

MARCH NEWSLETTER

Volume 10, Issue 3
March 2005

CUSTOM PLUG-IN SUPPORT ADDED TO THE BLACK ICE PRINTER DRIVERS

Inside this issue:

Custom Plug-In Support added to the Black Ice Printer Drivers	1
An Inside Look at the Fax C++/ActiveX	1
New Fax C++/ActiveX Retry Sample	2
New TIFF Tag Manipulation Sample Application	3

The BLACK ICE NEWSLETTER is published by Black Ice Software, Inc. The contents of this newsletter in its entirety are Copyright © 2005 by Black Ice Software, Inc. 292 Route 101, Salzbürg Square, Amherst, NH 03031, USA. Black Ice Software, Inc. 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) 673-1019
Fax: (603) 672-4112
E-mail: sales@blackice.com
www.blackice.com

Add a Custom Plug-in to the Black Ice Printer Drivers! Version 9.10 of the Black Ice printer drivers includes new Plug-In support which is accessed through a dynamic linked library or DLL, and is loaded by the printed driver during printing. The printer driver calls pre-defined functions from the Plug-in at each phase of printing, and will not continue printing

until the function returns. Developers will have full access to the DLL and can insert their own custom code to each DLL function to perform a variety of tasks as required by their application.

The Black Ice printer drivers can convert any file printed from any Windows application into an image or PDF file, which can then be faxed,

archived, etc. as necessary. Print Jobs can be submitted to the Black Ice printer drivers from either custom applications or from typical applications, such as Notepad or Microsoft Office applications. When using custom applications, developers can modify their printing application to perform exactly as they require, however when

(Continued on page 3)

AN INSIDE LOOK AT THE FAX C++/ACTIVE X

Internal Fax Queue

The Fax C++/ActiveX allows developers to send faxes in two unique ways:

1) By sending a fax immediately on an available port or channel or 2) By adding the fax to the internal fax queue. This intent of this article is to provide a detailed overview of the Fax C++/ActiveX fax queue and its benefits.

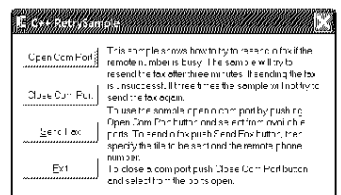
How it Works:

The Fax C++/ActiveX internal fax queue works the same as any queue, the first item added to the queue is the first item out of the queue. Fax Objects can be added to the queue by the developer's application, and the queue can accept and store any number of fax objects, and once the fax object has been added to the queue the application can essentially forget about them

and let Fax C++/ActiveX manage the fax objects.

Fax C++/ActiveX will take the first fax from the queue and will send it on the first available port or channel. As soon as the

(Continued on page 2)



New Retry Sample added, see page 2 for details

(Continued from page 1)

next port or channel becomes available, Fax C++/ActiveX will take the next object, and will continue in this manner until the queue has been emptied

Variations of the Queue Order

The order in which fax objects are added to the queue and subsequently removed from the queue for faxing can be manipulated by changing a given fax object's priority. The fax priority functionality was first released in version 11.04 of the Fax C++/ActiveX and allows developers to assign any positive integer value to a fax object. The higher the priority value, the more important that object is and the further up the queue it moves. The fax queue will automatically move higher priority faxes up the queue to their correct priority based position.

Tracking Objects in the Queue

Fax objects in the queue can be tracked in a number of ways, however the most flexible and powerful tracking method is through the use of the User Strings, also released in version 11.04 of the Fax C++/ActiveX. User strings are a property of each fax object and allow developers to store a string of information along with a particular fax object. Some possible uses can include storing the faxing users' name, a tracking number, etc. Developers can locate objects in the queue by searching for a particular user string and upon locating the string can update some property of the object (fax number, priority, etc.), they can even delete the object entirely if so desired

Retrying an Unsuccessful fax

Fax C++/ActiveX internally handles sending faxes on multiple ports by queuing the faxes and sending out the faxes on the first available ports.

