

HOWTO zu rsync (Remote Synchronisation)
 (C) 2008-2011 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
 OSTC Open Source Training and Consulting GmbH, <http://www.ostc.de>
 \$Id: rsync-HOWTO.txt,v 1.36 2011-11-13 16:31:19 tsbirn Exp \$

Dieses Dokument beschreibt die lokale und Remote-Duplizierung von UNIX-Dateisystemen mit "rsync".

Inhaltsverzeichnis

- 1) Einführung
- 2) Die wichtigsten Optionen
- 3) Weitere wichtige Optionen
- 4) Hinweise
- 5) Beispiele
- 6) Rollierende mehrfache Komplettsicherung mit "rsync + cp"
- 7) Optionen
 - 7a) Weitere interessante Optionen
 - 7b) Weitere interessante Optionen
 - 7c) Vollständige Liste der Optionen

1) Einführung

"rsync" (remote synchronization) ist ein leistungsfähiger und effizienter Ersatz für "rcp/scp". Es repliziert effizient großer Datenmengen (Verzeichnisbäume) lokal oder auch über ein Netzwerk hinweg auf andere Rechner. Die Software ist verfügbar unter <http://www.samba.org/rsync/>.

Zur Optimierung der Abgleich- und Übertragungsgeschwindigkeit werden dabei folgende Verfahren benutzt:

- * Dateien mit identischer Länge und gleichem Änderungsdatum ignorieren
- * Dateiteile per rollierender Prüfsumme auf Identität prüfen
- * NUR geänderte Teile einer Datei transferieren (Delta-Algorithmus)

Aufrufsyntax:

```
rsync [OPT] SRC... DEST           # Lokal replizieren
rsync [OPT] SRC... [USER@]HOST:DEST # Nach Remote replizieren
rsync [OPT] [USER@]HOST:SRC DEST   # Von Remote replizieren
rsync [OPT] [USER@]HOST:SRC       # Remote auflisten
```

Der Hauptspeicherbedarf steigt mit der Anzahl an zu synchronisierenden Dateien, da zum Synchronisieren auf beiden Seiten eine vollständige Dateiliste aufgebaut wird, die pro Datei etwa 100 Byte umfasst. Weiterhin ist der Rechenzeit- und I/O-Aufwand von rsync recht hoch, da der gesamte Dateibaum eingelesen wird und evtl. Prüfsummen über geänderte/neue Dateien gebildet werden. Ab mehr als 1 Mio zu synchronisierende Dateien sollten daher mehr als 1 GByte Hauptspeicher und ein leistungsfähiger Prozessor AUF BEIDEN SEITEN vorhanden sein.

- * Bis V2.6.9 wurde (bei Option --recursive) die Liste der zu transferierenden Dateien ZUERST VOLLSTÄNDIG aufgebaut und DANN der Transfer durchgeführt.
- * Ab V3.0.0 wird (bei Option --recursive) ein "inkrementeller" Modus beim Aufbau der Dateiliste verwendet. Bereits während dem Aufbau der Liste der zu synchronisierenden Dateien werden Dateien synchronisiert, d.h. die beiden Vorgänge laufen "verzahnt" ab. Abschalten kann man dieses Feature mit Option --no-inc-recursive bzw. --no-i-r.

Dieser inkrementelle Modus führt zu sehr grosser Speicherplatzersparnis (bei 10 Mio Dateien a 100 Byte pro Datei waren vorher bei einem lokalen Transfer etwa 2 GByte Hauptspeicher für die beiden Dateilisten erforderlich). Die reine Laufzeit wird um 10-25% geringer, da der Zeitaufwand für die Listenerstellung insgesamt konstant bleibt, aber bereits für Transfers genutzt werden kann. Bedingung ist auf BEIDEN Seiten eine rsync-Version ab 3.0.0 (Protokoll 30).

Die aktuelle Version ist "rsync version 3.0.9 protocol version 30".

