ELO Preview Converter

[Stand: 14.03.2019 | Programmversion: 12.00.000]

Dieses Dokument beschreibt den Aufbau, die Installation und Wartung des ELO Preview Converters.

Inhalt

1 Aufgaben und Aufbau 2
2 Installation und Inbetriebnahme 3
2.1 Fenstertitelüberwachung 6
2.1.1 Beispiel für eine Titelüberwachung 7
3 Überwachung und Monitoring 9
4 Konvertierung in mehreren ELO Archiven 11
5 Anhang 13
5.1 Teilpakete des ELO Preview Converters 13
5.2 Parameter der COM-Konverters 15
5.3 Parameter des EXE Konverters 17
5.4 Parameter des JPEG-Konverters 17
5.5 Parameter des Printer-Konverters 18
5.6 Anpassungen in den Standardpaketen Java und Tomcat 18
5.7 Bekannte Probleme 20
5.7.1 COM-Konverter 20
5.7.2 Aspose-Konverter 20
5.7.3 Print-Konverter 21

1 Aufgaben und Aufbau

Aufgabe des ELO Preview Converters ist es, neue Dokumente im ELO Archiv mit einer PDF-Vorschaudatei zu ergänzen. Diese kann verwendet werden, wenn auf dem Client kein passendes Originalprogramm zur Anzeige vorliegt oder wenn die Originaldatei zu umfangreich wäre.

Durch ein integriertes Konvertierungsprogramm erkennt und konvertiert der ELO Preview Converter folgende Dateitypen: DOC, DOCX, XLS, XLSX, PPT, PPTX, MSG und EML. Alle anderen Dateitypen werden nicht unterstützt.

Es wird ausschließlich die Arbeitsversion eines Archivdokuments konvertiert. Nach dem Einchecken einer neuen Dokumentversion wird diese dann vom ELO Preview Converter konvertiert.

Folgende Dokumente werden nicht konvertiert:

1. Gesperrte Archivdokumente kann der ELO Preview Converter nicht konvertieren. In diesem Fall wird folgende Info-Log-Ausgabe erstellt:

```
Locked entry, cannot create preview, ignored:
```

2. Zudem ignoriert der ELO Preview Converter „ReadOnly"-Archivdokumente. Es erscheint folgende Info-Log-Ausgabe:

```
Write protected entry, cannot create preview, ignored:
```

3. Gelöschte Archivdokumente werden nicht konvertiert.

2 Installation und Inbetriebnahme

Der ELO Preview Converter wird Ihnen als ZIP-Datei zur Verfügung gestellt, welche alle Programmteile enthält, die für den Betrieb notwendig sind. Je nach verwendeten Konvertern sind aber zusätzlich externe Programme notwendig, z. B. Microsoft Office, falls der COM-Konverter verwendet wird.

Der Konverter muss in einer eigenen physikalischen oder virtuellen Maschine laufen. Das ist notwendig, da bei Störungen verschiedene Prozesse zwangsweise beendet werden. Das könnte bei anderen Prozessen, die auf der gleichen Maschine laufen, Fehlfunktionen auslösen.

Da ein Konverter-Lauf durchaus mehrere Minuten pro Dokument in Anspruch nehmen kann, ist es sinnvoll, die Timeoutzeiten für die Anmeldung und Such-Haltezeit des Indexservers auf eine Stunde zu verlängern. Die Paketgröße der Suche sollte bei 50 liegen, sie kann über den Optionsparameter *packet* eingestellt werden.

Die ZIP-Datei muss auf dem Laufwerk C im Wurzelverzeichnis entpackt werden. Prinzipiell sind auch andere Pfade möglich, dann müssen aber an verschiedenen Stellen Konfigurationen und Skripte angepasst werden. Im Folgenden wird bei den Pfadangaben von einer Standardinstallation ausgegangen.

Falls der PDF-Printer-Konverter verwendet werden soll, muss nun noch der ELO PDF Drucker oder ein anderer PDF-Drucker installiert werden. Die einzige Bedingung an die Druckausgabe ist, dass zum Abschluss ein PDF-Dokument im Verzeichnis C:\ELOpreview\pdfprinter erzeugt wird. Falls der Drucker nicht auf dieses Verzeichnis eingestellt werden kann, ist es auch möglich, in den Optionen unter *pdfprinter* einen anderen Pfad einzutragen.

Als nächstes muss in der Datei C:\ELOpreview\config\cv-elo\config.xml der Indexserverzugriff eingerichtet werden. Diese Datei enthält dafür drei Parameter, die Indexserver URL, den ELO Benutzernamen und das ELO Passwort. Optional kann im gleichen Verzeichnis noch die Datei *log4j.properties* mit den Log-Ausgaben angepasst werden.

Vor dem Start muss nun noch die ELOcv Base Struktur im Archiv eingerichtet werden. Diese kann als Importdatensatz aus dem Paket c:\ELOpreview\download\ELOcv Base.zip geladen werden.

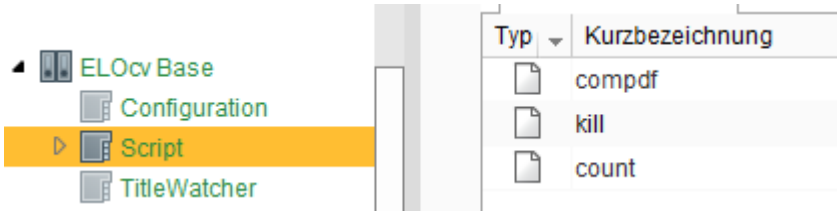


Abb. 1: ELOcv Base

Der Ordner *ELOcv Base* enthält anschließend drei Ordner, *Configuration*, *Script* und *TitleWatcher*. Im Zusatztext des Configuration-Ordners werden die Parameter für den Konverter festgelegt.

