Entire Connection

Befehle

Version 9.2.1

Oktober 2022

ADABAS & NATURAL

Dieses Dokument gilt für Entire Connection ab Version 9.2.1.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Release Notes oder Neuausgaben bekanntgegeben werden.

Dokument-ID: PCC-COMMANDS-921-20221013DE

Inhaltsverzeichnis

Vorwort

Entire Connection stellt folgende Befehle zur Verfügung:

BEEP	EXECUTE	PERFORM	RSPMONITOR
CANCEL	EXIT	POPDIR	SCHEDTOP/SCHEDULE
CD	GOTO	PUSHDIR	SET
CHDRIVE	IF/IFNOT	QA	SHIFT
CHMOD	INCR	QUIT	SLEEP
CLOSE	INPUT	RD	TOGGLE
CONNECT	LEARN	READ	TYPE
DECR	LOG	REC_BUFF	WAIT
DISCONNECT	MD	REC_SCR	WAITFOR
DOS	MONITOR	REC_XFER	WAITM
DOSDIR	MSG	RECALL	WAITUNTIL
ELAPSETIME	OPEN-I	RENAME	WRITE
EMULATE	OPEN-O	RESET	
ERASE	OS	RETURN	
EXECTASK	PAUSE	REVEAL	

Konventionen

Nachstehend finden Sie die folgenden Informationen:

- Befehlsformat
- Syntaxkonventionen
- Befehlseingabe

Siehe auch: *Befehle an den PC senden* im Abschnitt *Terminal-Emulation*.

Befehlsformat

Ein Befehl hat meist einen oder mehrere Operanden. Mehrere Operanden werden durch Leerzeichen voneinander getrennt.

```
befehl operand1 operand2 ... operandn
```

Es gibt erforderliche und optionale Operanden. Einige Operanden haben optionale Parameter. Operanden bestehen aus Zeichenketten oder Integer-Werten.

Syntaxkonventionen

Bei den Befehlsbeschreibungen gelten die folgenden Syntaxkonventionen:

Konvention	Beschreibung
GROSSBUCHSTABEN	Begriffe in Großbuchstaben stellen Befehle, Variablen usw. dar. Sie müssen diese Begriffe genauso eingeben wie sie in der Syntax angegeben sind.
kursive kleinbuchstaben	Begriffe in kursiven Kleinbuchstaben sind Platzhalter, für die Sie die entsprechenden Informationen eingeben müssen. Bei eingebetteten Leerzeichen oder wenn zwischen Groß- und Kleinschreibung unterschieden werden soll, muss der Begriff in doppelte oder einfache Anführungszeichen gesetzt werden (" " oder ' ').
[ ]	Eckige Klammern bedeuten, dass die Eingabe des eingeklammerten Begriffs optional ist.
{ }	Geschweifte Klammern bedeuten, dass Sie einen der eingeklammerten Begriffe eingeben müssen.
|	Vertikale Striche werden zur Trennung von alternativen Parameterwerten benutzt.
...	Der vor den Auslassungspunkten stehende Begriff kann wiederholt werden.

Befehlseingabe

Jede Befehlsbeschreibung enthält Informationen, an welcher Stelle Sie den Befehl benutzen können. Es gibt folgende Möglichkeiten:

Benutzung	Beschreibung
Prozedurdatei	Der Befehl kann in einer Prozedurdatei benutzt werden.
Befehlszeile	Der Befehl kann in der Befehlszeile eingegeben werden.
Taste	Der Befehl kann auf eine Taste oder Tastenkombination gelegt werden.
API	Der Befehl kann in einem Programm benutzt werden, das die Programmierschnittstelle (API) benutzt.

1 Über diese Dokumentation

Dokumentationskonventionen

Konvention	Beschreibung
Fettschrift	>Kennzeichnet Elemente auf einem Bildschirm.
`Nichtproportionale Schrift`	Kennzeichnet Namen und Orte von Diensten im Format *`Ordner.Unterordner.Dienst`*, Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
`Kursivschrift`	Kennzeichnet: Variablen, für die Sie situations- oder umgebungsspezifische Werte angeben müssen. Neue Begriffe, wenn sie erstmals im Text auftreten. Verweise auf andere Dokumentationsquellen.
`Nichtproportionale Schrift`	Kennzeichnet: Text, den Sie eingeben müssen. Meldungen, die vom System angezeigt werden. Programmcode.
{ }	Zeigt eine Reihe von Auswahlmöglichkeiten an, von denen Sie eine auswählen müssen. Geben Sie nur die innerhalb der geschweiften Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole { } ein.
|	Trennt zwei sich gegenseitig ausschließende Auswahlmöglichkeiten in einer Syntaxzeile voneinander ab. Geben Sie eine der Auswahlmöglichkeiten ein. Geben Sie nicht das Symbol | ein.
[ ]	Zeigt eine oder mehrere Optionen an. Geben Sie nur die innerhalb der eckigen Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole [ ] ein.
...	Zeigt an, dass Sie mehrere Auswahlmöglichkeiten desselben Typs eingeben können. Geben Sie nur die Informationen ein. Geben Sie nicht die drei Auslassungspunkte (...) ein.

Online-Informationen und Support

Produktdokumentation

Sie finden die Produktdokumentation auf unserer Dokumentationswebsite unter **https://documentation.softwareag.com**.

Zusätzlich können Sie auch über **https://www.softwareag.cloud** auf die Dokumentation für die Cloud-Produkte zugreifen. Navigieren Sie zum gewünschten Produkt und gehen Sie dann, je nach Produkt, zu „Developer Center“, „User Center“ oder „Documentation“.

Produktschulungen

Sie finden hilfreiches Produktschulungsmaterial auf unserem Lernportal unter **https://knowledge.softwareag.com**.

Tech Community

Auf der Website unserer Tech Community unter **https://techcommunity.softwareag.com** können Sie mit Experten der Software AG zusammenarbeiten. Von hier aus können Sie zum Beispiel:

- Unsere umfangreiche Wissensdatenbank durchsuchen.
- In unseren Diskussionsforen Fragen stellen und Antworten finden.
- Die neuesten Nachrichten und Ankündigungen der Software AG lesen.
- Unsere Communities erkunden.
- Unsere öffentlichen Repositories auf GitHub and Docker unter **https://github.com/softwareag** und **https://hub.docker.com/publishers/softwareag>** besuchen und weitere Ressourcen der Software AG entdecken.

Produktsupport

Support für die Produkte der Software AG steht lizenzierten Kunden über unser Empower-Portal unter **https://empower.softwareag.com>** zur Verfügung. Für viele Dienstleistungen auf diesem Portal benötigen Sie ein Konto. Wenn Sie noch keines haben, dann können Sie es unter **https://empower.softwareag.com/register>** beantragen. Sobald Sie ein Konto haben, können Sie zum Beispiel:

- Produkte, Aktualisierungen und Programmkorrekturen herunterladen.
- Das Knowledge Center nach technischen Informationen und Tipps durchsuchen.
- Frühwarnungen und kritische Alarme abonnieren.
- Supportfälle öffnen und aktualisieren.
- Anfragen für neue Produktmerkmale einreichen.

Datenschutz

Die Produkte der Software AG stellen Funktionen zur Verarbeitung von personenbezogenen Daten gemäß der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union zur Verfügung. Gegebenenfalls sind in der betreffenden Systemverwaltungsdokumentation entsprechende Schritte dokumentiert.

2 BEEP

Beschreibung

Lässt den Signalton des PC ertönen.

Die Systemvariable `BEEP` muss eingeschaltet sein.

Syntax

```
BEEP
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

3 CANCEL

Beschreibung

Bricht die Verarbeitung einer Prozedurdatei ab.

Wenn der Befehl in einer verschachtelten Prozedurdatei verwendet wird, werden alle Prozedurdateien abgebrochen. Der Befehl hat keine Auswirkung auf geplante Tasks und Prozedurdateien.

Syntax

```
CANCEL
```

Prozedurdateibeispiel

Parms.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

EXIT

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

4 CD

Beschreibung

Wechselt das aktuelle Verzeichnis auf dem aktuellen Laufwerk.

Syntax

```
CD {[laufwerk:]\verzeichnis\...}
```

Beispiele

- Zum Verzeichnis *SAG* auf dem aktuellen Laufwerk wechseln:

  ```
  CD \SAG
  ```

- Zu einem Verzeichnis wechseln, das durch den Inhalt der Variablen #FILEDRIVE und #FILEPATH definiert ist:

  ```
  CD #FILEDRIVE ':' #FILEPATH
  ```

- Zum Verzeichnis *SAG\PCC* auf dem aktuellen Laufwerk wechseln:

  ```
  CD \SAG\PCC
  ```

Zurückgegebene Variablen

#RC (SUCCESS, wenn CD erfolgreich war. FAILURE, wenn CD nicht erfolgreich war.)

Verwandte Befehle

CHDRIVE, MD, RD

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Nein
API:	Nein

5 CHDRIVE

Beschreibung

Wechselt das aktuelle Laufwerk.

Syntax

```
CHDRIVE laufwerk
```

Beispiele

- Zu Laufwerk A wechseln:

  ```
  CHDRIVE A
  ```

- Zum Laufwerk wechseln, das durch den Inhalt der Variable #FILEDRIVE definiert ist:

  ```
  CHDRIVE #FILEDRIVE
  ```

Zurückgegebene Variablen

#RC (SUCCESS, wenn CHDRIVE erfolgreich war. FAILURE, wenn CHDRIVE nicht erfolgreich war.)

Verwandte Befehle

CD, MD, RD

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Nein
API:	Nein

6 CHMOD

Beschreibung

Ändert die Dateiattribute.

Syntax

```
CHMOD pfad [optionen]
```

dabei ist *pfad*:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

optionen umfasst die folgenden Attribute:

A	Archiv
H	versteckt
R	schreibgeschützt
S	System

Wenn Sie keine Optionen angeben, werden alle Dateiattribute ausgeschaltet.

Beispiele

- Alle Attribute ausschalten:

```
CHMOD Test.ncp
```

- Das Attribut „Archiv“ setzen:

```
CHMOD Test.ncp A
```

- Die Attribute „Versteckt“ und „Schreibgeschützt“ für die in der Variable #FILESPEC definierte Datei setzen:

```
CHMOD #FILESPEC HR
```

Zurückgegebene Variablen

#RC (SUCCESS, wenn CHMOD erfolgreich war. FAILURE, wenn CHMOD nicht erfolgreich war.)

Verwandte Befehle

DOSDIR

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

7 CLOSE

Beschreibung

Schließt eine Datei.

Wenn der Befehl `CLOSE` weggelassen wird, werden beim Beenden der Prozedurdatei (und aller verschachteteln Prozedurdateien) automatisch alle zum Lesen geöffneten Dateien geschlossen. Für alle zum Schreiben geöffneten Dateien wird eine End-of-File-Markierung gesetzt.

Syntax

```
CLOSE dateinummer
```

wobei *dateinummer* eine Zahl zwischen 1 und 4 sein kann.

Beispiel

- Datei 1 schließen:

```
CLOSE 1
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

#RC (`SUCCESS`, wenn `CLOSE` erfolgreich war. `FAILURE`, wenn `CLOSE` nicht erfolgreich war.)

Verwandte Befehle

`OPEN-I`, `OPEN-O`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

8 CONNECT

Beschreibung

Öffnet eine Host-Session.

Wenn die Session geöffnet wird, enthält die Variable `*COMMTYPE` den Session-Namen.

Der Session-Name wird in der Titelleiste des Terminal-Anwendungsfensters angezeigt.

Syntax

```
CONNECT sessionname
```

wobei *sessionname* der Name einer Session ist, die in den Session-Eigenschaften definiert wurde.

Beispiel

- Host-Session mit dem Namen MEINIBM öffnen:

```
CONNECT MEINIBM
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

DISCONNECT

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Nein
API:	Ja

9 DECR

Beschreibung

Zieht 1 von den globalen und lokalen Zählervariablen oder Bildschirmpositionsvariablen ab.

Syntax

```
DECR {zähler|bildschirmpositionsvariable}...
```

zähler steht für eine der folgenden Variablen:

#CNT0 bis #CNT9 (lokal)
+CNT0 bis +CNT9 (global)

Der gültige Zahlenbereich für die Zählervariablen liegt zwischen 0 und 32767.

bildschirmpositionsvariable steht für eine der folgenden Variablen:

#ROW, #COL, #LENGTH (lokal)
+ROW, +COL, +LENGTH (global)

Beispiele

- 1 vom lokalen Zähler #CNT1 abziehen:

```
DECR #CNT1
```

- 1 vom globalen Zähler +CNT1 abziehen:

```
DECR +CNT1
```

- 1 von den lokalen Zählern #CNT1 und #CNT2 abziehen:

```
DECR #CNT1 #CNT2
```

- 1 vom lokalen Zähler #CNT1 und vom globalen Zähler +CNT1 abziehen:

```
DECR #CNT1 +CNT1
```

- 2 vom lokalen Zähler #CNT1 abziehen:

```
DECR #CNT1 #CNT1
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

INCR, RESET, SET

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

10 DISCONNECT

Beschreibung

Schließt eine Host-Session.

Wenn dieser Befehl aus der Befehlszeile oder mit einer Prozedurdatei ausgeführt wird, wird die aktuelle Session beendet.

Wenn dieser Befehl über die Programmierschnittstelle ausgeführt wird, wird die von der Programmierschnittstelle angesprochene Session beendet.

Wichtig: Dieser Befehl darf nicht in einer Prozedurdatei verwendet werden, die eine automatische An- und Abmeldung beim Host ausführt.

Syntax

```
DISCONNECT
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

CONNECT

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Nein
API:	Ja

11 DOS

Beschreibung

Führt einen DOS-Befehl aus.

Der Befehl öffnet bei Windows ein DOS-Fenster (MS-DOS-Eingabeaufforderung), in dem die mit diesem Befehl angegebenen Befehle ausgeführt werden.

Die Befehle DOS und OS sind identisch.

Syntax

```
DOS dosbefehl
```

Beispiele

- Alle Dateien mit der Namenserweiterung *exe* auflisten:

```
DOS DIR *.exe
```

- Die Stapeldatei *Test.bat* ausführen:

```
DOS Test
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

OS

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Nein

12 DOSDIR

Beschreibung

Zeigt DOS-Verzeichnis-Informationen an.

Der Verzeichnisinhalt wird in den unten aufgeführten Variablen zurückgegeben.

Der Befehl `DOSDIR` zeigt den Verzeichnisinhalt genauso an, wie der DOS-Befehl `DIR`.

Syntax

```
DOSDIR pfad [optionen]
```

dabei ist *pfad*:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Für *dateiname* und *erweiterung* können Sie die Platzhalterzeichen Stern (*) und Fragezeichen (?) verwenden.

Wenn Sie *pfad* nicht angeben, nimmt Entire Connection an, dass der Befehl `DOSDIR` mit Platzhalterzeichen eingegeben wurde, und sucht nach der nächsten Datei, die mit dem zuletzt benutzten Muster übereinstimmt.

Wenn sich die Datei nicht im aktuellen DOS-Pfad befindet, müssen Sie den kompletten Pfad zu dieser Datei angeben.

optionen umfasst Folgendes:

A	nur Dateien mit dem Attribut „Archiv“
D	nur Verzeichnisse
H	nur Dateien mit dem Attribut „Versteckt“
R	nur Dateien mit dem Attribut „Schreibgeschützt“
S	nur Dateien mit dem Attribut „System“

Wenn Sie keine *optionen* angeben, werden nur die normalen Dateien angezeigt. Verzeichnisse, schreibgeschützte, versteckte und Systemdateien werden in diesem Fall nicht angezeigt.

Beispiele

- Alle *ncp*-Dateien im aktuellen Verzeichnis auflisten:

```
DOSDIR *.ncp  /* erste "ncp"-Datei auflisten
DOSDIR        /* nachfolgende "ncp"-Dateien auflisten
```

- Alle Dateien mit dem Attribut „Archiv“ auflisten:

```
DOSDIR *.* A  /* erste Datei mit dem Attribut "Archiv" auflisten
DOSDIR        /* nachfolgende Dateien mit dem Attribut "Archiv" auflisten
```

- Alle Dateien auflisten, die dem Inhalt der Variable #FILESPEC entsprechen:

```
DOSDIR #FILESPEC /* erste Datei auflisten, die #FILESPEC entspricht
DOSDIR           /* nachfolgende Dateien auflisten, die #FILESPEC entsprechen
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

#RC (SUCCESS, FILE NOT FOUND, NO MORE FILES oder INVALID PATH)
#FILEDRIVE
#FILEPATH
#FILENAME
#FILEEXT
#FILESIZE
#FILEDATE
#FILEYEAR
#FILEMONTH
#FILEDAY
#FILETIME
#FILEHOUR
#FILEMINUTE

#FILESECOND
#FILEMODE
#FILETYPE
#FILESPEC
#FILEINFO

Siehe auch: *Lokale Variablen*

Verwandte Befehle

CD, CHDRIVE, CHMOD, PUSHDIR

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

13 ELAPSETIME

Beschreibung

Berechnet die Differenz in Sekunden zwischen zwei Datums- und Zeitangaben.

Syntax

```
ELAPSETIME datum zeit datum zeit
```

datum hat das Format JJJJ/MM/TT. Ab dem Jahr 2000 müssen Sie das Jahr im Format JJJJ angeben. Bis 1999 können Sie das Jahr im Format JJ oder JJJJ angeben.

zeit hat das Format HH:MM:SS.

Beispiel

- Wenn die dynamischen Variablen *DATE und *TIME die Werte 1998/03/21 und 12:00:00 enthalten, wird in der lokalen Variable #ELAPSETIME der Wert 900 zurückgegeben:

```
ELAPSETIME *DATE *TIME 1998/03/21 12:15:00
```

Prozedurdateibeispiel

Vars.ncp

Zurückgegebene Variablen

#ELAPSETIME - positive Zahl, die die Differenz in Sekunden angibt.

Verwandte Befehle

PAUSE, SLEEP, WAIT, WAITFOR, WAITM, WAITUNTIL

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

14 EMULATE

Beschreibung

Wechselt in den Terminal-Emulationsmodus.

Dieser Befehl hat zwei Funktionen:

- Wenn der Befehl `EMULATE` in einer Prozedurdatei benutzt wird, unterbricht er die Verarbeitung und wechselt zum Terminal-Emulationsmodus. Dies erlaubt dann manuelle Eingaben vom Benutzer.
- Mit der Terminal-Emulationstaste, die dem Befehl `EMULATE` zugeordnet wurde, wird die Steuerung an die Prozedurdatei zurückgegeben. Die Verarbeitung wird dann mit dem nächsten Befehl fortgesetzt.

Syntax

```
EMULATE
```

Prozedurdateibeispiel

Makete.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

15 ERASE

Beschreibung

Löscht eine Datei.

Syntax

```
ERASE pfad
```

dabei ist *pfad*:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Beispiele

- Die Datei *Test.ncp* löschen:

  ```
  ERASE Test.ncp
  ```

- Die Datei im Verzeichnis *SAG* löschen, die durch den Inhalt der Variablen #FILENAME und #FILEEXT definiert ist.

