

NCR

Mini VRX

Rel. 11

NCR GmbH
Postfach 10 00 90
8900 Augsburg 1
Telefon 0821/40 51

NCR

Printed in the Federal Republic of Germany

D00-0071875

INHALTSVERZEICHNIS	KAPITEL
Betriebssystem	1
Dienstprogramme	2
Text Editor	3
Programmierung	4
Daten-Management	5
Teleprocessing	6
(Reserve)	7
SDE	8
Tabellen + Codes	9
IVS	10
Stichwortverzeichnis	11

„Diese Unterlagen sind vertrauliches, geistiges Eigentum der NCR GmbH. Eine unbefugte Verwendung, Vervielfältigung oder Weitergabe ist nicht gestattet.“

© Copyright NCR GmbH 1985

Inhaber P. [REDACTED]

Ber. Schulung

August 85
(VRX REL. 11)

1 BETRIEBSSYSTEM VRX

VRX Start	1-20
VRX Abschluss	1-20
Disc-Controller Operationen	1-30
VS Firmware Control Functions	1-40
VRX SW-1 Dump	1-50
VRX Operator Communication: Abfragen	1-60
VRX Operator Communication: Befehle	1-70
VRX JOB CONTROL LANGUAGE	1-100
VRX JSL Parameters	1-110
VRX MCL Parameters	1-130
VRX FCL Parameters	1-150

Ausführliche und massgebende Informationen in:

- RM-0231-V11 V-Series Operator Information
- RM-0232-V11 VRX Operating System

Stand: V11 / 07.85

VRX-START

- FIRMWARE-Discette einlegen
- (NORMAL/ALTERNATE-Schalter auf NORMAL stellen)
- POWER-ON-Taste drücken
- Peripherie- und Telecommunication-Processors starten, falls vorhanden
- wenn die Firmware erfolgreich geladen ist, erscheint auf der Konsole die Meldung:
//V SYSRESET
//V HALT CR=h h h h h h VMID=h P=n TM=n X=n VS n/n.n
//V _
- BO eingeben
- nachfolgende Meldung beantworten

VRX START UP - ENTER TYPE OF INITIALIZATION
(FULL, REINIT, OR EDIT) FOLLOWED BY DAY, DATE,
TIME AND OPTIONAL EBF - (FULL, DAY, DD MMM YYYY,
HH:MM:SS, filename)
DAYS MON, TUE, WED, THU, FRI, SAT, SUN
MONTHS JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG,
SEP, OCT, NOV, DEC
SYSTEM ID - nnn,nn nn,nnn

(bei 86xx: nur 'InitType' + 'filename' angeben)

VRX-ABSCHLUSS

- DEINIT eingeben
- wenn alle Jobs zu Ende gelaufen sind, kommt die Meldung, ob das System endgültig gestoppt oder weiterlaufen soll
- Eingabe START oder DEINIT
- nach WAIT B1B1: an MP-Systemen: ESC-CNTL-I/Discs abstellen
(evtl. Disc-Controller abstellen, siehe nächste Seite)

DISC-CONTROLLER OPERATIONEN

DISC-CONTROLLER-BOOT 625

POWER	ON	Discs nicht starten
HALT	ON	
RESET	drücken	Ausgabe: FFFF
HALT	OFF	
NORMAL/DIAG	NORMAL	
Kassette «Controlprog»	einsetzen	
LOAD	drücken	dauert ca. 2 Minuten Ende: WAIT leuchtet konstant
MEMORY ADDRESS links	drücken	Ausgabe: F00C
Kassette «Log»	einsetzen	
COMPUTE	drücken	ok, wenn ACTIVE-Lampe blinkt
Discs	starten	

DISC-CONTROLLER 625 TAGES-START

POWER	ON	Discs nicht starten
HALT	ON	
Kassette «Log»	einsetzen	
NORMAL/DIAG	NORMAL	
RESET	drücken	
eintippen	1807	
CONTROL ADDRESS rechts	drücken	
HALT	OFF	
COMPUTE	drücken	ok, wenn ACTIVE-Lampe blinkt
Discs	starten	

Falls nicht erfolgreich: Zuerst DISC-CONTROLLER-BOOT ausführen

DISC-CONTROLLER 625 TAGES-ABSCHLUSS

Discs	ausschalten
NORMAL/DIAG	DIAG
warten	bis Kassette zurückgespult ist
HALT	ON
POWER	OFF

VSn FIRMWARE CONTROL FUNCTIONS

(Eingabe ohne ←)

Funktion	Beschreibung	Display
ESC H	HALT v. SYSTEM	//V HALT CR=h h h h h...
ESC C	COMPUTE	//V COMPUTE
ESC R	SYSTEM RESET (nur im Halt)	//V PERESET
ESC CNTL I	FULL SYSTEM-RESET FÜR MP-SYSTEME	//V PERESET
ESC Ø	DISPLAY OPTION SWITCH	//V OS=n n n n n n n n
ESC n	SET OPTION SWITCH n ON	//V OS=n n n n n n n n
ESC SHIFT+n	SET OPTION SWITCH n OFF	//V OS=n n n n n n n n
ESC CNTL L	FIRMWARE RELOAD	//V SYSRESET //V HALT CR=h h h h h h... //V_

To generate the switch-1 printout:

1. Enter
ESC 1

The screen displays:

```
//VWAIT 00DD01 xxxxxx
//VHALT CR=dddddd VMID=x P=n TM=x X=n VS1=n.n
```

where xxxxxx is the hex memory address to enter data for each printout request.

2. Enter
EN xxxxxx type of dump
where xxxxxx is the hex memory address in the WAIT 00DD01 xxxxxx message, aaaaaa is the TOB address,
and nnnnnn is the local area address.
3. Enter
ESC C
The screen will display the same message as above when printing is complete.
4. To terminate printing or exit the switch-1 function, enter
ESC 1
ESC SHIFT
then
ESC C

System Command	Function	Mag. Tape File
EN xxxxxx 10 nnnn ESC C	Dump to mag tape (LPN = nnnn)	
EN xxxxxx 11 ESC C	Stop dump to tape	
EN xxxxxx 01 ESC C	Dump all TDB's and job tables	TDB DUMP
EN xxxxxx 02 00 ssssss eeeee ESC C	Dump global memory	
EN xxxxxx 02 01 ssssss eeeee ESC C	Dump writable global memory	
EN xxxxxx 02 02 ssssss eeeee ESC C	Dump resident global memory	GLOB DUMP
EN xxxxxx 03 aaaaaa 00 ssssss eeeee ESC C	Dump local memory	
EN xxxxxx 03 aaaaaa 01 ssssss eeeee ESC C	Dump writable local memory	
EN xxxxxx 03 aaaaaa 02 ESC C	Dump stack analysis	LOC DUMP
EN xxxxxx 03 aaaaaa 03 ssssss ESC C	Dump local memory to CRT	
EN xxxxxx 04 ESC C	Dump I/O Control Block	IOCB DUMP
EN xxxxxx 24 ESC C	Dump JIF	JIF DUMP

ssssss = optional starting address
 eeeee = optional ending address
 aaaaaa = TDB address

VRX OPERATOR COMMUNICATION: ABFRAGEN

* Spezielle Kontroll-Funktionen:

- .C druckt Konsoleninhalt auf Hardcopy-Printer aus
- .D für Bildschirm-Updating
- .F für Bildschirm-Updating alle 12 sec
- .J nächste Bildschirm-Seite bei mehr als 20 Zeilen
- .K letzte Bildschirm-Seite bei mehr als 20 Zeilen

* Menu-Bilder

- A System-Display-Index (gibt Liste der Menu-Bilder)
 - H IP Host Status-Anzeigen
 - O noch zu beantwortende Konsolen-Meldungen
(.O / .O, I / .O, nnnn)
 - P alle Jobs im System (ACTIVE JOBS HELD JOBS,
PENDING JOBS)
(.P / .P, I / .P, B)
 - Q alle wartenden Jobs (PENDING JOBS), welche vom
System selber aufgerufen werden
 - R alle aktiven Jobs (ACTIVE JOBS)
(.R / .R, I / .R, B)
 - S alle wartenden Jobs (HELD JOBS), welche vom
Operator aufgerufen werden müssen
 - T alle voneinander abhängigen Jobs
 - U Angaben über Drucker-Spoolfiles (Anzahl Seiten etc.)
(.U / .U, y / .U, nnnn)
 - V Peripherie-Zuordnung (evtl. freie Disc-Stationen etc.)
(.V / .V, nnnn)
 - W System-Statistik
 - X Tasks des Systems
(.X / .X, nnnn)
 - Y RJE Links und Units
 - Z Remote Terminals
- zu beantwortende Meldungen (SET UP MESSAGES etc.)
 - Job-Detail-Information (mehrere Seiten)

VRX OPERATOR COMMUNICATION: BEFEHLE

* Format:

nnnn:VERB, Felder ☒

nnnn : = vierstellige Jobnummer
 VERB : = Funktion
 Felder : = evtl. Operanden
 ☒ : = Abschluss der Eingabe
 (←-Taste oder NEWLINE-Taste)

* Befehle zum Ändern der Job-Spezifikationen

nnnn:CPTY, n ☒ } Prozessor-Priorität ändern
 nnnn:IOCT, [T], ±n ☒ } Max. Zeit zwischen I-O's ändern
 nnnn:JPTY, n ☒ } Job-Priorität ändern
 nnnn:MAXT, [T], ±n ☒ } Max. Processor-Zeit ändern
 nnnn:OPTS, hh ☒ } Setzen der OPTION SWITCHES
 nnnn:OPTS, bbbbbb ☒ } für einen speziellen Job
 in Hex oder Binär

* Befehle zum Ändern des Job-Status

nnnn:ABORT ☒ } Job-Abbruch (o. SIMOPTSW!)
 nnnn:ABJ ☒ } (Spoolfile bleibt erhalten)
 nnnn:CKILL ☒ } Karten-Spoolfile löschen
 nnnn:DEBUG [,xxx] ☒ } System Debug Modus
 nnnn:ERRORSTART [,nn] ☒ } JOB STEP abbrechen
 nnnn:AB [,nn] ☒ } evtl. Setzen v. SIMOPTSW
 (vgl. S. 9-50)
 nnnn:HOLD ☒ } Job in Warteschlange einreihen
 nnnn:KILL ☒ } Job-Abbruch (mit Löschen
 des Drucker-Spoolfiles)
 nnnn:NEXT ☒ } Raschmöglichstes Aktivieren
 eines wartenden
 Jobs (JPTY = 0)
 nnnn:PURGE ☒ } Löschen eines beendeten und
 gedruckten Jobs
 nnnn:READY ☒ } Aktivieren eines Jobs aus der
 Warteschlange
 nnnn:RECOVER ☒ } auf Recovery Routine
 verzweigen
 nnnn:UNPOST ☒ } POST-Zähler um 1 reduzieren
 nnnn:UNWAIT, xx... ☒ } Eintragungen eines Jobs im
 Wait-Verzeichnis löschen

* Befehle zur Bearbeitung des Druckfiles

nnnn:DESTN, Filename, xx, y } Zuteilung von Printfile zum RJE-
 Code xx, mit Warteschlange y
 (xx = 00, AA-ZZ/y = 1-7)
 nnnn:PCOPY, Filename, n } Anzahl Kopien eines Druckfiles
 nnnn:PCOPY, Filename, ±n } ändern (auf/um n)
 nnnn:PFREE, Filename } Software löscht Druckfile nach
 dem Drucken (= >PKEEP)
 nnnn:PHOLD, Filename } Druckfile soll erst bei Abruf
 ausgedruckt werden
 nnnn:PKEEP, Filename } Druckfile soll nach dem Drucken
 nicht gelöscht werden
 nnnn:PKILL, Filename } Druckfile soll gelöscht werden,
 das im Moment weder gespooled
 noch gedruckt wird
 nnnn:PLIMIT, Filename, n } Anzahl Seiten pro Druckfile
 nnnn:PLIMIT, Filename, ±n } beschränken (auf/um n)
 nnnn:PMOVE, Filename, { |PRINTER| } , y } Umleiten
 { |MTNORMAL| } eines Druckfiles
 { |MTCOM| }
 nnnn:PNEXT, Filename } druckt ein wartendes oder
 suspendiertes Druckfile
 als nächstes aus
 (= >PHOLD, PSUSP)
 nnnn:PREDY, Filename } macht ein suspendiertes oder
 wartendes Druckfile zum
 Drucken bereit
 PSPAC, Lpn } Wiederholen oder Überspringen
 PSPAC, Lpn, ±n } des gegenwärtig druckenden
 Druckfiles um n Seiten resp.
 von Anfang an
 PSUSP, Lpn } suspendiert das gegenwärtig
 druckende Druckfile (kann
 später wieder mit PNEXT oder
 PREDY weiter gedruckt werden)

* Befehle zum Ändern des Peripherie-Status

ALLOCOK, Lpn } Systemzuordnung einer Unit
 zum System (kann nun von Jobs
 gebraucht werden)
 ATTACH, Lpn } Unit dem Betriebssystem zu-
 ATTACH, Lpn, y } ordnen (y = Nummer der Printer-
 Warteschlange)
 DETACH, Lpn } Systemeinheit vom Betriebs-
 system loslösen
 DOWN, Lpn } Unit defekt
 MARG, Lpn } Unit beschränkt einsatzfähig
 NOALLOC, Lpn } Unit nicht mehr neu zuteilbar
 UP, Lpn } Bring up

* verschiedene weitere Befehle

DEINIT		Befehle zum Anhalten des Systems (System wird angehalten, sobald alle aktiven Jobs abgeschlossen sind und das gerade druckende Druckfile zu Ende ist)
POCTF, { [PRINTER] [MTNORMAL] }, y [MTCOM]		OCT-File drucken
RUNJ, job- name/jobid { .DS=source .DD=del-digits } ... { .V=vvvvvv .D=ddmmyy }		Abrufen eines katalogisierten JOBS von CPL
TIME		Tageszeit displayen
JISTART		JOBS wieder aktivieren (nach JISTOP)
JISTOP		Keine neuen JOBS mehr aktivieren
VDATE, ddmonyy		Ändern Virtual Date

* Befehle für IVS-Software

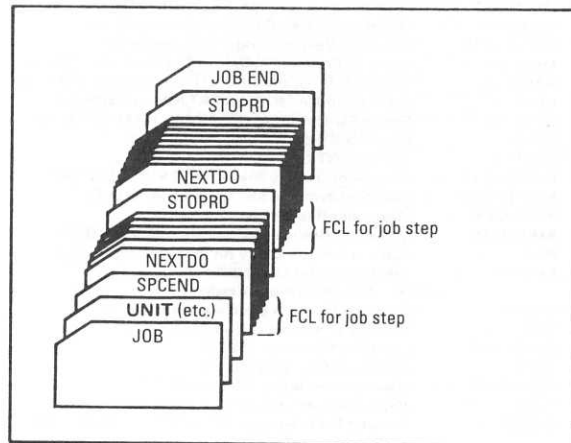
RUNS, IVS		Start von IVS
ROUTE [,nnnn]		Als Meldungs-Beantwortung: Meldung umleiten (nnnn=Session-Nr.)