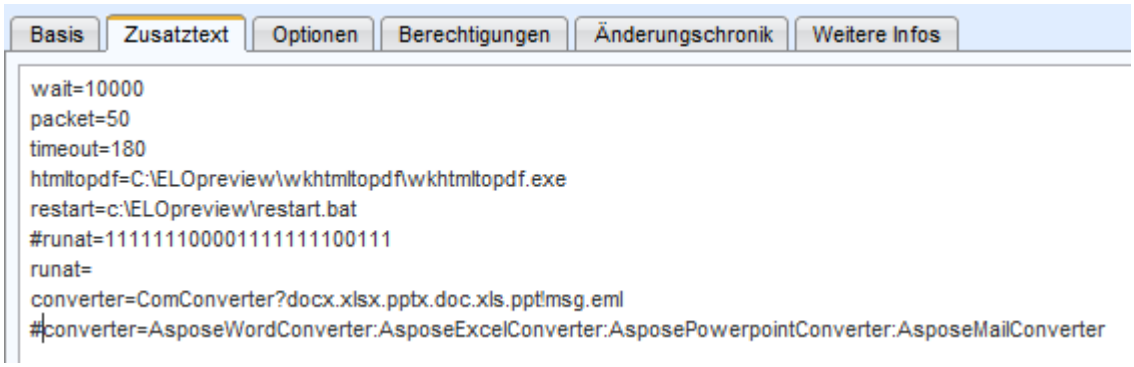


Abb. 2: Zusatztext des Ordners 'Configuration'

Nicht verwendete Einstellungen können per # in der ersten Spalte auskommentiert werden. Dieses ermöglicht im Testbetrieb einen schnellen Wechsel zwischen unterschiedlichen Konfigurationen.

wait	Der Konverter läuft in einer dauerhaften Schleife: er fragt beim Indexserver eine Liste der noch nicht konvertierten Dokumente ab und arbeitet diese Liste anschließend ab. Danach beginnt der nächste Durchlauf. Über den Parameter *wait* kann bestimmen, wie viele Sekunden Wartezeit zwischen zwei Durchläufen eingefügt werden soll.
packet	Die Liste der zu konvertierenden Dokumente wird in Paketen vom Indexserver abgeholt. Diese Paketgröße sollte nicht zu groß sein, da es sonst bei langlaufenden Konvertierungen zu Verbindungsabbrüchen kommen kann.
timeout	Der COM-Konverter muss immer damit rechnen, dass eine externe Konvertierungsanwendung hängen bleibt. Das kann z.B. dann passieren, wenn ein Word-Dokument ein Passwort enthält. Word zeigt dann einen Eingabedialog an und stoppt, da niemand da ist, der das Passwort eingeben könnte. Aus diesem Grund wird der Konvertierungsprozess mit einem Timer überwacht und nach einer maximalen Wartezeit abgebrochen. Der Abbruch erfolgt über das Skript *kill.vbs* aus dem ELOcv Base/Script-Ordner. Der timeout-Parameter gibt die maximale Wartezeit in Sekunden an.
htmltopdf	Einige externe Konverter können nicht direkt PDF Dateien erzeugen. Fall sie in der Lage sind HTML Dateien zu generieren können Sie trotzdem vom COM Konverter verwendet werden. Er fügt nach der Konvertierung dann automatisch eine HTML-nach-PDF-Konvertierung an. Diese wird über den

	hier eingerichteten Konverter durchgeführt. Das Programm *wkhtmltopdf.exe* ist im Standard ZIP bereits enthalten und benötigt keine zusätzliche Installation.
htmloptions	Die HTML-nach-PDF-Konvertierung wird über das Kommandozeilenprogramm *wkhtlmtopdf* vorgenommen. Dieses bietet verschiedene Optionen, die man über diesen Parameter setzen kann. Die Optionen sind auf der Webseite des Open Source Projekts einsehbar.
runat	Da der ELO Preview Converter eine erhebliche Systemlast erzeugen kann, besteht die Möglichkeit sein Betrieb auf bestimmte Stunden einzuschränken. Hierfür legt man im Parameter *runat* eine Zeichenkette von 24 Nullen oder Einsen ab – für jede Stunde des Tages ein Eintrag. Eine 0 signalisiert keine Aktivität, eine 1 gibt den Konverter frei.
converter	Über diesen Parameter kann man bestimmen, welche Dokumententypen konvertiert werden sollen und welcher Konverter dafür verwendet werden soll. Im Prinzip gibt es nur drei Gruppen: 1. Konvertierung über Microsoft Office mittels COM-Konverter. 2. Konvertierung mittels der Aspose-Konverter 3. Konvertierung mittels des PDF-Printers Alle Varianten haben Ihre Vor- und Nachteile (siehe Anhang). Ein Mischbetrieb ist zulässig. Im converter-Parameter wird eine Liste der zu verwendeten Konverter festgelegt, diese werden jeweils durch einen Doppelpunkt getrennt. Manche Konverter können wiederum Parameter empfangen, die werden dann durch ein Fragezeichen vom Konverternamen getrennt. Converter1:Converter2:Converter3 Converter1?Params1:Converter2:Converter3?Params3 Die Parameterliste besteht aus einer Liste der zu bearbeitenden Extensions, jeweils durch einen Punkt getrennt (docx.xlsx.pptx). Wenn die Anwendung über die COM-Schnittstelle keine PDF-Dateien erzeugen kann dafür aber HTML-Dateien, dann besteht noch die Möglichkeit, dass man eine automatische HTML-nach-PDF-Konvertierung aktiviert. Diese Extensions müssen in der Parameterliste nach einem Ausrufezeichen aufgeführt werden. converter=ComConverter?docx.xlsx.pptx!msg Dieses Beispiel verwendet den ComConverter mit einer Konvertierung nach PDF für DOCX, XLSX- und PPTX-Dokumente sowie einer Konvertierung von

