

```

Option Explicit
' Add a reference to PDFCreator.
' - in VBA IDE "Tools", "References", add tick to "PDFCreator".
' =====
' Sleep function necessary to stop PDFCreator from locking up.
Private Declare Sub Sleep Lib "kernel32.dll" (ByVal dwMilliseconds As Long)
' Save formats used by PDFCreator.
Private Enum eAutoSaveFormat
    asfPdf = 0
    asfPng = 1
    asfJpg = 2
    asfBmp = 3
    asfPcx = 4
    asfTif = 5
    asfPS = 6
    asfEPS = 7
    asfTxt = 8
End Enum
' Print the worksheet to PDF.
Sub PrintWksAsPDF()
    PrintToPdf True
End Sub
' Print all sheets in the active workbook to PDF.
Sub PrintWbkAsPDF()
    Dim sPrompt As String
    sPrompt = "Print all worksheets - Are you sure?"
    If MsgBox(sPrompt) Then PrintToPdf False
End Sub
' Print the active workbook to a PDF in the same folder,
' and named "<wbk> - <sht>.pdf".
Private Sub PrintToPdf(bSheetOnly As Boolean)
    Dim sOutFile As String
    Dim sDir As String
    ' Is there a worksheet open?
    If (ActiveSheet Is Nothing) Then
        MsgBoxCriticalError "No active spreadsheet to print!"
    Else
        If (ActiveWorkbook.Path = "") Then
            ' Never been saved; use temp folder, and generic file name.
            sDir = Environ("TEMP")
            sOutFile = "Workbook"
        Else
            ' Saved, so use its folder and file name.
            sDir = ActiveWorkbook.Path
            sOutFile = Replace(ActiveWorkbook.Name, ".xls", "")
        End If
        ' Tack on the sheet name and PDF extension.
        If bSheetOnly Then
            ' Are we printing multiple sheets?
            If (ActiveWindow.SelectedSheets.Count > 1) Then
                sOutFile = sOutFile & " - Selected Sheets.pdf"
            Else
                sOutFile = sOutFile & " - " & ActiveSheet.Name & ".pdf"
            End If
        Else
            sOutFile = sOutFile & " - All Sheets.pdf"
        End If
    End If

```

```

    ' Create the PDF.
    PrintWbkWksAsPDF sDir, sOutFile, bSheetOnly
End If
End Sub
' Print the current worksheet to a specified folder & file.
Sub PrintWbkWksAsPDF(sDir As String, sOutFile As String, bSheetOnly As Boolean)
    Dim pdfc As PDFCreator.clsPDFCreator
    Dim asf As eAutoSaveFormat
    Dim sSavePrinter As String
    Dim sFullOutFile As String
    Dim sPrompt As String
    '
    sFullOutFile = sDir & Application.PathSeparator & sOutFile
    If (Dir(sFullOutFile) <> "") Then
        sPrompt = "Output file '" & sFullOutFile
        sPrompt = sPrompt & "' already exists...overwrite it?"
        If MsgNo(sPrompt) Then
            Kill sFullOutFile
        Else
            Exit Sub
        End If
    End If
    asf = asfPdf          ' Save as PDF format
    ' Check that there is an active worksheet to print.
    If (ActiveSheet Is Nothing) Then
        MsgBoxCriticalError "No active spreadsheet to print!"
        Exit Sub
    End If
    ' Set a PDFCreator printer as default printer.
    sSavePrinter = Application.ActivePrinter
    If SelectPDFCreatorPrinter Then
        ' Got the printer, continue with printing.
        Set pdfc = New clsPDFCreator
        With pdfc
            ' Start the printer.
            If .cStart("/NoProcessingAtStartup", False) = False Then
                If .cStart("/NoProcessingAtStartup", True) = False Then
                    MsgBoxCriticalError "Can't initialize PDFCreator."
                    Exit Sub
                End If
            End If
        End With
        If (.cError.Number > 0) Then ShowPDFCreatorError .cError
        ' Set the print options.
        .cOption("OptionsEnabled") = 1
        .cOption("UseAutosave") = 1
        .cOption("UseAutosaveDirectory") = 1
        .cOption("UseStandardAuthor") = 1
        .cOption("AutosaveDirectory") = sDir
        .cOption("AutosaveFilename") = sOutFile
        .cOption("AutosaveFormat") = asf
        .cOption("PDFFontsEmbedAll") = 1
        .cOption("StandardAuthor") = Environ("USERNAME")
        .cClearCache
        If (.cError.Number > 0) Then ShowPDFCreatorError .cError
        DoEvents
        ' Select
        If bSheetOnly Then

```

```

        ' Print the selected worksheet(s).
        ActiveWindow.SelectedSheets.PrintOut
    Else
        ' Print all worksheets.
        ActiveWorkbook.PrintOut
    End If
    ' Wait until the printing has finished.
    Sleep 2000
    If (.cError.Number > 0) Then ShowPDFCreatorError .cError
    .cPrinterStop = False
    Do Until .cCountOfPrintjobs > 0
        DoEvents
    Loop
    .cPrinterStop = False
    Do Until .cCountOfPrintjobs = 0
        DoEvents
    Loop
    .cPrinterStop = True
    ' Report error in VBA.
    If (.cError.Number > 0) Then ShowPDFCreatorError .cError
    ' Pause before closing to avoid lockup.
    Sleep 8000
    .cClose
End With
' Restore previous printer.
Application.ActivePrinter = sSavePrinter
' Inform user of success, and location of PDF file.
If (Dir(sFullOutFile) <> "") Then
    sPrompt = "File '" & sFullOutFile & "' saved."
    sPrompt = sPrompt & vbCrLf & vbCrLf
    sPrompt = sPrompt & vbTab & vbTab & "View it?"
    If MsgYes(sPrompt) Then
        If (Not ShellOpenFile(sFullOutFile, sDir)) Then
            MsgBox "Could not display '" & sFullOutFile & "'."
        End If
    End If
End If
Else
    ' Couldn't get a PDFCreator printer.
    MsgBoxCriticalError "Can't select a PDFCreator printer"
    Exit Sub
End If
Exit_Func:
    ' Housekeeping.
    Set pdfc = Nothing
End Sub
' Display PDFCreator print error.
Private Sub ShowPDFCreatorError(pdfcErr As PDFCreator.clsPDFCreatorError)
    MsgBoxCriticalError Format(pdfcErr.Number) & ": " & pdfcErr.Description
End Sub
' Select a PDFCreator printer.
Private Function SelectPDFCreatorPrinter() As Boolean
    Dim pdfc As PDFCreator.clsPDFCreator
    Dim clnPDFCPrinters As Collection
    Dim sPrinter As String
    Dim sNetworkPrinter As String
    Dim sSavePrinter As String

```

```
'  
SelectPDFCreatorPrinter = False  
sSavePrinter = Application.ActivePrinter  
On Error GoTo Exit_Func  
Set pdfc = New clsPDFCreator  
With pdfc  
    Set clnPDFCPrinters = .cGetPDFCreatorPrinters()  
    If (clnPDFCPrinters Is Nothing) Then GoTo Exit_Func  
    If (clnPDFCPrinters.Count < 1) Then GoTo Exit_Func  
    sPrinter = clnPDFCPrinters(clnPDFCPrinters.Count)  
    .cDefaultPrinter = sPrinter  
    SelectPDFCreatorPrinter = True  
End With  
GoTo Exit_Func  
Exit_Func:  
On Error GoTo 0  
If (Not SelectPDFCreatorPrinter) Then  
    Application.ActivePrinter = sSavePrinter  
End If  
Set pdfc = Nothing  
End Function
```