If the called phone number is busy, a human answer is detected, or some similar error occurs then the fax should be resent after a waiting period. The waiting period ensures or at least makes it more likely that the phone number will not be busy the next time the fax is sent.

This retry feature is not directly implemented in Fax C++/ActiveX since the retries will be handled differently by every application, however implementing a retrying mechanism is very simple.

Black Ice recently added a new sample to the Fax C++ /ActiveX toolkit which demonstrates how to implement retrying in the case of a busy fax number.

The sample will try to resend an unsuccessful fax in three minutes. If sending the fax is unsuccessful again, the sample will retry two more times.

The current version of the sample is implemented only for modems, however the retry can be implemented exactly the same way for other fax hardware such as Brooktrout, GammaLink, NMS or Dialogic boards.

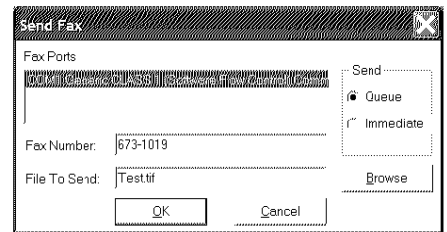
Sending a Fax

The retry sample is implemented for both queued faxes and faxes sent immediately.

If "Immediate" mode was used and remote number is busy the sample will try to resend the fax after three

minutes. It is important to note that the sample will try to send the fax on the same port as initially selected.

If "Queue" was checked and remote number is busy the sample will try to put the fax back into the queue after three minutes. If the remote number is not busy next time, the fax will be sent on the first available port.



The sample uses the following logic to implement the retries:

1. The Fax Object is placed in the Fax C++ queue for sending
2. The TERMINATE event is received, the fax number was busy
3. The Fax Object is removed from the Fax C++ queue and it is added to the retry queue of the sample.
4. The timer is started
5. When the previously established time period expires (3 minutes) the Fax Object is removed from the retry queue and it is added to the Fax C++ queue
6. Fax C++ will send the Fax Object on the first available port when the object reaches the top of the fax queue.
7. Steps 2 through 6 are repeated as many times as necessary (number of retries)

New Impact Fax Server 6.0 coming in March!



Visit www.blackice.com for more details and to download a fully functional evaluation version of new Impact Fax Server 6.0!

(Continued from page 1)

printing from a third party application, developer's are at the mercy of the application vendor and the printing user. Until now...

How It Works

The Black Ice printer driver will load the Plug-In DLL with each print job and will pass to each DLL function a pointer to the devmode of the current print job. When the print job starts, at the Start Document phase, the Plug-In DLL can pass custom data to the printer driver. The custom data will be stored by the driver for the duration of the print job. The size of the custom data is not limited by the printer driver, however using a large data block is not recommended since the printing can be negatively impacted. Specifying a data block of no larger than 10-20 KB should be sufficient in most cases and will not have an impact on performance. The printer driver will return a pointer to the custom data in each additional function call to the Plug-in DLL. The Plug-in DLL can then modify or use the custom data in any way desired.

The Plug-in DLL functions must return TRUE on success or FALSE on any failure. If any of the custom functions return FALSE, the Black Ice Printer driver will abort the current print job.

Plug-in DLL functions:

BiStartDoc:

The BiStartDoc() function will be called once for each print job, when the printer driver starts generating the images. The printer driver will pass the DEVMODE to the function and will allocate memory for the custom data passed back by the function.

BiStartPage:

The BiStartPage () function is called each time the

printer driver starts processing a new page. The printer driver is passing the devmode and the custom data to the function.

BiEndPage:

The BiEndPage () function is called each time the printer driver finishes processing a page. The printer driver will pass the DEVMODE, the custom data and the current page number.

BiEndDoc:

The BiEndDoc() function is called one for each print job when the printer driver each time the printer driver finishes processing a document. The printer driver will pass the DEVMODE, the custom data and the name of the group file. After the function returns the memory allocated for the custom data will be de-allocated.

BiAbort():

The BiAbort() function is called if the print job is aborted. The print job can be aborted by the user, or by the printer driver if an error occurs.

