CorelDRAW and Corel DFSIGNFR This tutorial demonstrates how to create macros for Corel-DRAW Graphics Suite (CDGS) X7 and CorelDRAW Technical Suite (CDTS) X7 using Visual Basic for Applications (VBA) development environment. This can be done in any of the CDGS/CDTS applications such as CorelDRAW, Corel DE-SIGNER or Corel PHOTO-PAINT. This article walks the user through the steps of creating a simple Image Resizer macro which takes images from a specified folder, opens them in Corel PHOTO-PAINT, resizes them to specific dimensions and saves them to a different folder. Even though the example described below is designed for Corel PHOTO-PAINT X7, the same principles can be easily applied to CorelDRAW and Corel DESIGNER. The macro project will contain a simple macro that shows a dialog box which allows the user to enter the location of images to process, required dimensions and the output folder. Creating New Macro Project First create a new macro project for Image Resizer macro. For this, open the Macro Manager docker (Alt-Shitf-F11), select "Visual Basic for Applications" in the list, right-click and select "New Macro Project..." Macro Manager Load...

CorelDR

Developer Area

Docs & Tutorials

More

Creating VBA Macros with User Interface in

■ ✓ Visual Studio Tools for Applications - Programme Visual Basic for Applications Load Macro Project... 🖮 🔜 GlobalMac New Macro Project... New Macro Tool... Show IDF... When the "Save As" dialog box appears, enter "ImageResizer" in the file name. This will create a new macro project

called ImageResizer.gms and it will be loaded into the Macro

×

Macro Manag

Manager docker now:

New ▼ Load... 🖆

■ ✓ Visual Studio Tools for Applications

🖮 🥕 Visual Basic for Applications

i ImageResizer

Macro Manager

Right-click "ImageResizer" and select "New Module" from the pop-up menu and rename the created module to "Resizer". Finally, right-click the "Resizer" mode and select "New Macro" which will open the VBA editor with the new "Macro1" open: Microsoft Visual Basic for Applications - ImageResizer - [Resizer (Code)] 🦂 <u>F</u>ile <u>E</u>dit <u>V</u>iew <u>I</u>nsert F<u>o</u>rmat <u>D</u>ebug <u>R</u>un <u>T</u>ools <u>A</u>dd-Ins <u>W</u>indow <u>H</u>elp i 🗿 🔯 - 🔚 | % 🗈 🖺 AA | 🔊 🖭 🕨 🔟 📵 🔛 🥞 🚰 🖖 🦟 🛛 Project - ImageResizer X (General) Macro1 **=** = | Sub Macro1() End Sub 🖽 🝇 GlobalMacros (GlobalMa 🗓 🝇 HTMLSlideshow (Slidesh ⊟ 🐉 ImageResizer Ė... € Modules Resizer Properties - Resizer ΧI Resizer Module Alphabetic Categorized ame) Resizer

Rename the macro and show a simple message by editing the macro code to be: Sub Start() MsgBox "Hello, world!" End Sub Now the basic macro project is set up. You can try running this macro by double-clicking the "Start" macro under ImageResizer > Resizer in Macro Manager docker: Corel PHOTO-PAINT X7 (64-Bit) <u>File Tools Window Help</u> F - 4 - 13 14 一日日 ※ 国 町 り か 41% · 100 % ○ 0.0 ° 💠

18

女.

30

2

٨,

Π,

X

Macro Manager

Load...

HTMLSlideshow

ImageResizer ThisMacroStorage

🛓 🞒 Resizer

Start

■ Visual Studio Tools for Applications 🏂 Visual Basic for Applications GlobalMacros

1 <u> ii</u> Þ **Creating Macro Dialog Box** Now that the macro project is set up, it's time to add a user interface to it. Switch back to VBA editor, select the "Resizer" module in VBA's Project Explorer docker (Ctrl-R), rightclick it and select Insert > UserForm from the pop-up menu: 췸 Microsoft Visual Basic for Applications - ImageResizer - [Resizer (Code)] File Edit View Insert Format Debug Run Tools Add-Ins Window <u>o</u> <u>e</u> - <u>|</u> | % • <u>|</u> 6 A | 9 € | • | |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 | Project - ImageResizer X (General) Sub Start() MsgBox "Hello, world!" 🖽 🝇 GlobalMacros (GlobalMa End Sub 🖮 🝇 HTMLSlideshow (Slidesh 🖮 🝇 ImageResizer - Modules View Code **1** View Object ImageResizer Properties...