	MSG nach HTML mit nachgeschalteter HTML-nach-PDF-Konvertierung.
pdfprinter	Ausgabeverzeichnis des PDF Druckers.  **Achtung:** Der Konverter geht davon aus, dass er exklusiven Zugriff auf den PDF-Drucker besitzt. Dieser Drucker muss als Standarddrucker eingerichtet sein und kein anderes Programm darf ihn verwenden. Vor dem Start einer Konvertierung wird der Inhalt des **Druckerverzeichnisses komplett gelöscht!**

Wenn die Konvertierung über den COM-Konverter mittels Microsoft Office durchgeführt werden soll, muss vor der Inbetriebnahme ein geeignetes Office-Paket installiert werden. Benötigt werden Word (für DOC und DOCX), Excel (für XLS und XLSX), PowerPoint (für PPT und PPTX) sowie Outlook (für MSG). Nach der Installation muss jede Komponente einmal gestartet werden, um zu prüfen, ob alles funktioniert und um bei Bedarf Einstellungen vorzunehmen. Insbesondere der Start von Outlook ist wichtig damit ein Postfach eingerichtet werden kann. Eine Verbindung mit einem Mailserver ist nicht notwendig, es kann auch im Offline-Betrieb gearbeitet werden.

Bei der Verwendung von Outlook muss beachtet werden, dass dieser im Normalfall den Zugriff über die COM-Schnittstelle beschränkt. Diese Einschränkung kann durch passende Exchange-Einstellungen, Registry Einträge oder Third-Party-Produkte aufgehoben werden.

Der Script-Ordner enthält mehrere Skripte für den COM-Konverter. Hier muss man im Normalfall keine Änderungen vornehmen. Ein Eingriff ist nur notwendig, wenn man selbst weitere Konverterprogramme einbinden möchte.

2.1 Fenstertitelüberwachung

Während der Konvertierung kann es vorkommen, dass Microsoft Office oder andere Konverterprogramme Dialoge anzeigen, die manuell bestätigt werden müssen. Da hier keine Person sitzt, die das machen könnte, hängt der Prozess bis er durch ein Timeout abgebrochen wird. Man sollte deshalb die COM-Konverterskripte nach Möglichkeit so anlegen, dass diese Abfragen unterdrückt werden. Das ist aber nicht immer möglich. Deshalb besitzt der Konverter eine Zusatzfunktion, die den Titel des aktuell aktiven Fensters überwacht und bei Bedarf eine vordefinierte Tastenfolge sendet.

Da Windows ein neu gestartetes Programm nicht immer zum Top-Window macht, sollte man darauf achten, dass auf dem Desktop des Konverters kein aktives Programmfenster sichtbar ist. Alle laufenden Programme sollten minimiert sein. So ist die Wahrscheinlichkeit am höchsten, dass das störende Fenster zum Top-Window wird, da es das einzige sichtbare Fenster ist.

2.1.1 Beispiel für eine Titelüberwachung

Wenn man in Word eine Datei mit Passwortschutz öffnet, zeigt es ein Fenster mit dem Titel *Kennwort* an und erwartet die Eingabe des Passworts. Das kann der Konverter nicht liefern, aber zumindest kann man die Wartezeit bis zum Timeout ersparen, indem man einen Title-Watcher einrichtet, der auf den Fenstertitel *Kennwort* anspricht und den Tastendruck *Escape* sendet. Dieser bricht den Dialog ab, die Konvertierung wird sofort beendet.

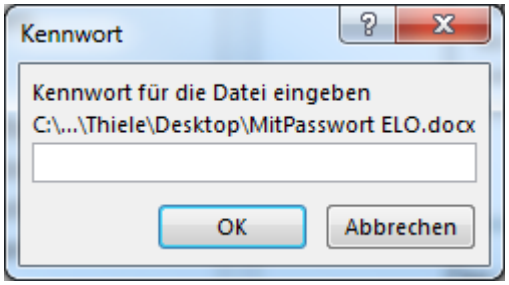


Abb. 3: Passwortabfrage

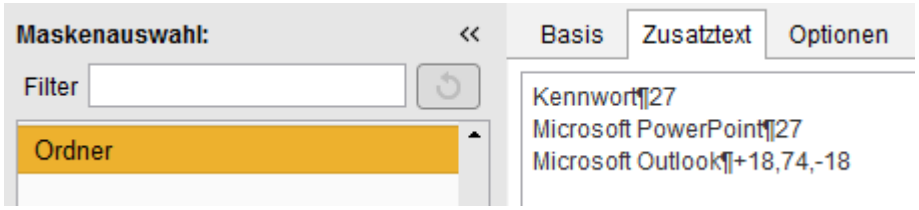


Abb. 4: Windows-KeyCodes für die gewünschten Tastenkombinationen

Eine ähnliche Situation ergibt sich bei PowerPoint. Wenn es mehrfach schnell hintereinander gestartet und beendet wird, kann es nach dem Beenden eine Fehlermeldung geben. Diese trägt den Fenstertitel *Microsoft Power Point* und kann ebenfalls mit *Escape* beendet werden.

Outlook gibt beim Speichern von signierten Mails eine Warnmeldung aus, die mittels *ALT-J* bestätigt werden kann.

Es ist zu bedenken, dass diese Erkennung nur funktioniert, wenn das Meldungsfenster zum aktiven Fenster wird. Sie kann also nicht 100% sicher arbeiten. In den anderen Fällen wird der Konverter durch die automatische Timeout-Erkennung den Vorgang abbrechen und mit dem nächsten Dokument weiterarbeiten.

Die Definition einer Titelüberwachung besteht aus einer Zeile im Zusatztext des Ordners *TitleWatcher*. Sie besteht aus zwei Teilen: dem regulären Ausdruck vor dem Pilcrow und der SendKeys-Folge nach dem Pilcrow.

Die Erkennung des Fenstertitels passiert über einen regulären Ausdruck. Im einfachsten Fall ist es ein statischer Text. Diesen kann man unter Beachtung der Sonderzeichen für reguläre Ausdrücke einfach eintragen. Wenn der Fenstertitel dynamische Anteile enthält, müssen diese durch einen passenden Ausdruck maskiert werden.

