Parameter Parsing / Embedded Commands

Parameters or "Embedded Commands" can be entered into the originating document to instruct pdfMachine on particular actions to take when generating the PDF. Parameter parsing makes pdfMachine the ideal product to produce PDF files from legacy applications.

Read more (/genp/legacyMerge.html) on how to quickly use pdfMachine to perform batch mode email merges from legacy applications.

The available parameters are :

Parameters	
Email	Email sending
SavePDF	Save to a file name
AppendPDF	Append a PDF
InsertPDF	Insert a PDF
OpenPass	Set the Open password
PermPass	Set the change permissions password
Stationery	Set or change the PDF stationery
DigSig	Apply the digital signature
Embed	Embed a file
InsertImage	Insert an image
SplitPDF	Split a large PDF into smaller PDF's
bm	Set a bookmark
print	Print PDF to another printer . This parameter only works when Adobe Reader XI or earlier is installed.
l	Link. Create a hyperlink.
i	Image button locator

Parameters	
ii	Insert Image
il	Image Link
ilqs	Image Link Query String - Name Value Pair
x_embed	Embed a ZUGFeRD XMLfile
x	Modify a ZUGFeRD XML tag
x_clone	Copy a ZUGFeRD XML tag
x_a	Modify a ZUGFeRD XML attribute

See Examples

e.g. If the text:

#_ savePDF C:\tmp\savedfile.pdf _#

appears in a Word document and it is printed to pdfMachine with parameter parsing enabled the file will be automatically saved to c:\tmp\savedfile.pdf and no pdfMachine user interface will be displayed.

If parameter parsing is enabled, pdfMachine will parse the PDF file for any of the specified parameters listed above and described in detail below. These parameters control both the content of the generated PDF file (eg inserting images, replacing keywords with text) and the way that pdfMachine works (eg setting passwords, emailing the PDF, saving the PDF).

This is particularly useful for integration with legacy software to do "mail merge" type operations.

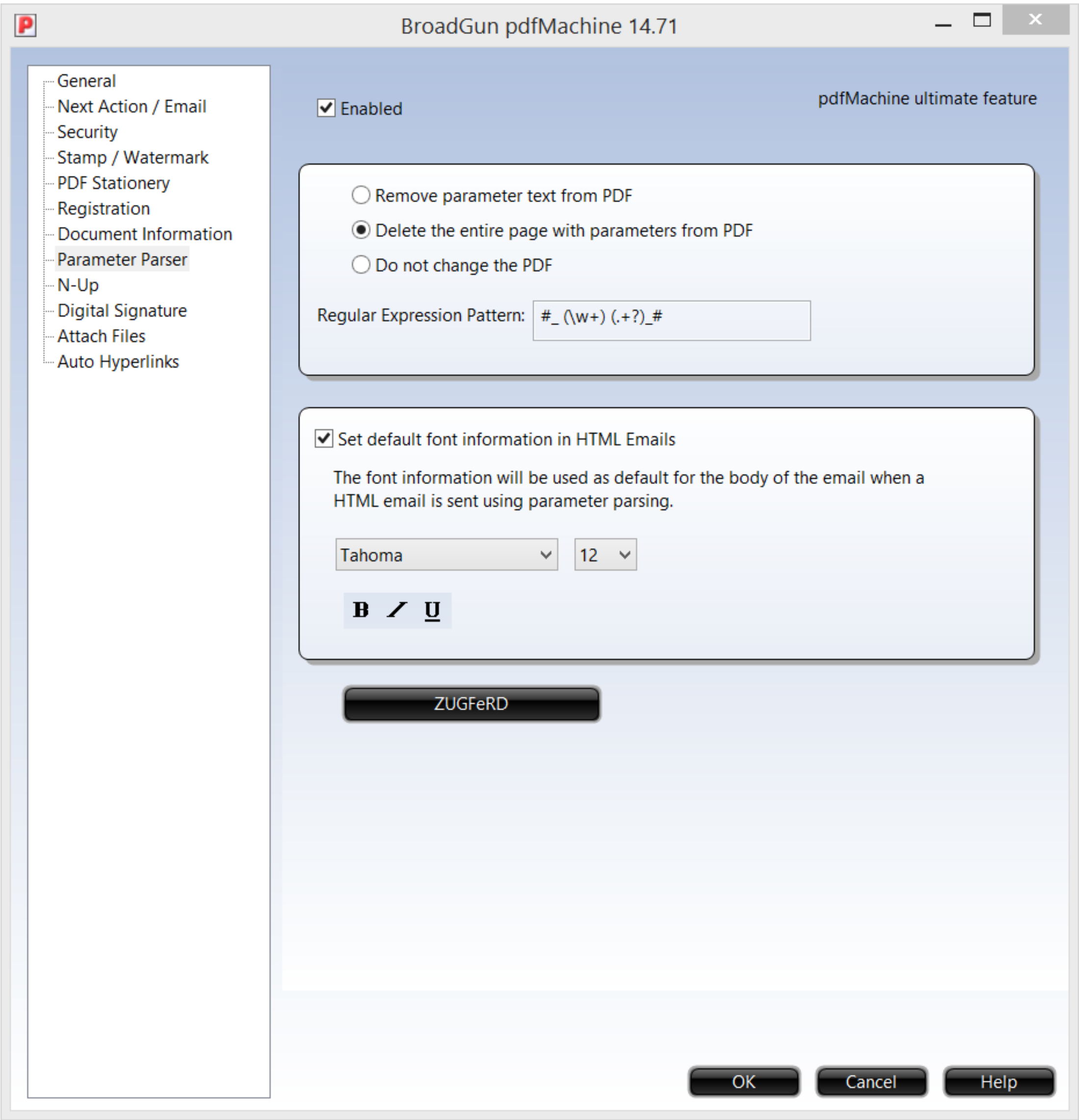
WARNING: Parameters in pdfMachine can be very powerful and complex. Please test each configuration thoroughly before using in a production environment.

