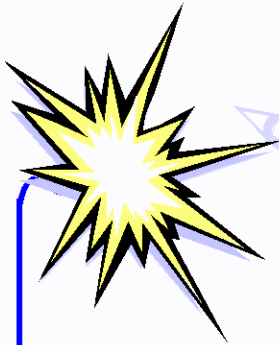


JULY DEVELOPER NEWSLETTER

BLACK ICE SOFTWARE

Inside this issue:

Printer Driver Sharing Utility	1
Using ePrintAnywhere to map black Ice Printer Drivers onto Terminal Servers	1
PDF Toolkit - Save Image to PDF in C#	3



The BLACK ICE NEWSLETTER is published by Black Ice Software. The contents of this newsletter in its entirety are Copyright © 2006 by Black Ice Software. 292 Route 101, Salzburg 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

Sales@blackice.com

www.blackice.com

Printer Driver Sharing Utility

Until now, it was not possible to share a Black Ice Printer Driver across a network on a different Windows operating system from where the Black Ice Printer Driver is installed. For example: if you installed a Black Ice Color Printer Driver to a Windows XP machine and shared the printer, you could only use this printer on other XP and Win2000 operating systems, you could not use the shared printer on a Windows 2000 Server, or 2003 Server.

The new Printer Driver Sharing Utility is available with the purchase of a Black Ice Printer Driver for using the shared printer on different Windows operating systems. You can use a shared XP printer driver on a Terminal Server operating system and vice versa. The utility doesn't support the Windows 9(X) and Windows ME printer drivers.

Using the Printer Driver Sharing Utility

The steps below demonstrate how to use the

Printer Driver Sharing Utility.

For example, when sharing a Color printer driver installed on a Windows 2003 Server on a Windows XP operating system:

1. Share the Color Printer Driver on the Windows 2003 machine. Call this machine "server."

2. Run the Printer Driver Sharing Utility on the Windows XP machine. Call this machine

(Continued on page 2)

J U L Y S A V I N G S !

Printer Driver 20% off
with Maintenance

Using ePrintAnywhere to Map Black Ice Printer Drivers onto Terminal Servers

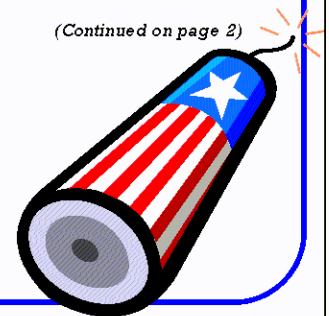
ePrintAnywhere is very popular with Terminal Server and Citrix Administrators due to the incredible ease with which they can map local client printers onto their Server. It requires no driver management by the Administrator and allows you to map

drivers that are not normally Terminal Server/Citrix compliant.

Software developers can use Black Ice Software's ePrintAnywhere application to map locally installed Black Ice printer drivers (or just about any other printer driver)

onto a Windows 2000 or

(Continued on page 2)



(Sharing Utility from page 1)

“client.”

The “YOUR COMPANY NAME” string contains your company as is indicated in the About box of your driver. The utility may only be used with the Black Ice Printer Driver it was purchased with.

Click the Next button.

3. Please note: If the utility is running on the Terminal Server, the installation of kernel level printer drivers must be enabled. At this point you have to connect to the server and click on the shared printer. Enable the installing of the printer driver automatically, if it's needed.

Click the Next button.

4. Select the newly installed printer from the “available printers” combo box. Once you have selected the printer click the Finish button.

The utility will then make some configuration changes to the driver.

Click the Finish button to close the

utility.

Shared Printer Driver Restrictions

The printed images will be generated on the server, not on the client.

On the client printer driver you must properly configure the following settings for it to work properly:

-Output Directory (on the Filename Generation tab): You have to select an output directory that exists on the server. If this directory does not exist on the server, the printed image is generated and placed in the TEMP directory for the machine.

If the output directory is a shared network directory (for example \\aaa\ccc), and the printed image is generated in the TEMP directory on the server, then the port monitor does not have read/write permission to generate the image in the network directory.

-Enable Start Application (on the Start Application tab): If you select a start application in the client printer, and this application exists on the server at the same location, the se-

lected application will run. This application won't run on the client machine.

-Watermarks (Watermark tab): If you select a watermark on the client machine, these files must exist on the server machine with the same full path.

If the watermark image doesn't exist on the server, the printed image will not contain the watermark.

-Using bates numbering (Bates Numbering tab): If you enable bates numbering and select a format from the combo box, the name of the format must exist on the server printer driver. If the format does not exist with the specified name on the server printer driver, the Bates number won't be added to the printed image. The printed image will contain the Bates numbering, however it will be in the format that the server's driver is configured for.

-Profile Manager (the Profile Manager tab): The management of the printer setting profiles does not work when the driver is shared.

(ePrintAnywhere Mapping from page 1)

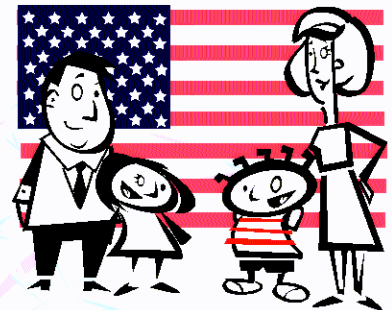
2003 Terminal Server or Citrix Server. This can be very useful when you want the document conversion to be completed and stored locally instead of only on the Terminal Server.

