



**Note: This product is distributed on a 'try-before-you-buy' basis. All features described in this documentation are enabled. The registered version does not insert a watermark in your generated pdf documents.**

## About A-PDF Content Splitter

A-PDF Content Splitter is a desktop utility program that lets you split large Acrobat files into smaller (multiple parts) pdf files base on location and text information within a PDF Page. It can be used to split composite PDF documents (such as invoices, records or salaries) to pieces by invoice number, account number or employees name which statement in a pdf file. A-PDF Content Splitter will locate the position, extract text and compare text, then generate small file. It provides a power split rule editor to allow you define how the file will be split and how the split output files are uniquely named. You even can define and save multiple rules to apply different type files.

## Typical applications for A-PDF Content Splitter

This feature is ideal for large invoicing, billing, reporting and enterprise payroll systems, where a single output file may contain many thousands of individual records that need to be extracted and distributed. Text within the document can be used to dynamically name output files and extracted to a delimited file for data storage.

A-PDF Content Splitter provides also a command line function, which can be called by script. It is useful when you want to split file automatically and/or periodically.

A-PDF Content Splitter does NOT require Adobe Acrobat, and produces documents compatible with Adobe Acrobat Reader Version 5 and above.



## Working Example for A-PDF Content Splitter

Page 1

Invoice Number
9000115519
Page
23 of 1

Split tag : Pos (120,125)

Page 3

Invoice Number
9000115520
Page
23 of 3

Page 5

Invoice Number
9000115521
Page
23 of 5

Original PDF (invoice.pdf)

A-PDF Content Split

Output pattern:  
{basename}\_{splittag}

Output

- invoice\_9000115519.pdf
- invoice\_9000115520.pdf
- invoice\_9000115521.pdf
- invoice\_9000115522.pdf
- invoice\_9000115523.pdf
- invoice\_9000115524.pdf
- invoice\_9000115525.pdf
- invoice\_9000115526.pdf
- invoice\_9000115527.pdf
- invoice\_9000115528.pdf
- invoice\_9000115529.pdf
- invoice\_9000115530.pdf

**A-PDF Content Splitter examples: split when an invoice Number (or other unique identifier) on the page changes.**

## A-PDF Content Splitter Feature List

- Generates multiple pdf's from a single pdf by splitting on textual content
- Splits on unique text by using word position searches
- Splits on find text by using word position searches
- Splits on unique text by using word position searches and group output by unique text
- Support transform and validate found text by using word position searches
- Support defining text location information by sampling visualization
- Support Batch Pdf file processing. can split on text information within many PDF's
- Support defined splits rules that can be used again. The rules can be imported and exported for use in other computers.
- Support define output files name pattern.
- Support Script for output files name (such as Splittag, Marcotag, Basename, etc...)
- Support password-protected pdf files
- Support adds dynamic Meta data to output split PDF.
- Support command line.
- Log support for split process
- Allow splitting part of pages, e.g. page range etc

