

Anwendungsspezifikation **arcunTAR v.4.2** Softwarelösung für die Benutzung des **arcun**-Speichers

Inhaltsverzeichnis

1	Zusammenfassung.....	1
2	Einleitung anhand eines Anwendungsbeispiels	2
3	Installationsanleitung.....	4
4	arcunTAR starten	5
5	Implementierungsdetails	6
5.1	Zugriffsschlüssel	6
5.2	Lokale Persistenz	7
5.3	Das Programm <i>arcunTAR</i>	7
5.4	Die einzelnen Optionen	8
5.5	Konfigurationsdatei.....	9
6	Datenstruktur XMP-Datei – Datenbank	13
6.1	Datenstruktur XMP-Datei	13
6.2	Datenbank Struktur	14
7	Webserver <i>arcunREST</i>	15
8	Change LOG	17
8.1	Version 3.6	17
8.2	Version 3.7	17
8.3	Version 3.8 und 3.9	18
8.4	Version 4.0	18
8.5	Version 4.1	18
8.6	Version 4.2	18
9	Index / Übersicht der verschiedenen Optionen	19

Veränderungen und Ergänzungen zu vorherigen Programm-Versionen von **arcunTAR** sind unter Change LOG aufgeführt

1 Zusammenfassung

Die Softwarelösung **arcunTAR** ist in der Lage, einzelne Dateien oder ganze Verzeichnisbäume in einen ZIP64¹ Container zu verpacken und auf dem **arcun**-Speicher abzulegen. Analog dazu kann unter Angabe eines Dateipfades oder einer *accessID* die Sicherheitskopie wieder auf einen lokalen Rechner zurückge-

¹ Das originale ZIP-Format (Version 4.5 der Spezifikation) weist etliche Grössenbeschränkungen auf, unter anderem ist die Dateigrösse auf max. 4GB beschränkt. PKWARE hat deshalb die "ZIP64" Formaterweiterung eingebracht, welche diese Begrenzungen aufhebt. Es existieren dafür bereits mehrere Implementierungen. Siehe dazu [http://en.wikipedia.org/wiki/ZIP_\(file_format\)](http://en.wikipedia.org/wiki/ZIP_(file_format))

holt werden. Die Zugriffsinformationen (*access/D* und ursprünglicher Speicherort²) werden in XMP-Dateien und/oder in einer Datenbank gespeichert.

Damit ist eine einfache Lösung gegeben, digitales Archivgut von einem lokalen Fileserver oder Rechner aus auf die Speicherplattform von FAST LTA-CH1 zu sichern. Die einzelnen Arbeitsschritte werden dabei vom Archivinformatiker manuell ausgeführt oder in einer Batch Datei³ automatisiert. Soll die *access/D* oder der Dateipfad als Zugriffsinformation im Findmittel gespeichert werden, kann dies von Hand oder mit entsprechender Datenbank-Programmierung gemacht werden.

arcunTAR ist eine von mehreren Lösungen für die Bewirtschaftung der LTA-CH1 Speicherplattform innerhalb des KOST-Projekts **arcun**.

2 Einleitung anhand eines Anwendungsbeispiels

Gestützt auf eine Umfrage bei den beteiligten Archiven können wir folgendes einfaches Anwendungsbeispiel skizzieren:

Die archivierten digitalen Daten werden nur auf ihrem originalen Speichermedium (CD oder DVD) vorgehalten. Der Zugriff im Archiv erfolgt über das Findmittel/Katalog mit Signatur und Lagerort des Speichermediums. Bei Bedarf wird die CD/DVD physisch aus dem Magazin geholt und im Lesesaal auf einem Rechner eingelesen. Mit **arcunTAR** sollen nun von den Daten auf diesen Speichermedien Kopien auf dem **arcun**-Speicher angelegt werden. Die archivinterne Bewirtschaftung erfolgt weiterhin über das Speichermedium CD/DVD, das heißt der Zugriffspfad zum **arcun**-Speicher wird nicht ins Findmittel eingetragen.

In diesem (fiktiven) Beispiel soll eine Sicherheitskopie einer CD auf dem **arcun**-Speicher mit **arcunTAR** angelegt werden:

Auf einem lokalen Rechner, USB-Festplatte, NAS⁴, oder Dateiserver wird ein Verzeichnis mit dem Namen von Bestand und Serie angelegt, unter denen sich das digitale Archivgut befindet und analog dazu wie sie im Findmittel eingetragen sind, z.B.

E:\ELAR\Audio-Sammlung\

In diesem Verzeichnis wird der Inhalt der entsprechenden CD oder ein ISO-Abbild⁵ des Datenträgers abgelegt. Im Falle einer Audio-CD wird ein Verzeichnis mit WAVE-Dateien angelegt. Als Verzeichnisname oder als Name des ISO-Abbildes wird die Signatur verwendet.

² *Speicherort* ist Pfad zusammen mit Dateiname ohne Laufwerksbuchstabe oder Name des Fileservers; das deshalb, weil Laufwerksbuchstabe aber auch Servername nur temporäre Zuordnungen zu einem Speicherort sind.

³ Batchverarbeitung oder auch Stapelverarbeitung bezeichnet die sequentielle, nicht-interaktive Bearbeitung von Aufgaben.

⁴ NAS (*Network Attached Storage*) bezeichnet einen einfach zu verwaltende Dateiserver.

⁵ ISO-Abbild (*engl. ISO image*) ist die Bezeichnung für eine Computer-Datei, die ein Speicherabbild des Inhalts einer CD oder DVD enthält, die im Format ISO 9660 strukturiert ist. Das Dateisystem wird bei der Erstellung eines ISO-Abbildes unverändert kopiert, dadurch bleiben Berechtigungen und andere Metadaten unverändert vorhanden. Die Norm ISO 9660 definiert ein Dateisystem für CDs bzw. DVDs. Zit. Wikipedia <http://de.wikipedia.org/wiki/ISO-Abbild>

```
E:\ELAR\Audio-Sammlung\UR-CD12br1999.iso
E:\ELAR\Audio-Sammlung\UR-CD12br1999\
```

Wenn die Daten-CD als ISO-Abbild gespeichert werden soll muss zuerst ein ISO-Abbild erstellt werden. Die Audio-CD muss in Audio Dateien umgewandelt werden, das heisst die einzelnen Titel/Tonspuren (*engl. Tracks*) auf der CD müssen in WAVE Dateien umgewandelt werden. Eine entsprechende Anleitung ist als KOST-Service verfügbar unter http://kost-ceco.ch/cms/index.php?service_audio-cd_de.

