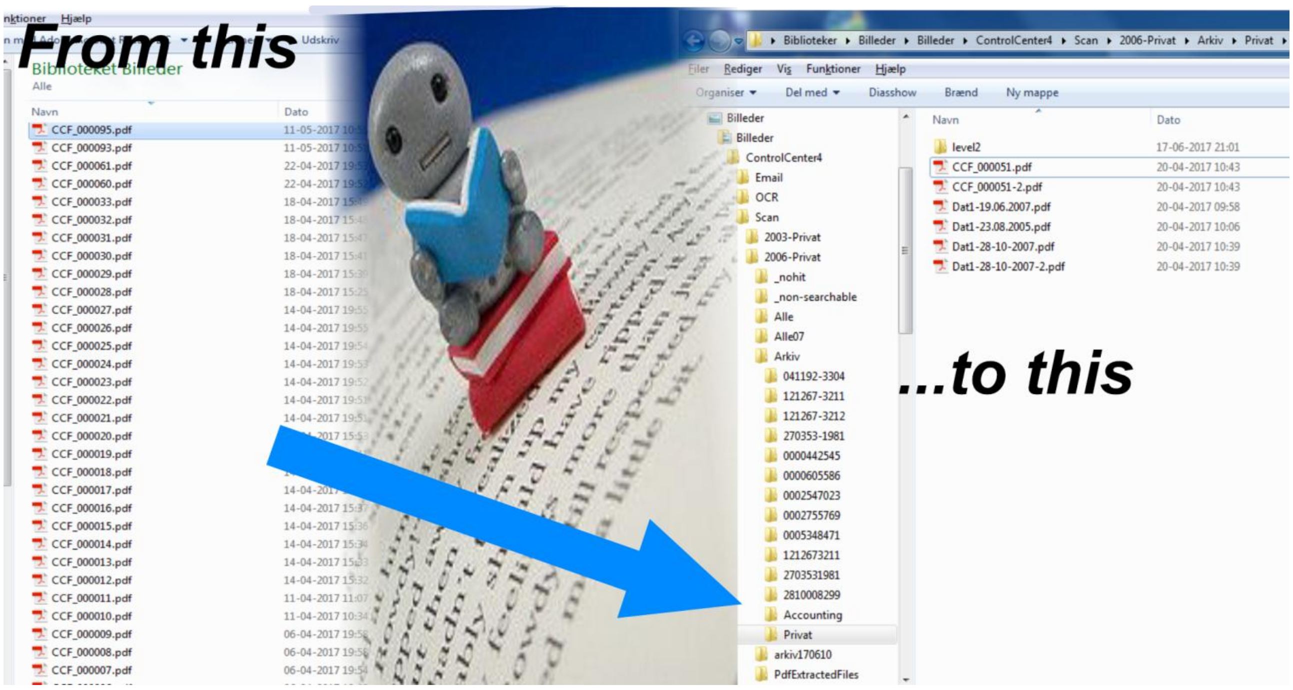
PDF Text Scanner

User Guide



User Guide ver. 1.0 en

InfoShare
Degnemose Allé 18
DK 2700 Brønshøj
Denmark

+45 38806010
pdftextscanner@infoshare.dk
www.infoshare.dk/pdf-text-scanner

PDF Text scanner

User Guide

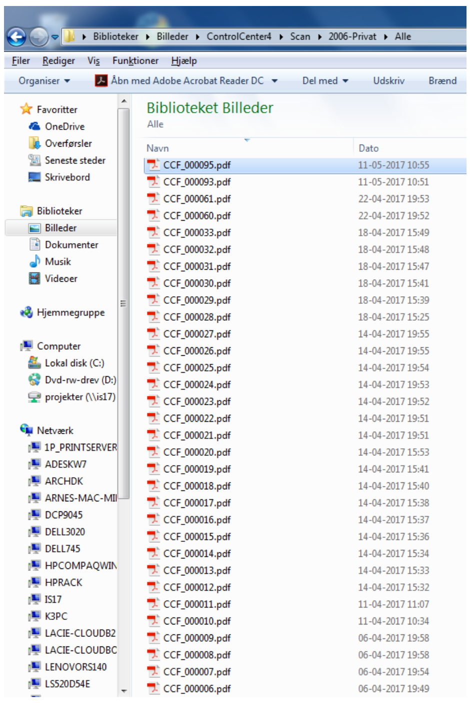
(Go to our homepage infoshare.dk/pdf-text-scanner-en to find this document in other versions or languages)

PdftextScanner is an application for automatic renaming and moving of searchable pdf files. When started the application is constantly watching the content of a specific folder (the source folder) for any pdf files and when there's new files they are being read and renamed and moved to another folder according to the rules set up in the rules configuration file.

If your source material is on paper you would normally send the paper through a scanner, then through an OCR program to make the scanned pdf files searchable and finally let PdfTextScanner move these files to relevant folders.