Additionally, using the redirect printing option of the Black Ice drivers and ePrintAnywhere, your user can print to the Terminal Server driver and the converted document can be stored on the Terminal Server and

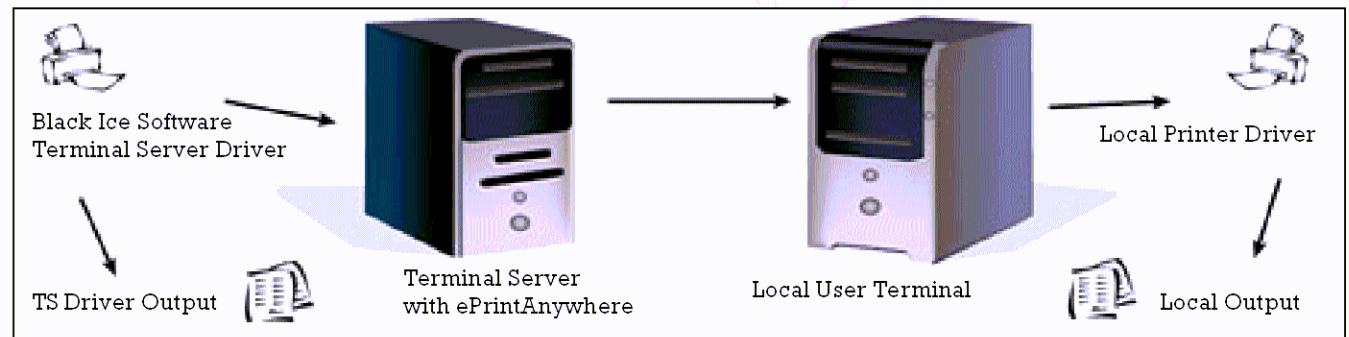
also redirected to the User's local driver allowing for the document to be archived on the Terminal Server and sent to the User's local machine or printed to the User's local printer.

For more information about ePrintAnywhere or to download the free fully functional 15 day demo of ePrintAnywhere visit

<http://www.blackice.com/ePrintAnywhere.htm>



Happy Fourth of July



PDF Toolkit - Save Image to PDF in C#

One way to save an image file to PDF format is by loading the image file into a DIB, and then saving the DIB as a PDF file. The Document Imaging Toolkit provides several ActiveX methods which load different types of image files into a DIB. For example: In BiImgFrm.ocx: LoadGIFIntoDIB and LoadJPEGIntoDIB methods, in BiDib.ocx: LoadDIBFromFile

method, in BiTiff.ocx: LoadTiffIntoDIB method. A DIB can be saved as a PDF file by using the SaveDIBAsColorPDFPage method (in BiPdf.ocx).

The following code shows the InpExample.TIFF file (this file must be created by the user) loaded into a DIB then the DIB is saved into the OutpExample.PDF file

(c:\OutpExample.PDF) in C#. TIFF is a multipage file format, and so more than one image could be in the InpExample.TIFF file. So the conversion is in a loop. BITiff1 is an instance of BiTiff.ocx, BIPDF1 is an instance of BiPdf.ocx. The ImageCount stores the number of images in the InpExample.TIFF file.

```
string TIFFFile, PDFFile, ErrorMessage;
long ImageCount, i, hDIB;

PDFFile = "c:\\OutpExample.PDF";
TIFFFile = "c:\\InpExample.TIFF";

ImageCount = 0;
ImageCount = BITiff1.GetNumberOfImagesInTiffFile(TIFFFile);
if (ImageCount == 0)
{
    MessageBox.Show("Error loading the TIFF file.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    BIPDF1.ReleasePDF();
    return;
}

if (ImageCount == -1)
{
    MessageBox.Show("The specified file is not a TIFF file!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    BIPDF1.ReleasePDF();
    return;
}

if (BIPDF1.InitPDF() == false)
    return;

if (BIPDF1.CreatePDF(PDFFile) == false)
{
    MessageBox.Show("Error creating the PDF file.", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
    BIPDF1.ReleasePDF();
    return;
}

for (i = 0; i < ImageCount; i++)
{
    hDIB = BITiff1.LoadTiffIntoDIB(TIFFFile, (short)i, false);
    if (hDIB == 0)
    {
        ErrorMessage = "Error loading image " + Convert.ToString(i+1) + " from the TIFF file!";
        MessageBox.Show(ErrorMessage, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        BIPDF1.ReleasePDF();
        return;
    }
    if (BIPDF1.SaveDIBAsColorPDFPage((int) hDIB, "TIFF32.DLL", 512, (short) -1) == false)
    {
        MessageBox.Show("Error saving image into the PDF file.", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        BIPDF1.FinishPDF();
        BIPDF1.ReleasePDF();
        return;
    }
}

if (BIPDF1.FinishPDF() == false)
    MessageBox.Show("Error finishing the PDF file.", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
else
    MessageBox.Show("PDF file created successfully.", "Information", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
BIPDF1.ReleasePDF();
```