- Folgende als Beantwortung von 'OPERATOR INPUT PERMITTED':

CEASE		IVS beenden
HELP		Alle IVS-Software Konsolen-Befehle erklären
RESTART		Nach Verbindungs-Unterbruch
SET MAXUSERS TO n		Max. Anzahl IVS-Terminal-Benutzer
SEND TO { nnnn } 'meldung' [ALL]		Meldung an IVS-Terminal(s) senden

JOB CONTROL LANGUAGE

(CC 18)
 JOB
 [UNIT]
 [PRINT]
 [JOBDET]
 [DSPLY]
 [DSPLYP]
 SPCEND
 Master MCL
 NEXTDO
 Detail-MCL (inkl. FCL)
 STOPRD
 JOB END



VRX JOB SPECIFICATION LANGUAGE PARAMETERS

(folgend JSL-Statements, ab CC 24)

JOB-Parameters

ab 24	Bedeutung
ACCESS = x	OLPD Print Spoolfile Zugriff (x = YES/NO)
ACCT x	Accounting-Nummer (1- bis 16stellig, alpha + num.)
CPTY n	CPU Time Priorität (0, 1, 2, ..., 15)
DELt x	Delete Charakter (0-9, A-Z)
DELt x-y	Delete Charakter (0-9, A-Z)
DEST xx/y	Druck-Output in Warteschlange y (xx = 00 oder AA-ZZ/y = 1-7)
HOLD	JOB einlesen und in Warteschlange versorgen
IOCT = n/T]	maximale CPU-Zeit zwischen I-O (1-9999999 ms)
JPTY = n	JOB-Ablauf-Priorität (1-15)
KEEP	JOB im Karten-Spoolfile nicht autom. löschen
LOCALDSA = x	Local dynamic Storage Area (4-4096 KB)
MAXSHRD = x	Maximale Shared-Data Lock-Zeit (1-99999 sec)
MAXT = n/T]	Maximale Verarbeitungszeit (1-99999 sec)
MSGCOM = x	Message Communication für OLPD Jobs (x = YES/NO)
NAME x	Name des Programmierers (1- bis 16stellig, alpha + num.)
NEXT x	nach laufendem JOB den CATJOB x verarbeiten
OPTS hh	Option Switch setzen (00000000 - 11111111, 00 - FF)
PBACK	Warten auf Piggyback-JOB-Parameter
POST n	JOB in POST-Liste aufnehmen (1-22)
*PROGOVLYS = n	Programm Overlay Area (0-9999 512-B.-Sektoren)
PROTECTDSA = n	Protected dynamic Storage Area (4-512 KB)
RMAXMEM = n	Maximales Reelles Memory (in KB)
RMINMEM = n	Minimales Reelles Memory (0 < n ≤ RMAXMEM)
RUNJ x/y	JOB x sofort von Disc y zur Verarbeitung bringen
SECURITY x	Job Security für OLPD-Jobs (X = 1 - 10 alphanum, Zeichen)
SHARE	Common Program Library sharen
*SIZE = n/n	Memory-Größen in KB für Programm
STACKSIZE = n	Local Stack Size (60-252 KB)
SUPP	JOB Detail Print unterdrücken
VMAXMEM = n	Maximales Virtuelles Memory (4-8192 KB)
WAIT x[/T]	JOB soll warten, bis JOB x abgeschlossen ist
WSEST = n	Working Set Estimator (RMINMEM ≤ x < RMAXMEM) in KB

* für B-Serie-JOBs

UNIT-Parameters

ab 24	Bedeutung
EMULATE = 45	NRZI Mode Parameter
LPN = n	Logical Peripheral Number (n = 4 Hex-Zeichen)
NOSETUP	Unterdr. der SETUP-Meldung
PACK = n	Volume Identification
REEL = n	(000000 - 999999)
VOL = n	(SCRATCH o. : NEWPACK)
SHARE = NO	kein Unit Sharing
SUD = xnn	Symbolic Unit Designator
TYPE = n	Peripheral Type Code

PRINT-Parameters

ab 24	Bedeutung
BLOCK = n	Print File Block Größe (1 - 4)
COPY = n	Anzahl Kopien zu drucken (0 - 255)
DEST = xx/y	Zuordnung von Print File zur Drucker-Warteschlange y (xx = 00 oder AA, AB, ... ZZ/y = 1 - 7)
FORMAT = x	Druckformat (x = COM, NORMAL)
KEEP	Print File nach Drucken nicht löschen
HOLD	Print File erst auf Abruf drucken
LIMIT = n/t]	Maximale Anzahl Druckseiten (1 - 999999)
NOHDRS	Kopfseiten von Print File unterdrücken
SUD = P0x	Symbolic Unit Designator (1 - 9)
TAPE = x	Spooled Printfile auf Magnetband (M01 - M99, SYSTEM)
TRAINNO = n	Printer Train Number (1 - 99)
TYPE = x	Printer Type Code (60 - 6B)

JOBDET-Parameters

ab 24	Bedeutung
COPY = n	Anzahl Kopien zu drucken (0 - 255)
DEST = xx/y	Zuordnung von JOBDETFL für RJE-Code x in Warteschlange y (xx = 00, AA-ZZ/y = 1 - 7)
FORMAT = x	Druckformat (x = COM, NORMAL)
HOLD	JOBDETFL erst auf Abruf drucken
KEEP	JOBDETFL nach drucken nicht löschen
TAPE = SYSTEM	JOBDETFL auf M-Band spoolen
TRAINNO = n	Print Train Number (1 - 99)
TYPE = x	Print Type Code (60-6B)

VRX MCL (MONITOR CONTROL LANGUAGE)

MASTER CONTROL STATEMENTS

CC 18	ab 24	Bedeutung
CALNDR	CALNUM=x	Kalender definieren (1-10, default = 0)
DATE	tt/mm/jj/tag, P	Virtuelles Datum definieren
DSPLY	ll, x...	Operator Meldung displayen (keine Antwort nötig)
DSPLYS	ll, x...	Operator Meldung displayen (Meldung mit CONT beantworten)
FINISH		Letztes Statement in CS
FREE	sud	Peripheriegerät freigeben
HEADCS		Kopf-Statement in CS
IF	bedingung	Vergleich in CS
AND IF	bedingung	Vergleich in CS
INTYPE	x	Eingabe von weiteren MCL-Statements über Console (x=I) oder Kartenl. (x=C)
LIBDSK	sud	Definieren einer User Library
MEMORY	RMINMEM=x SWAP OUT TIME=x REALTIME=YES PAGE FREE INT=x	Min. Reelles Memory Page Swap Time (x=3-999) oder NEVER! Thrash-Verhinderung bei Online-Jobs Steuert Paging-Charakteristik (x=0,01-9,99)
NEXTDO	x	Aufruf eines Programms oder CS
NEXTBR	x	Aufruf eines CS
NOLIB		Hebt das Statement LIBDSK auf
REMDSC	sud	Disc Wechsel (nur wenn nicht geshared!)
SCLBGN		Beginn: System Command Language
SITECD	SCNUM=n	Festsetzen der Übersetzungstabellen für SPUR (0-10)
SYSPER	sud, P	System Printer zuordnen
TRAIN	P0n, y	Definiert den Train y für Printer x
UTIL	x	Aufruf von VRX-Utility x
DIAL	sud, sud, P	SUD's von gleichartigen Geräten vertauschen

DETAIL CONTROL STATEMENTS

(zwischen NEXTDO und STOPRD)

CC 18	ab CC 24	Bedeutung
* Für VRX Programme		
FILE	x...	siehe unter FCL
HEXB	00,0,adress,11,x...	Ändern von Hex-Werten von Load Modul
LOAD	REPLICAS=nn RECOVERY=x... SHARE NOSHARE	Parallel Multi Tasking Parallel Multi Tasking Load-Modul Sharing kein L-M Sharing
STOPRD		Logisches Ende eines Job-Steps
* Für B-Serie-Programme		
CHAR	00,0,adress,11,x...	Ändern von Alpha-Zeichen im Memory
EXFILE	x...	Temporäre Änderung in Extremity File Table
FILEP	x...	File-Placement auf Disc
HEX	00,0,adress,11,x...	Ändern von Hex-Werten im Memory
STOPRD		Logisches Ende eines Job-Steps
* VRX Sort Utility		
CODE	x...	Sort/Merge Parameter (vgl Tab 2)
MERGE	x...	Sort/Merge Parameter (vgl Tab 2)
OPTION	x...	Sort/Merge Parameter (vgl Tab 2)
SEQ	x...	Sort/Merge Parameter (vgl Tab 2)
SORT	x...	Sort/Merge Parameter (vgl Tab 2)
SORTWF	x...	Sort/Merge Parameter (vgl Tab 2)
WORKFL	x...	Sort/Merge Parameter (vgl Tab 2)
* VRX Linkage Editor Utility		
EDIT	NAME=... BIND=... etc.	Link Editor Befehle (vgl Tab 4)
* VRX Modul Library Utility		
LIBRY	x...	Library Befehle (vgl Tab 4)

VRX FCL (FILE CONTROL LANGUAGE)

FCL-Statements auf CC 24, folgend auf das MCL-Statement (FILE)

FILE CHARACTERISTIC PARAMETERS

Statement	Bedeutung	Default
ACCESS MODE = SEQUENTIAL RANDOM DYNAMIC	Zugriffsart	SEQUENTIAL
ACQUIRE	Output-File nicht neu aufbauen	-
AGE = x x/y NEW	Datei-Schutz-Angabe für existierendes File (x = 0-255/TTMMJJ/JJTTT) x ≤ FILE-DATE ≤ y	NEW oder heutiges TTMMJJ oder 0
ALT SUD = xnn	Alternate SUD bei File Creation	-
ALT VOL = nnnnnn	Alternate Volume bei File Creation	-
ASSOCIATION = DATA CONTROL ADDRESS MODIFIED BLOCK KEY	definiert File-Art, zu der jetzt weitere Angaben folgen	DATA
BAT od. NOT CAM	nicht CAM-File	-
BLI NO BLI	Block Length Indicator für ein Magnetband-File	NO BLI
BLOCK HEADER = n	Block Header zusätzlich für User in Bytes	0 Bytes
BLK PER SGMT = n	Blöcke pro Segment (n = 2-255)	-
BLOCK SIZE = n n[//CHARACTERS] n/RECORDS	Blockgröße bei CAM-File-Creation (ohne CAM-Header)	-
BUFFERS = n	Anzahl Puffer (1-255)	seq : 2 sonst 1
CAM	File in VRX COBOL Programm ist CAM-File	-

Statement	Bedeutung	Default
CAM HEADER NO CAM HEADER	evtl. CAM Header für OUTPUTFILE weglassen (nicht für INDEXED!)	CAM HEADER
CAT	File-Catalog File	-
CODE SET = STANDARD ASCII EBCDIC EBCDIC-615 A H KATAKANA BCD ANSI NO TRANSLATION	Code zur Daten-Übersetzung	Code von SITE STANDARD CODE
CREATE	I-O-File soll zuerst kreiert werden	-
DATE SCHEME = GEN DATE JULIAN WORK DAY WORK WEEK WORK MONTH SCRATCH RUN ANY	definiert die Art des Datei-Schutzes	ANY (exist. File) RUN (kreiert. File) SCRATCH (BAT File)
FASTLOAD	Spezial-OPEN bei ICAM-Files	NO FASTLOAD
FILE SIZE = n/RECORDS od BLOCKS/ /APPROXIMATE od. EXACTLY	File-Grösse	100/BLOCKS/ EXACTLY
FILE TYPE = n DATA o 0 SOURCE o 1 OBJECT o 2 CONTROL STRING o 6	File Typ	DATA
FRAGMENTS NO FRAGMENTS	BAT-Multi-Section-File erlaubt	NO FRAGMENTS
FULL SHARE	B-File-Sharing	File nicht geshared
INIT TIME = OPEN NEVER	TOICAM-Block-Vorformatierung	Keine Vorformatierung
.O WAIT LIMIT	Zeitlimite bei Block-Lock	-
KEEP ALL SXNS	Multi-Section-Files in NEATVS	-

Statement	Bedeutung	Default
KEY OFFSET = n	definiert die relative Position des Schlüssels in einem Record (nur bei Erstellen eines CAM-INDEXED-File)	0
KEY SIZE = n	Länge des Schlüssels (1-255)	Wert aus Programm
LABEL = OMITTED STANDARD	Label Record für Tapes	STANDARD
MESSAGE = 'x...' NO MESSAGE	Meldung wird jedesmal beim Eröffnen des Files dem Operator gemeldet	NO MESSAGE
NAIL BUFFERS	Puffer für CAM-File memory-resident	-
NAME = 'x...' (für NEATVS) x... (für VRXCIBOL)	Name des Files (Phys. Name)	Angabe bei REFERENCE
NOCAT	File ist nicht im File-Catalog	-
NOSECURE	Kein Schutz durch File-Access-Control	NOSECURE
NOT CAM	siehe unter BAT	-
ORGANIZATION = SERIAL CHAINED SEQUENTIAL RELATIVE INDEXED (/TIME-ORDERED)	File Organisation (für BAT-Files nur CHAINED oder SERIAL)	SEQUENTIAL oder SERIAL
PAGE TRAILER = n	Anzahl Zeilen pro Seite (0-255)	(66)
POSITION = n	relative Position eines Files auf einem Multi-File-Tape	0
QUICK CLOSE	schneller File-Abschluss für Magnet-Band-Input-Files	-
RECORD SIZE = n n/n	Record-Länge in Bytes (bei var. Länge n n: min. und max. Länge)	Wert aus Programm

Statement	Bedeutung	Default
REFERENCE = x...	Log. File-Name (muss als erster Parameter vorhanden sein!)	OBLIGATORISCH
RETENTION = x...	Datei-Schutzangabe, wie lange ein File geschützt sein soll (0-99 bei GEN, TTMMJJ bei DATE, JJTTT bei JULIAN)	bei Datums-Schutz: heutiges virt. Datum, sonst 0
SECURE	File-Schutz durch File-Access-Control	NOSECURE
SET ID = x...	Identifikation für Files auf mehreren Tapes	Volume Header Nummer des Tapes
SGMT PER SXN n	Anzahl Segmente in einem CAM-Indexed-File pro Section	1
SHARE NO SHARE	Sharing von CAM-Files	SHARE
SHARE-ID = JOB	File Sharing innerhalb Job	-
SPECIAL FORMS = n	Spezielles Druckformular (n = 1-6stellige Nummer)	-
STATISTICS = OPERATIONS NONE	I-O-Statistik für TOICAM-File	Keine Statistik
TAPE MAPPING SINGLE FILE REEL MULTI-REEL FILE MULTI-FILE REEL	Tape File Relation	SINGLE FILE REEL
TEST PATTERN	2 Seiten Testdruck erwünscht	-
THRESHOLD	Meldung bei wenig Reserve in ICAM oder TOICAM-File	-
USAGE INPUT OUTPUT I-O EXTEND NON-KEYED INPUT	Art der File-Bearbeitung	-
VERSION = nn NEW	Versions-Nummer eines Source- oder Objekt-CAM-Programm-Files	NEW

Statement	Bedeutung	Default
VLI=BINARY[/n] DECIMAL[/n]	Typ und Länge des VLI	BINARY/2
WRITEBACK=IMMEDIATE DELAYED	Record-Write bei Non- Shared TOICAM-File	

SECTION CHARACTERISTIC PARAMETERS

Statement	Bedeutung	Default
ALLOC BOUNDS=x/y x/y/u/v	Speicherplatz auf Disc x=Start-Sector, y=End-Sector, u/v=EDA-Module von-bis	-
BLKFREESPACE=n/BYTES n/%	definiert Reserve im Block eines CAM-Indexed-Files	ca. 50%
DUP KEYS=YES NO n/%	Doppelte Schlüssel bei TOICAM Alternate Keys	Aus Programm
FREE-UP NO FREE-UP	nicht belegte Sec- toren freigeben f. seq. Files	NO FREE-UP
HIGH KEY=HEX/hh... CHAR/x...	definiert max. Schlus- selwert pro Section	HEX/FFFFFF..FF
KEY ID=PRIME ALT/n	Folgende Angaben für entspr. Key	PRIME
LOGICAL BLK=n BUCKET SIZE=n SEGMENT SIZE=n	Größen für BAT-Files	-
ON DISK INDEX=n/LEVELS	Anzahl TOICAM- Key-Section Levels auf Disc verbleibend	Alle ausser Hauptlevel
OVERFLOW=n/BLOCKS	Überlaufsectoren für BAT-Kettenfiles	-
SECTION=n MODEL oder Ø	File Sections, die zu bearbeiten ist (Ø für alle Section gültig)	alle Sections
SECTION SIZE=n/RECORDS od. BLOCKS/ APPROXIMATE od. EXACTLY	Section Grösse in Records oder Blocks	100/BLOCKS/ EXACTLY
SEGFREESPACE=n/BLOCKS n/%	Reserve am Schluss jedes Segmentes	ca 50%
SXN BLOCK SIZE=n/CHARACTERS /RECORDS	Block-Grösse pro Section (nicht möglich bei ICAM-Files)	-
SUD=xnn ANY	SUD des Files	-
UNIT TYPE=x/FIXED x/REMOVABLE DISC	Disc Typ zum Erstellen eines neuen CAM-Files	DISC
VOL=nnnnn ANY	Volume Serial Nummer	

2 DIENSTPROGRAMME VRX

Jobs katalogisieren	2-20
Datenträger Directory ausdrucken	2-30
Ganze Datenträger (Teile) kopieren	2-50
Einzelne Datei(en) kopieren	2-90
Ganze Datenträger löschen	2-110
Einzelne Datei(en) löschen	2-120
Datenträger (-Teilbereich) ausdrucken	2-130
Einzelne Datei (-Teile) ausdrucken	2-150
Datenträger (re-)initialisieren	2-180
Tape Spooling	2-220
Kalender Unterhalt	2-230
Sort/Merge	2-240
Datenträgervergleich	2-270
Disc-Sektoren-Inhalt ändern	2-280
SPUR	2-290
File-/Pack locken/entlocken	2-300
Disc-File-Angaben modifizieren	2-310
Waitcodes drucken	2-320

Ausführliche und massgebende Informationen in:

-RM-0235-V11 VRX Utility Routines
(Die TAB und PUBN ist bei jedem Utility angegeben)

DISC (-TEIL) AUF BAND KOPIEREN

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80		
	Jobnamen	JOB	UNIT	SUD=Dnn,VOL=nnnnnn,SHARE=NO,		TYPE=nn		SHARE=NO obl. bei QBACKUP																																																																									
		UNIT	SUD=Mnn,VOL=nnnnnn,LPN=nnnn																																																																														
		UNIT	SUD=Mww,VOL=nnnnnn,LPN=nnnn																																																																														
		SPCEND																																																																															
		NEXTDOBACKUP																																																																															
	SPECF	Q5758	Dnn	Mnn	aaaaaaaaaa															Mww																																																										} Spiegelbildkopie (Data oriented/5)			
		NEXTDOODTSCMGT																																																																															
	SPECF	ODMTOD	Dnn	Mnn	aaaaaaaaaa															Mww	mm	y	b																																																								} Alle Files kop. (Data oriented/3)		
		NEXTDOODTSCMGT																																																																															
	SPECF	ODMTOD	Dnn	Mnn	aaaaaaaaaa															Snnnnn	Ennnnn	Mww	mm	y	b																																																								} Disc-Teil kop. (Data oriented/3)
		STOPRD																																																																															
	JOB	END																																																																															

