通过子句 Calling OLE methodname 把菜单项和回调方法关联在一起,其中 methodname 是用户自定的方法名;

⑦ 在定义的方法中,处理 MapInfo 传递过来的参数。

### 4 回调参数的格式

当用户使用自定义工具点击地图窗口,或者选择了自定义的快捷菜单后,MapInfo 将调用和其相关联的 OLE 方法,并传递一个由“MI:”打头的字符串参数。该字符串分 8 个部分,各部分之间用逗号隔开。

如果是快捷菜单引发的事件,字符串格式为“MI:,,,,,101”,其中 101 为快捷菜单的 ID。如果是用自定义工具点击地图窗口引发的事件,字符串格式为:

“MI: - 73. 5548, 42. 122, F, T, - 72. 867702, 43. 025,

202, "

其中 -73.5548 和 42.122 为压下鼠标处的经纬度;接下来的两个字母分别表示是否按下 Shift 和 Ctrl 键(T 表示是, F 表示否); -72.867702 和 43.025 为释放鼠标处的经纬度(仅适用于拖-放模式的工具按钮); 202 为工具按钮的 ID。

MapInfo 还支持折线及多边形模式的自定义工具, 此时回调参数不包含坐标, 需要调用 MapBasic 函数 Command-Info(CMD\_INFO\_CUSTOM\_OBJ) 获取折线或多边形的顶点坐标。

请注意, 以上假定座标单位为经纬度, 在实际应用中, 座标单位和系统采用的座标系相关, 有可能为米、千米等。

## 5 举例

下面将通过两个实例, 介绍如何在用户程序中实现 MapInfo 的回调。其中 MapInfo 选用 5.0.1 中文版, 编程语言选用常见的 Visual Basic 6.0 中文版和 Visual C++ 6.0(以下分别简称为 VB6 和 VC6)。

### 5.1 用 VB6 实现回调

本例程实现两个功能: 在窗口的标题栏实时显示鼠标指针的地理座标; 创建一个自定义工具, 当用户使用它点击地图窗口时, 将弹出一个对话框显示回调参数。

运行 VB6, 创建一个“标准 EXE”工程。点击菜单“工程-添加类模块”, 在弹出的对话框中选择“类模块”, VB6 会创建类 Class1, 我们将用它接收回调事件。为该类添加两个方法 SetStatusText 和 UserTool, 前者用于接收状态信息, 后者处理自定义工具引发的事件, 代码如下:

```
Public Sub SetStatusText(msg As String)
    Form1.Caption = msg 在主窗口标题栏显示 Map-
    Info 的状态信息
End Sub
Public Sub UserTool(msg As String)
    MsgBox msg 弹出对话框显示回调参数
End Sub
```

接下来把 MapInfo 集成到程序中, 在 Form1 窗体中输入下列代码:

```
Dim map As Object  MapInfo 对象
Dim receiver As New Class1 定义并创建事件接收对象
Private Sub Form_Load()
    Set map = CreateObject("MapInfo. Applica-
    tion") 创建 MapInfo 对象
    map.Do "Set Application Window " & hWnd 设置
    主窗口
    map.Do "Set Next Document Parent " & hWnd &
```

" Style 1" 设置文档窗口的拥有者

```
map.Do "Open Table ""STATES"" Interactive"
打开地图文件 STATES.TAB
```

```
map.Do "Map From STATES" 生成地图窗口
```

```
map.Do "Set Map Display Position" 在状态信息
中包括光标位置
```

```
添加自定义工具按钮, Cursor 138 表示使用十字光标
```

```
DrawMode 32 表示支持矩形拖-放操作, 工具按钮
的 ID 设为 15
```

```
map.Do "Alter ButtonPad ID 1 Add ToolButton
Calling OLE ""UserTool"" Cursor 138 DrawMode 32 ID
15"
```

```
map.Do "Run Menu Command ID 15" 激活自定义工
具按钮
```

```
map.SetCallback receiver 设置事件接收对象
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Set map = Nothing 释放 MapInfo 对象
```

```
Set receiver = Nothing 释放事件接收对象
```

```
End Sub
```

修改工程属性, 点击菜单“工程-属性”, 在“通用”标签页中把工程类型设置为“ActiveX EXE”, 启动对象设为“Sub Main”; 在“部件”标签页中把启动模式设为“独立方式”; 在“调试”标签页设置启动工程时“等待创建部件”。

设置完工程属性后, 类 Class1 的属性窗口会多出一项 Instancing, 把它设为“5-MultiUse”, 这点很重要, 否则程序运行会出错。

由于修改了启动对象, 因此还需要定义 Main 过程, 点击菜单“工程-添加模块”生成一个模块文件, 输入下列代码:

```
Sub Main()
    Form1.Show 显示主窗体
End Sub
```

运行程序, 定位地图文件 STATES.TAB, 我们将看到地图被集成在窗体中, 同时标题栏显示出当前鼠标指针的地理座标。如果在地图窗口进行拖-放操作, 将弹出一个对话框显示相关信息, 如图 1 所示:

正常情况下, MapInfo 的初始工具为“选择工具”, 应该通过用户程序界面调用 MapBasic 语句来激活自定义工具。本文为了简化代码, 直接把 MapInfo 的初始工具设为自定义工具。如果读者想了解有关集成 MapInfo 工具按钮的详细信息, 请参考 MapBasic 帮助文档中的“Integrated Mapping - Integrating MapInfo Toolbar Buttons; Adding Toolbar Buttons and Handlers”。

