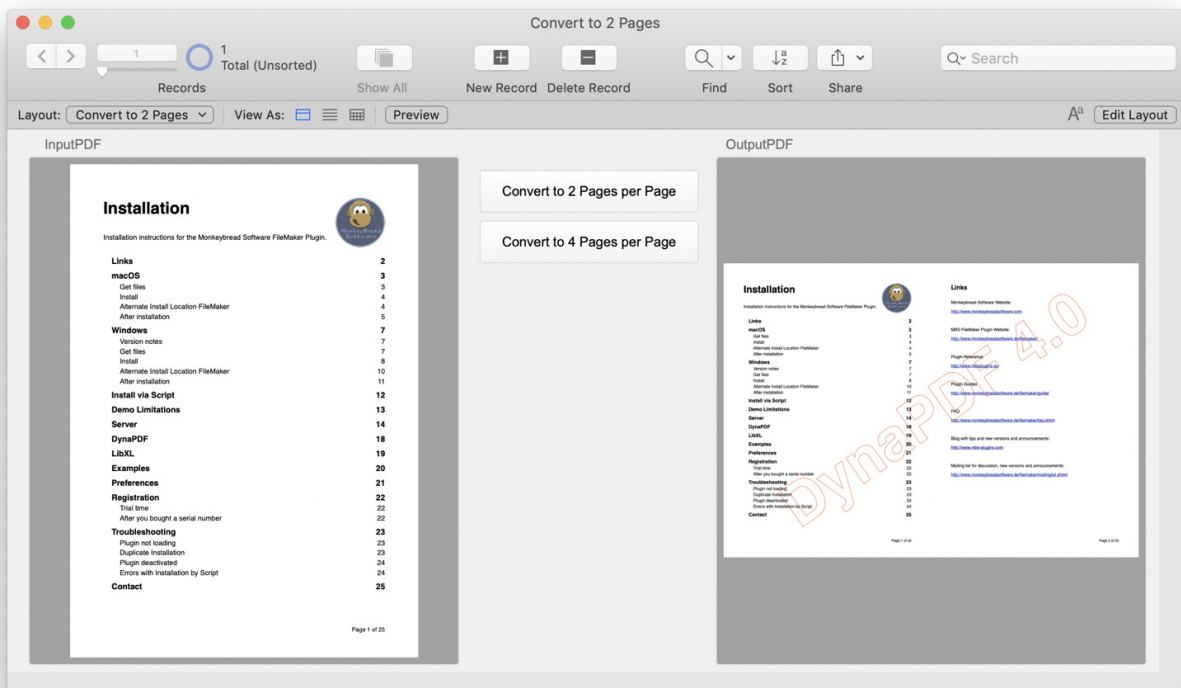


Create two page PDF document in FileMaker

You want to convert a single page document to a two pages side by side document or even more pages per page with FileMaker?

With the help of the [MBS FileMaker Plugin](#) and our [DynaPDF](#) functions you can do it. You find an example database in the example folder coming with the plugin.



Here the script step by step:

At first we initialize DynaPDF. Next we create a new PDF environment.

```
Set Variable [ $pdf ; Value: MBS("DynaPDF.New") ]
```

We load our single page document from the container. It is also possible to load it from a file with [DynaPDF.OpenPDFFromFile](#) function.

```
Set Variable [ $r ; Value: MBS("DynaPDF.OpenPDFFromContainer"; $pdf; Convert to 2 Pages::InputPDF) ]
```

We query the number of pages of the currently open import document.

```
Set Variable [ $pageCount ; Value: MBS("DynaPDF.GetImportPageCount"; $pdf) ]
```

If the document include pages we query the size of the first page of the import document. The media box defines the paper size:

```
Set Variable [ $bounds ; Value: MBS("DynaPDF.GetImportPageBounds";  
$pdf; 1; "MediaBox") ]
```

The function [DynaPDF.GetImportPageBounds](#) return a list with the left, top, right and bottom values of the page rectangle.

The list e.g. looks like this: 0.000000 841.000000 59.000000 0.000000
We must delete all "," from the list and replace after that the "." with "," because some countries have an other numerical presentation. Some countries, write a half in decimal as 0,5 and other countries as 0.5. Because of this we need sometimes the substitution. Afterwards the list look like this: 0,000000 841,000000 59,000000 0,000000

```
Set Variable [ $bounds ; Value: Substitute(Substitute($bounds; ","; "");  
"."; ",") ]
```

Then we save each item of the list in a separate variable and determine height and width of the page.

```
Set Variable [ $boundsLeft ; Value: GetAsNumber(MiddleValues  
( $bounds ; 1 ; 1 )) ]  
Set Variable [ $boundsTop ; Value: GetAsNumber(MiddleValues  
( $bounds ; 2 ; 1 )) ]  
Set Variable [ $boundsRight ; Value: GetAsNumber(MiddleValues  
( $bounds ; 3 ; 1 )) ]  
Set Variable [ $boundsBottom ; Value: GetAsNumber(MiddleValues  
( $bounds ; 4 ; 1 )) ]  
Set Variable [ $w ; Value: Abs($boundsRight - $boundsLeft) ]  
Set Variable [ $h ; Value: Abs($boundsBottom - $boundsTop) ]
```

Alternatively you can use [Math.TextToNumber](#), which always uses English format and not localized format.

After it we import each page as templates in a loop to place them on the new pages together.