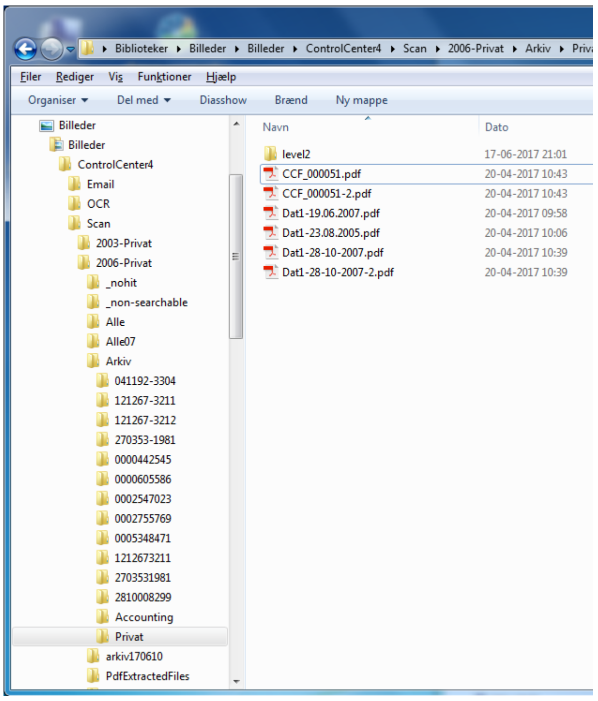
With most scanners and scanner software this can all be set to work fully automatically whenever you drop a document in the scanner and press the scan button.

You go from something like this



Navn	Dato
CCF_000095.pdf	11-05-2017 10:55
CCF_000093.pdf	11-05-2017 10:51
CCF_000061.pdf	22-04-2017 19:53
CCF_000060.pdf	22-04-2017 19:52
CCF_000033.pdf	18-04-2017 15:49
CCF_000032.pdf	18-04-2017 15:48
CCF_000031.pdf	18-04-2017 15:47
CCF_000030.pdf	18-04-2017 15:41
CCF_000029.pdf	18-04-2017 15:39
CCF_000028.pdf	18-04-2017 15:25
CCF_000027.pdf	14-04-2017 19:55
CCF_000026.pdf	14-04-2017 19:55
CCF_000025.pdf	14-04-2017 19:54
CCF_000024.pdf	14-04-2017 19:53
CCF_000023.pdf	14-04-2017 19:52
CCF_000022.pdf	14-04-2017 19:51
CCF_000021.pdf	14-04-2017 19:51
CCF_000020.pdf	14-04-2017 15:53
CCF_000019.pdf	14-04-2017 15:41
CCF_000018.pdf	14-04-2017 15:40
CCF_000017.pdf	14-04-2017 15:38
CCF_000016.pdf	14-04-2017 15:37
CCF_000015.pdf	14-04-2017 15:36
CCF_000014.pdf	14-04-2017 15:34
CCF_000013.pdf	14-04-2017 15:33
CCF_000012.pdf	14-04-2017 15:32
CCF_000011.pdf	11-04-2017 11:07
CCF_000010.pdf	11-04-2017 10:34
CCF_000009.pdf	06-04-2017 19:58
CCF_000008.pdf	06-04-2017 19:58
CCF_000007.pdf	06-04-2017 19:54
CCF_000006.pdf	06-04-2017 19:49

- to something like this:



Navn	Dato
level2	17-06-2017 21:01
CCF_000051.pdf	20-04-2017 10:43
CCF_000051-2.pdf	20-04-2017 10:43
Dat1-19.06.2007.pdf	20-04-2017 09:58
Dat1-23.08.2005.pdf	20-04-2017 10:06
Dat1-28-10-2007.pdf	20-04-2017 10:39
Dat1-28-10-2007-2.pdf	20-04-2017 10:39

Installation

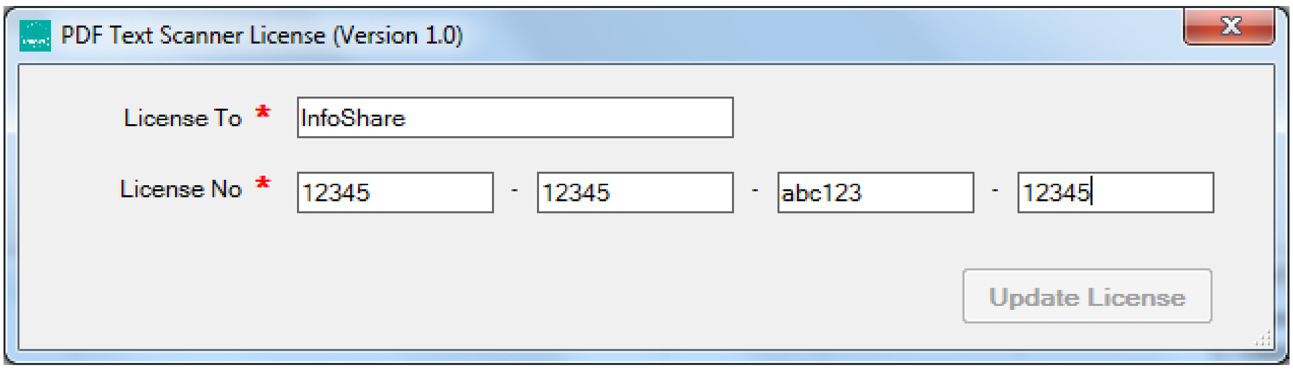
The application PdfTextScanner is easily installed on any Windows based computer by downloading and doubleclicking the installer file pdftextscanner.msi. You can at any time as long as you have a valid license download and install any new version of the application. Installing a new version will require the old version to be uninstalled.