## 5.2 用 VC6 实现回调

本例程实现的功能同上。

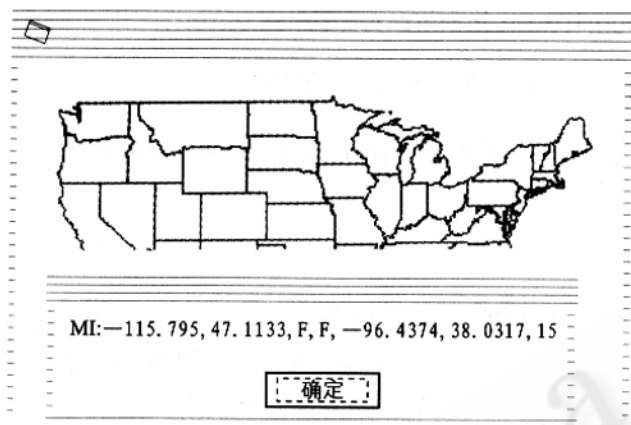


图 1

运行 VC6, 新建一个基于对话框的 MFC 应用程序 TestCallback, 生成选项使用缺省设置即可。生成后, 删除对话框中的所有控件。

用类向导从 CCmdTarget 派生一个类 CEventReceiver, 一定要选择“Automation”选项以支持自动化, 然后把它的构造函数和析构函数从保护成员改为公有成员。我们将用该类接收回调事件。

再次打开类向导, 点击“Automation”标签页, 为类 CEventReceiver 添加两个方法 SetStatusText 和 UserTool, 它们都带一个类型为 LPCTSTR 的参数 msg, 无返回值(即 void)。MapBasic 的帮助文档要求用 C/C++ 处理回调时, 方法应返回一个类型为 SCODE 的值。在实际应用中, 不要也可以。

在方法 SetStatusText 对应的类成员函数 CEventReceiver::SetStatusText 中插入语句:

```
AfxGetMainWnd()->SetWindowText(msg); //在主窗口标题栏显示 MapInfo 的状态信息
```

在方法 UserTool 对应的类成员函数 CEventReceiver::UserTool 中插入语句:

```
AfxGetMainWnd()->MessageBox(msg); //弹出对话框显示回调参数
```

接下来把 MapInfo 集成到用户程序中。打开类向导, 点击“Add Class”按钮, 在弹出的菜单中选择“From a type library”, 用 MapInfo 的类型库文件 mapinfow.tlb 产生所需的组件类, 该文件位于 MapInfo 软件的安装目录。

编辑 TestCallbackDlg.h, 为类 CTestCallbackDlg 添加两个数据成员:

```
#include "mapinfow.h" //组件类的头文件
#include "EventReceiver.h"
class CTestCallbackDlg : public CDialog
{
protected:
    DMapInfor_m_map; //定义 MapInfo 对象, DMapInfo
    //是我们刚生成的组件类之一
    CEventReceiver_m_receiver; //定义事件接收对象
    ... ..
    在类成员函数 CTestCallbackDlg::OnInitDialog 中
    插入下列语句:
    AfxOleInit(); //初始化 OLE
    m_map.CreateDispatch("MapInfo.Application"); //创建 MapInfo 对象
    char cmd[256];
    sprintf(cmd, "Set Application Window &H%p",
    m_hWnd);
    m_map.Do(cmd); //设置主窗口
    sprintf(cmd, "Set Next Document Parent &H%p
    Style 1", m_hWnd);
    m_map.Do(cmd); //设置文档窗口的拥有者
    m_map.Do("Open Table " STATES " Interactive"); //打开地图文件 STATES.TAB
    m_map.Do("Map From STATES"); //生成地图窗口
    m_map.Do("Set Map Display Position"); //在状态
    信息中包括光标位置
    //添加自定义工具按钮
    m_map.Do("Alter ButtonPad ID 1 Add ToolButton
    Calling OLE " UserTool " Cursor 138 DrawMode 32 ID
    15");
    m_map.Do("Run Menu Command ID 15"); //激活自
    定义工具按钮
    m_map.SetCallback(m_receiver.GetIDispatch
    (FALSE)); //设置事件接收对象
```

本例程为了说明问题, 专门用一个类负责接收回调事件。在实际应用中, 用户程序通常是一个单文档(SDI)程序, 地图窗口位于视图对象中。我们可以通过为视图对象添加 OLE 自动化功能, 使它既能做地图窗口的“容器”, 又可以处理 MapInfo 的回调事件, 这样有利于程序结构简洁明了。让视图对象支持自动化也很简单: 用类向导从 CView 派生一个新类, 在派生的时候选择“Automation”选项, 然后用该类替换掉原来的视图类即可。

(下转第 75 页)

(上接第 73 页)

为方便读者,本文中的例程已经放在互联网上,下载地址:<http://oldsong.nease.net/callback-sample.zip>

## 参考文献

- 1 王晓武、陈宗敏、杜兴国, MapBasic 程序设计[M], 电子工业出版社, 2000。
- 2 潘爱民、王国印 译, Visual C++ 技术内幕第四版[M], 清华大学出版社, 1999。

Practical Experience 实践经验 75