

ARTS Split Pro Assistant

Version 1.4



1.0 Table of Contents

1.0	Table of Contents	2
2.0	Introduction.....	3
3.0	ARTS Split Pro Assistant Tools.....	4
3.1	Create a new coordinate file.....	4
3.2	Open a coordinate file	4
3.3	Save a coordinate file.....	5
3.4	Close a coordinate file.....	5
3.5	View text as ARTS Split Pro does	5
3.6	View text runs.....	6
3.7	Activate the rectangle tool	6
4.0	Coordinate files	7
4.1	Structure of commands	8
4.2	Rectangle commands.....	9
4.3	Parameters	10
5.0	Frequently asked questions (FAQs).....	11
6.0	Troubleshooting.....	12
7.0	Appendix A – Rectangle Commands	13
8.0	Appendix B – Split by Co-ordinate File Examples.....	17

2.0 Introduction

The ARTS Split Pro Assistant is a plug-in for Adobe Acrobat that assists the ARTS Split Pro user to create co-ordinate files of which are utilized when splitting PDFs with ARTS Split Pro.

The ARTS Split Pro Assistant can be accessed via the ARTS Split Pro Assistant toolbar, or from the Adobe Acrobat plug-ins menu.

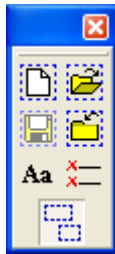


Figure 1.8 - ARTS Split Pro toolbar

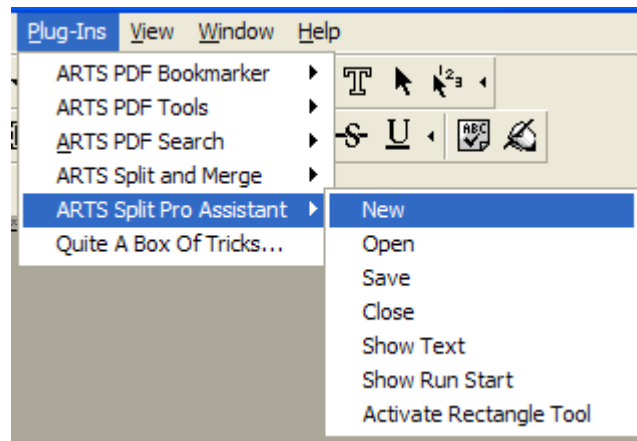




Figure 1.9 - ARTS Split Pro menu

3.0 ARTS Split Pro Assistant Tools

This section explains the tool buttons and menu items made available by the ARTS Split Pro Assistant plug-in.

3.1 Create a new coordinate file

To create a coordinate file:

1. Click the 'Create a New CRD File' button () on the ARTS Split Pro Assistant toolbar. You can alternatively go to 'Plug-ins > ARTS Split Pro Assistant > New' in the Adobe Acrobat plug-ins menu.
2. The Rectangle Tool () is now automatically activated/selected enabling you to draw a rectangle on your PDF page. Drag a box over the area on the page that contains the text you wish to apply a command to.
3. The 'Rectangle Tool Properties' window will now appear. Select the command you wish to use for the rectangle you just created on your page. A short description for each command is can be found below the rectangle command drop-down list.

Refer to [Rectangle Commands](#).

4. Enter any parameter that is required for your command in the 'parameters for command' text box. This is where you enter the text when you wish to see if a particular/literal string is present on the page.


Refer to [Parameters](#).

5. Click "OK". This command is now set to the rectangle currently selected.


Note: for each rectangle only one rectangle command can be set. However, there is no limit to the number of rectangles that can exist within a coordinate file. Therefore if you wish to apply several commands to a particular piece of text then it is possible by creating another rectangle over the top of the existing one.

3.2 Open a coordinate file

To open an existing coordinate file:

1. Select 'Open' from the ARTS Split Pro menu or click on the open button () located on the ARTS Split Pro toolbar.
2. Select the coordinate file to open.
3. Click "Open".

3.6 View text runs

To view text runs, select 'Show Run Start' from the ARTS Split Pro menu, or click on the close button () located on the ARTS Split Pro toolbar.