Das ISO-Abbild kann nun mit **arcunTAR** wie folgt auf FAST LTA-CH1, den Datenspeicher, übertragen werden:

```
arcunTAR -c -f 54881199937753545850 E:\ELAR\Audio-Sammlung\UR-CD12br1999.iso
```

Eine Audio Sammlung oder Datenstruktur im Verzeichnis UR-CD12br1999 wird wie folgt übertragen:

```
arcunTAR -c -f 54881199937753545850 E:\ELAR\Audio-Sammlung\UR-CD12br1999
```

Die Syntax folgt dem UNIX TAR Befehl. *-c* steht für *create*⁶ und bedeutet, eine neue Kopie anzulegen. *-f* steht für *archive file* bzw. *device*, womit das Speicherkonto auf LTA-CH1 spezifiziert wird. Der Servername oder Laufwerksbuchstabe **E:** wird entfernt, der Speicherort in LTA-CH1 ist *ELAR\Audio-Sammlung\UR-CD12br1999.iso*.

```
arcunTAR -x -f 54881199937753545850 E:\ELAR\Audio-Sammlung\UR-CD12br1999.iso
```

So wird die Datei UR-CD12br1999.iso wieder aus LTA-CH1 zurückkopiert; *-x* bedeutet *extract*. Der neue Speicherort ist das aktuelle Arbeitsverzeichnis, darin wird *\ELAR\Audio-Sammlung\UR-CD12br1999.iso* angelegt. Ist das aktuelle Arbeitsverzeichnis **E:**, wird die Datei an den ursprünglichen Speicherort zurückkopiert.

```
arcunTAR -t -f 54881199937753545850 E:\ELAR\Audio-Sammlung\UR-CD12br1999\
```

Die Option *-t* (für *list*) zeigt alle Dateien und Verzeichnisse in und unterhalb des spezifizierten Speicherortes (hier *.\UR-CD12br1999*) an. Wenn sich also drei Audiodateien im WAVE-Format im Verzeichnis *ELAR\Audio-Sammlung\UR-CD12br1999* befinden, erhalten wir folgendes Listing:

```
arcunTAR -t -f 54881199937753545850 E:\ELAR\Audio-Sammlung\UR-CD12br1999\
ELAR\Audio-Sammlung\UR-CD12br1999\
ELAR\Audio-Sammlung\UR-CD12br1999\track01.wav
ELAR\Audio-Sammlung\UR-CD12br1999\track02.wav
ELAR\Audio-Sammlung\UR-CD12br1999\track03.wav
```

Soll die gesamte ELAR Datensammlung übertragen werden, lautet der **arcunTAR** Befehl (für die Automatisierung stehen auch LOOP und INTERVAL Befehle zur Verfügung):

```
arcunTAR -c -f 54881199937753545850 E:\ELAR
```

Die Option *-l* (*--compare*) überprüft die Konsistenz der übertragenen Daten (sind alle Dateien in **N:ELAR** übertragen und lokal unverändert:

⁶ Vergleichbar zu *-c* ist die Option *-u*, *update*, welche das Anlegen einer neuen Kopie auf der Speicherplattform erzwingt; siehe dazu weiter unten die Beschreibung der einzelnen Optionen.

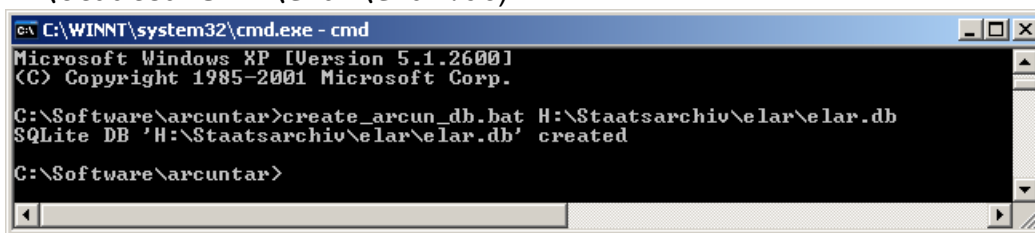
```
arcunTAR -l -f 54881199937753545850 E:\ELAR
```

3 Installationsanleitung

Die Installation von arcunTAR besteht darin, das Zip zu entpacken, eine Datenbank (da werden die Persistenzinformationen lokal abgelegt) anzulegen und das Erstellen einer Konfigurationsdatei pro Account mit dem entsprechenden Pfad der Persistenz-Datenbank.

Die einzelnen Schritte:

1. Den gesamten Inhalt der ZIP Datei **arcunTAR.zip** in ein geeignetes Programm-Verzeichnis auf dem Rechner kopieren. (im Beispiel `C:\Software\arcuntar\`)
2. Mit der Script Datei **create_arcun_db.bat** an einem geeigneten Ort die Persistenz-Datenbank mit beliebigem Namen anlegen (im Beispiel `H:\Staatsarchiv\elar\elar.db`)

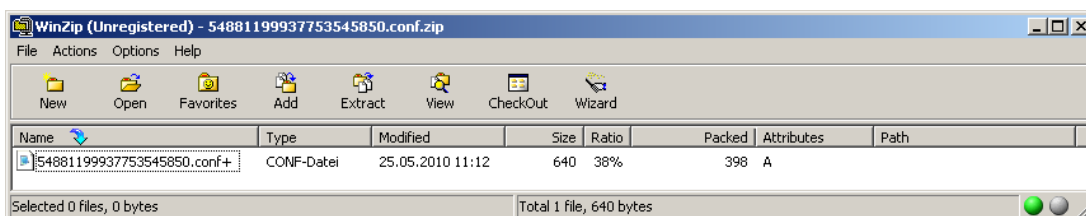


```
C:\WINNT\system32\cmd.exe - cmd
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Software\arcuntar>create_arcun_db.bat H:\Staatsarchiv\elar\elar.db
SQLite DB 'H:\Staatsarchiv\elar\elar.db' created

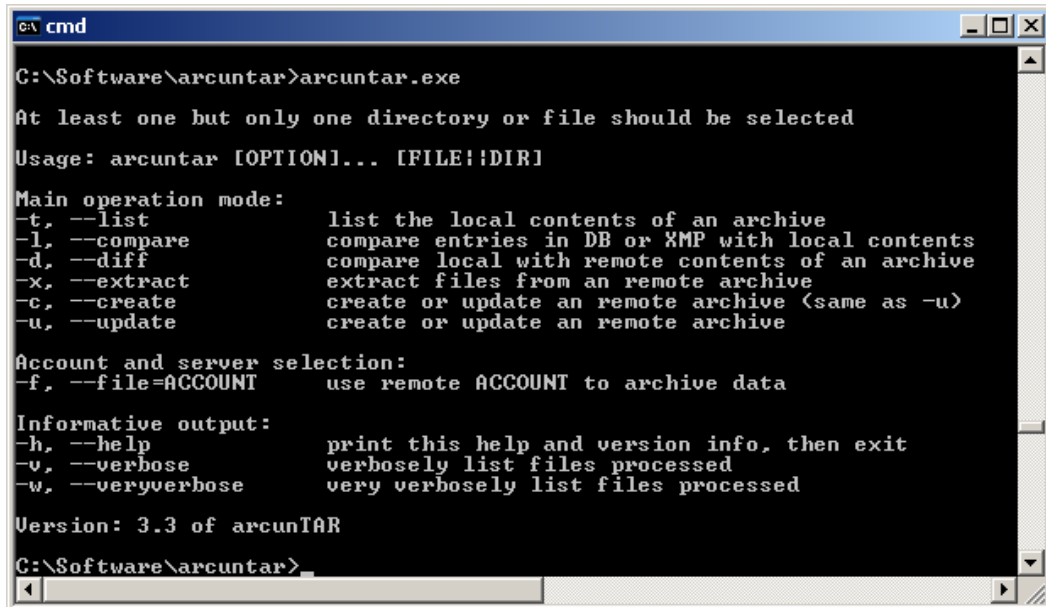
C:\Software\arcuntar>
```

