

# Re: Printing Issue

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.windowsforms/2005-09/msg00179.h>

---

- *From:* v-jetan@xxxxxxxxxxxxxxxxxxxxxx ("Jeffrey Tan[MSFT]")
  - *Date:* Wed, 07 Sep 2005 07:40:32 GMT
- 

Hi Alex ,

Thanks for your feedback.

I have created a test Power Users account on my machine, then test your sample code in this account. Yes, I can reproduce out this problem on my side. This exception occurs in PrintPreviewDialog.CreateHandle.

If we use Reflector to view the source code, we will see that:

```
protected override void CreateHandle()
{
if ((this.Document != null) && !this.Document.PrinterSettings.IsValid)
{
throw new
InvalidPrinterException(this.Document.PrinterSettings);
}
base.CreateHandle();
}
```

Then I added the test code:

```
bool f=this.printDocument1.PrinterSettings.IsValid;
```

For the test account, the result is false, which caused out the InvalidPrinterException exception.

Further invest in System.Drawing.Printing.PrinterSettings.IsValid in Reflector, I got the following code:

```
public bool IsValid
{
get
{
return (this.DeviceCapabilities(0x12, IntPtr.Zero, -1) != -1);
}
}
```

```
private int DeviceCapabilities(short capability, IntPtr pointerToBuffer,
int defaultValue)
{
```

## Re: Printing Issue

```
IntSecurity.AllPrinting.Assert();
string text1 = this.PrinterName;
CodeAccessPermission.RevertAssert();
IntSecurity.UnmanagedCode.Assert();
return this.FastDeviceCapabilities(capability, pointerToBuffer,
defaultValue, text1);
}
```

```
private int FastDeviceCapabilities(short capability, IntPtr
pointerToBuffer, int defaultValue, string printerName)
{
int num1 = SafeNativeMethods.DeviceCapabilities(printerName,
this.OutputPort, capability, pointerToBuffer, IntPtr.Zero);
if (num1 == -1)
{
return defaultValue;
}
return num1;
}
```

After setting breakpoint in these methods one-by-one (you should setup the debug symbols correctly, also, I used Win2003 RunAs function to run VS.net under the test account, so that I can do debugging under this test account), I find that DeviceCapabilities Win32 API failed and returns -1. Calling System.Runtime.InteropServices.Marshal.GetLastWin32Error(), I got the error message below: "The printer name is invalid".

It seems that PrinterSettings.PrinterName property returns the incorrect printer name. Then I checked into the PrinterName property, I get these:

```
public string get_PrinterName()
{
IntSecurity.AllPrinting.Demand();
return this.PrinterNameInternal;
}

private string get_PrinterNameInternal()
{
if (this.printerName == null)
{
return PrinterSettings.GetDefaultPrinterName();
}
return this.printerName;
}

private static string GetDefaultPrinterName()
{
IntSecurity.UnmanagedCode.Assert();
SafeNativeMethods.PRINTDLG printdlg1 =
PrinterSettings.CreatePRINTDLG();
printdlg1.Flags = 0x400;
```

## Re: Printing Issue

```
if (!SafeNativeMethods.PrintDlg(printdlg1))
{
return SR.GetString("NoDefaultPrinter");
}
IntPtr ptr1 = printdlg1.hDevNames;
IntPtr ptr2 = SafeNativeMethods.GlobalLock(new HandleRef(printdlg1,
ptr1));
if (ptr2 == IntPtr.Zero)
{
throw new Win32Exception();
}
string text1 = PrinterSettings.ReadOneDEVNAME(ptr2, 1);
SafeNativeMethods.GlobalUnlock(new HandleRef(printdlg1, ptr1));
ptr2 = IntPtr.Zero;
SafeNativeMethods.GlobalFree(new HandleRef(printdlg1,
printdlg1.hDevNames));
SafeNativeMethods.GlobalFree(new HandleRef(printdlg1,
printdlg1.hDevMode));
return text1;
}
```

I finally get that the `SafeNativeMethods.PrintDlg` return false, so no default printer name is returned (returned by `SR.GetString("NoDefaultPrinter");` statement).

`SafeNativeMethods.PrintDlg` is a Win32 API, which .Net p/invoked it. Then I write a MFC application to test this API in Win32 world:

```
void CMFCPrintDlgTestDlg::OnBnClickedButton1()
{
PRINTDLG printdlg1;
HWND hwnd;

// Initialize PRINTDLG
ZeroMemory(&printdlg1, sizeof(printdlg1));

printdlg1.lStructSize = 0x42;
printdlg1.hwndOwner = 0;
printdlg1.hDevMode = 0;
printdlg1.hDevNames = 0;
printdlg1.Flags = 0;
printdlg1.hwndOwner = 0;
printdlg1.hDC = 0;
printdlg1.nFromPage = 1;
printdlg1.nToPage = 1;
printdlg1.nMinPage = 0;
printdlg1.nMaxPage = 0x270f;
printdlg1.nCopies = 1;
printdlg1.hInstance = 0;
printdlg1.lCustData = 0;
printdlg1.lpfPrintHook = NULL;
```

## Re: Printing Issue

```
printdlg1.lpfmSetupHook = NULL;
printdlg1.lpPrintTemplateName = NULL;
printdlg1.lpSetupTemplateName = NULL;
printdlg1.hPrintTemplate = 0;
printdlg1.hSetupTemplate = 0;
printdlg1.Flags = 0x400;
```

```
if (PrintDlg(&printdlg1)==TRUE)
{
  MessageBox("Succeeded!");
  DeleteDC(printdlg1.hDC);
}
else
{
  char buf[50];
  wsprintf(buf, "error: %d", GetLastError());
  MessageBox(buf, buf);
}
}
```

Testing with the test account, I also can reproduce out the problem. Also, the key point is for setting PRINTDLG.Flags to 0x400, which is PD\_RETURNDEFAULT const.

Yes, after enabling "Manage Documents" permission on the printer for test account, the problem will go away. But I am not sure why PrintDlg with PD\_RETURNDEFAULT require "Manage Documents" permission.

Currently, I do not think there is any good solution for this issue, maybe we have to setup "Manage Documents" permission for all the non-admin accounts.(Yes, you can documented this in your application)

Finally, if you want to submit a bug for microsoft, I suggest you contact Microsoft PSS for it.

You can contact Microsoft Product Support by contacting us at 1-(800)936-5800 or by choosing one of the options listed at <http://www.microsoft.com/services/microsoftservices/supp.mspix>

Hope this information helps

Best regards,

Jeffrey Tan

Microsoft Online Partner Support

Get Secure! – [www.microsoft.com/security](http://www.microsoft.com/security)

This posting is provided "as is" with no warranties and confers no rights.

.

---

• *Follow-Ups:*

Re: Printing Issue

Re: Printing Issue

- ◆ **Re: Printing Issue**
  - ◇ From: Alex Clark
- ◆ **Re: Printing Issue**
  - ◇ From: Collin Bennett

• **References:**

- ◆ **Printing Issue**
    - ◇ From: Alex Clark
  - ◆ **Re: Printing Issue**
    - ◇ From: Collin Bennett
  - ◆ **Re: Printing Issue**
    - ◇ From: Alex Clark
- 
- Prev by Date: **RE: unhandled exception when opening window**
  - Next by Date: **Re: Editable tabPage title**
  - Previous by thread: **Re: Printing Issue**
  - Next by thread: **Re: Printing Issue**
  - Index(es):
    - ◆ **Date**
    - ◆ **Thread**