Mww : MT-Wechsel-Sud (fak.)
mm : Disc-EDA-Modul (fak.)
aaaaaaaaa : Filename auf MT
Snnnnn : Disc-Start-Sektor
b : N=nur File-Typen 1,2,6,7,
kopieren (k/Datenfile)
Ennnnn : Disc-End-Sektor

DATENTRÄGER (-TEILE) VERGLEICHEN

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
	Jobnamen	JOB	UNIT	SUD=xnn,VOL=nnnnnn																																																																											
		UNIT	SUD=xnn,VOL=nnnnnn																																																																												
		SPCEND																																																																													
		NEXTDOGENLCOMP																																																																													
	SPECF	CPYALT																																																																									} Bänder vergleichen, nach CPYALT (B-Util-Routines)				
		NEXTDOGENLCOMP																																																																													
	SPECF	GENCOM	xnnnnn	aaaaaa	bbbbbb	cccc	d	e	ms	md																																																																					
	SPECF	GENCOMSERIES	aaaaaa	bbbbbb	cccc	d	e	ms	md																																																													} Tapes oder Discs (-Teile) vergleichen (B-Util-Routines)									
		STOPRD																																																																													
	JOB	END																																																																													

aaaaaa : Start-Block auf 1. Sud
bbbbbb : Start-Block auf 2. Sud
cccc : Anzahl zu vergl. Blocks
d : N=volle Blocks vergleichen
Y=Vergleich hört bei '5E' auf
e : Y=Tape in File-Mode
N=Tape in Non-File-Mode
Space=Disc
ms : Modul, 1. Disc-Sud (fak.)
md : Modul, 2. Disc-Sud (fak.)
SERIES : Weiter auf denselben Suds (ab der 2. Zeile)

DISC-SEKTOREN-INHALT ÄNDERN

jobnamen	JOB		
	UNIT	SUD=Dnn,VOL=nnnnnn,SHARE=NO	[TYPE=nn]
	SPCEND		
SPEC\$mm	NEXTDD	ODISCTL	
	TRKFA	0nn,aaaaa,bbbb,c,1,dd,eee,mm	
	STOPPR		
	JOB	END	

(I + Sys Main/14)

aaaaa : Sektor-Nummer, dez.
 bbbb : Anfangspos. im Sektor/dez.
 c : Längencode:
 2=1 Bytes
 4=2 Bytes
 etc.
 E=7 Bytes
 0=8 Bytes
 1 : Fixwert 'eins'
 dd... : Alter Inhalt, hex
 ee... : Neuer Inhalt, hex
 m : Disc-Modul

DISC-FILE(S) KOPIEREN

jobnamen	JOB		
	UNIT	SUD=Dss,VOL=nnnnnn	[TYPE=nn]
	UNIT	SUD=Ddd,VOL=nnnnnn	[TYPE=nn]
	SPCEND		
SPEC\$mm	NEXTDD	ODISCOPY	
	NAME	DssDdd,a...	
SPEC\$mm	NEXTDD	ODISCOPY	
	PLACE	bbbbbbbbbb,Dss,Ddd,c	
SPEC\$mm	NEXTDD	ODISCOPY	
	ODISC	DssDdd,dddddd,A,bbbbbbbbbb,c,fffff,ggg,hhh	
	STOPPR		
	JOB	END	

File(s) mit best. Namen(-Teil) kop. (Data oriented/2+ SWISS-SOFT)
 File kopieren (Data oriented/2+ SWISS SOFT)
 File kopieren (Data oriented/2+ SWISS-SOFT)

mm : Modul-Nummer
 a... : File (-Teil) -Namen
 bbbbbbbbbbb : File-Namen
 c : File-Type (bei PLACE: Def.=2)
 dddd : Disc-Pack-Nummer
 ffff : Erstellungsdatum
 ggg : Generations-Nr.
 hhh : Sections-Nr.

Bei PLACE und QDISC: ab CC67 Angabe von Start- und Endsektor (+1) im Format 'sssss,eeee' möglich.

DISC (RE-)INITIALISIEREN: 6560/6580 an DC-625 oder PA

jobnamen		JOB			
	UNIT	SUD=D01,VOL=nnnnnn,SHARE=NO,LPN=nnnn			
	SPCEND				
	NEXTDOUTILCNTL				
SPEC\$	DINIT	D01,aaaaaa,9000,0	b,cc,d		gg
SPEC\$	DINIT	Heee			
SPEC\$	DINIT	Heee			
SPEC\$	DINIT	SDffff			
	NEXTDOUTILCNTL				
SPEC\$	DINIT	D01,aaaaaa,9000,0	3	b	d
SPEC\$	DINIT	Heee... (fak.)			
SPEC\$	DINIT	Heee... (fak.)			
	STOPRD				
	JOB	END			

VOL=:NEWPACK bei Initialisierung

Disc initialisieren (Init + S M/3)

Disc reinitialisieren (Init + S M/3)

- aaaaaa : Disc-Pack-Nummer
- b : Disc-Modell 6=6560 / 8=6580
- cc : Sektoren/Spur: 08,09,20 (6560: k/Eintrag)
- d : Bei 6580: 'D'
- eee... : Benützer-Identifikation max. 50stellig (CC24=H!)
- ffff : DD-Size: 6560=0030/6580=0100
- gg : V8=Sector-Verification (b/6580)

DISC (RE-)INITIALISIEREN: 6530/6531/6540

jobnamen		JOB			
	UNIT	SUD=D01,VOL=nnnnnn,SHARE=NO,LPN=nnnn			
	SPCEND				
	NEXTDOUTILCNTL				
SPEC\$	DIN65	D01,aaaaaa,9000			bb
SPEC\$	DIN65	Hccc			
SPEC\$	DIN65	Hccc			
SPEC\$	DIN65	SDddd			
	NEXTDOUTILCNTL				
SPEC\$	DIN65	D01,aaaaaa,9000	3		
SPEC\$	DIN65	Hccc... (fak.)			
SPEC\$	DIN65	Hccc... (fak.)			
	STOPRD				
	JOB	END			

VOL=:NEWPACK bei Initialisierung

Disc initialisieren (Init + S M/5)

Disc reinitialisieren (Init + S M/5)

- aaaaaa : Disc-Pack-Nummer
- bb : SV=Sector-Verification (6540: ca. 2 1/2 Std.)
- ccc... : Benützer-Identifikation, max 50stellig (CC24=H!)
- dddd : DD-Grösse je Modul

Parameter	Value
JOH	JOHN
LOC	LOCAL
MINMEM	256
LOCALDSA	256
UNIT	Dnn, VOl=nnnnnn, TYPE=nn
SUD	Dnn, VOl=nnnnnn, TYPE=nn
UNIT	SUD=Dnn, VOl=nnnnnn, TYPE=nn
SUD	Dnn, VOl=nnnnnn, TYPE=nn
SECTEN	
NEXTDSORTVRX	
FILE	REFERENCE=SORTINA, NAME=...
SUD	Dnn, BLOCKSIZE=...
FILE	REFERENCE=SORTOUT, NAME=...
SUD	Dnn, BLOCKSIZE=...
RETENTION	
SORT	RECORD=... etc.
MERGE	RECORD=... etc.
ORDER	
MORAP	FILESIZE=... etc.
SECTEN	SET=... etc.
STOPRD	
JOH	END

(SORTVRX:
(Data oriented/21)

Inputfile

(a=Inputfile-Nr./

1 bis 16)

Outputfile (2. Output-

file= SORTOUTDUP

Sortenweisung

Mergeanweisung

Workfile-Angaben

Sort-Sequenz-Ang.

Options

Mögliche Parameter: vgl. nächste Seiten

Parameters bei MCL-Befehl SORT bzw. MERGE

KEYn=lll/ppppp/s/tt

n Key-Nummer (1-20)
lll Key-Länge in Bytes
ppppp Rel. Pos. des Keys
s Sequenz (A oder D)
tt Key-Format
(Bl, PT, ST, ZT, AT...)

FORMAT=xx

Angabe des Key-Formates als
Default-Wert für Key-Parameter

RECORD=F/nnnnn

Länge des fixen Records

RECORD=V/xxxxx/yyyyy/zzzzz

Länge eines var. Records
xxxxx Max. Länge (excl. VLI)
yyyyy (fak.) Durchschn. Länge
zzzzz (fak.) Minimal Länge

Parameters bei MCL-Befehl WORKFL

(fak. bei MERGE)

FILESIZE=nnnnn

Grösse jedes Work-Files in Sektoren

SECTORSIZE=nnnnn

Sektorgrösse des/der

Work-File-Packs

SUD=Dnn/Dnn...

Angabe von Workfile-Suds (max. 16)

Parameters bei MCL-Befehl SEQ (fak.)

SET=xx/xx-xx/*xx/*xx-xx xx

Hex-Wert

* Hex-Wert(e) mit gleicher

Wertigkeit wie der letztdefinierte

DISC-SEKTOREN-INHALT ÄNDERN

.....	jobnamen	JOB
.....		UNIT	SUP=Dnn,VOL=nnnnnn,SHARE=NO [TYPE=nn]
.....		SPCEND
.....		NEXTDC	UTLLCNL
.....	SPEC\$	TRKFX	Dnn,aaaaa,bbbb,c,d,dd,ee,ff,gg
.....		STOPRD
.....		JOB	END

} (I + Sys Main/14)

- aaaa : Sektor-Nummer, dez.
- bbbb : Anfangspos. im Sektor/dez.
- c : Längencode:
 2=1 Bytes
 4=2 Bytes
 etc.
 E=7 Bytes
 0=8 Bytes
- l : Fixwert 'eins'
- dd... : Alter Inhalt, hex
- ee... : Neuer Inhalt, hex
- m : Disc-Modul

SPUR: PROG.-/CS-LOCHKARTEN AUF DISC

.....	jobnamen	JOB
.....		UNIT	SUP=D01/D02,VOL=nnnnnn [TYPE=nn]
.....		SPCEND
.....		NEXTDC	SPUR
.....		STOPRD
.....	000000E	
.....		
.....	END\$	
.....		JOB	END

} SPUR
(Prog. D.T/2)

a : 1=File ausdrucken

3 TEXT EDITOR

NCR VRX OLPD: System Commands	3-20
NCR VRX OLPD: Editing Commands	3-30
NCR VRX OLPD: Spoolfile Access Commands	3-60
NCR VRX OLPD: Message Communication Commands	3-60

Ausführliche und massgebende Informationen in:

–RM-0235 VRX Utility Routines/Program Devel. Tools/Pub 11+ 12

Stand: V08 / 11.82.

NCR VRX OLPD: SYSTEM COMMANDS

ACCESS,jobname[,jobid,rjcode],[filename],[security]

CREATE,filename,COBOL4,idcol,R

CREATE,filename,NEAT/3,,R

DELETE,sud,filenamevv,gen,DATA

DELETE,sud,filenamevv,gen,OBJT

DELETE,sud,filenamevv,gen,SRCE

DELETE,sud,filenamevv,gen

EDIT,filename,sud,idcol,R

EDIT,filenamevv,sud,,R

FINISH

JSTATUS

JSTATUS,jobnumber

JSTATUS,jobname/jobid

JSTATUS,jobnamekey,N

JSTATUS,programmerkey,P

JSTATUS,OLPD

LIST,sud

LIST,sud,filenamevv

LIST,sud,keycharacters,KEY

MAKE,B,c

MAKE,T,c

MAKE,V,c

MSGCOM,jobname[,jobid] [.RJE],security

ON,name

OPERATOR

OPERATOR,message,T

OPERATOR,message

OPERATOR,message,*

PASSWORD,WORD,password

PASSWORD,KEY,passkey

RECOVER,filename

RECOVER

SPACE,sud

TIME

UNITS

VIEW,B

VIEW,V

VIEW,T

XECUTE,jobname,jobid,deletedigits,sud

NCR VRX OLPD: EDITING COMMANDS

ADD

ADD,sourceline

ADD,sourceline,D

ADD,sourceline,R

COPY,sud,filenamevv

COPY,sud,filenamevv,seqno1,seqno2

COPY,sud,filenamevv,seqno1

COPY,sud,filenamevv,,seqno2

DOWN

DOWN,x

DOWN,,y

DOWN,x,y

DOWN,,*

DOWN,x,*

EXIT,KEEP,sud

EXIT,KEEP,sud,0

EXIT,UPDATE,sud

EXIT,UPDATE,sud,0

EXIT,SAVE

EXIT,FORGET

FIRST

FIRST,x

FIRST,*

GAUGE

HOOK,systemcommand

ID

ID,idcol

JOIN,seqno1

JOIN,seqno1,seqno2

LAST

LOCATE,/string/

LOCATE,/string/,x

LOCATE,/string/,*

NCR VRX OLPD: EDITING COMMANDS (Forts.)

MODIFY,/string1//
 MODIFY,string1/string2/
 MODIFY,/string1/string2/x,y
 MODIFY,/string1/string2/*
 MODIFY,/string1//z/D
 MODIFY,/string1/string2//z/D
 MODIFY,/string1/string2/x,y/z/D
 MODIFY,/string1/string2/*z/D
 MODIFY,/string1//*/D
 MODIFY,/string1/string2//*/D
 MODIFY,/string1/string2/x,y/*z/D
 MODIFY,/string1/string2/*z/D
 NOPRINT,P,+
 NOPRINT,P,-
 NOPRINT,I,+
 NOPRINT,I,-
 OMIT
 OMIT,seqno1
 OMIT,*x
 OMIT, { seqno1 } , { seqno2 }
 *x *y
 OMIT,,seqno2
 OMIT,,*y
 PAGELINE,seqno
 PAGELINE,*x
 PAGELINE,seqno,y
 PAGELINE,*x,y
 PAGELINE,seqno,*
 PAGELINE,*x,*

NCR VRX OLPD: EDITING COMMANDS (Forts.)

QUERY
 QUERY,x
 QUERY,*
 REFERENCE,section-name [$\begin{smallmatrix} U \\ D \end{smallmatrix}$]
 REFERENCE,paragraph-name [$\begin{smallmatrix} U \\ D \end{smallmatrix}$]
 REFERENCE,section-name, [$\begin{smallmatrix} U \\ D \end{smallmatrix}$],x
 REFERENCE,paragraph-name., [$\begin{smallmatrix} U \\ D \end{smallmatrix}$],x
 REFERENCE,section-name, [$\begin{smallmatrix} U \\ D \end{smallmatrix}$],*
 REFERENCE,paragraph-name., [$\begin{smallmatrix} U \\ D \end{smallmatrix}$],*
 SEQUENCE,seqno1,seqno2,x
 SEQUENCE,seqno1,seqno2,x,seqno3
 SEQUENCE,seqno1,*x
 SEQUENCE,seqno1,*x,seqno3
 TABS
 TABS,x,y
 UP
 UP,x
 UP,,y
 UP,x,y
 UP,,*
 UP,x,*
 VERSION,x
 XOMIT,seqno1
 XOMIT,*x
 XOMIT, { seqno1 } , { seqno2 }
 *x *y
 ZAPP,x,/string/,n

NCR VRX OLPD: SPOOLFILE ACCESS COMMANDS

DOWN,n[L oder P]

END

FIRST

HOOK, systemcommand

LAST

LIST

LOCATE,/string/

PRINT(copies),[FREE oder KEEP]

PRINT,[KILL oder STATUS]

UP,n[L oder P]

WINDOW, [* oder n], leftposition – rightposition

NCR VRX OLPD: MESSAGE COMMUNICATION COMMANDS

HOOK

MSG

MSG,*

:response

SEND

HOLD

END

4 PROGRAMMIERUNG

VRXCOBOL : Compilation	4-20
LINKEDIT : Load Modul erstellen	4-50
VRXCOBOL : Testhilfe COBUG	4-70
VRXCOBOL : Tuninghilfe: CARTUNE	4-90

VRX : Source Program Library	4-110
VRX : Object Program Library	4-150
VRXCOBOL : Identification Division	4-180
VRXCOBOL : Environment Division	4-190
VRXCOBOL : Data Division	4-230
VRXCOBOL : Procedure Division	4-280
VRXCOBOL : Reserved Words	4-420
VRXCOBOL : File Status Werte	4-440
VRXCOBOL : Sort-Job-Angaben	4-450

Ausführliche und massgebende Informationen in:

