A process to automatically open and display documents from Excel is described.

Author:

Date: 17/04/12

Bombardier Transportation UK Ltd

Estover Close

Estover

Plymouth PL6 7PU

Tel : +44 1752 725000

Fax: +44 1752 725001

This document and its contents are the property of Bombardier Inc. or its subsidiaries. This document contains confidential proprietary information. The reproduction, distribution, utilisation or the communication of this document or any part thereof, without express authorisation, is strictly prohibited. Offenders will be held liable for the payment of damages.

**Table of Contents**

[1 Introduction 3](#_Toc322679278)

[2 Revision History 3](#_Toc322679279)

[3 Operation 4](#_Toc322679280)

[4 Create a Shortcut Icon 4](#_Toc322679281)

[5 Limitations 4](#_Toc322679282)

[6 SAP Export 4](#_Toc322679283)

[7 Appendix 1 5](#_Toc322679284)

[8 Appendix 2 6](#_Toc322679285)

# Introduction

Engineers and others often require rapid access to released drawings that are stored in the current issue network folder.

A macro has been added to the drawing state Excel file.

Note that the drawing and design note spreadsheets may be merged into DesignNoteDrgstateSAP.xls

These files are stored in g:\uk\pl\dos\DrgState.

To store a macro in an Excel 2007 spreadsheet requires the use of the xlsb format.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Issue | |  |  |  |
| From | To | Date | Reason for Change | Sig. |
| - | 1 | 17/04/12 | First Issue | KS |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Operation

1. Make sure that macros are enabled in Excel. Check in Excel-options-macros.
2. Open the drawing list Excel file (DesignNoteDrgstateSAP.xls)
3. Use the Excel search or filter facilities to locate the drawings of interest.
4. Click the cursor on the drawing number in column A.
5. Either use the keyboard shortcut (CTRL+l) or shortcut icon on the Quick Access Toolbar (see section 4).
6. The macro will then locate the file(s) in current issue. If none exists a user message will report this. If only one file exists it will be opened in its native application.
7. If more than one file exists then a menu will be displayed showing the list of files available.
8. Select the number of the required file and the file will be opened.

# Create a Shortcut Icon

1. Right click on the menu bar (the bar with Home : Insert : Page Layout : etc) and select Customise Quick Access Toolbar.
2. In the ‘Choose commands from’ box select Macros and in the box below locate and select PERSONAL.XLS!DrgState or DrgState then click Add.
3. Select the new Macro and click Modify. Select a suitable icon (the default will suffice) and change the name in the box below to something meaningful (DrgState is suggested).

# Limitations

1. If files have not been stored on 1\_current\_iss then they will not be found.
2. SAP will always be slightly more up to date than the Excel file list which is only updated periodically.

# SAP Export

1. Export the drawings and design notes from SAP.
2. Combine both sets of data into the Excel workbook that contains the DrgState macro (DrgStateMacroTemplate.xls).
3. Create the index file either by allowing the macro to determine that the index is out of date or by double clicking on the CreateIndex.bat file stored in the DrgState folder.

# Appendix 1

'Option Explicit

Private Declare Function OpenProcess Lib "kernel32" (ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal dwProcessId As Long) As Long

Private Declare Function GetExitCodeProcess Lib "kernel32" (ByVal lnghProcess As Long, lpExitCode As Long) As Long

Private Const PROCESS\_ALL\_ACCESS = &H1F0FFF

Public Function ShlProc\_IsRunning(ShellReturnValue As Long) As Boolean

Dim lnghProcess As Long

Dim lExitCode As Long

Dim lRet As Long

'//Get the process handle

lnghProcess = OpenProcess(PROCESS\_ALL\_ACCESS, 0&, ShellReturnValue)

If lnghProcess <> 0 Then

'// The GetExitCodeProcess Function retrieves the

'// termination status of the specified process.

GetExitCodeProcess lnghProcess, lExitCode

If lExitCode <> 0 Then

'// Process still Running

ShlProc\_IsRunning = True

Else

'// Process completed

ShlProc\_IsRunning = False

End If

End If

End Function

Sub DrgOpen()

'

' Use with DrgstateSAP.xlsx to search for and open files directly.