Demo version

Whenever the application is installed without a valid license, it will work in demo-mode: The application will start, handle three pdf files and then you will have to terminate the application and start it again after a short delay to have it handle three more pdf files. This limitation does not exist in the licensed version: It will run from you start it and until you manually stop it.

License

PDF Text Scanner is not freeware and although it can be run in a demo mode, continued use requires a license. The license can be obtained from our homepage, www.infoshare/pdf-text-scanner-license The license is connected to your name, the name of your company or organization, whatever applies. The application can be installed and used on any number of computers belonging to the licensee, but it cannot be transferred or copied to another organization or company. You can find more details on this in the EUAL file provided by the installation program.


PDF Text Scanner License (Version 1.0)

License To * InfoShare

License No * 12345 - 12345 - abc123 - 12345

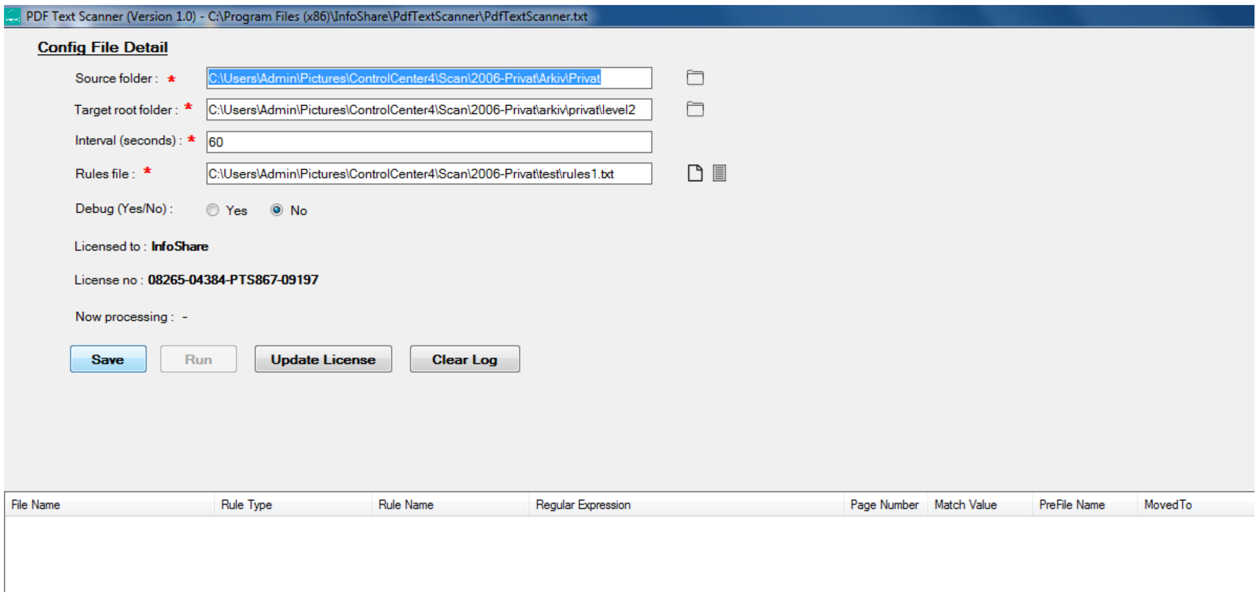
Program configuration

When the program starts it will show you the present configuration which mainly consists of 4 parameters:

1. Source folder where the application should look for pdf files
2. The root of the filestructure where the renamed files should be saved.
3. The interval in seconds between each scan of the source folder for new pdf files
4. The name of the file containing the rules, that are being used for the renaming and moving of the files.

You can change these parameters as well as enter valid license information from inside the program screen by clicking the Edit button.

Setup limitation: The path for source, target and rules file can be a maximum of 190 characters.


