I'm not robot
reCAPTCHA
Continue

Excel vba create pdf and attach to email

May 26, 2015 May 26, 2015 / Chris Newman I absolutely love to use email as attachment functionality provided by Excel. It's great that you quickly send one-time data requests to someone during the day. The only thing I dislike about this built-in feature is the lack of customization that I find myself repeatedly making the same changes over and over again. This irritation inspired me to create a couple of VBA macros so I could add a little more functionality to Microsoft's email as attachment idea. First, I hate not being able to mention the attached file. I can't tell you how many times I've attached a file called Book1.xlsx to anyone. I also like to add some kind of message to the email, such as Please see attached or Attached is your requested data. Having to print this out every time can be a little annoying. So I would like to share with you what I use to send my Excel workbooks and worksheets as attachments. Now this code was meant to suit my needs, but hopefully I've added enough comments so you can easily adjust it to suit your specific needs. Example of what this VBA code will send this VBA macro code with adding the entire ActiveWorkbook as an attachment to a brand new Outlook message. Sub EmailWorkbook()'PURPOSE: Create email message with ActiveWorkbook attached'SOURCE: www.TheSpreadsheetGuru.comDim SourceWB As WorkbookDim DestinWB as WorkbookDim OutlookApp As ObjectDim OutlookMessage as ObjectDim TempFileName As VariantDim ExternalLinks As VariantDim TempFilePath As StringDim FileExtStr As StringDim DefaultName As StringDim UserAnswer As LongDim x As LongSet SourceWB = ActiveWorkbook'Look for macro code resident in If Val (Application.Version) > = 12 So if SourceWB.FileFormat = 51 And SourceWB.HasVBProject = True Da UserAnswer = MsgBox (It was VBA code found in this xlsx file. & If you continue the VBA code will not be included in the email attachment. & _ Do you want to continue?, vbYesNei, VBA code found!) If UserAnswer = vbNo then Exit Sub 'Handle if user cancels end if'Determine temporary file path TempFilePath = Environ$(temp) & \'Determine the default input box file name if SourceWB.Saved Then DefaultName = Left(SourceWB.Name, InStrRev(SourceWB.Name, .) - 1) Else DefaultName = SourceWB.Name End If'Be user for a filename TempFileName = Application.InputBox(What do you want to name the attachment? (No special characters!), _ Filename, Type:=2, Default:=DefaultName) If TempFileName = False then Exit Sub 'Handle if user cancels 'Determine file name if SourceWB.Saved = True Then FileExtStr = . & LCase(Right(SourceWB.Name, Len(SourceWB.Name) - InStrRev(SourceWB.Name, ., , 1))) Else FileExtStr = .xlsx End If'Optimize Code Application.ScreenUpdating = False Application.EnableEvents = False Application.DisplayAlerts = False'Save Workbook Temporary SourceWB.SaveCopyAs TempFilePath & TempFileName & Set DestinWB = Workbooks.Open(TempFilePath & TempFileName & & External Links ExternalLinks = DestinWB.LinkSources (Type:=xlLinkTypeExcelLinks) 'Loop Through each external link in ActiveWorkbook and Break it by error Resume Next For x = 1 To UBound (ExternalLinks) DestinWB.BreakLink Name:= ExternalLinks (x), Type:=xlLinkTypeExcelLinks Next x On Error GoTo 0 'Save changes DestinWB.Save'Create instance of Outlook by error Resume Next set OutlookApp = GetObject (class:=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing then specify OutlookApp =CreateObject (class:=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing then specify OutlookApp=CreateObject (class:=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing then specify OutlookApp=CreateObject (class:=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing then specify OutlookApp=CreateObject (class:=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing :=Outlook.Application) = . Subject = TempFileName . Body = Please see attached. & vbNewLine & vbNewLine & Chris . Attachments.Add DestinWB.FullName . Show end with error GoTo 0'Close and delete the temporary file DestinWB.Close SaveChanges:=False Kill TempFilePath & TempFileName & FileExtStr'Clear Memory Set OutlookMessage = Nothing set OutlookApp = Nothing 'Optimize CodeExitSub: Application.ScreenUpdating=True Application.EnableEvents=True Application.DisplayAlerts=TrueEnd SubThis VBA macro will only attach the selected tabs in ActiveWorkbook to a new Outlook email message. Sub EmailSelectedSheets()'PURPOSE: Create email message with only selected worksheet attached'SOURCE: www.TheSpreadsheetGuru.comDim SourceWB WorkbookDim DestinWB as WorkbookDim OutlookApp As ObjectDim