

DynaPDF Parser for FileMaker

With [MBS FileMaker Plugin](#) 14.0 we include the [DynaPDF](#) Parser functions. These functions provides a top-level interface on the parser in DynaPDF. You can do various operation using them:

- Parse a page
- Find text on the page.
- Extract text on the page
- Delete text within a rectangle.
- Query coordinates for found text.
- Replace found text with new text.
- Set alternative font for new text.
- Finally write changes back to page.

The combination of these allows for a lot of things. Like finding text on a page and replacing it with new text for a little PDF editor. Or to find text and then:

- Draw rectangles around the found text to show it.
- Put highlight annotations on the text.
- Find website names or keywords and put WebLink annotations on them.
- Use DeleteText function to remove the text from the PDF.
- Find each character and then know the coordinates of every letter.

The possibilities from the handful of functions is enormous. Let's take a look on sample code:

Find and Replace Text all pages in file Find and Replace Text

```
# Find Text all pages and replace with new text
#
Set Variable [ $StartTime ; Value: Get(CurrentTimeUTCmilliseconds) ]
# Initialize DynaPDF if needed
If [ MBS("DynaPDF.IsInitialized") ≠ 1 ]
    Perform Script [ Specified: From list ; "InitDynaPDF" ; Parameter: ]
End If
# Clear current PDF document
Set Variable [ $pdf ; Value: MBS("DynaPDF.New") ]
# Load PDF from container
```

```

Set Variable [ $r ; Value: MBS("DynaPDF.OpenPDFFromContainer"; $pdf;
DynaPDF Replace Text::InputPDF) ]
# import a page
Set Variable [ $r ; Value: MBS("DynaPDF.ImportPDFFile"; $pdf; 1) ]
#
# initialize parser
Set Variable [ $OptimizeFlags ; Value: 0 ]
Set Variable [ $r ; Value: MBS( "DynaPDF.Parser.Create"; $pdf; $OptimizeFlags ) ]
Set Variable [ $r ; Value: MBS( "DynaPDF.Parser.SetAltFont"; $pdf; "Helvetica"; 0;
12) ]
#
#
# loop counting up from 1 to $count
Set Variable [ $count ; Value: MBS("DynaPDF.GetPageCount"; $pdf) ]
Set Variable [ $index ; Value: 1 ]
If [ $index ≤ $count ]
    Loop [ Flush: Always ]
        # Now parse a page
        Set Variable [ $r ; Value: MBS("DynaPDF.Parser.ParsePage"; $pdf;
$index; "EnableTextSelection") ]
        If [ MBS("IsError") ]
            Show Custom Dialog [ "Failed to parse page" ; $r ]
        Else
            # Run a find
            Set Variable [ $continueFind ; Value: 0 ]
            Set Variable [ $r ; Value: MBS( "DynaPDF.Parser.FindText"; $pdf;
DynaPDF Replace Text::SearchText; "CaseInsensitive"; $continueFind) ]
            If [ $r = 1 ]
                Loop [ Flush: Always ]
                    # Do the replace
                    Set Variable [ $r ; Value:
MBS( "DynaPDF.Parser.ReplaceSelText"; $pdf; DynaPDF Replace
Text::ReplaceText) ]
                    #
                    # Continue search
                    Set Variable [ $continueFind ; Value: 1 ]
                    Set Variable [ $r ; Value:
MBS( "DynaPDF.Parser.FindText"; $pdf; DynaPDF Replace Text::SearchText;
"CaseInsensitive"; $continueFind) ]
                    Exit Loop If [ $r ≠ 1 ]
                End Loop
            #
            # Save changes back
            Set Variable [ $r ; Value:
MBS( "DynaPDF.Parser.WriteToPage"; $pdf; $OptimizeFlags) ]

```

```

        End If
    End If
    #
    # next
    Set Variable [ $index ; Value: $index + 1 ]
    Exit Loop If [ $index > $count ]
End Loop
End If
#
# save
Set Field [ DynaPDF Replace Text::OutputPDF ; MBS( "DynaPDF.Save"; $pdf;
"output.pdf" ) ]
Set Variable [ $EndTime ; Value: Get(CurrentTimeUTCMilliseconds) ]
Show Custom Dialog [ "Time" ; ($EndTime - $StartTime) ]
# Cleanup
Set Variable [ $r ; Value: MBS("DynaPDF.Release"; $pdf) ]

```

This does a search and replace while looping over all pages. Instead of doing the replace, we could of course query the coordinates on the page for later using the [DynaPDF.Parser.SelectionBBox](#) function. Then after we found all the positions, we can call EditPage to make changes and for example draw rectangles into the found places. Or add annotations.

Please try the new functions and let us know.