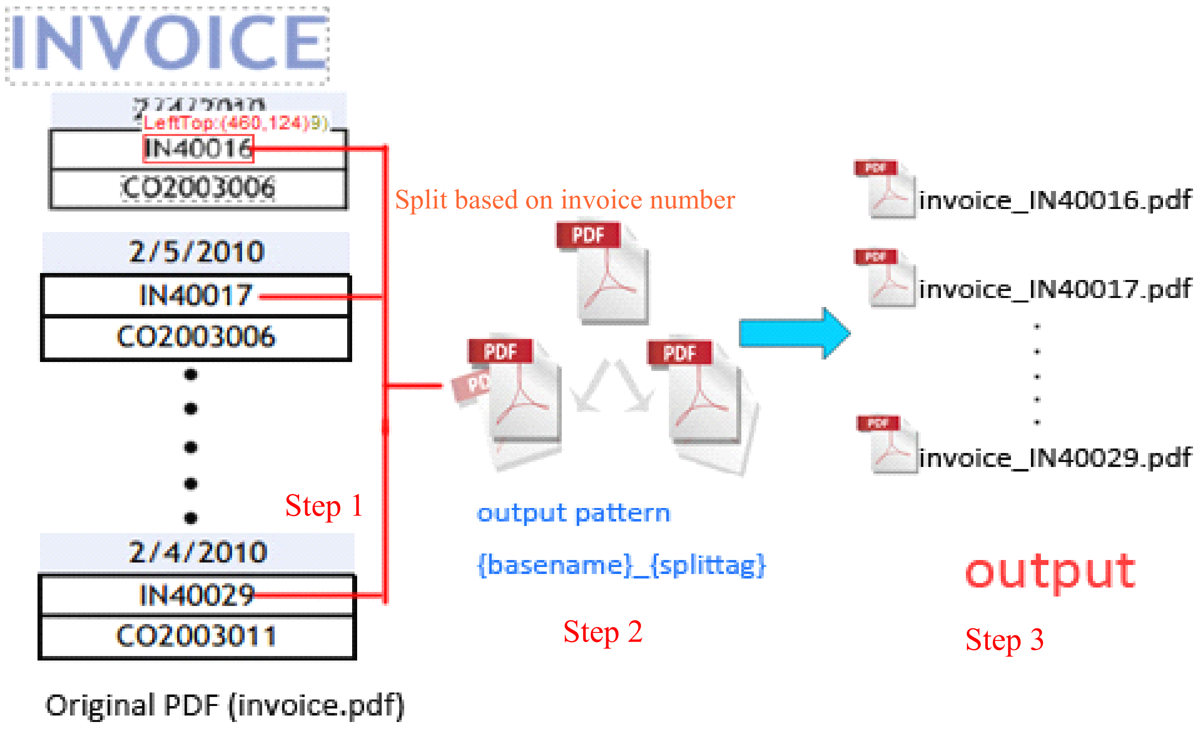
1 About Easy PDF Content Split

Easy PDF Content Split is a practically ideal standalone application (Acrobat NOT required) to split large PDF files that consist of numbers of individual records up to smaller segments, based on location and text within the files. It provides multiple options to separate documents such as invoice, report and payroll through keywords like invoice number, account number and employee name. More functions like hot directories,customized split rules and command line are included.

How does Easy PDF Content Split work?

Easy 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.

Functional diagram:



Step 1: Input the right PDF document that consists of multiple separated parts, such as invoice, payroll, report and so on.

Step 2: Edit splitting rule based on content within PDF with visual rule editor, and define output filename Pattern with macro parameter.

Step 3: Accomplished and you will get a series of split documents.

