BLACK ICE SOFTWARE

September Developer Newsletter

Volume 11, Issue 9 September, 2006

Inside this issue:

Phone: (603) 673-1019

Fax: (603) 672-4112

Sales@blackice.com

www.blackice.com

Custom Pop-Up Menu in Annotation SDK

When using the BiAnno dll and ActiveX control, you have the ability to modify annotation objects through a pop-up menu. This menu displays when you right click on an annotation object.

If you don't want to use the default annotation pop-up menu, you can easily change it in the latest Annotation SDK. You can now create your own pop-up menu. There are functions for creating and modifying menu items in the customized menu. The menu items can be checked and disabled, as well. In addition, you can have more than one custom pop-up menu at the same time, and you can choose which menu you want to use.

Every custom menu item has 4 properties:

Name: The displayed name of the menu item. The name can be a maximum of 64 characters. One custom pop-up menu can contain menu items with the same name.

Disabled: The menu item can be enabled or disabled. When the menu item is disabled, the user can not select the menu item.

Checked: The menu item can be checked or unchecked.

Menu Item ID: When the user selects a menu item from the custom annotation menu, the menu item ID signs what menu item was selected. If you use the Bi-anno.dll and the user selects a menu item from the custom pop-up menu, a window message will be sent. If you use the BiAnno ActiveX control, an event will be raised when the user clicks on a menu item.

The custom menu properties can be changed during run-time. For example, you can change the name of the menu item when the user selects it, or you can enable/disable the menu item whenever you want.

"Figure 1" on the next page shows a sample code snippet in VB.net. ■

How to Create a CAB File for Use with Document Imaging

This section describes creating cabinet (CAB) files for distributing ATL and MFC controls over the Internet. The CAB file contains the OCX controls, dlls for distribution, and an INF file. The INF file is a text file that specifies the files that need to be present or downloaded for your control to run. An INF file allows you to bundle all the needed files in one compressed CAB file.

For example, the sample INF in "Figure 2" on the next page will be used to create a CAB file for the BiDIB control.

This INF specifies the BiDib.dll, License_TIFF_SDK_demo.dat and License.dll files. If these files don't exist on the system, they will be downloaded from the CAB file created with this INF. "thiscab" is a keyword meaning the CAB containing this

(Continued on page 2)

Save 15% on any Black Ice Monochrome Printer Driver

&

Buy a 2D Barcode Reading toolkit, and get the 2D Writing toolkit free!

Annotation Custom Menu Sample Source Snippet in VB.Net

Figure 1

Create a custom pop-up menu

```
m_iCustomMenu = BiAnno.AnnoCreateCustomMenu()
```

Add menu item

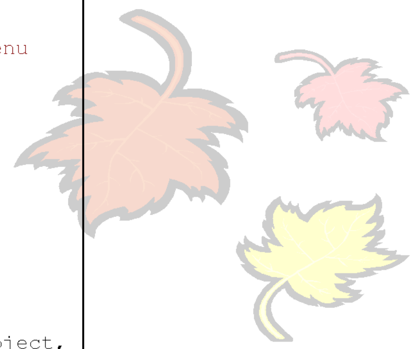
```
If Not BiAnno.AnnoAddCustomMenuItem(m_iCustomMenu,
dlg.m_szMenuItemText, Not dlg.m_bDisabled, dlg.m_bChecked,
dlg.m_iReturnValue) Then
                    MessageBox.Show("Error adding custom pop-up menu
item. Error: " + Convert.ToString(BiAnno.GetLastAnnoError()),
"Annotation Sample", MessageBoxButtons.OK, MessageBoxIcon.Error)
                    Exit Sub
                End If
```

Delete custom menu

```
BiAnno.AnnoDeleteCustomMenu(m_iCustomMenu)
```

Handling event

```
Private Sub BiAnno_CustomMenuItemClicked(ByVal sender As System.Object,
ByVal e As AxBIANNOLib._DBiAnnoEvents_CustomMenuItemClickedEvent) Han-
dles BiAnno.CustomMenuItemClicked
        If e.retValue <> 0 Then
            MessageBox.Show("Return Value: " + Convert.ToString
(e.retValue) + vbCrLf + "Object Type: " + GetTypeName(e.objectType),
"Annotation Sample", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub
```



(Creating CAB files Continued from page 1)

INF. You can also download a needed DLL from an HTTP location by specifying an absolute or relative path, for example:

```
file-win32-x86=http://
sample.comp.com/mydir/
BiDib.DLL
```

The keyword "file-win32-x86" identifies the platform as x86 specific.