Some Possible Uses

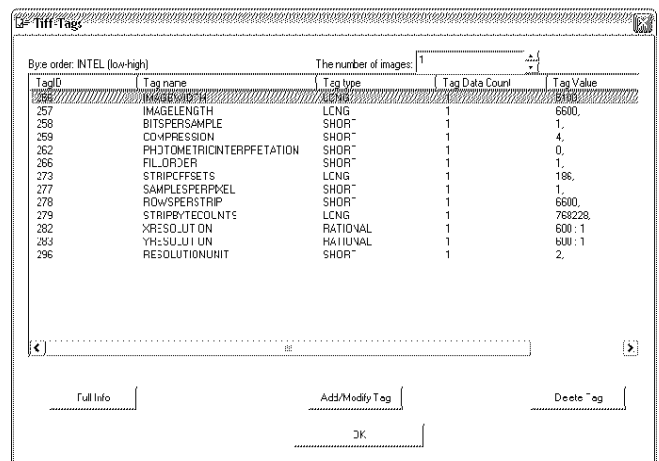
- Print Job Authorization – prompt for a username, password, etc. from the BiStartDoc function
- Display a custom database window for archiving applications – prompt for a database name, JobID, etc. from the BiStartDoc function
- Change printer driver settings on the fly (output directory, filename, etc)
- Alternate form of message handling - the printer driver will call a function of the DLL for each printing event which can be used to notify a developer's application of the current printing status.

NEW TIFF TAG MANIPULATION SAMPLE APPLICATION

The Black Ice Imaging Toolkits allow developers to perform a variety of imaging tasks, including reading and manipulating TIFF tags.

While TIFF Tag manipulation has always been available in the Black Ice Imaging Toolkits, as part of the C++ samples, previously there was no clear cut demonstration of this functionality using the Black Ice Imaging ActiveX controls.

The latest release of the Black Ice Imaging Toolkits, version 10.33, includes a new TIFF Tag sample written in Visual Basic which provides a clear cut example of this functionality.



•292 Route 101•
 •Amherst, NH 03031 USA•
 Tel: (603) 673-1019
 Fax: (603) 672-4112

Inside

New Printer Driver Plug-In Support,
 Fax C++/ActiveX Retry Sample and more

Time to upgrade?



Latest Version Numbers

Printer Drivers

Product	Version	Date
Printer Drivers for ME/98/95 Monochrome, Color, Metafile and PDF	5.65	10/28/04
Printer Drivers for XP/2000/NT Color, ColorPlus, Metafile, Monochrome and PDF	9.10	2/24/05
Printer Drivers for 2003, Citrix and Ter- minal Servers Color, ColorPlus, Metafile, Monochrome and PDF	9.10	2/24/05

Fax, Voice and Image Toolkits

Product	Version	Date
Fax C++/ActiveX/COM	11.06	02/04/05
Voice C++/ActiveX/COM	11.06	02/04/05
Tiff SDK/ActiveX/COM	10.33	02/22/05
Annotation SDK/ActiveX/COM	10.33	02/22/05
Cover Page Generator SDK/ActiveX	10.33	02/22/05
Image SDK/ActiveX	10.33	02/22/05
Document Imaging SDK/ActiveX	10.33	02/22/05
Barcode SDK/ActiveX	4.10	10/04/04

Impact Products

Product	Version	Date
Impact Fax Server	6.00	02/28/05
Impact Fax on Demand	4.06	12/01/00
Impact Fax Broadcast	4.30	12/13/04
Impact ColorFax	6.00	07/22/04
Drag and Drop Printing	2.00	12/24/02

Free Software

Product	Version	Date
Impact ColorFax Lite	6.00	07/22/04
Tiff Viewer Plug-in - Free Version	6.00	07/22/04
ModemWeasel	2.00	08/01/02

Internet Tools

Product	Version	Date
Print2Email	4.00	02/14/05
Tiff Viewer Plug-in - Complete	6.00	07/22/04
Print Monitoring Server	3.00	09/16/04