  ```
  ERASE C:\SAG\ #FILENAME #FILEEXT
  ```

- Die Datei *\SAG\Test.ncp* im aktuellen Laufwerk löschen:

```
ERASE \SAG\Test.ncp
```

Zurückgegebene Variablen

#RC (SUCCESS, wenn ERASE erfolgreich war. FAILURE, wenn ERASE nicht erfolgreich war.)

Verwandte Befehle

DOSDIR, RENAME

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

16 EXECTASK

Beschreibung

Führt einen unter Entire Connection definierten Task aus.

Syntax

```
EXECTASK taskname [taskparameter]...
```

taskname ist ein unter Entire Connection definierter Task.

taskparameter ist ein Parameter, der einem für den angegebenen Task erforderlichen Eingabeparameter entspricht. Sie können auch mehrere durch Leerzeichen getrennte Task-Parameter angeben.

Beispiele

- Den Task EDITOR, der einen lokalen Editor aufruft, ohne Parameter ausführen:

  ```
  EXECTASK EDITOR
  ```

- Den Task EDITOR, der einen lokalen Editor aufruft, mit Parametern (in diesem Fall der Name der zu bearbeitenden Datei) ausführen:

  ```
  EXECTASK EDITOR Test.abc
  ```

- Den Task MEINTASK ausführen und ihm den Wert der lokalen Variable #PARM1 übergeben:

```
EXECTASK MEINTASK #PARM1
```

- Den Task ausführen, der in der lokalen Variable `#PARM1` definiert ist:

```
EXECTASK #PARM1
```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Nein

17 EXECUTE

Beschreibung

Führt eine Prozedurdatei aus.

Sie können bis zu 7 Prozedurdateien verschachteln.

Syntax

```
EXECUTE pfad [prozedurdateiparameter] ...
```

dabei ist *pfad*:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Wenn Sie keine *erweiterung* angeben, wird für die Prozedurdatei automatisch die Namenserweiterung "ncp" vergeben.

Wenn Sie kein *laufwerk* und/oder *verzeichnis* angeben, versucht Entire Connection die Prozedurdatei im aktuellen Laufwerk und Verzeichnis zu finden. Wenn das nicht gelingt, sucht Entire Connection die Datei im Prozedurenverzeichnis, das in den Benutzereigenschaften definiert wurde.

Sie können bis zu 9 *prozedurdateiparameter* angeben (durch Leerzeichen getrennt), die den Eingabeparametern der angegebenen Prozedurdatei entsprechen. Diese werden in den lokalen Variablen #PARM1 bis #PARM9 gespeichert. Die Variable #PARM0 enthält die komplette Pfadangabe für die ausgeführte Prozedurdatei. Die Variable #PARMNO enthält die Anzahl der übergebenen Parameter (00 bis 09).

Beispiele

- Die Prozedurdatei *Test1.ncp* ohne Parameter ausführen:

```
EXECUTE Test1
```

- Die Prozedurdatei *Meinproz* ausführen und ihr den Wert der lokalen Variable #PARM1 übergeben:

```
EXECUTE Meinproz #PARM1
```

- Die Prozedurdatei *Test2.abc* mit 2 Parametern ausführen:

```
EXECUTE Test2.abc eins zwei
```

- Die Prozedurdatei *\MEINDAT\Test3.ncp* mit 3 Parametern ausführen:

```
EXECUTE \MEINDAT\Test3 eins zwei drei
```

- Die Prozedurdatei, die in der lokalen Variable #PARM1 definiert ist, ohne Parameter ausführen:

```
EXECUTE #PARM1
```

Prozedurdateibeispiele

Ncpnest.ncp, *Findfile.ncp*

Zurückgegebene Variablen

#RC (SUCCESS, wenn kein Laufzeitfehler auftrat. FAILURE, wenn ein Laufzeitfehler auftrat.)

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Ja

18 EXIT

Beschreibung

Verlässt eine Prozedurdatei und kehrt zur vorherigen Prozedurdatei oder zu Entire Connection zurück.

Mit diesem Befehl können Sie auch Informationen an die aufrufende Prozedurdatei zurückgeben.

Der Befehl `EXIT` wird am Ende einer Prozedurdatei immer ausgeführt, auch wenn der letzte Befehl nicht ausdrücklich `EXIT` ist.

Syntax

```
EXIT [{zeichenkette | variable}...]
```

Beispiele

- Die Prozedurdatei verlassen:

  ```
  EXIT
  ```

- Die aktuelle Uhrzeit an die aufrufende Prozedurdatei zurückgeben:

  ```
  EXIT 'Die Uhrzeit ist ' *TIME
  ```

- Eine Meldung zurückgeben, dass die in der lokalen Variable `#FILESPEC` definierte Datei nicht gefunden wurde:

```
EXIT 'Datei ' #FILESPEC ' wurde nicht gefunden'
```

- Eine Meldung zurückgeben, dass die Prozedurdatei erfolgreich ausgeführt wurde:

```
EXIT 'Erfolg'
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

`+RC` - enthält alle mit `EXIT` übergebenen Werte (globale Variablen sind immer verfügbar).

Verwandte Befehle

`CANCEL`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

19 GOTO

Beschreibung

Verzweigt zu einer anderen Stelle in der Prozedurdatei.

Die Sprungmarke kann vor oder nach der `GOTO`-Anweisung stehen.

Syntax

```
GOTO sprungmarke
```

Beispiel

- Zur `WEITER` genannten Sprungmarke verzweigen:

```
GOTO WEITER
```

Prozedurdateibeispiele

Findfile.ncp, Copyscr.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

`PERFORM`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

20 IF/IFNOT

Beschreibung

`IF` überprüft eine Bedingung. `IFNOT` ist die Kurzschreibweise für `IF *SCREEN NE`.

Wenn die Bedingung wahr ist, wird die Anweisung ausgeführt. Wenn die Bedingung falsch ist, wird die nächste Anweisung ausgeführt.

Text in einfachen oder doppelten Anführungszeichen wird auf Groß- und Kleinschreibung überprüft.

Syntax

```
IF [variable] [operator] variable befehl
```

```
IFNOT variable befehl
```

Für *`variable`* können Sie alle Variablen von Entire Connection benutzen (z.B. `#PARM1` oder `#CNT1`). Bei einer Großrechneranwendung ist die Variable `*SCREEN` hilfreich (Syntax siehe unten). Damit können Sie feststellen, welcher Bildschirm gerade übertragen wird, ohne ihn anzuzeigen. Sie können dann entsprechend reagieren.

`operator` ist einer der folgenden Werte:

`EQ`	gleich
`NE`	ungleich
`GE`	größer gleich
`LE`	kleiner gleich
`GT`	größer als
`LT`	kleiner als

Für *`befehl`* können Sie jeden Befehl von Entire Connection, außer den Folgenden, benutzen:

```
IF
IFNOT
WAITFOR
```

Wenn Sie die optionalen Operanden *variable* und *operator* bei der IF-Anweisung weglassen, wird die Variable *SCREEN (gesamter Bildschirm) und der Operator EQ benutzt.

Syntax für *SCREEN

Die Variable *SCREEN kann nur einmal pro IF/IFNOT benutzt werden.

```
*SCREEN [zeile spalte[länge]]
```

zeile ist ein Wert zwischen 1 und der maximalen Zeilenanzahl + 1.

spalte ist ein Wert zwischen 1 und der maximalen Zeilenlänge.

länge ist ein Wert zwischen 1 und der Bildschirmgröße.

Zum Beispiel:

*SCREEN steht für: gesamter Bildschirm.

*SCREEN 2 1 steht für: ab Zeile 2, Spalte 1 bis zum Ende des Bildschirms.

*SCREEN 2 1 80 steht für: ab Zeile 2, Spalte 1 die nächsten 80 Positionen.

Beispiele

- Den gesamten Bildschirm nach NEXT durchsuchen. Wenn NEXT vorkommt, zur Sprungmarke NEXT verzweigen:

  ```
  IF 'NEXT' GOTO NEXT
  ```

- Den gesamten Bildschirm nach NEXT durchsuchen. Wenn NEXT nicht vorkommt, zur Sprungmarke WEITER verzweigen:

  ```
  IFNOT 'NEXT' GOTO WEITER
  ```

- Ab Zeile 2, Spalte 1 die nächsten vier Positionen nach NEXT durchsuchen:

```
IF *SCREEN 2 1 4 EQ 'NEXT' GOTO WEITER
```

- Ab Zeile 2, Spalte 1 die nächsten 80 Positionen nach `NEXT` durchsuchen:

```
IF *SCREEN 2 1 80 EQ 'NEXT' GOTO WEITER
```

- Ab der in der Variable `+ROW` enthaltenen Zeile und der in `+COL` enthaltenen Spalte nach dem Inhalt der lokalen Variable `#PARM1` suchen; die Anzahl der zu durchsuchenden Positionen wird durch den Wert in der Variable `#CNT3` festgelegt:

```
IF *SCREEN +ROW +COL #CNT3 EQ #PARM1
GOTO WEITER
```

- Die definierte Tastatureingabe an den Host senden, wenn der Inhalt der lokalen Variable `#PARM1` nicht leer ist:

```
IF #PARM1 NE ' ' TYPE '*NAT' CR
```

Prozedurdateibeispiele

Findfile.ncp, *Copyscr.ncp*

Zurückgegebene Variablen

Wenn die Bedingung für den Befehl `IF *SCREEN EQ` wahr ist, wird die Bildschirmposition der Zeichenkette in den folgenden lokalen Variablen zurückgegeben:

`#ROW` - die gültigen Werte liegen zwischen 1 und der maximalen Zeilenanzahl + 1

`#COL` - die gültigen Werte liegen zwischen 1 und der maximalen Zeilenlänge

Verwandte Befehle

`WAITFOR`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Ja

21 INCR

Beschreibung

Fügt 1 zu den globalen und lokalen Zählervariablen oder Bildschirmpositionsvariablen hinzu.