The "DestDir" is the directory where the file will be loaded, "11" specifies the system directory: WINDOWS/SYSTEM or WINNT/SYSTEM32, "10" specifies the Windows directory: WINDOWS or WINNT. If no DestDir is specified (typical case), code is installed in the fixed OCCACHE directory. The "clsid" is the class ID of the control to be installed.

Once you have created an INF file, run the CABARC utility to create the CAB file. You should run the CABARC utility in the directory that contains your source files. On the command line, put the source files in the order they appear in the INF, and put the INF file last. For example, to make a CAB file for the BiDib control from the INF in Figure 2, use the following command:

(Continued on page 3)

Figure 2

```
; Sample INF file for BiDib control
[version]
signature="$CHICAGO$"
AdvancedINF=2.0

[Add.Code]
License.dll=License.dll
License_TIFF_SDK_demo.dat=License_TIFF_SDK_demo.dat
BiDib.dll=BiDib.dll
BiDib.ocx=BiDib.ocx

[License.dll]
file-win32-x86=thiscab
FileVersion=2,0,0,0
DestDir=11

[License_TIFF_SDK_demo.dat]
file-win32-x86=thiscab
FileVersion=1,0,0,0
DestDir=11

[BiDib.dll]
file-win32-x86=thiscab
FileVersion=10,0,0,0
DestDir=11

[BiDib.ocx]
file-win32-x86=thiscab
clsid={D2797899-BE27-4CDB-892F-4FDC26EA9BA9}
FileVersion=10,0,0,0
RegisterServer=yes
DestDir=11
; End of inf file
```

(Creating CAB Files Continued from page 2)

```
C:\MSSDK\BIN\CABARC -s 6144 BiDib.CAB License.dll License_TIFF_SDK_demo.dat BiDib.dll BiDib.ocx BiDib.INF
```

In this example, the INF file should list License.DLL first, then License_TIFF_SDK_demo.dat, BiDib.dll and then BiDib.OCX. The "-s" option reserves space in the cabinet for code signing. The "n" command specifies that you want to create a CAB file. For a list of CABARC commands and options, type CABARC alone on the command line. ■

Tips and Tricks About Printer Driver Messages

Q: I have a Black Ice printer driver installed. My application prints a document and gets messages from the printer driver. I would like to run multiple instances of my application using the same printer driver, but all instances get all printer driver messages. Can I make it so one instance of my application only gets the messages which belong to its printing?

A: You can not solve this problem with one installed printer driver, because of the following reasons:

The printer driver does not know which application started the printing, so it can not send any message explicitly to the printing application.

You only have three possibilities to send messages from the printer driver:

1. **Through a registered window message**. This method is NOT supported on Terminal Server systems. In this case if one application registers the windows message, it will get the messages from the printer driver. In this case if your application registers a message with the interface name of the printer driver, all running instances will get all messages from the printer driver. The instances can not decide if one message belongs to its printing or not. If you have more printer drivers installed, the interface name of the drivers are different, so one application can get messages that belong to only one printer driver. (The interface name is stored in the BlackIceDEVMODE structure.)
2. **Through the WM_COPYDATA Windows message**. This method is NOT supported on Terminal Server systems. The WM_COPYDATA message is sent by the driver to a specific window. The driver has to know the application's window handle. In this case you can specify only one window handle. This window will get all printer driver messages from the installed driver. So only one instance of your application can get printer driver messages.
3. **Through a named pipe interface**. This method is supported on Terminal Server Systems as well as non terminal server systems. In this case the printer drivers uses a named pipe interface to communicate with the application. If one instance reads the message from the pipe, the other instances can't read that pipe message.

You can read more about printer driver messages in the RTK. ■