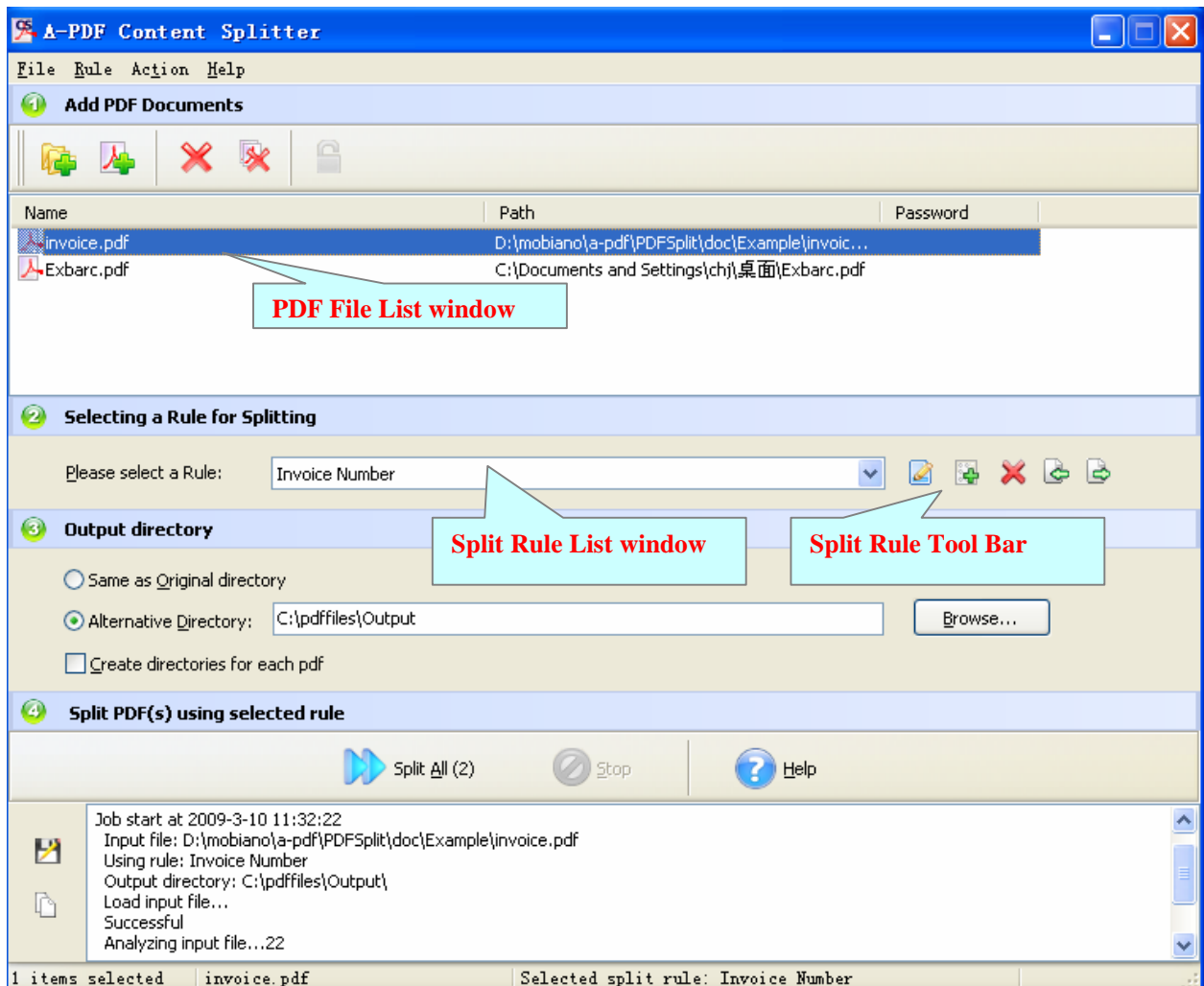


## Using A-PDF Content Splitter

A-PDF Content Splitter can be started either by launching the application via a desktop shortcut, or directly from within Windows/Windows Explorer.




When you start the program, you will be presented with the primary screen from which all actions are taken.




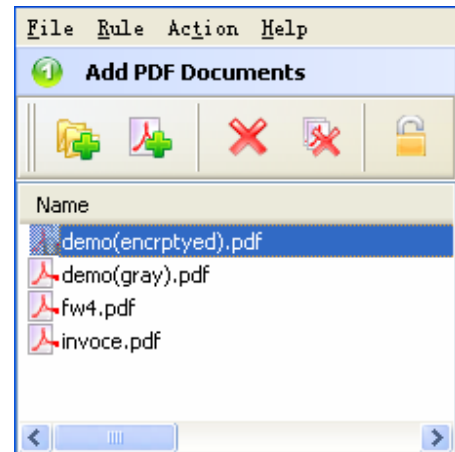


## Selecting PDF File

Clicking the **Add PDF Document** icon , or selecting **File** then choosing **Add PDF Document** will open the standard file browse and select window. From that window you can navigate to the folder and find the files you want to add to the list of files to split.

Alternatively, if you are using Windows Explorer to locate files, you have the convenience of being able to 'drag and drop' files into the document window.



A-PDF Content Splitter even allows you to select all files in a particular directory/folder by using the **Add a Directory**  option from the **File** menu! Add a directory also supports sub-folders.

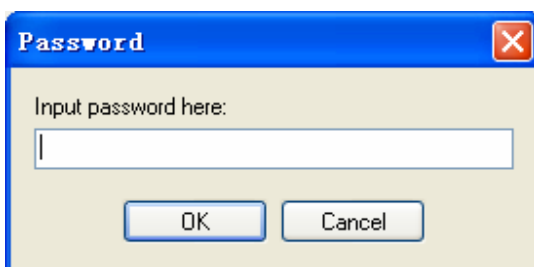


Each added file is displayed in the PDF File List window, as seen in the example below. You can view any of the files in the window by 'double-clicking' them. That will open pdf files in your default pdf reader program (usually Adobe Reader).



If any of the files you want to split are password-protected (security may be set within Adobe Acrobat for Read and other permissions), then before Adding Watermark to PDF files you need to provide the file password(s).

Files that contain passwords show a lock icon  in left. Click the file name and the **Password** icon  becomes active. Then clicking the **Password** icon produces an input box for each password required.

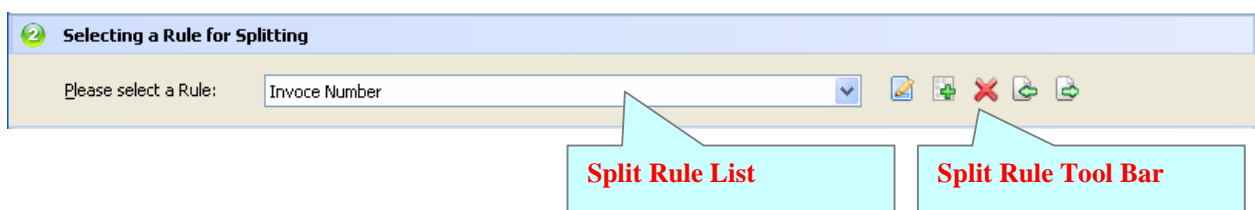




## Select Split Rule

**What is Split Rule?** A Split Rule defines text **position** information within the sample pdf. A-PDF Content Splitter according to the rule to search for text, compare text and then split the document up for you automatically. You can find an example to create a split rule below sections.

