

# vbactivex控件:如何在VB中实现ActiveX控件的IobjectSafety接口-vb教程

疯狂代码 <http://CrazyCoder.cn/>     <http://CrazyCoder.cn/VisualBasic/Article15406.html>

VB中实现ActiveXControl控件IobjectSafety接口

---

## 总述

本文叙述了如何在VB中实现Control控件IobjectSafety接口以标志该Control控件是脚本安全化和安全  
VBControl控件默认处理方式是在注册表中注册组件类来标识其安全性但实现IobjectSafety接口是更好思路方  
法本言语包括了实现过程中所需所有代码

请注意Control控件只有确实是安全才能被标识为“安全” 本文并未论及如何确保Control控件安全性这个问题  
请参阅Internet Client Software Development Kit (SDK)中相关文档 \“Safe Initialization and Scripting  
for ActiveX Controls\”它在Component Development 栏目中

## 相关信息:

<此处略去了段也许无关紧要警告>

现在开始循序渐进地举例介绍说明怎样创建个简单VBControl控件以及怎样将它标识为脚本安全化和安全  
首先新建个文件夹来存放在本例中所产生文件

从VB CD-ROM取得OLE 自动化类库制作工具将VB安装光盘中  
\\Common\\Tools\\VB\\Unsupprt\\Typelib\\目录下所有内容并拷贝到前面新建项目文件夹中

把下列内容拷贝到“记事本”中然后保存到上述文件夹文件名为ObjSAFE.odl:

[

```

uuid(C67830E0-D11D-11cf-BD80-00AA00575603),
help(\ "VB IObjectSafety Interface\ "),
version(1.0)
]
library IObjectSafetyTLB
{
importlib(\ "stdole2.tlb\ ");
[
uuid(CB5BDC81-93C1-11cf-8F20-00805F2CD064),
help(\ "IObjectSafety Interface\ "),
odl
]
interface IObjectSafety:IUnknown {
[help(\ "GetInterfaceSafetyOptions\ ")]
HRESULT GetInterfaceSafetyOptions(
[in] long riid,
[in] long *pdwSupportedOptions,
[in] long *pdwEnabledOptions);

[help(\ "SetInterfaceSafetyOptions\ ")]
HRESULT SetInterfaceSafetyOptions(
[in] long riid,
[in] long dwOptionsSetMask,
[in] long dwEnabledOptions);
}
}

```

在命令行提示符下切换到项目文件夹输入下列命令创建个.tlb 文件:

```
MKTYPLIB objsafe.odl /tlb objsafe.tlb
```

在VB中新建个ActiveX Control 项目修改属性把项目命名为IobjSafetyControl控件命名为DemoCtl在Control控件上放置个按钮命名为cmdTest在它Click事件中加入句代码 MsgBox \ "Test\ "

打开菜单 “工程->引用” 点 “浏览” 找到刚刚建立Objsafe.tlb把它加入到引用中

增加个新module名为basSafeCtl并在其中加入下列代码:

Option Explicit

```
Public Const IID_IDispatch = "{00020400-0000-0000-C000-000000000046}"
Public Const IID_IPersistStorage = _
"{0000010A-0000-0000-C000-000000000046}"
Public Const IID_IPersistStream = _
"{00000109-0000-0000-C000-000000000046}"
Public Const IID_IPersistPropertyBag = _
"{37D84F60-42CB-11CE-8135-00AA004BB851}"
```

```
Public Const INTERFACESAFE_FOR_UNTRUSTED_CALLER = &H1
Public Const INTERFACESAFE_FOR_UNTRUSTED_DATA = &H2
```

```
Public Const E_NOINTERFACE = &H80004002
Public Const E_FAIL = &H80004005
Public Const MAX_GUIDLEN = 40
```

```
Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" _
(pDest As Any, pSource As Any, ByVal ByteLen As Long)
Public Declare Function StringFromGUID2 Lib "ole32.dll" (rguid As _
Any, ByVal lpstrClsId As Long, ByVal cbMax As Integer) As Long
```

```
Public Type udtGUID
Data1 As Long
Data2 As Integer
Data3 As Integer
Data4(7) As Byte
End Type
```

```
Public m_fSafeForScripting As Boolean
Public m_fSafeForInitializing As Boolean
```

```
Sub Main
m_fSafeForScripting = True
m_fSafeForInitializing = True
End Sub
```