Die SendKeys-Folge wird als eine Folge von Zahlen, getrennt durch ein Komma definiert. Die Zahl ist der Windows-KeyCode der jeweiligen Taste. Wenn einfach eine Zahl eingetragen wird (z.B. 27), dann erzeugt der Konverter dafür einen Key-Down und Key-Up (in diesem Beispiel die Taste *Escape* wird gedrückt und losgelassen). Wird ein + vor die Zahl gestellt, wird nur ein Key-Down erzeugt, ein – vor der Zahl erzeugt nur ein Key-Up. Deshalb wird die Tastenfolge ALT-J durch +18 (ALT-Taste gedrückt) 74 (J Taste gedrückt und wieder losgelassen) -18 (ALT-Taste losgelassen) definiert.

3 Überwachung und Monitoring

Der ELO Preview Converter kann leicht manuell überwacht werden oder in ein vorhandenes Monitoring System eingebunden werden. Dafür stellt er eine Statusseite mit zwei Ausgabeformaten zur Verfügung.

Der Aufruf der normalen Statusseite kann durch einfachen Aufruf der cv-elo Seite erfolgen. Optional kann man noch den Parameter cmd=status anfügen.

http://localhost:8080/cv-elo

http://localhost:8080/cv-elo?cmd=xml

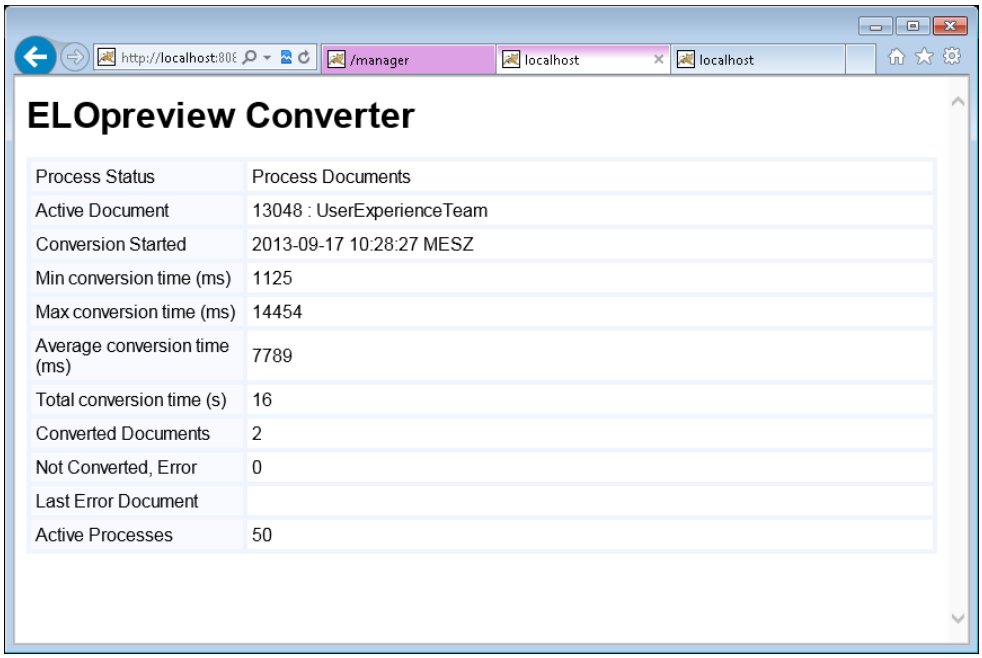


Abb. 5: Statusseite des ELO Preview Converters

Der Aufruf der Statusseite für den Monitoring Betrieb wird durch den Parameter cmd=xml durchgeführt. In diesem Fall wird die Statusseite im XML Format ausgegeben und kann somit leichter von Monitoring Systemen ausgewertet werden.

http://localhost:8080/cv-elo?cmd=xml

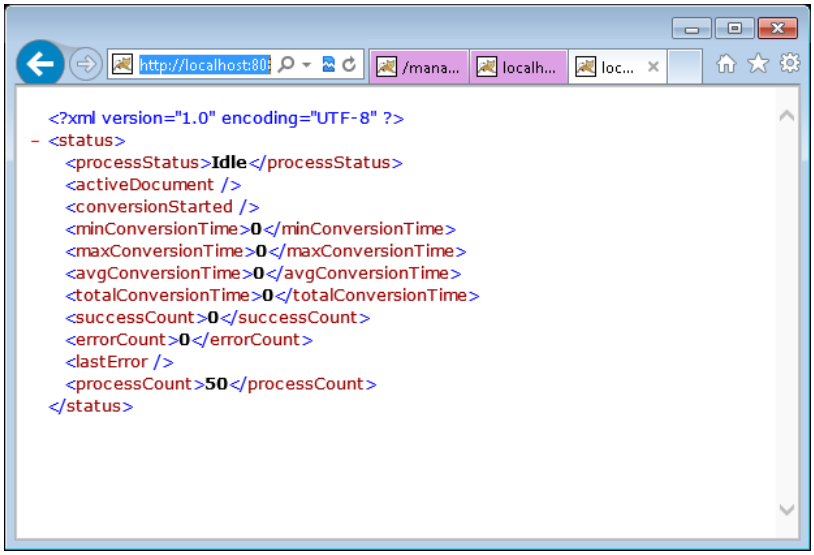


Abb. 6: Statusseite im XML-Format

Für das Monitoring sind vor allem folgende Einträge wichtig:

- conversionStarted: das ist der Startzeitpunkt der aktuellen Konvertierung. Wenn dieser zu weit in der Vergangenheit liegt, ist die Wahrscheinlichkeit groß, dass der Prozess hängt.
- errorCount: Anzahl der Fehlerdokumente bei der Konvertierung. Wenn ein Dokument nicht konvertiert werden kann, wird es in der Datenbank als *nicht konvertierbar* vermerkt. Wenn sich der error Count ändert, ist ein weiteres nicht konvertierbares Dokument aufgetreten.
- lastError: der Name des letzten fehlerhaften Dokuments.

