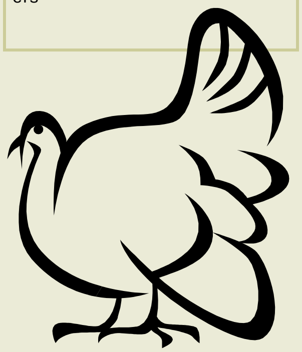
Black Ice Software

November Developer Newsletter

November 2008

Volume 13, Issue 11

Inside this issue:



The BLACK ICE NEWSLETTER is published by Black Ice Software, LLC. The contents of this newsletter in its entirety are Copyright © 2008 by Black Ice Software, LLC. 20 Broad St, Nashua NH 03064, USA. Black Ice Software, LLC. does hereby give permission to reproduce material contained in this newsletter, provided credit is given to the source, and a copy of the publication that the material appears in is sent to Black Ice Software at the above address.

Phone: (603) 882-7711

Fax: (603) 882-1344

sales@blackice.com

www.blackice.com

Black Ice PDF SDK Version 2.50 is Released

Black Ice Software announces the release of a fully functional PDF Software Development Tool Kit to read write and edit PDF documents.

New easy to use features were added to PDF SDK to create new PDF documents and modifying existing ones.

The Black Ice PDF SDK now contains 4 modules in which one may build a PDF enabled application:

- The Low Level PDF Reader and Writer Interface provide a high speed and well very freely programmable C++ interface for the PDF SDK.
- The PDF Reader and Writer ActiveX module gives the possibilities of the Low Level Interface for PDF SDK C++ library to many other programming languages, such as .NET and Delphi.
- The High Level Interface for PDF SDK is a fully object oriented system that gives a programmer many powerful tools to create an efficient PDF Editor application in C++.
- The PDF Editor ActiveX control gives a very easily usable control for any language supporting ActiveX controls. Just import the control into your developer environment and in only a couple of clicks and a few lines of code, one's got a PDF editing application!

Low Level Interface for PDF SDK

Until now the PDF SDK has only supported reading PDF documents. The new features that were added to the PDF SDK support creating new PDF documents and modifying existing ones too.

There are three well-separable modules in the PDF SDK:

- Reader module: contains the PDF loading routines and the rasterizing of the pages of the document. This module contains the text searching as well.
- Writer module: supports PDF document creation and modification. It supports settings of the document such as passwords, encryption (passwords), compression, font embedding, metadata (title, subject, author, creator, keywords); and page settings such as page size and page contents (text, images and other vector graphical shapes)
- Common module: that supports getting the PDF document's global properties.

The new PDF SDK release supports both the Reader and the Writer SDK's features and more.

If one wants to develop an application that supports viewing PDF documents and perhaps want to search text, one should use the Reader module. If one wants to develop an application that can create PDF files the Writer module will offer the necessary features. For creating a more complex program however, it is recommended to use both modules. For example if one wants to modify PDF files it requires loading, modifying and writing the PDF document. This SDK supports all of these operations by using one CPDF object that makes working with PDF documents easy.

The PDF SDK contains both a dynamic link li-

(Continued on page 2)

What is the High Level Interface for PDF SDK?

Black Ice Software's newly released PDF SDK contains PDF viewing and writing functionality. It contains an exciting module as well: the High Level Interface. Black Ice gives this module free of charge with the Black Ice PDF SDK. This module allows you the quick and simple creation of PDF Viewing and Editing applications in C++.

The High Level Interface can load the contents of a PDF document and its pages into a fully object oriented data structure and it can modify and save/unload that data structure easily. When the data structure is loaded, programmers can call a range of functions on it that help much when developing a PDF viewer or editor program. One such feature is the object rendering system. It can be put to action only by calling the Draw function of a page object after having loaded a page. While this feature is so easy to use it also supports a multitude of settings such as allowing users to use anti-aliasing and giving the ability to cancel rendering at any time (for multi threading). The High Level Interface also contains an object targeting system allowing the programmer to pass only the coordinate of a mouse click to get the clicked object(s).

The High Level Interface – as the name suggests – is an interface for the PDF SDK through which a programmer can interact with the Low Level Interface for PDF SDK itself. The base class of the High Level Interface is the CPDFInt. Instantiating an object of this class also associates it with a new or an existing PDF document and it manages the embedded Low Level Interface object. One can then create the main object oriented data structure by calling one function of CPDFInt, and it will create a PDF document object containing the document's properties and pages within itself. The programmer is then free to load any page's contents and display or edit them.

The SDK contains a PDF Editor sample, which uses the High Level Interface and shows the usage of the High Level Interface in the well-known MFC Document/View architecture. The sample includes the following features:

- Creating a new PDF document or loading an existing one
- Displaying the PDF document page by page with the PDF Interface's advanced rendering engine
- Viewing and editing the PDF document information, password/encryption and compression settings
- Adding new objects onto a PDF page. Supported object types are:
  - Text
  - Image
  - Rectangle
- The PDF Interface itself supports many more objects!
- Modifying existing objects on a page. (Fully supporting the previously mentioned types and sup-

(Low Level Interface Continued from page 1)

brary (dll) and an ActiveX control, so it is easy to use in C++ (with dll functions) and in all of the programming languages supporting ActiveX controls.