2 Functions and Features

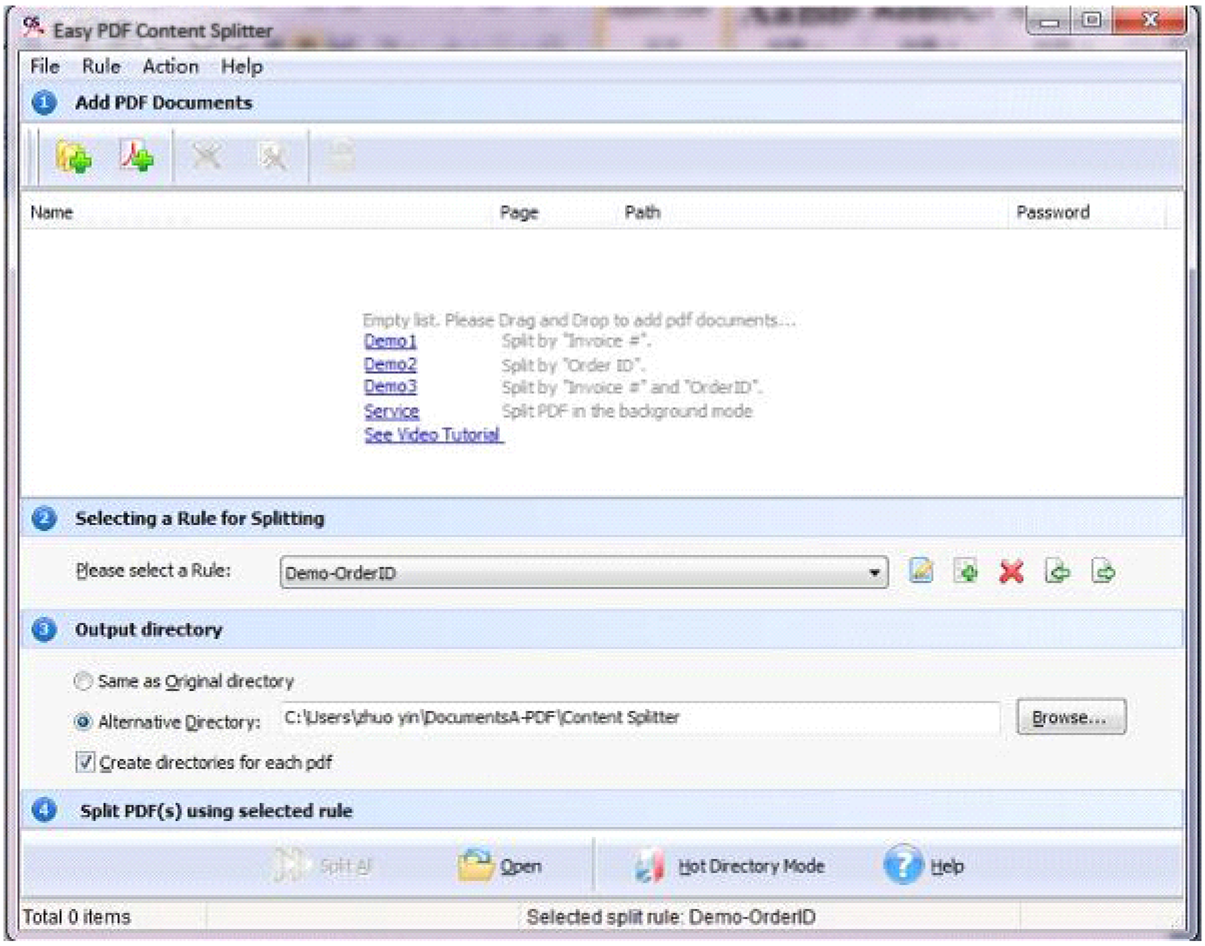
- Standalone utility program requires no other programs (e.g. Adobe Acrobat Pro).
- Intuitive interface and simple operation.
- Splits PDF file based on content, such as location and text within PDF file.
- Allows to split either all pages or only part of pages selected.
- Batch split amounts of PDF files based on content at amazing speed.
- Enables to work with hot directory or command line.
- Allows to edit rules for split and output filename with macro parameter.
- Supports use pascal-script to transform and validate tags, either split tag or macro tag.
- Auto replace chars like \ / *: with pre-set char when extracting text from PDF file as filename.
- To customize output filename pattern With macro parameters.
- Much more practical options like properties and security are available.
- Software compatible with Windows 98/2000/ME/XP/2003/Vista/7.

⚠ Easy PDF Content Split does NOT require Adobe Acrobat, and produces documents compatible with Adobe Acrobat Reader Version 5 and above.

3 Using Easy PDF Content Split

Under Windows Operation System, Easy PDF Content Split can be started either via the desktop shortcut, or directly from Windows explorer.

When you start the program, if you haven't registered our product, you will be informed to purchase the full version. Then you will be presented with the primary screen where you can complete split operation.



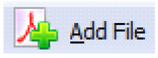
Empty list. Please Drag and Drop to add pdf documents...
Demo1 Split by "Invoice #".
Demo2 Split by "Order ID".
Demo3 Split by "Invoice #" and "OrderID".
Service Split PDF in the background mode
See Video Tutorial

Please select a Rule: Demo-OrderID

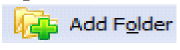
Same as Original directory
Alternative Directory: C:\Users\zhuo yin\DocumentsA-PDF\Content Splitter
[✓] Create directories for each pdf

Total 0 items
Selected split rule: Demo-OrderID

Add PDF document

Click the  icon in toolbar will open the standard file browse and select window, where you can browse the folders and find the PDF file you want to mail. Alternatively you can open the "**File**" menu and select "Add PDF Document" to add PDF file.

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 to add PDF file.

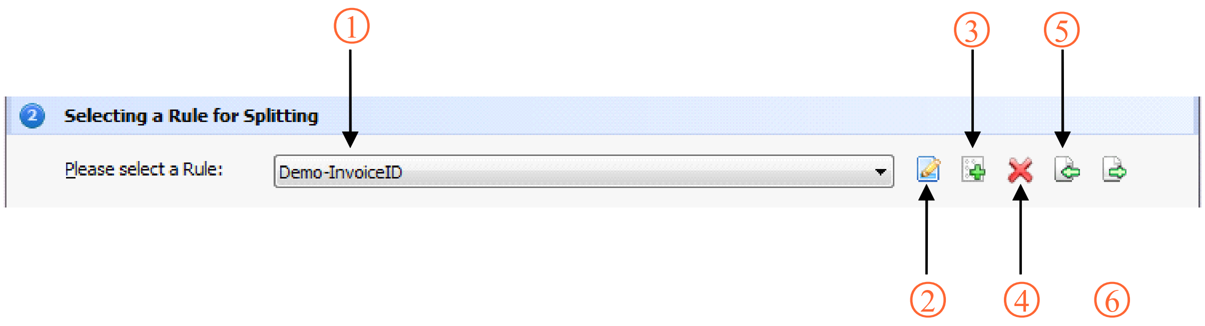
Easy PDF Content Split even allows you to select all files in a particular directory/folder by either using the **Add a Directory** option from the File menu or click the  icon! Add a directory also supports sub-folders.