How to use Parameter Parsing

1. Enable it in the options.
2. Create the originating document with parameters.
3. Print the document to pdfMachine by performing a File Print.

Setting up pdfMachine for parameter parsing

Check the "Enabled" check box in the Parameter Parser tab of the pdfMachine Options dialog to turn parameter parsing on. Enabling parameters will slow processing of pdfMachine down a little, so don't do it unless you need them.



☑ Enabled

○ Remove parameter text from PDF

● Delete the entire page with parameters from PDF

○ Do not change the PDF

Regular Expression Pattern: #_ (\w+) (.+?)_#

☑ Set default font information in HTML Emails

The font information will be used as default for the body of the email when a HTML email is sent using parameter parsing.

Tahoma 12

After parsing the parameters pdfMachine will either remove from or leave in the parameters from the resultant PDF file as selected by the radio buttons in this dialog.

The parameter parsing is done using the "Perl-Compatible Regular Expressions" library developed by Philip Hazel. A regular expression is a pattern that is matched against a subject string from left to right. For a detailed explanation of regular expressions, please read:

http://www.pcre.org/ (http://www.pcre.org/)

The default regular expression used is: #_ (\w+) (.+?)_#

We recommend this regular expression is not changed without a good understanding of regular expressions. Should this accidentally be changed it can be corrected using the default regular expression described on this page.

If you need to change the regular expression, it is stored under the following registry entry:

HKCU\Software\pdfMachine\RegExp

The default regular expression will allow parameters to be entered into the originating document in the following format:
#_ parameter argument(s) _#

Note: A parameter and an argument list for that parameter must both be specified, even if the arguments are not used, at least 1 argument must be present. For example:

A valid entry for setting a digital signature in the PDF file is : #_ digsig 1 _#
An invalid entry for setting a digital signatures is : #_ digsig _#

Initializing font for HTML

When you use parameter parsing to send an HTML email you can set the default font information for the body of your email. The font information will be used to surround the body of the email with appropriate HTML. This is only applicable for send methods which allow HTML emails to be sent.

Parameters

The parameters are not case sensitive. Choose from the parameters listed below:

Sending an email with the PDF Attachment

The following parameters can be used to cause pdfMachine to email the resultant PDF. See example.