OutlookMessage As ObjectDim TempFileName As VariantDim ExternalLinks As VariantDim TempFilePath As StringDim FileExtStr As StringDim DefaultName As StringDim UserAnswer So LongDim x As Long'Optimize Code Application.ScreenUpdating = False Application.EnableEvents = False Application.DisplayAlerts = False'Copy only selected sheets to new workbook Set SourceWB = ActiveWorkbook SourceWB.Windows (1). SelectedSheets.Copy Set DestinWB = ActiveWorkbook'Look for macro code resident in If Val(Application.Version) >= 12 Then if SourceWB.FileFormat = 51 and SourceWB.HasVBProject = True Then UserAnswer = MsgBox(There was VBA code found in this xlsx file. & _ If you continue vba code will not be included in your email attachment. & Do you want to continue?, vbYesNei, VBA Code Found!) 'Handle if user cancels If UserAnswer = vbNo then DestinWB.Close SaveChanges:=False GoTo ExitSub End If End If'Determine Temporary File Path TempFilePath = Environ$(temp) & \'Determine the default file name for Input Box if SourceWB.Saved Then DefaultName = Left (SourceWB.Name, InStrRev(SourceWB.Name, .) - 1) Else DefaultName = SourceWB.Name End If'Be user for a filename TempFileName = Application.InputBox(What do you want to name the attachment? (None _ File name, Type:=2, Type:=2, If TempFileName = False then goTo ExitSub 'Act if the user cancels 'Determine the extension if SourceWB.Saved = True Then FileExtStr= . & LCase(Right(SourceWB.Name, Len(SourceWB.Name) - InStrRev(SourceWB.Name, ., , 1))) Else FileExtStr = .xlsx End If'Break external links ExternalLinks = DestinWB.LinkSources (Type:=xlLinkTypeExcelLinks) 'Loop Through each external link in ActiveWorkbook and break it on error resume next for x=1 to UBound (ExternalLinks) DestinW B.BreakLink Name:= ExternalLinks(x) , Type:=xlLinkTypeExcelLinks Next x In case of error GoTo 0 'Save temporary workbook DestinWB.SaveCopyAs TempFilePath & TempFileName & FileExtStr'Create instance of Outlook by error Resume next set OutlookApp = GetObject (class :=Outlook.Application) 'Handles if Outlook is already open Err.Clear If OutlookApp is nothing then specify OutlookApp=CreateObject (class:=Outlook.Application) 'If not, open Outlook If Err.Number = 429 Then MsgBox Outlook could not be found, canceled., 16, Outlook not found GoTo ExitSub End If on error GoTo 0'Create a new email message Set OutlookMessage =OutlookApp.CreateItem(0)'Create Outlook emails with attachments On error resume next with OutlookMessage . To = . CC = . BCC = . Subject = TempFileName . Body = Please see attached. & vbNewLine & vbNewLine & Chris . Attachments.Add TempFilePath & TempFileName & FileExtStr . Show end with error Go To 0'Close and delete the temporary file DestinWB.Close SaveChanges:=False Kill TempFilePath & TempFileName & FileExtStr'Clear Memory Set OutlookMessage = Nothing Set OutlookApp = Nothing 'Optimize CodeExitSub: Application.ScreenUpdating=True Application.EnableEvents=True Application.DisplayAlerts=TrueEnd SubIf you found this VBA code useful, you are sure to love my highly reputable exporter template. This Excel utility allows you to automate how to save your files and who you send them as well. Click on the banner below to learn more about this incredibly powerful tool! Chances are this post doesn't give you the exact answer you were looking for. We all have different situations, and it is impossible to take into account all the special needs one may have. That's why I want to share with you: My guide to getting the solution to your problems QUICKLY! In this article, I explain the best strategies I've come up with over the years to get quick answers to complex problems in Excel, PowerPoint, VBA, you name it! I strongly recommend that you check this guide out before asking me or anyone else in the comments box to resolve your specific problem. I can guarantee 9 times out of 10, one of my strategies will get you the answer(s) you need faster than it will take me to get back to you with a possible solution. I try my best to help everyone, but sometimes I don't have time to fit everyone's questions in (there never seems to be quite enough hours today!). wish you good luck and I hope this tutorial gets you heading in the right direction! Chris Makro Newman :) May May 2015/ Chris Newman/ Newman/

block unknown numbers on android phone , motion_map_physics_worksheet.pdf , murder_mystery_codes_2020_new.pdf , normal_5f8903983e57d.pdf , ffxiv diadem gathering map , german grammar books for beginners pdf , hydralazine davis pdf , diagrama de colaboracion uml , fundamentals of sailplane design pdf , manual_pull_station_notifier.pdf , old_school_rs_fletching_guide.pdf , dedobijin.pdf ,