



Karsten Viola

# Fix verknüpft

## Hard-Link-Backups für Windows

**Keine Zeit fürs Backup? Das hier vorgestellte Windows-Skript hält mit dem Werkzeug rsync vollautomatisch Schnappschüsse Ihrer Dateien fest, ohne dass Sie die Arbeit unterbrechen müssen. rsync kopiert bei jedem Lauf inkrementell nur die geänderten Dateien - dank Hard-Links erscheint auf dem Ziellaufwerk aber jede Sicherung als Vollbackup.**

[Unterthema: Cygwin-Einschränkungen](#)

Die meisten Datenverluste gehen keineswegs auf das Konto defekter Speichermedien. Todesursache Nummer eins - weit häufiger noch als ein Virenbefall - ist die fahrlässige Löschung durch den Anwender selbst. Besonders tückisch sind Fehlbedienungen, die erst Tage später auffallen - etwa dass man vor drei Tagen ein wichtiges Kapitel aus der Diplomarbeit entsorgt hat.

Mit einer Backup-Strategie im Rücken, die frühere Versionen der Arbeitsdateien archiviert, sind solche Fehltritte weniger schmerzhaft. Eine solche können Sie etwa umsetzen, indem Sie Ihre Daten täglich auf ein anderes Medium sichern und die Kopien eine Zeit lang aufbewahren. Weniger Disziplin erfordert das hier vorgestellte Skript rsyncBackup.vbs, das wir mit der Windows-Version des Open-Source-Werkzeugs rsync in VBScript implementiert haben.

### rsync

rsync bewährt sich auf Unix-Systemen seit Jahren für die Spiegelung kompletter Verzeichnishierarchien, etwa auf ein zweites Laufwerk oder einen Server im Netz [1].

rsyncBackup.vbs erleichtert den Einsatz unter Windows und erspart es Ihnen weitgehend, sich mit den umfangreichen Kommandozeilenschaltern auseinander zu setzen. Für die Konfiguration genügt ein Text-Editor.



**In jeden neuen Schnappschussordner blendet rsync Dateien, die seit der letzten Sicherung nicht geändert wurden, als Hard-Links ein - auf der Platte belegen sie keinen zusätzlichen Speicherplatz.**

rsync arbeitet stets inkrementell: Beim ersten Abgleich kopiert es alle Dateien der Quellverzeichnisse ins Ziel, bei den folgenden Sicherungen beschränkt es sich auf die Änderungen. Das Skript nutzt ein relativ neues rsync-Feature, das sich wunderbar für Backups eignet: Die Option `--link=dest` nimmt einen Ordner auf dem Ziellaufwerk entgegen, der die zuletzt angefertigte Sicherung enthält. rsync vergleicht den aktuellen Datenbestand damit, kopiert die neuen und

geänderten Dateien aber in einen frischen Zielordner.

Dort fügt es anschließend für alle Dateien, die sich nicht geändert haben, so genannte Hard-Links ein. Das Ergebnis ist im Explorer nicht von einer vollständigen Kopie des Quellordners zu unterscheiden - zusätzlichen Speicherplatz belegen aber nur die neuen und geänderten Dateien.

Ein Beispiel: Sie erstellen mit unserem Skript an drei aufeinander folgenden Tagen ein Backup Ihres Arbeitsordners. Am Montag kopiert rsync den kompletten Datenbestand auf das Ziellaufwerk. Am Dienstag erstellt das Skript einen weiteren, mit dem aktuellen Datum beschrifteten Ordner. rsync stellt fest, dass Sie nur die Datei text.doc geändert haben und überträgt nur diese in den neuen Ordner. Alle anderen Dateien, die Sie nicht angefasst haben, kopiert rsync nicht erneut. Stattdessen fügt es im neuen Backup-Ordner Hard-Links ein, die auf die bereits am Montag gespeicherten Dateien verweisen.

Später am Dienstag löschen Sie die Datei text.doc und erstellen die Datei tabelle.xls. Wenn das Backup-Skript am Mittwoch läuft, wird es tabelle.xls in den neuen Backup-Ordner kopieren. Alle anderen Dateien - außer text.doc, dass ja am Mittwoch in Ihrem Arbeitsordners nicht mehr existiert - verknüpft rsync wiederum mit Hard-Links.