3. Eine Konfigurationsdatei '**[remote_account].conf**' erstellen (siehe beigelegtes Beispiel `54881199937753545850.conf` für das **arcun** Testkonto respektive Kapitel 5.5 Konfigurationsdatei)
4. Die Konfigurationsdatei in eine ZIP-Datei ohne weitere Verzeichnisstruktur umwandeln und gegebenenfalls die ZIP-Datei mit einem Passwort schützen (Das Passwort für die Testdatei ist '123'). Die Konfigurationsdatei muss sich am selben Ort wie die Programmdatei **arcunTAR.exe** befinden.



4 arcunTAR starten

Das Programm `arcunTAR.exe` in einem `command` Fenster aufrufen. Wenn sich das Programm nicht im Arbeitsverzeichnis befindet mit Pfadnamen aufrufen (wie im Beispiel unten `c:\Software\arcuntar\arcunTAR.exe`) und nicht via `%PATH%` starten.



```
c:\ cmd

C:\Software\arcuntar>arcuntar.exe

At least one but only one directory or file should be selected

Usage: arcuntar [OPTION]... [FILE!;DIR]

Main operation mode:
-t, --list           list the local contents of an archive
-l, --compare       compare entries in DB or XMP with local contents
-d, --diff          compare local with remote contents of an archive
-x, --extract       extract files from an remote archive
-c, --create        create or update an remote archive (same as -u)
-u, --update        create or update an remote archive

Account and server selection:
-f, --file=ACCOUNT use remote ACCOUNT to archive data

Informative output:
-h, --help         print this help and version info, then exit
-v, --verbose      verbosely list files processed
-w, --veryverbose  very verbosely list files processed

Version: 3.3 of arcunTAR

C:\Software\arcuntar>
```

Ein Beispiel:

Acht Dateien aus dem Verzeichnis `c:\tmp\arcun_123` werden im Speicherkonto `54881199937753545850` archiviert:

```
C:\Software\arcuntar\arcunTAR.exe -cvf 54881199937753545850 C:\tmp\arcun_123
```

Anschliessend wird ein Listing der bereits archivierten Dateien angefordert:

```
C:\Software\arcuntar\arcunTAR.exe -tf 54881199937753545850 C:\tmp\arcun_123
```

Der letzte Aufruf überprüft und vergleicht die auf der Speicherplattform archivierten Dateien mit den lokalen Dateien:

```
C:\Software\arcuntar\arcunTAR.exe -dvf 54881199937753545850 C:\tmp\arcun_123
```


5.2 Lokale Persistenz

Die beiden Zugriffsschlüssel, *accessID* und Pfad/Dateiname, werden auf dem **arcun**-Speicher in Form des LTA-CH1 Speichernamens (*accessID*) und der Metainformation in der ZIP-Datei festgehalten. Die ZIP-Datei enthält neben Pfad und Dateinamen auch *creation-date* und *modification-date* der Datei. Damit der Zugriff auf die Sicherungskopien sowohl via *accessID* als auch via Dateinamen und Dateipfad erfolgen kann (meist sogar nur via Dateinamen und Dateipfad) muss diese Information sinnvoller Weise auch lokal im Archiv vorhanden sein. **arcunTAR** bietet zwei Möglichkeiten, diese Information lokal zu speichern: in einer XMP-Metadatei⁷ und/oder in einer Datenbank. Die XMP-Metadatei bzw. die Datenbank enthält alle aus dem Filesystem des Speicherortes automatisch extrahierbaren Informationen (Server, Verzeichnispfad, Dateiname, *ctime*, *mtime*, *owner*, *mimetype* etc.) zusammen mit der LTA-CH1 *accessID* und dem Speicherzeitpunkt. Die Datenstruktur der XMP-Datei, beziehungsweise der Datenbankeintrag ist weiter unten beschrieben⁸.

Eine XMP-Datei wird mit dem Dateinamen *filename.xmp* am gleichen Ort wie die Originaldatei gespeichert. Für die XMP-Datei, welche als XML-Datei auch mit einem gängigen Browser betrachtet werden kann, ist ein entsprechendes Schema unter <http://kost-ceco.ch/arcun/arcun.xsd> abgelegt.

Als Datenbank ist per Default eine SQLite⁹ Datenbank vorgesehen.

5.3 Das Programm **arcunTAR**

arcunTAR bietet analog zum im UNIX-Umfeld sehr bekannten Archivierungsprogramm **tar** die Möglichkeit, Dateien und Verzeichnisbäume sequenziell in ein (entferntes) Archiv zu schreiben bzw. diese Dateien daraus wieder herzustellen. Das Programm folgt so weitgehend wie möglich der Syntax von **tar**. Das heisst, als Argumente werden zuerst die Optionen übergeben (siehe unten), dann die Datei(en). Diese können durch den vollständigen Pfad zu einem Verzeichnis oder einer einzelnen Datei identifiziert werden. Dabei kann anstelle des Laufwerkbuchstabens auch ein Servername (mit *//* beginnend) oder ein doppelter Schrägstrich */* als Indikator für die Dateisystem Root verwendet werden. Wenn als Pfad nur */* angegeben wird, werden sämtliche Dateien im aktuellen Arbeitsverzeichnis ausgewählt.

In Abweichung von **tar** wird das *archive file* bzw. *device* durch das Konto auf dem Speichersystem ersetzt. Alle zusätzlich für die Programmkonfiguration und Übermittlung notwendigen Parameter werden in einer (optional verschlüsselten)

⁷ Die Extensible Metadata Platform (XMP) ist ein Standard, um Metadaten in digitale Medien einzubetten. Der Standard wurde von Adobe im Jahr 2001 veröffentlicht. Der Standard steht mit Spezifikationen und einem SDK unter einer Open-Source-Lizenz zur Verfügung http://de.wikipedia.org/wiki/Extensible_Metadata_Platform <http://www.adobe.com/products/xmp/indepth.html>

⁸ XMP Datei und Datenbank Record sind weitgehend analog aufgebaut.