Beachten Sie: Bei einem Neustart des Konverters, auch bei einem erzwungenen automatischen Neustart, werden alle Statistikwerte auf 0 zurückgesetzt.

Wenn bei der Konvertierung Fehler aufgetreten sind, werden die fehlerhaften Dokumente unterhalb der Statusliste ausgegeben (maximal 100). Die ELO Objekt-ID dieser Einträge ist anklickbar und man kann somit ein Dokument erneut zur Konvertierung anmelden. Wenn ein Konverterproblem durch eine Konfigurationsänderung behoben werden kann (z.B. durch einen längeren TimeOut oder einen passenden TitleWatcher-Eintrag), kann der Administrator über diese Funktion eine erneute Konvertierung auslösen.

Beachten Sie bitte, dass eine Konfigurationsänderung in Normalfall keinen Neustart des Dienstes erfordert (der diese Liste löschen würde). Die Konfiguration wird bei jedem Konverter-Durchlauf automatisch neu eingelesen.

4 Konvertierung in mehreren ELO Archiven

Der ELO Preview Converter kann ab der Version 12 für mehrere ELO Archive konfiguriert werden und führt automatisch die Konvertierung in den eingestellten Archiven durch. Die einzelnen ELO Archive werden in der Konfiguration (Datei „config.xml“) des ELO Preview Converters eingetragen. Jedes Archiv wird durch eine eindeutige Instanznummer spezifiziert.



Beachten Sie: Bei der Einstellung von mehreren Archiven haben die Konfigurationseinträge für jedes Archiv die entsprechende Instanznummer.

Eine Konfiguration für zwei ELO Archive sieht folgendermaßen aus:

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>parameters for this web application</comment>
<entry key="instanceId1">1</entry>
<entry key="url1">http://pcvladov:9090/ix-LocalArchive/ix</entry>
<entry key="user1">ELO Service</entry>
<entry key="password1">elo</entry>
<entry key="maxDocCount1">25</entry>