Syntax

```
INCR {zähler|bildschirmpositionsvariable}...
```

zähler steht für eine der folgenden Variablen:

#CNT0 bis #CNT9 (lokal)
+CNT0 bis +CNT9 (global)

Der gültige Zahlenbereich für die Zählervariablen liegt zwischen 0 und 32767.

bildschirmpositionsvariable steht für eine der folgenden Variablen:

#ROW, #COL, #LENGTH (lokal)
+ROW, +COL, +LENGTH (global)

Beispiele

- 1 zum globalen Zähler +CNT1 hinzufügen:

```
INCR +CNT1
```

- 1 zu den lokalen Zählern #CNT1 und #CNT2 hinzufügen:

```
INCR #CNT1 #CNT2
```

- 1 zum lokalen Zähler `#CNT1` und globalen Zähler `+CNT1` hinzufügen:

```
INCR #CNT1 +CNT1
```

- 2 zum lokalen Zähler `#CNT1` hinzufügen:

```
INCR #CNT1 #CNT1
```

Prozedurdateibeispiele

Findfile.ncp, Copyscr.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

DECR, RESET, SET

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

22 INPUT

Beschreibung

Fordert den Benutzer zur Eingabe auf.

Syntax

```
INPUT parametervariable länge meldung [meldung]
```

parametervariable steht für eine der folgenden Variablen:

+PARM1 bis +PARM9 (global)
#PARM1 bis #PARM9 (lokal)

Bei einer globalen *parametervariable* wird jedes an der Eingabeaufforderung eingegebene Zeichen als Stern (*) angezeigt. Dies ist bei der Eingabe von Passwörtern oder anderen sensiblen Daten wichtig. Wenn eine lokale *parametervariable* bereits einen Wert enthält, wird dieser Wert angezeigt.

länge steht für einen Wert zwischen 1 und 72.

meldung kann eine Zeichenkette oder eine Variable sein. Der Benutzer wird durch eine ein- oder zweizeilige Meldung aufgefordert, einen Wert einzugeben. Dieser Wert wird in der *parametervariable* gespeichert.

Beispiele

- Aufforderung zur Eingabe eines 8 Zeichen langen Benutzerkennzeichens:

```
INPUT #PARM1 8 'Geben Sie Ihr Benutzerkennzeichen ein'
```

- Aufforderung zur Eingabe eines 8 Zeichen langen Passworts, das nicht angezeigt wird:

```
INPUT +PARM1 8 'Geben Sie Ihr Passwort ein'
```

- Aufforderung zur Eingabe eines Werts, dessen Länge durch die lokalen Variable #CNT1 festgelegt wird. Der Text der Eingabeaufforderung ist in der lokalen Variable #PARM2 enthalten:

```
INPUT #PARM1 #CNT1 #PARM2
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

23 LEARN

Beschreibung

Erstellt eine Prozedurdatei im Lernmodus.

Um den Lernmodus ein- und auszuschalten, müssen Sie die hierfür definierte Tastenkombination drücken.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+L
BS2000 / BS2000KEYS	ALT+T
VTxxx / VT220PC	ALT+L

Nach der Eingabe des Befehls `LEARN` werden Sie nach einem Dateinamen für die Prozedurdatei gefragt. Anschließend wird jede Tastatureingabe in die Prozedurdatei geschrieben.

Wenn der Lernmodus eingeschaltet ist, erscheint im Terminal-Emulationsbildschirm ein L in Spalte 76 der Statuszeile.

Syntax

```
LEARN
```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

24 LOG

Beschreibung

Schreibt eine Meldung in die Log-Datei.

Der Name der Log-Datei ist *<benutzername>.log*. Sie befindet sich im Log-/Trace-Verzeichnis. Wenn die Datei noch nicht vorhanden ist, wird sie erstellt. Ansonsten wird die Meldung der bestehenden Datei angehängt.

Syntax

```
LOG {zeichenkette|variable}...
```

Wenn die Meldung in die Log-Datei geschrieben wird, steht kein Leerzeichen zwischen den Operanden.

Beispiele

- Eine Meldung mit der aktuellen Uhrzeit in die Log-Datei schreiben:

  ```
  LOG 'Die Uhrzeit ist ' *TIME
  ```

- Die Meldung in die Log-Datei schreiben, dass der in der Variable `#FILESPEC` enthaltene Dateiname nicht gefunden wurde:

  ```
  LOG 'Datei ' #FILESPEC ' wurde nicht gefunden'
  ```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Ja

25 MD

Beschreibung

Erstellt ein Verzeichnis.

Syntax

```
MD {[laufwerk:]\verzeichnis\...}
```

Beispiele

- Ein Verzeichnis mit dem Namen *NCP* im aktuellen Verzeichnis erstellen:

  ```
  MD NCP
  ```

- Ein Verzeichnis mit dem Namen *NCP* auf der obersten Verzeichnisebene von Laufwerk C erstellen:

  ```
  MD C:\NCP
  ```

- Ein Verzeichnis mit dem Namen *NCP* im Verzeichnis *SAG* erstellen:

  ```
  MD \SAG\NCP
  ```

- Ein Verzeichnis mit Hilfe des Inhalts der Variablen #FILEDRIVE und #FILEPATH erstellen:

```
MD #FILEDRIVE ':' #FILEPATH
```

Zurückgegebene Variablen

#RC (SUCCESS, wenn MD erfolgreich war. FAILURE, wenn MD nicht erfolgreich war.)

Verwandte Befehle

CD, CHDRIVE, RD

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

26 MONITOR

Beschreibung

Schreibt kommunikationsspezifische Daten auf die Festplatte.

Dies entspricht der Funktion **Trace bei Kommunikation** auf der Eigenschaftenseite **Test** der Session-Eigenschaften.

Die Daten werden in die Datei *Mon<nn>.trc* im Log-/Trace-Verzeichnis geschrieben, wobei *nn* eine Zahl ist, die jeweils um 1 erhöht wird.. Wenn die Datei noch nicht vorhanden ist, wird die Datei *Mon00.trc* erstellt. Ansonsten werden die Informationen der bestehenden Datei angehängt.

Die Daten werden solange aufgezeichnet, bis Sie den Befehl `MONITOR` erneut eingeben. Sie können diesen Modus mit einer Tastenkombination ein- und ausschalten.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+M
BS2000 / BS2000KEYS	STRG+M
VTxxx / VT220PC	ALT+M

Wenn Daten aufgezeichnet werden, erscheint im Terminal-Emulationsbildschirm ein M in Spalte 5 der Statuszeile.

⚠ **Wichtig:** Dieser Befehl dient der Problemlösung bei Entire Connection und sollte nur unter Anleitung Ihres Technischen Supports benutzt werden.

Syntax

```
MONITOR
```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Nein
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

27 MSG

Beschreibung

Gibt eine Meldung aus.

Die Meldung wird im Ausgabefenster der Terminal-Anwendung angezeigt.

Syntax

```
MSG {zeichenkette|variable}...
```

Wenn die Meldung angezeigt wird, steht kein Leerzeichen zwischen den Operanden.

Beispiele

- Eine Meldung mit der aktuellen Uhrzeit ausgeben:

```
MSG 'Die Uhrzeit ist ' *TIME
```

- Die Meldung ausgeben, dass der in der Variable #FILESPEC enthaltene Dateiname nicht gefunden wurde:

```
MSG 'Datei ' #FILESPEC ' wurde nicht gefunden'
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

28 OPEN-I

Beschreibung

Öffnet eine Datei, aus der Daten gelesen werden. Der Buchstabe I im Befehlsname steht für „input file“ (Eingabedatei).

Die Datei ist für alle geschachtelten Prozedurdateien verfügbar. Sie können zum Beispiel den Befehl `READ` aus mehr als einer Prozedurdatei heraus benutzen.