Parameter	**Description**
Email	*Arguments:* Comma separated list of email addresses to send to. e.g. #_ email dave@broadgun.com _#
EmailCC	*Arguments:* Comma separated list of email addresses to cc to. e.g. #_ emailCC dave@broadgun.com _#
EmailBCC	*Arguments:* Comma separated list of email addresses to bcc to. e.g. #_ emailBCC dave@broadgun.com _#
EmailSubject	*Argument:* Subject of email e.g. #_ emailSubject Your invoice is attached _#

Parameter	**Description**
Emailbody	*Argument:* Text body of email. May span across multiple lines. e.g. #_ emailBody Hi, Please find attached a very useful PDF file. Best Regards, The Boss. _#
EmailAttachmentName	*Argument:* Name to use for attached PDF file e.g. #_ emailAttachmentName invoice.pdf _# Note: Do not use the following invalid Windows filename characters as they will be removed. \ / : * ? " < > |
EmailAttachmentsList	*Argument:* A list of files to attach to the PDF. They must exist, and have the full path name defined. The list should be separated by semi-colons with no spaces in between. e.g. #_ emailAttachmentsList c:\invoices\123.docx;c:\receipts\123.pdf _# Note: Do not use the following invalid Windows filename characters in the actual filename component as they are illegal characters for a filename and will result in an error. \ / : * ? " < & &rt; |

Parameter	Description
MergeField	*Arguments:* fieldname value The MergeField parameter allows the contents of the Email to be customized to a finer detail. That is, the subject, body and attachment name can be set in the pdfMachine options screens. The {} characters can be used to define field names within these and then the MergeField parameter used to set these field names. See example. MergeField can only be used when the Email parameter is set. Takes a fieldname to replace, and the value to replace that fieldname with. The mergeField is applied to the EmailSubject, EmailBody and EmailAttachment name settings. e.g. #_ MergeField first_name Fred _# This will replace all occurrences of {first_name} in the subject, body or attachment name settings, with the word Fred.

Launching the default email client with the PDF Attachment

The following parameters can be used to cause pdfMachine to prepare to email the resultant PDF. The email can then be checked or corrected before it is sent. The EmailTo parameter replaces the Email parameter. The other Email parameters described in the previous table can also be used. See example.

Parameter	Description
EmailTo	*Arguments:* Comma separated list of email addresses to send to. This is used *instead* of the Email parameter described above. This prefills the email parameter but does not send the email. e.g. #_ EmailTo dave@broadgun.com _#
NextAction	*Arguments:* 4 To launch the default email client with the prefilled values set by the parameters a NextAction of 4 should be used. NextAction is only valid to use when the EmailTo parameter is set. e.g. #_ NextAction 4 _#

Save the PDF File

The following parameter causes pdfMachine to automatically save the resultant PDF file to the specified location.

Parameter	**Description**
SavePdf	*Argument:* Path to the location to save the PDF file e.g. #_ SavePdf C:\savedFileHere.pdf _#

Inserting or Appending PDF Files

The following parameters insert or append files into the resultant PDF file .

Parameter	**Description**
AppendPdf	*Argument:* Path to the PDF file that is to be appended. e.g. #_ AppendPDF c:\somefile.pdf _# Multiple appendPdf parameters can appear in the file which will result in multiple PDF's being appended. See example
InsertPdf	*Argument:* Page number after which to insert (0 to insert at the start of file), path to the PDF file to insert. See example.

Encryption

The following parameters cause pdfMachine to enable encryption and to set the appropriate passwords in the resultant PDF file.

Parameter	**Description**
OpenPass	*Argument:* Password required to open the document (a.k.a. user password). e.g. #_ OpenPass mysecret _# NOTE : Set the other encryption parameters in the pdfMachine options screens.

Parameter	Description
PermPass	*Argument:* Permissions password (a.k.a. owner password). e.g. #_ PermPass mysecret _# NOTE : Set the other encryption parameters in the pdfMachine options screens.