2) Die wichtigsten Optionen

Option (lang+kurz)	Bedeutung
--help -h	Hilfe anzeigen
--archive -a	Archiv-Modus = -Dgloptr (-Do benötigen root-Rechte!) device group links owner permission times recursive
--group -D -g	Geräte + Spezialdateien transferieren (nur root!) Besitzer-Gruppe transferieren

--links	-l	Symbolische Links transferieren
--owner	-o	Besitzer transferieren (nur root!)
--perms	-p	Zugriffsrechte transferieren
--times	-t	Änderungszeit+datum transferieren
--recursive	-r	Verzeichnis-Bäume replizieren

--rsh=CMD	-e	Zu benutzende "Remote Shell" (z.B. ssh)
--compress	-z	Komprimierung der Datenübertragung (remote sinnvoll)

--quiet	-q	Ablaufmeldungen unterdrücken (nur Fehlermeldungen)
--verbose	-v	Meldung pro transferierter Datei (--itemize-changes)
2x --verbose	-vv	Meldung mit übersprungenen Dateien
3x --verbose	-vvv	Meldung mit Debuginfo (sehr viel!)

* Die häufigsten Aufrufe haben folgende Form (-a/--archive):

```
rsync -a SRC/ DEST # Lokal replizieren
rsync -e ssh -az SRC/ USER@HOST:DEST # Remote über SSH replizieren
rsync -e ssh USER@HOST:DEST # Remote-Dateien auflisten
rsync -e ssh -az -HSAV SRC/ USER@HOST:DEST # ALLES remote replizieren!
```

* Zur Replikation des Datei-Besitzers (Owner), von Geräten (Devices) und Spezialdateien (Named Sockets, Named Pipes) sind auf der Zielseite root-Rechte erforderlich (Optionen -o und -D).

* Option -D ist die Zusammenfassung von --devices + --special.

* Option -t/--times bzw. -a transferiert die Änderungszeiten der Dateien, nur dann funktioniert die rsync-Optimierung zum Ignorieren identischer Dateien auf der Basis der Änderungszeit. Lässt man diese Option weg, entspricht dies dem Setzen von -I/--ignore-times beim nächsten Transfer.

* Komprimierung -z/--compress lohnt sich nur bei Remote-Replikation der Daten über eine "dünne" Netzwerkleitung (z.B. DSL). Auf beiden Seiten werden die CPUs stärker belastet als ohne Komprimierung.

* Option -a/--archive umfasst NICHT die Optionen -A/--acls, -H/--hard-links, -S/--sparse und -X/--xattrs. Sollen ACLs, Hardlinks, Sparse-Dateien und Extended Attribute ebenfalls 1:1 repliziert werden, so sind diese Optionen zusätzlich zu -a/--archive anzugeben (siehe oben).

3) Weitere wichtige Optionen

Opt (lang+kurz)	Bedeutung
--delete	Dateien löschen, die auf dem Sender fehlen
--delete-during	Dateien WÄHREND Transfer löschen [--del]
--delete-before	Dateien VOR Transfer löschen
--delete-after	Dateien NACH Transfer löschen

--dry-run -n	Aktionen nur anzeigen, NICHT ausführen
--update -u	Neuere Dateien auf Zielseite NICHT überschreiben

--hard-links -H	Hardlinks erhalten (aufwendig!)
--sparse -S	Sparse Dateien erhalten (lange 0-Byte Sequenzen)
--acls -A	ACLs transferieren (enthält --perms)
--xattrs -X	Extended Attributes transferieren (ab 3.0.0)

* VORSICHT bei der Verwendung von --del...: Wird die FALSCHE RICHTUNG bei der Replikation gewählt, sind die NEUEN Dateien auf der Quellseite weg! Unbedingt VORHER TESTEN mit -n/--dry-run, sonst wird evtl. die falsche (zu sichernde) Seite zerstört!

* Option --delete entspricht bis V2.6.9 --delete-before, seit V3.0.0 auf beiden Seiten entspricht sie --del/--delete-during (performanter).

* Option --delete-after löscht erst nach dem erfolgreichen Übertragen aller Dateien die zu entfernenden.

* Option -u/--update verhindert Überschreiben von Dateien, die auf Zielseite NEUER sind als auf Quellseite (z.B. infolge von Änderungen auf beiden Seiten). Damit dies korrekt funktioniert, MUSS die Uhrzeit auf beiden Rechner identisch sein (NTP = Network Time Protocol aktivieren)!

* Optionen -H/-S/-A/-X replizieren weitere wichtige Eigenschaften von Dateien sofern beide Dateisysteme diese beherrschen (Hardlinks, Sparse-Dateien mit "Löchern", Access Control Lists, Extended Attributes).

