

rvs[®]

rvsbat

Batch- Schnittstelle

Beschreibung

Die in diesem Dokument aufgeführten Produkte sind urheberrechtlich geschützt und stehen dem jeweiligen Rechtsinhaber zu.

rvs®

Version 1.0

Beschreibung

© 2005 by gedas deutschland GmbH

Pascalstraße 11

10587 Berlin

Das vorliegende Dokument ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne Genehmigung von gedas in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache übertragen werden. Auch die Rechte der Wiedergabe durch Vortrag, Funk und Fernsehen sind vorbehalten.

Inhaltliche Änderungen dieses Dokuments behalten wir uns ohne Ankündigung vor. gedas haftet nicht für technische oder drucktechnische Fehler oder Mängel in diesem Dokument. Außerdem übernimmt gedas keine Haftung für Schäden, die direkt oder indirekt auf Lieferung, Leistung und Nutzung dieses Materials zurückzuführen sind.

Inhaltsverzeichnis

| | |
|--|-----------|
| Inhaltsverzeichnis..... | 3 |
| 1 Repräsentationsmittel und Darstellungskonventionen..... | 7 |
| 2 rvs® und seine Schnittstellen im Überblick | 9 |
| 3 Die Kommandozeilen-Schnittstelle rvsbat..... | 11 |
| 4 Starten der Batch-Schnittstelle (Kommandozeilen-Schnittstelle) rvsbat. Verwendete Parameter und korrespondierende rvs®- | |
| Kommandos | 12 |
| 4.1 Die rvs® Kommandozeilen-Schnittstelle (rvsbat) starten | 13 |
| Kommandozeilen-Parameter..... | 13 |
| Kommandobehandlung | 14 |
| 5 Beschreibung der Kommandos | 15 |
| 5.1 Syntax der Kommandos | 16 |
| 5.2 Das Kommando START..... | 17 |
| 5.3 Das Kommando END | 18 |
| 5.4 Das Kommando SEND | 19 |
| 5.5 Das Kommando RESENTR | 25 |
| 5.6 Kommando SENDJOB | 30 |
| 5.7 Das Kommando USER..... | 33 |
| 5.8 Das Kommando ACTIVATE | 35 |
| 5.9 Das Kommando MODST..... | 36 |
| 5.10 Das Kommando DELST | 37 |
| 5.11 Das Kommando LISTPARM..... | 38 |
| 5.12 Das Kommando SETPARM..... | 39 |

Das vorliegende Dokument enthält eine Beschreibung der rvs[®]-Schnittstelle `rvsbat`.

Zu Beginn des Dokuments finden Sie eine Übersicht der Repräsentationsmittel und Darstellungskonventionen, die im Text verwendet wurden.

Kapitel 2 bietet einen Gesamtüberblick zu allen rvs[®]-Schnittstellen.

Nach einer Beschreibung der Einbettung der `rvsbat`-Schnittstelle in die rvs[®]-Struktur im Kapitel 3, stellen Ihnen die Kapitel 4 und 5 die Möglichkeiten, die Ihnen die `rvsbat`-Schnittstelle bietet, sowie sämtliche Kommandos im Detail vor.

1 Repräsentationsmittel und Darstellungskonventionen

Dieser Abschnitt enthält die Beschreibung, welche Darstellungskonventionen in diesem Handbuch verwendet werden und welche Bedeutung besonders gekennzeichnete Ausdrücke haben.

Darstellungskonventionen

| | |
|---------------------------------------|---|
| <code>courier</code> | Kommandos, Menübefehle, Dateinamen, Pfadnamen, Programme, Beispiele, Script-Dateien, Optionen, Qualifiers, Datensätze, Felder, Modi, Fensternamen, Dialogboxen und Status |
| FETT und GROSS- BUCHSTABIG | Parameter, Umgebungsvariablen, Variablen |
| "Hochkommata" | Verweise auf andere Handbücher, Kapitel und Abschnitte, Literatur |
| fett | wichtige Begriffe, Betriebssystemnamen, Eigennamen, Schaltflächen (Buttons), Funktionstasten |

Begriffe

rvsX ist das Synonym für rvs[®] auf **UNIX** Systemen.

rvsNT ist das Synonym für rvs[®] auf **Windows NT** Systemen.

rvsXP ist das Synonym für rvs[®] auf **Windows XP** Systemen.

rvs400 ist das Synonym für rvs[®] auf **OS/400** Systemen.

Verzeichnisse

Weil Benutzerverzeichnisse auf unterschiedlichen Plätzen bei den unterschiedlichen Betriebssystemen zu finden sind, benutzen wir

in diesem Handbuch die Variable **\$RVSPATH**. Die Standardwerte sind:

- /home/rvs/ für **AIX, Solaris, IRIX, Linux** und **SCO**
- /users/rvs/ für **HP-UX**
- /defpath/rvs/ für **SINIX**
- C:\Programme\rvs[®] für **Windows NT** und **Windows XP**

Ersetzen Sie diese Variable durch Ihren richtigen Pfad.

Dateinamen bei **OS/400** Systemen werden immer groß geschrieben.

2 r^{vs}® und seine Schnittstellen im Überblick

r^{vs}® bietet einen zuverlässigen Datenübertragungsdienst, in dem es als eigenständiges Werkzeug Daten senden, empfangen und verteilen kann. r^{vs}® kann in Anwendungen für die Automatisierung des Datenaustausches zwischen Netzknoten und Benutzern integriert werden. Typische Anwendungsbereiche von r^{vs}® sind EDI (**E**lectronic **D**ocument **I**nterchange), CAD (**C**omputer **A**ided **D**esign), Finanztransaktionssysteme und sichere Übertragung von Stammdaten und Daten von Medienkonzernen.