Stationery

The following parameter overrides the current pdfMachine stationery settings. There are two settings, one for all pages and one for first page only. If you want to completely control the stationery in parameter parsing then you should specifically set both of them.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Stationery</td><td><i>Arguments:</i> pages drawn path
<table>
<tr><td>pages:
all | first | none</td><td>Optionally specify which pages the stationary is applied:
"all" (default) applies the stationery to all pages of document.
"first" applies it to first page only.
"none" turns off all stationery.</td></tr>
<tr><td>drawn:
<b>below</b> | above | none</td><td>Optionally specify how the stationery is drawn:
"below" (default) draws the stationery underneath the page text.
"above" draws the stationery above the page text.
"none" turns off either the stationery for all pages or for the first page only, depending on the first argument.</td></tr>
<tr><td>path</td><td>Path to PDF file to be used as stationery.</td></tr>
</table>
e.g. Set the stationery for 'all pages' to be c:\stationery.pdf and draw it on top of the page text. Any 'first page only' stationery will still be used.
#_ Stationery all above c:\stationery.pdf _#

e.g. Set the stationery for 'all pages' to be c:\stationery.pdf and draw it on top of the page text. Turn off any 'first page only' stationery that has been defined.
#_ Stationery all above c:\stationery.pdf _#
#_ Stationery first none _#

e.g. Turn off all stationery.
#_ Stationery none _#</td></tr>
</table>

Digital Signature

The following parameter causes pdfMachine to Inserts the selected digital signature at the current place in the file.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>DigSig</td><td><i>Argument:</i> visible enabled
<table>
<tr><td>visible:
<b>show</b> | noshow</td><td>Optionally specify if signature is visible:
"show" (default) shows the signature appearance as specified in the options settings.
"noshow" document is signed but has no signature picture or words.</td></tr>
<tr><td>enabled:
<b>on</b> | off</td><td>Optionally specify if signature is done:
"on" (default) signs document.
"off" does not sign document. Allows you to override digsig being enabled in the options.</td></tr>
</table>

e.g. #_ DigSig noshow _#

NOTE : the DigSig parameter *must* be followed by at least one other piece of text so that the regular expression can find it. Anything other that the parameters shown above are ignored, so it is fine to just put something simple like '1'. e.g. #_ DigSig 1 _#

NOTE : Set up the other signature details in the pdfMachine options.</td></tr>
</table>

Files Attached / Embedded within the PDF

The following parameter causes pdfMachine to attach file(s) to the resultant PDF file. The attached files can be seen in the attached files list in Acrobat Reader 6 and 7.

Parameter	Description
embed	*Argument:* Path of the file to be embedded. e.g. #_ embed C:\afile.pdf _# #_ embed C:\somefile.pdf _# Call embed multiple times to embed multiple files. See example.

Insert Image

The following parameter causes pdfMachine to insert images into the resultant PDF file.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>InsertImage</td><td><i>Arguments:</i> width height path
e.g. #_ InsertImage 100 100 C:\somefile.jpg _#
<table>
<tr><td>width</td><td>optionally specify the width in points for the picture in the PDF file.</td></tr>
<tr><td>height</td><td>optionally specify the height in points for the picture in the PDF file.</td></tr>
<tr><td>path</td><td>path to the image file.</td></tr>
</table>
Note: 1 point = 1/72th inch
InsertImage can be called more than once per page.</td></tr>
</table>

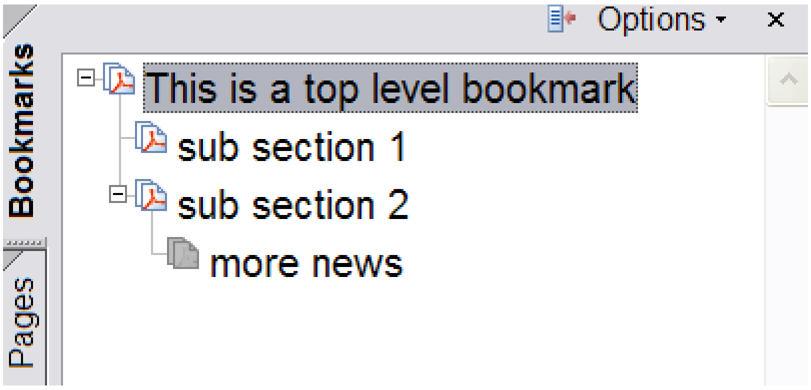
Split Document

Parameter	Description

SplitPDF	*Argument:* dummy argument to help regular expression parsing e.g. #_ SplitPDF 1 _# Note the SplitPDF parameter must be followed by some other text, which is '1' in this case. Splits document at the end of the current page. This should be used with extreme caution, as it may result in multiple emails being sent for a single print job. **Please be sure you do adequate testing to ensure the emails are as you wish.** see example Make sure that there is no more that one SplitPDF command on a single page. This can be useful for integrating with a legacy application, or Crystal Reports solution that creates one big print job with each customer statement on a new page. e.g. An accounting application may print out a number customer invoices, one after the other in one big print job. Normally this would generated 1 large PDF. However, using a combination of "SplitPDF" and "email" parameters each customer would be emailed a unique invoice belonging to them.