<entry key="instanceId2">2</entry>
<entry key="url2">http://testserver-elo11-1:9090/ix-elo110/ix</entry>
<entry key="user2">ELO Service</entry>
<entry key="password2">elo</entry>
<entry key="switchWaitTime">10000</entry>
</properties>
```

Der ELO Preview Converter beginnt mit dem ersten eingestellten ELO Archiv und konvertiert die eingestellte maximale Anzahl an Archivdokumenten. Der Konfigurationsparameter „maxDocCount“ definiert die maximale Anzahl an Archivdokumenten in einem Archiv, die bei einem Konvertierungsdurchlauf konvertiert werden. Die Eingabe dieses Parameters ist optional. Wenn dieser Konfigurationsparameter nicht vorhanden ist, hat der ELO Preview Converter keine Einschränkung für die maximale Anzahl der zu konvertierenden Archivdokumente.

Nach der Konvertierung der Archivdokumente im ersten Archiv wechselt der ELO Preview Converter zum nächsten Archiv. Nach dem Beenden der Konvertierung im letzten Archiv springt der ELO Preview Converter wieder zum ersten Archiv und führt die Konvertierung durch. Die Dauer des Archivwechsels (Wartezeit zwischen dem Wechsel zum nächsten Archiv) kann über den Parameter „switchWaitTime“ eingestellt werden. Die Eingabe der Wartezeit erfolgt in Millisekunden.

Auf der Statusseite sind alle konfigurierten Archive und deren Status zu sehen. Ein Archiv, in dem gerade die Konvertierung stattfindet, ist als aktiv gekennzeichnet. Ein Archiv hat den Status „Disconnected“, wenn sich der ELO Preview Converter in einem Wartezustand befindet und keine Indexserver-Verbindung vorhanden ist.

ELOpreview Converter	
Version	12.00.000 000
Archive Name	LocalArchive
Process Status	Disconnected
Active Document	
Conversion Started	
Min conversion time (ms)	0
Max conversion time (ms)	0
Average conversion time (ms)	0
Total conversion time (s)	0
Converted Documents	0
Not Converted, Error	0
Pending Redo List Size	0
Last Error Document	
Active Processes	126

ELOpreview Converter	
Version	12.00.000 000
Archive Name	Teamroom
Process Status	Process Documents - Download document
Active Document	120370 : Betr WG 12-er Skripting Seite
Conversion Started	2018-12-07 14:26:06 MEZ
Min conversion time (ms)	1152
Max conversion time (ms)	3737
Average conversion time (ms)	2260
Total conversion time (s)	10
Converted Documents	4
Not Converted, Error	2
Pending Redo List Size	0
Last Error Document	120372 : asdf
Active Processes	126

Abb. 7: Statusseite des ELO Preview Converters bei der Konvertierung in mehreren Archiven

5 Anhang

Der ELO Preview Converter kommt als vorinstalliertes ZIP-Paket. Im Folgenden werden Erstellung und Aufbau dieses Pakets beschrieben.

5.1 Teilpakete des ELO Preview Converters

Der ELO Preview Converter ist ein Servlet, welches wie alle anderen ELO Servlets in einem Tomcat-Container läuft, der in einer Java-Virtual-Machine betrieben wird.

Der Konverter besteht aus folgenden Teilpaketen:

Name	Funktion	Verzeichnis
Java Runtime	Dieses Paket beinhaltet die Java Runtime zum Betrieb des Tomcat Containers. Dabei handelt es sich um eine nicht-Installierte lokale Version nur für den ELO Preview Converter. Sie kann nicht von anderen Programmen verwendet werden und der ELO Preview Converter verwendet keine andere Runtime.	\Java
Application Server	Das ELO Preview Converter Servlet läuft in einem vorkonfigurierten Tomcat Application Server. Dieser enthält nur dieses eine Servlet und darf nicht um weitere eigene Funktionen ergänzt werden, da bei einem Timeout der komplette Prozess beendet und neu gestartet wird.	\Tomcat
Konfiguration	Das Standard ELO Konfigurationsverzeichnis enthält die Einstellungen für das Konvertierungsprogramm. Dieses wird, unabhängig vom Archivnamen, als *cv-elo* eingerichtet. Da innerhalb dieser Tomcat Instanz nur dieses eine Servlet aktiv ist, muss es zum Betrieb nicht umbenannt werden. Es enthält nur ein Unterverzeichnis *cv-elo* welches die Konfigurationsdatei enthält: *config.xml*: Standardkonfiguration des ELO Preview Converters. Diese muss vor dem Start angepasst werden. Sie enthält die Indexserver-URL, den ELO Anmeldenamen und das Passwort.	\config

Log-Dateien	Hier werden die Log-Dateien des ELO Preview Converters abgelegt.	\logs
Austauschverzeichnis	Das Austauschverzeichnis enthält verschiedene Dateien, die für den Betrieb des Konverters nur indirekt benötigt werden. • ELOcv Base.zip: ein Importdatensatz mit der Standardstruktur *ELOcv Base* im Archiv. Dieses kann mit dem Java Client importiert werden und anschließend bei Bedarf an die lokalen Anforderungen angepasst werden. • cv-elo.war: Webarchiv des Konverters. Wenn eine neue Version verfügbar wird, kann diese Version hier ausgetauscht werden und über die deployment-Funktion in der Tomcat Management Console aktualisiert werden. • cv-elo.xml: deployment descriptor für das Servlet. Er wird ebenfalls beim Update benötigt.	\download
HTML-nach-PDF-Konverter	Wenn ein Konverterprogramm nicht direkt PDF-Dateien erzeugen kann, dafür aber HTML-Ausgaben erzeugen kann, wird vom ELO Preview Converter automatisch eine HTML-nach-PDF-Konvertierung angefügt. Diese wird über das Programm *wkhtmltopdf* durchgeführt. Im Prinzip handelt es sich dabei um eine Kommandozeilenversion des Chrome-Browsers.	\wkhtmltopdf
Skript- und Batch-Dateien	Im Startverzeichnis gibt es noch eine Reihe von Skript- und Batch-Dateien welche für den Betrieb des ELO Preview Konverters benötigt werden. • autorestartpreview.vbs: wenn der Konverter über dieses Skript gestartet wird, prüft es alle 20 Sekunden ab, ob der Konverter noch ausgeführt wird. Falls der Konverter nicht in der Prozessliste gefunden wird, führt das Skript einen automatischen Neustart aus. Diese Startart sollte für den normalen Betrieb verwendet werden. Wenn man den	\

	Konverter beenden möchte, muss man zuerst den WSCRIP.EXE-Prozess stoppen, danach den ELOpreview.EXE-Prozess. • Start.bat: Startet den ELO Preview Converter ohne Automatischer-Neustart-Funktion. Dieser Start ist für den Test gedacht. • Killpreview.vbs: Beendet den Konverter. Dieser Aufruf wird auch aus dem Konverter heraus aufgerufen, wenn er sich in einem gestörten Zustand befindet. Wurde der Prozess mittels *autorestartpreview* aufgerufen, wird er hierüber automatisch wieder gestartet. • Restart.bat: Beendet den aktuellen Konverter und startet ihn neu. Diese Funktion ist für den Testbetrieb gedacht und darf nur aufgerufen werden, wenn der Konverter über *start.bat* gestartet wurde.	

5.2 Parameter der COM-Konverters

Der COM Konverter besitzt in der Konfiguration weitere Parameter zu den zu konvertierenden Dateitypen. Diese werden in der converter-Zeile mit einem Fragezeichen hinter dem Konvertiernamen angefügt.

converter=ComConverter?docx.xlsx.pptx.doc.xls.ppt!msg.eml

Diese zusätzlichen Parameter beinhalten eine Liste aller Extensions, die vom COM-Konverter bearbeitet werden sollen. Die Extensions sind jeweils durch einen Punkt getrennt. Der Teil der Liste vor dem Ausrufezeichen erzeugt dabei direkt PDF-Dateien. Der Teil hinter dem Ausrufezeichen erzeugt HTML-Dateien, die noch eine Nachkonvertierung nach PDF benötigen.

Der COM-Konverter führt für jedes Dokument einen Aufruf des Skriptes *compdf.vbs* durch. Dieses Skript hat für jede Extension einen Bereich, der die notwendigen COM-Aufrufe für die Konvertierung durchführt.

```
if (ext = "doc") or (ext = "docx") then
  set word = CreateObject("Word.Application")
  word.visible = false

  set docs = word.Documents
  set doc = docs.Open(fromFile)
```

```
  call doc.SaveAs(toFile, 17)
  set window = doc.ActiveWindow
  call window.Close()
end if
```

Das Skript erhält dabei als Parameter die Informationen

1. Name der originalen Dokumentendatei
2. Name der Zieldatei im PDF oder HTML Format
3. Name der Extension, klein geschrieben und ohne Punkt

Falls die Konvertierung hängen bleibt, wird der COM-Konverter nach der eingestellten timeout-Zeit diesen Prozess abbrechen. Das geschieht über den Aufruf des Skripts *kill.vbs*, welches nach dem Prozessnamen sucht (z.B. *EXCEL.EXE*) und alle Prozesse mit diesem Namen beendet.

Im Projekt können leicht weitere Konvertierungen hinzugefügt werden. Es muss nur ein geeignetes Konvertierungsprogramm zur Verfügung stehen und es müssen die Skripte *compdf.vbs* und *kill.vbs* um die Befehle für die zusätzliche Extension ergänzt werden.

5.3 Parameter des EXE Konverters

Für Programme, die keine Ansteuerung über die COM-Schnittstelle erlauben, dafür aber ein *Speichern als PDF* über Kommandozeilenoptionen ermöglichen, gibt es den EXE-Converter. Er ruft das externe Programm mit Parametern für Quell- und Zieldatei auf und speichert dann die erzeugte PDF-Datei im System ab.

Die Aktivierung des Konverters und Zuordnung der Extension erfolgt genauso wie bei den anderen Konvertern.

Beispiel: Microsoft Office Dokumente über den ComConverter, CAD Dateien über den ExeConverter verarbeiten:

```
converter=ComConverter?docx.xlsx.pptx.doc.xls.ppt:ExeConverter?dwg.dxf
```

Diese Einstellung teilt dem ELO Preview Converter mit, dass DWG- und DXF-Dateien an den Exe-Konverter übergeben werden sollen. Als nächstes ist nun noch eine Konfiguration der Kommandozeile für das externe Programm notwendig.