PDF Text Scanner (Version 1.0) - C:\Program Files (x86)\InfoShare\PdfTextScanner\PdfTextScanner.txt

Config File Detail

Source folder : * C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\Arkiv\Privat

Target root folder : * C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\arkiv\privat\level2

Interval (seconds) : * 60

Rules file : * C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\test\rules1.txt

Debug (Yes/No) : Yes [✓ No]

Licensed to : InfoShare

License no : 08265-04384-PTS867-09197

Now processing : -

File Name	Rule Type	Rule Name	Regular Expression	Page Number	Match Value	PreFile Name	MovedTo

Rules using regular expressions

The rules is at the center of what this program does: This is where you decide what to look for and what to do whenever a hit is found. You might want to work with different rules files according to scanning of different types of documents and you definitely would revise the rules after a few scans when you gradually come to realize how you would like to have your documents ordered.

You can use two kinds of rules: Rules for moving the pdf file (folder rules) and rules for renaming the pdf file. These rules have one thing in common: The look for a text you have specified as a regular expression in the rule set.

All rules look like this: name of the rule####Regular expression to look for####What to do.

The name of the rule is a free text only used for easier identification on the application debug screen if you need to use this.

Whenever a text conforming to a rule is found, the two ruletypes will act different and independent of each other:

A **renaming rule** will change the name of the file to a short text defined by you followed by the found text.

Sample 1: Date-in-march####[0-3][0-9] March [1-2][0-9]{3}####Date-+

This rule will look for a date like 11 March 2011 and if found the file will be renamed to Date-11 March 2011.pdf (or whatever date in March was found)

The "+" at the end signals that this is a renaming rule and whatever text is between the last # and the + is put in front of the found text to form the new filename. You can look for whatever you need for renaming files, but dates might be a good choice if you are doing a mass scanning of a lot of old documents.

A **folder rule** will move the file to another folder as according to the rule – and create the folder if it is missing. The destination folder can be an absolute path (like c:\My archive\private) or a folder relative to the root target on the setup screen. The folder can either be a constant text or it can be whatever text was found according to the regular expression:

Sample 2: My name####Firstname Lastname####Private

This rule will look for the text Firstname Lastname and whenever that is found the pdf fle will be moved to the subfolder Private.

You might want to add another rule or two:

Sample 3: My name####Firstname Middlename Lastname####Private
Sample 4: My name####FIRSTNAME LASTNAME####Private

For a more detailed ordering of the document you would use folders that are named after the found test:

Sample 5: Personal identification number####[0-9{4} [0-9{4} [0-9{4}####Found value

This rule would look for any number consisting of 3 groups of 4 digits, say 1234 5678 0246, create a folder with that text and move the pdf into that folder. You will ofcourse have to edit the rule to fit the format of the personal identification number, that is used in your documents.

The rules are interpreted sequential: The first renaming rule with a hit stops any further looking for a renaming rule and the first folder rule with a hit would stop looking for more folder rules.

No hit: If you have tried to process a pdf file, that isn't a searchable pdf file, the file will be moved without namechange to the subfolder _non-searchable. If you believe these files contains information, that can be recognized by an OCR program you can run them through your OCR program again – most likely this is a program, that came with your scanner.

It the pdf file is searchable, but doesn't contain any information, that fits with the folder-rules in your rules file, the file will be moved to the subfolder _nohit. This is the case nomatter if the rules file contains name-rules, that give a hit or not (i.e. in _nohit you can find files with their original name as well as renamed files.)

Hierarchical rules sets: You could let the ordering of documents work in a hierarchical way: A first run sort all documents in "Private" and "Company" and the next run, with another set of rules will sort the contents of these folders into a second level of folders. And so forth. You can do this by running the PDF text scanner with different rulefiles one after the other. Or you can do it by running more instances of the program, using different rulefiles. The installation creates a shortcut on your desktop and if you want to have more instances with different rules running, the easiest way is to copy this shortcut and then edit the "Run in" foldername. Whatever foldername you enter in this field in the shortcut must exist (created manually) and the folder must contain a copy of the setup configuration file PdfTextScanner.txt that was originally installed. You'll find this In the installation folder: Program Files (x86)\InfoShare\PdfTextScanner. You then edit this general pdftextscanner.txt file from inside the program to point to the source, targetroot and rulefile, that you need for this instance of the program execution.