* Optionen -S/--sparse und -H/--hard-links unbedingt setzen, sonst kann auf

der Zielseite wesentlich mehr Plattenplatz verbraucht werden. Die Option `-H/--hard-links` erhöht allerdings den Aufwand zur Erstellung der Dateiliste enorm.

- * Ein BackupPC-Repository MUSS unbedingt mit `-H/--hard-links` kopiert werden, sonst werden die zur Speicherplatzersparnis bei den mehrfachen Sicherungen eingesetzten Hardlinks als echte Dateien kopiert (vervielfacht Platzbedarf).

4) Hinweise

- * Häufig benötigte Optionen fertig zum Kopieren per Maus:

```
rsync SRC/ USER@HOST:DEST
-a                # --archive = -Dgloptr
-glptr           # (--group --links --perms --times --recursive)
-D -o            # (--devices --specials --owner (root-Rechte nötig!))
-z              # --compress (Komprimierung, lokal sinnlos!)
-H -S -A -X     # --hard-links --sparse --acls --xattrs
--del            # --delete-during
--exclude=/lost+found # Mehrfach erlaubt
--numeric-ids    # UID/GID statt User/Group-Namen transferieren (speed)
--bwlmit=1000    # Transferbandbreite beschränken 1000 KByte/s = 1 MB/s
--modify-window=2 # Bei Windows Zeitvergleich nur auf 2 Sec genau (FAT)
--modify-window=3602 # Bei Sommerzeit/Winterzeit + Windows (FAT)
-v              # Verbose (transferierte Dateien auflisten)
--progress       # Fortschrittmeldungen während Transfer
--stats          # Statistik am Schluss
-c              # --checksum (immer Prüfsumme statt Datum+Größe vgl.)
-I              # --ignore-times (gl. Datum+Größe auch transferieren)
-W              # --whole-file (kein Delta-Algorithmus)
--human-readable # Größen in menschlich lesbarer Form (K,M,G)
--8-bit-output   # Sonderzeichen NICHT im Escape-Format \#123
--append         # Sofort anhängen, keine Zwischendatei
--partial        # Teilweise übertragene Dateien erhalten (Abbruch!)
--inplace        # Zieldatei direkt ändern (keine Zwischendatei)
```

- * `"/` am Ende der Quell-Verzeichnisnamen "SRC" NICHT vergessen, sonst wird nicht der INHALT des Verzeichnisses, sondern das Verzeichnis selbst repliziert (d.h. der Pfad "SRC" ist auf der Zielseite unter "DEST" sichtbar).

- * Bei Windows-Rechnern sollte `--modify-window=2` gesetzt werden, das eine Differenz $\leq 2s$ als gleiche Zeit interpretiert (da das Änderungsdatum unter Windows nur auf 2s genau gespeichert wird).

- * Es empfiehlt sich, auf der Quell- und Zielseite die mount-Optionen "noatime" + "nodiratime" oder "relatime" zu setzen, um beim Aufbau beider Dateilisten (Lesen von Verzeichnissen und Inodes) keine Schreiboperationen auszulösen.

- * Beim Einsatz von LVM (Logical Volume Manager) empfiehlt sich, einen READ-ONLY SNAPSHOT des zu transferierenden Dateisystems SRCDEV zu erstellen und nach dem Transfer wieder freizugeben. Die Größe des Snapshots an die Änderungsrate des Quelldateisystems während der Replikation anpassen (hier 5G).

```
lvcreate --permission r --size 5G --snapshot --name SNAPNAME SRCDEV
mount -o ro,noatime,nodiratime,async /dev/.../SNAPNAME MOUNTPOINT
rsync MOUNTPOINT/ DESTINATION
umount MOUNTPOINT
lvremove -f /dev/.../SNAPNAME
```

- * Der Delta-Algorithmus lohnt sich bei rein LOKALER Replikation auf der gleichen Maschine nicht, da er 2x vollständiges Lesen auf beiden Seiten + 1x Delta schreiben auf Zielseite bedingt. Lohnt sich nur bei Remote-Transfers über ein (relativ langsames) Netzwerk (z.B. DSL). Mit Option `--whole-file` abschalten.