Für verschiedene Grade der Automatisierung sind passende Schnittstellen zu r^{vs}® für die Erzeugung von Sende- und Empfangseinträgen verfügbar:

| | |
|---|--|
| Dialog Schnittstelle rvsdia | Ist ein interaktives Werkzeug für die Erzeugung jeweils einzelner Einträge; Abfragefunktionen informieren Sie über den Status Ihrer Anforderungen. |
| Batch-Schnittstelle (auch: Kommandozeilen-Schnittstelle) rvsbat | Liest Kommandos aus einer Datei. Diese Eingabedatei ist eine einfache Textdatei, die mit einem beliebigen Editor vorbereitet oder als Ausgabedatei eines Anwendungsprogrammes erzeugt werden kann. |
| C-CAL-Schnittstelle rvscal | Erlaubt Anwendungsprogrammen durch Aufrufe von Funktionen der Programmiersprache C die direkte Generierung von r ^{vs} ® Kommandoeinträgen. |
| J-CAL-Schnittstelle | Diese Java-Schnittstelle wurde im Zusammenhang mit dem r ^{vs} Client Server, einer netzwerkfähigen Erweiterung von r ^{vs} ® portable, entwickelt. |
| XML-Schnittstelle | XML-Schnittstelle ermöglicht Export und Import von r ^{vs} ® bezogenen Daten im XML-Format. |

rvs[®] analysiert die Sendeeinträge, erzeugt Sendekommandos und startet die Sendeprozesse, die Ihre Dateien an den oder die richtigen Nachbarknoten im OFTP-Netzwerk für die Zustellung oder die Weiterleitung (Routing) übertragen. Der jeweilige Sendeeintrag bleibt im Wartezustand, bis eine positive Bestätigung vom Empfänger (EERP) kommt.

Empfangsprozesse nehmen eintreffende Dateien an und legen sie in temporären Dateien ab. Danach werden die Dateien dem lokalen Empfänger zugestellt oder wenn nötig, werden passende Sendeeinträge für die Weiterleitung an ferne Stationen erzeugt.

Die Zustellung eintreffender Dateien kann durch residente Empfangseinträge beeinflusst werden, die rvs[®] mitteilen, unter welchem Namen die Datei gespeichert werden soll und ob ein Job für die Weiterverarbeitung der eintreffenden Daten gestartet werden soll.

Jede der oben vorgestellten Schnittstellen erzeugt je einen Eintrag für jede Datei.

3 Die Kommandozeilen-Schnittstelle rvsbat

Die Kommandozeilen-Schnittstelle ermöglicht es, Aufträge als eine einzelne Kommandozeile auf der Ebene der Kommandosprache des entsprechenden Systems einzugeben. Ein solches Kommando kann leicht in die Liste der Kommandoprozeduren aufgenommen werden. Es kann entweder interaktiv genutzt werden oder in der Kommandoliste eines Datensets, das im Stapelverarbeitungsmodus ausgeführt wird.

4 Starten der Batch-Schnittstelle (Kommandozeilen-Schnittstelle) `rvsbat`. Verwendete Parameter und korrespondierende `rvs`[®]-Kommandos

Dieses Kapitel beschreibt, wie Sie die `rvsbat`-Schnittstelle starten und welche Parameter Sie verwenden können. Außerdem erhalten Sie eine Beschreibung der korrespondierenden globalen Kommandos von `rvs`[®].

Für alle Aufgaben (Senden und Empfangen von Dateien ebenso wie die Administration von `rvs`[®]), die Sie mit `rvs`[®] zu erledigen haben, können Sie entweder

- Kommandos in eine Textdatei schreiben und an die Kommandozeilen-Schnittstelle (`rvsbat`) übergeben,
- Kommandos an die Funktion `rvscal()` der C-Cal-Schnittstelle übergeben oder
- C-Funktionen für die C-Cal-Schnittstelle benutzen (für jedes Kommando eine eigene Funktion)

Die Kommandos von `rvsbat` (Kommandozeilen-Schnittstelle) sind identisch mit denen von `rvscal()` (C-Cal-Schnittstelle) und werden im **Kapitel 5: Beschreibung der Kommandos** erläutert.

4.1 Die rvs® Kommandozeilen-Schnittstelle (rvsbat) starten

Die Kommandozeilen-Schnittstelle kann aufgerufen werden durch:

```
rvsbat [/c] [/e<envdsn>] [/i<cmdfile>]  
[/l<language>] [/q] [/t]
```

Kommandozeilen-Parameter

- **/c:** nach einem Fehler während der Ausführung des Kommandos rvsbat fortsetzen. Im Standard beendet rvsbat nach einem Fehler die Ausführung.
- **/e:** vom Standard abweichende Umgebungsdatei RVSPATH/rvsenv.dat verwenden.
- **/icmdfile:** Kommandos werden nicht von stdin gelesen, sondern aus der Kommando-Eingabedatei cmdfile. Die Kommando-Eingabedatei enthält folgende Elemente:
 - Kommandos: darf mehrere Zeilen lang sein; am Ende jeder fortzusetzenden Zeile muss das + stehen.
 - Kommentarzeilen: beginnend mit *.
- **/language:** Nachrichtensprache verwenden, wie in language angegeben.
- **/q:** Kommando ohne Echo (Stumm-Modus) ausführen; Rückmeldungen über Erfolg oder Fehler der Operation werden weiterhin unterstützt.
- **/t:** Testmodus verwenden.