Edit Rule for Splitting

What is Split Rule?

A Split Rule defines text position information within the sample PDF. Easy PDF Content Split 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.

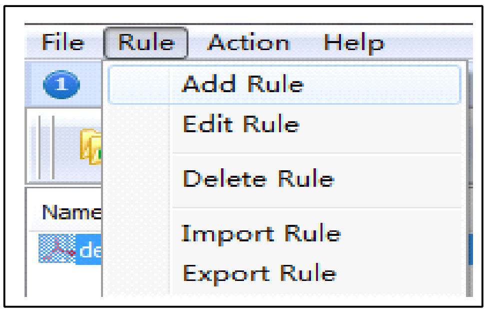
The Split Rule for applying to PDF(s) are listed in the Split Rule List window, where you can find rule items list in here, the current Rule you selected will to be applied to PDF(s).



① Current Rule List	② Edit selected rule
③ Add new rule	④ Delete existing rule
⑤ Import existing rule	⑥ Export rule for reusing

How to create split rule?

Software allows to either edit existing rule or create new rule for splitting. You can directly click the icon in rule toolbar to open the visual rule editor, or select "**Rule**" in menu bar then choose "**Add Rule**".



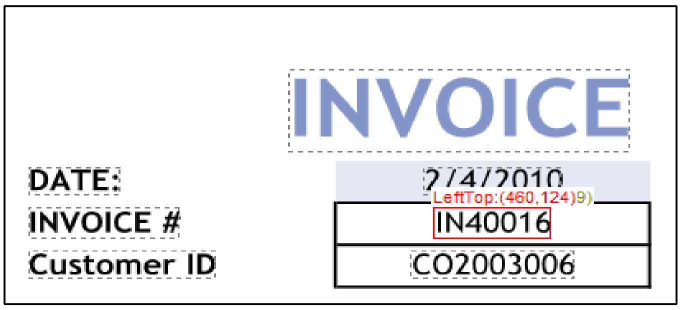
A simple example of how A-PDF Content Splitter is used will demonstrate how easy and convenient this utility is for splitting PDF files into parts.

Example for creating and applying an "Invoice Number" rule

Step 1: In the primary screen, click to open the rule editor window, where you can edit splitting rule with visual tag and Preview Area. Once you finish your editing, simply click Save button to save your work.

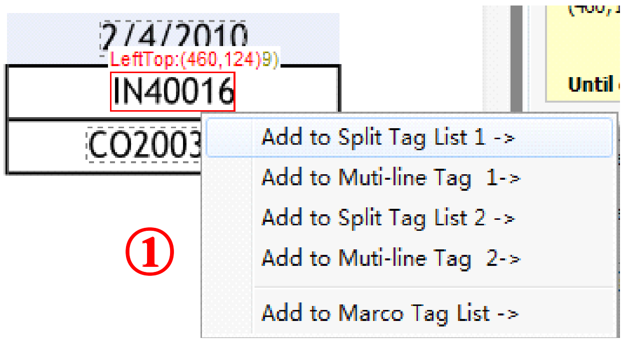
Step 2: Click the Browse... button in configuration window. You can select a PDF file (for example: "invoice.pdf") as sampling document. Normally the sampling document can be the PDF document that you want to split.

Step 3: In the opened **sample PDF view**. You can find the **Invoice Number** words (for example the sample words is: IN40016) in it. Click left mouse button to select it. Then you will find have a Red dashed border on it.



Step 4:

① Click the target words and select such as "Add to Split Tag List 1" in Pop-up windows

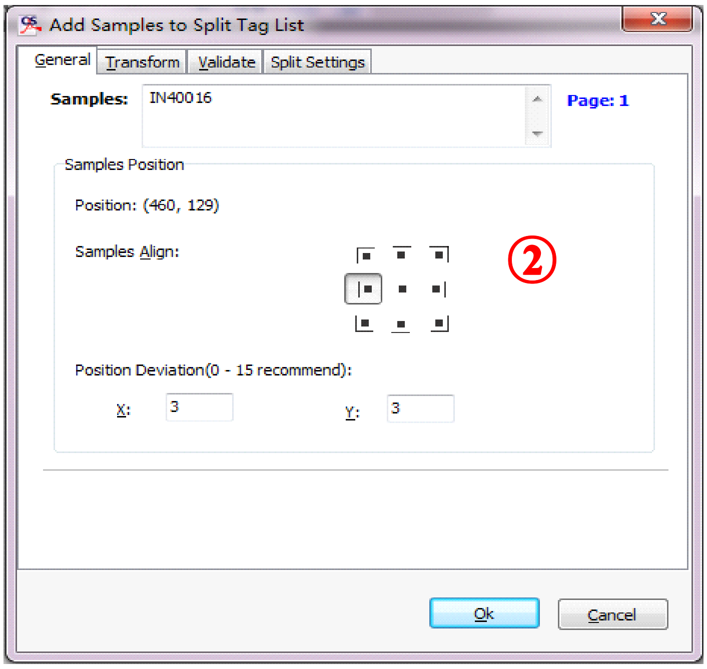


② Then in the opened "**Add Samples to Split Tag List**" configuration window, you can adjust the Position deviation (X,Y) to mitigate the impact of position deviation in sample PDF.