Syntax

```
OPEN-I dateinummer pfad
```

dateinummer ist eine Zahl zwischen 1 und 4.

pfad ist:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Beispiele

- *Test.ncp* als Eingabedatei Nummer 1 öffnen:

  ```
  OPEN-I 1 Test.ncp
  ```

- Die in der Variable `#FILESPEC` angegebene Datei als Eingabedatei Nummer 2 öffnen:

```
OPEN-I 2 #FILESPEC
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

#RC (SUCCESS, wenn OPEN-I erfolgreich war. FAILURE, wenn OPEN-I nicht erfolgreich war.)

Verwandte Befehle

READ, CLOSE

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

29 OPEN-O

Beschreibung

Öffnet eine Datei, in die Daten geschrieben werden. Der Buchstabe O im Befehlsname steht für „output file“ (Ausgabedatei).

Die Datei ist für alle geschachtelten Prozedurdateien verfügbar. Sie können zum Beispiel den Befehl `WRITE` aus mehr als einer Prozedurdatei heraus benutzen.

Wenn Sie in eine bestehende Datei schreiben, werden die neuen Daten angehängt.

Syntax

```
OPEN-O dateinummer pfad
```

dateinummer ist eine Zahl zwischen 1 und 4.

pfad ist:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Beispiele

- *Test.ncp* als Ausgabedatei Nummer 1 öffnen:

  ```
  OPEN-O 1 Test.ncp
  ```

- Die in der Variable #`FILESPEC` angegebene Datei als Ausgabedatei Nummer 2 öffnen:

```
OPEN-O 2 #FILESPEC
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

#RC (SUCCESS, wenn OPEN-O erfolgreich war. APPEND, wenn die Datei bereits existiert. FAILURE, wenn OPEN-O nicht erfolgreich war.)

Verwandte Befehle

WRITE, CLOSE

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

30 OS

Beschreibung

Die Befehle `OS` und `DOS` sind identisch. Aus Kompatibilitätsgründen wird der Befehl `OS` noch unterstützt.

Siehe die Beschreibung des Befehls `DOS`.

31 PAUSE

Beschreibung

Hält die Verarbeitung für eine bestimmte Zeitspanne an, maximal bis zur angegebenen Zeitspanne, minimal bis zum Eintreffen einer Antwort vom Host.

Während der Pause wird geprüft, ob eine Antwort vom Host vorliegt. Wenn die Antwort vor dem definierten Ende der Pause eintrifft, wird die Ausführung der Prozedurdatei sofort fortgesetzt.

Syntax

```
PAUSE millisekunden
```

millisekunden kann ein Wert zwischen 0 und 32000 sein, d.h. zwischen 0 Tausendstel einer Sekunde und 32 Sekunden. Der Wert 1000 entspricht 1 Sekunde.

Beispiele

- Verarbeitung für 1 Sekunde (1000 Millisekunden) anhalten:

  ```
  PAUSE 1000
  ```

- Verarbeitung für die Anzahl der Sekunden anhalten, die in der Variable #CNT1 definiert sind:

  ```
  PAUSE #CNT1
  ```

Prozedurdateibeispiel

Vars.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

ELAPSETIME, WAIT, WAITFOR, WAITM, WAITUNTIL

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Ja

32 PERFORM

Beschreibung

Verzweigt zu einer anderen Stelle in der Prozedurdatei und führt die dort definierten Anweisungen aus.

Der Befehl `RETURN` setzt die Verarbeitung mit der Anweisung fort, die direkt nach dem Befehl `PERFORM` steht. Die Sprungmarke kann vor oder nach dem `PERFORM`-Befehl stehen.

`PERFORM`-Befehle können sieben Ebenen tief verschachtelt werden.

Syntax

```
PERFORM sprungmarke
```

Beispiel

- Zur `ROUTINE1` genannten Sprungmarke verzweigen und die dort definierten Anweisungen ausführen:

  ```
  PERFORM ROUTINE1
  ```

Zurückgegebene Variablen

Keine

Verwandte Befehle

`RETURN`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

33 POPDIR

Beschreibung

Kehrt an die Position in der Verzeichnishierarchie zurück, die mit dem Befehl `PUSHDIR` gespeichert wurde.

Sie können diesen Befehl nur verwenden, wenn vorher der Befehl `PUSHDIR` verwendet wurde.

Syntax

```
POPDIR
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

`PUSHDIR`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

34 PUSHDIR

Beschreibung

Speichert die aktuelle Position in der Verzeichnishierarchie.

Sie können diesen Befehl nur verwenden, wenn vorher der Befehl `DOSDIR` verwendet wurde.

Dieser Befehl ist zum Beispiel hilfreich, wenn Sie verschiedene Verzeichnisse nach bestimmten Dateien durchsuchen.

Syntax

```
PUSHDIR
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

`DOSDIR`, `POPDIR`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

35 QA

Beschreibung

Zeichnet Sessions, Terminal-Emulationsbildschirme und Benutzereingaben im Entire Test Client Format auf Festplatte auf. Diese Daten werden vom Support und der Entwicklung der Software AG zum Reproduzieren von Problemen verwendet.

Mit der Tastenkombination STRG+Q können Sie diesen Modus ein- und ausschalten.

Wenn Sie diesen Modus mit dem Befehl QA einschalten, erscheint ein Dialogfeld, in dem Sie ein Verzeichnis und einen Dateinamen für die aufzuzeichnenden Daten eingeben müssen. Die Namenserweiterung der Datei ist immer *qau*.

Die Daten werden solange aufgezeichnet, bis Sie den Befehl QA erneut eingeben. Dann erscheint ein anderes Dialogfeld, in dem Sie eine kurze Beschreibung eingeben können (bis zu 39 Zeichen). Wenn Sie die Befehlsschaltfläche **Abbrechen** wählen, wird keine Beschreibung mit der Datei gespeichert.

Wenn Daten aufgezeichnet werden, erscheint im Terminal-Emulationsbildschirm ein Q in Spalte 61 der Statuszeile.

Syntax

```
QA
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

REC_BUFF, REC_SCR, REC_XFER

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

36 QUIT

Beschreibung

Schließt ein Terminal von Entire Connection.

Wenn eine Prozedur zum Abmelden definiert wurde, wird diese vor dem Schließen des Terminals ausgeführt.

Syntax

```
QUIT
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

`CANCEL`, `EXIT`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Nein

37 RD

Beschreibung

Löscht ein Verzeichnis.

Syntax

```
RD {[laufwerk:]\verzeichnis\...}
```

Beispiele

- Das Verzeichnis mit dem Namen *NCP* im aktuellen Verzeichnis löschen:

  ```
  RD NCP
  ```

- Das Verzeichnis mit dem Namen *NCP* auf der obersten Verzeichnisebene von Laufwerk C löschen:

  ```
  RD C:\NCP
  ```

- Das Verzeichnis mit dem Namen *NCP* im Verzeichnis *SAG* löschen.

  ```
  RD \SAG\NCP
  ```

- Das Verzeichnis löschen, das durch den Inhalt der Variablen #FILEDRIVE und #FILEPATH definiert ist:

```
RD #FILEDRIVE ':' #FILEPATH
```

Zurückgegebene Variablen

#RC (SUCCESS, wenn RD erfolgreich war. FAILURE, wenn RD nicht erfolgreich war.)

Verwandte Befehle

CD, CHDRIVE, MD

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

38 READ

Beschreibung

Liest Daten aus einer Eingabedatei.

Die Eingabedatei ist für alle geschachtelten Prozedurdateien verfügbar. Sie können zum Beispiel den Befehl `READ` aus mehr als einer Prozedurdatei heraus benutzen.

Syntax

```
READ dateinummer parametervariable ...
```

`dateinummer` ist eine Zahl zwischen 1 und 4.

`parametervariable` steht für eine oder maximal 9 der folgenden Variablen:

`+PARM1` bis `+PARM9` (global)
`#PARM1` bis `#PARM9` (lokal)

Das Ergebnis einer `READ`-Operation wird in diesen Variablen gespeichert.

Wenn Sie nur eine *`parametervariable`* angeben, wird der gesamte Datensatz in dieser Variable gespeichert.

Wenn Sie mehr als eine *`parametervariable`* angeben, wird der Datensatz zuerst in Felder zerlegt, die durch Leerzeichen getrennt sind. Danach wird das erste Feld in die erste Variable gestellt usw.

Wenn Sie mehr Variablen als Felder angeben, werden die unbenutzten Variablen auf Null zurückgesetzt. Wenn es mehr Felder als Variablen gibt, enthält die letzte Variable den Rest des Datensatzes.

Der Befehl `READ` unterstützt eine maximale ASCII-Datensatzlänge von 255 Bytes. Deshalb können in einer *`parametervariable`* auch bis zu 255 Bytes gespeichert werden.

Beispiele

- Eingabedatei Nummer 1 lesen und den gesamten Datensatz in einer Variable speichern.

```
READ 1 #PARM1
```

- Eingabedatei Nummer 2 lesen und den Datensatz in vier Variablen speichern:

```
READ 2 #PARM1 #PARM2 #PARM3 #PARM4
```

Prozedurdateibeispiel

Copyscr.ncp

Zurückgegebene Variablen

#RC (SUCCESS, wenn ein Datensatz erfolgreich gelesen wurde. EOF, wenn die Datei-Ende-Marke (End-of-File) erreicht wurde. FAILURE, wenn der Befehl READ nicht erfolgreich war.)

Verwandte Befehle

OPEN-I, CLOSE

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

39 REC_BUFF

Beschreibung

Zeichnet Daten unübersetzt vom Terminal-Emulations-Buffer auf Festplatte auf.

Die Buffer werden in die Datei *Buffer.trc* im Log-/Trace-Verzeichnis geschrieben. Wenn die Datei noch nicht vorhanden ist, wird sie erstellt. Ansonsten werden alle neuen Buffer einer bestehenden Datei angehängt.