```
exe_dwg="C:\Program Files (x86)\Any DWG to PDF Converter Pro\dp.exe"
/InFile "%1" /OutFile "%2" /hide
```

Für jede verwendete Extension muss in der Konfiguration eine Zeile hinterlegt werden, die mit *exe_* beginnt und danach die gewünschte Extension aufführt. Nach dem Gleichheitszeichen folgt dann die Kommandozeile, welche die Parameter *%1* für den Quelldateinamen und *%2* für den Zieldateinamen (PDF) enthalten muss.

5.4 Parameter des JPEG-Konverters

Wenn im System sehr große JPEG-Dateien enthalten sind, für die eine kleinere Vorschau erstellt werden soll, kann man den JPEG-Konverter einsetzen. Hier kann man als Parameter die maximale Bildbreite oder Höhe einstellen. Wenn das Bild größer ist, wird es vor der PDF Erstellung entsprechen herunterskaliert.

```
converter=ComConverter?docx.xlsx.pptx.doc.xls.ppt!msg:JpegConverter?512
```

5.5 Parameter des Printer-Konverters

Wenn Dateien konvertiert werden müssen, für die es eine Windows Anwendung gibt, welche das Explorer Verb *Print* unterstützt, kann der PrintConverter eingesetzt werden. Dieser startet für die konfigurierten Datei-Extensions einen ShellExecute-Aufruf und wartet anschließend darauf, dass im konfigurierten PDF-Druckerverzeichnis eine PDF-Datei auftaucht. Diese wird dann als Preview-Datei gespeichert. Damit das Ausgabeverzeichnis überwacht werden kann, muss der Parameter *pdfprinter* entsprechend gesetzt werden.