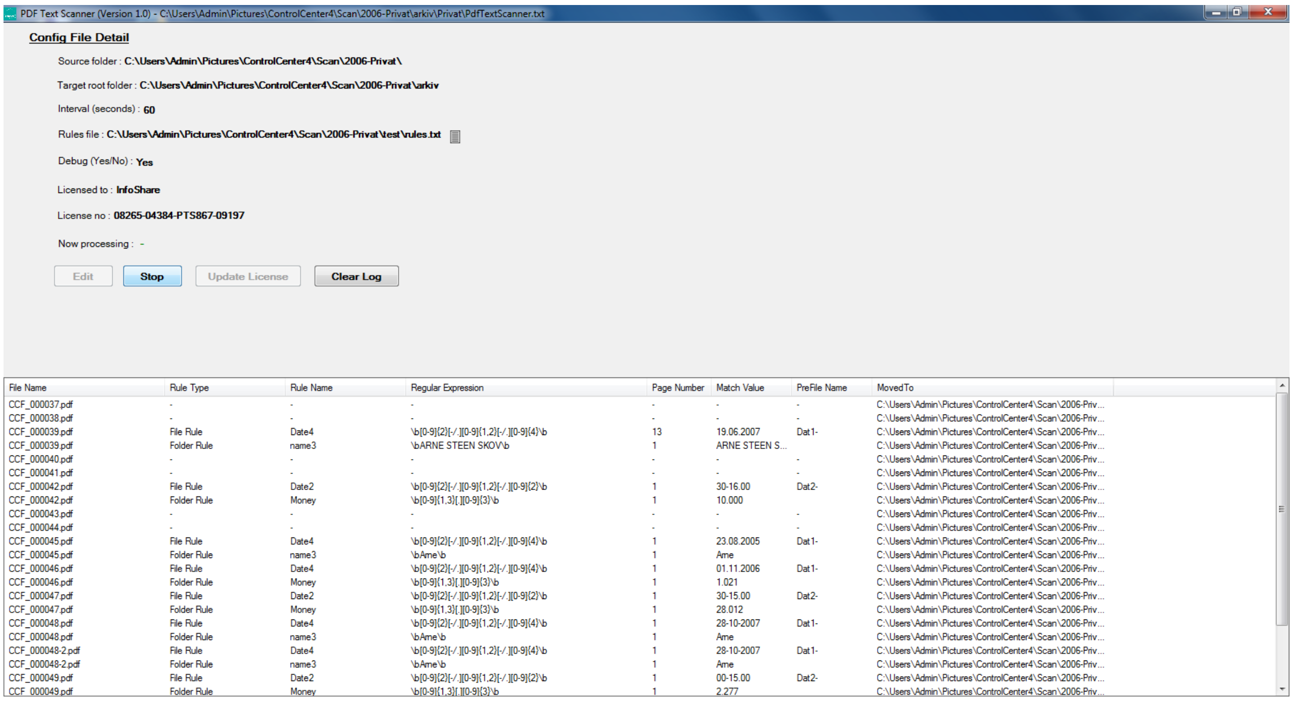
Further information about Regular expressions: A short introduction is in appendix A. You can visit our homepage for more information and for to consult our FAQ (frequently asked questions), http://infoshare.dk/pdf-text-scanner-en or study the Microsoft documentation at:

https://msdn.microsoft.com/en-us/library/system.text.regularexpressions(v=vs.110).aspx

and https://msdn.microsoft.com/en-us/library/ae5bf541(v=vs.100).aspx

Debug option

If you enable the debug option in the setup you will see the debug screen while PDF Text Scanner is working (this is also the case if you run in demo mode):


PDF Text Scanner (Version 1.0) - C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\arkiv\Privat\PdfTextScanner.txt
Config File Detail
Edit
Stop
Update License
Clear Log

Source folder : C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\

Target root folder : C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\arkiv

Interval (seconds) : 60

Rules file : C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Privat\test\rules.txt

Debug (Yes/No) : Yes

Licensed to : InfoShare

License no : 08265-04384-PTS867-09197

Now processing : -

File Name	Rule Type	Rule Name	Regular Expression	Page Number	Match Value	PreFile Name	MovedTo
CCF_000037.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000038.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000039.pdf	File Rule	Date4	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b	13	19.06.2007	Dat1-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000039.pdf	Folder Rule	name3	\bARNE STEEN SKOV\b	1	ARNE STEEN S...		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000040.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000041.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000042.pdf	File Rule	Date2	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{2}\b	1	30-16.00	Dat2-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000042.pdf	Folder Rule	Money	\b[0-9]{1,3}[.][0-9]{3}\b	1	10.000		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000043.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000044.pdf	-	-	-	-	-	-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000045.pdf	File Rule	Date4	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b	1	23.08.2005	Dat1-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000045.pdf	Folder Rule	name3	\bArne\b	1	Arne		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000046.pdf	File Rule	Date4	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b	1	01.11.2006	Dat1-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000046.pdf	Folder Rule	Money	\b[0-9]{1,3}[.][0-9]{3}\b	1	1.021		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000047.pdf	File Rule	Date2	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{2}\b	1	30-15.00	Dat2-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000047.pdf	Folder Rule	Money	\b[0-9]{1,3}[.][0-9]{3}\b	1	28.012		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000048.pdf	File Rule	Date4	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b	1	28-10-2007	Dat1-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000048.pdf	Folder Rule	name3	\bArne\b	1	Arne		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000048-2.pdf	File Rule	Date4	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b	1	28-10-2007	Dat1-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000048-2.pdf	Folder Rule	name3	\bArne\b	1	Arne		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000049.pdf	File Rule	Date2	\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{2}\b	1	00-15.00	Dat2-	C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...
CCF_000049.pdf	Folder Rule	Money	\b[0-9]{1,3}[.][0-9]{3}\b	1	2.277		C:\Users\Admin\Pictures\ControlCenter4\Scan\2006-Priv...