5) Beispiele

Alle C-Dateien des aktuellen Verzeichnisses auf Server "HOST" in Verzeichnis "src" replizieren:

```
rsync *.c HOST:src/
```

Alle Dateien in Verzeichnis "src/bar" von Maschine "HOST" rekursiv in Verz. "/data/tmp/bar" (!) replizieren (d.h. dort Unterverz. "bar" erzeugen!):

```
rsync -avz HOST:src/bar /data/tmp
```

TIPP: `"/` am Ende von SRC heißt "Inhalt des Verzeichnisses" replizieren. Sonst wird Verzeichnis selbst repliziert (1 Stufe mehr, vor allem wichtig bei `--del`):

```
rsync -avz HOST:src/bar/ /data/tmp
```

Aktuelles Verzeichnis komplett auf Maschine "HOST" in Verzeichnis "save" replizieren (-C=CVS-Dateien weglassen, -z=komprimieren)

```
rsync -avzC . HOST:save
```

Drei Skripte zum Backup und Restore des lokalen eigenen Heimatverz. auf den Rechner "backup":

```
# 1) Heimat-Verz. nach Rechner "backup" synchronisieren
USER="dummy"
HOST="backup"
cd
rsync -avzHSAX --del -e ssh . "$USER@$HOST:."
```

```
# 2) Inhalt des Heimatverz. auf Rechner "backup" auflisten
USER="dummy"
HOST="backup"
rsync -r -e ssh "$USER@$HOST:."
```

```
# 3) Wiederherstellung von Dateien bzw. Verz. mit ihrem Inhalt von Rechner
# "backup" (. = alle Dateien)
USER="dummy"
HOST="backup"
cd
for FILE
do
    rsync -avzHSAX -e ssh "$USER@$HOST:./$FILE" "./$FILE"
done
```

6) Rollierende mehrfache Komplettsicherung mit "rsync + cp"

Man kann "rsync" und "cp" kombinieren, um einen rollierende mehrfache Komplettsicherung eines Dateibaums zu erstellen, ohne dafür den mehrfachen Platz zu verbrauchen. In aller Kürze geht das so (ACHTUNG: "/" am Ende ist notwendig!):

```
mkdir -p /backup/0          # 1x notwendig
rm -rf /backup/3          # Vorgestern löschen
mv /backup/2 /backup/3    # Vorgestern -> Vorgestern
mv /backup/1 /backup/2    # Gestern -> Vorgestern
cp -al /backup/0 /backup/1 # Aktuell -> Gestern (Hardlinks!)
rsync -a --del SRCDIR /backup/0/ # 1:1-Kopie + gelöschte Dateien weg
```

Werden diese Kommandos täglich durchgeführt, dann SCHEINEN "/backup/1", "/backup/2" und "/backup/3" jeweils eine komplette Vollsicherung des Ausgangsverzeichnisses zu sein (gestern, vorgestern und vorgestern).

In Wirklichkeit wird nur EINE "echte" Vollsicherung vom heutigen Tag unter "/backup/0" gemacht. Die "älteren" Vollsicherungen "/backup/[123]" entstehen durch HARDLINKS auf diese Vollsicherung. Alle Sicherungen zusammen müssen daher auf EINER physikalischen Platte liegen.

Komplett sind die älteren Vollsicherungen nur in dem Sinne, dass Änderungsdatum, Zugriffsrechte und Besitzverhältnisse inhaltlich unveränderter Dateien immer dem neuesten Stand dieser Dateiattribute entsprechen. D.h. eine Änderungshistorie dieser Attribute ist nicht möglich.

Der Speicherplatz für alle Sicherungen entspricht daher der Größe des aktuellen Dateibaums + der Gesamtgröße aller Änderungen der letzten 3 Tage. Dies ist exakt der gleiche Platzbedarf, den eine vollständige + 3 inkrementelle Sicherungen benötigen.