Kommandobehandlung

Kommandos können in die Kommando-Eingabedatei oder mit dem Aufruf von `rvsbat` geschrieben werden. In beiden Fällen ist die Syntax des Kommandos identisch.

Beispiel:

```
SEND /C DSN=c:\temp\readme.txt (SID=LOC  
DSNNEW=new.txt)
```

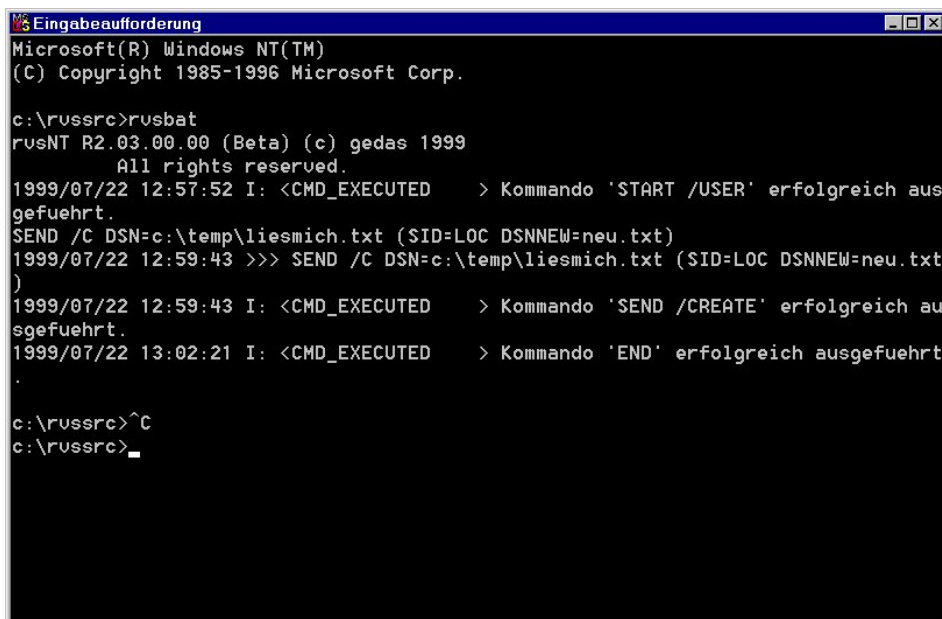
Eine genaue Beschreibung der Kommandos ist in **Kapitel 5: Beschreibung der Kommandos** enthalten.

Für jedes von der Kommandoeingabe gelesene Kommando wird eine Nachricht zu **stdout** geschrieben, die Informationen über den Erfolg oder Fehler bei der Bearbeitung des Kommandos enthält.

Wenn `rvsbat` ohne Eingabe einer Kommandodatei gestartet wird, kann es mit **<STRG> C** abgebrochen werden.

Beispiele:

Die folgende Abbildung zeigt Ihnen ein Beispiel für das Senden einer Datei mittels `rvsbat` unter **Windows NT**:



```
Eingabeaufforderung
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

c:\russrc>rvsbat
rvsNT R2.03.00.00 (Beta) (c) gedas 1999
All rights reserved.
1999/07/22 12:57:52 I: <CMD_EXECUTED > Kommando 'START /USER' erfolgreich ausgeführt.
SEND /C DSN=c:\temp\liesmich.txt (SID=LOC DSNNEW=neu.txt)
1999/07/22 12:59:43 >>> SEND /C DSN=c:\temp\liesmich.txt (SID=LOC DSNNEW=neu.txt)
)
1999/07/22 12:59:43 I: <CMD_EXECUTED > Kommando 'SEND /CREATE' erfolgreich ausgeführt.
1999/07/22 13:02:21 I: <CMD_EXECUTED > Kommando 'END' erfolgreich ausgeführt.
.

c:\russrc>^C
c:\russrc>_
```

5 Beschreibung der Kommandos

Dieser Abschnitt beschreibt die Syntax der Kommandos sowie den von der C-CAL-Schnittstelle (*rvscal*) und der Kommandozeilen-Schnittstelle (*rvsbat*) benutzten gültigen Kommandosatz.

Die unten stehenden Beispiele benutzen zwei Syntax- Erweiterungen, die nur für die Kommandozeilen-Schnittstelle verfügbar sind:

- Kommentarzeilen beginnen mit einem * in der Spalte **1**,
- Kommandos können in der nachfolgenden Zeile weitergeführt werden, indem ein + als letztes Zeichen der fortzusetzenden Zeile steht.

5.1 Syntax der Kommandos

Ein Utility-Kommando muss der untenstehenden Syntax folgen:

- Es besteht aus
 - einem Kommando-Verb,
 - einem optionalen Bestimmungswort wie /CREATE, /DELETE, usw. Das Bestimmungswort kann mit / oder – beginnen und darf auf einen Buchstaben abgekürzt sein.
 - Werte nicht-wiederholbarer Parameter, angegeben durch <Parametername>=<Parameterwert>. Vor und nach dem Gleichheitszeichen "=" darf kein Leerzeichen stehen.
 - Werte wiederholbarer Parameter (nur Kommando send); diese folgen den selben Syntaxregeln wie für nicht-wiederholbare Parameter. Ein Gruppe wiederholbarer Parameter wird in Klammern gesetzt. Es gibt beliebig viele Gruppen wiederholbarer Parameter.
- Kommando-Verb, Bestimmungsworte und Parameternamen können in Groß-, Kleinbuchstaben oder gemischt angegeben sein.
- Parameterwerte werden in Großbuchstaben konvertiert, wenn sie nicht durch einfache oder doppelte Anführungszeichen geschützt sind.
Wenn ein Parameterwert das schützende Zeichen selbst enthalten soll, muss das schützende Zeichen zweimal angegeben sein. Z.B. sind die folgenden Angaben gleichbedeutend.

```
PARM=' "test string" ' or parm="" "test string" ""
```

Nicht geschützte Parameterwerte können beliebige alphanumerische Zeichen enthalten. Sie können einen Satz spezieller Zeichen nicht enthalten, z.B. Leerzeichen, einfache oder doppelte Anführungszeichen und Klammern.