bookmark

Parameter	**Description**

bm	*Argument:* bookmark name\sub bookmark name A back slash separated list of heading names, forming a table of contents or bookmark hierarchy. Bookmarks provide a table of contents and usually represent the chapters and sections in a document. Bookmarks appear in the navigation pane of adobe reader. When a bookmark is clicked, the reader will be presented with the section where the "bm" command was placed. e.g #_ bm This is a top level bookmark _# #_ bm This is a top level bookmark\sub section 1 _# #_ bm This is a top level bookmark\sub section 2\more news _# Produces PDF with a bookmarks window in Adobe Reader:  Options Bookmarks Pages This is a top level bookmark sub section 1 sub section 2 more news

Print

Parameter	Description

print	*Argument:* printer name Used to print to a secondary printer. **This parameter only works when Adobe Reader XI or earlier is installed.** This could be useful if a hardcopy is desired as well as the PDF. If the word "default" is used for the printer name the default printer is used. e.g 1 - Print to a specific printer name #_ print HP Office Jet v49 _# e.g 2 - Print to the default printer #_ print default _#

insert hyperlink

Parameter	**Description**
l	*Argument:* link_text URL A hyperlink is created pointing to URL. The text 'link_text' will remain. e.g #_ l link to broadgun http://www.broadgun.com _# #_ l hi there http://www.google.com (http://www.google.com) _#

image button locator

The 'i' parameter defines the position of the image that will be inserted into the PDF by the 'ii' command.

Parameter	**Description**

i	*Argument:* tag A tag name that is used by the ii, il and ilqs commands. e.g #_ i 1 _# A complete example (http://www.broadgun.com/genp/parm_parsing_link.html)

insert image

Inserts an image into the PDF. The image is inserted at the location defined by the 'i' command.

Parameter	Description
ii	*Arguments:* tag width URL tag - An alphanumeric string that refers to the image location previously defined by the 'i' command. width - Optional. The width of the image in points. URL - Can be a http or https URL or a file path on the local PC. File types supported are: gif, png, jpg, bmp. e.g.1 #_ ii 1 100 http://www.broadgun.com/somepic.png _# e.g.2 #_ ii 2 c:\tmp\somepic.jpg _# A complete example (http://www.broadgun.com/genp/parm_parsing_link.html)

image link

Inserts a hyperlink into the PDF. The position and size is defined previously by the 'i' and 'ii' commands.

Parameter	Description
il	*Arguments:* tag URL tag - An alphanumeric string that refers to the image location previously defined by the 'i' command. URL - Can be a http or https URL. e.g. #_ il http://www.broadgun.com _# A complete example (http://www.broadgun.com/genp/parm_parsing_link.html)

Parameter	Description

image link query string

Used to build a query string for a URL. The base of the URL is defined by the 'il' command. Several 'ilqs' commands can appear all relating to the same URL.

Parameter	Description
ilqs	*Arguments:* tag name value tag - An alphanumeric string that refers to the image location previously defined by the 'i' command. name - Parameter name. This will be URL parameter encoded. value - Parameter value. This will be URL parameter encoded. A complete example (http://www.broadgun.com/genp/parm_parsing_link.html)

x_embed

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>x_embed</td><td><em>Arguments:</em> basic | ZUGFeRDprofile path_to_xml_file