The Split Rule for applying to PDF(s) are listed in the **Split Rule List window**, where you can find rule items list in here, the current **Rule** you selected will to be applied to PDF(s)



## Split Rule Tool Bar Function





## Set Output Directory


**Same as Original directory:** The spitted PDFs will be saved to the same as Original directory.

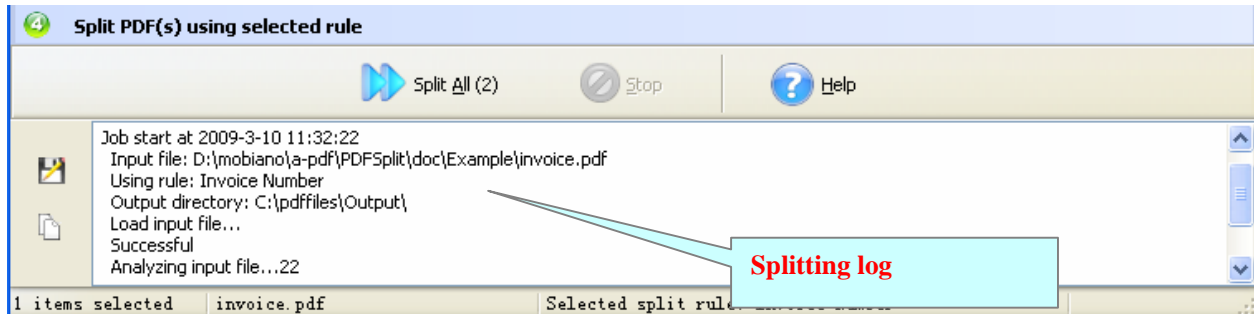
**Alternative Directory:** You can browse an alternative folder to save output

**Create directories for each pdf:** The option checked means A-PDF Content Splitter will create folders for each source PDF under the folder specified above, then save the output files to proper folders.



## Splitting PDF

Click the **Split All** button  to split all of PDF files in the PDFs document windows. A-PDF Content Splitter will to split PDF files base on the selected rule.



The log splitting will be shown in log list.




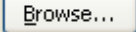
***A simple example of how A-PDF Content Splitter is used will demonstrate how easy and convenient this utility is for splitting pdf files into parts.***

## How to Create a Split Rule (Example)?

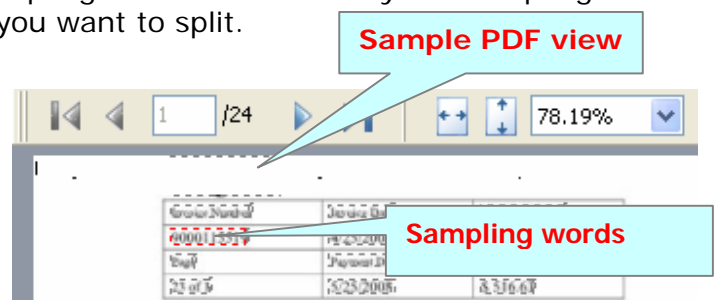
I have a PDF document generated by my financial management system (for example, "invoice.pdf"). The file contains hundreds of invoices. I want to split it on invoice number information within the PDF.

### Example for creating and applying an "Invoice Number" rule

**Step 1,** In **Split Rules list window**, click the icon  to open the **New Split Rule** configuration window. You can configure the split rule properties here. Once you set the desired options for the rule, you can click Save Button to save it.

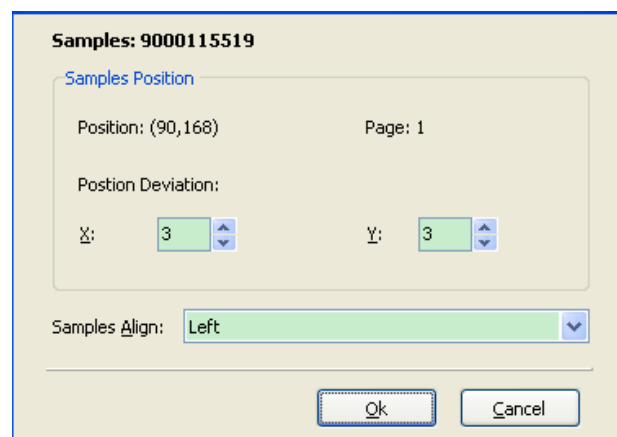
**Step 2,** Click the **Browse** button  in configuration window. You can select a pdf file (for example: "invoice.pdf") as sampling document. Normally the sampling document can be the pdf document that you want to split.

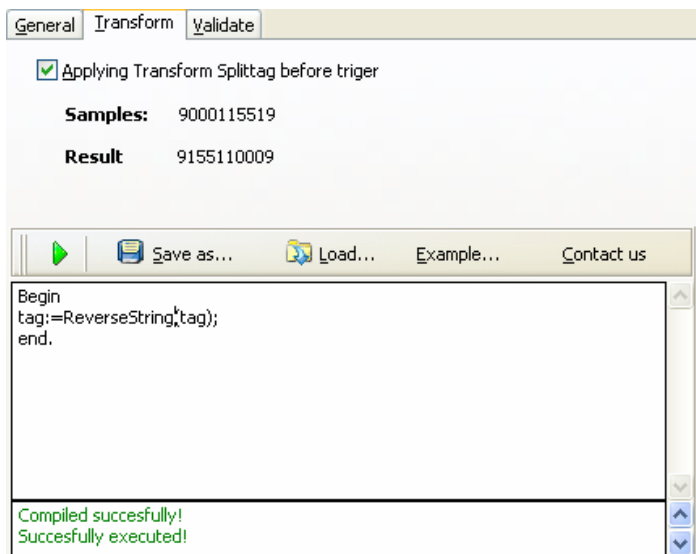
**Step 3,** In the opened **sample PDF view**. You can find the Invoice Number words (for example the sample words is: **90001155**) in it. Click left mouse button to select it. Then you will find have a **Red dashed border** on it.



**Step 4,** Click the **Add to Split Tag List ->** button  to add the **Position** of selected sample words to Split Tag list. Then open the **Add Samples to Split Tag List** configuration window. You can adjust the **Position deviation (X,Y)** to mitigate the impact of position deviation in sample PDF.