– NCR VRX COBOL USER MANUAL

Stand: V11 / 0Z85

VRXCIBOL COMPILATION

Kolonne	Eintrag	Beschreibung
51	C	Löschen aller Monitor-Flags
	☐	Nicht löschen der Monitor-Flags
55	1	Memory Clear
	☐	Kein Memory Clear
61	E	Druck: Nur Error Meldungen
	W	Druck: Error und Warning Meldungen
	M	Druck: Error, Warning und Minor Meldungen
	A	Druck: Error, Warning, Minor und Non-Ansi Meldungen
	☐	Druck: Error, Warning und Minor Meldungen
62	C	Lochkarten-Eingabe
	I	Disc-Eingabe (80-Byte)
	M	Magnetband-Eingabe (80-Byte)
	S	Spur-File-Eingabe
	☐	Lochkarten-Eingabe / Ausnahme: SPUR-Datei-Eingabe, wenn in erster Karte in Kol. 7 ein P und in Kol. 37 ein S
63	X	Benutzte und nicht benutzte Daten- und Prozedur-Namen separat ausgedruckt
	Y	Benutzte und nicht benutzte Daten- und Prozedur-Namen miteinander ausgedruckt
	☐	Kein Cross-Reference Ausdruck
64	F/☐	Compilation mit Erstellung eines Object-Moduls
	S	Compilation ohne Erstellen eines Object-Moduls (Nur Compiler Listing)
65	A	Apostroph (') ersetzt Anführungszeichen (") in QUOTE-Klausel
	☐	QUOTE-Klausel = Anführungszeichen (*)
66	S/☐	Einfache Zeilenschaltung bei Listing
	D	Doppelte Zeilenschaltung bei Listing
67	Y/☐	Generierung der COBUG-Tabelle
	N	Kein Generieren der COBUG-Tabelle
68	Y	Mit WITH DEBUGGING MODE Funktion
	N	Ohne WITH DEBUGGING MODE Funktion
	☐	WITH DEBUGGING MODE Angabe wird aus der Environment Division genommen
69	Y	Druck: Object Code Map und Object Code Listing
	M	Druck: Object Code Map
	N/☐	Kein Druck von Object Code Map oder - Listing
70	Y	Generierung von Spezial-Instruktionen
	N/☐	Kein Generieren von Spezial-Instruktionen
71	Y	CANCEL Befehl kann gebraucht werden
	N/☐	CANCEL Befehl kann nicht gebraucht werden
72	☐	Ohne CARTUNE
	Y	Mit CARTUNE
73	Y	Num. Felder: ☐→-ß, ill. Vz→+
	N/☐	Kein Umsetzen
74	Y	Gen. von Zusatz-Spezial-Instruktionen
	N/☐	Kein Gen. von Zusatz-Spezial-Instruktionen

LINKAGE EDITOR: LOAD MODUL ERSTELLEN

JOB		LOCALLDJA=256
jobnamen	UNIT	SUB=Dnn,VOL=nnnnnn,TYPE=nl
	SPCENE	
	NEXTDLINKEDIT	
	FILE	REFERENCE=OUTPUTFILE,BLOCKSIZE=nnnnn
	EDIT	NAME=loadmod199/Dnn
		BIND=ob1lmod100/Dnn
		BIND=ob12mod100/Dnn
		etc.
	STOPRE	
	JOB	END

LINKAGE EDITOR: Parameters nach 'EDIT'

NAME=loadmodlvv/Dnn	Name/Ort des entstehenden Load-Moduls
BIND=objmodlvv [/Dnn] [/libnamen]	Name/Ort des zu bindenden Object-Moduls
AUTOLIB=Dnn/libnamen	Name/Ort der Object-Modul-Library
LOC=Dnn [/libnamen]	Ort/Library der nachfolgend stehenden BIND-Anweisungen
ORDER=modnamevv/sectname	Festlegen der Reihenfolge der Object-Modul-Section
AUGMAP=modnamevv/sectname/n	Zuordnen einer neuen Basis-Adresse
LOADENT=modnamevv/sectname/ep1/ep2	Definition von Eintrittspunkten (bei NEATVS: obligatorisch)
NOTLOADENT=modnamevv/sectname/ep1/ep2	Nichtig-Erklären eines Eintrittspunktes
XREF	Ausdruck des Alpha Cross Reference Reports
EO	Ausdruck des Extended Object Listings
REPLICAS	Replizieren der folgenden Object-Module erlaubt
NOREPLICAS	Nicht-Replizieren der folgenden Object-Module
ENDREP	Nicht-Replizieren der folgenden Object-Module
SHARE	Load-Modul wird bei Ausführung re-entrant gebraucht
ATTRSORT	Object-Modul-Sections sortieren

VRXCOBOL-TESTHILFE: COBUG

- VRXCOBOL-Compilation: MF17 auf Y oder ☐
- LINKEDIT: Modul 'COBUG' dazubinden:
"EDIT BIND=COBUG/D90"
- Testjob:
 - VMAXMEM mind. 256
 - Programmaufruf mit MF06=C(=COBUG)
 - Unmittelbar nach STOPRD müssen die COBUG-Parameterzeilen folgen.
sofern ein Parameter > 1 Zeile: %-Zeichen am Ende der fortzusetzenden Zeile
 - Abschluss der COBUG-Parameters mit einer END\$-Zeile (anschl. evtl. Daten-zeilen)

COBUG: Parameters

```
DATA data-name-1 [IN data-name-2] ... [OF program-name]
```

```
DUMP {ALL  
{identifier-1} ... }
```

```
AT {ALL  
paragraph-name [IN section-name]}
```

```
[OF program-name]
```

```
[ON x TIME] [FOR y TIMES]
```

```
[ WHEN {identifier-2} {literal-1} {
  IS [NOT] GREATER THAN
  IS [NOT] LESS THAN
  IS [NOT] EQUAL TO
  IS [NOT] >
  IS [NOT] <
  IS [NOT] =
  IS >=
  IS <=
  IS /= } {identifier-3}
  {literal-2} ]
```

FLOW ALL

FLOW paragraph-name-1 [IN section-name-1]
 [TO paragraph-name-2 [IN section-name-2]]
 [OF program-name]
 [ON x TIME] [FOR y TIMES]

PROGRAM program-name

SECTION section-name [OF program-name]

TRACE ALL

TRACE paragraph-name-1 [IN section-name-1]
 [TO paragraph-name-2 [IN section-name-2]]
 [OF program-name]
 [ON x TIME] [FOR y TIMES]

VRXCOBOL-TUNINGHILFE: CARTUNE

- VRXCOBOL-Compilation: MF22 auf Y
- LINKEDIT: Modul 'CARTUNE' dazubinden:
 "EDIT BIND=CARTUNE/D90"
- Testjob: – VMAXMEM überprüfen
 - Programmaufruf mit MF09=C
 - CARTUNE verlangt 2 FILE-Zeilen:
 - FILE REFERENCE=CARTIN, SUD=P11
 CODESET=A, RECORDSIZE=80/CHARACTERS
 - FILE REFERENCE=CARTOUT, SUD=P0n
 RECORDSIZE=132/CHARACTERS
 BLOCKSIZE=1/RECORDS
 - Unmittelbar nach STOPRD (bzw. nach evtl. COBUG-Parameters) können die CARTUNE-Parameterzeilen folgen (wenn keine stehen, werden alle Reports ausgegeben)
 - Abschluss der CARTUNE-Parameters mit einer END\$-Zeile (anschl. evtl. Daten-Zeilen)

Source-Programm-Library: Anweisungen

(CCØ7="+"")

- Neuhinzufügen eines Library-Textes:
+LIB ADD text-name
(anschliessend Source-Zeilen oder eine READ-Anweisung)
- Reorganisation einer Library:
+LIB COMPRESS
(MCL: "REFERENCE=LIBRARY-2 etc." muss vorhanden sein)
- Erstellung der Library:
+LIB CREATE
- Inaktivieren eines Library-Textes:
+LIB DELETE text-name
- Library-Inhaltsverzeichnis ausdrucken:
+LIB DIRECTORY
- Inhalt eines Library-Textes ausdrucken:
+LIB LIST text-name
- Inhalt aller Library-Texte der Library ausdrucken:
+LIB LISTALL

- Identifizierung der Library:
+LIB NAME libr-namen
(1. Library-Befehl/libr-namen hier='xxxxxxxxxxx'
bei FCL: NAME=)
- Einlesen eines 80-Bytes-Record Files:
+LIB READ CARD-IMAGE-INPUT file-namen
•(MCL: eine entsprechende FILE-Zeile muss angegeben sein)
- Einlesen eines SPUR-Files:
+LIB READ SPUR-INPUT filename
(MCL: Eine entsprechende File-Zeile muss angegeben sein)
- Ausgeben eines 80-Bytes-Record Files:
+LIB WRITE text-name TO CARD-IMAGE-OUTPUT filename
(MCL: Eine entsprechende File-Zeile muss angegeben sein)
- Ausgeben eines SPUR-Files:
+LIB WRITE text-name TO SPUR-OUTPUT filename
(MCL: Eine entsprechende File-Zeile muss angegeben sein)

Object-Modul-Library: Anweisungen (ab CC24)

- Neuerstellung einer Library:
CREATE=librarynam/Dnn/nnnnnK/n%
 (nnnnnK=Grösse in K/n%=Index-Anteil)
- Identifikation einer bereits bestehenden Library:
LIBRARY=librarynam/Dnn/A/Dnn/nnnnnK/n%
 (/A/Dnn...: Parameters für Library-Neuerstellung bei zu wenig Platz)
- Integrieren eines zusätzlichen Moduls:
ADD=modulnamvv/Dnn
- Inaktivieren eines Moduls:
DELETE=modulnamvv
- Ersetzen eines Moduls:
REPLACE=newmodulvv/Dnn/oldmodulvv
- Bilden einer neuen Generation einer Library:
NEWLIB oder NEWLIB=librarynam/Dnn/nnnnnK/n%
- Ausdrucken eines Library-Inhaltsverzeichnisses:
PRINT

Object-Modul-Library: LINKEDIT-Anweisungen

- Ein einzelner Modul aus Library:
BIND=objmodulvv/Dnn/librarynam
BIND=objmodulvv//librarynam
BIND=objmodulvv
- Angabe einer Library, die bei den folgenden BIND-Anweisungen verwendet werden soll:
LOC=Dnn/librarynam
- Automatisches Integrieren aller benötigten Object-Moduls ins Load-Modul:
AUTOLIB=Dnn/librarynam
 (Mind. das Haupt-Object-Modul muss mit BIND=... angegeben werden)

FORMATE DER IDENTIFICATION DIVISION

IDENTIFICATION DIVISION.PROGRAM-ID. program-name.[AUTHOR. [comment-entry] ...][INSTALLATION. [comment-entry] ...][DATE-WRITTEN. [comment-entry] ...][DATE-COMPILED. [comment-entry] ...][SECURITY. [comment-entry] ...]

FORMATE DER ENVIRONMENT DIVISION

ENVIRONMENT DIVISION.CONFIGURATION SECTION.SOURCE-COMPUTER. NCR-VRX (WITH DEBUGGING MODE).OBJECT-COMPUTER. NCR-VRX [, MEMORY SIZE integer { WORDS
CHARACTERS
MODULES }]

[, PROGRAM COLLATING SEQUENCE IS alphabet-name]

[, SEGMENT-LIMIT IS segment-number].

[SPECIAL-NAVES. [, { MONITOR-FLAG-n }]

{ IS mnemonic-name-1 [, ON STATUS IS condition-name-1 [, OFF STATUS IS condition-name-2]]
IS mnemonic-name-2 [, OFF STATUS IS condition-name-2 [, ON STATUS IS condition-name-1]] } ...
{ ON STATUS IS condition-name-1 [, OFF STATUS IS condition-name-2]
OFF STATUS IS condition-name-2 [, ON STATUS IS condition-name-1] }
[, SWITCH-n { ON STATUS IS condition-name-1 [, OFF STATUS IS condition-name-2] }] ...
[, OFF STATUS IS condition-name-2 [, ON STATUS IS condition-name-1]] } ...

[(CARD-READER
CONSOLE
EXT
FLAGS
LINE-n
MONITOR-FLAGS
PARM-CARDS
PIFA
PRINTER
RVI
SYSTEMS-LOC
TRAN
TYPEWRITER
control-character) IS mnemonic-name ...]

(ENVIRONMENT DIVISION)

```
[ , ALPHABET alphabet-name IS
  {
    STANDARD-1
    NATIVE
    literal-1 [ { THROUGH }
               { THRU } literal-2
               ALSO literal-3 [, ALSO literal-4] ... ]
    [ literal-5 [ { THROUGH }
                 { THRU } literal-6
                 ALSO literal-7 [, ALSO literal-8] ... ] ] ... ]
  }
, ALPHABET alphabet-name IS { EBCDIC
                              EXTENDED-H-SET
                              KATAKANA }
```

```
[ , SYMBOLIC CHARACTERS ARE { ASCII-CONTROL [, HEXADECIMAL] }
                              { HEXADECIMAL [, ASCII-CONTROL] }
```

```
[ , CURRENCY SIGN IS literal-9 ]
```

```
[ , DECIMAL-POINT IS COMMA . ]
```

[NCR-RECOVERY SECTION.

[NCR-SAVE-DATA.

```
data-name-1 [, data-name-2] ... .]
```

[NCR-CRITICAL-DATA.

```
data-name-3 [, data-name-4] ... .]
```

[INPUT-OUTPUT SECTION.

FILE-CONTROL.

```
{file-control-entry} ...
```

[I-O-CONTROL.

```
[ ; RERUN ON RESCUE-1 EVERY integer-1 RECORDS OF file-name-1 ] ...
```

```
[ ; SAME [ RECORD
          SORT
          SORT-MERGE ] AREA FOR file-name-2 {file-name-3} ... ] ...
```

```
[ ; MULTIPLE FILE TAPE CONTAINS file-name-4 [POSITION integer-2]
```

```
  [, file-name-5 [POSITION integer-3]] ... ] ... .]
```

FORMATE DER FILE-CONTROL EINTRAGUNG

SEQUENTIAL FILE:

```
SELECT [OPTIONAL] file-name

ASSIGN TO { DISC
            MAGNETIC-TAPE
            CARD-READER
            CARD-PUNCH
            PRINTER }

[ ; RESERVE integer-1 [ AREA
                      AREAS ] ]
[ ; ORGANIZATION IS SEQUENTIAL ]
[ ; ACCESS MODE IS SEQUENTIAL ]
[ ; FILE STATUS IS data-name-1 ].
```

RELATIVE FILE:

```
SELECT file-name

ASSIGN TO DISC

[ ; RESERVE integer-1 [ AREA
                      AREAS ] ]

[ ; ORGANIZATION IS RELATIVE ]
[ ; ACCESS MODE IS { SEQUENTIAL [, RELATIVE KEY IS data-name-1]
                   { RANDOM
                     DYNAMIC } , RELATIVE KEY IS data-name-1 } ]
[ ; FILE STATUS IS data-name-2 ].
```

(FILE-CONTROL EINTRAGUNG)

INDEXED FILE:

SELECT file-name

ASSIGN TO DISC

[; RESERVE integer-1 [AREA
AREAS]]

; ORGANIZATION IS INDEXED

[; ACCESS MODE IS { SEQUENTIAL
RANDOM
DYNAMIC }]

; RECORD KEY IS data-name-1

[; ALTERNATE RECORD KEY IS data-name-2 [WITH DUPLICATES]] ...

[; FILE STATUS IS data-name-3].

SORT OR MERGE FILE:

SELECT file-name ASSIGN TO DISC.

FORMATE DER DATA DIVISION

DATA DIVISION.

[FILE SECTION.

[FD file-name

[; BLOCK CONTAINS [integer-1 TO] integer-2 {RECORDS
CHARACTERS}][; RECORD {CONTAINS [integer-3 TO] integer-4 CHARACTERS
IS VARYING IN SIZE [[FROM integer-3] [TO integer-4] CHARACTERS]}
[DEPENDING ON data-name-1]]; LABEL {RECORD IS } {STANDARD}
{RECORDS ARE } {OMITTED}[; VALUE OF {ALLOCFLAGS
DATESCHEME
ENTRY-TYPE
FILE-ID
FILE-SIZE
SUD
VOLUME-ID} IS {data-name-2}
{literal-2}][; {ALLOCFLAGS
DATESCHEME
ENTRY-TYPE
FILE-ID
FILE-SIZE
SUD
VOLUME-ID} IS {data-name-3}
{literal-2}] ...]

[; DATA {RECORD IS } data-name-4 [, data-name-5] ...]

[; LINAGE IS {data-name-6}
{integer-5} LINES [, WITH FOOTING AT {data-name-7}
{integer-6}][, LINES AT TOP {data-name-8}
{integer-7}] [, LINES AT BOTTOM {data-name-9}
{integer-8}]]

[; CODE-SET IS alphabet-name].

[record-description-entry] ...] ...

(DATA DIVISION)

[SD file-name

```
[ : RECORD { CONTAINS [integer-3 TO] integer-4 CHARACTERS
              { IS VARYING IN SIZE [[FROM integer-3] [TO integer-4] CHARACTERS] } }
  [ [DEPENDING ON data-name-1]
```

```
[ : DATA { RECORD IS } data-name-1 [, data-name-2] ... . ]
  { RECORDS ARE }
```

[record-description-entry] ...] ...]

[WORKING-STORAGE SECTION.

[77-level-description-entry] ...]

[record-description-entry] ...]

[NCR-SHARED-STORAGE SECTION.

[77-level-description-entry] ...]

[record-description-entry] ...]

[NCR-CONSTANT-STORAGE SECTION.

[77-level-description-entry] ...]

[record-description-entry] ...]

[LINKAGE SECTION.

[77-level-description-entry] ...]