⁹ SQLite implementiert in einer Programmbibliothek ein relationales Datenbanksystem. Die Datenbank selber liegt in einer Datei, ein Datenbank-Serverprozess ist nicht notwendig. Siehe www.sqlite.org.

Konfigurationsdatei aufbewahrt. Der Dateiname der Konfigurationsdatei entspricht dem Konto, so können mehrere Konten nebeneinander benutzt werden.

Das Passwort für die Konfigurationsdatei wird bei Programmstart abgefragt.

Folgende Anwendungsarten und Programmparameter stehen zur Verfügung:

```
arcunTAR --help
Usage: arcuntar [OPTION]... [FILE|DIR]
Main operation mode:
-c, --create          create or update an remote archive (same as -u)
-u, --update          create or update an remote archive
-t, --list            list the local contents of an archive
-l, --compare         compare entries in DB or XMP with local contents
-d, --diff            compare local with remote contents of an archive
-x, --extract         extract files from an remote archive
Account and server selection:
-f, --file=ACCOUNT   use remote ACCOUNT to archive data
Informative output:
-h, --help            print this help and version info, then exit
-v, --verbose         verbosely list files processed
-w, --veryverbose     very verbosely list files processed
```

5.4 Die einzelnen Optionen

Main operation mode:

`-c, --create` create or update an remote archive (same as `-u`)

Sämtliche Dateien im als Argument übergebenen Verzeichnis (inkl. Unterverzeichnisse) werden mit ihrem vollständigen Pfad auf dem **arcun**-Speicher gespeichert. Es kann im Gegensatz zu **tar** beim Programmaufruf nur ein Verzeichnis gewählt werden.

Für diese und alle weiteren Optionen im **Main operation mode** gilt: Das entsprechende **arcun**-Konto wird mit der Option `-f` identifiziert (siehe dazu unten). Als Argument kann der vollständige Pfad zu einem Verzeichnis oder zu einer Datei angegeben werden. Dabei kann anstelle des Laufwerkbuchstabens auch ein Servername (mit `//` beginnend) oder ein doppelter Schrägstrich `/` als Indikator für die Dateisystem Root verwendet werden (unter MS-Windows steht `/` für das aktuelle Laufwerk). Wenn als Pfad nur `/` angegeben wird, werden sämtliche Dateien ausgewählt.

`-u, --update` create or update an remote archive

Entspricht der Option `-c, --create`. Da identische Dateien mit gleichem Pfad und Dateinamen nur einmal auf der Speicherplattform LTA-CH1 abgelegt werden ist ein Update nur zu Testzwecken sinnvoll. Zum Vergleichen von lokalen Dateibeständen mit den Daten auf der Speicherplattform dient die Option `-d, --diff`

`-t, --list` list the local contents of an archive

Eine Liste aller Dateien (mit Pfad und Dateinamen) unter dem angegebenen Pfad im entsprechenden Konto auf dem **arcun**-Speicher wird ausgegeben. Zusammen mit der Option `-v, --verbose` wird auch *modification-time*, *size* und *access/ID* angezeigt. Diese Option wird ohne Zugriff auf die LTA-CH1 Speicherplattform ausgeführt; die angezeigten Informationen stammen aus der Datenbank oder den XMP Dateien.

-l, --compare compare entries in DB or XMP with local contents
Analog zu **-t, --list** wird eine Liste aller Dateien (mit Pfad und Dateinamen) ab dem angegebenen Pfad im entsprechenden Konto auf dem **arcun**-Speicher ausgegeben. Zusätzlich werden die lokalen Dateien via md5-Hash mit dem entsprechenden Eintrag in der Datenbank oder den XMP Dateien verglichen. Auch diese Option wird ohne Zugriff auf die LTA-CH1 Speicherplattform ausgeführt.

-d, --diff compare local with remote contents of an archive
Alle Dateien (mit Pfad und Dateinamen) unter dem angegebenen Pfad im entsprechenden Konto auf dem **arcun**-Speicher werden mit den auf der LTA-CH1 Speicherplattform Dateien verglichen. Dazu werden die Dateien von der Speicherplattform heruntergeladen, in einem temporären Verzeichnis gespeichert, verglichen und anschliessend wieder gelöscht.

Falls eine Datei lokal fehlt, wird sie am entsprechenden Ort wieder hergestellt (wie mit der Option **-x**).

Unterscheiden sich die lokale Datei und die Datei auf der **arcun** Speicherplattform wird das mit einer Meldung angezeigt und die herunter geladene Datei im temporären Verzeichnis belassen.

-x, --extract extract files from an remote archive

Alle Dateien ab dem angegebenen Pfad im entsprechenden Konto auf dem **arcun**-Speicher werden unter ihrem archivierten Pfad lokal gespeichert. Der Pfad wird im aktuellen Arbeitsverzeichnis angelegt. Wird das entsprechende Root Verzeichnis oder der entsprechende Laufwerksbuchstabe gewählt werden die Dateien nur mit den bereits vorhandenen verglichen, d.h. lokal vorhandene Dateien können nicht versehentlich überschrieben werden.

Account and server selection:

-f, --file=ACCOUNT use remote ACCOUNT to archive data

Angabe des Kontos auf der Speicherplattform LTA-CH1, auf dem die Sicherungskopie erstellt werden soll. Das Passwort für das Konto ist in der gleichnamigen Konfigurationsdatei gespeichert. Der Kontoname ist gleichzeitig auch der Name der Konfigurationsdatei, wo weiter Optionen für die Übertragung zu einem bestimmten Konto spezifiziert sind (siehe Abschnitt: Konfigurationsdatei).

Informative output:

-h, --help print this help and version info, then exit

Zeigt Hilfetext und Programmversion an

-v, --verbose verbosely list files processed

Pro übertragene Datei wird Pfad, Dateiname und accessID ausgegeben. Die Ausgabe wird auch in die LOG Datei geschrieben.

-w, --veryverbose very verbosely list files processed

Diese Optionen erlaubt, die Übertragung im Detail nachzuverfolgen, und ist eher für Testzwecke gedacht.

5.5 Konfigurationsdatei

Die Syntax der Konfigurationsdatei orientiert sich an der Syntax von Microsoft **robocopy.exe**. Der Name der Konfigurationsdatei lautet **'remote_account'.conf.zip**, also zum Beispiel: **54881199937753545850.conf.zip**. Dabei handelt es sich um eine Zip.Datei, welche eine Konfigurationsdatei mit selbem Namen enthält (im Beispiel **54881199937753545850.conf**).