Bedeutung der obigen Optionen von "cp":

```
+-----+
| -a | archive (-dpR)
+-----+
| -d | no-dereference + preserve-link
| -p | preserve mode, ownership, timestamps
| -R | Rekursiv
| -l | Hardlinks statt Kopien
+-----+
```

Bedeutung der obigen Optionen von "rsync":

```
+-----+
| --archive      -a | Archiv-Modus = -Dgloptr (-Do benötigen root-Rechte!) |
+-----+
| --devices      -D | Geräte + Spezialdateien transferieren (nur root!)
| --devices      (-D) | Character/Blockgeräte transferieren (nur root!)
| --specials     (-D) | Spezialdateien transferieren
+-----+
```

--group	-g	Besitzer-Gruppe transferieren
--links	-l	Symbolische Links transferieren
--owner	-o	Besitzer transferieren (nur root!)
--perms	-p	Zugriffsrechte transferieren
--times	-t	Datum transferieren
--recursive	-r	Verzeichnis-Bäume replizieren

7) Optionen

7a) Weitere interessante Optionen

Option (lang+kurz)	Bedeutung	
--checksum	-c	VOR Transfer Änd. per Prüfsumme erk., NICHT Dat/Größe
--ignore-times	-I	Dateien gl. Datum+Größe NICHT überspringen (langsam!)
--whole-file	-W	Datei vollständig kop. = kein rsync Delta-Algorithm. (Std. lok. Tr., Delta-Alg. nur bei Netz-Transfer ok)

- * Diese Optionen DEAKTIVIEREN die drei Optimierungen von rsync:
 - + -c: Dateiteile per rollierender Prüfsumme auf Identität prüfen
 - + -I: Dateien mit identischer Länge UND gleichem Änderungsdatum ignorieren
 - * -W: NUR geänderte Teile einer Datei transferieren (Delta-Algorithmus)
- * Option --whole-file (Verzicht auf Delta-Algorithmus) ist Standard bei REIN LOKALEN Transfers, da der Delta-Algorithmus 2x vollständiges Lesen auf beiden Seiten + 1x Delta schreiben auf der Zielseite bedingt. Dies lohnt sich nur bei Remote-Transfer über ein (relativ langsames) Netzwerk (z.B. DSL).

7b) Weitere interessante Optionen

Option (lang+kurz)	Bedeutung	
--exclude=MUSTER		Verz./Dateien gemäß MUSTER ausschließen
--exclude-from=FILE		Dateien in FILE ausschließen
--include=MUSTER		Dateien gemäß MUSTER einschließen
--include-from=FILE		Dateien in FILE einschließen
--filter=RULE	-f	Dateien gemäß RULE auswählen/ausschließen
--cvs-exclude	-C	CVS-Dateien ausschließen
<hr/>		
--bwlimit=NNN		Transferbandbreite beschränken (KByte/sec)
--numeric-ids		UID/GID statt User/Group-Namen transferieren (speed)
--partial	(-P)	Teilweise übertragene Dateien erhalten (Abbruch!)
<hr/>		
--progress	(-P)	Fortschritts-Meldungen ausgeben (impl. --verbose)
--stats		Transfer-Statistik ausgeben
--itemize-changes	-i	Fortschritts-Meldungen ausg. [--progress + --partial] Übersicht aller Update-Änderungen (--verbose)

- * Option --exclude=MUSTER schliesst Verzeichnisse und Dateien mit Hilfe von Shell-Mustern vom Transfer aus (z.B. /tmp, /var, /lost+found, *~). Mit Hilfe von --filter=RULE sind komplexere Formen von Ein- und Ausschlusskriterien möglich. Mit Hilfe von --exclude-from=FILE und --include-from=FILE können die Filterregeln auch von Datei gelesen werden.
- * Option --bwlimit=NNN beschränkt die Übertragungsrate auf ein Maximum von NNN KByte/s, um konkurrierende Transfers auf der gleichen Verbindung nicht abzuwürgen.
- * Option --numeric-ids beschleunigt den Transfer, da nicht die Namen von Benutzern und Gruppen, sondern ihre UID/GID transferiert werden. Erfordert allerdings einheitliche ID-Vergabe auf beiden Seiten, z.B. durch ein zentrales LDAP-Verzeichnis oder lokalen Transfer.
- * Option --partial behält beim Verbindungsabbruch bereits transferierte Teile einer Datei und verwendet sie beim erneuten Aufsetzen der rsync-Vorganges. Besonders für langsame Verbindungen und SEHR GROSSE Dateien interessant.
- * Option --progress gibt Fortschrittsmeldungen folgender Form aus (32.77K = Dateigröße, 100% = bereits übertragener Teil, 30.04MB/s = Übertragungsrate, 0:00:00=Übertragungsdauer, xfer#10 = Anzahl bereits übertragener Dateien, to-check=14446/19441 = Anzahl noch zu prüfender Dateien/Anzahl insgesamt geprüfter Dateien):

...