[record-description-entry] ...]

[COMMUNICATION SECTION.

[communication-description-entry

[record-description-entry] ...] ...]

FORMAT DER DATENBESCHREIBUNGS-EINTRAGUNGlevel-number {data-name-1}
{FILLER}

[; REDEFINES data-name-2]

[; {PICTURE
PIC} IS character-string][; [USAGE IS] { COMPUTATIONAL
COMPE
COMPUTATIONAL-4
COMP-4
DISPLAY
INDEX }][; [SIGN IS] { LEADING
TRAILING } [SEPARATE CHARACTER][; OCCURS {integer-1 TO integer-2 TIMES DEPENDING ON data-name-3}
integer-2 TIMES }[{ASCENDING
DESCENDING } KEY IS data-name-4 [, data-name-5] ...]

[INDEXED BY index-name-1 [, index-name-2 ...]]

[; { SYNCHRONIZED } [LEFT
SYNC RIGHT][; { JUSTIFIED } RIGHT
JUST]

[; BLANK WHEN ZERO]

[; VALUE IS literal].

FORMAT DER 88-EINTRAGUNG88 condition-name; {VALUE IS } literal-1 [{THROUGH } literal-2]
{VALUES ARE } [THRU]

[, literal-3 [{THROUGH } literal-4]] ...]

FORMAT DER 66-EINTRAGUNG66 data-name-1; RENAMES data-name-2 [{THROUGH } data-name-3]
{THRU }

NCR VRX COBOL DATEN DEFINITIONEN

Datentyp	Anzahl Bits per Charakter	PICTURE Charakters		USAGE Angabe	SIGN Angabe
		S	Daten Charakters		
Alphanumerisch	8	Nein	X	[DISPLAY]	Keine
Dezimal, ohne Vorzeichen	8	Nein	9	[DISPLAY]	Keine
Dezimal, Vorzeichen hinten	8	Ja	9	[DISPLAY]	TRAILING SEPARATE
Dezimal, Vorzeichen vorne	8	Ja	9	[DISPLAY]	LEADING SEPARATE
Dezimal, Zonen-Vorzeichen hint.	8	Ja	9	[DISPLAY]	[TRAILING]
Dezimal, Zonen-Vorzeichen, vorne	8	Ja	9	[DISPLAY]	LEADING
Dezimal gepackt, ohne Vorzeichen	4	Nein	9	COMP	Keine
Dezimal gepackt, mit Vorzeichen	4	Ja	9	COMP	Keine
Binär, mit Vorzeichen	-	Ja	9	Comp-4	Keine

FORMATE DER COMMUNICATION SECTION

INPUT CD EINTRAG: FORMAT 1

```

CD cd-name; FOR INPUT
[[: SYMBOLIC QUEUE IS data-name-1]
[; SYMBOLIC SUB-QUEUE-1 IS data-name-2]
[; SYMBOLIC SUB-QUEUE-2 IS data-name-3]
[; SYMBOLIC SUB-QUEUE-3 IS data-name-4]
[; MESSAGE DATE IS data-name-5]
[; MESSAGE TIME IS data-name-6]
[; SYMBOLIC SOURCE IS data-name-7]
[; TEXT LENGTH IS data-name-8]
[; END KEY IS data-name-9]
[; STATUS KEY IS data-name-10]
[; MESSAGE COUNT IS data-name-11]].

```

INPUT CD EINTRAG: FORMAT 2

```

CD cd-name; FOR INPUT
[data-name-1, data-name-2, ... , data-name-11].

```

OUTPUT CD EINTRAG:

```

CD cd-name; FOR OUTPUT
[[: DESTINATION COUNT IS data-name-1]
[; TEXT LENGTH IS data-name-2]
[; STATUS KEY IS data-name-3]
[; DESTINATION TABLE OCCURS integer-1 TIMES
[; INDEXED BY index-name-1 [, index-name-2] ... ]]
[; ERROR KEY IS data-name-4]
[; SYMBOLIC DESTINATION IS data-name-5]].

```

FORMATE DER PROCEDURE DIVISION

FORMAT 1:

PROCEDURE DIVISION [USING data-name-1 [, data-name-2] ...].

[DECLARATIVES.

(section-name SECTION [segment-number]. declarative-sentence

[paragraph-name. [sentence] ...] ...) ...

END DECLARATIVES.]

(section-name SECTION [segment-number].

[paragraph-name. [sentence] ...] ...) ...

FORMAT 2:

PROCEDURE DIVISION [USING data-name-1 [, data-name-2] ...].

(paragraph-name. [sentence] ...) ...

FORMAT DER USE ANWEISUNG (DECLARATIVES)

USE AFTER STANDARD {EXCEPTION
ERROR} PROCEDURE ON {file-name-1 [, file-name-2] ... }
INPUT
OUTPUT
I-O
EXTEND}

FORMAT DER USE FOR DEBUGGING ANWEISUNG (DECLARATIVES)

USE FOR DEBUGGING ON {cd-name-1
{ALL REFERENCES OF identifier-1}
file-name-1
procedure-name-1
ALL PROCEDURES}

[, {cd-name-2
{ALL REFERENCES OF identifier-2}
file-name-2
procedure-name-2
ALL PROCEDURES} ...] .

(PROCEDURE DIVISION: ANWEISUNGEN)

ACCEPT identifier [FROM mnemonic-name]

ACCEPT identifier FROM $\left. \begin{array}{l} \text{DATE} \\ \text{DAY} \\ \text{DAY-OF-WEEK} \\ \text{TIME} \end{array} \right\}$

ACCEPT cd-name MESSAGE COUNT

ADD {identifier-1} [literal-1] [, identifier-2] [literal-2] ... TO identifier-m [ROUNDED]
[, identifier-n [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

ADD {identifier-1} [literal-1] , {identifier-2} [literal-2] [, identifier-3] [literal-3] ...
GIVING identifier-m [ROUNDED] [, identifier-n [ROUNDED]] ...
[; ON SIZE ERROR imperative-statement]

ADD $\left. \begin{array}{l} \text{CORRESPONDING} \\ \text{CORR} \end{array} \right\}$ identifier-1 TO identifier-2 [ROUNDED]
[; ON SIZE ERROR imperative-statement]

ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2
[, procedure-name-3 TO [PROCEED TO] procedure-name-4] ...

CALL {identifier-1} [literal-1] [USING data-name-1 [, data-name-2] ...]
[; ON OVERFLOW imperative-statement]

CANCEL {identifier-1} [literal-1] [, identifier-2] [literal-2] ...

(PROCEDURE DIVISION: ANWEISUNGEN)

CLOSE file-name-1 $\left[\begin{array}{l} \text{REEL} \\ \text{UNIT} \end{array} \right] \left[\begin{array}{l} \text{WITH NO REWIND} \\ \text{FOR REMOVAL} \end{array} \right]$
WITH $\left. \begin{array}{l} \text{NO REWIND} \\ \text{LOCK} \end{array} \right\}$

[, file-name-2 $\left[\begin{array}{l} \text{REEL} \\ \text{UNIT} \end{array} \right] \left[\begin{array}{l} \text{WITH NO REWIND} \\ \text{FOR REMOVAL} \end{array} \right]$] ...
WITH $\left. \begin{array}{l} \text{NO REWIND} \\ \text{LOCK} \end{array} \right\}$ *

COMPUTE identifier-1 [ROUNDED] [, identifier-2 [ROUNDED]] ...
= arithmetic-expression [; ON SIZE ERROR imperative-statement]

DELETE file-name RECORD

DELETE file-name RECORD [; INVALID KEY imperative-statement]

DISABLE INPUT [TERMINAL] cd-name WITH KEY {identifier-1} [literal-1]

DISABLE OUTPUT cd-name WITH KEY {identifier-1} [literal-1]

DISPLAY {identifier-1} [literal-1] [, identifier-2] [literal-2] ... [UPON mnemonic-name]

DIVIDE {identifier-1} [literal-1] INTO identifier-2 [ROUNDED]
[, identifier-3 [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

DIVIDE {identifier-1} [literal-1] INTO {identifier-2} [literal-2] GIVING identifier-3 [ROUNDED]
[, identifier-4 [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

(PROCEDURE DIVISION: ANWEISUNGEN)

DIVIDE {identifier-1} BY {identifier-2} GIVING identifier-3 [ROUNDED]
 {literal-1} {literal-2} {literal-3} [ON SIZE ERROR imperative-statement]
 [, identifier-4 [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

DIVIDE {identifier-1} INTO {identifier-2} GIVING identifier-3 [ROUNDED]
 {literal-1} {literal-2} {literal-3} [REMAINDER identifier-4 [; ON SIZE ERROR imperative-statement]]

DIVIDE {identifier-1} BY {identifier-2} GIVING identifier-3 [ROUNDED]
 {literal-1} {literal-2} {literal-3} [REMAINDER identifier-4 [; ON SIZE ERROR imperative-statement]]

ENABLE INPUT [TERMINAL] cd-name WITH KEY {identifier-1}
 {literal-1}

ENABLE OUTPUT cd-name WITH KEY {identifier-1}
 {literal-1}

EXIT.

EXIT PROGRAM.

GO TO [procedure-name]

GO TO procedure-name-1 [, procedure-name-2] ... , procedure-name-n

DEPENDENT ON identifier

IF condition THEN {statement-1; ELSE statement-2}
 {NEXT SENTENCE; ELSE statement-2}
 {statement-1; ELSE NEXT SENTENCE}
 statement-1.

(PROCEDURE DIVISION: ANWEISUNGEN)

INSPECT identifier-1 TALLYING

{identifier-2 FOR {ALL LEADING CHARACTERS} {identifier-3} {literal-1}} {BEFORE AFTER} INITIAL {identifier-4} {literal-2}} ... }

INSPECT identifier-1 REPLACING

{CHARACTERS BY {identifier-6} {literal-4} {BEFORE AFTER} INITIAL {identifier-7} {literal-5}}
 {ALL LEADING FIRST} {identifier-5} {literal-3} BY {identifier-6} {literal-4} {BEFORE AFTER} INITIAL {identifier-7} {literal-5}} ... }

INSPECT identifier-1 TALLYING

{identifier-2 FOR {ALL LEADING CHARACTERS} {identifier-3} {literal-1}} {BEFORE AFTER} INITIAL {identifier-4} {literal-2}} ... }

REPLACING

{CHARACTERS BY {identifier-6} {literal-4} {BEFORE AFTER} INITIAL {identifier-7} {literal-5}}
 {ALL LEADING FIRST} {identifier-5} {literal-3} BY {identifier-6} {literal-4} {BEFORE AFTER} INITIAL {identifier-7} {literal-5}} ... }

MERGE file-name-1 ON {ASCENDING DESCENDING} KEY data-name-1 [, data-name-2] ...

[ON {ASCENDING DESCENDING} KEY data-name-3 [, data-name-4] ...] ...

{COLLATING SEQUENCE IS alphabet-name}

USING file-name-2, file-name-3 [, file-name-4] ...

{OUTPUT PROCEDURE IS section-name-1 {THROUGH THEN} section-name-2}}
 {GIVING file-name-5}

(PROCEDURE DIVISION: ANWEISUNGEN)

MOVE {identifier-1} TO identifier-2 [, identifier-3] ...

MOVE {CORRESPONDING
CORR} identifier-1 TO identifier-2

MULTIPLY {identifier-1}
{literal-1} BY identifier-2 [ROUNDED]
[, identifier-3 [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

MULTIPLY {identifier-1}
{literal-1} BY {identifier-2}
{literal-2} GIVING identifier-3 [ROUNDED]
[, identifier-4 [ROUNDED]] ... [; ON SIZE ERROR imperative-statement]

NCR-LOCK identifier-1 FOR RETRIEVAL [; STATUS IS identifier-2]

NCR-LOCK identifier-1 FOR UPDATE [; STATUS IS identifier-2]

NCR-UNLOCK ALL

NCR-UNLOCK identifier-1 [; STATUS IS identifier-2]

OPEN {INPUT file-name-1 [WITH NO REWIND] [, file-name-2 [WITH NO REWIND] ... }
{OUTPUT file-name-3 [WITH NO REWIND] [, file-name-4 [WITH NO REWIND] ... } ...
{I-O file-name-5 [, file-name-6] ... }
{EXTEND file-name-7 [, file-name-8] ... }

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2 {identifier}
{integer} TIMES

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2 UNTIL condition-1

(PROCEDURE DIVISION: ANWEISUNGEN)

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2

VARYING {identifier-1}
{index-name-1} FROM {identifier-2}
{index-name-2} BY {identifier-3}
{literal-1} {literal-2}

UNTIL condition-1

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2

VARYING {identifier-1}
{index-name-1} FROM {identifier-2}
{index-name-2} BY {identifier-3}
{literal-1} {literal-2}

UNTIL condition-1

AFTER {identifier-4}
{index-name-3} FROM {identifier-5}
{index-name-4} BY {identifier-6}
{literal-3} {literal-4}

UNTIL condition-2

PERFORM procedure-name-1 {THROUGH
THRU} procedure-name-2

VARYING {identifier-1}
{index-name-1} FROM {identifier-2}
{index-name-2} BY {identifier-3}
{literal-1} {literal-2}

UNTIL condition-1

AFTER {identifier-4}
{index-name-3} FROM {identifier-5}
{index-name-4} BY {identifier-6}
{literal-3} {literal-4}

UNTIL condition-2

AFTER {identifier-7}
{index-name-5} FROM {identifier-8}
{index-name-6} BY {identifier-9}
{literal-5} {literal-6}

UNTIL condition-3

PURGE cd-name

(PROCEDURE DIVISION: ANWEISUNGEN)

SET identifier-1 [, identifier-2] ... TO index-name

SORT file-name-1 ON {ASCENDING
DESCENDING} KEY data-name-1 [, data-name-2] ...

[ON {ASCENDING
DESCENDING} KEY data-name-3 [, data-name-4] ...] ...

[COLLATING SEQUENCE IS alphabet-name]

{ INPUT PROCEDURE IS section-name-1 {THROUGH
THRU} section-name-2 }
{ USING file-name-2 [, file-name-3] ... }

{ OUTPUT PROCEDURE IS section-name-3 {THROUGH
THRU} section-name-4 }
{ GIVING file-name-4 }

START file-name [KEY { IS EQUAL TO
IS =
IS GREATER THAN
IS >
IS NOT LESS THAN
IS NOT < } data-name]

[; INVALID KEY imperative-statement]

STOP {RUN
literal}

STRING { identifier-1 } [, identifier-2] ... DELIMITED BY { identifier-3 }
{ literal-1 } [, literal-2] { literal-3 }
{ SIZE }

[{ identifier-4 } [, identifier-5] ... DELIMITED BY { identifier-6 }
{ literal-4 } [, literal-5] { literal-6 }
{ SIZE }

INTO identifier-7 [WITH POINTER identifier-8]

[; ON OVERFLOW imperative-statement]

(PROCEDURE DIVISION: ANWEISUNGEN)

SUBTRACT { identifier-1 } [, identifier-2] ... FROM identifier-m [ROUNDED]

[, identifier-n [ROUNDED] ... [; ON SIZE ERROR imperative-statement]

SUBTRACT { identifier-1 } [, identifier-2] ... FROM { identifier-m }
{ literal-1 } [, literal-2] { literal-m }

GIVING identifier-n [ROUNDED] [, identifier-o [ROUNDED] ...

[; ON SIZE ERROR imperative-statement]

SUBTRACT { CORRESPONDING
CORE } identifier-1 FROM identifier-2 [ROUNDED]

[; ON SIZE ERROR imperative-statement]

UNSTRING identifier-1

[DELIMITED BY [ALL] { identifier-2 } [, { identifier-3 }] ...]
{ literal-1 } [, { literal-2 }] { literal-3 }

INTO identifier-4 [, DELIMITER IN identifier-5] [, COUNT IN identifier-6]

[, identifier-7 [, DELIMITER IN identifier-8] [, COUNT IN identifier-9] ...