UserForm

Module

A new user form will be added to the macro project and the

UserForm1

🚰 Microsoft Visual Basic for Applications - ImageResizer - [UserForm1 (Us... <mark>₩ File Edit View Insert Format Debug Run T</mark>ools Add-Ins <u>W</u>indow <u>H</u>elp

Project - ImageResizer X

Class Module

Insert

Print... Dockable <u>H</u>ide

form designer window will open:

Import File...

Export File...

Remove Resizer...

Properties - Res

Alphabetic Catego

Name) Resizer

= =

- Forms

Corel PHOTO-PAINT X7 (64-Bit) <u>F</u>ile <u>T</u>ools <u>W</u>indow <u>H</u>elp

Q,

ijij,

۵,

1

>> Þ

↔ ○ □ Ø Ø X: 0 px

Toolbox

N A abl 国

ᄪᅜᅊᆛ

♦| 🙈

Project - ImageResizer X

🖽 🝇 GlobalMacros (GlobalMa

🞚 🐰 HTMLSlideshow (Slidesh

UserForm1

Properties - cmdResize X

cmdResize

1 - fmBackStyle

Resize Images V

<

False ■ 8H8000000

False

Caption/Text

Destination Fold-

Source Folder:

Browse...

Browse...

Width:

Height:

Maintain propor-

Resize Images

Resizer

Properties - cmdResize cmdResize CommandButton Alphabetic Categorized

Accelerator AutoSize

BackColor

BackStyle 'ance Caption

End Sub

button is clicked:

Source Folder

Destination Fo

ageResizer project:

⊕ · · · male Forms 🖃 ··· 🥞 Modules

—── Forms

i ... @ Modules

Project - HTMLSlideshov X

🖽 🝇 GlobalMacros (GlobalMa M HTMLSlideshow (Slidesh

> ----- FolderPicker SlideShow

..... UserForm1

Dim folder As String

nderDFrame", Me.Caption))

Dim folder As String

nderDFrame", Me.Caption))

If folder = "" Then Exit Sub

txtSourceFolder.Text = folder

If folder = "" Then Exit Sub

txtDestFolder.Text = folder

Private Sub cmdBrowseSource Click()

tion Folder",

End Sub

Text,

er", _

End Sub

Resizer

🖃 🝇 ImageResizer

cmdResize

1 - fmBackStvl

Resize Images

dler to be the following:

Private Sub cmdResize Click()

MsgBox "Button clicked"

∃ <

Change the implementation of cmdResize_Click() event han-

This adds a simple code to show a message box when the

Image Resizer

OK

Corel PHOTO-PAINT X7

Resize Images

Now implement the Browse buttons for source and destina-

tion folders. This requires a "Browse For Folder" dialog from

Windows which is not available in VBA directly. However VBA

can use low-level Windows API functions, and there is al-

ready code available for this which is part of Corel PHOTO-

directly from that macro into your own. Expand the HTML-

the "FolderPicker" module. Select and drag it into the Im-

This will copy the FolderPicker module to ImageResizer and

Add the two button click event handlers for both Browse...

folder = BrowseForFolderDlg(txtDestFolder.Text,

folder = BrowseForFolderDlg(txtSourceFolder.-

"Select Destina-

GetWindowHandle ("Thu

"Select Source Fold-

GetWindowHandle ("Thu

buttons and provide their implementations as follows:

its code can now be used inside our macro too.

Private Sub cmdBrowseDest Click()

PAINT's HTMLSlideshow macro. You can just copy that code

Slideshow macro project in VBA's Project Explorer and locate

Button clicked

rosa...

· Applica

croStora

ros show

er

...

Wisual Studio Tools for.

Browse...

Browse...

False 8H8000000

100

рх

100

tions

docker:

er:

cmdResize CommandButton Alphabetic Categorized

(Name) Accelerator AutoSize

BackColor BackStyle

Cancel

Caption

docker.

🖃 🝇 ImageResizer

- Torms

---- Modules Resizer

Controls

🖪 🝇 HTMLSlideshow (Slidesh 🗏 🀹 ImageResizer

.... UserForm1

Resizer Module

Corel PHOTO-PAINT X7

OK

Hello, world!