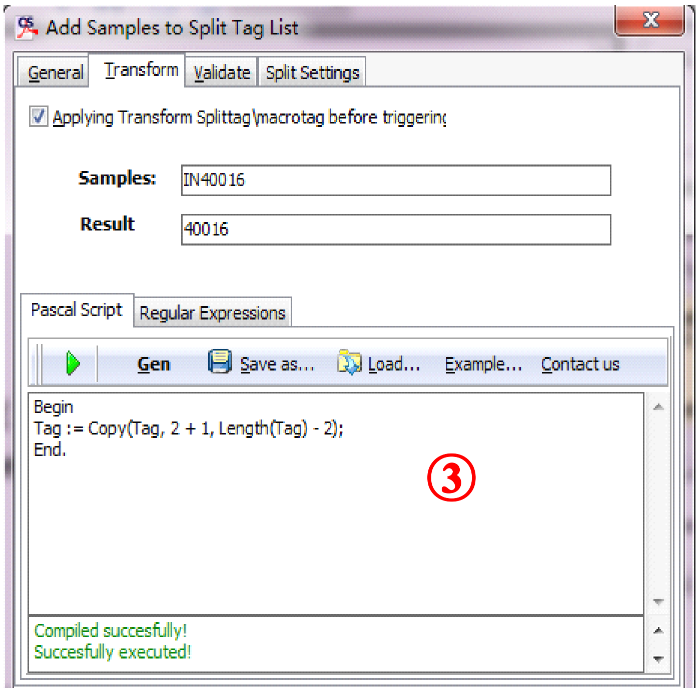
If the text align of sampling is left. The Samples Align value should be "**Left**".

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**".



Samples: IN40016
Page: 1
Position: (460, 129)
Position Deviation(0 - 15 recommend):
X: 3
Y: 3



☑ Applying Transform Splittag\macrotag before triggering

Samples: IN40016

Result: 40016

```
Begin
Tag := Copy(Tag, 2 + 1, Length(Tag) - 2);
End.
```

```
Compiled succesfully!
Succesfully executed!
```

③ **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 two characters from splittag and use it to separate source PDF.

Example :

tag:=leftstr(tag,2);

//take the first two characters from splittag

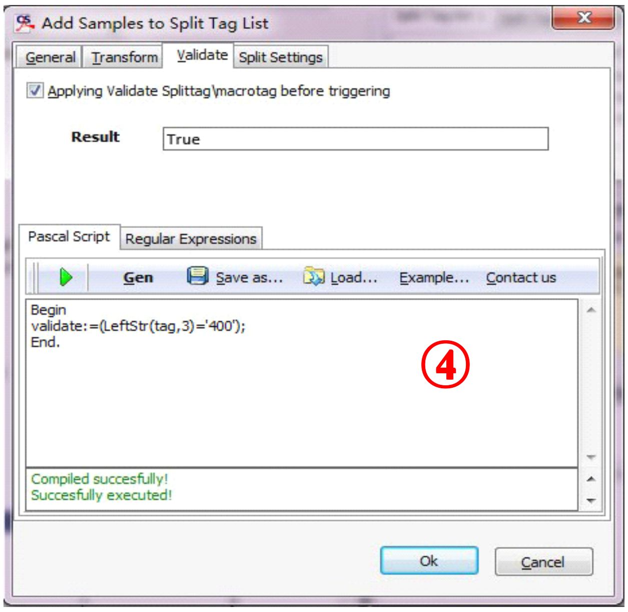
④ Validate Macro (Optional):

You also can verify Macro by script before you use the Macro, the result is only valid can be used as input split tag conditions. The benefits of Validate is: You can filter the Macro line with the conditions, such as I would like to verify first three characters of Macro is the '400'

Example :

validate:=(LeftStr(tag,3)='400');

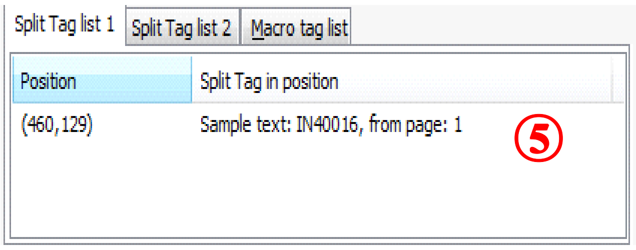
// If the Macro of the first three characters is the '400', then the return value is true, otherwise, return false.



```
Begin
validate:=(LeftStr(tag,3)='400');
End.
```

```
Compiled succesfully!
Succesfully executed!
```

⑤ 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 (460,124) and in each page) until "Invoice Number text" changed.