[WITH POINTER identifier-10] [TALLYING IN identifier-11]

[; ON OVERFLOW imperative-statement]

WRITE record-name [FROM identifier1]

[{ BEFORE
AFTER } ADVANCING { identifier-2 } [LINE
integer] [LINES]]
{ mnemonic-name }
{ PAGE }

[; AT { END-OF-PAGE
EOF } imperative-statement]

WRITE record-name [FROM identifier] [; INVALID KEY imperative-statement]

FORMATE DER KONDITIONS-EINTRAGUNGEN

RELATION KONDITION:

$$\left. \begin{array}{l} \{ \text{identifier-1} \\ \text{literal-1} \\ \text{arithmetic-expression-1} \\ \text{index-name-1} \end{array} \right\} \left\{ \begin{array}{l} \text{IS [NOT] GREATER THAN} \\ \text{IS [NOT] LESS THAN} \\ \text{IS [NOT] EQUAL TO} \\ \text{IS [NOT] } > \\ \text{IS [NOT] } < \\ \text{IS [NOT] } = \\ \text{EXCEEDS} \\ \text{EQUALS} \\ \text{IS UNEQUAL TO} \end{array} \right\} \left. \begin{array}{l} \{ \text{identifier-2} \\ \text{literal-2} \\ \text{arithmetic-expression-2} \\ \text{index-name-2} \end{array} \right\}$$

CLASS KONDITION:

$$\text{identifier IS [NOT] } \left\{ \begin{array}{l} \text{NUMERIC} \\ \text{ALPHABETIC} \end{array} \right\}$$

SIGN KONDITION:

$$\left. \begin{array}{l} \{ \text{identifier} \\ \text{arithmetic-expression} \end{array} \right\} \text{ IS [NOT] } \left\{ \begin{array}{l} \text{POSITIVE} \\ \text{NEGATIVE} \\ \text{ZERO} \end{array} \right\}$$

SWITCH-STATUS KONDITION:

$$\text{condition-name}$$

CONDITION-NAME KONDITION:

$$\text{condition-name}$$

NEGATED SIMPLE KONDITION:

$$\text{NOT simple-condition}$$

COMBINED KONDITION:

$$\text{condition-1 } \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \text{ condition-2 } \dots$$

ABBREVIATED COMBINED RELATION KONDITION:

$$\text{relation-condition } \left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \text{ [NOT] [relational-operator] object } \dots$$

VERSCHIEDENE EINTRAGUNGEN

QUALIFIKATION:

$$\left. \begin{array}{l} \{ \text{record-name} \\ \text{data-name} \\ \text{condition-name} \end{array} \right\} \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ file-name}$$

$$\left. \begin{array}{l} \{ \text{data-name-1} \\ \text{condition-name} \end{array} \right\} \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ data-name-2 } \dots$$

$$\text{paragraph-name } \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ section-name}$$

$$\text{text-name } \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ library-name}$$

SUBSCRIPTING:

$$\left. \begin{array}{l} \{ \text{data-name} \\ \text{condition-name} \end{array} \right\} (\text{subscript-1 } [, \text{subscript-2 } [, \text{subscript-3}]])$$

INDEXING:

$$\left. \begin{array}{l} \{ \text{data-name} \\ \text{condition-name} \end{array} \right\} (\left. \begin{array}{l} \{ \text{index-name-1 } [(\{\pm\} \text{literal-2})] \\ \text{literal-1} \end{array} \right\} \left[\begin{array}{l} \{ \text{index-name-2 } [(\{\pm\} \text{literal-4})] \\ \text{literal-3} \end{array} \right] \left[\begin{array}{l} \{ \text{index-name-3 } [(\{\pm\} \text{literal-6})] \\ \text{literal-5} \end{array} \right] \right] \right)$$

IDENTIFIER: FORMAT 1

$$\text{data-name-1 } \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ data-name-2 } \dots$$

$$[(\text{subscript-1 } [, \text{subscript-2 } [, \text{subscript-3}]])]$$

IDENTIFIER: FORMAT 2

$$\text{data-name-1 } \left[\begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right] \text{ data-name-2 } \dots \left[(\left. \begin{array}{l} \{ \text{index-name-1 } [(\{\pm\} \text{literal-2})] \\ \text{literal-1} \end{array} \right\} \left[\begin{array}{l} \{ \text{index-name-2 } [(\{\pm\} \text{literal-4})] \\ \text{literal-3} \end{array} \right] \left[\begin{array}{l} \{ \text{index-name-3 } [(\{\pm\} \text{literal-6})] \\ \text{literal-5} \end{array} \right] \right] \right)$$

NCR VRX COBOL: RESERVED WORDS

ACCEPT	COMPUTE	END	INDEXED
ACCESS	CONFIGURATION	END-OF-PAGE	INDICATE
ACK	CONSOLE	ENQ	INITIAL
ADD	CONTAINS	ENTER	INITIATE
ADVANCING	CONTROL	ENTRY-TYPE	INPUT
AFTER	CONTROLS	ENVIRONMENT	INPUT-OUTPUT
ALL	COPY	EOP	INSPECT
ALLOCFLAGS	CORR	EOT	INSTALLATION
ALPHABET	CORRESPONDING	EQUAL	INTO
ALPHABETIC	COUNT	EQUALS	INVALID
ALSO	CR	ERROR	IS
ALTER	CURRENCY	ESC	
ALTERNATE		ESI	JUST
AND	DATA	ETB	JUSTIFIED
ARE	DATE	ETX	
AREA	DATE-COMPILED	EVERY	KATAKANA
AREAS	DATE-WRITTEN	EXCEEDS	KEY
ASCENDING	DATESCHEME	EXCEPTION	
ASCII-CONTROL	DAY	EXIT	LABEL
ASSIGN	DAY-OF-WEEK	EXT	LAST
AT	DC1	EXTEND	LEADING
AUTHOR	DC2	EXTENDED-H-SET	LEFT
	DC3		LENGTH
BEFORE	DC4	FD	LESS
BEL	DE	FF	LF
BLANK	DEBUG-CONTENTS	FILE	LIMIT
BLOCK	DEBUG-ITEM	FILE-CONTROL	LIMITS
BOTTOM	DEBUG-LINE	FILE-ID	LINAGE
BS	DEBUG-NAME	FILE-SIZE	LINAGE-COUNTER
BY	DEBUG-SUB-1	FILLER	LINE
	DEBUG-SUB-2	FINAL	LINE-COUNTER
CALL	DEBUG-SUB-3	FIRST	LINE-n
CAN	DEBUGGING	FLAG-n	LINES
CANCEL	DECIMAL-POINT	FLAGS	LINKAGE
CARD-PUNCH	DECLARATIVES	FOOTING	LOCK
CARD-READER	DELETE	FOR	LOW-VALUE
CD	DELIMITED	FROM	LOW-VALUES
CF	DELIMITER	FS	
CH	DEPENDING		MAGNETIC-TAPE
CHARACTER	DESCENDING	GENERATE	MEMORY
CHARACTERS	DESTINATION	GIVING	MERGE
CLOCK-UNITS	DETAIL	GO	MESSAGE
CLOSE	DISABLE	GREATER	MODULES
COBOL	DISC	GROUP	MONITOR-FLAG-n
CODE	DISPLAY	GS	MONITOR-FLAGS
CODE-SET	DIVIDE		MOVE
COLLATING	DIVISION	HEADING	MULTIPLY
COLUMN	DLE	HEXADECIMAL	
COMMA	DOWN	HIGH-VALUE	
COMMUNICATION	DUPLICATES	HIGH-VALUES	NAK
COMP	DYNAMIC	HT	NATIVE
COMP-3		I-O	NCR-CONSTANT-
COMP-4	EBCDIC	I-O-CONTROL	STORAGE
COMP-6	EGI	IDENTIFICATION	NCR-CRITICAL-
COMPUTATIONAL	ELSE		DATA
COMPUTATIONAL-3EM		IF	NCR-RECOVERY
COMPUTATIONAL-4EM		IN	NCR-LOCK
COMPUTATIONAL-6ENABLE		INDEX	NCR-REPLY

(NCR VRX COBOL: RESERVED WORDS)

NCR-SAVE-DATA	READ	SPACES	VOLUME-ID
NCR-SHARED-	RECEIVE	SPECIAL-NAMES	VT
STORAGE	RECORD	STANDARD	WHEN
NCR-TIMER	RECORDS	STANDARD-1	WITH
NCR-UNLOCK	REDEFINES	START	WORDS
NCR-VRX	REEL	STATUS	WORKING-
NCR-VRX-	REFERENCES	STOP	STORAGE
OUTCOME	RELATIVE	STRING	WRITE
NEGATIVE	RELEASE	STX	
NEXT	REMAINDER	SUB	ZERO
NO	REMOVAL	SUB-QUEUE-1	ZEROS
NOT	RENAMES	SUB-QUEUE-2	ZEROS
NUL	REPLACING	SUB-QUEUE-3	
NUMBER	REPORT	SUBTRACT	+
NUMERIC	REPORTING	SUD	-
	REPORTS	SUM	*
OBJECT-	RERUN	SUPPRESS	/
COMPUTER	RESCUE-1	SWITCH-n	**
OCCURS	RESERVE	SYMBOLIC	>
OF	RESET	SYN	<
OFF	RETRIEVAL	SYNC	>
OMITTED	RETURN	SYNCHRONIZED	>
ON	REVERSED	SYSTEMS-LOG	=
OPEN	REWIND		
OPTIONAL	REWRITE	TABLE	
OR	RF	TALLYING	
ORGANIZATION	RS	TAPE	
OUTPUT	RH	TERMINAL	
OVERFLOW	RIGHT	TERMINATE	
	ROUNDED	TEXT	
PAGE	RUN	THAN	
PAGE-COUNTER	RVI	THEN	
PARM-CARDS		THROUGH	
PERFORM	SAME	THRU	
PF	SD	TIME	
PH	SEARCH	TIMES	
PIC	SECTION	TO	
PICTURE	SECURITY	TOP	
PIPA	SEGMENT	TRAILING	
PLUS	SEGMENT-LIMIT	TRAN	
POINTER	SELECT	TYPE	
POSITION	SEND	TYPEWRITER	
POSITIVE	SENTENCE		
PRINTER	SEPARATE	UNEQUAL	
PRINTING	SEQUENCE	UNIT	
PROCEDURE	SEQUENTIAL	UNSTRING	
PROCEDURES	SET	UNTIL	
PROCEED	SI	UP	
PROGRAM	SIGN	UPDATE	
PROGRAM-ID	SIZE	UPON	
PURGE	SO	US	
	SOH	USAGE	
QUEUE	SORT	USE	
QUOTE	SORT-MERGE	USING	
QUOTES	SOURCE		
	SOURCE-	VALUE	
RANDOM	COMPUTER	VALUES	
RD	SPACE	VARYING	

NCR VRX FILE STATUS WERTE

Content of File Status Data Item	Status of Input-Output Operation	OPEN	CLOSE	READ	WRITE	REWRITE	DELETE	START
00	Successful completion of input-output operation	SRI	SRI	SRI	SRI	SRI	RI	RI
10	At end condition			SRI				
21	Invalid key due to sequence error				I	I		
22	Invalid key due to duplicate key				RI			
23	Invalid key due to no record found			RI		RI	RI	RI
24	Invalid key due to boundary violation				RI			
30	Permanent error condition	SRI	SRI	SRI	SRI	SRI	RI	RI
34	Permanent error condition due to boundary violation				S			
90	Internal software error	SRI	SRI	SRI	SRI	SRI	SRI	SRI
91	Temporary resource error; e.g. • DSA not available • Disc volume has been removed	SRI			SRI			
92	Permanent resource error; e.g. • Block header has incorrect format number • Block numbers do not match • Records out of sequence on indexed or relative file • Key index directory out of order with associated indexed file • VLI is incorrect	SRI		SRI	SRI	SRI	RI	RI
93	Parameter error; e.g. • Fixed length records specified but variable length records present • Key length error • Attempt to read an output file • Record size error • Keys don't match on a REWRITE • Record size error during WRITE	SRI	SRI	SRI	SRI	SRI	R*	RI
94	Logic error; e.g. • Attempt to read an unopened file • REWRITE or DELETE does not follow a successful READ	SRI	SRI	SRI	SRI	SRI	RI	RI
95	Outside intervention could correct error; e.g. • disc unit down	SRI	SRI	SRI	SRI	SRI	RI	RI
96	Illogical operation due to action of another task	SRI	SRI	SRI	SRI	SRI	RI	RI
97	Record truncated on READ; e.g. • 200 byte record on file but 100 byte input area				SRI			
98	Translation error				SRI	SRI	SRI	
9A	Lock aborted on locked block (shared file only)			SRI	SRI	SRI	RI	
9B	Requested block is located and not available because locked by another user (shared file only)				SRI	SRI	RI	RI

S = sequential file organization
R = relative file organization
I = indexed file organization

VRXCIBOL: SORT Job-Angaben

- Für jeden COBOL-SORT muss die folgende Zeile im Job-Step stehen:

CC18 CC24

SORTWF SORT-Parameters

- Mögliche SORT-Parameters:

FILESIZE=nnnnnn	Größe jedes WF
IDNAME=xxxxxxxxx	WF-Namen aus Programm
WORKFILESUDS=Dnn/Dnn/...	WF-Suds
RECORD=F/nnnnn	Übersteuern der Record-
RECORD=V/xxxxx/yyyy/zzzzz	Angaben aus Programm (fak.)
SECTORSIZE=nnnnn	WF-Sektorgröße (Def.=512)
INPUTVOLUME=nnnnnnnn	Anzahl zu sort. Records (fak.)
EQUALKEY=x	Originalreihenfolge bei gleichen Keys (x=Y oder N/fak.)
ORDER=x	Record-Input-Sequenz (x=B,N oder R/fak.)
INPUTPROSZE=nnnnnnn	Real-Memory-Bedarf der Input-Procedure (Def.=4096 Bytes/fak.)
OUTPUTPROSZE=nnnnnnn	Real-Memory-Bedarf der Output-Procedure (Def.=4096 Bytes/fak.)
MERGECHECK=x	Prüfen der Input-Sequenz bei MERGE (x=Y oder N/fak.)
SORTALG=x	Sort-Algorithmus (x=T, Q oder S/fak.)

5 DATEN MANAGEMENT

TOTAL DATENBANK SYSTEM

TOTAL DML Befehle	5-20
TOTAL DML Status Codes	5-30

CAM DATEN MANAGEMENT SYSTEM

CAM: Aufbau einer seq. oder rel. Datei	5-500
CAM: Aufbau einer Indexed-Datei (ICAM)	5-520
CAM: Reorganisation einer Indexed-Datei (ICAM)	5-550
CAM: Aufbau eines TOICAM-Indexed-Files	5-560
CAM: Reorganisation eines TOICAM-Indexed-Files	5-560

NCR-DMS/LOGICAL VIEW

LESEN (GET)	5-800
EINFÜGEN (STORE)	5-800
UPDATEN (MODIF 4)	5-800
LOSCHEN (DELETE)	5-800
SPEZIAL-BEFEHLE	5-810

Ausführliche und massgebende Informationen in:

- RM-0900 NCR TOTAL
- RM-0236-V11 VRX FILE MANAGEMENT
- RM-0323-A NCR-DMS/LOGICAL VIEW REL. 2.0

TOTAL DML FUNCTIONS

ADD-M	ADD MASTER	Record einfügen
ADDVA	ADD VARIABLE AFTER	Record einfügen
ADDVB	ADD VARIABLE BEFORE	Record einfügen
ADDVC	ADD VARIABLE CONTINUE	Record einfügen
ADDVR	ADD VARIABLE REPLACE	Record neu verketten
*BTRAN	BEGIN TRANSACTION	1. logischer Befehl in Transaktion
DEL-M	DELETE MASTER	Record löschen
DELVD	DELETE VARIABLE DIRECT	Record löschen
*ETRAN	END TRANSACTION	1. logischer Befehl in Transaktion
*FREEEX	FREE HELD RESOURCES	gelockte Records freigeben
+ MARKL	MARK LOG RECORD	User Record ins Log schreiben
+ QUIET	SET QUIET POINT IN DATA BASE	Checkpoint setzen
RDNXT	READ NEXT	Record sequentiell lesen
READD	READ DIRECT	Record lesen
READM	READ MASTER	Record lesen
READR	READ REVERSE	Record lesen
READV	READ VARIABLE	Record lesen
RQLOC	REQUEST LOCATION	RRN berechnen lassen
+ SINOF	SIGN OFF	DB abschliessen
+ SINON	SIGN ON	DB eröffnen
+ WRITD	WRITE DIRECT	Record direkt schreiben
WRITM	WRITE MASTER	Record zurückschreiben
WRITV	WRITE VARIABLE	Record zurückschreiben

* f. NCR TOTAL CENTRAL (TOTAL 80)

+ IN TRANPRO-APPLIKATIONS-MODULE NICHT ERLAUBT

TOTAL DML STATUS CODES

STATUS	FATAL	BEDEUTUNG
BCHN		BROKEN CHAIN
BCTL		BLANK CONTROL FIELD
DUPM		DUPLICATE MASTER RECORD
DUPO	X	DUPLICATE OPEN OF A DATA SET
ENTF		ELEMENT NOT FOUND
EXSO		EXTRA SINON COMMAND
FNOP	X	FILE NOT OPEN
FNTF	X	FILE NOT FOUND
FTYP		INVALID FILE TYPE
FULL		FILE LOADED TO CAPACITY
FUNC		INVALID FUNCTION CODE
HELD		RECORD LOCKED
IFAM		INVALID FILE ACCESS MODE
IMDL		INVALID MASTER DELETE
IPAR		INVALID NUMBER OF PARAMETERS
IRLC	X	INVALID RECORD LOCATION
IVRC		INVALID RECORD CODE
IVRD		INVALID VARIABLE ENTRY DATA SET READ
IVRP		INVALID REFERENCE PARAMETER
IVWD		WRITD NOT ALLOWED
LOAD		VARIABLE ENTRY FILE LOADED BEYOND CYLINDER LOAD LIMIT
LOCK	X	DATA SET LOCKED
LGNA		LOG FILE NOT ACTIVE
LSZE		LOG SIZE ERROR
MFNF		MASTER FILE NOT FOUND
MLNF		MASTER LINK NOT FOUND
MRNF		MASTER RECORD NOT FOUND
MTBL	X	NO MODULE TABLE FOUND
NHLD		NO "READ WITH HOLD"
NOJO	X	NO ASSIGNED I-O AREA
NOSO		NO SINON
NOST		NO START LOGICAL TRANSACTION
NSMR		NO SECONDARY MASTER RECORD FOUND
REPC		RECORD TO MODIFY IS UNLOCKED
RSVD	X	INVALID RESERVE DATA SET
UACM	X	UNDEFINED ACCESS MODE
UCTL		UNEQUAL CONTROL FIELD
ULGO	X	UNDEFINED LOGGING OPTIONS
UPDE		UPDATE MODE ERROR
VMRE	X	VARIABLE READ MASTER ERROR
WAIT		WAIT FOR QUIET

TOTAL ZUGRIFF (TOTAL BASIC)

MASTER FILES:

CALL "DATBAS" USING OPERATION,
 TOTAL-STATUS,
 DATA-SET,
 CONTROL-KEY,
 TOTAL-LIST
 TOTAL-DATA,
 ENDP.