Der Buffer wird nur auf die Festplatte geschrieben, wenn CR an den Großrechner übermittelt wird. Die Buffer werden solange aufgezeichnet, bis Sie den Befehl `REC_BUFF` erneut eingeben.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+B
BS2000 / BS2000KEYS	STRG+B
VTxxx / VT220PC	ALT+B

⚠ **Wichtig:** Dieser Befehl dient der Problemlösung bei Entire Connection und sollte nur unter Anleitung Ihres Technischen Supports benutzt werden.

Wenn Buffer aufgezeichnet werden, erscheint im Terminal-Emulationsbildschirm ein B in Spalte 75 der Statuszeile.

Syntax

```
REC_BUFF
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

QA, REC_SCR, REC_XFER

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

40 REC_SCR

Beschreibung

Zeichnet Terminal-Emulationsbildschirme auf Festplatte auf.

Die Bildschirme werden in die Datei *Screen.trc* im Log-/Trace-Verzeichnis geschrieben. Wenn die Datei noch nicht vorhanden ist, wird sie erstellt. Ansonsten werden alle neuen Bildschirme einer bestehenden Datei angehängt.

Die Bildschirme werden solange aufgezeichnet, bis Sie den Befehl `REC_SCR` erneut eingeben. Sie können diesen Modus mit einer Tastenkombination ein- und ausschalten.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+S
BS2000 / BS2000KEYS	STRG+S
VTxxx / VT220PC	ALT+S

Die Bildschirme werden nur im Terminal-Emulationsmodus aufgezeichnet oder wenn eine Prozedurdatei oder ein API-Programm ausgeführt wird und die Systemvariable `DISPLAY` auf `ON` gesetzt ist.

Wenn Bildschirme aufgezeichnet werden, erscheint im Terminal-Emulationsbildschirm ein S in Spalte 79 der Statuszeile.

Syntax

```
REC_SCR
```

Prozedurdateibeispiel

Recscr.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

QA, REC_BUFF, REC_XFER

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Ja

41 REC_XFER

Beschreibung

Zeichnet die Datentransfer-Buffer auf Festplatte auf.

Die Datentransfer-Buffer werden in die Datei *Xfer.trc* im Log-/Trace-Verzeichnis geschrieben. Wenn die Datei noch nicht vorhanden ist, wird sie erstellt. Ansonsten werden alle neuen Datentransfer-Buffer einer bestehenden Datei angehängt.

Die Datentransfer-Buffer werden solange aufgezeichnet, bis Sie den Befehl `REC_XFER` erneut eingeben. Sie können diesen Modus mit einer Tastenkombination ein- und ausschalten.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+F
BS2000 / BS2000KEYS	STRG+X
VTxxx / VT220PC	ALT+F

Wichtig: Dieser Befehl dient der Problemlösung bei Entire Connection und sollte nur unter Anleitung Ihres Technischen Supports benutzt werden.

Wenn Datentransfer-Buffer aufgezeichnet werden, erscheint im Terminal-Emulationsbildschirm ein X in Spalte 80 der Statuszeile.

Syntax

```
REC_XFER
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

QA, REC_BUFF, REC_SCR

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Ja

42 RECALL

Beschreibung

Zeigt im aktuellen Eingabefeld bis zu 20 der bei der Terminal-Emulation zuletzt eingegebenen Befehle und/oder Zeichenketten an.

Die `RECALL`-Funktion muss hierfür in den Session-Eigenschaften eingeschaltet werden.

Syntax

```
RECALL
```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Nein
Befehlszeile:	Nein
Taste:	Ja
API:	Nein

43 RENAME

Beschreibung

Benennt eine Datei um oder verschiebt eine Datei.

Syntax

```
RENAME pfad-1 pfad-2
```

dabei ist *pfad*:

{[[*laufwerk*:]*verzeichnis*\...\]*dateiname*[.*erweiterung*]}

Um eine Datei zu verschieben, geben Sie für *pfad-2* ein anderes Verzeichnis an (das Laufwerk muss dasselbe sein). Beim Verschieben können Sie (optional) die Datei auch umbenennen. Dazu geben Sie einen neuen Dateinamen und/oder eine neue Namenserweiterung ein.

Beispiele

- Die Datei *Test.ncp* in *Test.xyz* umbenennen:

  ```
  RENAME Test.ncp Test.xyz
  ```

- Die Datei *Test.ncp* in das Verzeichnis *\SAG\NCP* verschieben:

  ```
  RENAME Test.ncp \SAG\NCP\Test.ncp
  ```

- Die Datei *Test.ncp* in das Verzeichnis *\SAG\NCP* verschieben und dabei in *Test2.ncp* umbenennen:

```
RENAME Test.ncp \SAG\NCP\Test2.ncp
```

Zurückgegebene Variablen

#RC (SUCCESS, wenn RENAME erfolgreich war. FAILURE, wenn RENAME nicht erfolgreich war.)

Verwandte Befehle

ERASE

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

44 RESET

Beschreibung

Setzt lokale oder globale Variablen auf Null oder Leerzeichen zurück.

Syntax

```
RESET {globale-variable|lokale-variable}...
```

oder

```
RESET {GLOBALS|LOCALS}
```

Beispiele

- Die Variable #PARM1 zurücksetzen:

  ```
  RESET #PARM1
  ```

- Alle globalen Variablen zurücksetzen:

  ```
  RESET GLOBALS
  ```

- Alle lokalen Variablen zurücksetzen:

  ```
  RESET LOCALS
  ```

- Die lokalen Variablen #PARM1 und #CNT1 und die globale Variable +PARM1 zurücksetzen:

```
RESET #PARM1 #CNT1 +PARM1
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

DECR, INCR, SET, SHIFT

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

45 RETURN

Beschreibung

Setzt die Verarbeitung mit der Anweisung fort, die direkt nach dem Befehl `PERFORM` steht.

Sie können diesen Befehl nur verwenden, wenn vorher der Befehl `PERFORM` verwendet wurde.

Syntax

```
RETURN
```

Zurückgegebene Variablen

Keine

Verwandte Befehle

`PERFORM`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

46 REVEAL

Beschreibung

Zeigt die Feldattribute der Emulation und den Wert des ASCII-Zeichens an der aktuellen Cursor-Position an.

Dieser Befehl gilt nur für die Terminal-Emulation. Die folgende Information erscheint im Terminal-Emulationsbildschirm ab Spalte 38 der Statuszeile:

```
aaa/bbb fields ccccc dd
```

aaa steht für die Anzahl der Eingabefelder auf dem Bildschirm.

bbb steht für die Anzahl der ungeschützten Eingabefelder auf dem Bildschirm.

ccccc gibt an, dass sich der Cursor in einem der folgenden Felder befindet:

FM	Field Mark (Attribut)
PROT	geschütztes Feld
NUM	ungeschütztes numerisches Feld
ALPHA	ungeschütztes alphanumerisches Feld
NON-DISP	Non-Display-Feld

dd steht für den ASCII-Wert des Zeichens (in Hexadezimaldarstellung) an der aktuellen Cursor-Position.

Beispiel:

```
043/001 fields PROT 6e
```

Wenn der Befehl `REVEAL` aktiv ist, kann kein Datentransfer durchgeführt werden. Die Dateitransfer-Buffer werden auf dem Bildschirm dargestellt. Die Informationen werden solange angezeigt, bis Sie den Befehl `REVEAL` erneut eingeben. Sie können diesen Modus mit einer Tastenkombination ein- und ausschalten.

Die Standardtastenkombination ist abhängig von der Art der Session und dem entsprechenden Tastenschema.

Art der Session / Standardtastenschema	Standardtastenkombination
TN3270 / SAGKEYS1	STRG+R
BS2000 / BS2000KEYS	ALT+R
VTxxx / VT220PC	ALT+R

Syntax

```
REVEAL
```

Prozedurdateibeispiel

Revattr.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Ja

47 RSPMONITOR

Beschreibung

Schaltet die Antwortzeit-Überwachung für die Terminal-Emulation ein und aus.

Wenn Sie die Antwortzeit-Überwachung einschalten, werden die unten aufgeführten Variablen auf Null zurückgesetzt.

Wenn die Antwortzeit-Überwachung eingeschaltet ist, erscheint im Terminal-Emulationsbildschirm ein R in Spalte 75 der Statuszeile.

Syntax

```
RSPMONITOR
```

Zurückgegebene Variablen

Die folgenden dynamischen Variablen werden immer gesetzt (berechnet), wenn Sie eine Terminal-Emulationstaste drücken (CLEAR, CR, PF1 usw.).

*RSPCOUNT	Anzahl der Transaktionen (Tastatureingaben)
*RSPTIME	aktuelle Antwortzeit
*RSPAVG	durchschnittliche Antwortzeit
*RSPMIN	minimale Antwortzeit
*RSPMAX	maximale Antwortzeit
*RSPTOTAL	gesamte Antwortzeit

Die Antwortzeit wird in Sekunden und hundertstel Sekunden angegeben.

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Nein

48 SCHEDTOP/SCHEDULE

Beschreibung

Plant die Ausführung von weiteren Tasks oder Prozedurdateien.

Die Ausführung beginnt, wenn die aktuelle Prozedurdatei beendet wird.

Mit `SCHEDTOP` planen Sie Tasks oder Prozedurdateien auf der Grundlage von „first-in-first-out“ (FIFO). Mit `SCHEDULE` planen Sie Tasks oder Prozedurdateien auf der Grundlage von „last-in-first-out“ (LIFO).

Vorsicht: Verwenden Sie die Befehle `SCHEDULE` und `SCHEDTOP` nicht in derselben Prozedurdatei oder innerhalb einer Gruppe verschachtelter Prozedurdateien.

Syntax