This defines what XML is used as a template file to embed into the PDF. The ZUGFeRD metadata for the PDF is inserted and the xml file is renamed to ZUGFeRD-invoice.xml. If the word "basic" is used, the ZUGFeRD profile is BASIC and the template used is the one that is edited by the ZUGFeRD template form. Alternatively you can provide your own template by specifying the ZUGFeRD profile type and a full path to another file.
<table>
<tr><td>basic</td><td>The ZUGFeRD profile is BASIC and the template used is the one that is edited by the ZUGFeRD template form.
For example : x_embed basic.</td></tr>
<tr><td>ZUGFeRDprofile path_to_xml_file</td><td>A ZUGFeRDprofile (eg BASIC, COMFORT or EXTENDED) followed by a full path to another file can be used. The file is used as the template. The specified ZUGFeRDprofile is used in the PDF MetaData.
For example : x_embed COMFORT d:\tmp\invoice.xml.</td></tr>
</table>
</td></tr>
<tr><td></td><td></td></tr>
</table>

x

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>x</td><td>

Argument: XML_tag_name

This is how XML elements are set. The XML_tag_name must be unique in the XML file defined by the x_embed command.

e.g. If the XML defined by the x_embed entry was :

```
<sometag>
  <apples>
    <style></style>
    <quality>
       <type></type>
    </quality>
  </apples>
  <oranges>
    <quality>
       <type></type>
    </quality>
  </oranges>
</sometag>
```

The parameters:
#_ x style pink lady _#
#_ x apples/quality/type fresh _#
#_ x oranges/quality/type old _# Would result in the following XML being embedded into the PDF:

```
 <sometag>
  <apples>
    <style>pink lady</style>
    <quality>
       <type>fresh</type>
    </quality>
  </apples>
  <oranges>
    <quality>
       <type>old</type>
    </quality>
  </oranges>
</sometag>
```

Note that the 'type' tag was defined in multiple places, so we needed to be more specific and include the parent tag names. For the tag 'style', it was only in one place so there was no need to include the parent tag names.

</td></tr>
<tr><td></td><td></td></tr>
</table>

x_clone

Parameter	Description
x_clone	*Argument:* XML_tag_name This command copies the last occurring XML tag (and children) in the XML and appends it as a sibling to the XML tag. Its a mechanism for including creating collections in the XML.

x_a

Parameter	Description
x_a	*Arguments:* XML_tag_name XML_attribute_name

XML_tag_name	The name of the tag to which the attribute belongs.
XML_attribute name	The name and value of the XML attribute. This must be in the form attribute_name="attribute_value". For example currency="EUR"

This is how XML attributes are set. The XML_tag_name must be unique in the XML file defined by the x_embed command. XML_attribute_name will identify which attribute to modify. If the attribute exists already it will be replaced with the new value. If it doesn't exist the attribute will be created.

e.g. If the XML defined by the x_embed entry was :

```
<sometag>
  <apples>
    <style></style>
    <quality>
       <type></type>
    </quality>
  </apples>
  <oranges quantity="weight">
    <quality>
       <type></type>
    </quality>
  </oranges>
</sometag>
```

The parameters:
#_ x_a oranges quantity="number" _#
#_ x_a apples quantity="number" _#
Would result in the following XML being embedded into the PDF:

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td><pre>&lt;sometag>
  <apples quantity="number">
    <style></style>
    <quality>
      <type></type>
    </quality>
  </apples>
  <oranges quantity="number">
    <quality>
      <type></type>
    </quality>
  </oranges>
</sometag></pre>
Note that the attribute already existed for the 'oranges' tag and was therefore modified, whereas it was created for the 'apples' tag.</td></tr>
<tr><td></td><td></td></tr>
</table>

Examples

Sending an email

Print the following text to pdfMachine to send an email to user@broadgun.com, with the attachment called "mypdf.pdf" and the subject "pdfMachine is great" and the body "Hi, Please have a look at the attachment." (spread over 2 lines).

```
#_ email user@broadgun.com _#
#_ emailSubject pdfMachine is great _#
#_ EmailAttachmentName mypdf.pdf _#
#_ Emailbody Hi,
Please have a look at the attachment. _#
This is the attachment text, it will appear in the PDF.
```