- Unterschiedliche Parameterangaben müssen durch mindestens ein Leerzeichen voneinander getrennt sein.

5.2 Das Kommando `START`

Funktion:

- Eine Sitzung mit den rvs[®] Utilities starten
- rvs[®] Datenbank öffnen,
- Prüfen, ob der aktuelle Benutzer die rvs[®] Utilities benutzen darf
- `start /USER` wird implizit durch die C-CAL-Schnittstelle aufgerufen, wenn es nicht explizit aufgerufen wird. Wenn die Kommandozeilen-Schnittstelle die Ausführung startet, wird `start /USER` immer implizit aufgerufen.

Bestimmungsworte:

`/USER` (Standard) startet die Sitzung für den rvs[®] Benutzer

Parameter:

RVSENV (optional) Name der rvs[®] Umgebungsdatei

Beispiele:

Umgebungsdatei in (rvs400 Bibliothek) **RVS_NEW/DAT(RVSENV)** benutzen:

```
START /USER RVSENV="RVS_NEW/DAT(RVSENV) "
```

5.3 Das Kommando END

Funktion:

- Beendet die Sitzung mit den rvs[®] Utilities
- Schließt die rvs[®] Datenbank
- END muss bei der Verwendung der C-CAL-Schnittstelle aufgerufen werden. Wenn die Kommandozeilen-Schnittstelle die Ausführung startet, wird END immer implizit aufgerufen.

Bestimmungsworte:

keine

Parameter:

keine

Beispiele:

END

5.4 Das Kommando SEND

Funktion:

- Erstellt
- ändert oder
- löscht

einen Sendeauftrag für die Übertragung einer lokalen Datei an einen anderen rvs[®] Knoten.

Bestimmungsworte:

`/CREATE` (Standard) erstellt einen Sendeauftrag
oder `/C`

`/DELETE` löscht einen Sendeauftrag
oder `/D`

`/HOLD` oder `/H` setzt den wartenden Sendeauftrag in den Status
"angehalten" (held)

`/RELEASE` oder `/R` Sendeauftrag freigeben, der zuvor in den Status
"angehalten" gesetzt wurde

SEND /CREATE Parameter:

DSN (erforderlich) Name der zu sendenden lokalen Datei, hier ist der ganze Pfad der Datei anzugeben.

CODEIN (optional) Code der lokalen Datei (**A**=ASCII, **E**=EBCDIC);
Standard: lokaler Code des Systems.

| | |
|-----------------|--|
| DISP | (optional) Disposition für die lokale Datei, nachdem der Sendeauftrag erfolgreich ausgeführt ist: K =halten (keep, Datei wird nicht gelöscht nach dem Versand, Standard), D =löschen (delete, Datei wird gelöscht nach dem Versand). |
| FORMAT | (optional) das für die Übertragung mit O-DETTE benutzte Format: T =Text (eine Folge von ASCII-Zeichen), U = unstrukturiert (binär), F =feste Satzlänge, V =variable Satzlänge; Standard: Satzformat der lokalen Datei (U für rvsNT, rvsX und rvs2, F für rvs400) |
| ACCOUNT | Berechnungsnummer oder Code. |
| INITTIME | (optional) frühest möglicher Zeitpunkt, zu dem der Sendeauftrag ausgeführt werden kann; möglich sind: H =anhalten (hold), N =jetzt (now, Standard), oder eine bestimmte Zeit im Format JJJJ/MM/TT HH:MM (Beispiel: 2004/09/04 10:43) . |
| SERIAL | (optional) bei Y (=Ja) wird der Sendeauftrag in der von Ihnen eingegebenen Reihenfolge gesendet (siehe auch LABEL). Die Datei wird erst gesendet, wenn der vorhergehende Sendeauftrag vollständig erledigt ist. |
| LABEL | (optional) Benutzerkennung (bis zu 20 Zeichen) für die Serialisierung eines vorangehenden Sendeauftrages (bei SERIAL=Y). Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (LABEL) haben. |

VFTYP

(optional) Textdateien können auch im Format **Fest(F)** oder **Variable(V)** versendet werden, ohne mit dem `rvs®`-Hilfswerkzeug `rvsut2fv` bearbeitet zu werden.

VFTYP=T (text) bedeutet, dass Ihre Textdatei ohne `rvsut2fv` versendet wird.