The screen shows you which files are processed, how the first name and/or folder rule fitted that document and the resulting filename.

Furthermore the debug option will save a textfile with the full text version of the document. This file is saved in the subfolder PdfExtractedFiles – use it, if you cannot understand why you don't get a hit: The OCR program might have produced another result, than you expect. (Sample: Some OCR programs fill in blanks if the font is a monospaced font – 7913 might become 79 1 3)

Minimized screen

If you minimize the debug/configuration screen it will be inimized to the systemtray and keep on running. You can right click on the icon in the systemtray and maximize the screen, stop or start the processing.

Contact

You can look at our homepage for further hints, information in other languages or newer versions: http://www.infoshare.dk/pdf-text-scanner-en (there's a link in the program as well) and feel free to contact us with questions, proposals, request for development of rules files at our email address: pdftextscanner@infoshare.dk.

Or visit us at InfoShare, Degnemose Alle 18, DK 2700 Brønshøj, Denmark.

Appendix A. regular expressions

PDF Text Scanner searches for “regular expressions” and that can be straight forward text or it can be a bit more complicated. Some samples might give you a general expression:

MyCompany searches for the exact text MyCompany.

Mycompany searches for the exact text Mycompany

My[Cc]ompany searches for MyCompany or Mycompany

\r\nMyCompany searches for MyCompany at the beginning of a line – i.e. after CR LF)

123 searches for the exact string 123

\d searches for any number

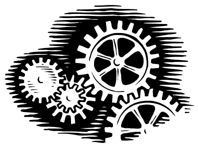
\d{3} searches for any 3 consecutive numbers (like 123 or 542 etc)

\d{3}\s\d{4} searches for 3 numbers followed by whitespace followed by four numbers (like 234 4543)

You can also take a look at the sample rules.txt file, that is installed with the program and you can use the Microsoft cheatsheet below.

Searching the web for regular expressions will give you many more sources for information, but beware: There might be minor differences in the implementation of regular expressions – we use the Microsoft library.

.NET FRAMEWORK REGULAR EXPRESSIONS



SINGLE CHARACTERS

Use	To match any character
[*set*]	In that set
[^*set*]	Not in that set
[*a–z*]	In the *a-z* range
[^*a–z*]	Not in the a-z range
.	Any except \n (new line)
char	Escaped special character

CONTROL CHARACTERS

Use	To match	Unicode
\t	Horizontal tab	\u0009
\v	Vertical tab	\u000B
\b	Backspace	\u0008
\e	Escape	\u001B
\r	Carriage return	\u000D
\f	Form feed	\u000C
\n	New line	\u000A
\a	Bell (alarm)	\u0007
\c *char*	ASCII control character	–

NON-ASCII CODES

Use	To match character with
octal	2-3 digit octal character code
\x *hex*	2-digit hex character code
\u *hex*	4-digit hex character code

CHARACTER CLASSES

Use	To match character
\p{*ctgry*}	In that Unicode category or block
\P{*ctgry*}	Not in that Unicode category or block
\w	Word character
\W	Non-word character
\d	Decimal digit
\D	Not a decimal digit
\s	White-space character
\S	Non-white-space char

QUANTIFIERS

Greedy	Lazy	Matches
*	*?	0 or more times
+	+?	1 or more times
?	??	0 or 1 time
{*n*}	{*n*}?	Exactly *n* times
{*n,*}	{*n,*}?	At least *n* times
{*n,m*}	{*n,m*}?	From *n* to *m* times

ANCHORS

Use	To specify position
^	At start of string or line
\A	At start of string
\z	At end of string
\Z	At end (or before \n at end) of string
$	At end (or before \n at end) of string or line
\G	Where previous match ended
\b	On word boundary
\B	Not on word boundary

GROUPS

Use	To define
(*exp*)	Indexed group
(?<*name*>*exp*)	Named group
(?<*name1-name2*>*exp*)	Balancing group
(?:*exp*)	Noncapturing group
(?=*exp*)	Zero-width positive lookahead
(?!*exp*)	Zero-width negative lookahead
(?<=*exp*)	Zero-width positive lookbehind
(?<!*exp*)	Zero-width negative lookbehind
(?>*exp*)	Non-backtracking (greedy)

INLINE OPTIONS

Option	Effect on match
i	Case-insensitive
m	Multiline mode
n	Explicit (named)
s	Single-line mode
x	Ignore white space

Use	To
(?imnsx-imnsx)	Set or disable the specified options
(?imnsx-imnsx:*exp*)	Set or disable the specified options within the expression

June 2014

BACKREFERENCES

Use	To match
\n	Indexed group
\k<*name*>	Named group

ALTERNATION

Use	To match
a |*b*	Either *a* or *b*
(?(*exp*) *yes* | *no*)	*yes* if *exp* is matched *no* if *exp* isn't matched
(?(*name*) *yes* | *no*)	*yes* if *name* is matched *no* if *name* isn't matched