P:\KOST\Pilotloesungen\arcun\Ausschreibung_2(5Partner)\9_arcunLösung\1_workbench\arcunTAR_v4.2.doc

Die Konfigurationsdatei kann mit einem ZIP Passwort vor unberechtigtem Zugriff geschützt werden, das Passwort muss in diesem Fall beim Start von **arcunTAR** interaktiv eingegeben werden.

Beispiel einer typischen Konfigurationsdatei:

```
:: Transfer Options :
/LTA:54881199937753545850:k145ST1@my.fotobank.de
/PROXY:www-proxy.archiv.ch:8080
/C
/ENC:mypasswd

:: Synchronize Options :
/DB:H:\Staatsarchiv\elar\elar.db
/LOG+:H:\Staatsarchiv\log\elar.log
```

Zwei Doppelpunkte leiten einen Kommentar ein. Leerzeilen werden ignoriert.

Hier eine Liste aller möglichen Konfigurationsoptionen:

```
:: Transfer Options :
/LTA:user:password@server
      :: LTA-CH1 server (https://my.fotobank.de) user (20 digits) password
/PROXY:[user:pass@]server:port
      :: Example: /PROXY:muster:mypwd@www-proxy.archiv.ch:8080
/TIMEOUT:sec :: Connection timeout: default 30 seconds
/TMP[:dir] :: Use dir instead of temp directory for ZIP file
/LOCALDIR :: Use local path instead of temp directory for ZIP file
      :: (/LOCALDIR:TIMESTAMP timestamp in path on LTA-CH1 server)
/C :: Compress ZIP file
/ENC:passwd :: Encrypt ZIP file with password
      :: Password is stored in plain text in XMP or DB
/HYDE :: Do not store password in XMP or DB

:: Synchronize Options :
/ANYWAY :: Transfer sourcefile despite already existing entry
/DB:path :: Path and filename of existing SQLite DB to be used
/XMP :: Write XMP metadata file ('sourcefile.xmp')
/LIST :: List all entries in DB or XMP persistence
/COMP :: Compare entries in DB or XMP with local source
/NOCHECKSUM :: Do not calculate checksum for local file entries
      :: only check for filename, filesize and creationdate
/RETRIEVE[://server/destination]
      :: Retrieve sourcefile from LTA-CH1 and compare with local source
      :: download will be restored in optional destination path

:: Retry Options :
/NMI:n :: Number of errors before shutdown: default 10
/W:n :: Wait time between retries: default 30 seconds
/LOOP :: Loop upload until terminated with ctrl-c
/INTERVAL:hh:mm-hh.mm :: Loop upload only in dedicated time range: from-to

:: Logging Options :
/LOG:file :: Output to LOG file (default: no LOG file)
/LOG+:file :: Appends to existing LOG file
/DEBUG:level :: Debug level: message = 1, full backtrace = 2

:: REST Server Options :
/REST:server:port
      :: Example: /REST:sv0029.archiv.ch:8080: default localhost:8080
/ROOT:[dir] :: Use dir instead of temp directory as sever root

:: Information :
/V :: Produce verbose output
/VV :: Produce very verbose output
/VER :: Version 4.1-49 of arcun
/? :: Display this help file
```

Die einzelnen Optionen

:: Transfer Options :

/LTA:user:password@server

LTA-CH1 server (<https://my.fotobank.de>) Benutzerkonto (20 digits) und Passwort z.B. **/LTA:54881199937753545850:k145ST1@my.fotobank.de**

/PROXY:[user:pass@]server:port

Die notwendigen Angaben zum Proxy Server. Der Proxy Server muss die HTML Funktion **CONNECT** unterstützen. Beispiel **/PROXY:muster:mypwd@www-proxy.archiv.ch:8080**

/TIMEOUT:sec Connection timeout: default 30 seconds seconds

/TIMEOUT erlaubt den Wert für *Connection Timeout* in der Konfigurationsdatei zu setzen, für den Fall, dass der *default* Wert 30 Sekunden nicht hinreichend ist. Minimaler Wert ist 15 Sekunden.

/TMP[:dir] Use dir instead of temp directory for ZIP file

Mit der Option **/TMP** kann ein anderes Verzeichnis als das Windows Temporärverzeichnis für die Übertragung gewählt werden. Achtung, das Verzeichnis muss leer und es muss genügend Speicherplatz vorhanden sein.

/LOCALDIR Use local path instead of temp directory for ZIP file

Hier wird das zu archivierende Verzeichnis als Temporärverzeichnis gewählt. Im Gigabank GUI¹⁰ ist dann dieses Verzeichnis mit den archivierten Dateien sichtbar. Einschränkung: es muss Schreibrecht und genügend Speicherplatz auf dem lokal zu archivierende Verzeichnis vorhanden sein und es sind nur die Zeichen a-z, A-Z, 0-9 und _ in Datei- und Verzeichnisnamen erlaubt.

/C Compress ZIP file

Die auf dem Speichersystem abgelegte ZIP Datei wird komprimiert.

/ENC:passwd Encrypt ZIP file with password

Die auf dem Speichersystem abgelegte ZIP Datei wird verschlüsselt (die Übertragung erfolgt ohnehin verschlüsselt). Die Stärke der Verschlüsselung hängt von der Schlüssellänge ab. Der Schlüssel wird in der Datenbank und in der XMP Datei lokal gespeichert. Mit der Option **/HYDE** kann dieses Verhalten so geändert werden, dass der Schlüssel nirgends gespeichert wird.

/HYDE Do not store password in XMP or DB

Das Passwort für die Verschlüsselung der ZIP Datei wird nicht lokal gespeichert. Es wird einzig die Tatsache der Verschlüsselung festgehalten. (zur Schreibweise, siehe Dr. Jekyll und Mr. Hyde).

:: Synchronize Options:

/DB:path Path and filename of existing SQLite DB to be used

Die Persistenzinformationen sollen alternativ zu **xmp** Dateien lokal in einer Datenbank abgelegt werden. Im Augenblick ist die einzige implementierte Datenbank eine **sqlite** Datenbank¹¹. Die Datenbank mit dem notwendigen Daten-

¹⁰ Zu jedem Account auf der Speicherplattform gibt es auch ein Web GUI unter <https://www.gigabank.de/>.

¹¹ Das Programm `sqlite3.exe` erlaubt den Inhalt der Datenbank zu betrachten, siehe dazu `sqlite3.exe --help`

bankschema und der Tabelle **XMP** können mit dem Script `create_arcun_db.bat` angelegt werden. Diese Datenbank liegt mit Vorteil nicht innerhalb des zu sichernden lokalen Bereiches und muss selber regelmässig gesichert¹² werden.