Textdatei bedeutet eine Folge von ASCII-Zeichen, wobei bei den einzelnen Satzlängen, der Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen) nicht dazugechnet wird. Neben dem Parameter **VFTYP** müssen auch die Parameter **MAXRECL** und **FORMAT** gesetzt werden. **Beispiel:** Wenn Sie eine Textdatei im Format **F** mit Satzlänge 80 versenden möchten, muss Ihre Textdatei in jeder Zeile 80 Zeichen lang sein (ohne CR/LF oder LF). Folgende Parameter sind dann wie folgt zu setzen: **VFTYP=T, FORMAT=F, MAXRECL=80.**

Wenn Sie aber Ihre Textdatei im Format **V** versenden möchten, bedeutet dies, dass Sie eine Textdatei (eine Folge von ASCII-Zeichen) mit unterschiedlichen Zeilenlängen vorbereitet haben (z.B. der längste Satz ist 120 Zeichen lang) und die notwendigen Parameter sind dann wie folgt zu belegen:

VFTYP=T, FORMAT=V, MAXRECL=120.

Als **MAXRECL** ist die maximale Länge eines Satzes bis zum Zeilenwechsel gedacht (ohne CR/LF oder LF). Siehe auch Beispiele am Ende des Abschnittes.

VFTYP=V (variabel) bedeutet, dass die Datei vor dem Versand mit `rvsut2fv` behandelt wurde. Dann ist nur das Format (**FORMAT**) der zu sendenden Datei anzugeben, ohne den Parameter **MAXRECL**. Siehe Beispiele am Ende des Abschnittes.

MAXRECL

(optional) maximale Satzlänge für Dateien im Format **T**, **F** oder **V**; dieser Parameter wird nur benötigt, wenn die Dateien nicht mit dem `rvs®`-Hilfswerkzeug `rvsut2fv` davor konvertiert wurden. Siehe auch Parameter **VFTYP**.

SIDORIG ID der virtuellen Station beim Senden; der Wert dieses Parameters wird beim Senden in die SFID (Start File ID) eingefügt.

Die folgende Gruppe der Sendeparameter sind innerhalb der runden Klammern () anzugeben:

SID (erforderlich) Stations-ID des Empfängers
UID (optional) Benutzer-ID des Empfängers; wenn kein Wert oder einer leerer String angegeben ist, ist dies des System auf dem fernen Station

COMPRESSION (optional) bei **Y** (=Ja) wird die Datei vor dem Senden komprimiert. Dieser Parameter muss beim Aufruf nach dem **SID** Parameter in Klammern angegeben werden; z.B.

```
SEND /C DSN=/home/test/test11.txt  
(SID=RTZ COMPRESSION=Y)
```

ENCRYPTION (optional) bei **Y** (=Ja) wird die Datei vor dem Senden verschlüsselt. Dieser Parameter muss beim Aufruf nach dem **SID** Parameter in Klammern angegeben werden; z.B.

```
SEND /C DSN=/home/test/test11.txt  
(SID=RTZ COMPRESSION=Y ENCRYP-  
TION=Y)
```

DSNNEW (optional) Für die Übertragung benutzter virtueller Dateiname (VDSN).

Bei Übertragungen an einen MVS-Hostrechner, muss dies ein gültiger MVS-Dateiname sein (den RACF-Regeln entsprechen).

Standard: VDSN wird aus dem lokalen Dateinamen gebildet. Die maximale Länge von VDSN ist nach ODETTE 26 Zeichen.

| | |
|------------------|--|
| CODEOUT | (optional) Gewünschter Code (A =ASCII, E =EBCDIC) der Datei beim Empfänger. Z.B. <code>send /c dsn=C:\test22.dat codein=a format=v(sid=rtt codeout=e dsnnew=FIX0GBE.TEXT)</code> |
| CODETABLE | (optional) Definiert die Codetabelle, die zur Codeumwandlung (siehe auch Abschnitt über Codeumwandlung im Benutzerhandbuch) verwendet werden soll. Hier ist der ganze Pfad der Codetabelle-Datei anzugeben. Beispiel: <code>send /c dsn=/home/send/test22.dat codein=a format=v (sid=rtt codeout=e codetable=/home/tables /rtcusrdat dsnnew=FIX0GBE.TEXT)</code> |
| TSTAMP | (optional) Gibt an, ob die Datei bei dem Empfänger einen Zeitstempel erhalten soll. Zur Zeit kann diese Anforderung an einen oder von einem MVS-Hostrechner nicht übergeben werden. |

SEND /DELETE, /HOLD, /RELEASE Parameter:

| | |
|--------------|---|
| DSN | (optional) Name der zu sendenden lokalen Datei |
| SID | (optional) Stations-ID des Empfängers |
| UID | (optional) Benutzer-ID des Empfängers; im Standard ein leerer String, d.h. das ferne System |
| CMDID | (optional) Eindeutige Kommandonummer des Sendeauftrages. |

Der zu bearbeitende Sendeauftrag kann entweder durch **DSN**, **SID**, und **UID** oder durch die eindeutige Kommandonummer **CMDID** des Sendeauftrags identifiziert werden.

Wenn kein Parameter angegeben wurde oder mehr als ein Sendeauftrag mit den angegebenen Parametern übereinstimmt, ändert `rvscal` keinen Sendeauftrag und gibt einen Fehler zurück.

Die folgenden Beispiele zeigen, wie Kommandos in einer Datei angegeben werden können, die als Eingabe-Datei für `rvsbat` benutzt wird. Der Gebrauch von Fortsetzungszeilen (die vorhergehende Zeile endet mit `+`) und Kommentaren (`*` in Spalte 1) wird ebenso gezeigt.