ARTS Split Pro Assistant allows the user to see where the start of each text run starts on the page. A small red cross represents the start of each text run. This can be seen in the diagram below (displaying text runs in ARTS Split Pro Assistant).

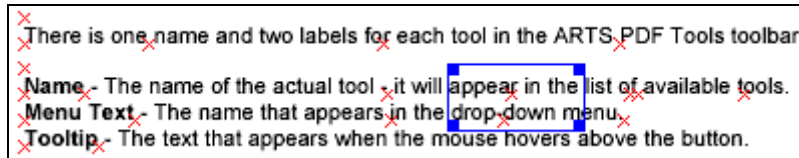


Figure 1.12 - displaying text runs in split pro assistant

Within a PDF file, text is drawn by moving to a specific location on the page and drawing a line of text, which is formally known as a run of text. What appears to be one line of text can actually be made of a number of runs of text put together.


In ARTS Split Pro, if a line of text begins inside a rectangle that you have created on your screen then the entire run of text is considered inside that rectangle. The rectangle that has been drawn in figure 1.12 contains two runs of text inside it, and as mentioned earlier the red crosses denote these runs of text.

The text that ARTS Split Pro would recognize from this triangle and the two runs of text inside it are:

- R in the list o
- Down menu

Note: if a line of text begins inside a rectangle, the entire run of text is considered "inside" that rectangle. The end of the rectangle in the figure 1.12 does not denote where ARTS Split Pro will stop reading the text, rather the end of the text runs that begin within the rectangle will.

3.7 Activate the rectangle tool

To activate the rectangle tool, select 'Activate Rectangle Tool' from the ARTS Split Pro menu, or click on the 'Activate the Rectangle Tool' button () located in the ARTS Split Pro toolbar. Following these same steps can also deactivate the rectangle tool.

4.0 Coordinate files

A coordinate file is used by the ARTS Split Pro to split a pdf file based on the text on each page. The coordinate file uses the .crd extension, but it is a plain text file that can be edited with any text editor. Each line of the coordinate file starts with a command and is followed by the appropriate operands for the command.

```
splitontextchange 507.406555 56.816055 535.184265 24.408707  
addtofilename 507.406555 56.816055 535.184265 24.408707 "pageno "  
providefilename 504.628784 67.001236 543.517578 21.630936
```

An example of commands contained within a coordinate file (.crd)

Editing the coordinate file with a text editor isn't required, as the ARTS Split Pro Assistant graphical user interface has been designed to allow the user to create coordinate files quickly and easily without the need to directly create or edit coordinate (text) files.

4.1 Structure of commands

The structure for a complete rectangle command as it appears within the co-ordinate file is shown below:

Command *[space]* left co-ordinate *[space]* top co-ordinate *[space]* bottom co-ordinate *[space]* right co-ordinate *[space]* "string"

Example:

```
splitiftextcontainedinbox 70.093918 736.859451 141.122421 703.214371 "arts"
```

Where:

Command = splitiftextcontainedinbox

Left co-ordinate = 70.093918

Top co-ordinate = 736.859451

Bottom co-ordinate = 141.122421

Right co-ordinate = 703.214371

String parameter = arts

Co-ordinates (left, top, bottom and top) are used to view the rectangle using ARTS Split Pro Assistant. It is the rectangle that is used to look at text in a PDF page used for the splitting of PDFs. Text within a rectangle area can be compared against the specified string parameter passed.

Example:

```
splitiftextcontainedinbox 70.093918 736.859451 141.122421 703.214371 "arts"
```

The rectangle co-ordinates, 70.093918 736.859451 141.122421 703.214371, form a rectangle. The rectangle area is viewable on a PDF page using ARTS Split Pro Assistant. A rectangle can be viewed on each page of the PDF document. Depending on where the text run begins on the page, this rectangle may have text inside it. Using the splitIfTextContainedInBox command, if ARTS Split Pro finds that the "arts" string is found inside the rectangle area then the PDF will split at that page (i.e. PDF will split if the word "arts" is found within the text inside the rectangle).

4.2 Rectangle commands

ARTS Split Pro allows the user to split PDF files based on the text that appears on the pages throughout a PDF document. The rectangle tool is used to select which text on the page will determine at which point/s the file is split, and this is where rectangle commands fit in.