VARIABLE FILES:

CALL "DATBAS" USING OPERATION,
 TOTAL-STATUS,
 DATA-SET,
 REFERENCE,
 LINKAGE-PATH,
 CONTROL-KEY,
 TOTAL-LIST,
 TOTAL-DATA.

TOTAL ZUGRIFF (TOTAL CENTRAL)

MASTER FILES:

CALL "DATBAS" USING OPERATION,
 TOTAL-STATUS,
 DATA-SET,
 CONTROL-KEY,
 TOTAL-LIST,
 TOTAL-DATA,
 END-PARA.*

VARIABLE FILES:

CALL "DATBAS" USING OPERATION,
 TOTAL-STATUS,
 DATA-SET,
 REFERENCE,
 LINKAGE-PATH,
 CONTROL-KEY,
 TOTAL-LIST,
 TOTAL-DATA,
 END-PARA.*

* END-PARA: "END." nur Current Record locken
 "RLSE" Current Record nicht locken
 "MHLD" Current Record nicht locken, vorher
 gelockte Records nicht entlocken

ACHTUNG: "END." DARF IN TRANPRO-UMGEBUNG
 NICHT BENÜTZT WERDEN!

CAM: DER AUFBAU EINES SEQ. ODER REL. FILES*** Minimal-FCL für Aufbau eines Ein-Section-Files**

FILE REFERENCE=...
 SUD=Dnn
 DATE SCHEME=... 1)
 RETENTION=... 2)
 BLOCK SIZE=... 3)
 FILE SIZE=... 4)
 NO SHARE=... 6)
 NAIL BUFFERS=... 6)

*** Minimal-FCL für Aufbau eines Mehrfach-Section-Files**

FILE REFERENCE=...
 DATE SCHEME=... 1)
 RETENTION=... 2)
 BLOCK SIZE=... 3)
 NO SHARE 6)
 NAIL BUFFERS 6)
 SECTION=1
 SECTION SIZE=... 5)
 SUD=Dnn
 SECTION=2
 SECTION SIZE=... 5)
 SUD=Dnn
 etc.

*** Erklärungen zu den FCL-Parameters**

- 1) Art und Weise des Dateischutzes
- 2) Dauer des Dateischutzes
- 3) Blockgröße (o/CAM-Block-Header)
- 4) Filegröße in 'n/RECORDS' oder 'n/BLOCKS'
- 5) Sectiongröße in 'n/RECORDS' oder 'n/BLOCKS'
- 6) Angaben zur Optimierung des File-Aufbaus.

Zusätzlich: – Bei rel. Files; Input wenn möglich sortiert
 – evtl. bei entspr. Disc-Unit: 'SHARE=NO'

CAM: DER AUFBAU EINES ICAM-FILES*** Minimal-FCL für Aufbau eines Ein-Section-Files**

FILE REFERENCE=...

SUD=Dnn

DATE SCHEME=...

1)

RETENTION=...

2)

BLOCK SIZE=...

3)

BLK PER SGMT=...

4)

BLKFREESPACE=...

5)

SEGFREESPACE=...

6)

SGMT PER SXN=...

7)

NO SHARE

8)

NAIL BUFFERS

8)

* Minimal-FCL für Aufbau eines Mehrfach-Section-ICAM-Files

FILE REFERENCE=...	
DATE SCHEME=...	1)
RETENTION=...	2)
BLOCK SIZE=...	3)
BLK PER SGMT=...	4)
BLKFREESPACE=...	5)
SEGFREESPACE=...	6)
SGMT PER SXN=...	9)
NO SHARE	8)
NAIL BUFFERS	8)
SECTION=1	
SUD=Dnn	
HIGH KEY=...	10)
SECTION=2	
SUD=Dnn	
HIGH KEY=...	10)
etc.	

* Erklärungen zu den FCL-Parameters

- 1) Art und Weise des Dateischutzes
- 2) Dauer des Dateischutzes
- 3) Blockgrösse (Errechnen lassen mit 'RUNJ,CAMBSIZE')
- 4) Anzahl Blöcke im Segment (berechnen, sehr oft 19 oder 39)
- 5) Platzreserve pro Segment/ca. 25% der Gesamtreserve hier angeben
- 6) Platzreserve pro Segment/ca. 25% der Gesamtreserve hier angeben
- 7) Anzahl Segmente im ganzen File/ca. 50% der Gesamtreserve hier einplanen
- 8) Angaben zur Optimierung des File-Aufbaus.
Zusätzlich: – Input sortiert
– evtl. bei entspr. Disc-Unit:
 'SHARE=NO'
– ACCESS SEQUENTIAL/OPEN OUTPUT
 (im VRXCOBOL-Programm)
- 9) Anzahl Segmente pro Section/eher mehr als 50% (zulasten von BLKFREESPACE und SEGFREESPACE) der Gesamtreserve hier einplanen (verteilt auf die einzelnen Sections).
- 10) Schlüsselspektrum der entsprechenden Section

CAM: REORGANISATION EINES ICAM-FILES*** Minimal-Angaben für Reorganisation**

(eines Ein-Section-Files)

- RMINMEM mind. 128KB

- LOCALDSA mind. 256KB

- JOBSTEP für Reorganisation:

```

NEXTDO  SORTVRX
FILE     REFERENCE=SORTIN1,SUD=Dnn
        NAME=altindexDF
        BLOCKSIZE=                1)
        KEY1=III/pp/A/AS          2)
        RECORD=V/III              3)
WORKFL   FILESIZE=                4)
        SUD=Dnn                    4)
FILE     REFERENCE=SORTOUT,SUD=Dnn
        NAME=neuindexDF
        ORGANIZATION=INDEXED
        DATE SCHEME=GEN,RETENTION=
        BLOCKSIZE=                5)
        SGMTPERSXN=                6)
        BLKPERSGMT=                7)
        KEYOFFSET=                 8)
        KEYSIZE=                   9)
        BLKFREESPACE=              10)
        SEGFREESPACE=              10)
        NOSHARE,NAILBUFFERS

```

STOPRD

- 1) Blockgröße des zu reorg. Files
- 2) Key-Angaben: Länge, Position
- 3) Record-Längen-Angabe
- 4) Workfile-Definitionen (vgl. SORTVRX)
- 5) Blockgröße des neuen Files
- 6) Anzahl Segmente pro Section
- 7) Anzahl Blöcke pro Segment
- 8) Position des Keys (rel. 0)
- 9) Länge des Keys in Bytes
- 10) Platzreserve

CAM: DER AUFBAU EINES TOICAM-FILES*** Minimal-FCL für Aufbau** (eines Ein-Section-Files)

```

FILE     REFERENCE=
        SUD=Dnn
        DATE SCHEME=GEN,RETENTION=n
        BLOCK SIZE=
        FILE SIZE=
        NO SHARE, NAIL BUFFERS
        ASSOCIATION=KEY
        BLOCK SIZE=10220

```

CAM: REORGANISATION EINES TOICAM-FILES*** Minimal-Angaben für Reorganisation**

(eines Ein-Section-Files)

```

NEXTDO  TOIREBLD
FILE     REFERENCE=OLD-DATA-FILE,SUD=Dnn
        NAME=altefileDF

```

(***)

STOPRD

(***) Bei Bedarf kann zusätzlich angegeben werden:

```

FILE     REFERENCE=NEW-DATA-FILE, und weitere
        FCL-Parameters (z.B. NAME=, SUD=,
        SECTION= etc.)

```

NCR-DMS/LOGICAL VIEW

LESEN

GET [TEST DIRECT] [LOCK UNLOCK HOLD] [NEXT PRIOR LAST FIRST] {RECORD record-name} {element-name [...]} .

[USING cursor-name] [THRU THROUGH FROM] path-name set-name [WHERE FOR] condition .

EINFUGEN

STORE [TEST] [AFTER BEFORE LAST] {RECORD record-name} {THROUGH THRU FROM} set-name {element-name [...]} .

[WHERE FOR] {condition} {KEY=literal} .

UPDATEN

MODIFY [TEST] RECORD {record-name} {element-name [...]} .

LÖSCHEN

DELETE [TEST] RECORD {record-name} {element-name [...]} .

SPEZIAL-BEFEHLE

SCHEMA ERÖFFNEN

OPEN [TEST] [DATABASE] entity-name [FOR]

```

LOAD
EXCLUSIVE
SHARE
DICTIONARY
UNCONDITIONAL [USE]
}READ-ONLY {
}UPDATE { [AND]
}EXCLUSIVE [USE]
}SHARE [FOR] {UPDATE
}READ-ONLY {

```

Open Command		Total Central	Total Basic	CAM
This User	Others			
Exclusive	-	PRIV	PRIV	Update
Load	-	PRIV	PRIV	Update
Share	-	SHRE	SHRE	Update
Dictionary	-	SHRE	SHRE	Update
Read	Exclusive	EXCL	PRIV	Update
Update	Exclusive	EXCL	PRIV	Update
Read-Only	Update	SHRE	SHRE	Update
Read-Only	Read-Only	IUPD	SHRE	Update
Update	Update	SUPD	SHRE	Update
Update	Read-Only	PRIV	PRIV	Update
Unconditional	-	OPEN	OPEN	Update

OPEN Options for Physical Files

SCHEMA ABSCHLIESSEN

CLOSE [TEST] [DATABASE] entity-name.

CLEAR REALMS

CLEAR [CURSOR-NAME].

START EINER LOGISCHEN TRANSAKTION

START

ENDE EINER LOGISCHEN TRANSAKTION

COMMIT

ANNULLIEREN EINER LOGISCHEN TRANSAKTION

ROLLBACK.

6 TELEPROCESSING

NCR-DMS/TRANPRO III

Operator-Befehle	6-20
Runtime-Parameters	8-60
Format-Tabellen	6-90
CAMBAS-BEFEHLE	6-150
CAMBAS-STATUS	6-160

Ausführliche und massgebende Informationen in:

- VRX TRANPRO RM-0220-A

07.85

OPERATOR-BEFEHLE

nnnn = Job-Nummer

TRANPRO-Abschluss

nnnn:C SHUT FINISH

TRANPRO-Unterbruch

nnnn:C SHUT STOP

Transaktion starten

nnnn:C PUT ttttt { $\begin{matrix} H \\ C \end{matrix}$ } dddd...ddd

ttttt : TX-Nummer

ddd..d: Daten

H : Eingabe der Daten in HEX

C : Eingabe der Daten in ASCII

DISPLAY MESSAGE LEVEL

nnnn:C MSGL

MESSAGE LEVEL ändern

nnnn:C MSGL nn

nn: 00-05

DISPLAY V. QUIET POINT LIMITE

nnnn:C DISQP

Ändern v. QUIET POINT LIMITE

nnnn:C SETQP CCCCCCCC, tttt

ccccccc: Anzahl Tx (0-16777215)

tttt : Zeitintervall (0-1440 Minuten)

QUIET ausführen

nnnn:C QUIET

FORMAT LÖSCHEN

nnnn:C DELETE ffffftt

fffff: Name des Formates, sechsstellig

tt : Terminal-Suffix (T0, T2 etc.)

FORMAT SPERREN

nnnn:C DISABL ffffftt

FORMAT ENTPERREN

nnnn:C ENABLE ffffftt

MODULE TRACE

nnnn:C TRACE mmmmmmmm

nnnn:C NOTRAC mmmmmmmm

mmmm: Module-Name

SECURITY-SIGNON-LOGS aktivieren

nnnn:C AUDIT

nnnn:C NOAUDIT

DISPLAY ALLE OFFENEN TRANPRO-MELDUNGEN

nnnn:C LIST

DISPLAY DATUM/ZEIT

nnnn:C TIME

OPERATOR MESSAGE LOG AKTIVIEREN

nnnn:C MSGLOG

OPERATOR MESSAGE LOG INAKTIVIEREN

nnnn:C NOMSGLOG

OPERATOR MESSAGE LOG DRUCKEN

nnnn:C PUT 091000 C 010200/ALL, ALL, HC

TERMINAL-BEFEHLEnnnn: { T } OPEN { tttttttttt }
 { S } { ssssss } } TERMINAL ERÖFFNENnnnn: { T } CLOSE { tttttttttt }
 { S } { ssssss } } TERMINAL SCHLIESSENnnnn: { T } RESET { tttttttttt }
 { S } { ssssss } } TERMINAL REAKTIVIEREN

ttttttttt: Name des Terminals oder ALL

sssss: Nummer des Terminals

MELDUNGEN AN TERMINALS SCHICKEN

nnnn:T MADD ALL "xxxxxxxx" an alle Terminals

nnnn:T MADD ttttttttt "xxxx" an Terminal mit Name tttttttttt

nnnn:T MADD ALL tttt... "xxxxxxxx" an alle Terminals, deren
Name mit tttt... beginnen**TERMINAL STATUS ANSCHAUEN (SWISS SOFT)**

nnnn:C PUT 810161 C xxxxx

xxxxx:	ALL	alle Terminals
	OPEN	nur offene Terminals
	CLOSE	nur geschlossene Terminals
	tttt...	Terminal mit bestimmten Namen (auch partiell)

RUNTIME PARAMETER

(ÜBERSTEUERN die Angaben, die bei der TRANPRO-Generierung gemacht wurden)

```

NEXTDO      ppppppppp
LOAD        REPLICAS=nn
FILE        REFERENCE=xxxxxx,...
:
SPEC$ OPTION xxxxxx
SPEC$ OPTION xxxxxx
:
SPEC$ TOTAL xxxxxx
SPEC$ TOTAL xxxxxx
:
SPEC$ CAM   xxxxxx
SPEC$ CAM   xxxxxx
:
STOPRD

```

OPTION-KARTEN

```

REPLICAS=nn      Anzahl User-Tasks, muss gleich sein
                  wie bei LOAD-Parameter
MSGLEVEL=nn      MESSAGE LEVEL (00-05)
LGFULL=nn        FÜLLQUOTIENT v.
                  TOTAL BEFORE IMAGE LOG in %
FMTLIB=ddd/ttt/xxxxxxxx/Dnn  FORMAT-LIBRARY
SYSLIB=ddd/ttt/xxxxxxxx/Dnn  SYSTEM-FILE
QUIET            QUIET-PUNKTE SETZEN
NOQUIET         KEINE QUIET-PUNKTE SETZEN
RECOVERY        EVTL. RECOVERY DURCHFÜHREN
NORECOVERY      KEIN RECOVERY DURCHFÜHREN
                  (nur im Test-Betrieb empfohlen!)

```

```

TERMOPEN        alle aktiven Terminals automatisch
                  eröffnen
NOTERMOPEN      keine Terminals eröffnen
SDDELETE        SAVE-DATA löschen
NOSDELETE       SAVE-DATA nicht löschen
QPREC          QUIET-POINT-RECOVERY erwünscht
                  (für TOTAL)
NOQPREC        KEIN QUIET-POINT-RECOVERY
MSGLOG         OPERATOR MESSAGE LOG
                AKTIVIEREN
NOMSGLOG       OPERATOR MESSAGE LOG NICHT
                AKTIVIEREN
TOTAL          TOTAL DB-SYSTEM WIRD BENÜTZT
                  (Default)
NOTOTAL        TOTAL DB-SYSTEM NICHT BENÜTZT
CAM            CAM-FILES WERDEN BENÜTZT
                  (Default)
NOCAM          CAM-FILES WERDEN NICHT BENÜTZT

```

TOTAL KARTEN

NOFILES

in TRANPRO-Tabelle definierte

TOTAL-Files nicht eröffnen

PROGRAM=pppppppp

Name des Loadmodules

DBMOD=dddddd

Name des DB-Modules

aaaaaa

Access Mode (RONLY, UPDATE)

lllll

LOGGING (LOG, NOLOG)

ffff=mmmm

TOTAL-File, das zu eröffnen ist,

:

ffff: Name

mmmm: File-Mode

(PRIV, SHRE, OPEN)

→ OPEN nur beim Testen !

CAM-KARTEN

xxxxx,yyyy,zzzz

Namen der CAM-Files, die zu
eröffnen sind (entsprechende

FCL-Statements müssen

vorhanden sein !)