在工程属性中把启动对象改成Sub Main确保上述代码会被执行m\_fSafeForScripting 和 m\_fSafeForInitializing两件变量值分别指定了脚本安全化和安全取值

打开Control控件代码窗口在声明部分加入如下代码(如果有Option Explicit语句当然要保证代码放在其后):

Implements IObjectSafety

把下面两个过程代码拷贝到Control控件代码中:

```
Private Sub IObjectSafety_GetInterfaceSafetyOptions(ByVal riid As _
Long, pdwSupportedOptions As Long, pdwEnabledOptions As Long)
```

```
Dim Rc As Long
Dim rClsId As udtGUID
Dim IID As String
Dim bIID As Byte
```

```
pdwSupportedOptions = INTERFACESAFE_FOR_UNTRUSTED_CALLER Or _
INTERFACESAFE_FOR_UNTRUSTED_DATA
```

```
If (riid <> 0) Then
CopyMemory rClsId, ByVal riid, Len(rClsId)
```

```
bIID = String$(MAX_GUIDLEN, 0)
Rc = StringFromGUID2(rClsId, VarPtr(bIID(0)), MAX_GUIDLEN)
Rc = InStr(1, bIID, vbNullChar) - 1
IID = Left$(UCase(bIID), Rc)
```

```
Select Case IID
```

```
Case IID_IDispatch
pdwEnabledOptions = Iif(m_fSafeForScripting, _
INTERFACESAFE_FOR_UNTRUSTED_CALLER, 0)
Exit Sub
Case IID_IPersistStorage, IID_IPersistStream, _
IID_IPersistPropertyBag
pdwEnabledOptions = Iif(m_fSafeForInitializing, _
INTERFACESAFE_FOR_UNTRUSTED_DATA, 0)
Exit Sub
Case Else
Err.Raise E_NOINTERFACE
```

```
Exit Sub
End Select
End If
End Sub
```

```
Private Sub IObjectSafety_SetInterfaceSafetyOptions(ByVal riid As _
Long, ByVal dwOptionsSetMask As Long, ByVal dwEnabledOptions As Long)
Dim Rc As Long
Dim rClsId As udtGUID
Dim IID As String
Dim bIID As Byte
```

```
If (riid <> 0) Then
CopyMemory rClsId, ByVal riid, Len(rClsId)
```

```
bIID = String$(MAX_GUIDLEN, 0)
Rc = StringFromGUID2(rClsId, VarPtr(bIID(0)), MAX_GUIDLEN)
Rc = InStr(1, bIID, vbNullChar) - 1
IID = Left$(UCase(bIID), Rc)
```

```
Select Case IID
Case IID_IDispatch
If ((dwEnabledOptions And dwOptionsSetMask) <> _
```

```
INTERFACESAFE_FOR_UNTRUSTED_CALLER) Then
Err.Raise E_FAIL
Exit Sub
Else
If Not m_fSafeForScripting Then
Err.Raise E_FAIL
End If
Exit Sub
End If
```

```
Case IID_IPersistStorage, IID_IPersistStream, _
IID_IPersistPropertyBag
If ((dwEnabledOptions And dwOptionsSetMask) <> _
INTERFACESAFE_FOR_UNTRUSTED_DATA) Then
Err.Raise E_FAIL
Exit Sub
Else
If Not m_fSafeForInitializing Then
Err.Raise E_FAIL
End If
```

```
Exit Sub
End If
```

```
Case Else
Err.Raise E_NOINTERFACE
Exit Sub
End Select
End If
End Sub
```

保存后把工程编译成OCX文件现在Control控件已经实现了IObjectSafety 接口在.htm中加入这件Control控件试试吧

2009-2-12 5:12:04

疯狂代码 <http://CrazyCoder.cn/>