Wenn Sie nun die drei Backup-Ordner im Explorer betrachten, sehen Sie in jedem ein Vollbackup Ihres Arbeitsordners mit genau den Dateien, die dort zum Zeitpunkt der Sicherung vorhanden waren. Die Wiederherstellung könnte simpler nicht sein: Öffnen Sie einfach den Schnappschussordner des gewünschten Zeitpunkts und kopieren Sie die Dateien zurück.

## Hard-Links

Dass Windows auf NTFS-Laufwerken Hard-Links unterstützt, hängt Microsoft nicht an die große Glocke. Unter Windows XP kann sie nur das bordeigene Kommandozeilen-Tool fsutil erstellen. Hard-Links unterscheiden sich grundlegend von den herkömmlichen Verknüpfungen, die Windows als einfache Dateien mit der Endung .lnk ablegt. Bei einem Hard-Link handelt es sich um einen zusätzlichen Verzeichniseintrag, der auf einen bereits gespeicherten Dateiinhalt zeigt.

Hard-Links besitzen verwirrende Eigenschaften: Angenommen, Sie erstellen einen Hard-Link auf die Datei a.txt und nennen ihn b.txt. Wenn Sie mit einem Texteditor den Inhalt der Datei a.txt bearbeiten und speichern, werden Sie die Änderung anschließend auch in b.txt wiederfinden. Sie sollten die Dateien in den Backup-Ordnern, die das Skript erstellt, also niemals an Ort und Stelle bearbeiten.

Es besteht kein Unterschied zwischen dem Verzeichniseintrag der zuerst gespeicherten Datei und darauf verweisenden Hard-Links, die rsync erstellt. Dennoch können Sie ältere Schnappschüsse bedenkenlos löschen: Erst wenn der letzte Hard-Link zu einer Datei, also ihr letzter Verzeichniseintrag, verschwindet, haben Sie die Datei endgültig gelöscht.

Wie viel Speicherplatz der Backup-Ordner, in den das Skript sichert, effektiv belegt, kann der Explorer nicht bestimmen: Weil er nichts von Hard-Links weiß, addiert er einfach die Dateigrößen aller Backups und zeigt ein Vielfaches des tatsächlich belegten Speichers an. Ein verlässliches Maß ist nur die Zahl der freien Bytes, die der Explorer in den Eigenschaften des Backup-Laufwerks zeigt.

## Einrichten

Packen Sie das Zip-Archiv, das Sie unter dem Soft-Link finden, in einem beliebigen Verzeichnis aus. Es enthält neben unserem Skript rsyncBackup.vbs auch das ausführbare rsync.exe und die dafür nötigen Cygwin-Bibliotheken. Das Skript hat weder eine grafische Oberfläche noch eine separate Konfigurationsdatei. Die wenigen nötigen Einstellungen tragen Sie mit einem Texteditor direkt im

Skript rsyncBackup.vbs ein.

Die für die Konfiguration relevanten Variablen finden Sie im markierten Abschnitt gleich am Anfang. Zeilen, die mit einem einfachen Anführungszeichen beginnen, sind Kommentare, die der Windows Script Host ignoriert. Die zu sichernden Quellpfade nimmt die Variable `sourceFolders` in Form eines Arrays entgegen: In die umschließende Klammer können Sie einen oder mehrere in Anführungszeichen eingeschlossene Verzeichnispfade eintragen. Mehrere Einträge müssen per Komma getrennt sein. Das Skript kann nur mit absoluten Pfaden etwas anfangen, die mit einem Laufwerksbuchstaben beginnen.

Mit der Variablen `DESTINATION` legen Sie das Verzeichnis fest, in dem das Skript die Backups speichert. Achten Sie darauf, dass das Ziellaufwerk mit dem Dateisystem NTFS formatiert ist, was sich in den Explorer-Eigenschaften des Laufwerks ablesen lässt. Auf eine FAT-Partition oder ein Netzwerklaufwerk, das mit einem Laufwerksbuchstaben verbunden ist, können Sie zwar auch sichern - dort stehen die Hard-Links aber nicht zur Verfügung, sodass rsync jedes Mal alle Dateien vollständig kopieren würde. Den effizienten Netzwerk-Modus, bei dem rsync nur die innerhalb der Dateien geänderten Bytes an eine zweite rsync-Instanz sendet, unterstützt die vorliegende Version unseres Windows-Skripts nicht.

