

HOWTO zu rsync (Remote Synchronisation)

(C) 2008-2014 T.Birnthal/H.Gottschalk <[howtos\(at\)ostc.de](mailto:howtos(at)ostc.de)>
OSTC Open Source Training and Consulting GmbH
<http://www.ostc.de>

Version 1.46 (28.08.2014)

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

HINWEIS: Die Begriffe "Verzeichnis" und "Verzeichnisse" werden wann immer sinnvoll als "Verz." abgekürzt

INHALTSVERZEICHNIS

- 1) Einführung
 - 1.1) Hauptspeicherbedarf
- 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
- 8) Tips

1) Einführung

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

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
```

1.1) Hauptspeicherbedarf

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 3.1.1 protocol version 31".

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	Geräte + Spezialdateien transferieren (nur root!)
--links -g	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 inkl. Opt.)
--rsync-path=PATH	Zu benutzendes remote rsync-Kommando (/usr/bin/rsync)
--compress -z	Komprimierung der Datenübertragung (lokal sinnlos!)
--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, -z/--compress):


```
rsync -a SRC/ DEST # Lokal replizieren
rsync -e ssh -z -a SRC/ USER@HOST:DEST # Remote über SSH replizieren
rsync -e "ssh -p 6300" -a SRC/ USER@HOST:DEST # R. über SSH auf Port 6300
rsync -e ssh USER@HOST:DEST # Remote-Dateien auflisten
rsync -e ssh -HSAX -z -a 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 Extending Attribute ebenfalls 1:1 repliziert werden, so sind diese Optionen zusätzlich zu -a/--archive anzugeben (siehe oben).
- * Option -e/--rsh startet ein Kommando inkl. Optionen für den Aufbau der Verbindung zum Remote-Rechner und den Start eines Remote-Logins auf dem Remote-Rechner. Beispiele:


```
rsync -e "ssh" HOST:/PATH/FILE . # SSH
rsync -e "ssh -l ssh-user" HOST:/PATH/FILE . # Login-User
rsync -e "ssh -p 2234" HOST:/PATH/FILE . # SSH-Port 2234 statt 22
rsync -e 'ssh -o "ProxyCommand nohup ssh firewall nc -w1 %h %p"\' \
HOST:/PATH/FILE .
```

3) Weitere wichtige Optionen

Opt (lang+kurz)	Bedeutung
--delete	Dateien löschen, die auf Sender fehlen
--delete-during	Dateien WÄHREND Transfer löschen (analog --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!)

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 3/9

--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 entspricht sie --delete-during (performanter). Sie kann mit einer der --delete-WHEN Optionen kombiniert werden.
- * Option --del entspricht --delete-during.
- * Option --delete-after löscht erst nach dem erfolgreichen Übertragen aller Dateien die zu entfernen.
- * 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 (--hard-links/--sparse/--acls/--xattrs) replizieren weitere wichtige Eigenschaften von Dateien sofern beide Dateisysteme diese beherrschen (Hardlinks, Sparse-Dateien mit "Löchern", Access Control Lists, Extended Attributes).
- * Optionen -H/--hard-links und -S/--sparse 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 der Datenübertragung (lokal sinnlos!) |
-H -S -A -X      # --hard-links --sparse --acls --xattrs
-x --one-file-system # NUR im gl. Dateisystem transf. (Symlink NICHT folgen)|
--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       # Fortschrittsmeldungen während Transfer
--stats          # Statistik am Schluss
-i --itemize-changes # Übersicht aller Update-Änderungen (--verbose)
--human-readable # Dateigrößen in menschlich lesbarer Form (K,M,G)
--8-bit-output   # Sonderzeichen NICHT im Escape-Format \#123
-c              # --checksum (immer Prüfsumme statt Datum+Größe vgl.)
-I              # --ignore-times (trotz Datum+Größe gleich transf.)
-W              # --whole-file (kein Delta-Algorithmus)
--partial        # Teilweise übertragene Dateien erhalten (Abbruch!)
--inplace        # Zielfeile direkt ändern (keine Zwischendatei)
--append         # Zielfeile kürzer -> sofort anhängen (keine Zw.datei)
--append-verify  # Analog (vorher bereits existierenden Teil prüfen)
```

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

```
rsync -r SRC/ DEST # --> SRC/* --> DEST/*
rsync -r SRC DEST # --> SRC/* --> DEST/SRC/*
```

- * Bei Windows-Rechnern sollte --modify-window=2 gesetzt werden, das eine Differenz <= 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"

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 4/9

+ "nodirctime" 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,nodirctime,async /dev/.../SNAPNAME MOUNTPPOINT
rsync MOUNTPPOINT/ DESTINATION
umount MOUNTPPOINT
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
```

