**Automatisierte Uploads mit WinUpload**

**Pfaffhausen, 13. April 2010:** Mit dem Release 2010/IV gibt es auf der ArchivistaBox neu eine Versionierung. Wie diese aktiviert bzw. mit ihr gearbeitet wird, dies haben wir im **ersten Teil zum Release 2010/IV** aufgezeigt. Das Auschecken eines Dokumentes bereitet dabei kaum Schwierigkeiten. Und auch das Hochladen einer geänderten Datei ist an sich einfach. Folgende Optionen stehen zur Auswahl:

a) Upload der geänderten Datei per WebClient
b) Upload über FTP-Order (/office/dbname)
c) Upload über API-Schnittstelle

Gerade die letzte Variante ist sehr mächtig, sie erfordert aber ein kleines Skript (als Beispiel hier mit unserer Hausspache Perl):

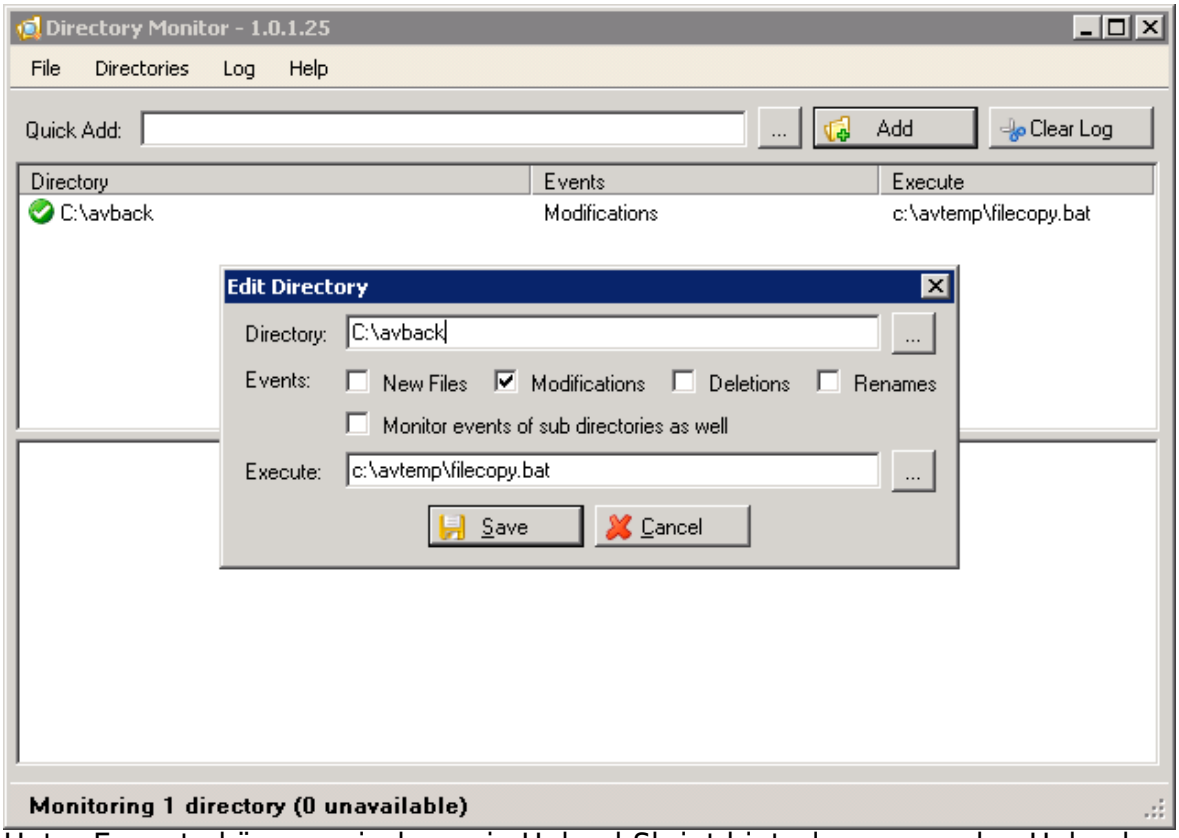
```
#!/usr/bin/perl
my $fout = shift; # get here the filename
use strict;
use LWP::UserAgent; # we work with UserAgent (our batch web browser)
use HTTP::Request::Common qw(POST); # the post method must be imported
my $server = "http://192.168.0.220/perl/avclient/index.pl";
my $www = LWP::UserAgent->new; # new www session
my $res = $www->request(POST "$server",
  Content_Type => 'form-data', # multipart/form-data
  Content => [ # structure for our file
  MAX_FILE_SIZE => 134217728, # max. size (WebClient won't accept more)
  upload => [ $fout, $fout ], # file to upload, file name to use
  go => 'go_action', # we need to call the go_action command
  action => 'upload', # inside of go_action we need to use upload
  uploaddef => 0, # scan def
  host => 'localhost', # connection information
  db => 'archivista',
  uid => 'Admin',
  pwd => 'archivista',
  frm_Laufnummer => 1,
])
```