---- Modules Properties - UserForm1 X UserForm1 UserForm Alphabetic Categorized UserForm1 ☐ &H8000000 BackColor &H8000001 BorderColor 0 - fmBorderSt BorderStyle Caption UserForm 1 Cycle 0 - fmCycleAllF DrawBuffer 32000 Note the Properties docker in VBA shows various customization options for the user form. For example, the Caption property specifies the text shown in the title bar of the form. Replace the default "UserForm1" string with "Image Resizer". Now switch to "Resizer" macro module editor (double click on "Resizer" under "Modules" in the VBA Project Explorer). And change the macro to show the new dialog box instead of the "Hello, world" message box but editing the code to read: Sub Start() UserForm1.Show End Sub Now you can run the macro to see how the new dialog box looks like. Run it from Corel PHOTO-PAINT's Macro Manager docker or you can do this directly from VBA editor by placing text cursor anywhere within the Start subroutine code and hitting F5 to run it:

1 41%

Image Resizer

100 %

Y A

l⇔l 0 рх

Now that a dialog box can be shown by Image Resizer

macro, it's time to add some controls to it. Toolbox in VBA

form editor contains all the user controls currently available:

X

Manager

<u> iii</u>

×

Browse...

Name

cmdBrowseSource

IblSourceFolder txtSourceFolder

IblDestFolder

txtDestFolder cmdBrowseDest

IblWidth

txtWidth

lblHeight

txtHeight

tions

IblWidthUnit

IblHeightUnit

cmdResize

cbMaintainPropor-

- 4 B 2 0

asic for Applications

> alMacros /ILSlideshow geResizer

Start

. his Macro Storage

Macro Manager New ▼ Load... 🎒

The toolbox contains the following controls: text label, textbox, combobox, listbox, checkbox, radio button, toggle button, group box, command button, tab strip, multipage control, scroll bar, spin button, image. More controls can be added to the toolbox from third-party control libraries. To add a control to a form, just drag the control out of toolbox and drop it onto the form at the desired location. Once dropped onto the form, controls can be selected, moved and resized just like objects in CDGS applications. Clicking an already-selected control allows to edit the contents of the control (e.g. the caption of a text label control or the text of a text box). Place label controls, text boxes, command buttons and the checkbox as shown in the following screenshot: 🚰 Microsoft Visual Basic for Applications - ImageResizer - [ImageResizer -

🦂 <u>F</u>ile <u>E</u>dit <u>V</u>iew <u>I</u>nsert F<u>o</u>rmat <u>D</u>ebug <u>R</u>un <u>T</u>ools <u>A</u>dd-Ins <u>W</u>indow <u>H</u>elp

Source Folder:

Destination Folder:

Width:

When a control is selected, you can also give it a more de-

scriptive name to identify the control in the macro code, be-

cause the automatic generated control names such as Text-

Box1 and Label2 are not very descriptive. The control name

can be specified in the "(Name)" property in the Properties

The following controls are used in the form, including their

Control Type

CommandBut-

CommandBut-

type and name (left-to-right and top-to-bottom):

Label

ton

ton

Label

Label

Label

Label

ton

TextBox

TextBox

CheckBox

CommandBut-

Now that all the controls are properly placed on the form, try

running the macro again to see how the real form will look like when run from Corel PHOTO-PAINT's Macro Manager

Label

TextBox

TextBox

Image Resizer

Maintain proportions Resize Images

👸 🔤 - 🔛 | X 📭 🖺 AA 🔊 (*) 🕨 🕨 🔳 🗷 💥 🚰 👺 😥

Corel PHOTO-PAINT X7 (64-Bit) g P X <u>F</u>ile <u>T</u>ools <u>W</u>indow <u>H</u>elp 41% >> X: 0 px ⊫ 0 px 100 % C F # K H ○ 0.0 ° 💠 页 0 px 100 % Macro Manager R æ New ▼ Load... 🎒 Macro Manager ■ Visual Studio Tools for Applications Applications Image Resizer os 섷. show Г Browse... Source Folder: croStorage Destination Folder: 12 Width: Height: 100 DX Maintain proportions V p Π, 1 <u> iii</u> 0 ightharpoonsHowever you will notice that clicking any buttons has no effect because there is no code written to specify what should happen when a button is clicked. This can be done simply by double-clicking a command button in VBA form designer. For example, double-click the "Resize Images" button. This will open the code window for the user form and add an event handler for mouse click for cmd-Resize control: 🚰 Microsoft Visual Basic for Applications - ImageResizer - [UserForm1 (Co... File Edit View Insert Format Debug Run Tools Add-Ins Window 🛅 💹 - 🔛 | % 📭 🖺 🗚 | 🤟 (* | 🕨 - III Project - ImageResizer 🗶 cmdResize Click Private Sub cmdResize_Click() 🖽 🝇 GlobalMacros (GlobalMa End Sub HTMLSlideshow (Slidesh 🖃 🝇 ImageResizer UserForm1 ⊡.... (See Modules

