Delete

Mit der Funktion **Delete** (`del.pm`), kann der Anwender komplette Archive, Unterverzeichnisse in Archiven und einzelne Dateien löschen. Zum Löschen wird der Parameter `-d` und der relative oder absolute Pfad des zu löschenden Objektes angegeben. Beim Löschvorgang ist dann zu prüfen, ob Verknüpfungen auf das zu löschende Objekt existieren. Diese müssen aktualisiert werden, damit sie weiterhin gültig sind, wenn das angegebene Objekt gelöscht wird.

Ablauf von Delete

Mit der Methode `delete_d()` wird der Löschvorgang nach der Parameterübergabe von `Invoker.pm` gestartet. Zuerst wird der Anwender gefragt, ob er das angegebene Objekt wirklich löschen will, damit es zu keinem versehentlichen Löschvorgang kommt. Wenn der Löschvorgang fortgesetzt werden soll, wird im Archiv nach gleich benannten Archiven gesucht und die Ergebnisse zeitlich sortiert.

Wenn keine gleichnamigen Archive gefunden wurden und ein komplettes Archiv gelöscht werden soll, muss dieser Eintrag aus `Hashtable.txt` entfernt werden. Wenn kein weiterer Eintrag in `Hashtable.txt` enthalten ist, wird das ganze Dokument gelöscht und die eigentliche in **Delete** enthaltene Löschfunktion `del()` wird ausgeführt. Wurden gleichnamige Archive gefunden, müssen diese auf enthaltene Verknüpfungen geprüft werden. Die Funktion `del()` kann ausgeführt werden, wenn keine Verknüpfungen, gefunden wurden oder wenn die die mit `checkLink()` geprüften Verknüpfungen auf kein zu löschendes Objekt zeigen.

Zu löschende Dateien, auf die eine Verknüpfung zeigt, werden in das zeitlich vorhergehende Archiv kopiert und die Verknüpfungen aus diesem Archiv entfernt. Dann wird mit `changeLinks()` in den weiteren gleichnamigen Archiven nach dieser Verknüpfung gesucht und bei einem Fund wird diese Verknüpfung mit `newLink()` aktualisiert. Nun kann der endgültige Löschvorgang `del()` gestartet werden, der das übergebene Objekt endgültig löscht.

Vorgehensweise beim Löschen eines Archivs, Verzeichnisses oder einer Datei

1. Erzeugen einer neuen Instanz von del mittels der Methode `new()`.
2. Hinzufügen des Verboselevels mittels der Methode `setVerboseLevel()`.
3. Hinzufügen des zu löschendem Objektes mit Verzeichnispfad mittels der Methode `addDestination`().
4. Löschen des Objektes mittels der Methode `delete_d()`.

Klassenbeschreibung Delete

Attribute der Klasse Delete (del.pm)

deleteFile	In diesem Attribut befindet sich das zu löschende Objekt mit Pfadangabe.
mainArchivpath	In diesem Attribut befindet sich der Pfad zum Archiv ohne Objektangabe.
archivFullName	Hier ist der Name des zu löschenden Objektes mit Datumsstempel und ohne Pfad angegeben.
archivName	Hier ist der Name des zu löschenden Objektes ohne Datumsstempel und ohne Pfad angegeben.
verbosity	In diesem Attribut befindet sich eine Instanz der Klasse Verbosity, welche für die Ausgabe von Meldungen dient.
message	In diesem Attribut befindet sich eine Instanz der Klasse Message, welche für die Ausgabe von Meldungen dient, auch wenn Verbosity deaktiviert ist.

Methoden der Klasse Delete (del.pm)

new

Beschreibung: Erzeugt ein Objekt der Klasse del bei einer Löschanweisung.
Parameter: Erhält keinen Parameter.
Rückgabe: Gibt die Hashreferenz $self ohne zugewiesene Informationen zurück.

addDestination

Beschreibung: Sucht aus dem erhaltenen Pfad die Werte für deleteFile (=$destination), mainArchivpath, archivFullName und archivName und weist sie zu. Des Weiteren wird geprüft, ob dieses Objekt wirklich existiert.
Parameter: $destination ist die Angabe des zu löschenden Objektes inkl. Pfad.
Rückgabe: Enthält keinen Rückgabewert.

setVerboseLevel

Beschreibung: Setzt den Level für die Verbose-Ausgabe.
Parameter: $level = 0 für keine Ausgabe und 1 für aktivierte Ausgabe.
Rückgabe: Enthält keinen Rückgabewert.

delete_d

Beschreibung: Ist die Hauptfunktion des Skripts und ruft nacheinander weitere Methoden für den Löschvorgang auf. Diese Funktion wird von Invoker.pm aufgerufen. Kopiert erhaltene Daten aus checkLink() von dem aktuellen Archiv in das vorhergehende und löscht dort die Verknpüpfungen.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors.
Rückgabe: Enthält keinen Rückgabewert.

check

Beschreibung: Fragt auf der Konsole, ob angegebenes Objekt wirklich gelöscht werden soll und bricht den Vorgang ab, wenn er verneint wurde.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors.
Rückgabe: Enthält keinen Rückgabewert.

findPreDir

Beschreibung: Sucht nach Archiven mit dem gleichen Namen und sortiert diese zeitlich. Führt updateHashtable() aus, wenn keine Archive gefunden wurden und zu löschendes Objekt ein Archiv ist.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors.
Rückgabe: Array mit den älteren gefundenen Archiven.

updateHashtable

Beschreibung: Löscht den Archiveintrag von archivName aus Hashtable.txt. Wenn kein Eintrag mehr in Hashtable vorhanden ist wird die Textdatei komplett gelöscht.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors.
Rückgabe: Enthält keinen Rückgabewert.

findLinksPreDir

Beschreibung: Sucht im erhaltenen Ordner nach Verknüpfungen oder nach Verknüpfungen mit dem gleichen Namen, wie der des erhaltenen Dateinamens.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors. $preDir enthält Verzeichnis- oder Dateiname.
Rückgabe: Array mit den gefundenen Verknüpfungen.

checkLink

Beschreibung: Prüft, ob die erhaltenen Verknüpfungen auf eine zu löschende Datei zeigen.

Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors. Erhält Array mit Verknüpfungen.
Rückgabe: Gibt ein Hash zurück. Im Key steht die zu löschende Datei inkl. Pfad und in Value ist ein Array mit den auf den Key verweisenden Verknüpfungen.

changeLink
Beschreibung: Prüft in allen gleichnamigen Archiven, ob enthaltene Verknüpfung wirklich auf zu löschende Datei verweist. Gibt gefundene Verknüpfungen an newLink() weiter.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors. $newLink enthält Pfad zu früherer Dateiverknüpfung aus vorhergehenden Archiv. $newDat enthält Pfad zu kopierter Datei in vorhergehenden Archiv. @foundDir enthält Namen mit Zeitangabe der älteren gleichnamigen Archive.
Rückgabe: Enthält keinen Rückgabewert.

newLink
Beschreibung: Aktualisiert die im Hash enthaltenen Verknüpfungen auf den neuen Pfad von $newDat.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors. $newDat enthält Name und Pfad zu der neu zu verknüpfenden Datei. %newLinks enthält im Key die zu löschende Datei inkl. Pfad und in Value ein Array mit den auf den Key verweisenden Verknüpfungen.
Rückgabe: Enthält keinen Rückgabewert

del
Beschreibung: Löscht das vom Anwender angegebene Verzeichnis inkl. Unterverzeichnisse oder die vom Anwender angegebene Datei.
Parameter: $inself erhält die Informationen aus der Hashreferenz des Konstruktors.
Rückgabe: Enthält keinen Rückgabewert.

DESTROY
Beschreibung: Zerstört das Objekt
Parameter: Erhält keinen Parameter
Rückgabe: Enthält keinen Rückgabewert

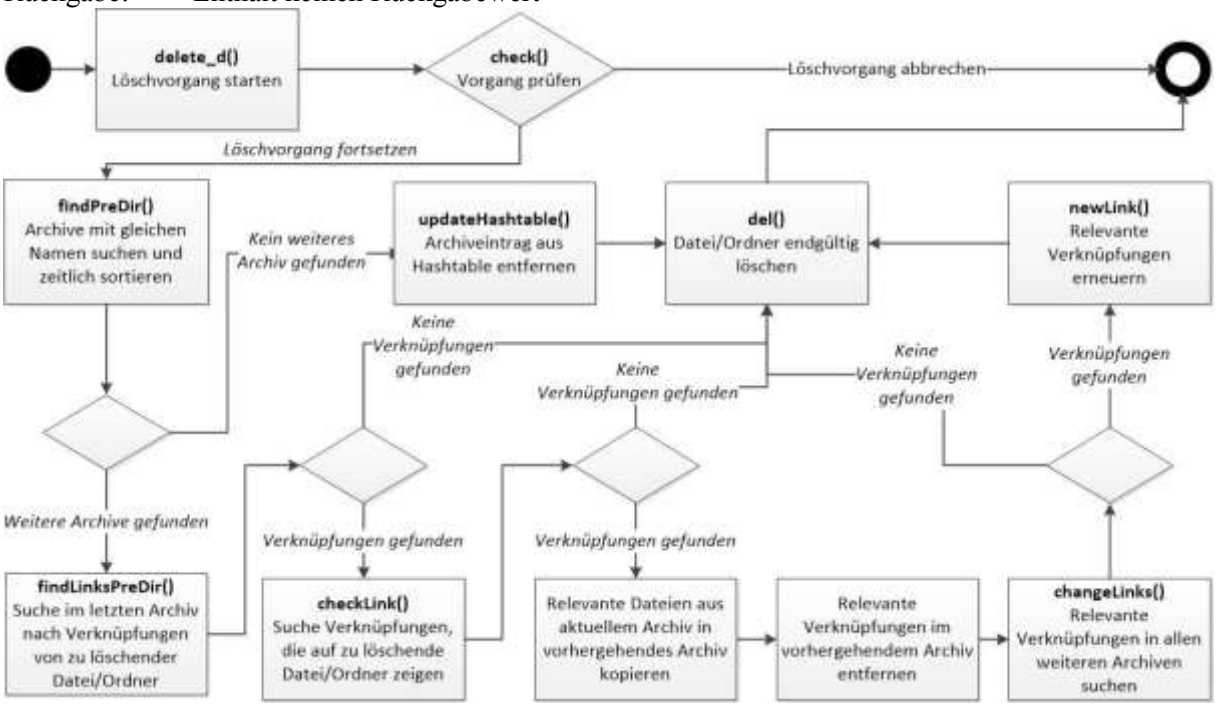

Abbildung 1 Ablaufdiagramm für Delete