OCR

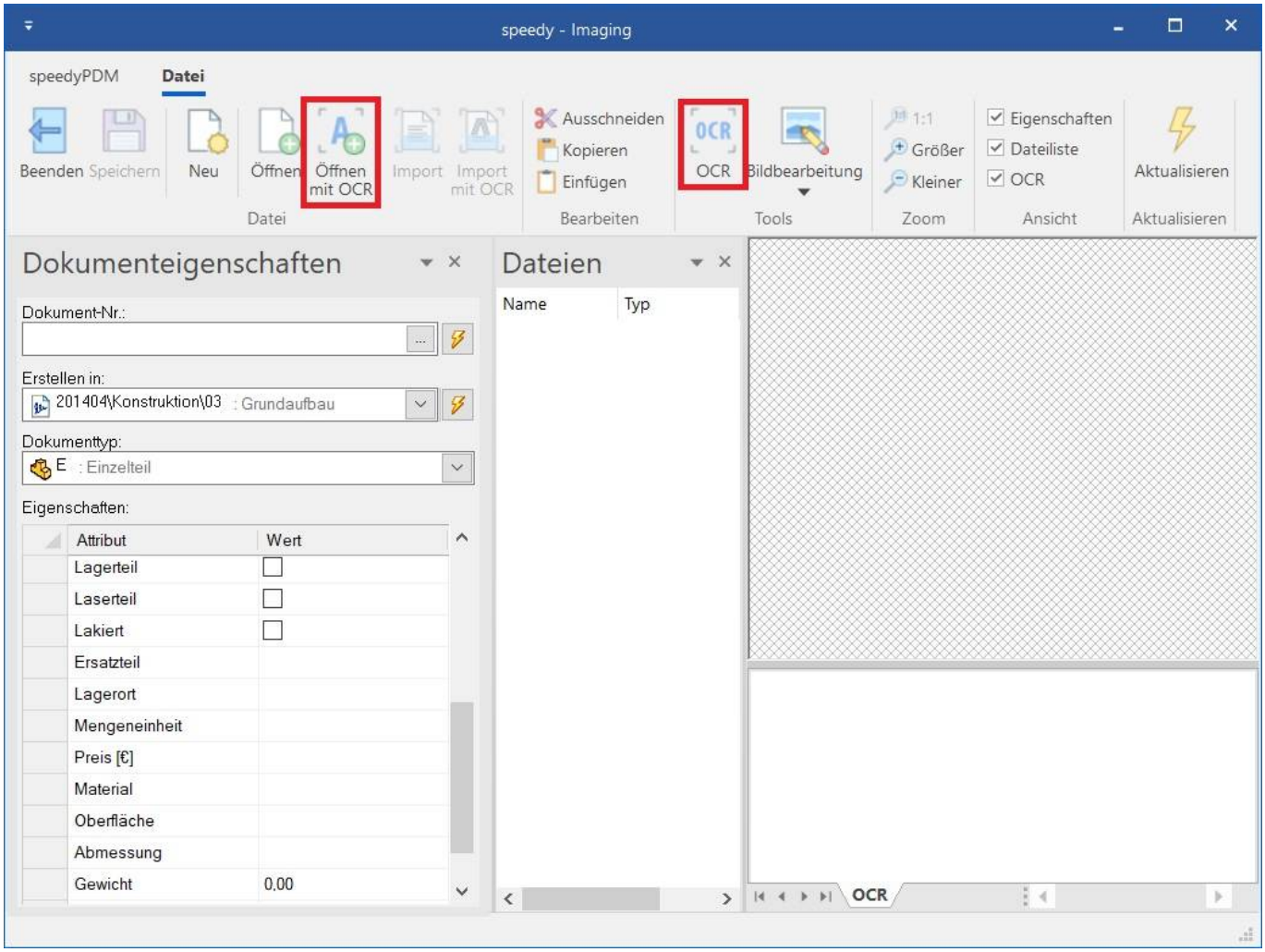
In speedyPDM gibt es die Möglichkeit eine OCR Indizierung einzurichten, um z.B. Rechnungen oder Angebote automatisch in speedy einzupflegen. Um dieses Modul nutzen zu können muss eine Lizenzierung für dieses Modul vorhanden sein. Es gibt 2 Möglichkeiten die OCR Funktionalität in speedy zu nutzen:

Ablauf der OCR Indizierung

Als erstes wird falls es noch keins ist eine Bilddatei erstellt. D.h. dass z.B. eine PDF in eine .tif umgewandelt wird. Diese Dateien werden im Temp-Ordner des Users abgelegt. Ist dies geschehen wird mit Hilfe des Tesseract Tool der Text aus den Bildern gelesen und eine TXT-Datei ebenfalls im Temp-Ordner abgelegt. Danach werden die einzelnen Rules abgearbeitet. Ist nach vollständiger Abarbeitung keine Eigenschaft gefunden worden, wird die Indizierung mit einer anderen Einstellung nochmals durchgeführt.

Halbautomatische Indizierung (Imaging)

Um für die ersten schritte und das Einstellen des OCR´s auf die verschiedenen Rechnungsarten zu Vereinfachen gibt es im Imaging Dialog die Möglichkeit eine OCR Indizierung durchzuführen. Den Imaging Dialog befindet sich unter **Dokument→Neu→Imaging**.



Imaging

Um die Indizierung ausführen zu können öffnet man die gewünschte Datei und klickt dann auf den OCR-Button in der Taskleiste.
Des weiteren gibt es die Möglichkeit eine oder mehrere Dateien direkt mit einer OCR-Indizierung zu öffnen. Dafür ist der Öffnen Button mit OCR.

Konnten Eigenschaften aus der Datei gelesen werden, werden sie nach der Indizierung links angezeigt. Dies bedeutet dass Dokumentennummer, Ordner, Dokumenttypen und alle Eigenschaften Theoretisch ausgelesen werden können, je nachdem wie die Indizierung Konfiguriert ist.

Vollautomatische Indizierung

Um OCR Vollautomatisch zu nutzen wird der speedy-Spooler verwendet. Dieser Überwacht ein Verzwichnis in dass z.B. Rechnungen direkt vom Drucker aus kommen und arbeitet dieses dann sofort ab sobald Dateien sich darin befinden. Damit ist es möglich einen Automatischen Import zu realisieren.

Regelwerk/Einrichtung

Die OCR Indizierung wird durch ein Regelwerk, dass in der Datenbank beschrieben ist durchgeführt. Dieses Regelwerk steht in der ocr_rules Tabelle. Das Regelwerk wird nach einander abgearbeitet, d.h. dass die erste ebene (pid = 0) danach die dazugehörige 2. Pid usw. abgearbeitet werden. Die pid ist immer die id nachdem diese Rule abgearbeitet werden soll. Die Pattern werden in Regex-Ausdrücken ausgelesen. Dies gibt einem die Möglichkeit viele verschiedene Fälle in Bezug auf Texterkennung abzudecken. Ein Tool zur Erstellung des Regex-Ausdruckes kann hier gefunden werden: https://regex101.com/. Zur Einrichtung einer Vorlage wird empfohlen dies im Imaging Modul zu machen und die Einstellung [ocr.deletefiles] zu nutzen um die txt Datei abzufangen und damit im Regex Editor die Pattern zu erstellen oder das Fenster mit dem OCR-Text anzeigen lassen. Um z.B. aus einer pdf-Datei einen Text zu extrahieren nutzen wir ein Tool namens Tesseract. Dieses Tool erzeugt aus Bilddateien einen Text den wir dann durch das Pattern indizieren können. Um ein besseres Ergebnis der Indizierung und der Trefferquote zu bekommen können verschiedene Einstellungen getroffen werden. Diese werden weiter hinten beschrieben. Um ein bestmögliches Ergebnis zu bekommen wird empfohlen immer den gleichen Scanner zu verwenden, da unterschiedliche Auflösungen, das Ergebnis verschlechtern können. Sollte ein Ausschnitt genommen werden ist es essenziell notwendig den gleichen Scanner zu benutzen! Die OCR-Indizierung steht im Imaging Modul und in der dwImportOcr.exe zur Verfügung. Die dwImportOcr.exe ist dafür da eine Automatische Erkennung in Verbindung mit der dwSpool zu nutzen. D.h. es können z.B. Angebote direkt in ein Verzeichnis gescannt werden und wenn die Erkennung Funktioniert werden die Dokumente automatisch in Speedy eingepflegt. Wenn ein Fehler auftritt wird diese Dateien in ein Unterverzeichnis Error verschoben.