Preparing an email to be sent

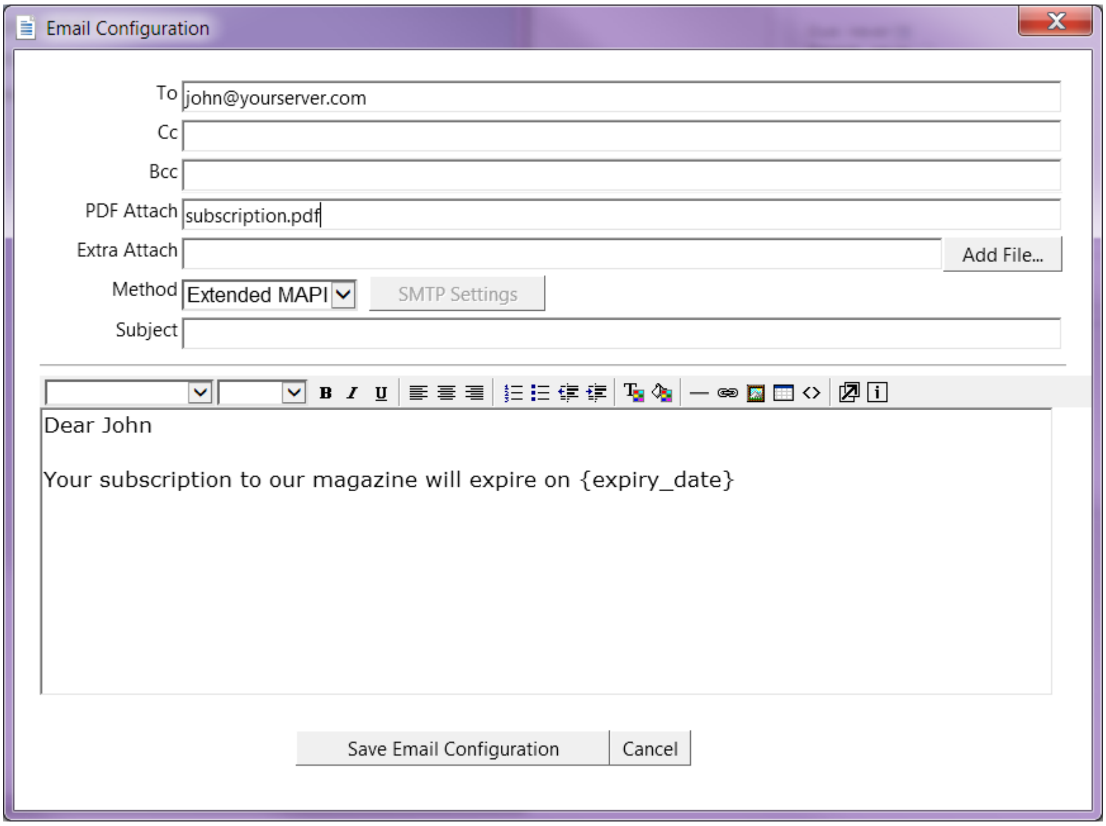
Print the following text to pdfMachine to launch the default email client without actually sending the email.

```
#_ emailTo user@broadgun.com _#
#_ NextAction 4 _#
#_ emailSubject pdfMachine is great _#
#_ EmailAttachmentName mypdf.pdf _#
#_ Emailbody Hi,
Please have a look at the attachment. _#
This is the attachment text, it will appear in the PDF.
```

Sending an email with the PDF Attachment using the MergeField parameter

You can either:

Set the pdfMachine options to have the email configuration as below:



To john@yourserver.com

Cc

Bcc

PDF Attach subscription.pdf

Extra Attach

Method Extended MAPI

Subject

Dear John

Your subscription to our magazine will expire on {expiry_date}

Then print the following text to pdfMachine to send an email to john@yourserver.com, with the attachment called "subscription.pdf" and the subject "Subscription expires on 1st March 2006" and the body "Dear John Your subscription to our magazine will expire on 1st March 2006" (spread over 2 lines).

```
#_ email john@yourserver.com _#

#_ EmailAttachmentName subscription.pdf _#
#_ EmailSubject Subscription expires on {expiry_date} _#
#_ Mergefield expiry_date 1st March 2006 _#
```

