

PrintCapture

WM_COPYDATA Automation

PrintCapture was written in Microsoft Visual C++ 6.0. In an effort to add some automation to the PrintCapture application we added processing for WM_COPYDATA Windows messages. This automation allows you to send a file name to PrintCapture from another Windows application. When the file name is received via the WM_COPYDATA Windows message PrintCapture opens and processed the file as if you used the "File\Open..." menu or received the file via a serial, USB, or GPIB interface.

Two global ASCII strings that hold the file names being set from an application to PrintCapture.

```
CString      m_cszInstName
CString      m_cszImageName
```

The COPYDATA structure that will be sent to PrintCapture via the WM_COPYDATA Windows message.

COPYDATASTRUCT

The **COPYDATASTRUCT** structure contains data to be passed to another application by the WM_COPYDATA message.

```
typedef struct tagCOPYDATASTRUCT
{
    ULONG_PTR    dwData;
    DWORD        cbData;
    PVOID        lpData;
} COPYDATASTRUCT, *PCOPYDATASTRUCT;
```

Members

dwData

Specifies data to be passed to the receiving application. If this member is 1, PrintCapture will automatically overwrite an existing file with the same name as that passed in **lpData**. If this member is 0, PrintCapture will ask for your permission to overwrite an existing file.

cbData

Specifies the size, in bytes, of the data pointed to by the **lpData** member.

lpData

Pointer to data to be passed to the receiving application. This member can be NULL.

Here is an actual piece of code that sends two file names from an application to PrintCapture via WM_COPYDATA Windows messages.

```
/******
```

```
FUNCTION:          CDlg::OnBtnDisplayHarvestData
```

```
DESCRIPTION:
```

```
This function processes the "Display Image" button. It sends a command to PrintCapture to process files into a bitmap images.
```

```
The files sent to PrintCapture must have the extension:
```

```
.ESC      For Epson ESC files.  
.HGL      For HP Graphics Language (HPGL) plotter files.  
.PCL      For HP Printer Control Language (PCL) printer files.  
.PCW      For PrintCapture raw data files.  
.TXT      For Pure Text (TXT) files.
```

```
RETURNS:
```

```
Nothing.
```

```
PARAMETERS:
```

```
None.
```

```
*****
```

```
-----  
Date      Init      Description  
-----  
02/17/11  MDS      Initial construction.  
-----  
*****/
```

```
void CDlg::OnBtnDisplayHarvestData (void)  
{
```

```
//----- find PrintCapture's window so a message can be sent.  the window ID --  
//----- was found by using Spy++ to locate the PrintCapture window. -----
```

```
CWnd *pPrintCaptureWnd =  
    CWnd::FindWindow (_T("FF840EE0-F84A-11D4-BACF-005004779DB6"), NULL);  
if (pPrintCaptureWnd)  
{  
    INT          nLen;  
    LPARAM       lParam;  
    WPARAM       wParam = 0;  
    COPYDATASTRUCT stCopyData;  
    CFileFind     cFind;  
    CString      cszPath;
```

```
TRACE0 ("dlg: OnBtnDisplayHarvestData() PrintCapture found\n");
```

```

//---- if the instruction file exists, get its full path. In this case ----
//---- m_cszInstName is a file name "Inst.txt". -----

if (cFind.FindFile (m_cszInstName))
{
    cFind.FindNextFile ();
    cszPath = cFind.GetFilePath ();
    nLen = cszPath.GetLength() + 1;

//---- load the COPYDATASTRUCT for sending to PrintCapture. -----

    stCopyData.dwData = 1; //automatic overwrite
    stCopyData.cbData = static_cast<DWORD>(nLen);
    stCopyData.lpData = static_cast<void*>(cszPath.GetBuffer (nLen));

//---- send the WM_COPYDATA message to PrintCapture. -----

    CWinApp* pApp = AfxGetApp ();
    if (pApp)
        wParam =
            reinterpret_cast<WPARAM>(pApp->m_pMainWnd->GetSafeHwnd ());
    lParam = reinterpret_cast<LPARAM>(&stCopyData);
    pPrintCaptureWnd->SendMessage (WM_COPYDATA, wParam, lParam);
}

//---- wait about 1 second for PrintCapture to process the image file. -----

::Sleep (1000);

//---- if the image file exists, get its full path. In this case ----
//---- cszImageName is a file name "Image.hgl". -----
-----

if (cFind.FindFile (m_cszImageName))
{
    cFind.FindNextFile ();
    cszPath = cFind.GetFilePath ();
    nLen = cszPath.GetLength() + 1;

//---- load the COPYDATASTRUCT for sending to PrintCapture. -----

    stCopyData.dwData = 1; //automatic overwrite
    stCopyData.cbData = static_cast<DWORD>(nLen);
    stCopyData.lpData = static_cast<void*>(cszPath.GetBuffer (nLen));

//---- send the command to PrintCapture. -----

    CWinApp* pApp = AfxGetApp ();
    if (pApp)
        wParam =
            reinterpret_cast<WPARAM>(pApp->m_pMainWnd->GetSafeHwnd ());
    lParam = reinterpret_cast<LPARAM>(&stCopyData);
    pPrintCaptureWnd->SendMessage (WM_COPYDATA, wParam, lParam);
}
}

```

```
//---- evidently PrintCapture is not active. -----  
  
    else  
        MessageBox ("The program for displaying the data was not foud.",  
                    "Display Data",  
                    MB_ICONEXCLAMATION | MB_OK);  
} //CDlg::OnBtnDisplayHarvestData
```