Die ocr_rules Tabelle hat folgende Spalten:

Spalte	**Beschreibung**
ocr_id	Fortlaufender Primary Key.
ocr_pid	Parent ID mit ihr wird auf den Primary Key der vorhergehenden Rule verwiesen, d.h. die vorhergehende Rule muss abgearbeitet sein damit diese Rule ausgeführt wird.
ocr_pcontext	Hier wird der Kontext angegeben unter dem diese Rule abgearbeitet werden soll. Ein Beispiel: diese Rule gilt nur für einen bestimmten Lieferanten (MMH) dann steht in dieser Spalte MMH. Diese Rule wird dann nur abgearbeitet, wenn zuvor der Lieferant MMH gefunden wurde.
ocr_index	Momentan nicht verwendet.
ocr_rulename	Name der Rule.
ocr_ruledesc	Beschreibender Text der Rule.
ocr_pattern	Regex Ausdruck um in dem erkannten Text nach bestimmten Textzeichen zu suchen.
ocr_matchindex	Gibt an welches Match des Regex Ausdruckes das Ergebnis ist.
ocr_searchlevel	Wird momentan nicht verwendet.
ocr_propname	Eigenschaften Name unter welchem diese Eigenschaft bzw. das Ergebnis gespeichert wird.
ocr_propdefault	wenn ocr_flag = 2 ist dann wird dieser Wert in die Eigenschaft geschrieben
ocr_extscript	Momentan nicht verwendet

Spalte	Beschreibung
ocr_selectstate	Ein Select Statement, dass abgearbeitet wird: - wenn ocr_flag = 3: Dann wird ein Platzhalter in diesem Select Statement durch einen Wert der schon ermittelt wurde ausgetauscht. Dieser Platzhalter wird mit < > Signalisiert. Zum Beispiel <Ben1>. - wenn ocr_flag = 1: Dann wird direkt das Ergebnis dieser Abfrage in die Eigenschaft geschrieben. Das Select Statement sollte folgendermaßen aussehen: `SELECT [ocr_pattern], [ocr_matchindex], [ocr_propdefault], [ocr_pcontext]`
ocr_sector	Es kann ein Rechteck ausgeschnitten werden um z.B. ein besseres lese Ergebnis zu bekommen. Beispiel: {322,201,536,314} (Linkes oberes Eck + Rechtes unteres Eck) Hinweis: Dazu das Setting **ocr.deletefiles** auf 0 setzen und im Temp Ordner die dazugehörige .png-Datei in Paint öffnen und die Koordinaten bestimmen.
ocr_flag	gibt an, welcher Wert zu einer Eigenschaft genommen wird: := 1: Wert der durch Regex oder SQL Abfrage ermittelt wurde wird genommen. := 2: Wert der in ocr_propdefault steht wird genommen. := 3: SQL Statement wird ein Platzhalter ersetzt.