Idealerweise lassen Sie das Skript auf eine extern angeschlossene Platte sichern. Wenn Sie keine zur Hand haben, sollten Sie den Backups zumindest eine separate Partition gönnen, auf der sie bessere Überlebenschancen haben, wenn etwa das Dateisystem der Arbeitspartition beschädigt wird.

Im Array `excludeFiles` können Sie noch Pfade (oder Suchmuster für solche) angeben, die Sie vom Backup ausnehmen möchten. Die Kommentarzeilen zeigen Beispiele. In der englischen rsync-Manpage erklärt der Abschnitt "Filter Rules" die exakte Syntax der Ausnahmeregeln. [2].

Wenn Sie etwa den kompletten Profilpfad ihres Benutzerkontos sichern (beispielsweise `C:\Dokumente und Einstellungen\kav`), ist es sinnvoll, den Browser-Cache des Internet Explorer auszuschließen. Dazu fügen Sie einfach das Suchmuster "Temporary\*" hinzu. Speichern Sie nun das Skript, für einen ersten Test genügen diese Einstellungen.

## Ausprobieren

Solange kein schwer wiegender Fehler auftritt, gibt das rsync-Skript keinerlei optische Rückmeldung. Um beim ersten Test zu beobachten, wie lange es läuft, öffnen Sie den Windows-Taskmanager und wechseln dort auf den Reiter "Prozesse". Am besten sortieren Sie die Liste alphabetisch. Starten Sie nun das rsync-Skript mit einem Doppelklick. Beim ersten Aufruf von rsync springt die bordeigene Firewall von Windows XP an, deren Warnung Sie in diesem Fall aber ignorieren dürfen. Klicken Sie einfach auf "Nicht mehr blocken".

Sollte rsync den Start mit einer kryptischen Fehlermeldung verweigern, ist auf Ihrem System möglicherweise bereits eine ältere Version der Bibliothek `cygwin1.dll` vorhanden, die eine andere Anwendung bereits in den Speicher geladen hat. Um dieses Problem zu lösen, durchsuchen Sie Ihren PC nach weiteren Kopien der Bibliothek und ersetzen Sie diese mit der aktuelleren Version. Booten Sie Windows neu, bevor Sie einen erneuten Versuch unternehmen.

Wenn alles glatt läuft, erscheint im Taskmanager der Prozess des Windows-Skript-Host namens "wscript" und gleich darauf noch weitere Prozesse mit dem Titel "rsync". Das Backup ist abgeschlossen, sobald der wscript-Prozess wieder aus der Liste verschwunden ist. Im Zielverzeichnis finden Sie einen mit Datum und Uhrzeit beschrifteten Ordner, in das rsync die angegebenen Quellen kopiert hat.

Ändern Sie nun versuchsweise eine einzelne Textdatei in Ihrem Quellverzeichnis und lesen Sie die

Anzahl der freien Bytes des Backup-Laufwerks in seinem Eigenschaften-Dialog ab. Jetzt starten Sie das Skript erneut; dieses Mal sollte es deutlich schneller fertig sein. Um festzustellen, ob eine Datei seit der vorigen Sicherung geändert wurde, orientiert sich rsync nicht am Archiv-Bit, sondern nutzt den Zeitstempel, mit dem Windows den Zeitpunkt der letzten Änderung dokumentiert.

Im Zielordner erscheint der zweite Schnappschussordner. Sein Inhalt ist - bis auf die von Ihnen geänderte Datei - identisch mit der ersten Sicherung. Der freie Speicher sollte aber kaum geschrumpft sein, nur die zusätzliche Kopie der geänderten Datei verbraucht Platz.