Position	Split Tag in position
(460,129)	Sample text: IN40016, from page: 1

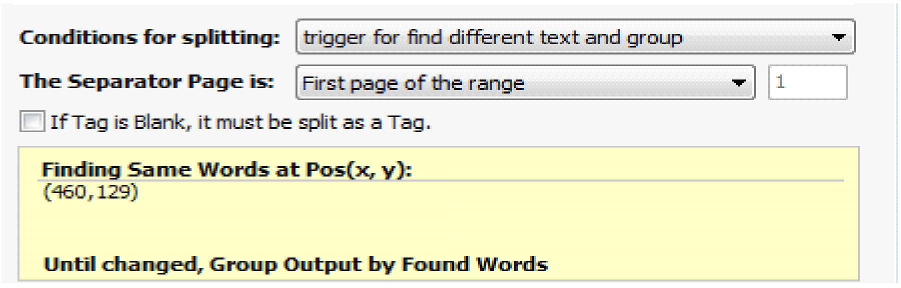
Step 5:

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.



Conditions for splitting: trigger for find different text and group

The Separator Page is: First page of the range 1

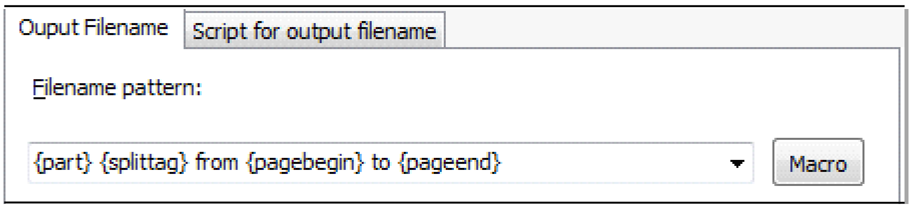
☐ If Tag is Blank, it must be split as a Tag.

Finding Same Words at Pos(x, y):
(460,129)

Until changed, Group Output by Found Words

Step 6: Entering a name for the Rule (note the text that will appear in the Split Rules list, must be a unique name, different from existing rule name). For example: Invoice Number.

Step 7: Easy PDF Content Split 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**".



Filename pattern: {part} {splittag} from {pagebegin} to {pageend}

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

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.

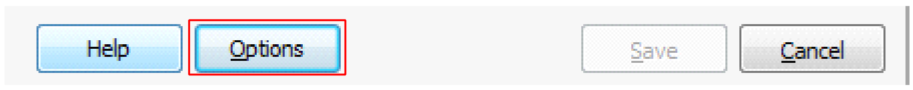
Patterns and their associated macros are a great convenience for naming your split files conveniently and logically with almost no effort on your part.

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**".

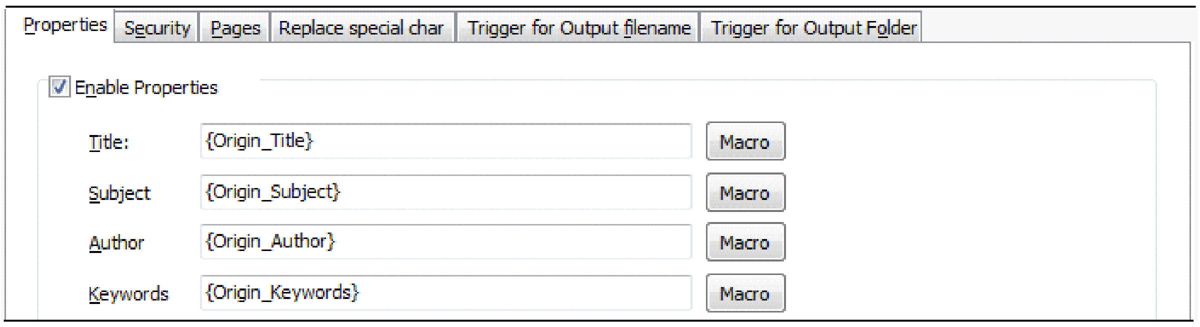
MACRO	DESCRIPTION
{basename}	The unsplit file name without extension name.
{splittag}	The text searched. For example, 900001234.
{part:0...0}	A sequential number from 1. 0...0 is place-holder. For example, you are using {part:0000}, will get 0001, 0002 etc. You can use other place-holder: A..A : the 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:0..0}	The count of resultant files. The place-holder is same as above.
{pagebegin:0...0}	The output file begin page number in original file. The place-holder is same as above.
{pageend:0...0}	The output file end page number in original file. The place-holder is same as above.
{pagecount:0..0}	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(optional): Easy PDF Content Split also provide varied practical options, such as properties, security and so on, and you can apply them according to per need.

Under Rule Editor window, click "**Options**" button to open the settings window.



Properties- Define the metadata of output PDF, including Title, Author, Subject and Keywords.