' Periodically when DrgstateSAP.xlsx is updated run the following command to create the index:

' dir G:\UK\PL\DOS2\1\_current\_iss /s/b > g:\uk\pl\dos\DrgState\index.txt

' Use DrgstateSAP.xlsx to find the drawing number wanted and click in the drawing number cell.

' Run the find macro using CTRL+L and if only one file exists it will be opened.

' If there are multiple files then a menu will be presented listing the files available.

' Choose the menu option number and that file will be opened.

'

Dim Cmd As String

Dim file As String

Dim reply As Variant

Dim DataArray(1 To 10) As String

Dim strBuf As String

Dim intIndex As Integer

Dim IndexFile As String

Dim RepositoryFolder As String

Dim ResultFile As String

Dim BatchFile As String

Dim TaskId As Long

Dim RepoDate, IndexDate As Variant

RepositoryFolder = "g:\uk\pl\dos2\1\_current\_iss"

IndexFile = "g:\uk\pl\dos\drgstate\index.txt"

ResultFile = "c:\temp\result.txt"

BatchFile = "g:\uk\pl\dos\drgstate\CreateIndex.bat"

' Locate file in index and return full path to file (s)

file = ActiveCell.Value

' Find and replace '/' with '-' for file name.

file = Replace(file, "/", "-")

' Check if index file is out of date.

RepoDate = FileDateTime(RepositoryFolder)

IndexDate = FileDateTime(IndexFile)

' Change the < to > for normal operation (< is for test)

If RepoDate > IndexDate Then

Choice = MsgBox("Index is out of date." & vbLf & "Do you want to update it now?." & vbLf & "This could take a couple of minutes.", vbOKCancel)

'MsgBox ("Index is out of date." & vbLf & "Please wait whilst index is updated." & vbLf & "This could take a couple of minutes.")

If Choice = vbOK Then

Cmd = Environ$("comspec") & " /c " & BatchFile

TaskId = Shell(Cmd, vbNormalFocus)

' Wait for process to finish

Do While ShlProc\_IsRunning(TaskId) = True

DoEvents

Loop

End If

End If

Cmd = Environ$("comspec") & " /c find """ & file & """ " & IndexFile & " > " & ResultFile

TaskId = Shell(Cmd, vbNormalFocus)

' Wait for process to finish

'Do While ShlProc\_IsRunning(TaskId) = True

' DoEvents

'Loop

' Delay while file is written

newHour = Hour(Now())

newMinute = Minute(Now())

newSecond = Second(Now()) + 2 ' 2 second delay

waitTime = TimeSerial(newHour, newMinute, newSecond)

Application.Wait waitTime

' Check that user has a c:\temp folder, if not create one.

If Len(Dir("c:\temp\")) = 0 Then MkDir ("c:\temp")

' Read in paths to found files

Open ResultFile For Input As #1

Line = 0

' while not eof or max array size

Do Until EOF(1) Or Line = 9

Line = Line + 1

Input #1, DataArray(Line)

Loop

Close #1

' More than 2 lines indicates that at least 1 file has been found

If Line > 2 Then

If Line > 3 Then

strBuf = "0 = None" & vbLf

For intIndex = 3 To Line

strBuf = strBuf & intIndex - 2 & " = " & Right(DataArray(intIndex), 40) & vbLf

Next

Choice = -9

Do Until (Choice >= 0) And (Choice < Line - 1)

Choice = Int(InputBox(strBuf, "Choose file:", 1))

Loop

filepath = DataArray(Choice + 2)

Else

filepath = DataArray(Line)

Choice = 1

End If

If Choice > 0 Then

' Create link to file

link = "file:///" & filepath

' Open file in applicaion

ActiveWorkbook.FollowHyperlink link

End If

Else

' no paths returned from search

MsgBox ("File not found")

End If

End Sub

Sub TextFilter()

'

' TextFilter Macro

' Set up data filters on Document and Description columns.

' Enter up to 2 words in each search box, these will be OR'd

Dim DrgNum, DrgDesc As String

Dim DrgNumWords() As String

Dim DrgDescWords() As String

Dim DrgNumWordCount, DrgDescWordCount As Integer

Dim test() As String

DrgNumWordCount = 4

DrgDescWordCount = 4

Do Until (DrgNumWordCount < 3)

' Get search data from user & convert to upper case

DrgNum = StrConv(InputBox("Enter part of the Drawing Number" & vbLf & "Up to 2 words can be entered" & vbLf \_

& "Use & | for AND OR", "Drawing Search"), vbUpperCase)

' Split the input strings into an array of words.