Each rectangle that is created on the pdf document using the rectangle tool in ARTS Split Pro Assistant has a user determined command/s related to it which will trigger when the pdf file is split, or also perform a number of other functions.

How to set command/s for a rectangle:

1. Double click on a rectangle created by the rectangle tool in ARTS Split Pro.
2. The properties window should now appear and the drop down list of rectangle commands will be located near the top. The default command is 'no command selected' which means there is no active command selected for this rectangle.

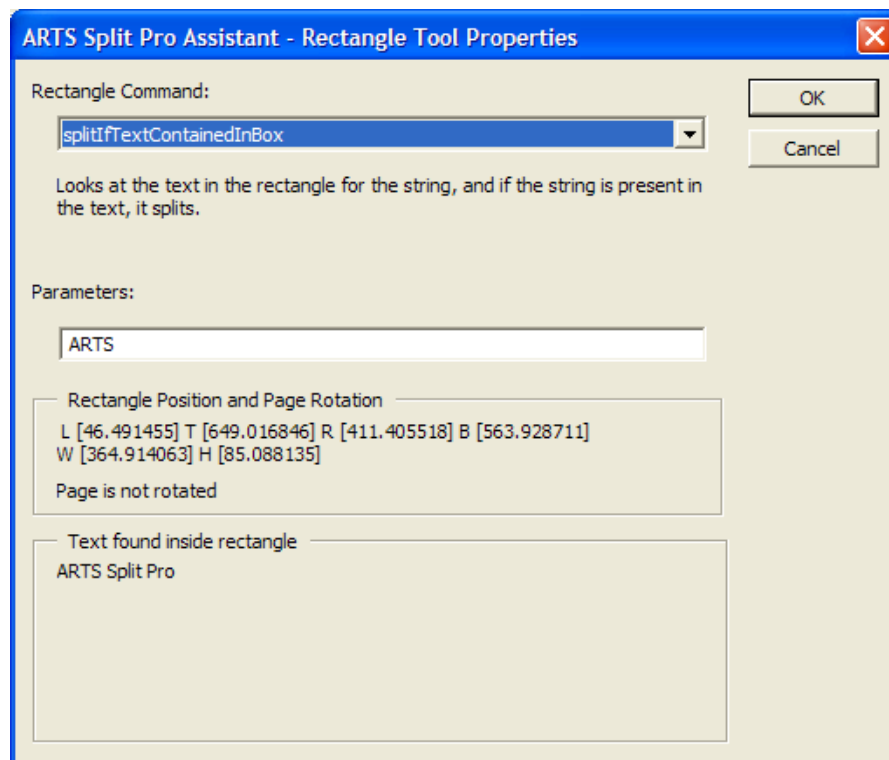


Figure 1.13 - rectangle properties window

3. Select the command from the rectangle command drop down list. A short description will appear below the drop down list when you have selected a command.

4.3 Parameters

When creating co-ordinate files, particular rectangle commands may require a user specified text string to be entered in order to split PDFs the way the particular command is expected to. There is a text box in the rectangle tool properties window named 'Parameters for command' where these text strings can be entered.

An example for a parameter is when utilizing the 'split if text contained in box' command when creating a co-ordinate file. This command checks if the parameter specified by the user within the parameter text box is present within the related ARTS Split Pro rectangle in the pdf file. If this string is present, then the file is split.

Other commands that require a user specified text string are:

- SplitIfTextIsPresent
- SplitIfTextIsInARun
- SplitOnTextChangeAfterString
- ProvideFilenameAfterString
- AddToFilename
- DeleteCharactersFromFilename
- GetFilenamesFromListInFile
- IncludeIfTextIsPresent
- IncludeIfTextContainedInBox
- IncludeIfTextChangeAfterString
- ExcludeIfTextIsPresent
- ExcludeIfTextContainedInBox
- ExcludeIfTextChangeAfterString
- ExtractFilename
- ExtractTextSelectedWords
- ExtractTextSkipWords
- ExtractTextSkipThisWord
- ExtractTextSkipCharacters
- ExtractOnlyIfThisTextIsInRectangle
- ExtractOnlyIfThisTextIsNotInRectangle

5.0 Frequently asked questions (FAQs)

Important note

Please check the ARTS Split Pro FAQs on our web site for recent additions at:

http://www.aroundtablesolution.com/faq_arts_split_pro.asp

Q. How do I stop my old ARTS Split Pro Assistant plug-in window from appearing each time acrobat starts up?

A. If you were using a past version of ARTS Split Pro or Ari's Split Pro, the assistant plug-in appeared each time Adobe Acrobat was opened. To stop this, go into the Acrobat plug-ins directory and delete the 'coordinator.api' file.