```
SCHEDTOP {task|prozedurdatei} [parameter] ...
```

```
SCHEDULE {task|prozedurdatei} [parameter] ...
```

task ist ein unter Entire Connection definierter Task.

prozedurdatei ist eine unter Entire Connection definierte Prozedurdatei.

parameter ist ein gültiger Eingabeparameter für den Task oder die Prozedurdatei.

Beispiele

- Den Task `BUDGET` planen, der ein Natural-Programm auf dem Großrechner ausführt und die Daten in die ASCII-Datei *Budget.ncd* herunterlädt:

```
SCHEDULE BUDGET Budget.ncd
```

- Den Task `BUDGET` als den ersten Task planen, der nach dem Beenden der aktuellen Prozedurdatei ausgeführt werden soll:

```
SCHEDTOP BUDGET Budget.ncd
```

- Einen Task oder eine Prozedurdatei planen, dessen Name in der Variable `#PARM1` enthalten ist:

```
SCHEDULE #PARM1
```

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Ja

49 SET

Beschreibung

Weist einer lokalen Variablen, globalen Variablen oder Systemvariablen einen Wert zu. Einer dynamischen Variablen können Sie keinen Wert zuweisen.

Die mit `SET` zugewiesenen Werte gelten nur für die aktuelle Session mit Entire Connection.

Über Befehlszeile und Terminal-Emulationstasten können Sie nur globalen Variablen und Systemvariablen Werte zuweisen.

Syntax

```
SET variable {zeichenkette|variable}...
```

Beispiele

- Die Systemvariable `LOGON` auf `YES` setzen:

```
SET LOGON YES
```

- Der globalen Variablen `+PARM1` den Wert `LOGON = YES` zuweisen:

```
SET +PARM1 'LOGON = YES'
```

- Den Dateinamen für `PCFILE 5` auf *Test.ncd* setzen:

```
SET PCFILE 5 DOWN DATA Test.ncd
```

- Den XSL-Stylesheet-Typ und den Namen des Stylesheets für das Herunterladen nach XML festlegen:

```
SET PCFILE 7 DOWN CONVERT text/xsl http://PCxyz/xml/employ2.xsl
```

  Siehe auch: *Dateiname mit dem Befehl SET angeben* in der Dokumentation zur Terminal-Emulation.

- Der lokalen Variablen `#PARM1` den Wert zuweisen, der auf dem Terminal-Emulationsbildschirm ab Zeile 2, Spalte 1 steht und zwar für eine Länge von 80 Zeichen:

```
SET #PARM1 *SCREEN 2 1 80
```

  Die Syntax für `*SCREEN` ist:

```
*SCREEN [zeile spalte länge]
```

  zeile ist ein Wert zwischen 1 und 25.

  spalte ist ein Wert zwischen 1 und 80.

  länge ist ein Wert zwischen 1 und 80.

- Der lokalen Variablen `#ENVIRONMENT` den Wert des DOS-Umgebungsparameters `PATH` zuweisen:

```
SET #ENVIRONMENT PATH
```

Prozedurdateibeispiel

Findfile.ncp

Zurückgegebene Variablen

Keine

Regeln für `SET PCFILE`

Die mit `SET PCFILE` definierte Namenserweiterung gibt an, welche dynamische Formatkonvertierung angewendet wird.

Format	Namenserweiterung
ASCII	Jede andere nicht in dieser Tabelle aufgeführte Namenserweiterung (z.B. *.*ncd*, *.*ncs*, *.*ncm*, *.*ncr* usw.)
Basic	*.*prn*
Binär	Alle Binärdateien unabhängig von der Erweiterung. Das Übertragungsformat von Natural ist ein Satz mit einem einzigen Binärfeld.
COBOL	*.*ncc*
dBase III	*.*dbf*
Data Interchange Format	*.*dif*
Encryption	*.*enc*
Excel	*.*xls* oder *.*xlsx* (abhängig von der Excel-Version)
HTML	*.*htm* oder *.*html*. Dies ist ein besonderes HTML-Format, das auch mit Excel bearbeitet werden kann.
Lotus	*.*wks*, *.*wk1*, *.*wkl*
XML	*.*xml*

Vom System generierte Dateinamen (nur beim Herunterladen)

Der Dateiname wird automatisch erzeugt, wenn Sie `~~RANDOM` anstelle eines Dateinamens mit der Anweisung `SET PCFILE` angeben. Der automatisch erzeugte Dateiname hat die folgende Form:

```
tthhmmss.xxx
```

tt steht für den aktuellen Tag des Monats, der vom Systemdatum übernommen wird.

hhmmss steht für die aktuelle Uhrzeit, die vom Systemdatum übernommen wird.

xxx steht für eine der folgenden Namenserweiterungen:

- *ncd*, wenn für die Daten keine Namenserweiterung angegeben wurde. Beispiel:

  ```
  SET PCFILE 5 DOWN DATA ~~RANDOM
  ```

- *ncr*, wenn für einen Report keine Namenserweiterung angegeben wurde. Beispiel:

  ```
  SET PCFILE 5 DOWN REPORT ~~RANDOM
  ```

- Die mit `~~RANDOM` angegebene Namenserweiterung. Beispiel:

```
SET PCFILE 5 DOWN DATA ~~RANDOM.XYZ
```

Verwandte Befehle

DECR, INCR, RESET, SHIFT

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja - abhängig vom Variablentyp. Bei Systemvariablen gibt es Einschränkungen, ob und wann SET gilt.
Taste:	Ja
API:	Ja

50 SHIFT

Beschreibung

Verschiebt den Inhalt der globalen oder lokalen Parametervariablen von PARM2 bis *PARM9* nach PARM1 bis PARM8, damit PARM9 der Wert Null zugewiesen werden kann.

Dieser Befehl ist zum Beispiel bei interaktiven Prozedurdateien (oder bei verschachtelten Prozedurdateien) hilfreich, bei denen ein externer Wert an +PARM1 übergeben wird. Wenn der Wert von +PARM1 bei jedem Durchlauf dynamisch geändert werden soll, übergeben Sie Werte für die Variablen +PARM1 bis + PARM9 und führen vor jedem Durchlauf den Befehl SHIFT aus.

Syntax

```
SHIFT {GLOBALS|LOCALS}
```

Beispiele

- Alle Werte der globalen Parameter verschieben:

  ```
  SHIFT GLOBALS
  ```

- Alle Werte der lokalen Parameter verschieben:

  ```
  SHIFT LOCALS
  ```

Prozedurdateibeispiel

Vars.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

51 SLEEP

Beschreibung

Hält die Verarbeitung einer Prozedurdatei für eine bestimmte Zeitspanne an.

Im Gegensatz zum Befehl `PAUSE` wird bei `SLEEP` nicht geprüft, ob Daten vom Host eintreffen.

Syntax

```
SLEEP millisekunden
```

millisekunden kann ein Wert zwischen 0 und 32000 sein, d.h. zwischen 0 Tausendstel einer Sekunde und 32 Sekunden. Der Wert 1000 entspricht 1 Sekunde.

Beispiele

- Verarbeitung einer Prozedurdatei für 1 Sekunde (1000 Millisekunden) anhalten:

  ```
  SLEEP 1000
  ```

- Verarbeitung einer Prozedurdatei für die Anzahl der Sekunden anhalten, die in der Variable `#CNT1` definiert sind:

  ```
  SLEEP #CNT1
  ```

Zurückgegebene Variablen

Keine

Verwandte Befehle

`ELAPSETIME`, `PAUSE`, `WAIT`, `WAITFOR`, `WAITM`, `WAITUNTIL`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

52 TOGGLE

Beschreibung

Schaltet zwischen zwei möglichen Zuständen einer Systemvariablen um.

Dies sind die Systemvariablen, für die nur die Werte `ON`/`OFF`, `YES`/`NO` oder `TRUE`/`FALSE` gesetzt werden können.

Syntax

```
TOGGLE systemvariable
```

Beispiele

- Variable `LOGON` von `ON` auf `OFF` oder von `OFF` auf `ON` umschalten:

  ```
  TOGGLE LOGON
  ```

- Variable `STATUS` von `ON` auf `OFF` oder von `OFF` auf `ON` umschalten:

  ```
  TOGGLE STATUS
  ```

Prozedurdateibeispiel

Vars.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

`RESET`, `SET`

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Ja
Taste:	Ja
API:	Ja

53 TYPE

Beschreibung

Sendet simulierte Tastatureingaben an den Host oder PC.

Entire Connection wartet bei allen Eingaben über Terminal-Funktionstasten (z.B. CLEAR oder CR) auf eine Antwort vom Host.

Entire Connection wartet, bis eine Antwort eintrifft oder bis die in der Systemvariable `RESPONSE` definierte Zeitspanne abgelaufen ist. Wenn innerhalb der definierten Zeitspanne eine Antwort eintrifft, wird die nächste Anweisung in der Prozedurdatei ausgeführt.

Wenn die definierte Zeitspanne überschritten wird, wird die Verarbeitung der Prozedurdatei abgebrochen. Ausnahme: wenn die Sprungmarke `$TIMEOUT` in der Prozedurdatei definiert ist, wird zur weiteren Verarbeitung zu dieser Sprungmarke verzweigt.

Syntax