Here, both event handlers are pretty much identical, except that they use different text controls (txtSourceFolder and txtDestFolder) to obtain the current folder name entered in those controls, call BrowseForFolderDlg() function from FolderPicker module and the new folder path returned is set back into the respective text box. Adding Image Processing Functionality Now we are ready to implement the main image processing function to do all the image manipulation. Change the cmd-Resize_Click() function to be the following: Private Sub cmdResize Click() Dim sourceFolder As String, destFolder As String Dim width As Long, height As Long Dim maintainAspect As Boolean sourceFolder = txtSourceFolder.Text If (CorelScriptTools.FileAttr(sourceFolder) And vbDirectory) = 0 Then MsqBox "Invalid source folder " & sourceFolder, vbCritica 1 Exit Sub End If If Right(sourceFolder, 1) <> "\" Then source-Folder = sourceFolder & "\" destFolder = txtDestFolder.Text If (CorelScriptTools.FileAttr(destFolder) And vb Directory) = 0 **Then** MsqBox "Invalid destination folder " & destFolder, vb-Critical Exit Sub End If If Right(destFolder, 1) <> "\" Then destFolder = destFolder & "\"

If Not IsNumeric(txtWidth.Text) Then MsgBox "Invalid width", vbCritical Exit Sub End If width = CLng(txtWidth.Text) If Not IsNumeric(txtHeight.Text) Then MsgBox "Invalid height", vbCritical Exit Sub End If height = CLng(txtHeight.Text) maintainAspect = cbMaintainProportions.Value ResizeImages sourceFolder, destFolder, width, height, maintainAspect Unload Me

the output folder specified by destFolder. tional logic to preserve the image aspect ratio. al destFolder As String, As Long, Dim f As String Dim doc As Document, w As Long, h As Long

End Sub This function takes the input from source and destination folder text boxes and validates that they contain valid folder names. Then it converts the required width and height of the image from the content of the text boxes (strings) to integer numbers which will be used when resizing images. Finally, it calls the Resizelmages function to do the actual image conversion. After all the images are processed inside ResizeImages, the dialog is closed by invoking the Unload method.

When maintainAspect variable is set to True, there is addi-ByVal maintainAspect As Boolean)

ResizeImages iterates over all the files inside a folder identified by sourceFolder, opens each one of them in Corel PHO-TO-PAINT, resize each image and saves them as JPEGs into

Sub ResizeImages (ByVal sourceFolder As String, ByV ByVal width As Long, ByVal height

f = Dir(sourceFolder & "*.*") While f <> "" doc = OpenDocument(sourceFolder & f) w = width

h = height

If maintainAspect Then If w * doc.SizeHeight < h * doc.SizeWidth Th</pre> en h = h * doc.SizeHeight / doc.SizeWidth Else w = w * doc.SizeWidth / doc.SizeHeight

End If End If doc.Resample w, h doc.SaveAs(destFolder & f, cdrJPEG).Finish doc.Close f = Dir()

Wend End Sub lems you may encounter in your day-to-day work.

Now we have a fully functional macro that shows a custom dialog, accepts user input and processes a bunch of images using Corel PHOTO-PAINT. Feel free and experiment with the source code and build more complex solutions to the prob-Download source: MacroUI.zip Share History

Related Creating a VBA for ...

6 - Making macros ... I want to see if this wa... An important part of ... API 18, Point interf... 6.3 - Providing use... The point interface ha... One way to make your...

> Addons: Extending Application Functionality with VBA and **VSTA** General Articles

Controlling CorelDRAW or Corel DESIGNER Applications From Other Processes **Creating Color Palettes**

Creating Custom Outline Enhanced Pattern Styles Creating VBA Macros with User Interface in CorelDRAW and Corel DESIGNER Macro Management in CorelDRAW Graphics Suite and CorelDRAW Technical Suite Storing Custom Information in Documents Using Corel Query Language (CQL) to Search for Objects

in CorelDRAW and Corel DESIGNER Documents

© Corel Corporation. All rights reserved. The content herein is in the form of a personal web log ("Blog") or forum posting. As such, the views expressed in this

site are those of the participants and do not necessarily reflect the views of Corel Corporation, or its affiliates and their respective officers, directors, employees

and agents. Terms of Use / Privacy / Cookies / Terms and Conditions / User Guidelines.

Developer Area requires membership for participation - click to join

Using JavaScript with CorelDRAW