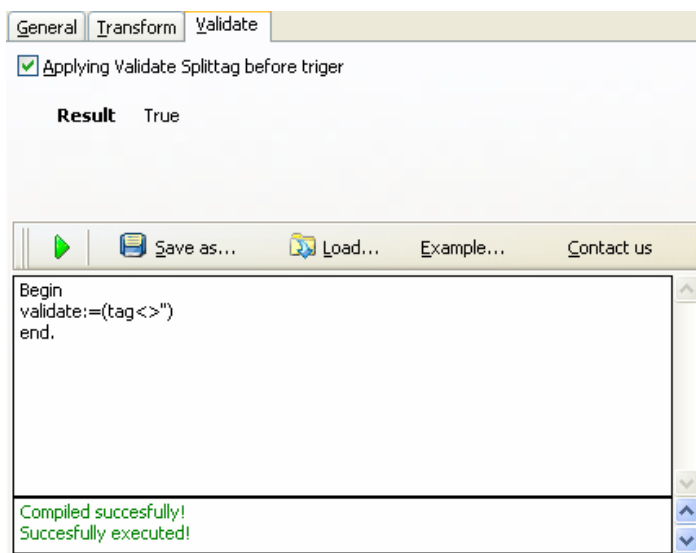
If the text align of sampling is left. The **Samples Align** value should be "**Left**".  
If the text align of sampling is middle. The **Samples Align** value should be "**middle**".  
and if the text align of sampling is right, the **Samples Align** value should be "**right**".



**Transform Splittag (Optional):**

You can transform splittag by script before you use the splittag, The benefits of transform is: you can split out the your expect results from the more complex PDF, such as I would like to just take the first three characters from splittag and use it to separate source PDF

Example : `tag:=leftstr(tag,3);`  
//take the first three characters from splittag

**Validate Splittag (Optional):**

You also can verify splittag by script before you use the splittag, the result is only valid can be used as input split tag conditions

The benefits of Validate is: You can filter the splittag line with the conditions, such as I would like to verify first three characters of splittag is the '010'

Example :  
`validate:=(LeftStr(tag,3)='010');`  
// If the splittag of the first three characters is the'010', then the

return value is true, otherwise, return false

Next Click the **OK** button to add the Samples to tag list (That means the split condition (in this case): **Finding unique "Invoice Number text" using text position searches (in same position (90,168) and in each page) until "Invoice Number text" changed**)





Position	Split Tag in position
(90,168)	Sample text : 9000115519, from page : 1)

**Step5.** Select split conditions:

**Trigger for find different text:** means search-by-page PDF to find same words at same position (x,y), until words changed, then begin to split

**Trigger for find text:** means search-by-page PDF to find words at same position (x,y), Began to split as long as found words

**Trigger for find different text and group:** means search-by-page PDF to find same words at same position (x,y), until words changed, then begin to split , a last to merge output which Have the same word

<b>Conditions for splitting:</b>	trigger for find different text and group
<b>Finding Same Words at</b>	trigger for find different text trigger for find text trigger for find different text and group
(90,169)	
<b>Until changed, Group Output by Found Words</b>	

**Step6.** Entering a name for the Rule (note the text that will appear in the Split Rules list, must be a unique name). (For example: Invoice Number)

Rule Name:	Invoice Number
------------	----------------

**Step7.** A-PDF Content Splitter allows you to choose how the split output files will be named. The '**Filename pattern**' dropdown lets you choose from several 'patterns' of nomenclature so that each of the split files is logically named and can be easily identified later. *If you need to custom output filename, please refer to "Script for Output"*

These name patterns are macros that automatically increment as necessary for the Final split files.

<b>Output Filename</b>
Filename pattern:
{basename}_{splittag}_{pagebegin}/{pageend}
Example: <b>invoice_9000115519_(1/24).pdf</b>

For example, choosing the {basename}.{part:0000} macro will name your split files from the unsplit filename and add sequential part numbers for each split file. 0000 is a place-holder and can be substituted with your own choice of size and starting part



number. An example of this pattern and macro could name split files like example.part 0002.pdf. Choosing {basename}.{pagebegin:0000}-{pageend:0000} will generate split files which include the start and end page of the original unsplit file that each split file contains.

For even more flexibility, you can enter your own macro selection to give split files any rational name and sequence you want.

Patterns and their associated macros are a great convenience for naming your split files conveniently and logically with almost no effort on your part.

If you use splitag for the purpose of splitting only, and at the same time you need to use the text information in a specific position relate to split PDF page to name the output file, you can use Marcotag Marco, for detail, please refer to "Define Marcotag Marco"

MACRO	DESCRIPTION
{basename}	The unsplit file name without extension name.
{splittag}	The text searched. For example, 900001234.
{part:O...O}	A sequential number from 1. O...O is place-holder. For example, you are using {part:0000}, will get 0001, 0002 etc. You can use other place-holder: A..A : the number will be uppercase letter, instead of digit, such as: A, B, C B..B: The number will be lowercase letter. Such as: a,b,c R..R: The number will be uppercase roman letter. Such as: I, II, III M..M: The number will be lowercase roman letter. Such as: i, ii, iii
{filecount:O..O}	The count of resultant files. The place-holder is same as above.
{pagebegin:O...O}	The output file begin page number in original file. The place-holder is same as above.
{pageend:O...O}	The output file end page number in original file. The place-holder is same as above.
{pagecount:O..O}	The page count in the output file. The place-holder is same as above.
{pagerange}	The output file page range in original file. For example, 10-15
{totalpage}	The page count in the original file.