Q. There is no switch listed for using the 'includeInASingleFragmentIfTextIsPresent' command from the command line. How do I do this?

A. This command currently isn't supported in the command-line version of ARTS split pro. Contact info@aroundtablesolution.com for more information.

6.0 Troubleshooting

Before contacting us, please check the ARTS split pro conference at the ARTS forum on our web site at: <http://forum.aroundtablesolution.com>

If you have no luck there, please e-mail techsupport@aroundtablesolution.com and supply the information below to help us replicate the problem you are experiencing.

1. The exact version of ARTS Split Pro you are using (this is located by running split pro, and then clicking "About"). Please also specify whether you are using a demo or full registered version.
2. The exact version of the operating system you are using (this is located in 'Start > Settings > Control Panel > System')
3. The amount of free disk space remaining [on all hard disks] (found by double clicking on 'My Computer', and then right mouse clicking on the drive and selecting 'Properties').
4. Processor speed and amount of ram for the system on which ARTS split pro is running (e.g. Pentium 233 mhz, 32mb ram).
5. Any other programs that are running at the time of the error (e.g. Outlook, Internet Explorer, etc).
6. All error messages that were displayed when the error occurred.
7. The exact series of steps that led to the error.

Feedback

If you have ideas and suggestions on how we could improve ARTS Split Pro, we would love to hear your thoughts. Please send them to info@aroundtablesolution.com.

Legal notes

Acrobat and exchange are registered trademarks of adobe systems incorporated. ARTS Split Pro is copyright © A Round Table Solution (ARTS) 1998-2003.

7.0 Appendix A – Rectangle Commands

Contained in this appendix is a list of all the commands that can be used in conjunction with the rectangle tool found in the ARTS Split Pro Assistant.

SplitIfTextIsPresent

If the specified parameter is contained in any run of text on the page, the PDF will be split and the page will be the start of a new file. The comparison is case sensitive.

SplitIfTextContainedInBox

Text inside the rectangle on a page is searched to see if it contains the specified parameter. If the string parameter is found inside the rectangle on a page, then the PDF will be split and that page becomes the start of a new file.

SplitIfTextIsInARun

If the specified parameter is found in any text run then the PDF will be split. Text that spans multiple runs of text will not activate a split.

SplitOnTextChange

The text inside the rectangle on a page is searched, and if the text inside the rectangle on a page is different from the text inside the rectangle on the previous page, the PDF will be split and a new file is started.

SplitOnTextChangeAfterString

The text inside the rectangle on a page is searched for the specified parameter. If the text after the string parameter is different from (the text after the string of the) previous page, the PDF will be split and a new file is started.

SplitOnTextChangeAfterStringOnly

The text inside the rectangle on a page is searched for the specified string parameter. If the specified string is found, it will look if the text after the specified parameter changes from page to page, the PDF will be split and a new file started. Ignores pages where the string is not present.

SplitIfThisTextRepeats

The text inside the rectangle on a page is searched to see if it contains the specified parameter. If the string parameter is found inside the rectangle on two pages in a row, the PDF will split between the two pages so that they end up in different fragments.

SkipHeader

Normally, the `splitiftextcontainedinbox` command causes a new fragment to be started whenever the specified text is found within the box. The first fragment starts at page one and continues up to the page before the first page that contains the text in the box. The `Skipheader` command causes the first fragment to start with the first page that contains the text in the box, leaving out any pages that came before it.