If InStr(1, DrgNum, "&") > 0 Then

DrgNumWords = Split(DrgNum, "&")

ElseIf InStr(1, DrgNum, "|") > 0 Then

DrgNumWords = Split(DrgNum, "|")

Else: DrgNumWords = Split(DrgNum)

End If

If DrgNum = "" Then DrgNum = "\*"

' Return the number of elements in the arrays.

DrgNumWordCount = UBound(DrgNumWords) - LBound(DrgNumWords) + 1

Loop

'ReDim Preserve DrgNumWords(2) As String

'For i = UBound(DrgNumWords) To 1

' DrgNumWords(i) = ""

'Next

Do Until (DrgDescWordCount < 3)

' Get search data from user & convert to upper case

DrgDesc = StrConv(InputBox("Enter part of the Drawing Description" & vbLf & "Up to 2 words can be entered" & vbLf \_

& "Use & | for AND OR", "Drawing Search"), vbUpperCase)

If InStr(1, DrgDesc, "&") > 0 Then

DrgDescWords = Split(DrgDesc, "&")

ElseIf InStr(1, DrgDesc, "|") > 0 Then

DrgDescWords = Split(DrgDesc, "|")

Else: DrgDescWords = Split(DrgDesc)

End If

If DrgDesc = "" Then DrgDesc = "\*"

DrgDescWordCount = UBound(DrgDescWords) - LBound(DrgDescWords) + 1

Loop

'ReDim Preserve DrgDescWords(2) As String

'For i = UBound(DrgDescWords) To 1

' DrgDescWords(i) = ""

'Next

Range("A7").Select

If DrgNum = "\*" Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=1, \_

Criteria1:="=\*", Operator:=xlAnd

ElseIf InStr(1, DrgNum, "&") = 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=1, \_

Criteria1:="=\*" & DrgNumWords(0) & "\*", Operator:=xlAnd

ElseIf InStr(1, DrgNum, "&") > 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=1, \_

Criteria1:="=\*" & DrgNumWords(0) & "\*", Operator:=xlAnd, Criteria2:="=\*" & DrgNumWords(1) & "\*"

ElseIf InStr(1, DrgNum, "|") > 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=1, \_

Criteria1:="=\*" & DrgNumWords(0) & "\*", Operator:=xlOr, Criteria2:="=\*" & DrgNumWords(1) & "\*"

End If

If DrgDesc = "\*" Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=2, \_

Criteria1:="=\*", Operator:=xlAnd

ElseIf InStr(1, DrgDesc, "&") = 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=2, \_

Criteria1:="=\*" & DrgDescWords(0) & "\*", Operator:=xlAnd

ElseIf InStr(1, DrgDesc, "&") > 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=2, \_

Criteria1:="=\*" & DrgDescWords(0) & "\*", Operator:=xlAnd, Criteria2:="=\*" & DrgDescWords(1) & "\*"

ElseIf InStr(1, DrgDesc, "|") > 0 Then

ActiveSheet.Range(Selection, Selection.SpecialCells(xlLastCell)).AutoFilter Field:=2, \_

Criteria1:="=\*" & DrgDescWords(0) & "\*", Operator:=xlOr, Criteria2:="=\*" & DrgDescWords(1) & "\*"

End If

ActiveWindow.SmallScroll Down:=-1000

End Sub

Sub CreateIndex()

' Generate new index file

Dim TaskId As Long

BatchFile = "g:\uk\pl\dos\drgstate\CreateIndex.bat"

Cmd = Environ$("comspec") & " /c " & BatchFile

TaskId = Shell(Cmd, vbNormalFocus)

' Wait for process to finish

Do While ShlProc\_IsRunning(TaskId) = True

DoEvents

Loop

End Sub

# Appendix 2

CreateIndex.bat

dir G:\UK\PL\DOS2\1\_current\_iss /s/b > g:\uk\pl\dos\DrgState\index.txt