```
pdfprinter=C:\ELOpreview\pdfprinter
converter=ComConverter?docx.xlsx.pptx:PrintConverter?txt
```

5.6 Anpassungen in den Standardpaketen Java und Tomcat

Das ZIP Installationspaket enthält bereits vorkonfigurierte Versionen der benötigten Java Runtime und des Tomcat Application Servers. Diese müssen im Normalfall nicht ausgetauscht werden. Falls im besonderen Fall doch mal ein Update durchgeführt werden muss, sollten folgende Anpassungen nachvollzogen werden.

Die Java Runtime muss nicht installiert werden. Insbesondere sollte sie nicht das Browser-Plug-in beinhalten, welches immer wieder ein Sicherheitsproblem darstellt und im ELO Umfeld nicht benötigt und nicht verwendet wird. Beim Austausch sollte das neue Paket direkt unter ELOpreview\Java liegen. Falls sich der Verzeichnisname mit der Java-Versionsnummer ändert (z.B. von jre7 auf jre8), muss anschließend im *start.bat* Batchprogramm die JAVA_HOME-Einstellung entsprechend angepasst werden.

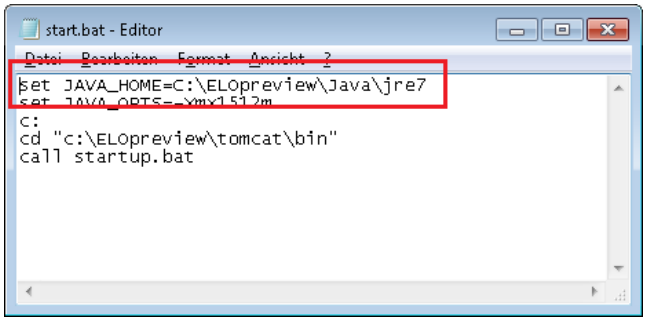


Abb. 8: Anpassung der JAVA_HOME-Einstellung

Damit der Konverter-Prozess in der Windows-Taskleiste nicht einfach nur unter dem Namen *Java* aufgeführt wird, muss als nächstes eine Kopie der Java.exe nach ELOpreview.exe durchgeführt werden.

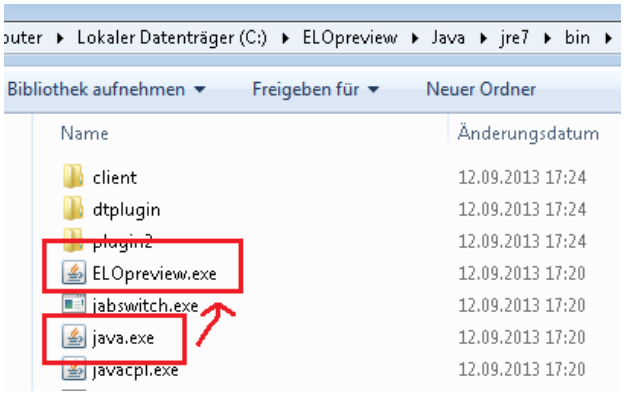


Abb. 9: Kopie der Datei 'Java.exe' auf' ELOpreview.exe'

Die Tomcat-Installation ist ebenfalls im Wesentlichen ein Standard-Tomcat aus dem ZIP-Archiv ohne Installation. Er soll nicht als Service laufen, sondern als ganz normaler Prozess auf dem Desktop. Hierdurch vermeidet man eine Vielzahl von Problemen, die sich einstellen, wenn Hintergrundprozesse mit Desktopanwendungen wie Microsoft Office kommunizieren wollen.

Die Standardinstallation wird in ein Verzeichnis ohne Versionsnummer durchgeführt damit nicht bei jedem Update eine Konfigurationsanpassung notwendig wird. Im Normalfall liegt er somit im Verzeichnispfad c:\ELOpreview\tomcat.

Damit der Tomcat-Server nicht mit *Java.exe* in der Taskliste auftaucht, muss die Datei *setclasspath.bat* im Verzeichnis c:\ELOpreview\tomcat\bin geändert werden. Es muss nur der Aufruf von Java nach ELOpreview angepasst werden. Da die Datei *ELOpreview.exe* im Java/bin-Verzeichnis eine Kopie der originalen *Java.exe* ist, sind keine weiteren Konfigurationsänderungen notwendig.

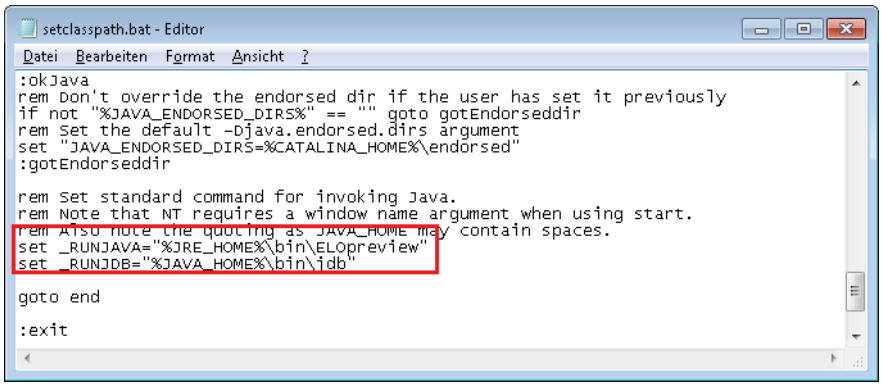


Abb. 10: Anpassung der Datei 'setclasspath.bat'

5.7 Bekannte Probleme

Bei der Dokumentenkonvertierung muss man immer zwischen zwei Übeln wählen: einer Konvertierung durch die Originalanwendung, welche für solche Aktionen nicht geschaffen ist, dafür aber die beste Qualität erzeugt. Oder einer Konvertierung durch eine Fremdbibliothek, welche die Dokumentenanzeige möglichst gut aber eigentlich nie perfekt abbilden kann. Der ELO Preview Converter überlässt diese Entscheidung dem Projektverantwortlichen indem für die meisten Formate mehrere Konverter zur Verfügung gestellt werden. Es muss jeweils im Einzelfall entschieden werden, welcher Weg beschritten werden soll.

5.7.1 COM-Konverter

Der COM-Konverter verwendet die Originalanwendung zur Konvertierung. Im Prinzip wird die Applikation gestartet, ein *Laden aus Datei* aufgerufen und ein *Speichern als PDF* oder *Speichern als HTML* schließt den Vorgang ab.

Dabei können verschiedene Komplikationen auftreten. Zum einen kann es vorkommen, dass die Applikation einen Anwenderdialog anzeigt, z.B. zur Eingabe von Passwörtern oder eine Abfrage nach Versionsänderungen. Dieser Dialog wird auf einer Serveranwendung natürlich nicht beantwortet und geschlossen werden, somit hängt der Prozess.

Dieses Problem wird vom ELO Preview Converter durch ein Timeout abgefangen. Wenn die Anwendung nicht innerhalb einer vorgegebenen Zeit abgeschlossen ist, wird sie über einen Betriebssystemaufruf beendet.

Weiterhin kann es vorkommen, dass der Prozess zwar die PDF-Datei erzeugt aber selbst danach nicht terminiert aber auch keinen Fehler meldet. Dieser Zustand kann vom Konverter nicht leicht erkannt werden. Deshalb bietet er auf der Statusseite eine Übersicht über die Anzahl der Prozess an. Der Administrator kann diesen Wert überwachen und bei einem starken Anwachsen eingreifen. Falls die Zahl unbemerkt so hochläuft, dass beim nächsten Prozessstart ein Fehler auftritt, werden allerdings alle Instanzen entfernt und das System läuft weiter.

5.7.2 Aspose-Konverter

Die Aspose-Konverter benötigen kein Microsoft Office. Allerdings unterliegen sie deshalb einer Reihe von Einschränkungen. Auch wenn die Konvertierungsqualität im Allgemeinen recht gut ist, kann man nicht im Allgemeinfall genau vorhersagen, welche Formatierungen oder Inhalte korrekt übernommen werden und welche nicht. Diese Aussage kann man nur im Projekt anhand einer repräsentativen Auswahl von echten Dokumenten treffen – und dort auch nur als Abschätzung.

Allgemein kann man nur sagen, dass die Aspose-Konverter natürlich keine aktiven Inhalte (Skripte) ausführen können. Zudem gibt es Einschränkungen bei der Auswahl der unterstützten Grafikformate. Word bietet eine sehr breite Palette von Möglichkeiten für eingebettete Grafiken. Welche tatsächlich verwendet werden und ob diese konvertiert werden können, muss im Test entschieden werden.

Bei Dokumenten mit sehr großen oder sehr vielen Grafiken stößt der Aspose-Konverter an seine Grenzen. In diesem Fall wird die Konvertierung sehr langsam oder kann gar komplett scheitern. Für diesen Fall gibt es im ELO Preview Converter einen Watchdog-Timer, der bei Bedarf den kompletten Prozess abbricht und neu startet.

5.7.3 Print-Konverter

Der Print-Konverter geht davon aus, dass der PDF-Drucker als Standarddrucker eingerichtet ist und NUR von ihm selbst angesteuert wird. Deshalb wird bei der Konvertierung auch die erste PDF-Datei gespeichert, die der Konverter im Ausgabeverzeichnis vorfindet.