HOWTO zu rsync (Remote Synchronisation)

(C) 2008-2023 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: rsync-HOWTO.txt,v 1.77 2025/03/17 13:48:45 tsbirn Exp $

Dieses Dokument beschreibt die lokale und die 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 Vollsicherung mit "rsync + cp"
7) Optionen
7a) Weitere interessante Optionen
7b) Weitere interessante Optionen
7c) Vollständige Liste der Optionen
8) Tipps
9) Links

---

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 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 -e ssh [OPT] [USER@]HOST:SRC DEST    # Von Remote replizieren
  rsync -e ssh [OPT] [USER@]HOST:SRC         # Remote auflisten (analog "ls -l")
```

ACHTUNG: Es ist ein FUNDAMENTALER Unterschied, ob der Quell-Verzeichnisname
"SRC" mit "/" abgeschlossen wird oder NICHT:

```
  rsync -r SRC/ DEST   # SRC/* --> DEST/*     --> INHALT von Verz. SRC nach DEST
kopiert
  rsync -r SRC  DEST   # SRC/* --> DEST/SRC/* --> Verz. SRC nach DEST kopiert
```

D.h. bei "/" am Ende wird nicht der INHALT des Verz. "SRC", sondern das Verz.
selbst repliziert (d.h. Verz. "SRC" auf der Zielseite unter "DEST" sichtbar):

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 sind 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 sychronisierende 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 3.4.1 protocol version 32" (15. Jan 2025)

2) Die wichtigsten Optionen

Option (lang+kurz)	Bedeutung
--help	Hilfe anzeigen
--archive -a	Archiv-Modus = -Dgloptr (-Do benötigen root-Rechte!) device group links owner permission times recursive
-D	Geräte + Spezialdateien 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	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!)
--remote-option= -M	Optionen nur auf Zielseite setzen

* Typische 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 repl.
    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 ignorieren identischer Dateien
  auf 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
  Extendend 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 --del	Dateien löschen, die auf Sender fehlen
--delete-during	Dateien WÄHREND Transfer löschen (STD ab V3.0.0)
--delete-before	Dateien VOR Transfer löschen (STD bis 2.5.9)
--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)
--copy-dirlinks -k	Sender-Symlink auf Verz. in echtes Zielverz. umwand.
--keep-dirlinks -K	Empfänger-Symlink auf Verz. als echtes Verz. behand.

* VORSICHT bei 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 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 (--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).

* Option -k/--copy-dirlinks sorgt dafür, dass der Sender einen Symlink auf
  ein Verz. als ECHTES Verz. behandelt. Symlinks auf alles außer Verz. bleiben
  erhalten. OHNE diese Option wird beim Ersetzen eines Verz. durch einen
  Symlink auf Senderseite die Empfängerseite ALLES löschen, was dem Symlink
  im Weg steht (auch einen kompletten Verz.baum).

* Option -K/--keep-dirlinks sorgt dafür, dass der Empfänger einen Symlink auf
  ein Verz. als ECHTES Verz. behandelt (nur wenn er ein echtes Verz. auf dem
  Sender referenziert). OHNE diese Option wird der Empfänger-Symlink entfernt
  und durch ein echtes Verz. ersetzt.

4) Hinweise
----
* Häufig benötigte Aufrufe und Optionen fertig zum Kopieren per Maus:

```
    rsync SRC           USER@HOST:DEST     # Verz. "SRC" mit Inhalt nach "DEST"
kopieren
    rsync SRC/          USER@HOST:DEST     # Inhalt von Verz. "SRC" nach "DEST"
kopieren
    rsync SRC           DEST               # Von Localhost nach Localhost
    rsync SRC           USER@HOST:DEST     # Von Localhost nach Remote-HOST
    rsync USER@HOST:SRC DEST               # Von Remote-HOST nach Localhost
    rsync USER@HOST:SRC USER2@HOST2:DEST   # Von Remote-HOST nach Remote-HOST2
    -n --dry-run          # Aktionen nur anzeigen, NICHT ausführen (zum TEST!)
    -e ssh                # Remote über SSH-Verbindung anmelden + transferieren
    -a                    # --archive = -g -l -p -t -r -D -o
    -g -l -p -t -r        # --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 transferieren
    -E --executability    # Apple/OSX resource forks (extended attr.) transfer.
    -x --one-file-system  # NUR im gl. Dateisystem transf. (Symlink NICHT folgen)
    --del                 # --delete-during
    --exclude=/lost+found # Mehrfach erlaubt
    --exclude="lost+found"# Mehrfach erlaubt
    --numeric-ids         # UID/GID statt User/Group-Namen transferieren (speed)
    --bwlimit=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          # 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,T)
    --8-bit-output        # Sonderzeichen NICHT im Escape-Format \#123
    --block-size=2048     # Feste Prüfsummen-Blockgröße (Byte)
    --checksum-seed=32761 # Startwert für Block/Datei-Prüfsumme
    -c --checksum         # Immer Prüfsumme statt Datum+Größe vergleichen
    -I --ignore-times     # Trotz Datum+Größe gleich transferieren
    -W --whole-file       # Keinen Delta-Algorithmus verwenden
    --partial             # Teilweise übertragene Dateien erhalten (Abbruch!)
    --inplace             # Zieldatei direkt ändern (keine Zwischendatei)
    --append              # Zieldatei kürzer --> anhängen (keine Zw.datei)
    --append-verify       # Analog (vorher bereits existierenden Teil prüfen)
    -P                    # --partial --progress
```

* ACHTUNG: "/" 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   # ergibt: SRC/* --> DEST/*
   rsync -r SRC  DEST   # ergibt: 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" + "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
```