`/XMP` Write XMP metadata file ('sourcefile.xmp')

Es soll eine lokale XMP Datei mit dem Namen `filename.xmp` zu jeder übertragenen Datei angelegt werden.

`/NOCHECKSUM` Do not calculate checksum for local file entries

Beim Überprüfen ob eine Datei schon übertragen ist, wird statt auf die Checksumme auf Dateigrösse und letztes Speicherdatum von Datei bzw. Verzeichnis abgestellt. Das ist wesentlich schneller, aber die Dateien werden dabei nicht gelesen, d.h. es findet kein *data scrubbing* statt.

:: **Retry Options:**

`/NMI:n` Number of errors before shutdown: default 10

`/NMI` legt fest, nach wie vielen Warnungen das Programm beendet werden soll, default ist 10. `/NMI` ohne Zähler bedeutet: kein Programmabbruch.

`/LOOP` Loop upload until terminated with ctrl-c

Mit der Option `/LOOP` wird **arcunTAR** fortlaufend mit den gleichen Argumenten ausgeführt. Eine kleine Pause unterbricht die einzelnen Loops. Das Passwort für die Konfigurationsdatei muss nur einmal eingegeben werden.

`/INTERVAL:hh:mm-hh.mm` Loop upload only in dedicated time range: from-to
Mit der Option `/INTERVAL:hh:mm-hh.mm` wird **arcunTAR** fortlaufend aber nur im ausgewählten Intervall, z.B. Nachts oder in Randzeiten ausgeführt. Diese Option ist nützlich, wenn Programm zur Überwachung eines bestimmten Ordners automatisch über eine Batch Datei in Windows *Startup* oder *AutoStart* gestartet werden soll.

:: **Logging Options:**

`/LOG:file` Output to LOG file (default: no LOG file)

Eine LOG Datei wird geschrieben. Mit der Option `/LOG+:file` wird eine bestehende LOG Datei ergänzt. Nur mit der Option `/v` oder `-v` bei Programmstart wird für jede übertragene Datei eine LOG-Zeile geschrieben.

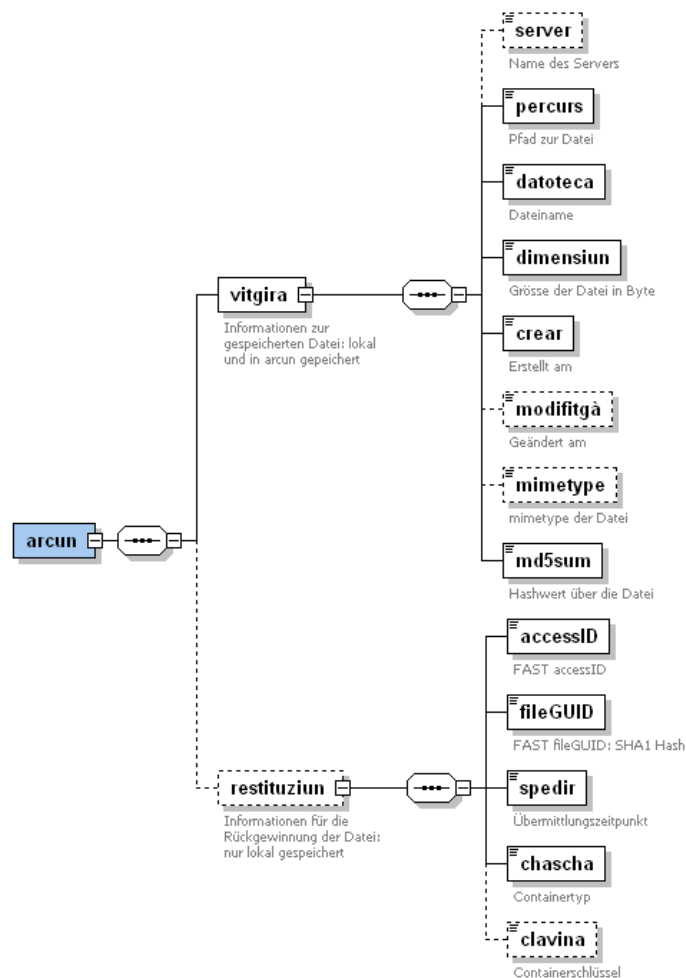
¹² Bei Verlust der Datenbank und, falls vorhanden, der XMP Dateien sind die lokalen Informationen über die auf der Speicherplattform bereits gespeicherten Daten verloren. Sie müssen dann alle mit entsprechendem Zeitaufwand wieder zurück gelesen werden.

6 Datenstruktur XMP-Datei – Datenbank

XMP Datei und Datenbank Record sind weitgehend analog aufgebaut. Die Element-, bzw. Feldnamen sind dem Rätoromanischen entnommen. Hier die Übersetzung:

<i>vitgira:</i>	Informationen zur gespeicherten Datei (lokal und in arcun gespeichert)
<i>server:</i>	Name des Servers
<i>percur:</i>	Pfad zur Datei
<i>datoteca:</i>	Dateiname
<i>dimensiun:</i>	Grösse der Datei in Byte
<i>crear:</i>	erstellt am
<i>modifitgà:</i>	geändert am
<i>mimetype:</i>	mimetype der Datei
<i>md5sum:</i>	Hashwert über die Datei
<i>restituziun:</i>	Informationen für die Rückgewinnung der Datei (nur lokal gespeichert)
<i>accessed:</i>	FAST accessID
<i>fileGUID:</i>	FAST fileGUID (sha1 Hash)
<i>spedir:</i>	Übermittlungszeitpunkt
<i>chascha:</i>	Containertyp; zip, jar, tar
<i>clavina:</i>	Containerschlüssel

6.1 Datenstruktur XMP-Datei



6.2 Datenbank Struktur

Hier das entsprechende *CREATE TABLE* Statement für die Datenbank:

```
CREATE TABLE xmp (  
    percurs VARCHAR(258),  
    datoteca VARCHAR(260),  
    dimensiun LONG, crear DATE,  
    modifitga DATE,  
    mimetype VARCHAR(80),  
    md5sum CHAR(32),  
    accessID CHAR(80),  
    fileGUID CHAR(40), spedir DATE,  
    chascha VARCHAR(4),  
    clavina VARCHAR(128),  
    path VARCHAR(520),  
    server VARCHAR(80)  
);  
CREATE INDEX idx_percurs ON xmp(percurs);  
CREATE INDEX idx_datoteca ON xmp(datoteca);  
CREATE INDEX idx_spedir ON xmp(spedir);  
CREATE INDEX idx_path ON xmp(path);
```


7 Webserver *arcunREST*

Mit *arcunREST* steht eine Webserver basierte Lösung zum Browsen der lokalen *arcun* Datenbank zur Verfügung. *arcunREST* startet einen HTTP Server, mit dem sich die in einem *arcun* Konto gespeicherten Dateien wie auf einem Webserver betrachten lassen. Dabei wird die ursprüngliche Speichersituation der Dateien und Verzeichnisse im lokalen Betriebssystem über das Lesen der lokalen Datenbank als REST (*Representational State Transfer*)¹³ Server abgebildet. Wird via Webserver auf eine lokal nicht mehr vorhandene Datei zugegriffen, wird diese Datei mit *arcunTAR* in das Temporärverzeichnis kopiert und angezeigt. Mit der Option `/REST` kann Server Name und Server Port festgelegt werden.