Nov 14, 11 3:00

rsync-HOWTO.txt

Page 7/8

--version		Versionsnummer und Featurezusammenfassung anzeigen
--quiet	-q	Ablaufmeldungen unterdrücken (nur Fehlermeldungen)
--verbose	-v	Meldung pro transferierter Datei
2x --verbose	-vv	Meldung auch übersprungener Dateien
3x --verbose	-vvv	Meldung auch mit Debuginfo (sehr viel!)
--stats		Transfer-Statistik ausgeben
--progress	(-P)	Fortschritts-Meldungen ausgeben (impl. --verbose)
	-P	Fortschritts-Meldungen ausg. [--progress --partial]
--dry-run	-n	Aktionen nur anzeigen, nicht ausführen
--list-only		Dateien nur auflisten statt replizieren (ls-Ersatz)
--archive	-a	Archiv-Modus = -Dgloptr (-Do benötigen root-Rechte!)
	-D	Geräte + Spezialdateien transferieren (nur root!)
--devices	(-D)	Character/Blockgeräte transferieren (nur root!)
--specials	(-D)	Spezialdateien transferieren
--group	-g	Besitzer-Gruppe transferieren
--links	-l	Symbolische Links transferieren
--owner	-o	Besitzer transferieren (nur root!)
--perms	-p	Zugriffsrechte transferieren
--times	-t	Datum transferieren
--recursive	-r	Verzeichnis-Bäume replizieren
--executability	-E	Ausführungsrecht transferieren (in --perms drin)
--chmod=...		Ausführungsrecht für Dateien/Verz. setzen
--hard-links	-H	Harte Links erhalten (wichtig für Snapshot-Archive!)
--sparse	-S	Sparse Dateien erhalten (lange 0-Byte Sequenzen)
--acls	-A	ACLs transferieren (umfasst -p/--perms)
--xattrs	-X	Extended Attributes transf. (analog ACLs) ab 3.0.0
--rsh=PATH	-e	Zu benutzende "Remote Shell" (z.B. "ssh")
--compress	-z	Komprimierung der Datenübertragung
--delete		Dateien löschen, die auf dem Sender fehlen
--delete-before		Dateien VOR Transfer löschen (Std bis 2.6.9)
--delete-during		Dateien WÄHREND Transfer löschen (Std ab 3.0.0 --del)
--delete-after		Dateien NACH Transfer löschen
--delete-excluded		Vom Transfer ausgeschlossene Dateien auch löschen
--force		Nichtleere Verz. löschen die Sender fehlen
--max-delete=NUM		Max. NUM Dateien löschen
--existing		NUR bereits vorhandenen Dateien updaten
--ignore-existing		Bereits vorhandenen Dateien NICHT überschreiben
--update	-u	Neuere Dateien auf Zielseite NICHT überschreiben
--checksum	-c	VOR Transfer Änd. per Prüfsumme erk., NICHT Dat/Größe
--ignore-times	-I	Dateien gl. Größe/Datum NICHT überspringen (langsam!)
--whole-file	-W	Datei vollständig kop. = kein Delta-Algorithmus (Std. lok. Tr., Delta-Alg. nur für Netz-Transfer ok)
--exclude=MUSTER		Verz./Dateien gemäß MUSTER ausschließen
--exclude-from=FILE		Dateien in FILE ausschließen
--include=MUSTER		Dateien gemäß MUSTER einschließen
--include-from=FILE		Dateien in FILE einschließen
--filter=RULE	-f	Dateien gemäß RULE auswählen/ausschließen
--cvs-exclude	-C	CVS-Dateien ausschließen
--files-from=FILE		Liste zu übertragender Dateien aus FILE holen
--human-readable	-h	Größen in menschlich lesbarer Form ausgeben (K,M,G)
--8-bit-output	-8	Sonderzeichen NICHT im Escape-Format \#123 ausgeben
--copy-links	-L	Symbolische Links als Datei kopieren, NICHT erhalten
--one-file-system	-x	NUR im gl. Dateisystem transf. (Symlink NICHT folgen)
--numeric-ids		UID/GID statt User/Group-Namen transferieren (speed)
--omit-dir-times	-O	Änderungszeit für Verzeichnisse NICHT transf. (NFS!)
--modify-window=NUM		Datum identisch bei Abweichung <= NUM Sek. (Std: 0)
--bwlimit=NNN		Transferbandbreite beschränken (KByte/sec)
--append		Neue Daten direkt an Zieldatei anhängen (keine Zw.d.)
--append-verify		Analog (vorher bereits existierenden Teil prüfen)
--inplace		Zieldatei direkt ändern (keine Zwischendatei)
--partial	(-P)	Teilweise übertragene Dateien erhalten (Abbruch!)
--partial-dir=DIR		Teilweise übertragene Dateien in Verz. DIR ablegen
--delay-updates		Erst alles übertragen, dann Zieldateien ersetzen
--temp-dir=DIR	-T	Verz. für temporäre Dateien (Platzprobleme)
--max-size=SIZE		Dateien größer SIZE nicht transferieren