SUBSTITUTION

Use	To substitute
$*n*	Substring matched by group number *n*
${*name*}	Substring matched by group *name*
$$	Literal $ character
$&	Copy of whole match
$`	Text before the match
$'	Text after the match
$+	Last captured group
$_	Entire input string

COMMENTS

Use	To
(?# *comment*)	Add inline comment
#	Add x-mode comment

For detailed information and examples, see **http://aka.ms/regex**

To test your regular expressions, see **http://regexlib.com/RETester.aspx**

SUPPORTED UNICODE CATEGORIES

Category	Description
Lu	Letter, uppercase
Ll	Letter, lowercase
Lt	Letter, title case
Lm	Letter, modifier
Lo	Letter, other
L	Letter, all
Mn	Mark, nonspacing combining
Mc	Mark, spacing combining
Me	Mark, enclosing combining
M	Mark, all diacritic
Nd	Number, decimal digit
Nl	Number, letterlike
No	Number, other

N	Number, all
Pc	Punctuation, connector
Pd	Punctuation, dash
Ps	Punctuation, opening mark
Pe	Punctuation, closing mark
Pi	Punctuation, initial quote mark
Pf	Puntuation, final quote mark
Po	Punctuation, other
P	Punctuation, all
Sm	Symbol, math
Sc	Symbol, currency
Sk	Symbol, modifier
So	Symbol, other
S	Symbol, all
Zs	Separator, space
Zl	Separator, line
Zp	Separator, paragraph
Z	Separator, all
Cc	Control code
Cf	Format control character
Cs	Surrogate code point
Co	Private-use character
Cn	Unassigned
C	Control characters, all

For named character set blocks (e.g., Cyrillic), search for "supported named blocks" in the MSDN Library.

REGULAR EXPRESSION OPERATIONS

Class: System.Text.RegularExpressions.Regex

Pattern matching with Regex objects

To initialize with	Use constructor
Regular exp	Regex(String)
+ options	Regex(String, RegexOptions)
+ time-out	Regex(String, RegexOptions, TimeSpan)

Pattern matching with static methods

Use an overload of a method below to supply the regular expression and the text you want to search.

Finding and replacing matched patterns

To	Use method
Validate match	Regex.IsMatch
Retrieve single match	Regex.Match (first) Match.NextMatch (next)
Retrieve all matches	Regex.Matches
Replace match	Regex.Replace
Divide text	Regex.Split
Handle char escapes	Regex.Escape Regex.Unescape

Getting info about regular expression patterns

To get	Use Regex API
Group names	GetGroupNames GetGroupNameFromNumber
Group numbers	GetGroupNumbers GetGroupNumberFromName
Expression	ToString
Options	Options
Time-out	MatchTimeOut
Cache size	CacheSize
Direction	RightToLeft

Appendix B. Notes on the sample rules file

The included rules.txt file contains some sample rules, that you can use, modify or study for making your own rules.

Anything before the first #### is just for identification purposes. Here we have used F to signal a folder rule and N to signal a name rule. Here the name rules are last but you can mix them t your likings. The rules in each group are evaluated in the order the are found in the file and whenever there is a hit in a group (folder or name) no more evaluation takes place.

```
01F Company1####\b(Ii)nfo(Ss)hare\b####InfoShare
02F Company2####\b(nyborg|Nyborg|NYBORG)\b####Nyborg
03F Person1a####\bArne Skov\b####Private
04F Person1b####\bARNE SKOV\b####Private
05F Person1c####\bArne\b####Privat
06F Person1d####\bARNE\b####Privat
07F Person2a####\b(?ix)Anton Bach Carlsen(?-ix)\b####Anton
08F Person2b####\b(?ix)Anton Carlsen(?-ix)\b####Anton
09F Money1####\b[0-9]{1,3}[.][0-9]{3}\b####Accounting
10F Money2####\b[0-9]{1,3},[0-9]{2}\b####Accounting
11F Carreg1a####\b\w(2)[0-9]{5}\b####Found value
12F Carreg1b####\b\w(2) [0-9]{5}\b####Found value
13F Carreg1c####\b\w(2) [0-9]{2} [0-9]{3}\b####Found value
14F Personid1####\b(?x)\d{10}(?-x)\b####Found value
15F Personid2####\b(?x)\d{6}-\d{4}(?-x)\b####Found value
16F Personid3####\b[0123]\d[01]\d{3}[-\s]{0,1}\d{4}\b####Found value
17N Date1####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{2}\b####Date-+
18N Date2####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
19N Date3####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
20N Date4####\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
21N Month1####\b(?i)January(?-i) [0-9]{2}\b####Mth-+
22N Month2####\b(?i)January(?-i) [0-9]{4}\b####Mth-+
23N Month2####\b(?i)February(?-i) [0-9]{2}\b####Mth-+
24N Month2####\b(?i)February(?-i) [0-9]{4}\b####Mth-+
25N Month2####\b(?i)Marts(?-i) [0-9]{2}\b####Mth-+
26N Month2####\b(?i)Marts(?-i) [0-9]{4}\b####Mth-+
27N Month2####\b(?i)April(?-i) [0-9]{2}\b####Mth-+
28N Month2####\b(?i)April(?-i) [0-9]{4}\b####Mth-+
```