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 vollständig 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":

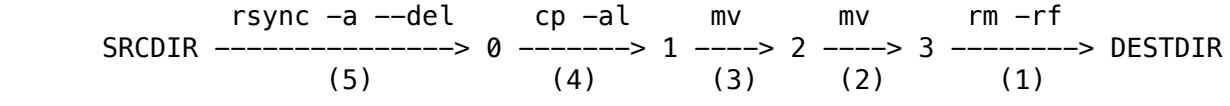
```
  # 1) Heimat-Verz. nach Rechner "backup" synchronisieren
  USER="dummy"
  HOST="backup"
  cd
  rsync -avzHSAX --del -e ssh . "$USER@$HOST:."

  # 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 Vollsicherung mit "rsync + cp"

Man kann "rsync" und "cp" kombinieren, um per Hard-Links eine rollierende mehrfache VOLLSICHERUNG eines Dateibaums zu erstellen, ohne dafür den mehrfachen Platz zu verbrauchen. Jede Datei ist physisch nur 1x vorhanden (mit ihren Metadaten), auch wenn sie in vielen Vollsicherungen enthalten ist.



Prinzipielle Vorgehensweise (ACHTUNG: "/" am Ende ist notwendig!):

```
  mkdir -p /backup/{0,1,2,3}        # 1x notwendig (path)
  rm -rf /backup/3                  # 1) Vorvorgestern löschen (recursive, force)
  mv     /backup/2 /backup/3        # 2) Vorgestern --> Vorvorgestern
  mv     /backup/1 /backup/2        # 3) Gestern    --> Vorgestern
  cp -al /backup/0 /backup/1        # 4) Aktuell    --> Gestern (archive, hardlinks!)
                                    #    (keine echte Kopie sondern per Hardlinks)
                                    #    (-a=-dR --preserve=all  -d=--no-dereference)
  rsync -a --del SRCDIR /backup/0/  # 5) 1:1-Kopie + gelöschte Dateien entfernen
                                    #    (-a=rlptgoD)
```

Durch Schritt 4) wird eine "Kopie" von Backup "0" in Backup "1" per Hardlinks erzeugt (d.h. keine echte Datenkopie erstellt, sondern nur Verweise dupliziert).

Durch Schritt 5) werden neue/geänderte/gelöschte Dateien vom ORIGINAL in Backup "0" kopiert. Bereits vorhandene "identische" Dateien werden dabei nicht mehr übertragen. Von geänderten Dateien wird nur der Unterschied übertragen. Die per "--del" gelöschte Dateien sind erst dann wirklich gelöscht, wenn sie in keinem der Backups "0", "1", "2" mehr per Link adressiert werden.

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

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

Vollständig 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!)
-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-Algorith. (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
--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	Am Schluss 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 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):

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

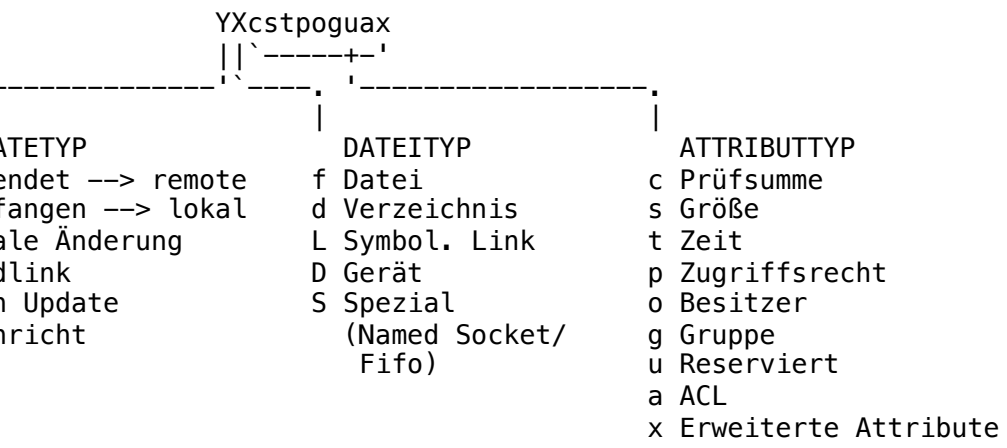
* Option --stats gibt am Schluss eine Zusammenfassung der Kenngrößen des Transfers in folgender Form aus:

```
  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:



* 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	Punkteliste 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 (--itemize-changes)
2x --verbose -vv	Meldung mit übersprungenen Dateien
3x --verbose -vvv	Meldung mit Debuginfo (sehr viel!)
--remote-option= -M	Optionen nur auf Zielseite setzen
--stats	Am Schluss 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")
--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=N	Max. N 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=N -@	Datum identisch bei Abweichung <= N Sek. (Std: 0)

| --bwlimit=N | Transferbandbreite beschränken (KByte/sec) |
| --outbut=N\|L\|B | Ausgabe-Puffer auf None, Line, Block setzen |

--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	Sender-Symlink auf Verz. in echtes Zielverz. umwand.
--keep-dirlinks -K	Empfänger-Symlink auf Verz. als echtes Verz. behand.
--dirs -d	Verz. ohne Inhalt übertragen (oder --recursive)

| --no-inc-recursive | Dateil. ERST vollst. erz., DANN übertragen (ab 3.0.0) |

| --iconv=LOCAL,REMOTE | 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
--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=N      | Set block/file checksum seed (advanced)           |
  +------------------------+---------------------------------------------------+
```

* Option --cvs-exclude lässt Dateien + Verz. analog der CVS-Auschlussliste 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/  .bzr/
```

* Option --human-readable liefert Einheiten K, M, G, T als Vielfaches von 1000. Wird sie 2x angegeben, dann als Vielfaches von 1024.

* Option --iconv=LOCAL,REMOTE (z.B. --iconv=utf8,iso88591 oder --iconv=UTF-8,UTF8-MAC) 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-Seite nicht setzbare Dateiattribute (Besitzer, Besitzergruppe, Geräte, Zugriffsrechte) in ACLs, um sie zumindest aufzuzeichnen.

8) Tipps

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

```
    rsync SRC   DEST   # A) Kop. "SRC/Inhalt" nach DEST (Verz. SRC in DEST)
    rsync SRC   DEST/  # A) Analog
    rsync SRC/  DEST   # B) Kop. SRC-"Inhalt" nach DEST (KEIN Verz. SRC in DEST)
    rsync SRC/  DEST/  # B) Analog
    rsync SRC/* DEST   # C) Analog B), ohne Konfig.-Dateien/Verz. ".XXX"
    rsync SRC/* DEST/  # C) Analog
```

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

```
    rsync SRC USER@HOST    # Falsch (lokale Kopie)
    rsync SRC USER@HOST:   # OK
```

* Ohne Zielverz. oder bei relativem Zielverz. ist das Heimatverz. des (lokalen) Benutzers USER das Ziel:

```
    rsync SRC USER@HOST:             # (":" nicht vergessen!)
    rsync SRC USER@HOST:.            # Analog (selber Benutzer wie lokal)
    rsync SRC USER@HOST:~            # Analog (selber Benutzer wie lokal)
    rsync SRC USER@HOST:~USER2       # Analog (Benutzer USER2 remote)
    rsync SRC USER@HOST:/home/USER2  # Analog (Benutzer USER2 remote)
```

* Relativer Zielpfad bezieht sich auf Heimatverz. des (lokalen) Benutzers USER:

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

* Mit --rsync-path=/PFAD/ZU/RSYNC wird beim remote Synchronisieren per
  -e ssh/--rsh=ssh auf der Gegenseite die rsync-Version /PFAD/ZU/RSYNC
  aufgerufen (Std: "/usr/bin/rsync").

* -S/--sparse und --inplace sind nicht gemeinsam verwendbar!

9) Links
----

http://en.wikipedia.org/wiki/Rsync	Rsync (Wikipedia Artikel, eng)
http://rsync.samba.org/	Rsync (Quellcode + Dokumentation)
http://www.samba.org/~tridge/phd_thesis.pdf	Efficient Algorithms for Sorting and Searching (Doktorarbeit von rsync-Erfinder Andrew Tridgell)
http://www.mikerubel.org/computers/rsync_snapshots/	Easy Automated Snapshot (Snapshot-Style Backups with Linux and Rsync)
http://www.rsnapshot.org/	Rsnapshot (inkrementelle Snapshots per rsync, ssh und cp)
http://www.opbyte.it/grsync/	Grsync (GUI für Rsync)
http://everythinglinux.org/rsync/rsync_content.html	Rsync (A Tutorial on Using Rsync)
http://github.com/schlomo/rbme/	RBME (Rsync Backup Made Easy)
http://samba.anu.edu.au/rsync/how-rsync-works.html	How Rsync Works (A Practical Overview)
http://www.rsync-backup.de/	Rsync-Backup (Effiziente Sicherung großer Datenmengen über das Internet)
http://backuppc.sourceforge.net/	BackupPC (GUI + rsync für Backup)

==========