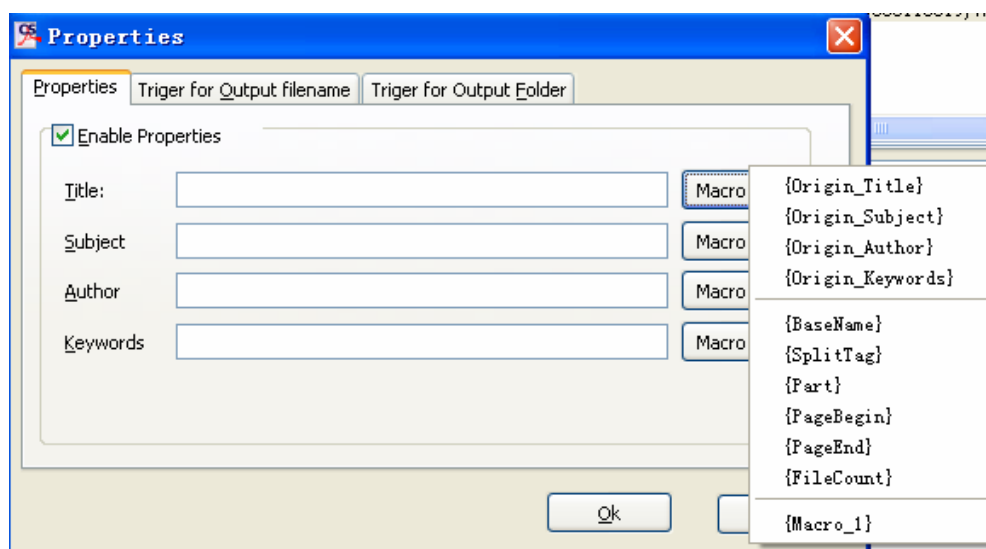
**Step 8**, A range of pages can be selected – either the entire contents of your original



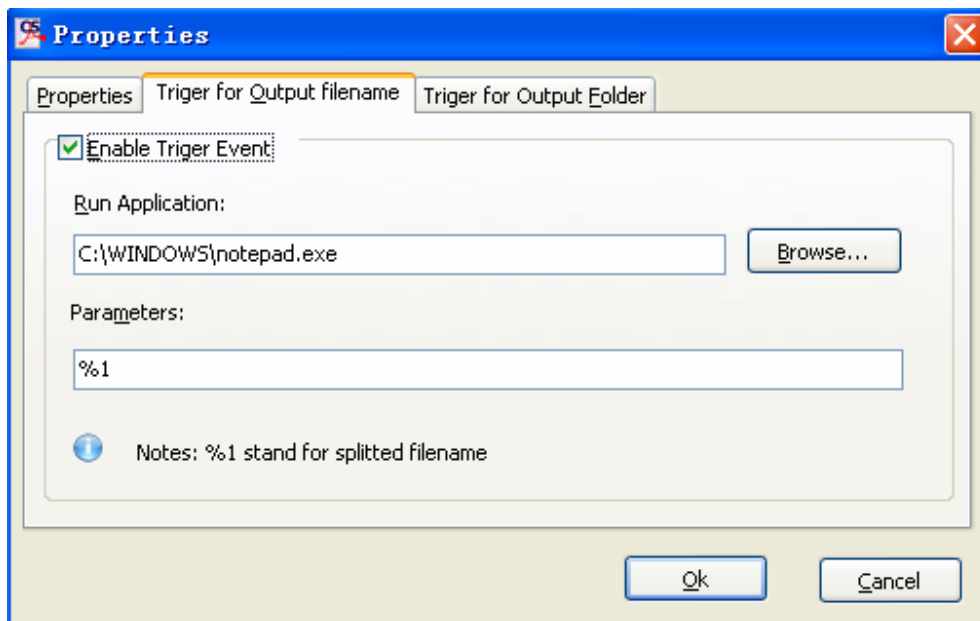
file, or only part of the file. This is a particularly useful option when you really only want a smaller set of split files produced from part of your original.

**Step 9(Optional)**, A-PDF Content Splitter allows you to add Meta data to output split PDF. Click **“Set properties”** button, then the Properties configuration window will be opened. The Properties configuration window allows you to add/edit Title, Subject, Author, and Keywords for the output spitted PDF document. Clicking the ‘Macro’ button displays a list of available macros for each field and allows you to quickly select data for each field.