Damit wir uns richtig verstehen, diese 20 Zeilen Code laden ein Dokument über unser API in den WebClient hoch. Diese Zeilen können genauso gut mit einer jeden anderen Programmiersprache realisiert werden. Allerdings, werden Sie einwenden wollen, ein normale/r Benutzer/in mag sich mit solchen "Kleinigkeiten" nicht herumschlagen. Auch hier gilt, einer unserer Kunde wollte einen automatisierten Upload der Dateien direkt aus einem Windows-Verzeichnis heraus.

Als Vorbemerkung sei angemerkt, dass gängige Office-Pakete problemlos auf einen FTP-Ordner Dateien speichern können. Es gibt zudem auch Programme, die einen beliebigen FTP-Ordner als Laufwerk bzw. Pfad einbinden können. Eine weitere Option kann darin bestehen, ein Verzeichnis zu überwachen, und sobald eine geänderte Datei eintrifft, ein Upload-Programm zu starten. Eine **Übersicht über solche Programme finden Sie hier.**

**DirectoryMonitor überwacht Verzeichnisse**

Mein Favorit aus der Liste wäre **Directory Monitor (getestet in Version 1.0.25).** Wichtig beim Einrichten war, die Aktion auf 'Modifications' zu aktivieren.




Unter Execute können wir dann ein Upload-Skript hinterlegen, um den Upload auszuführen. Im konkreten Falle haben wir das nachfolgende Skript filecopy.bat verwendet:

```
cd c:avtemp@echo offecho user ftp> ftpcmd1.datecho ftp>>
ftpcmd1.datecho cd office>> ftpcmd1.datecho cd archivista>>
ftpcmd1.datecho bin>> ftpcmd1.datecho put %1>>
ftpcmd1.datecho quit>> ftpcmd1.datPING -n 3 127.0.0.1>nulftp
-n -s:ftpcmd1.dat 192.168.0.220PING -n 1 127.0.0.1>nuldel
ftpcmd1.datdel %1
```

DriveMonitor stellt dabei sicher, dass nach dem Ändern einer Datei in einem Ordner der Upload der Datei automatisch vorgenommen wird. Grundsätzlich funktioniert dieses Verfahren einwandfrei. Allerdings gibt es zwei Dinge zu beachten. Erstens erfordert DriveMonitor eine neuere Dot.Net-Umgebung. Die Applikation selber ist zwar beschaulich klein, die Dot.Net-Umgebung dagegen erschreckend gross (und ohne einen Neustart geht gar nichts). Zweitens ist DriveMonitor eine Freeware-Applikation. D.h. Sie dürfen die Applikation kostenfrei verwenden, erhalten aber z.B. die Sourcen nicht und von daher kann ich keine Aussagen machen wie stabil bzw. gut DriveMonitor wirklich ist.

**Lösung mit WinUpload und WinCopy**

Die obenstehende Lösung hat noch einen anderen Nachteil. Beim FTP-Upload wird eine Text-Datei erstellt, um den Upload über den internen FTP-Client (ftp.exe) von Windows durchzuführen. Dabei werden Benutzerdaten bzw. Passwörter im Klartext gespeichert (sowohl in der Batch-Datei als in der Steuerungsdatei). Zudem muss filecopy.bat (unser Upload-Programm) jeweils von Hand angepasst werden.

Aus diesen Gründen haben wir uns entschlossen zwei neue Hilfsprogramme zu erstellen. Es handelt sich dabei um die Programme **winupload.pl** bzw. **wincopy.pl.** Diese zwei Programme stehen ab **Release 2010/IV im Ordner /home/cvs/archivista/jobs** zur Verfügung. Die beiden Perl-Programme laufen plattformunabhängig (getestet unter Linux und Windows, auf Mac mangels

Rechner nicht überprüft). Unter Linux wird Perl wohl bereits zur Verfügung stehen, unter Windows dürfte das eher nicht der Fall sein. Aus diesem Grunde haben wir die Programme kompilliert, d.h. Sie können die beiden Hilfsprogramme auch als EXE-Datei beziehen. Sie finden die beiden Programme (gezippt) hier:
**www.archivista.ch/de/media/version/winupload.zip**
**www.archivista.ch/de/media/version/wincopy.zip**