Beispiele:

```
SEND /C DSN=C:/RVS/LPDBI.C +  
SERIAL=n LABEL=l1 inittime=NOW +  
CODEIN=A FORMAT=T DISP=d +  
(SID=st1 UID=user1 CODEOUT=e TSTAMP=n  
DSNNEW=dsnnew1)
```

```
SEND /C DSN=/HOME/LPDBI.C +  
SERIAL=y LABEL=l1 inittime=HOLD +  
FORMAT=U DISP=k (SID=st1)
```

```
SEND /C DSN=rpu.c +  
SERIAL=y LABEL=l1 inittime='1991/07/01 10:35' +  
FORMAT=U (SID=st1)
```

```
*----- serialize on data set (without specifying  
* the full data set name)
```

```
SEND /C DSN=lpdbi.c SERIAL=y FORMAT=T (SID=st1)
```

```
*----- serialize again on data set (and default  
* for FORMAT)
```

```
*
```

```
SEND /C DSN=lpdbi.c SERIAL=y (SID=st1)
```

```
*----- delete SEND request (unique)  
SEND /D DSN=/home/rpu.c (SID=st1)
```

```
* --- Textdatei zum Host senden ohne rvsut2fv  
send /c dsn=C:\test.text codein=a format=f  
vftyp=t maxrecl=80(sid=rtt codeout=e  
dsnnew=FIX0GBE.TEXT)
```

```
* --- Textdatei, die mit rvsut2fv konvertiert  
* --- wurde zum Host senden  
send /c dsn=C:\test22.text codein=a format=v  
(sid=rtt codeout=e dsnnew=FIX0GBE.TEXT)
```


5.5 Das Kommando RESENTR

Funktion:

Erstellen, Überarbeiten und Löschen von residenten Empfangseinträgen.

Bestimmungsworte:

`/CREATE` (Standard) erstellt einen residenten Empfangseintrag
oder `/C`

`/UPDATE` überarbeitet einen residenten Empfangseintrag
oder `/U`

`/DELETE` löscht einen residenten Empfangseintrag
oder `/D`

Parameter:

DSN (erforderlich) Virtueller Name der eintreffenden Datei (VDSN, darf max. 26 Zeichen lang sein).

SID (erforderlich) Stations-ID des Senders.

ACCOUNT (optional) Berechnungsnummer oder Code

COMMENT (optional) Kommentar zur Beschreibung des residenten Empfangseintrages (bis zu 50 Zeichen lang);

Standard: leerer String

DSNNEW (optional) der neue Dateiname auf dem lokalen System. Hier ist der ganze Pfad der neuen Datei anzugeben.

Standard: Der lokale Dateiname wird aus dem virtuellen Dateinamen gebildet.

JOB

Name einer Kommandozeilen-Datei die nach der Zustellung der Datei gestartet wird¹ (.bat unter Windows-Systemen; .sh für UNIX-Systeme).

Diese Kommandozeilen-Datei darf Ersetzungszeichen enthalten. rvs[®] ersetzt sie, bevor es den Job zur Ausführung an das Betriebssystem übergibt.

- **?DSN?**: Name der lokalen Datei, in der die empfangene Information gespeichert wurde.
- **?VDSN?**: Virtueller Dateiname, unter welchem die Datei übertragen wurde.
- **?DTAVAIL?**: Datum, zu dem die Datei zum Senden verfügbar war.
- **?FORMAT?**: Satzformat der empfangenen Datei.
 - **F** feste Satzlänge
 - **V** variable Satzlänge
 - **T** Textdatei (ASCII-Zeichenfolge)
 - **U** unstrukturierte Datei (binär)
- **?MAXRECL?**: Die Bedeutung dieses Feldes hängt vom Satzformat der empfangenen Datei ab.
 - **F** Format: Länge jedes Satzes.
 - **V** Format: Maximale Länge, den ein Satz haben kann.
 - **T** und **U** Format: Immer **0** (null)
- **?BYTES?**: Zahl der übertragenen Bytes
- **?RECORDS?**: Zahl der übertragenen Sätze für Dateien im **F** und **V** Format; immer **0** (null) für Dateien im **T** und **U** Format
- **?DTRCV?**: Datum, an dem die Datei dem lokalen Benutzer zugestellt wurde
- **?LUID?**: ID des (lokalen) Empfängers
- **?UID?**: ID des Senders
- **?SID?**: Stations-ID des Empfängers
- **?SIDDEST?** StationsID der virtuellen Empfangsstation

¹ Die Art des geforderten Jobs hängt vom Betriebssystem ab. Z.B. wird unter OS/2 eine CMD-Datei als Hintergrundprozeß ausgeführt, während unter OS/400 rvs[®] ein Job mit Hilfe von **SBMDBJOB** startet.

- **?CNQS?** Kommandonummer der Quittungs-Sendung (EERP) für die empfangene Datei
- **?DSNTEMP?**: Name der temporären Datei (Diese Datei kann am Ende des Sendeauftrages mit dem Befehl
 - `DELETE ?DSNTEMP? (NT, XP, 2000)`
 - `rm ?DSNTEMP? (UNIX)`
 - `DLTF ?DSNTEMP? (OS/400)`gelöscht werden.)