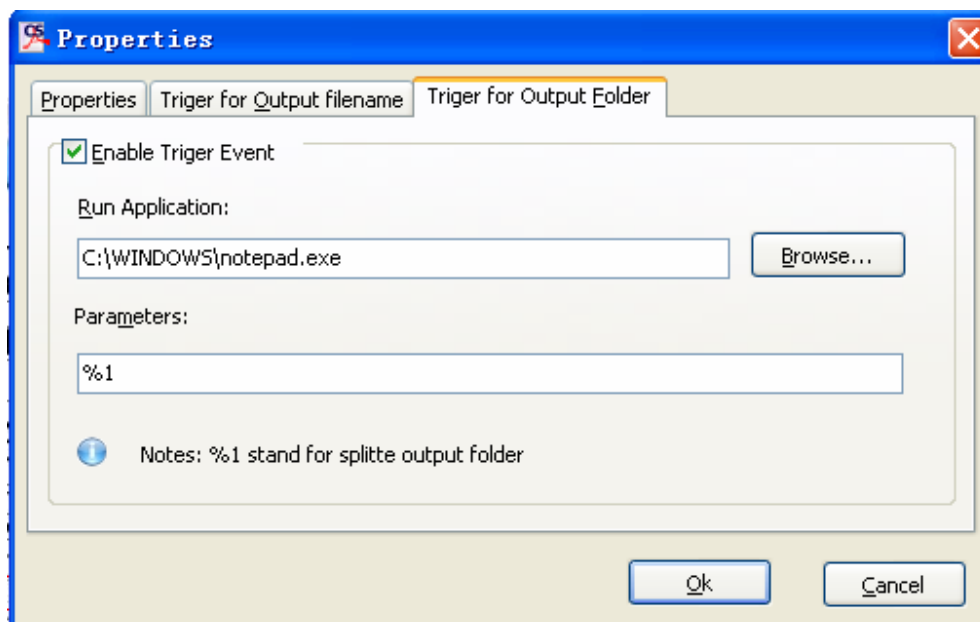
Set Properties



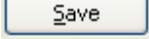
A-PDF Content Splitter Allows you to call external executable programs after split the each output file and set output filename %1 as a macro and pass it to the executable program as input parameter




A-PDF Content Splitter also allows you to call external executable programs after split the each source file and set output folder %1 as a macro and pass it to the executable program as input parameter

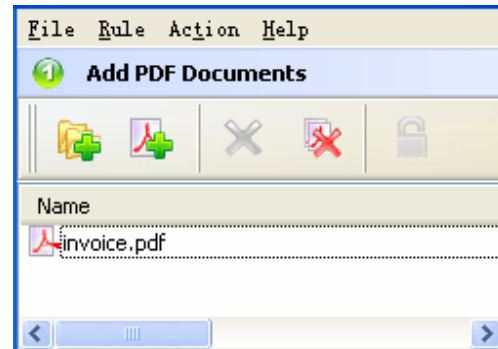




**Step 10**, Next click save button  to save the 'Invoice Number' Rule.


**Step 11**, Return to the main window. In the PDF File List window, click the **Add PDF**













**Document icon**  to add PDF file which you what to split. (For example: "invoice.pdf")



**Step 12**, In the **Split Rule List window**, select a rule (for example: Invoice Number)



**Step 13**. Click the **Split All** button  to split, the output file will split on "Invoice Number" and the result as blew (splitting base on invoice number)

 invoice\_9000115519\_(1\_13).pdf  
 invoice\_9000115520\_(2\_13).pdf  
 invoice\_9000115521\_(3\_13).pdf  
 invoice\_9000115522\_(4\_13).pdf  
 invoice\_9000115523\_(5\_13).pdf  
 invoice\_9000115524\_(7\_13).pdf  
 invoice\_9000115525\_(8\_13).pdf  
 invoice\_9000115526\_(9\_13).pdf  
 invoice\_9000115527\_(10\_13).pdf  
 invoice\_9000115528\_(11\_13).pdf  
 invoice\_9000115529\_(12\_13).pdf  
 invoice\_9000115530\_(13\_13).pdf

## Add Script for Output

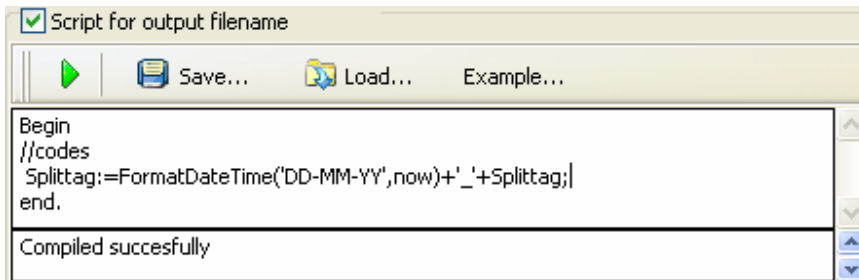
A-PDF Content Splitter allows you to custom how the split output files will be named by using Script. To enable script function, you need to check 'Script for Output Filename' option, then a script editor will be opened, which you can add script to



change 7 variables (BaseName, Part, PageBegin, PageEnd, PageRange, FileCount, SplitTag).

The script is based on Object Pascal, more about Object Pascal syntax please visit:

[http://en.wikipedia.org/wiki/Object\\_Pascal](http://en.wikipedia.org/wiki/Object_Pascal)



A script for variable 'Basename' example here

// example #1 - Trim Basename

// Script code:

begin

  Basename:=trim(Basename);

end;

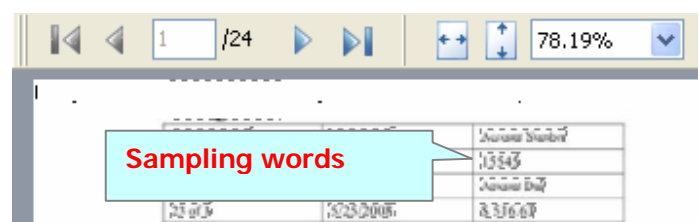
// example #1 end;

More examples please see **CS-Script-example.txt**

## Define Marcotag Marco

A-PDF Content Splitter allows you to custom how the split output files will be named by using Script. If you use splitag for the purpose of splitting only, and at the same time you need to use the text information in a specific position relate to split PDF page to name the output file, you can use Marcotag Marco

**Step 1**, In the opened **sample PDF view**. You can find the Account Number words as marcotag (for example the sample words is: 5545) in it. Click left mouse button to select it.