**WinUpload zum Hochladen von Dateien**

Mit dem ersten Program (winupload) können Dateien bequem hochgeladen werden. Ein Aufruf ohne Parameter gibt einen kurzen Hilfetext aus:

```
C:avuploadwinupload.exe v1.0 (c) 2010 by Archivista GmbH,
upload files to WebClientThis program has three modes, a)
single, b) config and c) uploada) single mode:
C:avuploadwinupload.exe host db user password def https file
[meta]b) config mode: C:avuploadwinupload.exe host db user
password [def=0..x https=0/1]c) upload mode:
C:avuploadwinupload.exe file
[fieldname1=value1:fieldname2=value2..]
```

Daraus können wir entnehmen, dass wir winupload in drei Varianten aufrufen können. Im Single-Modus geben wir sämtliche Parameter mit an und der Upload der Datei erfolgt direkt (ohne Konfiguration). Im Config-Modus können wir die Uplaod-Parameter so hinterlegen, dass wir anschliessend nur noch den Dateinamen angeben können. Im Upload-Modus schliesslich geben wir nur die Datei und allfällige Meta-Informationen mit, um eine Datei hochzuladen. Die Verbindungsoptionen werden aus der Datei winupload.dat ausgelesen. Diese Datei liegt nicht im Klartext vor, doch sei hier auch gesagt, dass falls Sie eine qualitative hochstehende Verschlüsslung benötigen, die vorliegende Verion 1.0 von WinUpload diesen Ansprüchen nicht genügt. Nachfolgend die drei Modi mit je einem Beispiel:
Single-Modus: winupload 192.168.0.220 archivista Admin archivista 0 0 c:14282d.pdf
Config-Modus: winupload 192.168.0.220 archivista Admin archivista 0 0
Upload-Modus: winupload c:14282d.pdf Titel=Hallo

**WinCopy zum Überprüfen eines Verzeichnisses**

Bevor wir eine Datei hochladen können, müssen wir ein Verzeichnis überwachen. Dafür ist das zweite Programm wincopy.exe gedacht. Sie können das Programm ohne Parameter starten und erhalten dann eine Auflistung der Optionen:

```
C:avuploadwincopy.exe v1.0 (c) 2010 by Archivista GmbH,
watch a dir and call upload prgC:avuploadwincopy.exe dirin
dirtemp seconds (to wait between checks for uploads)
[upload]   => [upload] optional, without it call
winupload.pl or winupload.exe   => you can stop the program
if you cre
ate a file wincopy.stp
```

**Achtung: Wir emfpehlen sehr, das Programm einzig innerhalb eines leeren Programmordners zu starten. Wird wincopy.exe mit einem gültigen Verzeichnis gestartet, so werden umgehend sämtliche Dateien ins Temporärverzeichnis verschoben und anschliessend wird versucht, diese Dateien hochzuladen.**

Doch zurück zu den Parametern. An erster Stelle muss das zu überwachende Verzeichnis stehen, an zweiter Stelle folgt ein Temporärverzeichnis, um zu testen, ob eine Datei die hochgeladen werden soll, nicht mehr von einer Applikation "gesperrt" wird. In diesem Falle könnte der Upload unter Umständen nicht korrekt erfolgen. An dritter Stelle kann eine Pausenzeit in Sekunden

eingeben werden (3-60 Sekunden). In diesem Intervall wird das an erster Stelle angegebene Verzeichnis überprüft.

**Komplettes Beispiel zum Abschluss**

a) Erstellen Sie den Ordner `'c:avupload'`
b) Kopieren Sie die Dateien `'winupload.exe'` und `'wincopy.exe'` in dieses Verzeichnis
c) Starten Sie wincopy.exe 192.168.0.100 archivista Admin archivista 0 0 (bzw. andere Anmeldeoptionen)
d) Überprüfen Sie im Verzeichnis `'c:avupload'`, ob die Datei `'winupload.dat'` besteht.
e) Überpürfen Sie, ob das Verzeichnis `'c:temp'` besteht (ansonsten erstellen!).
f) Starten Sie `'wincopy.exe .'` (mit dem Punkt am Schluss verwenden Sie das aktuelle Verzeichnis)
g) Speichern Sie eine Office-Datei ins Verzeichnis `'c:avupload'.` Sie wird umgehend hochgeladen.

**Hinweis: Punkt g) funktioniert nur, wenn die Datei nicht mehr vom Verarbeitungsprogramm gesperrt wird. Sie werden daher in der Regel die Datei nach dem 'Speichern' auch noch 'Schliessen' müssen.**

Selbstverständlich werden Sie das Hilfsprogramm 'wincopy.exe .' beim Hochstarten bzw. über einen Dienst anwerfen wollen. Und natürlich dürfen Sie die Perl-Programme winupload.pl und wincopy.pl beliebig an ihre eigenen Bedürfnisse anpassen, d.h. sofern Sie den Quellcode auch wieder unter die GPL-Lizenz stellen.