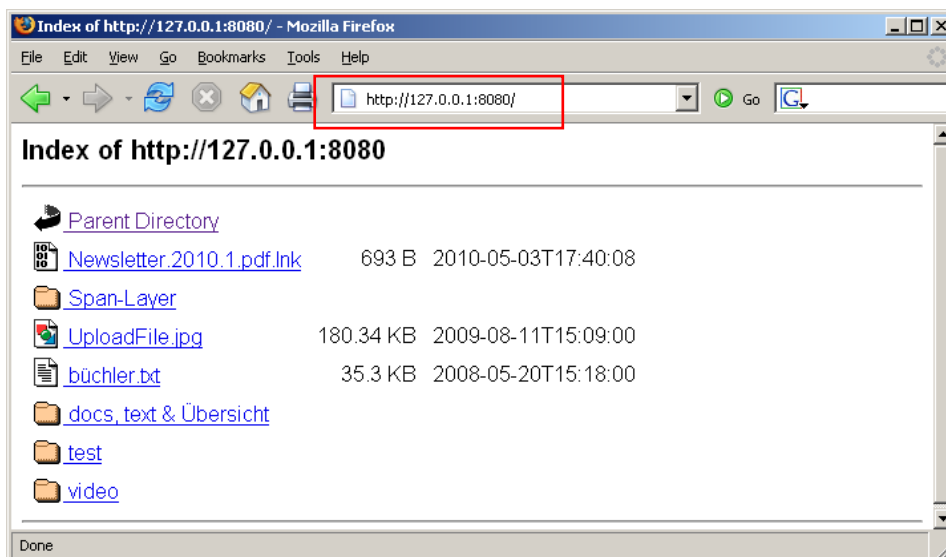
arcunREST lässt sich gleichermassen wie *arcunTAR* starten. Als Parameter stehen `-r -v -f` zur Verfügung. `-r` bedeutet REST Server starten, `-v -f` wie gewohnt. Wird also mit folgendem Commandozeilen-Befehl

```
arcunTAR -c -f 54881199937753545850 E:\ELAR
```

alle Dateien im Verzeichnis `E:\ELAR` inklusive Unterverzeichnisse in das *arcun* Konto `54881199937753545850` archiviert, zeigt

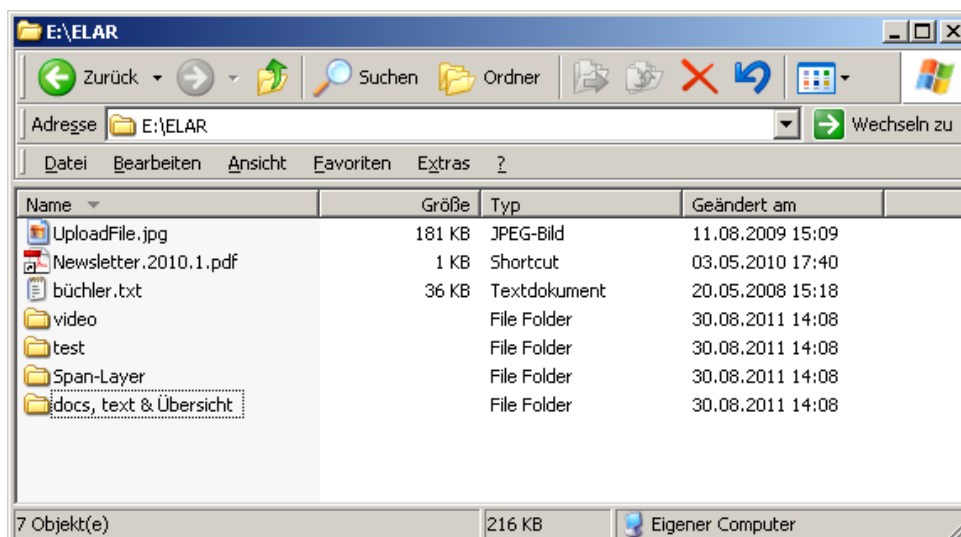
```
arcunREST -r -f 54881199937753545850 E:\ELAR
***** miniHTTPServer started *****
Copyright (C) 2009 by KOST-CECO
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
for details.see source code miniHTTPServer.php
Server on 127.0.0.1:8080
arcun root /ELAR
Document root C:/DOCUME~1/u1942/LOCALS~1/Temp/ELAR
```

eben diese Dateien und Verzeichnisse wieder via HTTP Server. Server Adresse ist wie bei Programmstart angezeigt `http://127.0.0.1:8080/`



¹³ Vereinfacht ausgedrückt bildet das REST Konzept Daten-(Bank) Inhalte in einfachen URLs ab, siehe dazu: http://de.wikipedia.org/wiki/Representational_State_Transfer

Das gleiche Verzeichnis im Microsoft Explorer Fenster.



Nach dem Löschen von **E:\ELAR** kann weiterhin mit dem Browser die gespeicherte Verzeichnisstruktur betrachtet und Dateien angezeigt werden. Dateien werden von *arcunREST* im Windows Temporärverzeichnis (hier in der Document Root bezeichnet) `c:\DOCUME~1\u1942\LOCALS~1\Temp\ELAR` mit der ursprünglichen Verzeichnisstruktur abgelegt. Dateien, welche bereits in der Document Root liegen werden nicht erneut aus dem *arcun* Speicher kopiert.

Das mit *arcun Root* bezeichnete Verzeichnis, ist das beim Start von *arcunREST* spezifizierte Verzeichnis und wählt den im Webserver sichtbaren Teil des Speicherkonto Inhaltes. Wird `/` angegeben, werden sämtliche Dateien ausgewählt.

Mit der Konfigurationsoption `/root` kann als *Document Root* ein anderes Verzeichnis als das Windows Temporärverzeichnis gewählt werden. Wird als *Document Root* gerade das archivierte Verzeichnis gewählt und die Dateien darin nicht gelöscht, funktioniert *arcunREST* wie ein normaler Webserver. Dabei ist zu beachten, dass der `/root` Parameter zusammen mit der *arcun Root* die *Document Root* des Webservers ergibt.