☑ Enable Properties

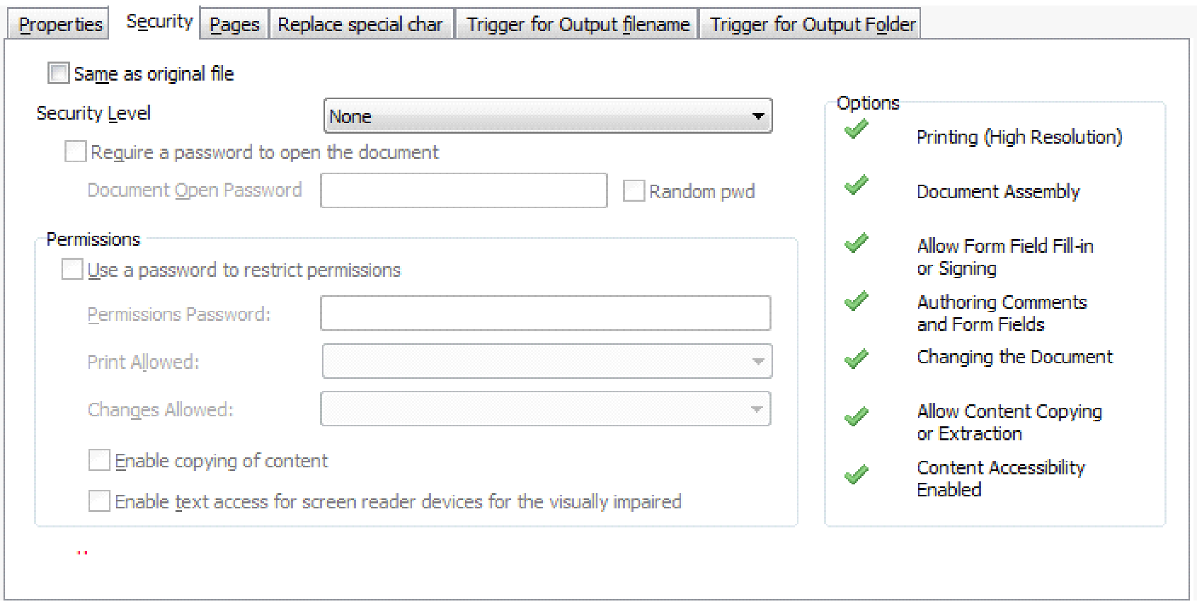
Title: {Origin_Title}

Subject {Origin_Subject}

Author {Origin_Author}

Keywords {Origin_Keywords}

Security- Set standard password to avoid unauthorized access and restrict permissions like copying, printing,etc.


Properties | Security | Pages | Replace special char | Trigger for Output filename | Trigger for Output Folder

☐ Same as original file

Security Level: None

☐ Require a password to open the document

Document Open Password: ☐ Random pwd

Permissions

☐ Use a password to restrict permissions

Permissions Password:

Print Allowed:

Changes Allowed:

☐ Enable copying of content

☐ Enable text access for screen reader devices for the visually impaired

Options

✓ Printing (High Resolution)

✓ Document Assembly

✓ Allow Form Field Fill-in or Signing

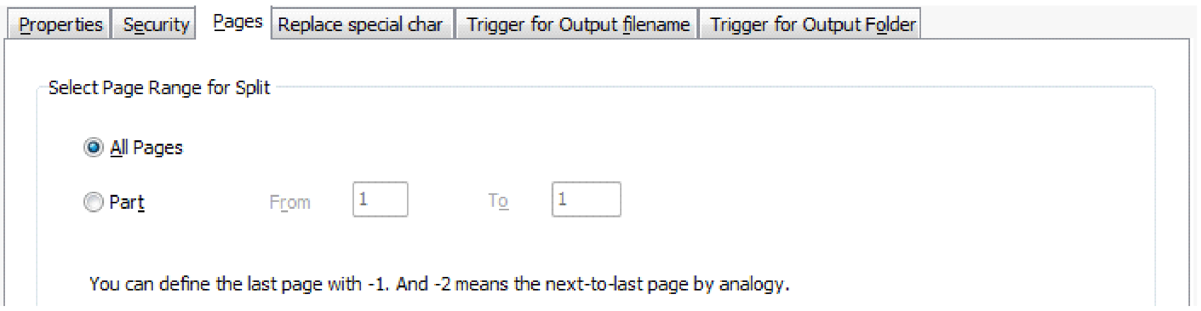
✓ Authoring Comments and Form Fields

✓ Changing the Document

✓ Allow Content Copying or Extraction

✓ Content Accessibility Enabled

Pages- Select page range for splitting


Properties | Security | Pages | Replace special char | Trigger for Output filename | Trigger for Output Folder

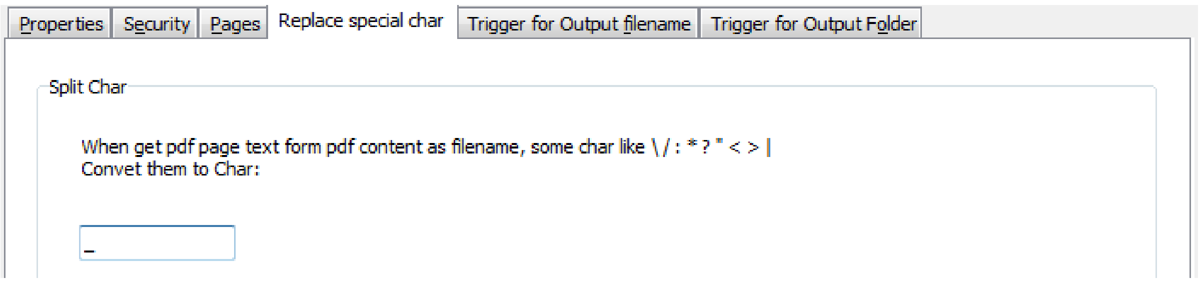
Select Page Range for Split

[✓ All Pages]

Part From 1 To 1

You can define the last page with -1. And -2 means the next-to-last page by analogy.