Comments:
01F Company1####\b(Ii)nfo(Ss)hare\b####InfoShare: Between word boundaries (space, new line etc) look for InfoShare where each of I and S can be either a capital letter or not, i.e. infoShare would be valid. Put any pdf file where this is found into the subfolder InfoShare and stop looking for match I folder names.

02F Company2####\b(nyborg|Nyborg|NYBORG)\b####Nyborg: Between word boundaries look for nyborg, Nyborg or NYBORG and put any found file into the subfolder Nyborg and stop searching.

03F Person1a####\bArne Skov\b####Private: Between word boundaries look for the name Arne Skov and put any found file into the folder Private and stop searching.
04F Person1b####\bARNE SKOV\b####Private: Between word boundaries look for the name ARNE SKOV and put any found file into the folder Private and stop searching.
05F Person1c####\bArne\b####Privat: Between word boundaries look for the name Arne and put any found file into the folder Private and stop searching.
06F Person1d####\bARNE\b####Privat: Between word boundaries look for the name ARNE and put any found file into the folder Private and stop searching.

07F Person2a####\b(?ix)Anton Bach Carlsen(?-ix)\b####Anton: Between word boundaries look for the name Anton Bach Carlsen where you ignore the case of all letters and ignore whitespace (i.e none, one or more spaces between the names is ok) and put any found file into the subfolder Anton and stop searching.
08F Person2b####\b(?ix)Anton Carlsen(?-ix)\b####Anton: Between word boundaries look for the name Anton Carlsen where you ignore the case of all letters and ignore whitespace and put any found file into the subfolder Anton and stop searching.

Looking for documents that contain amounts typical for a bank statement and putting the files into the subfolder Accounting.
09F Money1####\b[0-9]{1,3}[.][0-9]{3}\b####Accounting
10F Money2####\b[0-9]{1,3},[0-9]{2}\b####Accounting

Danish car licenseplates have two letters and 5 digits. It will be written as AB12345, AB 12345 or AB 12 345. The following lines search for these combinations and put found files into a folder, that is named as the found value (i.e. AB12345, AB 12345 or AB 12 345)
11F Carreg1a####\b\w(2)[0-9]{5}\b####Found value
12F Carreg1b####\b\w(2) [0-9]{5}\b####Found value
13F Carreg1c####\b\w(2) [0-9]{2} [0-9]{3}\b####Found value

Danish personident consist of 10 digits, normally written as 1212560123 or 121256-0123. For good measure the following lines look for this and ignore any additional whitespace. Document where this is found is put into a subfolder with the same number combination (i.e. 1212560123, 121256-0123 or 121256 0123)
14F Personid1####\b(?x)\d{10}(?-x)\b####Found value
15F Personid2####\b(?x)\d{6}-\d{4}(?-x)\b####Found value

In fact we know that the first 6 digits represent ddmmyy, so we can use this as a limitation in the search. The following line will only accept 0,1,2,3 in the first position, the a digit and only 0 or 1 in the third position. After the date you can have either – or whitespace or nothing (0 occurences of – or whitespace). The file with the match will again be put into a folder with the found value as the name.
16F Personid3####\b[0123]\d[01]\d{3}[-\s]{0,1}\d{4}\b####Found value

The following rules are used to rename the file. The all look for a date with different kinds of syntax. If found the new filename will be “Date-“ followed by the found value.

```
17N Date1####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{2}\b####Date-+
18N Date2####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
19N Date3####\b[0-9]{1,2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
20N Date4####\b[0-9]{2}[-/.][0-9]{1,2}[-/.][0-9]{4}\b####Date-+
```

Similar for another type of dates: Monthname followed by a space and 2 or 4 digit (year). Resulting filename will be “Mth-“ followed by whatever value was matched.

```
21N Month1####\b(?i)January(?-i) [0-9]{2}\b####Mth-+
22N Month2####\b(?i)January(?-i) [0-9]{4}\b####Mth-+
23N Month2####\b(?i)February(?-i) [0-9]{2}\b####Mth-+
24N Month2####\b(?i)February(?-i) [0-9]{4}\b####Mth-+
25N Month2####\b(?i)Marts(?-i) [0-9]{2}\b####Mth-+
26N Month2####\b(?i)Marts(?-i) [0-9]{4}\b####Mth-+
27N Month2####\b(?i)April(?-i) [0-9]{2}\b####Mth-+
28N Month2####\b(?i)April(?-i) [0-9]{4}\b####Mth-+
```