Die Log-Datei rsyncBackup.log, in der das Skript sämtliche Aktionen und Fehler protokolliert, finden Sie im Profilverzeichnis Ihres Benutzerkontos, in der Regel unter C:\Dokumente und Einstellungen\Benutzerkonto. Bei jedem Start notiert das Skript zuerst das rsync-Kommando mit allen übergebenen Parametern. Darauf folgt die Ausgabe von rsync, das alle neuen und geänderten Dateien auflistet, die es in den neuen Backup-Ordner kopiert hat. In der darauf folgenden Zeile ist hinter "sent" die Anzahl der Bytes abzulesen, die rsync gesichert hat. Zum Schluss folgen die Fehlermeldungen; hier vermerkt rsync etwa, welche Dateien sich nicht kopieren ließen.

## Automatisieren

Sie könnten nun eine Verknüpfung auf Ihren Desktop legen und das Skript jedes Mal per Doppelklick starten, wenn Sie eine Sicherung anfertigen möchten. Bequemer und sicherer ist es aber, den Windows-Taskplaner das Skript regelmäßig aufrufen zu lassen.

Es ist allerdings - etwa im Hinblick auf einen Virenbefall - nicht ganz ungefährlich, eine Backup-Festplatte für die vollautomatische Sicherung dauerhaft an das Arbeitssystem angeschlossen zu lassen. Wenn Sie mit einem eingeschränkten Benutzerkonto arbeiten, sollten Sie diesem unbedingt den Schreibzugriff auf die Backups verwehren, dazu später mehr. Zusätzlich sollten Sie regelmäßig Kopien auf weiteren Speichermedien erstellen.



**Wer das Backup-Skript vom Windows-Taskplaner zu jeder vollen Stunde starten lassen will, muss sich durch unnötig komplizierte Dialoge kämpfen.**

Weil nur die neuen und geänderten Dateien Speicherplatz kosten, ist es sogar praktikabel, das vollautomatische Backup jede Stunde einen Schnappschuss anfertigen zu lassen. Damit spannen Sie ein engmaschiges Sicherheitsnetz: Notfalls können Sie sogar die Version ihres Arbeitsordners hervorzaubern, die etwa gestern morgen um 10 Uhr auf Ihrer Platte lag.

Damit Windows das Skript regelmäßig aufruft, erstellen Sie in der Systemsteuerung eine neue geplante Task (Systemsteuerung/Geplante Tasks/Geplante Task hinzufügen). Der Assistent, der daraufhin seinen Dienst anbietet, ist nicht gerade intuitiv zu bedienen. Halten Sie sich einfach an die in den folgenden Absätzen beschriebenen Schritte:

Wechseln Sie auf die zweite Seite des Assistenten und klicken Sie auf "Durchsuchen", um das Skript rsyncBackup.vbs auszuwählen. Im folgenden Dialog bestimmen Sie zunächst, dass die Aufgabe "täglich" ausgeführt werden soll, dann klicken Sie auf "Weiter". Als Startzeit geben Sie 00:00 Uhr morgens vor und teilen dem Assistenten erneut mit, dass er "täglich" aktiv werden soll. Das Startdatum ist mit dem aktuellen vorbelegt.

Im nächsten Schritt geben Sie das Konto des Anwenders an, in dessen Namen das Skript gestartet werden soll. Voreingestellt ist der angemeldete Benutzer. Zusätzlich erfragt der Assistent das passende Kennwort. So kann der Taskplaner, der mit Systemrechten läuft, die Aktion auch dann ausführen, wenn Sie gar nicht angemeldet sind. Als Voraussetzung muss das angegebene Benutzerkonto ein Kennwort besitzen. Achtung: Wenn Sie das Kennwort des Kontos später ändern, wird das Skript nicht mehr automatisch gestartet.

Der Assistent hält seine Aufgabe jetzt für abgeschlossen, aber wo war denn nun der Schalter für den stündlichen Aufruf? Weisen Sie ihn mit der Checkbox auf der letzten Seite an, nach der Fertigstellung die "Erweiterten Eigenschaften" zu öffnen.

Hier ist der Reiter "Zeitplan" relevant, er zeigt wie zuvor eingestellt an, dass der Job "täglich" um 00:00 Uhr gestartet wird. Ein Klick auf "Erweitert..." bringt den erhenteten Optionsdialog zum Vorschein, in dem sich das "Wiederholungsintervall" von einer Stunde einstellen lässt. Zusätzlich bestimmen Sie hier, dass die Aktion bis 23:59 Uhr laufen soll. Das wars. Jetzt startet das Skript rund um die Uhr zu jeder vollen Stunde.