Replace special char

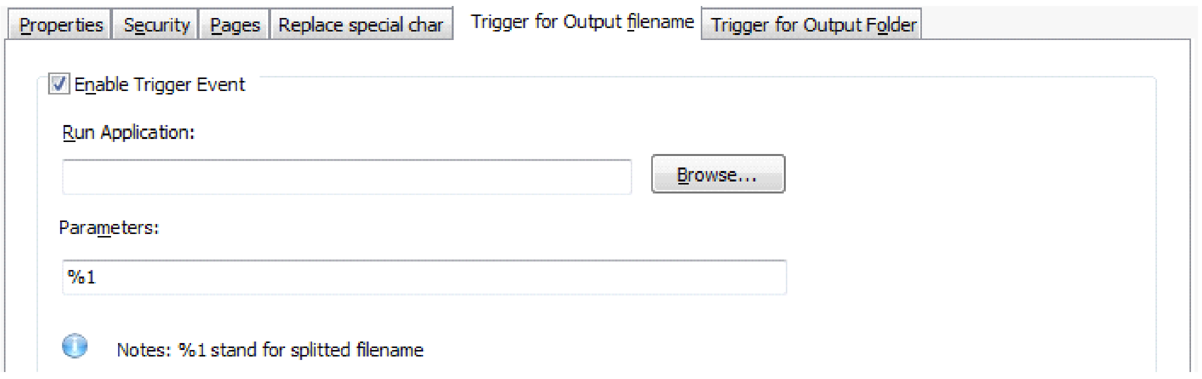

Properties | Security | Pages | Replace special char | Trigger for Output filename | Trigger for Output Folder

Split Char

When get pdf page text form pdf content as filename, some char like \ / : * ? " < > |
Convet them to Char:

_

Trigger for Output Filename



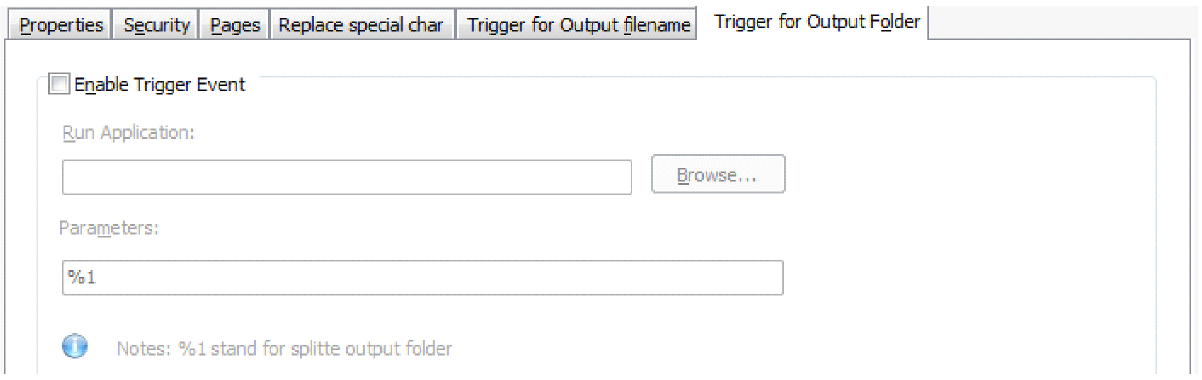
☑ Enable Trigger Event

Run Application:

Parameters: %1

Notes: %1 stand for splitted filename

Trigger for Output Folder



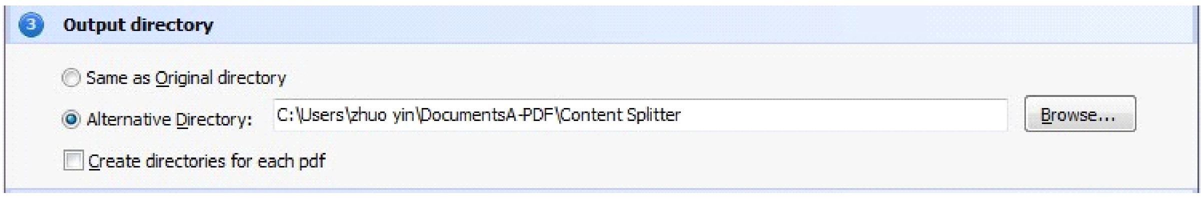
☐ Enable Trigger Event

Run Application:

Parameters: %1

Notes: %1 stand for splitte output folder

Step 9: Confirm the rule you have edited and click Save button to save your work. Return to the main window and appoint directory as output directory to save the output PDF documents.