Some words about the Writer module:

During the development of this module it was the most important goal to implement methods that are as easy as possible to use. A PDF document contains page objects and these pages contain the visual contents. The writer module uses a class of such page objects as well with all needed methods to add any kind of object onto the PDF page. The page class's interface is very similar to the MFC CDC class's interface, so if one has already made any programs using the Microsoft GDI, getting used to the PDF Writer will take no time! It is almost exactly like you were using a CDC object!

There are also some extra methods of the CPDFPage class for example to support the BITMAP and DIB image structures more easily (StretchDIB, StretchBitmap, BltDIB, BltBitmap). These functions make it even easier to use this SDK and other Black Ice SDKs together. One can manipulate an image with the Document Imaging methods and the output can be used as a content of a PDF page.

After one has created a PDF document, inserted the pages into it and loaded the content into the pages, one may set the properties of the whole document before save it.

The Writer supports the following settings which will also be the topics of later newsletter articles:

- Metadata that contains the document properties.
- Encryption, owner and user password.
- Compression of the different images and other streams.
- Embedding the different fonts.

How to Programmatically Specify the Name of the Message Interface on Terminal Servers

We have already discussed the goal of setting manually the session ID of the printer driver in the October newsletter. Now we demonstrate how you can specify programmatically the session ID.

You can determine the session ID of the printer driver with the *SetSessionID* method of the BlackIceDEVMODE. There is a *GetSessionID* method to get the current session ID that is used by the printer driver for generating the Message Interface name. After you set the session ID you can use the specified session ID in your Message Capture application.

The following code snippet shows how to set the session ID in the C# sample using the BlackIceDEVMODE ActiveX control:

```
// Load DEVMODE

int iDevmode =
BlackIceDEVMODE.LoadBlackIceDEVM
ODE(m_szPrinter.ToString());


if (iDevmode == 0)

{

    // Error

    MessageBox.Show("Error load-
ing the DEVMODE of " +
m_szPrinter + ".", "SessionID C#
Sample", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

    Application.Exit();

}


bool bRet =
BlackIceDEVMODE.SetSessionID
(iDevmode, 2);


if (!bRet)

{

    MessageBox.Show("Error set-
ting the session ID of " +
m_szPrinter + ".", "SessionID C#
Sample", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);

    return;

}


// Save the devmode

BlackIceDEVMODE.SaveBlackIceDEVM
ODE(m_szPrinter.ToString(),
iDevmode);

// Release devmode structure and
free allocated memory

BlackIceDEVMODE.ReleaseBlackIceD
EVMODE(iDevmode);
```

You can set the sessionID through printer driver's INI file at installation time. For example:

```
[Default Settings]

Interface-
Name=ColorPlusMessageInterface

; Session ID for generating the
MSG interface name

SessionID=2
```

After the session ID is set in the printer driver the Message Capture application is able to use the specified session ID. The following code snippets demonstrate how to set the session ID manually in a C++ and a C# application.

- In the C++ PIPE Message Capture application we use the Blicectr.dll. The Message Interface name is parameter of ***WaitForPrnPipe*** function:

```
WCHAR szPipeName[MAX_PATH];

DWORD SessionID = 2;


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

WaitForPrnPipe(szPipeName, hwnd,
WM_USER+1000 );

SetListeningPriority
( THREAD_PRIORITY_TIME_CRITICAL
);
```

- In the C# PIPE Message Capture application we use the BiPrnDrv.ocx. The BiPrnDrv.ocx has a new property to specify the session ID. The name of the new property is **SessionID**.

```
BiPrnDrv.SessionID = 2;

// MESSAGE_CAPTURE_METHOD_PIPE =
3

BiPrnDrv.StartCapture(szPrinter,
3);

BiPrnDrv.PrinterName =
szPrinter;
```

Note: The session ID is part of the Message Interface name only in TS printer drivers. In case of NT printer drivers you don't need to specify the session ID.

BLACK ICE
SOFTWARE, LLC

20 Broad St
Nashua, NH 03064

Phone: 603-882-7711
Fax: 603-882-1344
E-mail: sales@blackice.com

Time to upgrade? Latest Version Numbers

Development Tools

Printer Drivers	10.23	08/01/08
Fax & Voice C++/ActiveX	12.50	06/18/07
Document Imaging SDK/ActiveX	10.95	04/25/08
Image PDF Plug-in	10.95	04/25/08
PDF SDK/ActiveX Professional	2.50	11/01/08
Annotation SDK/ActiveX	10.95	04/25/08
Image SDK/ActiveX	10.95	04/25/08
Tiff SDK/ActiveX	10.95	04/25/08
Cover Page Generator SDK/ActiveX	10.95	04/25/08
Barcode SDK/ActiveX	5.10	05/07/07

Impact Products

Impact Fax Server	8.03	05/21/08
Impact Fax Broadcast	6.5	07/02/08
Impact ColorFax	8.02	02/25/07
Print2Email	7.10	04/25/08
Tiff Viewer Plug-in - Complete	8.13	09/04/08
Print Monitoring Server	4.10	03/14/08
Print2RDP	4.80	09/04/08
Print2FTP	2.02	08/15/06
FileMorph	2.13	05/20/08
FileMorph DS	2.12	04/25/08

Free Software

Impact ColorFax Lite	8.02	02/25/07
Tiff Viewer Plug-in - Free Version	8.13	09/04/08
ModemWeasel	2.00	8/01/02