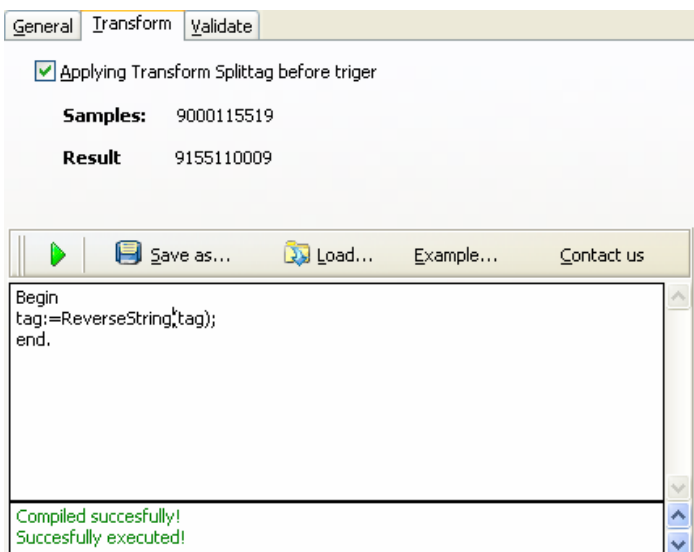
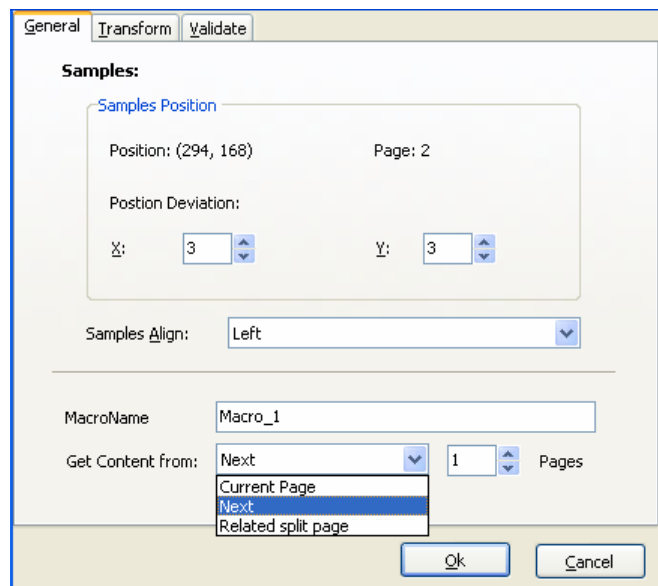
**Step 2**, Click the **Add to Marco Tag List ->** button to add the **Position** of selected sample words to Marco Tag list. Then open the **Add Samples to Marco Tag List** configuration window. You can adjust the **Position deviation (X,Y)** to mitigate the impact of position deviation in sample PDF.



If the text align of sampling is left.  
The **Samples Align** value should be "**Left**"

If the text align of sampling is middle.  
The **Samples Align** value should be "**middle**" and if the text align of sampling is right, the **Samples Align** value should be "**right**".

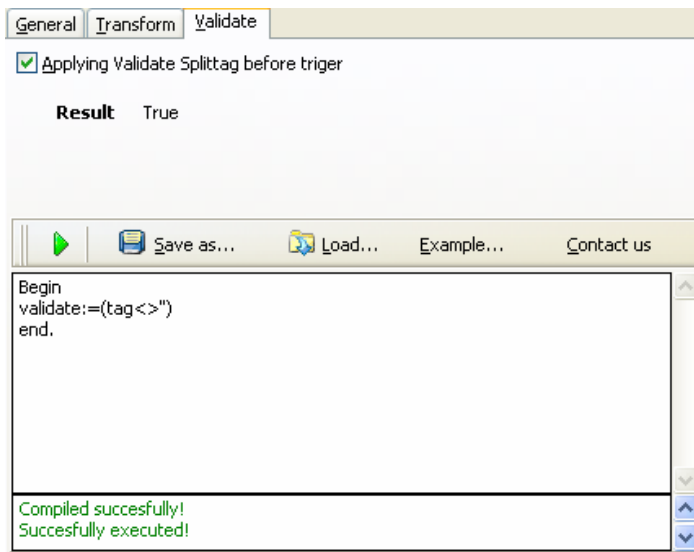
You can Set read Marco tag from the Specify scope of the page  
: Current page, Next N page (N stand for number), all range related split page



#### Transform Marcotag (Optional):

You can transform marcotag by script before you use the marcotag, The benefits of transform is: you can split out the your expect results from the more complex PDF, such as I would like to just take the first three characters from marcotag and use it to separate source PDF

Example : `tag:=leftstr(tag,3);`  
//take the first three characters from **marcotag**

**Validate marcotag (Optional):**

You also can verify marcotag by script before you use the marcotag, the result is only valid can be used as input split tag conditions

The benefits of Validate is: You can filter the marcotag line with the conditions, such as I would like to verify first three characters of marcotag is the '010'

Example :

```
validate:=(LeftStr(tag,3)='010');//  
If the Marcotag of the first three characters is the '010', then the
```

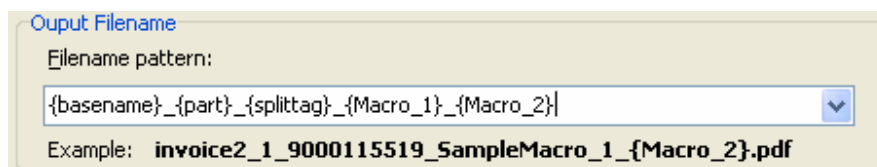
return value is true, otherwise, return false

**Step 3,** Next Click the **Ok** button to add the Samples to Marco tag list

Marco Name	Position	Marco Tag in position
Macro_1	(294,168)	
Macro_2	(294,169)	G1-2

**Step 4**

Now, the name patterns can include macro tag (such macro\_1, macro\_2, etc...) that automatically increment as necessary for the Final split files.

**Ending Your Session**

When you have completed your work, choose the **File** option and select **Exit** from the dropdown list.