ProvideFilename

Whatever text falls inside the rectangle (see runs of text) is used as the file name for the fragment. The text is appended to the output file name. If this or a similar command appears more than once in the coordinate file, the text from each rectangle is appended together in the same order that the commands appear in the coordinate file. This allows you to use the text from multiple locations on the page as part of the file name.

ProvideFilenameFromFirstWord

This is similar to the `providefilename` command, except that only the first word of the text inside the rectangle is used (see runs of text). Any initial spaces are skipped, and the word continues until a space or the end of the run of text.

ProvideFilenameFromSelectedWord

This is similar to the `providefilenamefromfirstword` command, except that only the number determines which word of the text inside the rectangle is used.

ProvideFilenameFromRangeOfCharacters

This looks at the text inside the rectangle and takes a range of characters from that text and adds the characters to the filename. The range of characters is specified by `Firstnum` and `Lastnum`.

ProvideFilenameAfterString

This is similar to the `providefilename` command, except that the software searches the text inside the rectangle for string, and if it finds the string, only the text inside the rectangle that comes after string is used for the file name. If string is not found, nothing is added to the file name.

AddToFilename

This command adds the string parameter to the output filename.

DeleteCharactersFromFilename

If any character in the text inside the quotes is found in the run of text for any of the `providefilename` commands, the character will be deleted as the text is added to the filename. The order of the command matters as it will only delete characters from the filename that are added by commands that come after this command in the coordinate file. This means that the `deletecharactersfromfilename` command must come before any of the `providefilename` commands that it will operate on.

GetFileNamesFromListInFile

This command opens a next file lists the names that will be given to each fragment created. The text file should contain one file name per line followed by a `cr/lf` pair.

IncludeAll

Normally, all pages go into one fragment or another, and the commands control where fragments end and another fragment starts. Passing a false parameter to the `includeall` command causes all pages to not be included in any fragment unless they are explicitly included by commands such as `includeiftextispresent` and `includeiftextcontainedinbox`.

IncludeIfTextIsPresent

All text on the page is searched for the string parameter. If the string parameter is found somewhere on the page, then the page will be included in the fragment. Since by default all pages go into one fragment or another, this command is only effective if a false parameter is passed to the `includeall` command.

IncludeInASingleFragmentIfTextIsPresent

If the literal string is found to occur in the text that is inside the rectangle, the page is added to the list of pages to be included in a single fragment. This is meaningless unless you set `includeall` to false.

IncludelfTextContainedInBox

All text within the rectangle on the page is searched for the string parameter. If the string parameter is found somewhere inside the rectangle, then the page will be included in the fragment. Since by default all pages go into one fragment or another, this command is only effective if a false parameter is passed to the includeall command.

IncludelfTextChangeAfterString

Looks at the text inside the rectangle for the string, if the string exists it will check if the text after the string changes. If the text after the string changes from the previous page that contained the string then the page will be included into the fragment. Since by default all pages go into one fragment or another, this command is only effective if a false parameter is passed to the includeall command.

ExcludelfTextIsPresent

If the string parameter is found somewhere in the text on the page, then this command causes the page to be excluded from all fragments.

ExcludelfTextContainedInBox

If the string parameter is found somewhere in the text on the page inside the rectangle, then this command causes the page to be excluded from all fragments.

ExcludelfTextChangeAfterString

Looks at text inside the rectangle for the string parameter, if the string exists it will check if text after the string changes. If the text after the string changes then the page will be excluded from the fragment.

ExtractFilename

This command opens a text extract file that will receive the text extracted by the various extract commands.

ExtractText

This command extracts the run or runs of text that begin within the rectangle into the extract file specified by the extractfilename command. See the runs of text section.

ExtractTextSelectedWords

This command works similarly to the extracttext command, but it extracts only the words indicated by the range number-number. It looks at the run of text that falls within the rectangle, and takes only the words that are number-number in order. For instance, if a run of text is text 15.839996,826.880005 (this is text on the page) and the extract command is extracttextselectedword 12 829 18 823 2-4, then only the text is text on will be extracted. This command uses the run or runs of text that begin within the rectangle into the extract file specified by the extractfilename command. See the runs of text section.

ExtractTextSkipWords

This command works similarly to the extracttext command, but it skips over number words. If the text in the run is text 15.839996,826.880005 (this is text on the page) and the extract command is extracttextskipwords 12 829 18 823 2, then the text text on the page will be extracted, because the first two words were skipped.