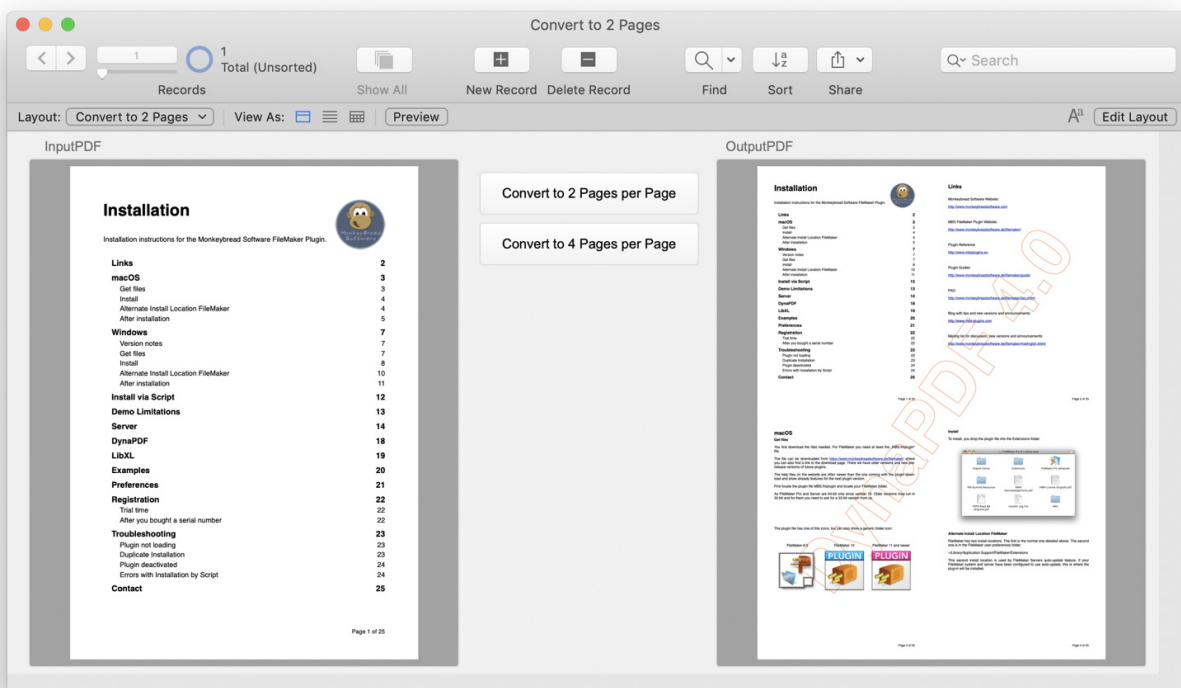
In that loop we take a look whether the actual imported page has an even or an uneven page number. We test int using the modulo operation. The modulo of a number is the remainder of a division with this number.

e.g

$$5 : 3 = 1 \text{ remainder } 2$$

So the result of $\text{Mod}(5; 3)$ would be 2.

We loop over the pages and count up. If we are on an uneven page number, we know that it is the first page we want insert on a new page. So we may need to end the existing page and append a new page to the document. Then we set the width and height of the new page. The value in our example is twice as large as the original page format, because we want to place one page left and one page right. The height of a page in this case is still the same. You can modified this example on the positions e.g. for four pages on one side. In this case you would calculate the modulo of 4 and if you get 1 as result, you would start a new page. Now you may want to place the following three pages on right and bottom parts. Of course you would double the page height to make room available.



After the determination of the page size, we position the page on the new side. To do this, we enter the rectangle of the page with the values left, top, right and bottom in the function [DynaPDF.PlaceTemplateEx](#). That call we must do separately for the left and the right page, because they have different positions on the side.

```

Set Variable [ $page ; Value: 1 ]
Loop
  MBS("DynaPDF.ImportPageAsTemplate"; $pdf; $page) ]
  Set Variable [ $m ; Value: Mod ( $page ; 2 ) ]
  If [ $m = 1 ]
    # first page of two
    If [ $page > 1 ]
      Set Variable [ $r ; Value: MBS("DynaPDF.EndPage"; $pdf) ]
    End If
    Set Variable [ $r ; Value: MBS("DynaPDF.AppendPage"; $pdf) ]
    Set Variable [ $r ; Value: MBS("DynaPDF.SetPageWidth"; $pdf; $w
* 2) ]
    Set Variable [ $r ; Value: MBS("DynaPDF.SetPageHeight"; $pdf;
$h) ]
    Set Variable [ $r ; Value: MBS("DynaPDF.PlaceTemplateEx"; $pdf;
$template; 0; 0; $w; $h) ]
  Else If [ $m = 0 ]
    # second page of two
    Set Variable [ $r ; Value: MBS("DynaPDF.PlaceTemplateEx"; $pdf;
$template; $w; 0; $w; $h) ]
  End If
  #
  # next...
  Set Variable [ $page ; Value: $page + 1 ]
  Exit Loop If [ $page > $pageCount ]
End Loop

```

After we finished the import of all pages we close the last editing page, save the document in the output container and release the PDF environment.

```

Set Variable [ $r ; Value: MBS("DynaPDF.EndPage"; $pdf) ]
Set Field [ Convert to 2 Pages::OutputPDF ; MBS("DynaPDF.Save";
$pdf; "Merged.pdf") ]
# done, save to container
Set Variable [ $r ; Value: MBS("DynaPDF.Release"; $pdf) ]

```

If you want using our functions in your solution, you need a license of the [MBS FileMaker Plugin](#) and a [DynaPDF](#) Pro license at least. You can both test with a trail license or buy both licenses on our website.

If you have any questions about functions or licenses, don't hesitate to contact me.

By Stefanie Juchmes