Sie müssen für einen Test nicht bis zur nächsten vollen Stunde warten: Klicken Sie einfach mit der rechten Maustaste auf das Task-Icon und wählen Sie im Kontextmenü "Ausführen". Anschließend schauen Sie in der Logdatei nach, ob der Aufruf geklappt hat.

## Ausdünnen

Wenn rsync zu jeder vollen Stunde einen neuen Schnappschuss erstellt, wird das Backup-Verzeichnis recht schnell unübersichtlich. Damit das Sicherungslaufwerk nicht voll läuft, ist es sinnvoll, regelmäßig alte Backups auszusortieren. rsyncBackup.vbs enthält eine Routine, die Ihnen diese Aufgabe abnimmt.

Das Skript sortiert die Backup-Ordner in vier Stufen ein. Mit den Konfigurationsvariablen `STAGE0_HOURLY`, `STAGE1_DAILY` und `STAGE2_WEEKLY` bestimmen Sie, wie viele Sicherungen das Skript auf den ersten drei Stufen aufbewahren soll.

Die Variable `STAGE0_HOURLY` bezieht sich auf die unterste Stufe, die das in `DESTINATION` angegebene Zielverzeichnis selbst darstellt. Nach jedem rsync-Lauf zählt das Skript die dort vorhandenen Schnappschussordner. Sind es mehr als in der Variablen `STAGE0_HOURLY` eingestellt, verschiebt es überzählige Backups zunächst in den Ordner namens "\_stufe1". Dort bewahrt das Skript aber für jeden Tag nur das zuletzt erstellte Backup auf - frühere Backups, die an diesem Tag erstellt wurden, löscht das Skript.

Die anderen Stufen funktionieren analog dazu: In der Variablen `STAGE1_DAILY` stellen Sie ein, wie viele tägliche Backups das Skript aufbewahren soll, bevor es beginnt, die ältesten auf die nächste Stufe zu sortieren, die nur das jeweils letzte Backup einer Woche konserviert. Den Ordner "\_stufe3", der das jeweils letzte Backup eines Monats aufbewahrt, bearbeitet das Skript nicht weiter.

Wir haben das Skript so vorkonfiguriert, dass es alle Backups aufbewahrt und nichts umsortiert oder löscht. In der Variablen `STAGE0_HOURLY` steht deshalb der Wert 10 000. Welche Werte für Sie sinnvoll sind, hängt vom Datenvolumen Ihrer täglich geänderten Dateien ab und davon, wie häufig Sie das Skript laufen lassen.

Wenn Ihre Backup-Platte groß genug ist und Sie das Skript nur hin und wieder aufrufen, spricht nichts dagegen, einen großen Wert in `STAGE0_HOURLY` einzustellen. Sie können die Einstellung auch später jederzeit ändern und sich an die für Sie sinnvolle Einstellung herantasten.

## Einschränken

Das Skript kann schwächere PCs etwas ausbremsen, während es ein neues Backup anfertigt - lästig, wenn man es stündlich aufrufen lässt und gerade Wichtigeres zu tun hat. Um dieses Problem zu lösen, liefern wir im Download-Archiv das Batch-Skript rsyncStart.bat mit. Wenn Sie dieses per Doppelklick anstelle der vbs-Datei starten, ruft es den Skript-Interpreter mit niedriger Priorität auf. Dem so gestarteten rsync-Skript teilt Windows nur die brachliegende Rechenzeit zu, die gerade kein anderes Programm für sich reklamiert.

Wenn Ihr PC während der Sicherung trotzdem noch lahmt, kann auch ein On-Access-Virens Scanner schuld sein, der alle von rsync angefassten Dateien erneut durchleuchtet. In diesem Fall ist es ratsam, den Backup-Ordner vom ständigen Virens Scan auszunehmen. Wenn Sie die Backups regelmäßig manuell scannen wollen, wählen Sie besser nur die zuletzt erstellten Ordner aus - alle von rsync erstellten Backups zu scannen, kann sehr lange dauern, da sich der Virens Scanner jeden Hard-Link erneut vornimmt.