Folgende Anwendungsarten und Programmparameter stehen zur Verfügung:

```
arcunREST --help
Usage: arcunREST [OPTION]... [FILE|DIR]

Main operation mode:
-r, --restserver      start REST Server and list the local contents of an archive
                      extract requestet files from remote archive

Account and server selection:
-f, --file=ACCOUNT   use remote ACCOUNT to archive data

Informative output:
-h, --help           print this help and version info, then exit
-v, --verbose        verbosely list files processed

Version: 4.1-42 of arcunREST:
```

Zusätzliche REST Server Konfigurationsoptionen:

```
:: REST Server Options :  
/REST:server:port  
    :: Example: /REST:sv0029.archiv.ch:8080: default localhost:8080  
/ROOT:[dir]  :: Use dir instead of temp directory as sever root
```

8 Change LOG

8.1 Version 3.6

Die Version 3.6 behebt einen Fehler in der `/INTERVAL` Funktion von **arcun-TAR v3.5** (waren oder sind bereits alle Dateien übertragen, wurde das Intervall nicht beendet sondern im `/LOOP` Mode fortgefahren).

Ein neues Script, erlaubt die **arcun** SQLite-Datenbank in eine XMP-Datei umzuwandeln. Die Datenbank wird dabei auf keine Art verändert. Die XMP Datei enthält für jede übertragene Datei einen `<arcun>` Knoten. Zum Ausführen die Dateien `sqlite3.exe`, `arcun_db2xmp.bat` und `arcun_db2xmp.sql` in den selben Ordner kopieren und die Batch Datei mit Pfad und Dateiname der **arcun** Datenbank als Argument aufrufen. Die XMP Datei wird im Arbeitsverzeichnis angelegt. (zum Beispiel: `arcun_db2xmp.bat H:\Staatsarchiv\elar\elar.db`)

8.2 Version 3.7

In der Version 3.7 sind zwei neue Optionen eingebaut und zwei Fehler behoben. Die Datenbank und die XMP Struktur sind durch die Neuerungen nicht betroffen und können unverändert beibehalten werden.

Behobene Fehler

Die Sonderzeichen `&` und `'` in Datei- und Pfadnamen werden nun richtig behandelt.

Bei Datei- und Pfadnamen wird die gross/klein Schreibung genau so wie im Dateisystem angegeben übertragen, obwohl MS-Windows keine Unterscheidung zwischen Gross- und Kleinschreibung bei Dateinamen macht. Zum Beispiel wird eine Datei `Upload.jpg` im Dateisystem wird unter diesem Namen auf dem **arcun**-Speicher gespeichert, auch wenn sie beim Übertragen mit **arcunTAR** als `upload.jpg` angesprochen wurde. Sie wird auch unter dem korrekten Namen zurückgeholt.

Neue Konfigurationsoptionen

```
/DEBUG:level :: Debug level: message = 1, full backtrace = 2  
/DEBUG erlaubt ein ausführlicheres Debugging mit Backtrace. Level 1 nur mit Funktions-Backtrace, bei Level 2 wird der ganze PHP Process Stack angezeigt.
```

```
/W:n :: Wait time between retries: default 30 seconds
```

Diese **wait time** beim lesen der Datenbank-Datei behebt Probleme die beim Backup der Datenbank-Datei durch kurzzeitige nicht Verfügbarkeit eine Netzlaufwerkes entstehen können. `/w:n` legt fest, wie lange auf die Datenbank bis zu einem erneuten Verbindungsversuch gewartet wird. `/nmi:n` legt fest, wie oft ein Verbindungsversuch unternommen wird, bis das Programm abbrechen soll.

8.3 Version 3.8 und 3.9

- Verbesserte Mime-Type Erkennung, insbesondere bei Video Dateien mit GnuWin32 file-4.26.
- Programmstart aus beliebigem Verzeichnis.
- Leer- und Sonderzeichen in Datei- und Pfadnamen werden beim schreiben der XMP Dateien richtig behandelt.

8.4 Version 4.0

Teil dieser Version ist eine neuere FAST-LTA Library (**LTA Middleware version: 1.82**)

`/TIMEOUT:sec :: Connection timeout: default 30 seconds seconds`
`/TIMEOUT` erlaubt den Wert für *Connection Timeout* in der Konfigurationsdatei zu setzen, für den Fall, dass der *default* Wert 30 Sekunden nicht hinreichend ist. Minimaler Wert ist 15 Sekunden.

8.5 Version 4.1

arcunREST erlaubt das browsen eines **arcun** Speicherkontos mit einem Web-Browser. Siehe weiter oben Kapitel *Webserver arcunREST*.

8.6 Version 4.2

`/NOCHECKSUM Do not calculate checksum for local file entries`
Beim Überprüfen ob eine Datei schon übertragen ist, wird statt auf die Checksumme auf Dateigrösse und letztes Speicherdatum von Datei bzw. Verzeichnis abgestellt. Das ist wesentlich schneller, aber die Dateien werden dabei nicht gelesen, d.h. es findet kein *data scrubbing* statt.

9 Index / Übersicht der verschiedenen Optionen

<i>Main operation mode:</i>	8
-c, --create	create or update an remote archive (same as -u)8
-u, --update	create or update an remote archive8
-t, --list	list the local contents of an archive8
-l, --compare	compare entries in DB or XMP with local contents ...9
-d, --diff	compare local with remote contents of an archive ...9
-x, --extract	extract files from an remote archive9
<i>Account and server selection:</i>	9
-f, --file=ACCOUNT	use remote ACCOUNT to archive data9
<i>Informative output:</i>	9
-h, --help	print this help and version info, then exit9
-v, --verbose	verbosely list files processed9
-w, --veryverbose	very verbosely list files processed9
<i>Konfigurationsdatei / Transfer Options:</i>	11
/LTA:user:password@server	11
/PROXY:[user:pass@]server:port	11
/TIMEOUT:sec	Connection timeout: default 30 seconds seconds11
/TMP[:dir]	Use dir instead of temp directory for ZIP file11
/LOCALDIR	Use local path instead of temp directory for ZIP file11
/C	Compress ZIP file11
/ENC:passwd	Encrypt ZIP file with password11
/HYDE	Do not store password in XMP or DB11
<i>Konfigurationsdatei / Synchronize Options:</i>	11
/DB:path	Path and filename of existing SQLite DB to be used11
/XMP	Write XMP metadata file ('sourcefile.xmp')12
<i>Konfigurationsdatei / Retry Options:</i>	12
/NMI:n	Number of errors before shutdown: default 1012
/LOOP	Loop upload until terminated with ctrl-c12
/INTERVAL:hh:mm-hh.mm	Loop upload only in dedicated time range: from-to ..12
<i>Konfigurationsdatei / Logging Options:</i>	12
/LOG:file	Output to LOG file (default: no LOG file)12