July Developer Newsletter

BLACK ICE SOFTWARE

July, 2007

Volume 12, Issue 7

Inside this issue:

Phone: (603) 673-1019

Fax: (603) 672-4112

sales@blackice.com

www.blackice.com

How to Programmatically Create a Text Annotation Object

There have been some questions in the past about the creation of annotation objects programmatically. Often the source of the problems are the settings.

The code snippet in Figure 1 on page 2 shows the loading of a tiff file, putting a text annotation onto the image and saving the modified tiff file.

Calling the AnnoUISetExtent method is necessary before creating and burning the annotations, because it sets the background bitmap size (it is used for clipping the paint area and computing rotation and flip). If you do not set it, the default size is (0;0), so the annotations will not appear. You have to set the size of the text annotation, because if it is too small, the text can not be seen. The problem can also be caused by wrong positioning.

If you want to display other annotation objects you can follow the same steps.

Fax C++ Now on the Internet

The Fax C++ toolkit has a new feature! We've added an internet sample, and we've created a cabinet file you can use on your website to remotely host the program. There is a sample of this new feature at:

```
http://
www.blackice.com/
FAXCPP_HTML_SAMPLE/
```

Why would you want to use this technology? For example, maybe your company uses a web server to host the software which your employees use. Your communications department has to use faxing machines to fax important letters, but now they may be able to fax those letters directly from your web application with only a fax modem!

Internet Explorer supports using ActiveX controls, and Fax C++ takes advantage of that feature now. You can insert the control into your web application very easily, and then you can use JavaScript or VBScript to interact with it.

The sample makes use of the TIFF SDK in order to show a preview of the TIFF file to be faxed. You can simply browse for a tiff file, and click the load picture button to load that picture. If you have the TIFF SDK installed, it will automatically display the picture. You may switch between pages of the picture with the previous/next buttons.

In order to send the loaded picture, you need to enter the com port that the fax modem you wish to send with is connected to. You also need to supply the telephone number you wish to fax to. The sample only supports COM ports in order to keep its code simple, however the Fax C++ SDK supports other faxing boards as well (like Brooktrout , Dia-

(Continued on page 2)

(Fax C++ on Internet Continued from page 1)

logic). When you've finished entering the parameters, just click the "Send the FAX" button, and the sample's sending the fax! You can see the progress of the faxing session in the DOS-like console window that opens.

In order to host the faxing web application on a remote server, so not everyone would need to have a Fax C++ installation directly on their machines, you need to use the cab file that is now added into the FaxCpp32\Cab directory. In that directory, you will find a .cab and .lpk file. You need to copy both onto your server into the same directory, and when you add the <object> tag into your HTML code, you can reference the .cab as below:

```
<object clas-
sid="CLSID:2E980303-C865-11CF-
BA24-444553540000" code-
```

Figure 1

```
String sourFileName = "test.tif";
String destFileName = "testOut.tif";

// load the source tiff file
int hDib = axBITiff1.LoadTiffIntoDIB(sourFileName, 0, false);

// get the size of the image
int width = axBIDIB1.GetDIBWidth(hDib);
int height = axBIDIB1.GetDIBHeight(hDib);

// set the background bitmap size
axBiAnno1.AnnoUISetExtent(width, height);

// create the text annotation object
axBiAnno1.CreateAnnoObj((short)BIANNOLib.ObjectType.aotText);

// set the text annotation object
axBiAnno1.AnnoObjSetPos(10, 10);
axBiAnno1.AnnoObjSetSize(width, 800);
axBiAnno1.AnnoObjSetColor(0);
axBiAnno1.AnnoObjSetText("Test Test Test Test Test Test Test");
axBiAnno1.AnnoObjSetFont(100, 50, 0, 0, 100, false, false, false, 0, 0, 0,
0, 0, "Arial");

// burn the text into the image
int nNewDib = axBiAnno1.AnnoBurnin(hDib);

// delete annotation objects from the page
axBiAnno1.DeleteObjectsFromPage(0);

// save the modified image
axBITiff1.SaveDIBInTiffFormat(destFileName, nNewDib, (short)
BITIFFLib.enumCompressionTypes.TCOMP_NOCOMP, false);
```

How to Create a General Message Capture Application

If you need to create a program for catching PIPE messages of the printer driver that works on NT and terminal server operating systems, you need to know the following differences.:

Using the *Blicectr.dll* for getting messages.