```
TYPE {zeichenkette|terminal-funktionstastenname|physische-funktionstaste}...
```

Beispiele

- Tastatureingabe `CLEAR` an den Host übermitteln:

```
TYPE CLEAR
```

- Die Zeichenketten `*NAT` und `%+` jeweils mit der Tastatureingabe `CR` an den Host übermitteln:

```
TYPE '*NAT' CR '%+' CR
```

- Die Tastatureingaben `CLEAR` und `ALT+A` an den Host übermitteln (für `ALT+A` kann zum Beispiel `'*NAT' CR` definiert sein):

```
TYPE CLEAR A-A
```

Siehe auch: *Tastenschemata*

Prozedurdateibeispiel

Makete.ncp

Zurückgegebene Variablen

Keine

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Ja
API:	Ja

54 WAIT

Beschreibung

Hält die Verarbeitung einer Prozedurdatei solange an, bis der Benutzer im Dialogfeld eine Befehlsschaltfläche wählt.

Die Meldung (Zeichenkette und Variable) darf bis zu 184 Zeichen lang sein.

Syntax

```
WAIT [{zeichenkette|variable}...]
```

Beispiele

- Verarbeitung ohne Ausgabe einer Meldung anhalten:

```
WAIT
```

- Verarbeitung anhalten, mit Ausgabe der Meldung, dass der in der Variable #FILESPEC definierte Dateiname nicht gefunden wurde:

```
WAIT 'Datei' #FILESPEC 'wurde nicht gefunden'
```

- Verarbeitung anhalten, mit Ausgabe der Meldung, dass eine Prozedurdatei erfolgreich beendet wurde:

```
WAIT 'Prozedurdatei wurde erfolgreich beendet'
```

Prozedurdateibeispiel

Parms.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

PAUSE, SLEEP, WAITM, WAITFOR, WAITUNTIL, ELAPSETIME

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

55 WAITFOR

Beschreibung

Wartet auf eine bestimmte Bedingung auf dem vom Host gesendeten Terminal-Emulationsbildschirm.

Wenn die Bedingung wahr ist, wird der entsprechende Befehl ausgeführt.

Wenn die Bedingung falsch ist oder wenn die in der Systemvariable `RESPONSE` definierte Zeitspanne abgelaufen ist ohne dass ein Bildschirm vom Host empfangen wurde, wird die nächste Anweisung ausgeführt.

Syntax

```
WAITFOR screen-variable operator zeichenkette befehl
```

screen-variable steht für die Variable `*SCREEN` (Syntax siehe unten).

operator ist einer der folgenden Werte:

`EQ`	gleich
`NE`	ungleich
`GE`	größer gleich
`LE`	kleiner gleich
`GT`	größer als
`LT`	kleiner als

Für *befehl* können Sie jeden Befehl von Entire Connection, außer den Folgenden, benutzen:

```
IF
IFNOT
WAITFOR
```

Syntax für *SCREEN

Die Variable *SCREEN kann nur einmal pro WAITFOR benutzt werden.

```
*SCREEN [zeile spalte [länge]]
```

zeile ist ein Wert zwischen 1 und der maximalen Zeilenanzahl + 1.

spalte ist ein Wert zwischen 1 und der maximalen Zeilenlänge.

länge ist ein Wert zwischen 1 und der Bildschirmgröße.

Zum Beispiel:

*SCREEN steht für: gesamter Bildschirm.

*SCREEN 2 1 steht für: ab Zeile 2, Spalte 1 bis zum Ende des Bildschirms.

*SCREEN 2 1 80 steht für: ab Zeile 2, Spalte 1 die nächsten 80 Positionen.

Beispiele

- Das Vorkommen der Zeichenkette CP READ prüfen und, wenn die Bedingung wahr ist, zur Sprungmarke LOGON verzweigen:

  ```
  WAITFOR *SCREEN EQ 'CP READ' GOTO LOGON
  ```

- Das Vorkommen der Zeichenkette NEXT ab Zeile 2, Spalte 1 prüfen und, wenn die Bedingung wahr ist, CR an den Host übermitteln:

  ```
  WAITFOR *SCREEN 2 1 EQ 'NEXT' TYPE CR
  ```

- Den in der lokalen Variable #PARM1 definierten Wert prüfen und, wenn die Bedingung wahr ist, zur Sprungmarke WEITER verzweigen:

  ```
  WAITFOR *SCREEN EQ #PARM1 GOTO WEITER
  ```

Prozedurdateibeispiel

Waitcmds.ncp

Zurückgegebene Variablen

Wenn die Bedingung wahr ist, wird die Bildschirmposition der Zeichenkette in den folgenden lokalen Variablen zurückgegeben:

#ROW - die gültigen Werte liegen zwischen 1 und der maximalen Zeilenanzahl + 1

#COL - die gültigen Werte liegen zwischen 1 und der maximalen Zeilenlänge

Verwandte Befehle

IF, PAUSE, SLEEP, WAIT, WAITM

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

56 WAITM

Beschreibung

Hält die Verarbeitung einer Prozedurdatei für eine bestimmte Zeitspanne an.

Dieser Befehl ist hilfreich bei Prozedurdateien, die im UA-Modus ausgeführt werden sollen. Außerdem kann es nötig sein, dem Benutzer das Weiterarbeiten für eine bestimmte Zeitspanne zu unterbinden. Dazu kann eine erklärende Meldung ausgegeben werden. Die Meldung wird im Ausgabefenster der Terminal-Anwendung angezeigt.

Während des Wartens prüft Entire Connection, ob eine Antwort vom Host vorliegt. Bei einer Antwort, die kein Dateitransfer-Buffer ist, wird das Warten automatisch beendet und die nächste Anweisung in der Prozedurdatei wird ausgeführt. Das ist hilfreich, wenn Sie darauf warten, dass ein Prozess auf dem Host beendet wird, Sie jedoch die Prozedurverarbeitung nicht länger als nötig anhalten wollen.

Die einzige andere Möglichkeit, das Warten zu beenden, ist die Verarbeitung der gesamten Prozedurdatei abzubrechen.

Syntax

```
WAITM minuten [meldung]
```

minuten steht für einen Wert zwischen 1 und 1440. *meldung* darf bis zu 184 Zeichen lang sein.

Beispiele

- Eine Minute warten:

```
WAITM 1
```

- Eine Minute warten und eine Meldung ausgeben:

```
WAITM 1 'Daten noch nicht verfügbar - neuer Versuch in 1 Minute'
```

Prozedurdateibeispiel

Waitcmds.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

PAUSE, SLEEP, WAIT, WAITFOR, WAITUNTIL, ELAPSETIME

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

57 WAITUNTIL

Beschreibung

Hält die Verarbeitung einer Prozedurdatei bis zu einem definierten Zeitpunkt an.

Syntax

```
WAITUNTIL datum zeit
```

datum hat das Format JJJJ/MM/TT. Ab dem Jahr 2000 müssen Sie das Jahr im Format JJJJ angeben. Bis 1999 können Sie das Jahr im Format JJ oder JJJJ angeben.

zeit hat das Format HH:MM:SS.

Beispiele

- Bis zum 21. März 1999 um 23:00 Uhr warten:

  ```
  WAITUNTIL 1999/03/21 23:00:00
  ```

- Bis zu dem Zeitpunkt warten, der durch die dynamische Variable `*DATE` und die lokale Variable `#PARM1` definiert ist:

  ```
  WAITUNTIL *DATE #PARM1
  ```

Prozedurdateibeispiel

Waitcmds.ncp

Zurückgegebene Variablen

Keine

Verwandte Befehle

PAUSE, SLEEP, WAIT, WAITFOR, WAITM, ELAPSETIME

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein

58 WRITE

Beschreibung

Schreibt Daten in eine Ausgabedatei. Die maximale Satzlänge beträgt 255 Bytes.

Die Datei ist für alle geschachtelten Prozedurdateien verfügbar. Sie können zum Beispiel den Befehl `WRITE` aus mehr als einer Prozedurdatei heraus benutzen.

Syntax

```
WRITE dateinummer [{zeichenkette|variable}...] [;]
```

dateinummer ist eine Zahl zwischen 1 und 4.

Wenn Sie keinen Operand angeben, wird ein leerer Datensatz geschrieben (d.h. die Datei enthält nur CR/LF).

Wenn Sie ein Semikolon (;) angeben, wird kein CR/LF in die Datei geschrieben.

Beispiele

- Leerzeile in die Datei 1 schreiben:

```
WRITE 1
```

- Den Inhalt der lokalen Variablen `#PARM1` bis `#PARM4` in die Datei 2 schreiben:

```
WRITE 2 #PARM1 #PARM2 #PARM3 #PARM4
```

- Den Inhalt der lokalen Variablen #PARM1 bis #PARM4 ohne Angabe von CR/LF in die Datei 2 schreiben:

```
WRITE 2 #PARM1 #PARM2 #PARM3 #PARM4 ';'
```

- Eine Zeichenkette und das aktuelle Datum in die Datei 1 schreiben:

```
WRITE 1 'Heute ist' *DATE
```

Prozedurdateibeispiel

Vars.ncp

Zurückgegebene Variablen

#RC (SUCCESS, wenn WRITE erfolgreich war. FAILURE, wenn WRITE nicht erfolgreich war.)

Verwandte Befehle

OPEN-O, CLOSE

Benutzung

Prozedurdatei:	Ja
Befehlszeile:	Nein
Taste:	Nein
API:	Nein