Nov 14, 11 3:00

rsync-HOWTO.txt

Page 8/8

<code>--min-size=SIZE</code>	Dateien kleiner SIZE nicht transferieren	
<code>--backup</code> <code>--backup-dir=DIR</code> <code>--suffix=SUFFIX</code>	Backup von geänderten Dateien machen Startverz. für Backup-Hierarchie Endung für <code>-b/--backup</code> (StdSuffix: "~")	
<code>--itemize-changes -i</code> <code>--out-format=FMT</code> <code>--log-file=FILE</code> <code>--log-file-format=.</code>	Übersicht aller Update-Änderungen (<code>--verbose</code>) Ausgabe pro Update-Änderung festlegen Ausgaben in Logdatei FILE schreiben Format der Ausgabe in Logdatei festlegen	
<code>--super</code> <code>--fake-super</code>	Empfänger versucht als Super-User "root" zu arbeiten Emuliert Rechte die nur "root" setzen kann durch ACLs	
<code>--copy-dirlinks -k</code> <code>--keep-dirlinks -K</code> <code>--dirs -d</code>	Symlink auf Verz. transferieren Symlink auf Verz. auf Empfänger als Verz. behandeln Verz. ohne Inhalt übertragen (oder <code>--recursive</code>)	
<code>--no-inc-recursive</code>	Dateil. ERST vollst. erz., DANN übertragen (ab 3.0.0)	
<code>--iconv=SPEC</code> <code>--protect-args -s</code>	Dateinamen-Zeichensatz konvertieren Remote-Sh belässt Metazeichen (Leerz.)	ab 3.0.0 ab 3.0.0

- * Option `--cvs-exclude` lässt Dateien + Verz. analog der CVS-Ausschlussliste weg:
RCS SCCS CVS CVS.adm RCSLOG cvslog.* tags TAGS
.make.state .nse_depinfo *~ #* .*#* ,*_ \$* *\$ *.old *.bak
*.BAK *.orig *.rej .del-* *.a *.olb *.o *.obj *.so *.exe
*.Z *.elc *.ln core .svn/ .git/ .bzz/
- * Option `--human-readable` liefert Einheiten K, M, G als Vielfaches von 1000.
Wird sie zweimal angegeben, dann als Vielfaches von 1024.
- * Option `--iconv=LOCAL,REMOTE` (z.B. `--iconv=utf8,iso88591`) wandelt die Zeichensatz-Codierung der Dateinamen beim Transfer um (ab 3.0.0).
Der Befehl `iconv --list` liefert eine Liste der möglichen Zeichensätze.
- * Option `--protect-args` verhindert die Interpretation aller Dateinamen und der meisten Optionen durch die Shell auf Remote-Seite. Leerzeichen zerlegen keine Dateinamen mehr und alle speziellen Zeichen ausser den Wildcards bleiben erhalten (z.B. `~ $; &`)
- * Option `--super` versucht die Wirkung der Optionen `--owner`, `--groups` und `--devices` auf Remote-Seite auszuführen, auch wenn der (lokale) Benutzer dort kein "root" ist (falls das möglich ist, ansonsten Fehlermeldung).
- * Option `--fake-super` speichert auf Remote-Seiten nicht setzbare Dateiattribute (Besitzer, Besitzergruppe, Geräte, Zugriffsrechte) in ACLs, um sie zumindest aufzuzeichnen.