You can use the *Blicectr.dll* for example if you develop applications in C++. You can use the same version of the *Blicectr.dll* on NT and TS operating systems. But the name of the PIPE is different on NT and TS. Before you call the *WaitForPrnPipe* you have to generate the name of the PIPE properly:

- On NT operating systems:

```
wsprintf(szPipeName,
L"\\\\.\\pipe\\%s",
DEVMODE_GetInterfaceName
(lpDevMode));
```

- On TS operating systems:

```
wsprintf(szPipeName,
L"\\\\.\\pipe\\%s%d",
DEVMODE_GetInterfaceName
(lpDevMode), SessionID);
```

As you can see the name of the PIPE contains the ID of the session on TS. So if you want to create a general message capture application you should detect the operating system first. After that you can create the name of the PIPE properly.

Using the *BiPrnDrv.ocx* for getting messages.

You can use the *BiPrnDrv.ocx* for example in C#, J# or VB .NET applications. The *BiPrnDrv.ocx* calls the *Blicectr.dll*. The *BiPrnDrv.ocx* creates the name of the PIPE internally. Because of this there are 2 versions of the *BiPrnDrv.ocx*. One for NT and one for TS operating systems. So if you want to create a general message capture application you should install the proper version of the *BiPrnDrv.ocx* and *Blicectr.dll* to the target machine. After that you can use the same PIPE message capture application on every operating systems.

Printing Tips: How the Built-in Print Dialog Works on Windows Operating Systems

Question: Why changes the resolution value when I try to print to non default Black Ice printer driver from Notepad?

Answer: This issue doesn't depend on the printer driver or the printing application. The resolution is changed when you try to print with an other printer (for example HP LaserJet) from another printing application (for example Internet Explorer) too.

The Microsoft's Print dialog changes the common printing settings when you change the printer selection on the built-in print dialog before printing the document.

Let's see the following example:

We have 3 installed printers on our system:

Black Ice TIFF

HP LaserJet 1010

Microsoft Office Document Image Writer

The default printer is the *Microsoft Office Document Image Writer.*

In our test we will check 2 common printer settings: **Paper size** and **Resolution**.

Here is the list of **Paper size** and **Resolution** settings of the printers:

Black Ice TIFF: Letter, 200x200

HP LaserJet 1010: Letter, 600x600

Microsoft Office Document Image Writer: Legal, 300x300

Open a text document with notepad.

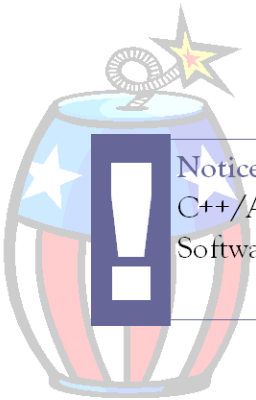
Select File\Print

In this case the *Microsoft Office Document Image Writer* is the selected printer in the built-in Print dialog, because it is the default printer of the system. If we select the *HP LaserJet 1010* and click the Preferences button, the resolution will be 300x300 and the paper size will be Legal, because the Print dialog changes the settings depending on *Microsoft Office Document Image Writer.*

We changed the resolution to 600x600 and paper size to A4 of the *HP printer* and click OK. After that we selected the *Black Ice TIFF* printer and clicked Preferences button. The print dialog changed the setting of the *Black Ice TIFF printer* depending on *HP printer*. So the paper size is A4 and the resolution is 600x600. But if we selected the *Microsoft Office Document Image Writer* printer again instead of *Black Ice TIFF*, the settings of the *Microsoft Office Document Image Writer* will be changed to 600x600 and A4.

You can avoid this issue if you check the printing preferences before every printing or if your printer driver is the default and

you don't change the selection of the printer in the print dialog. In this case the printing settings won't be changed before printing.



Notice: The Printer Drivers for Windows 95 / 98 / ME, and the Voice C++/ActiveX Toolkit will be discontinued at the end of this year. Black Ice Software will no longer support these products after January 1st, 2008.