TIP: "/" 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 eine 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
```

```
# Gestern -> Vorgestern
```

```
mv /backup/1 /backup/2
```

```
# Gestern -> Gestern
```

```
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

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 5/9

Ausgangsverzeichnis 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":

Option (lang+kurz)	Bedeutung
--archive -a	Archiv-Modus = -DgloptR (-Do benötigen root-Rechte!)
--devices -D	Geräte + Spezialdateien transferieren (nur root!)
--specials (-D)	Character/Blockgeräte transferieren (nur root!)
--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

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 6/9

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

* 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):

```
...
  32.77K 100%  30.04MB/s   0:00:00 (xfer#10, to-check=14446/19441)
.mozilla/firefox/sjldutp.default/places.sqlite-wal
  601.55K 100%   5.57MB/s   0:00:00 (xfer#11, to-check=14445/19441)
.mozilla/firefox/sjldutp.default/prefs.js
  663.71K 100%   2.67MB/s   0:00:00 (xfer#12, to-check=14443/19441)
...
```

* Option --stats gibt am Schluss eine Zusammenfassung der Kenngrößen des Transfers in folgender Form aus:

```
Number of files: 11821
Number of files transferred: 314
Total file size: 45.84G bytes
Total transferred file size: 99.18M bytes
Literal data: 190.75K bytes
Matched data: 98.99M bytes
File list size: 227.66K
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 267.73K
Total bytes received: 297.86K
sent 267.73K bytes received 297.86K bytes  49.18K bytes/sec
total size is 45.84G speedup is 81048.17
SRCDIR: Mon May 18 12:59:13 CEST 2009 - Mon May 18 12:59:45 CEST 2009
DESTDIR: Mon May 18 12:59:45 CEST 2009 - Mon May 18 12:59:56 CEST 2009
```

* Option -P ist die Zusammenfassung von --partial + --progress.

* Option --itemize-changes / -i gibt für jede Datei eine Änderungsliste inkl. Attributänderungen aus und entspricht exakt --out-format='%i %n%L'. %n ist der Dateiname, %L ist "-> SYMLINK", "-> HARDLINK" oder "" falls die Datei kein Link ist) und Format %i erzeugt folgende 11 Zeichen:

YXcstpoguax		
UPDATE TYP	DATEI TYP	ATTRIBUTE TYP
< Gesendet -> remote	f Datei	c Prüfsumme
> Empfangen -> lokal	d Verzeichnis	s Größe
c Lokale Änderung	L Symbol. Link	t Zeit
h Hardlink	D Gerät	p Zugriffsrecht
. Kein Update	S Spezial	o Besitzer

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 7/9

* Nachricht (Named Socket/
Fifo) g Gruppe
u Reserviert
a ACL
x Erweiterte Attribute

* Option --verbose verwendet --out-format="%n%L" und ist ohne Option
--itemize-changes / -i fast wertlos,

* Als Formatkürzel in --out-format sind möglich (siehe "man rsyncd.conf")

ESC	Bedeutung
%a	IP-Adresse des Remoterechners
%b	Anzahl tatsächlich übertragener Byte
%B	Zugriffsrechte der Datei (z.B. rwxrwxrwt)
%c	Gesamtgröße der Blockprüfsummen für Basisdatei (gesendet)
%f	Dateiname (lange Form auf dem Sender, kein "/" nach Verz.)
%G	GID der Datei (dezimal) oder "DEFAULT"
%h	Hostname des Remote-Rechners
%i	Punktliste der upgedateten Dateieigenschaften
%l	Dateilänge in Byte
%L	Text "-> SYMLINK", " => HARDLINK" oder ""
%m	Modulename
%M	Datum+Uhrzeit der letzten Dateiänderung
%n	Dateiname (Kurzform, "/" nach Verz.)
%o	Operation ("send", "recv" oder "del.")
%p	PID der rsync Sitzung
%P	Modulpfad
%t	Aktuelles Datum+Uhrzeit
%u	Benutzername (authentifiziert) oder leerer Text
%U	UID der Datei (dezimal)

Default log format:

```
"%o %h [%a] %m (%u) %f %l"
```

Option --log-file fügt folgenden Präfix hinzu:

```
"%t [%p] "
```

7c) Vollständige Liste der Optionen

Option (lang+kurz)	Bedeutung
--help -h	Hilfe anzeigen
--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 = -Dglopt (-Do benötigen root-Rechte!)
--devices -D	Geräte + Spezialdateien transferieren (nur root!)
(-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