ExtractTextSkipThisWord

This command works similarly to the `extracttext` command, but it skips over any initial Whitespace and skips over the first occurrence of text in the run of text. If text is not present, the entire run of text inside the rectangle is extracted. If text is present, then the text that comes after text in the run of text inside the rectangle is extracted.

ExtractTextSkipCharacters

This command works similarly to the `extracttext` command, but it skips over any initial Whitespace and then skips over number characters. If the text in the run is text 15.839996,826.880005 (this is text on the page) and the `extract` command is `extracttextskipcharacters 12 829 18 823 2`, then the text is text on the page will be extracted, because the first two characters were skipped.

ExtractOnlyIfThisTextInRect

This command causes `extracttext` commands that follow it in the coordinate file to extract only if the text is found inside the rectangle on a page. The order of the commands in the coordinate file is important.

ExtractOnlyIfThisTextNotInRect

This command is similar to the `extractonlyifhistextinrect` command, except that it will cause following `extracttext` commands to extract only if the text is not in the rectangle.

ExtractTextLineAfterString

If the parameter specified by the user is found on a visual line of text in the pdf, the rest of the text on that line after the specified parameter is extracted.

ExtractTextLineAfterStringInRect

If the parameter specified by the user is found on a visual line of text using text runs within the rectangle, the rest of the text on that line is extracted.

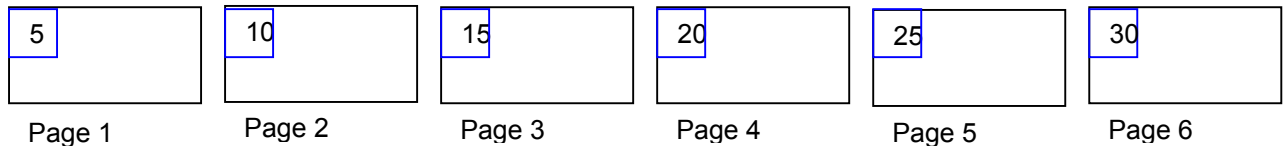
FillInfoDictEntry

Takes the text that is inside the rectangle and stores it within the specified key of the info dictionary.

8.0 Appendix B – Split by Co-ordinate File Examples

Example 1. Splitting source pdf when the number changes with the 'pageno' and the selected number as the output filename.

Content of source file:



Content of co-ordinate file:

```
splitontextchange 507.406555 56.816055 535.184265 24.408707  
addtofilename 507.406555 56.816055 535.184265 24.408707 "pageno "  
providefilename 504.628784 67.001236 543.517578 21.630936
```

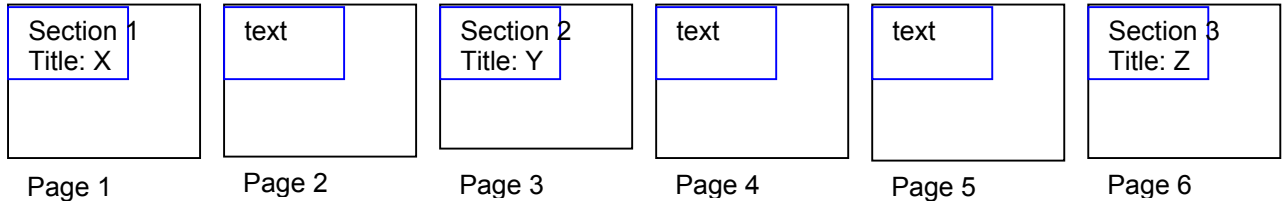
Where rectangle co-ordinates select the text run area where the number appears.

Output fragments using the co-ordinate file:

```
'pageno 5.pdf' containing page 1  
'pageno 10.pdf' containing page 2  
'pageno 15.pdf' containing page 3  
'pageno 20.pdf' containing page 4  
'pageno 25.pdf' containing page 5  
'pageno 30.pdf' containing page 6
```

Example 2. Splitting source pdf when the page number changes with the page number as the output filename.