More information please visit: <http://www.a-pdf.com>





## A-PDF Content Splitter Command Line Help

A-PDF Content Splitter Command line (PCSCMD) is a Windows console utility that extracts pieces of PDF files based on the content position user defined. PCSCMD is a standalone program. It does not need Adobe Acrobat. You can call the command line at your script or batch file.

A-PDF Content Splitter Command Line is included in the package installed.

### USAGE

```
PCSCMD <input file> [-S<password>] <rule file> [-O<output dir>]
```

**Parameters:**

<Input file> : The PDF file to be split. Such as "C:\invoice.pdf"  
<Output dir> : The output directory. If it is blank, will be the "output" under current directory.  
<rule file> : Content Splitter Split Rule file name, PCSCMD will split <input File> base on the rule, the rule can be defined by A-PDF Content Splitter GUI  
<password> : A password to open <input file> (if the pdf have open Password Security). If the <input file> no Password Security, it is not need.

**Examples:**

```
PCSCMD.exe "C:\invoice.pdf" "InvoiceNumber.rul"
```

```
PCSCMD.exe "C:\invoice.pdf" -Spwd "IN.rul" -o"C:\split" >c:\log.txt
```

**Return Code:**

0: Split successful  
1: Show help message only. Parameters error.  
2: Input file does not exist.  
3: Rule file does not exist.  
4: Load input file (PDF file) error.  
5: Split files error.  
6: Create output directory failed.  
99: Unknown exception error.



## Available Function in A-PDF Content Splitter Script

```
function UpperCase(const S: string): string;
function LowerCase(const S: string): string;
function CompareStr(const S1, S2: string): Integer;
function CompareText(const S1, S2: string): Integer;
function SameText(const S1, S2: string): Boolean;
function Trim(const S: string): string;
function TrimLeft(const S: string): string;
function TrimRight(const S: string): string;
function QuotedStr(const S: string): string;
function IntToStr(Value: Integer): string;
function StrToInt(const S: string): Integer;
function StrToIntDef(const S: string; Default: Integer): Integer;
function StrToBool(const S: string): Boolean;
function StrToBoolDef(const S: string; const Default: Boolean): Boolean;
function BoolToStr(B: Boolean; UseBoolStrs: Boolean = False): string;
function ExtractFilePath(const FileName: string): string;
function ExtractFileDir(const FileName: string): string;
function ExpandFileName(const FileName: string): string;
function StrLen(const Str: PChar): Cardinal;
function StrEnd(const Str: PChar): PChar;
function StrMove(Dest: PChar; const Source: PChar; Count: Cardinal): PChar;
function StrCopy(Dest: PChar; const Source: PChar): PChar;
function Format(const Format: string; const Args: array of const): string;
function StrToFloat(const S: string): Extended;
function StrToFloatDef(const S: string; const Default: Extended): Extended;
function EncodeDate(Year, Month, Day: Word): TDateTime;
function EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime;
function DayOfWeek(const DateTime: TDateTime): Word;
function Date: TDateTime;
function Time: TDateTime;
function Now: TDateTime;
function CurrentYear: Word;
function IncMonth(const DateTime: TDateTime; NumberOfMonths: Integer = 1): TDateTime;
procedure ReplaceTime(var DateTime: TDateTime; const NewTime: TDateTime);
procedure ReplaceDate(var DateTime: TDateTime; const NewDate: TDateTime);
function IsLeapYear(Year: Word): Boolean;
function DateToStr(const DateTime: TDateTime): string;
function TimeToStr(const DateTime: TDateTime): string;
function DateTimeToStr(const DateTime: TDateTime): string;
function StrToDate(const S: string): TDateTime;
function StrToDateDef(const S: string; const Default: TDateTime): TDateTime;
function StrToTime(const S: string): TDateTime;
function StrToTimeDef(const S: string; const Default: TDateTime): TDateTime;
function StrToDateTime(const S: string): TDateTime;
function StrToDateTimeDef(const S: string; const Default: TDateTime): TDateTime;
function IncludeTrailingPathDelimiter(const S: string): string;
function SameFileName(const S1, S2: string): Boolean;
function StringReplace(const S, OldPattern, NewPattern: string; Flags: ReplaceFlags): string;

function AnsiContainsText(const AText, ASubText: string): Boolean;
function AnsiStartsText(const ASubText, AText: string): Boolean;
function AnsiEndsText(const ASubText, AText: string): Boolean;
```



```
function AnsiReplaceText(const AText, AFromText, AToText: string): string;
function AnsiMatchText(const AText: string;const AValues: array of string): Boolean;
function AnsiIndexText(const AText: string; const AValues: array of string): Integer;
function AnsiContainsStr(const AText, ASubText: string): Boolean;
function AnsiStartsStr(const ASubText, AText: string): Boolean;
function AnsiEndsStr(const ASubText, AText: string): Boolean;
function AnsiReplaceStr(const AText, AFromText, AToText: string): string;
function AnsiMatchStr(const AText: string; const AValues: array of string): Boolean;
function AnsiIndexStr(const AText: string; const AValues: array of string): Integer;
function ReverseString(const AText: string): string;
function StuffString(const AText: string; AStart, ALength: Cardinal; const ASubText:string): string;
function LeftStr(const AText: String; const ACount: Integer):String;
function RightStr(const AText: String; const ACount: Integer): String;
function MidStr(const AText: String; const AStart, ACount: Integer): String;
function PosEx(const SubStr, S: string; Offset: Integer): Integer;
```

More information please visit: <http://www.a-pdf.com>