- DISP** (optional) Bestimmt die Behandlung der Datei, wenn die Bearbeitung abgeschlossen ist.
- **K** hält die Datei
 - **D** löscht die Datei
- Standard: **K**
- REPLACE** (optional) Wenn **DISP=K** (halten) eingestellt ist, legt **REPLACE** die Aktionen fest, die rvs[®] durchführt, nach dem Empfang einer Datei, die im Namen mit einer lokalen Datei übereinstimmt.
- R** Vorhandene Datei ersetzen
- N** Neuen eindeutigen Dateinamen verwenden
- I** Eintreffen der Datei ignorieren
- Standard: **N**
- TSTAMP** (optional) Kann **Y=Ja** oder **N=Nein** sein; teilt rvs[®] mit, ob der Dateiname zur eindeutigen Kennzeichnung mit Zeitstempel versehen werden soll, wenn die Datei eingetroffen ist..
- Standard: **N**
- CODETRANS** (optional) Gibt an, ob die empfangene Datei mit rvs-internen Codeumwandlungstabellen in ASCII oder EBCDIC konvertiert werden soll, oder ob zu Codeumwandlung eine eigene Codetabelle benutzt werden soll (siehe auch Kapitel über Codeumwandlung im rvs[®] Benutzerhandbuch).
- A** für die Codeumwandlung nach ASCII
- E** für die Codeumwandlung nach EBCDIC
- T** für die Codeumwandlung mit Codetabelle
- z.B. `resentr /c dsn="<empfangene ASCII Datei>" codetrans=e sid="<Sender>"`

CODETABLE (optional) Definiert die eigene Codeumwandlungstabelle (siehe auch Abschnitt über Codeumwandlung im Benutzerhandbuch), die verwendet werden soll.

```
z.B. resentr /c dsn="<empfangene EBC-  
DIC Datei>" codetrans=t codetable  
="<User Code Tabelle, z.B.  
/home/rvs/arcdire/rtcusrdat>"  
sid="<Sender>"
```

VFTYP (optional) Sie können hier bestimmen, ob die empfangene Datei als Textdatei mit Zeilenwechsel (CR/LF bei MS Windows, LF bei Unix-Systemen) nach jedem Satz abgelegt werden soll. Diese Angabe gilt nur für Dateien, die im Format **Fixed** oder **Variable** empfangen werden.

VFTYP=T (text) bedeutet, dass die empfangene Datei als Textdatei mit Zeilenwechsel gespeichert werden soll.

VFTYP=V (variabel) bedeutet, dass die empfangene Datei im Format **Fixed** oder **Variable** ohne Umwandlung in Textdateien im rvs[®] internen Format gespeichert wird.

VFTYP=S (Sinix) bedeutet, dass die empfangene Datei im Sinix-Format gespeichert wird.

Beispiele:

```
*
*----- use all parameters
*
RESETR /C DSN=incoming1  SID=st2 +
  DSNNEW=/home/local.dsn REPLACE=n DISP=k +
  JOB=/home/rvs/bin/rcv.sh COMMENT='This is a test
RE'
*
*----- use defaults
*
RESETR /C DSN=incoming2  SID=st2 +
  REPLACE=i      DISP=d
*
*----- no UID, DISP
*
RESETR /C DSN=incoming3  SID=st2 +
  REPLACE=r
*
*----- delete RESETR
*
RESETR /D DSN=incoming3  SID=st2
*
*----- update RESETR
*
RESETR /U DSN=incoming2  SID=st2 +
  REPLACE=n      JOB=/home/rvs/bin/rcv.sh
```

5.6 Kommando SENDJOB

Funktion:

Erstellen, Überarbeiten und Löschen von Einträgen für Jobstart nach Sendeversuch.

Bestimmungsworte:

- /CREATE (Standard) erstellt einen Eintrag für Jobstart nach Sendeversuch
- /UPDATE überarbeitet einen residenter Empfangseintrag
- /DELETE löscht einen residenten Empfangseintrag

Parameter:

- VDSN** (erforderlich) Virtueller Name der gesendeten Datei.
- SID** (erforderlich) Stations-ID des Empfängers
- ATTEMPTS** (optional) Zahl der Sendeversuche; legt fest, bei welcher Bedingung der Job starten soll
- Wert **0**: Der Job startet bei erfolgreicher Übertragung
- Wert **>0**: Der Job startet, wenn die angegebene Zahl fehlgeschlagener Sendeversuche erreicht ist
- Standard: **0**
- JOB** (optional) Name des Jobs, der nach erfolgreicher Übertragung oder erfolglosen Sendeversuch gestartet werden soll. Diese Kommandozeilen-Datei (der Job) darf Ersetzungszeichen enthalten. rvs[®] ersetzt sie, bevor es den Job zur Ausführung an das Betriebssystem übergibt:
- **?DSN?**: Name der lokalen Datei, die gesendet wurde. Bei **EERP** gibt es keinen lokalen Dateinamen. Der Wert von **?DSN?** hat die Form **QS (SIDORIG - SIDDEST)** mit der Bedeutung:
 - **SIDORIG** Stations-ID des Senders
 - **SIDDEST** Stations-ID des Empfän-

gers

- **?VDSN?**: virtueller Dateiname unter dem die Datei übertragen wurde.
- **?DTAVAIL?**: Datum, an dem die Datei für das Senden verfügbar war.
- **?FORMAT?**: Satzformat der gesendeten Datei
 - **F** fest
 - **V** variabel
 - **T** Text
 - **U** unstrukturiert
- **?BYTES?**: Zahl der übertragenen Bytes
- **?RECORDS?**: Zahl der im Format **F** oder **V** übertragenen Sätze
- **?DTRCV?**: Datum, an dem die Datei dem lokalen Benutzer zugestellt wurde.
- **?LABEL?**: Zeichenfolge, wenn das Sendekommando ein **LABEL** Parameter enthielt.. Kann zur Identifikation des Sendekommandos benutzt werden.
- **?SECN?**: Kommandonummer des Sendekommandos (**CN** des **SE**). Kann zur Identifikation des Sendekommandos benutzt werden.
- **?SKCN?**: Nummer des Sendekommandos
- **?UID?**: Benutzer-ID des Senders; bei **EERP** ist der Wert immer "**!-QS-!**".
- **?SID?**: Stations-ID des Empfängers
- **?SIDORIG?** Stations-ID der virtuellen Senderstation.
- **?SENDATT?** Anzahl der Sendeversuche, nach denen der Job gestartet werden soll.
- **?DSNTEMP?**: Name der temporären Datei (Diese Datei kann am Ende des Sendeauftrages mit dem Befehl
 - DELETE ?DSNTEMP? (**NT**)
 - rm ?DSNTEMP? (**UNIX**)
 - DLTf ?DSNTEMP? (**OS/400**) gelöscht werden.)