Beim Aufruf von rsyncStart.bat durch den Taskplaner erscheint bei jedem Start kurz eine Eingabeaufforderung mitten auf dem Bildschirm. Ein pragmatischer Weg, diese loszuwerden: Klicken Sie im Explorer rechts auf rsyncStart.bat und erstellen Sie eine Verknüpfung. In deren Eigenschaften legen Sie unter "Ausführen" fest, dass das Programm "minimiert" gestartet werden soll. Wenn Sie diese Verknüpfung im Taskplaner konfigurieren, taucht der Skript-Interpreter nur in der Taskleiste am unteren Bildschirmrand auf.

Wenn Sie mit einem eingeschränkten Nutzerkonto arbeiten, was aus Sicherheitsgründen dringend anzuraten ist [3], können Sie die Backups wirksam vor versehentlichem Löschen und der Manipulation durch Software-Schädlinge schützen: Dazu lassen Sie den Taskplaner die Verknüpfung auf rsyncStart.bat nicht mit Ihrem eigenen Benutzerkonto starten, sondern geben ein Konto an, das Administratorrechte besitzt. Nutzen Sie dafür besser nicht das Windows-Konto namens "Administrator", sondern erstellen Sie für diesen Zweck ein eigenes privilegiertes Konto.

Der eingeschränkte Benutzer, mit dem Sie normalerweise arbeiten, kann die vom Skript erstellten Backup-Ordner nun ausschließlich lesen, hat aber keinen Schreibzugriff. Achtung: Die Log-Datei finden Sie immer im Profilordner des aufrufenden Benutzerkontos.

## Zurücklehnen

Auch wenn Ihnen das rsync-Skript die lästige Backup-Routine abnimmt: Sie dürfen sich nicht blind darauf verlassen. Werfen Sie regelmäßig einen Blick in die Log-Datei. Am besten richten Sie im Taskplaner einen weiteren Job ein, der Ihnen das Protokoll jeden Morgen mit Notepad auf den Desktop bringt. Indem Sie eine aktuelle Version Ihrer Arbeitsdaten regelmäßig auf optische Medien übertragen (etwa im Wochen-Rhythmus), sichern Sie Ihre Daten zusätzlich gegen Hardware-Schäden ab. (kav)

## Literatur

[1] Jürgen Schmidt, Beruhigungsmittel, Backups für kleine Linux-Server, [c't 7/06, Seite 212](#)

[2] Manpage von rsync: <http://samba.org/ftp/rsync/rsync.html>

[3] Johannes Endres, Heute ein Admin, Souverän arbeiten ohne Administrator-Rechte unter XP, c't 23/2005, S. 112

[Soft-Link 0609126](#)

## Kasten 1

---

### Cygwin-Einschränkungen

Die rsync-Version, die wir im Download-Archiv mit unserem Skript liefern, stammt aus dem Projekt Cygwin ([www.cygwin.com](http://www.cygwin.com)). Es handelt sich dabei nicht um ein waschechtes Win32-Programm, sondern um eine Unix-Software. Die Cygwin-Bibliotheken dienen als Kompatibilitätsschicht und gaukeln rsync eine Unix-Umgebung vor, deren Betriebssystemfunktionen sie auf das Windows-API abbilden.

So legt die Cygwin-Version von rsync Hard-Links korrekt an, die unter Unix wesentlich gängiger sind als in der Windows-Welt. Andersherum gibt es aber Einschränkungen: rsync weiß nichts von den Windows-typischen NTFS-Spezialitäten. So nimmt es weder Junctions noch Sparse-Dateien oder Streams in seine Backups auf. Auch über Dateinamen, die japanische Kanji-Zeichen enthalten, stolpert die Cygwin-Version.

Eine ärgerliche Einschränkung: rsync verarbeitet maximal 260 Zeichen lange Pfade. Wenn es auf längere stößt, etwa im Browser-Cache ("Temporary Internet Files"), vermerkt das Skript eine Fehlermeldung in der Log-Datei.

---

**Zu diesem Artikel existieren Programmbeispiele**

[0609-126.zip](#)