Beispiel eines Regelwerks für ein Angebot[1] dass die Angebotsnummer herausliest[2], in einer Angebots-Tabelle nach der Kundennummer sucht[3], ein Rechtecksauschnitt macht[4] und in diesem Ausschnitt dann das Angebotsdatum sucht[5].

ocr_id	ocr_pid	ocr_pcontext	ocr_index	ocr_rulename	ocr_ruledesc	ocr_pattern	ocr_matchindex	ocr_searchlevel	ocr_propname	ocr_propdefault	ocr_extscript	ocr_selectstate	ocr_sector	ocr_flag
1	0		0	Angebot		(Angebot:\s+)([\w]+)	0	-1	doc_doctype	Angebot				2
2	1	MMH	0	Angebotsnummer		(Angebot:\s+)([\w][0-9]+)	2	-1	dm_docno					1
3	1		0	Kunde	Kunde aus Datenbank suchen			-1	kd_nr			select kd_name from angebot_db.angebot a inner join angebot_db.kunden k ON a.ang_kunde=k.kd_id where a.ang_nr='<BEN1>';		3
4	2	MMH	0	Rechteck			3	-1	Rectangle				{322,201,536,314}	1
5	4	MMH	0	Angebotsdatum		(Datum:\s)([0-9]+.[0-9]+.[0-9]+)	2	-1	KOMMENTAR					1

Creation Code der ocr_rules Tabelle:

```
CREATE TABLE `ocr_rules` (
`ocr_id` INT(11) NOT NULL AUTO_INCREMENT,
`ocr_pid` INT(11) NOT NULL DEFAULT '0',
`ocr_pcontext` VARCHAR(255) NULL DEFAULT NULL,
`ocr_index` INT(11) NOT NULL DEFAULT '0',
`ocr_rulename` VARCHAR(50) NULL DEFAULT NULL,
`ocr_ruledesc` VARCHAR(50) NULL DEFAULT NULL,
`ocr_pattern` VARCHAR(255) NULL DEFAULT NULL,
`ocr_matchindex` INT(11) NULL DEFAULT NULL,
`ocr_searchlevel` INT(11) NULL DEFAULT '-1',
`ocr_propname` VARCHAR(50) NULL DEFAULT NULL,
`ocr_propdefault` VARCHAR(50) NULL DEFAULT NULL,
`ocr_extscript` VARCHAR(50) NULL DEFAULT NULL,
`ocr_selectstate` VARCHAR(255) NULL DEFAULT NULL,
`ocr_sector` VARCHAR(255) NULL DEFAULT NULL,
`ocr_flag` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`ocr_id`)
```

```
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB
```

Settings

Setting	Beschreibung
ocr.tesseract.exe	Gibt den Pfad zur Tesseract Exe an. Default: .\\tools\\tesseract\\tesseract.exe
ocr.tesseract.tessdata	tessdata Directory Default: .\\tools\\tesseract\\tessdata
ocr.multitiff	Gibt an ob nur die 1.Seite OCR Indiziert werden soll oder nicht. := 1: Alle Seiten werden Indiziert. := 0: Nur die erste Seite wird Indiziert (Default)
ocr.tesseract.language	Gibt die Sprache an mit der Indiziert werden soll. Default: deu
ocr.tesseract.psm1	Page Segmentation Mode, gibt an mit welcher Einstellung im ersten Durchgang der Tesseract Indizieren soll (tesseract hilfe). Default:0
ocr.tesseract.psm2	Page Segmentation Mode, gibt an mit welcher Einstellung im zweiten Durchgang der Tesseract Indizieren soll (tesseract hilfe). Default:12
ocr.tesseract.oem	Gibt den Ocr-Engine Mode an. Default:3
ocr.tesseract.configvar	Hier kann man die Variablen angeben die für die Konfiguration des Tesseract notwendig sind. Es können mehrere Variablen hintereinander angegeben werden. VAR=Value Default: keine Variablen gesetzt
ocr.color.colored	Gibt an ob die Indizierung Farbig oder Schwarz weiß stattfindet. Default: 1 (Farbig)
ocr.deletefiles	Löscht die erzeugten Dateien im Temp Ordner. Default: 1 (Löschen)
ocr.zoom	Zoomfaktor der beim Umwandeln von PDF in TIFF verwendet wird um ein besseres OCR Ergebnis zu erzielen. Standardwert := 2.0

From:
https://wiki.speedy-pdm.de/ - **speedyPDM - Wiki**

Permanent link:
https://wiki.speedy-pdm.de/doku.php?id=speedy:30_modules:ocr

Last update: **2020/12/29 12:24**