Content of source file:



Content of co-ordinate file #1:

```
splitiftextcontainedinbox 507.406555 56.816055 535.184265 24.408707 "Section"  
providefilenameafterstring 504.628784 67.001236 543.517578 21.630936 "Title:"
```

Where

splitiftextcontainedinbox rectangle co-ordinates select the text run area where the 'Section' appears.

splitiftextcontainedinbox rectangle co-ordinates select the text run area where 'Title:' appears.

Content of co-ordinate file #2:

```
splitontextchangeafterstring 507.406555 56.816055 535.184265 24.408707 "Section"  
providefilenameafterstring 504.628784 67.001236 543.517578 21.630936 "Title:"
```

Where

splitontextchangeafterstring rectangle co-ordinates select the text run area where the 'Section' and number appears.

splitiftextcontainedinbox rectangle co-ordinates select the text run area where 'Title:' appears.

Output fragments using either co-ordinate file #1 or #2:

'x.pdf' containing pages 1,2

'y.pdf' containing pages 3-5

'z.pdf' containing page 6

Example 3. Extract text from a pdf into a text file.

Content of co-ordinate file:

```
extractfilename "c:\output\extract.txt"  
extracttext -2.000000 788.000000 591.000000 12.000000
```

Where

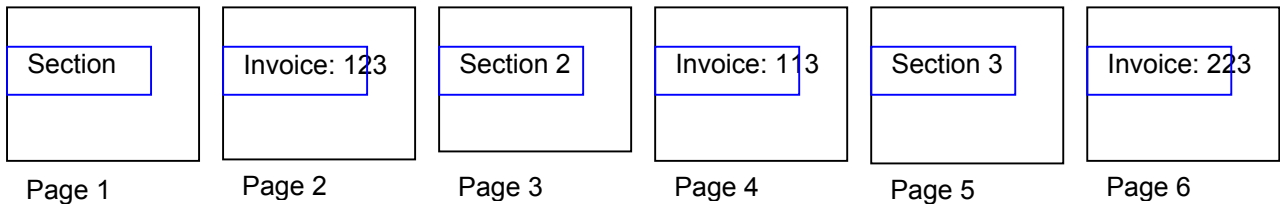
extracttext rectangle co-ordinates select the text run area selects all text of the whole page.

Output fragment using the co-ordinate file:

Extract.txt file within the c:\output\ folder containing text from the page.

Example 4. Excluding pages.

Content of PDF source file:



Content of co-ordinate file:

```
excludeiftextchangeafterstring 46.398743 644.541016 147.459015 629.922180 "Invoice:  
"
```

Where

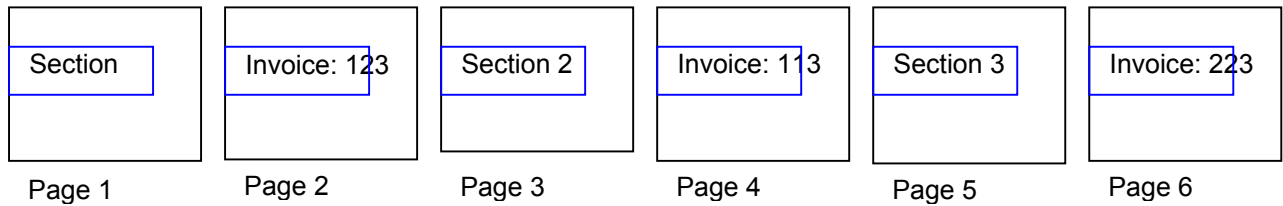
excludeiftextchangeafterstring rectangle co-ordinates select the text run area where the "Invoice:" appears.

Output fragment using co-ordinate file:

A single fragment containing pages 1, 3 and 6.

Example 5. Including pages.

Content of PDF source file:



Content of co-ordinate file:

```
includeall 347.037170 733.524902 507.208160 698.566895 "false"  
includeiftextcontainedinbox 347.037170 733.524902 507.208160 698.566895 "invoice"
```

Where

includeiftextcontainedinbox rectangle co-ordinates selects the text run area where the "invoice" appears.

Output fragments using co-ordinate file:

Fragment #1 containing page 2
Fragment #2 containing page 4
Fragment #3 containing page 6