Or you can do the whole thing using parameter parsing. You do not need to setup the Email Configuration in the options. You can print the following text to pdfMachine to send an email to john@yourserver.com, with the attachment called "subscription.pdf" and the subject "Subscription expires on 1st March 2006" and the body "Dear John Your subscription to our magazine will expire on 1st March 2006" (spread over 2 lines).

```
#_ email john@yourserver.com _#

#_ EmailAttachmentName subscription.pdf _#
#_ EmailSubject Subscription expires on {expiry_date} _#
#_ EmailBody Dear {first_name}

Your subscription to our magazine will expire on {expiry_date}. _#

#_ Mergefield first_name John _#
#_ Mergefield expiry_date 1st March 2006 _#
```

Appending PDF file

Print the following text to pdfMachine to append the contents of the file 'one.PDF' and 'two.pdf' into the resultant PDF.

```
#_ appendPdf c:\one.pdf _#
#_ appendPdf c:\two.pdf _#

Put any text  here , this forms the main pdf.
```

Inserting PDF file

Print the following text to pdfMachine to append the contents of the file otherPDF.PDF into the resultant PDF at the start of the file.

```
#_ insertPDF 0 My Documents\otherPDF.pdf _#

Put any text  here - this will appear after the contents of the file otherPDF.pdf
```

Attaching/Embeding files

Print the following text to pdfMachine to embed files in the resultant PDF.

```
#_ embed c:\somefile.txt _#
#_ embed c:\anotherfile.txt _#

Put any text  here - this will appear as the contents of the PDF file.  The PDF file will
have an "attachments" window containing the two text files.
```

Splitting single print job into multiple emailed documents - SplitPDF

Print the following text to pdfMachine to send two emails.

> #_ email john@yourserver.com _#
> #_ emailSubject hello, here is the news for john _#
> #_ splitPDF 1 _#
>
> Customer Name: John
> Amount Owed: $100
> Date Due: 23 Jan 2006
>
> ..
> ..
>
> **(make sure there is a page break before this next text)**
>
> #_ email dave@yourserver.com _#
> #_ emailSubject hello, here is the news for dave _#
> #_ splitPDF 1 _#
>
> Customer Name: Dave
> Amount Owed: $200
> Date Due: 23 Jan 2006

ZUGFeRD e-invoicing

ZUGFeRD is a German e-invoicing standard. A ZUGFeRD PDF is a PDF/3-A compliant PDF file that is a visual invoice, just like any other. It also contains structured invoice data in an embedded XML document that can be processed by ZUGFeRD compliant applications. For details, please go to www.ferd-net.de (http://www.ferd-net.de)

What follows is an example set of parameters that will produce a ZUGFeRD XML when printed to pdfMachine. The XML tag names are all taken from the ZUGFeRD specification.

Note: No validation is performed on the XML, so its up to you to make sure the field values are valid.

```
#_ x_embed basic _#  // embed basic ZUGFeRD profile

// invoice id and dates
#_ x rsm:HeaderExchangedDocument/ram:ID 2323 _# // unique invoice ID
#_ x ram:IssueDateTime/udt:DateTimeString 20140711 _#
#_ x ram:OccurrenceDateTime/udt:DateTimeString 20140711 _#

// set the totals
#_ x ram:LineTotalAmount 845.00 _#
#_ x ram:ChargeTotalAmount 0 _#
#_ x ram:AllowanceTotalAmount 0 _#
#_ x ram:TaxBasisTotalAmount 845.00 _#
#_ x ram:TaxTotalAmount 160.55 _#
#_ x ram:GrandTotalAmount 1005.55 _#

// line items
#_ x ram:LineID 1 _#
#_ x ram:BilledQuantity 1 _#
#_ x ram:LineTotalAmount 800 _#
#_ x ram:SpecifiedTradeProduct/ram:Name  Pink Hammers _#

#_ x_clone ram:IncludedSupplyChainTradeLineItem _#   // copy previous line
#_ x ram:LineID 2 _#
#_ x ram:BilledQuantity 1 _#
#_ x ram:LineTotalAmount 100 _#
#_ x ram:SpecifiedTradeProduct/ram:Name  Blue Nails _#

#_ x_clone ram:IncludedSupplyChainTradeLineItem _#   // copy previous line
#_ x ram:LineID 3 _#
#_ x ram:BilledQuantity 2 _#
#_ x ram:LineTotalAmount 105.55 _#
#_ x ram:SpecifiedTradeProduct/ram:Name  Green Widgets _#
```