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 8/9

--rsh=PATH -e	Zu benutzende "Remote Shell" (z.B. "ssh")
--rsync-path=PATH	Zu benutzendes remote rsync-Kommando (/usr/bin/rsync)
--compress -z	Komprimierung der Datenübertragung (lokal sinnlos!)
--delete	Dateien löschen, die auf Sender fehlen
--delete-before	Dateien VOR Transfer löschen (Std bis 2.6.9)
--del[ete-during]	Dateien WÄHREND Transfer löschen (Std ab 3.0.0)
--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	Zieldatei kürzer -> sofort anhängen (keine Zw.datei)
--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
--min-size=SIZE	Dateien kleiner SIZE nicht transferieren
--backup -b	Backup von geänderten Dateien machen
--backup-dir=DIR	Startverz. für Backup-Hierarchie
--suffix=SUFFIX	Endung für -b/--backup (StdSuffix: "~")
--itemize-changes -i	Übersicht aller Update-Änderungen (--verbose)
--out-format=FMT	Ausgabe pro Update-Änderung festlegen
--log-file=FILE	Ausgaben in Logdatei FILE schreiben
--log-file-format=.	Format der Ausgabe in Logdatei festlegen
--super	Empfänger versucht als Super-User "root" zu arbeiten
--fake-super	Emuliert Rechte die nur "root" setzen kann durch ACLs
--copy-dirlinks -k	Symlink auf Verz. transferieren
--keep-dirlinks -K	Symlink auf Verz. auf Empfänger als Verz. behandeln
--dirs -d	Verz. ohne Inhalt übertragen (oder --recursive)
--no-inc-recursive	Dateil. ERST vollst. erz., DANN übertragen (ab 3.0.0)
--iconv=SPEC	Dateinamen-Zeichensatz konvertieren ab 3.0.0
--protect-args -s	Remote-Sh belässt Metazeichen (Leerz.) ab 3.0.0
Neue Optionen von rsync V3.1.1 (Preallokation)	
--info=FLAGS	Fine-grained informational verbosity
--debug=FLAGS	Fine-grained debug verbosity
--msgs2stderr	Special output handling for debugging
--munge-links	Munge symlinks to make them safer (but unusable)
--omit-link-times -J	Omit symlinks from --times
--preallocate	Allocate dest files before writing them

Aug 29, 14 3:00

rsync-HOWTO.txt

Page 9/9

--ignore-missing-args	Ignore missing source args without error
--delete-missing-args	Delete missing source args from destination
--usermap=STRING	Custom username mapping
--groupmap=STRING	Custom groupname mapping
--chown=USER:GROUP	Simple username/groupname mapping
--remote-option=OPT -M	Send OPT to the remote side only
--bwlimit=RATE	Limit socket I/O bandwidth
--outbuf=N L B	Set output buffering to None, Line, or Block
--checksum-seed=NUM	Set block/file checksum seed (advanced)

- * 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.

8) Tips

- * Folgende Fälle sind bei Verz. SRC und DEST zu unterscheiden:

```

rsync SRC DEST # A) Kop. "SRC/Inhalt" nach DEST (Unterverz. SRC in DEST)
rsync SRC/ DEST # B) Kop. nur "Inhalt" nach DEST (KEIN Unterverz. SRC in DEST!)
rsync SRC/* DEST # C) Analog B), allerdings ohne .-Dateien/Verz.
rsync SRC DEST/ # Analog A)
rsync SRC/ DEST/ # Analog B)
rsync SRC/* DEST/ # Analog C)

```
- * Fehlender Doppelpunkt beim Ziel führt zu einer lokalen Kopie
(häufiger Fehler):

```

rsync SRC USER@HOST

```
- * Ohne Zielverz. ist das Heimatverz. des Benutzers USER das Ziel:

```

rsync SRC USER@HOST: #
rsync SRC USER@HOST:. # Analog

```
- * Ein relativer Zielpfad bezieht sich auf das Heimatverz. des Benutzers USER:

```

rsync SRC USER@HOST:PATH # Ohne "/" am Anfang von PATH
rsync SRC USER@HOST:./PATH # Analog

```
- * --rsync-path=/PFAD/ZU/RSYNC verwenden, um beim remote Synchronisieren per
-e ssh / --rsh=ssh auf der Gegenseite eine bestimmte rsync-Version aufzurufen.
(sonst wird "/usr/bin/rsync" aufgerufen).
- * -S/--sparse und --inplace sind nicht gemeinsam verwendbar.