COMMENT (optional) Ein Kurztext zur Beschreibung des Zwecks des Jobstarteintrages (bis zu 50 Zeichen lang);Standard: leerer String

Beispiele:

```
*
*----- use all parameters
*
SENDJOB /C VDSN=sending1  SID=st2 ATTEMPTS=1 +
  JOB=/home/rvs/bin/send-fail1.sh COMMENT='This is
a test JS'
*
*----- use defaults
*
SENDJOB /C VDSN=sending2  SID=st2
JOB=/home/rvs/bin/snd.sh
*
*----- Job should start after data have been
transmitted successfully
*
SENDJOB /C VDSN=sending3  SID=st2
JOB=/home/rvs/bin/snd.sh
*
*----- delete SENDJOB
*
SENDJOB /D VDSN=sending3  SID=st2
*
*----- update SENDJOB
*
SENDJOB /U DSN=/home/sending2  SID=st2
JOB=/home/rvs/bin/sendok.sh
```


5.7 Das Kommando `USER`

Funktion:

- `rvs`[®] Benutzertabelle ändern
- Benutzer einstellen
- Dialogsprache für Benutzer festlegen
- Privilegien für die Benutzer festlegen

Bestimmungsworte:

`/CREATE` erstellt einen Eintrag in die `rvs`[®] Benutzertabelle
`/DELETE` löscht einen Eintrag in die `rvs`[®] Benutzertabelle
`/UPDATE` überarbeitet einen Eintrag in die `rvs`[®] Benutzertabelle

Wenn kein Bestimmungswort angegeben ist, wird der Eintrag erstellt oder überarbeitet, je nachdem, ob der Eintrag schon besteht oder nicht.

Parameter:

UID (optional) Benutzer-ID, die den zu ändernden Eintrag identifiziert; Sie können nur als privilegierter Benutzer etwas anderes als die aktuelle Benutzer-ID angeben.
Standard: current user

NAME (optional) neuer Benutzername

LANGUAGE (optional) Neue Sprache angeben (z.B. **E**=Englisch, **D**=Deutsch)

PRIV (optional) neues Benutzerprivileg; (**U**=Benutzer, **O**=Operator, **A**=Administrator)

Beispiele:

```
*----- create new user (default=own UID)
*
USER /C
*
*----- create new user
*
```

rvsbat Schnittstellenbeschreibung

```
USER /C UID=newuser LANGUAGE=d NAME=x PRIV=O
USER /C UID=extrauser
*
*----- update existing user
*
USER /U UID=newuser LANGUAGE=e NAME=y PRIV=U
*
*----- delete user
*
USER /D UID=newuser
```

5.8 Das Kommando `ACTIVATE`

Funktion:

- Aktiviert eine Partnerstation
- `rvs`[®] startet einen Sendeprozess, der die Verbindung zur Partnerstation herstellt. Alle Dateien in Wartestellung werden übertragen. Danach wird die Verbindung wieder getrennt.

Parameter:

SID (erforderlich) Stations-ID des Partners

Beispiele:

```
*
*----- activate station ABC
*        (connect and send or receive queued data
sets)
*
ACTIVATE SID=ABC
```

5.9 Das Kommando MODST

Funktion:

Stationstabelle ändern.

Achtung: Dieses Kommando überschreibt alte Datenbank-einträge; andere vorhandene Einträge werden nicht gelöscht () nur **DELST** löscht einen Eintrag).

Parameter:

DSN (erforderlich) Name der Datei, die die Stationstabelle enthält. Der Dateiname kann eine einfache Eingabedatei oder ein Verzeichnis mit mehreren darin enthaltenen Eingabedateien sein.

Beispiele:

```
*
*----- modify station table:
*      (open the file (here: UNIX file name),
*      read contents and put it into database):
*
MODST DSN="/home/rvs/init/new_rdstat.dat"
```

5.10 Das Kommando DELST

Funktion:

Stationstabelle löschen.

Parameter:

SID Stations-ID der zu löschenden Station

Beispiele:

```
*
*----- delete entry ABC in database:
*
DELST SID=ABC
```

5.11 Das Kommando LISTPARM

Funktion:

- Gibt den Wert eines rvs[®] Parameters aus.
- Im Kommandozeilen-Modus erscheint die Meldung "I: Wert".
- In C-Programmen wird der Wert als String im `rvscal()`-Ausgabeparameter **S_MSG** übergeben.

Parameter:

parameter Name des rvs[®] Parameters.

Beispiele:

```
*
*----- show the value of the parameter SLEEP
*
LISTPARM SLEEP
```

5.12 Das Kommando SETPARAM

Funktion:

Setzt den Wert eines rvs[®] Parameters.

Parameter:

parameter Name des rvs[®] Parameters.

Beispiele:

```
*
*----- set the value of the parameter SLEEP
*
SETPARM SLEEP=2
```