PrintCapture
WM_COPYDATA Automation

This automation method is aimed at programmers. 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 processes the file as if you used the "File\Open..." menu or received the file via a Serial Port, GPIB Interface, Prologix GPIB-Ethernet Controller, or Network Interface.

Note: The "Auto Close" setting (see Setup> Configuration> Image Processing> Image Processing "Auto Close") is used with WM_COPYDATA automation. You should use the "Auto Close" setting to keep PrintCapture from using up your memory resources.

A member ASCIIZ string that holds the file names being sent from an application to PrintCapture.

```
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.  PrintCapture
   processes thisdata as bit-mapped data. At this time, only two bits are
   used.  They are:

   Bit 0, xxx0: If this bit is 0, PrintCapture will ask for your permission
   to overwrite an existing image file.

   Bit 0, xxx1: If this bit is 1, PrintCapture will automatically overwrite
   an existing image file with the same name as that passed in lpData.

   Bit 1, xx0x: If this bit is 0, the file name in lpData is an ASCII string.

   Bit 1, xx1x: If this bit is 1, the file name in lpData is a UNICODE
   string.

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.  For
   PrintCapture this is a pointer to a file name string.  The file name
   string can contain a wildcard specifier (e.g. "Image*.hgl"). If this
   pointer is NULL, points to an empty file name string, or the file name
   does not have a valid raw data file extension then PrintCapture will
   notify you with a message box.
```

Here is an actual piece of code that sends a file name from an application to PrintCapture via WM_COPYDATA Windows message.

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

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
--------   ----   ------------------------------------------------------------
05/01/12    MDS   Initial construction.
*****************************************************************************/

void CDlg::OnBtnDisplayHarvestData (void)
{

//---- find PrintCapture's window so a message can be sent.  the window ID --
//---- is a globally unique ID (GUID) set for PrintCapture's window. --------

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

      TRACE0 ("dlg: OnBtnDisplayHarvestData() PrintCapture found\n");
//---- if the image file exists, get its full path.  In this case ----------
//---- m_cszImageName is a file named "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));

//---- build parameters fo WM_COPYDATA Windows message. ---------------------

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

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

         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
```