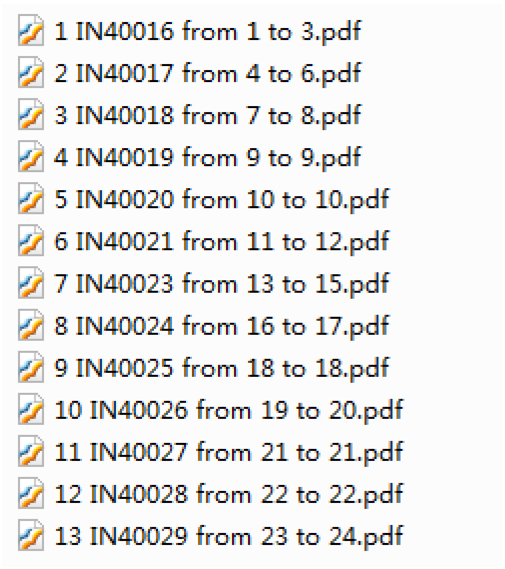
○ Same as Original directory

● Alternative Directory: C:\Users\zhuo yin\DocumentsA-PDF\Content Splitter

☐ Create directories for each pdf

Step 10: After you complete all the configuration, you can click

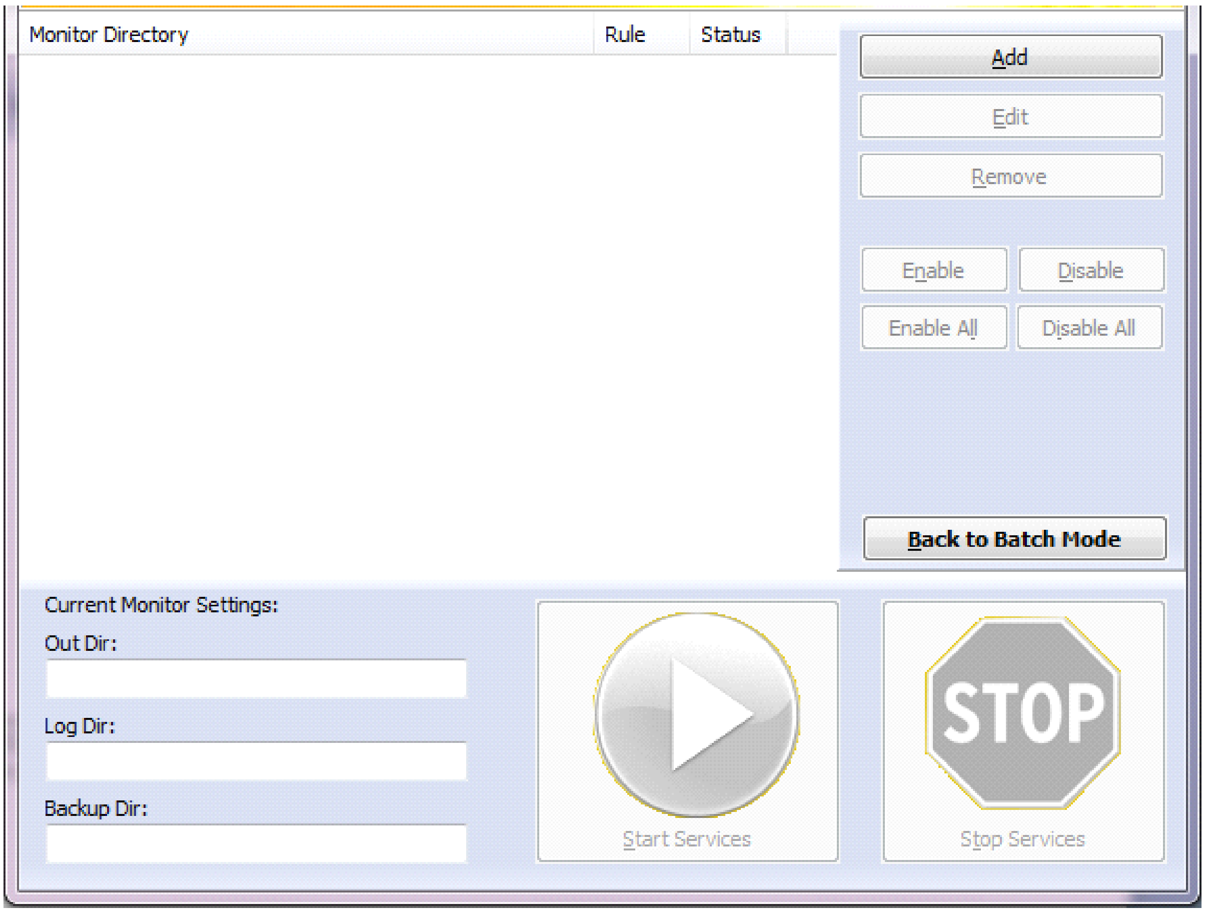
button in main window to start splitting. In a few seconds, you will get a series of separated PDF documents.



1 IN40016 from 1 to 3.pdf
2 IN40017 from 4 to 6.pdf
3 IN40018 from 7 to 8.pdf
4 IN40019 from 9 to 9.pdf
5 IN40020 from 10 to 10.pdf
6 IN40021 from 11 to 12.pdf
7 IN40023 from 13 to 15.pdf
8 IN40024 from 16 to 17.pdf
9 IN40025 from 18 to 18.pdf
10 IN40026 from 19 to 20.pdf
11 IN40027 from 21 to 21.pdf
12 IN40028 from 22 to 22.pdf
13 IN40029 from 23 to 24.pdf

4 Hot Directory Mode

Easy PDF Content Split also supports hot directory mode. Under hot directory mode, firstly you have to define folder as monitored directory, when some one writes the PDF files in, the files will be split with pre-set rule and sent to an output directory automatically. In Easy PDF Content Split main window, select the option Hot Directory Mode, a Hot Directories Service window will be opened.



After starting HOT Directory service, software will work silently, supervising the monitored directory and split automatically when new PDF files are written in. Unless you manually stop the service, Easy PDF Content Split will work continually.

5 Command Line Mode

Easy 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.

Easy 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.
```

6 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.easy-pdf-tools.com