FORMAT-KARTEN

Card Type	Card Column	Description
	1- 6 7 11-12	ALL CARDS Page and line number Asterisk Card type number
10	14-19 21-30 35-38 45-47 49-50 55-56 58	FORMAT ID CARD Format identification Author Terminal device type identification Line length for device Extra null characters Logical terminal type (format ID suffix) Comma replacement of decimal point
15	14 16 18 21-70	STANDARD FIRST LINE CARD (Optional) Function switch Standard first line switch Disconnect transaction Titel of format
16	21-33	STANDARD FIRST LINE MODIFICATION CARD (Optional) Replacement for NEXT FUNCTION field
20	14-19 21-26 60	CHAINING CARD Transaction code Next format ID Partial Screen Output=V
21	14-18 21-26	ALTERNATE FORMAT ID CARD (Optional) Alternate format number Alternate format ID
25	14-15 17-19 21-23 31 33-37 39-43 45 47-51 53 55-59 61	COMMON ERROR MESSAGE CARD (Optional) Line position of message Column position of message Message length Multiple error, first error, last error switch Multiple error message number Incorrect data type – message number Incorrect data type – fill character Required field missing – message number Required field missing – fill character Incorrect field length – message number Incorrect field length – fill character
31	14-15 17-19 21-23 26 28 30-31	FIELD LIMIT CARD Line position of the field Column position of the field Length of the field Input field indicator (I) Cursor position indicator (X) Numeric positions right of decimal point

FORMAT-KARTEN

Card Type	Card Column	Description
32	14-17	FORMS CONTROL CARD Channel selection (FORM or VTAB)
35	14-15 17-19 21-23 25-28 31 39-43 45 47-51 53 55-59 61	SPECIFIC FIELD ERROR/TRIGGERED MESSAGE CARD Line position of the message field Column position of the message field Length of the message field Displacement in the output string Fill or redisplay code 1 = fill character, special message 2 = redisplay input, special message 3 = program message trigger 4 = fill character, special message, message trigger 5 = redisplay input, special message, message trigger Incorrect data type – message number Incorrect data type – fill character Required field missing – message number Required field missing – fill character Incorrect field length – message number Incorrect field length – fill character
40	21-80	INITIAL VALUE CARD Constant data
51	14-17 19-21 23 24 26-27 29	OUTPUT VARIABLE CARD Displacement in output parameter data string Output field length Data type of field in output data string } s. S. 6-110 Data type of output display field } Edit rule s. S. 6-120 Fill character
61	14-17 19-21 23 24 26-27 29	INPUT VARIABLE CARD Displacement in the input parameter data string Input field length Data type of field in input display Data type of field in input data string } s. S. 6-110 Edit rule code for converting data s. S. 6-130 Fill character
66	14-17 19-21	INPUT CONSTANT CARD Displacement in input data string Length of constant field
70		END FORMAT CARD
80	14-18 20-22	MESSAGE DEFINITION CARD Message number Message length

PERMITTED DATA TYPE COMBINATIONS

Output Data Types	
Output Data String Type (Card 51 CC 23)	Display Types Allowed (Card 51 CC 24)
A	A
X	X
U	U, Y, T
D	D, S
K	U, Y, T
P	D
B	U, D, H
Input Data Types	
Input Display Type (Card 61 CC 23)	Input Data String Types Allowed (CC 24)
A	A
X	X
U	B, K, P, U
D	P, D
S	D
H	B
Codes	
A = alpha A-Z or space X = character or space U = numeric 1-9 D = numeric, trailing sign S = numeric, leading sign H = hexadecimal Y = date field; output only T = time field, output only K = packed numeric, unsigned P = packed numeric, signed B = binary	

Datenkonvertierung vom OUTPUT-DATA-STRING auf den Bildschirm:

- 00: genauso ausgeben, wie die Daten im Output-Data-String definiert sind, d.h. ohne Umwandlung
- 04: Vornullen unterdrücken, nur gültig für numerische Daten
- 0C: Vornullen ersetzen durch Füllzeichen (CC 29), nur gültig für numerische Daten
- 20: Dezimalpunkt hinzufügen, nur gültig für numerische Daten
- 24: Dezimalpunkt hinzufügen und Vornullen unterdrücken, nur gültig für numerische Daten
- 2C: Dezimalpunkt hinzufügen, Vornullen unterdrücken und füllen mit Füllzeichen, nur gültig für numerische Daten

Datenkonvertierung vom Bildschirm in den INPUT-DATA-STRING

- 00: nicht obligatorisches Feld, Eingabe ohne Umwandlung, für numerische Felder volle Länge erforderlich, gültig für alle Datentypen
- 09: nicht obligatorisches Feld linksbündig gefüllt und rechts mit dem Füllzeichen (CC 29) ergänzt, nur gültig für Typen A oder X
- 0A: nicht obligatorisches Feld, rechtsbündig gefüllt und mit Nullen aufgefüllt, nur gültig für U- und H-Typen; wenn keine Daten eingegeben werden, wird das Feld mit Nullen initialisiert oder mit dem Füllzeichen (CC 29) gefüllt
- 10: obligatorisches Feld, volle Länge der Eingabe verlangt für numerische Felder, gültig für alle Datentypen
- 19: obligatorisches Feld, linksbündig übertragen und rechts mit dem Füllzeichen (CC 29) gefüllt, nur gültig für A- und X-Typen
- 1A: obligatorisches Feld, rechtsbündig übertragen und links mit Nullen aufgefüllt, nur gültig für U- und H-Typen
- 20: nicht obligatorisches Feld mit Dezimalseparator, volle Länge verlangt, nur gültig für numerische Daten
- 2A: nicht obligatorisches Feld mit Dezimalseparator, rechtsbündige Übertragung und links mit Nullen gefüllt, nur gültig für numerische Daten
- 30: obligatorisches Feld mit Dezimalseparator, volle Länge verlangt, nur gültig für numerische Daten
- 3A: obligatorisches Feld mit Dezimalseparator, rechtsbündige Übertragung, links mit Nullen gefüllt nur gültig für numerische Felder

CAMBAS-BEFEHLE (INDEXED + RELATIVE FILES)

READ	RECORD LESEN
WRITE	RECORD EINFÜGEN
REWRITE	RECORD MODIFIZIEREN
DELETE	RECORD LÖSCHEN
STARTEQ	START EQUAL
STARTGT	START GREATER
STARTGE	START GREATER ODER EQUAL
READNEXT	NÄCHSTER RECORD LESEN
READHOLD	RECORD LESEN + RECORD LOCKEN
READLOCK	RECORD LESEN + BLOCK LOCKEN
UNLOCK	BLOCK ENTLOCKEN
*RDNEXT	NÄCHSTER RECORD LESEN (START KEY > xxx + READ NEXT)
*GETFIRST	WIE RDNEXT, JEDOCH WIRD DER GELESENE RECORD NICHT ENTLOCKT
*GETNEXT	NÄCHSTER RECORD LESEN (NACH EINEM GETFIRST)

ACHTUNG: VOR DEM BENÜTZEN VON GETFIRST UND
GETNEXT STUDIERT MAN DEN
SYSTEM-SOFTWARE STATUS REPORT V-0115,
REF. 7.1!

* SWISS-SOFT-BEFEHLE

CAMBAS-STATUS

- OK (CAM '00') – Successful completion
- OKDP (CAM '02') – Successful, but duplicate key
- ENDF (CAM '10') – End of File
- IKOS (CAM '21') – Invalid key out of sequence
- IKDP (CAM '22') – Invalid key – duplicate
- IKNF (CAM '23') – Invalid key – no record found
- IKBO (CAM '24') – Invalid key – boundary overrun
- PE (CAM '30') – Permanent error
- PEBO (CAM '34') – Permanent error – boundary overrun
- SE (CAM '90') – Internal software error
- TERE (CAM '91') – Temporary software error
- PERE (CAM '92') – Permanent resource error
- PARM (CAM '93') – Parameter error
- LE (CAM '94') – Logic error
- INVT (CAM '95') – Intervention required
- LETK (CAM '96') – Illogical operation due to action of another task
- TREC (CAM '97') – Truncated record
- TRNS (CAM '98') – Translating error
- VEIL (CAM '99') – Veiled record
- ABLK (CAM '9A') – Aborted lock
- LOCK (CAM '9B') – Locked block
- FLNF – File not found
- FUNC – Invalid function
- UNKN – Unknown error
- TASK – Task ID not in tables
- FLNO – File not opened
- RDON – Update attempted on read-only File
- NOST – Update attempted with no Start logical Transaction

8 SOFTWARE DEVELOPMENT ENVIRONMENT

Components of SDE	8-20
Reserved Words	8-40
Symbols	8-50
Tool Syntax	8-80
Linedit Commands	8-200
Textform Commands	8-240
Pilot Commands	8-290

Ausführliche und massgebende Informationen in folgenden Manuals (SDE Reference Library):

- D1-0085-A
- D1-0086-A
- D1-0087-A
- D1-0089-A
- D1-0091-A

Stand: VRX Rel. 11, SDE Rel. 2.5. / 0785

COMPONENTS OF SDE

BASE

The Base Support Environment is the nucleus of SDE. It supports the SDE Data Base and provides the session and access control for the environment. BASE also allows you to modify and expand the environment to adapt it to your specific software development requirements.

SOLIAC

The System of Library and Change Control is the information management component of SDE. It provides you with an organized storage system for files, and a system for tracking and maintaining multiple versions of a file in a minimum amount of space.

STARS

Software Test Analysis and Reporting System. It allows you to test your software designs quickly and efficiently at various stages of development. It creates the data needed to test the software, simulates the output for parts of the program that are not yet developed and prints a report of the test results.

MODULAR TOOLS

The data manipulation component of SDE. Modular Tools are a collection of single-function tools specifically designed to perform the manipulations and comparisons of data and text files that you need so frequently during your software development process.

PLUME

The text manipulation component of SDE. It allows you to create, edit, and format programs or text. It also contains a Spelling Checker/Dictionary that you can expand or modify to fit your requirements.

PILOT

The instructional language of SDE. You can present on-line computer-assisted instruction to the users of the software that you develop. You can present instruction, provide drill and practice, design simulations, and test students to assure that they are well trained and ready to use your software to its fullest extent.

RESERVED WORDS

Word	Function
CANCEL	Cancels command before execution is complete.
HELP	Obtains further information on a parameter, tool, or subject matter.
HUGE	Largest allowable integer in SDE [(2**31)-1 or 2147483647].
INPUT*	Refers to the terminal keyboard or to a procedure file as Standard Input.
KEYBOARD*	Refers to the terminal keyboard as Standard Input.
MAXLINE	The maximum length of a record or line in SDE.
OUTPUT*	Refers to the user terminal display as Standard Output.
PRINTER*	Refers to the system printer as Standard Output.

*May be used to direct Standard Input or Standard Output for a tool, pipeline, or procedure.

SYMBOLS

Listed below are the special symbols that are used in SDE command lines. Consult the LINEDIT and PILOT Command Lists for other special symbols used with those commands.

COMMAND SYNTAX SYMBOLS

Function	Symbol	Example
command terminator	: Colon	COPYPART COPY=4:
pipeline indicator	/ Slash	CAP < MYFILE / REVERSE > YOURFILE
string delimiter	() Parentheses	PULL TXT=(JOHN MARY)
parameter indicator	, Comma	GAP 10,20
parameter separator	Blank	NUMBER 1..10
comment initiator	# Number Sign	#Comment line
parameter delimiter	' Single or Double Quotes	OVERLAY ?' < MYFILE
keyword identifier	= Equal Sign	HELP SUBJECT=FILES

FILE INPUT/OUTPUT SYNTAX SYMBOLS

Function	Symbol	Example
Standard Output	> Greater-than	COPYPART COPY=4 >PRINTER
Standard Input	< Less-than	COPYPART COPY=4 <FILEA
filename separator	+ Plus Sign	MYFILE+CAT+S
qualifier	. Period	CREATELE ELEMENT=GAMES.ASTRO

PROCEDURE FILE SUBSTITUTION SYMBOL

Function	Symbol	Example
parameter substitutor	! Exclamation Mark	COPYPART !1 !2 !3

PATTERN SYMBOLS

Function	Symbol	Example
character class delimiter	[] Bracket	REMOVE CHARS=[A-Z] <MYFILE
negator	~ Tilde	REMOVE CHARS=[~A-Z] <MYFILE
Range Separator	- Hyphen	REMOVE CHARS=[A-Z] <MYFILE
Pattern Repeater	* Asterisk	TRANSLIT <A OLD=%[0-9]* NEW=' '
Wild Card	? Question Mark	EXTRACT PAT=NUMBER??????
Beginning Anchor	% Percent Sign	EXTRACT PAT=%The
Ending Anchor	\$ Dollar	EXTRACT PAT=.s
Symbol Identifier	@ Escape	GETBLOCK @T.@S

TOOL SYNTAX

The following is an alphabetical tool syntax list for the tools available to you in BASE, Modular Tools, SOLIAC, STARS, PILOT, and Plume. Following each tool syntax is a brief description of the tool's function. At the end of some of the descriptions, you will find a symbol indicating that the tool can be pipelined. If no symbol is shown, the tool cannot be used in a pipeline.

- [P] Indicates that the tool can be used any place in a pipeline.
- [P] Indicates that the tool can be used only at the beginning of a pipeline.
- P] Indicates that the tool can be used only at the end of a pipeline.

ADDWORD WORD=string [DICTFILE=objname]
Adds a new word into the dictionary (Plume)

BUILDTEST TEST=string MODULE=string
ENTRY=string DDEF=string
Builds a user test tool (STARS)

CAP [CASE=x]
Makes all the alphabetical characters in the same file the same case (Modular Tools) [P]

CHANGEACTL OBJECT=objname
[ULIST=(string,...) DELUSER=(string,...)]
Changes Access Control (BASE)

CHANGEDD DDEF=ddefname
[DDESC=ddescid QUALIFY=x]
Changes an existing Data Definition (STARS)

CHANGEDONE CHANGEID=string
Finalizes a change set (SOLIAC) [P]

CHANGEHELP SUBJECT=subject
Changes a Help Definition (BASE) P]

CHANGEMSG SUBJECT=subject NUMBER=99
Changes a Message Definition (BASE) P]

CHANGEOBJ OBJECT=objname
Changes an Object Definition (BASE)

CHANGETOOL TOOL=objname
Changes a user Tool Definition (BASE)

CHANGEVER [VERSION=string]
Modifies a version definition (SOLIAC) [P]

CHANGEWORD FROM=string TO=string
[DICTFILE=objname]
Changes a word in the dictionary (Plume)

CLOAK KEY=string

Makes the contents of a file unreadable
(Modular Tools) [P]

CLOCK

Writes the date and the time (Modular Tools) [P]

COB [CHARS=cclass]

Copies, omitting specified characters (Modular
Tools) [P]

COMBINE FILE=filename

[COPY=99 SKIP=99 FCOPY=99]

Inserts a portion of one file into another file
(Modular Tools) [P]

COMMON FILE=filename

Merges lines from two files, tagging them to
indicate source and identity (Modular Tools) [P]

COMPARE FILE=filename

Compares the lines in two files (Modular Tools) [P]

COMPRESS

Compresses repeated characters in a file
(Modular Tools) [P]

COMPRESSVER [FILE=filename]

Compresses version definitions (SOLIAC) [P]

CONCAT [FILES=(filename,...)]

Combines files into one (Modular Tools) [P]

CONTRAST FILE2=filename [MAXWORK=99]

Compares two files, showing the differences
(Modular Tools) [P]

COPYPART [SKIP=99 COPY=99

LEND=99 REND=99]

Copies all or part of a file (Modular Tools) [P]

COUNT

Counts the number of characters, words,
and lines (Modular Tools) [P]

**CREATEDD DDEF=ddefname [DDESC=ddescid
QUALIFY=x]**

Creates a Data Definition or adds to an existing
one (STARS)

CREATEHELP SUBJECT=subject

Creates a Help Definition (BASE) [P]

CREATELE [ELEMENT=libname]

Creates a library element (SOLIAC) [P]

CREATELIB [LIBRARY=libname]

Creates a library (SOLIAC)

CREATEMSG SUBJECT=subject NUMBER=99

Creates a Message Definition (BASE) [P]

CREATEOBJ OBJECT=objname

Adds a new Object Definition (BASE)

CREATETOOL TOOL=objname

Adds a new Tool Definition (BASE)

CREATEVER VERSION=string IDLIST=string

Creates a version definition (SOLIAC) [P]

CUT CHARS=cclass

Starts a new line at specified characters
(Modular Tools) [P]

**DELETE [COPY=99 SKIP=99 LEND=99
REND=99]**

Deletes part of a file (Modular Tools) [P]

DELETEACTL OBJECT=objname

Deletes Access Control (BASE)

**DELETEDD DDEF=ddefname [DESC=ddescid
QUALIFY=x]**

Deletes a Data Definition or description (STARS)

DELETEHELP SUBJECT=subject

Deletes a Help Definition (BASE)

DELETELE [ELEMENT=libname]

Deletes a library element (SOLIAC)

DELETEDLIB [LIBRARY=libname]

Deletes a library (SOLIAC)

DELETEDMSG SUBJECT=subject NUMBER=99

Deletes a Message Definition (BASE)

DELETEOBJ OBJECT=objname

Deletes an Object Definition (BASE)

DELETEDOOL Tool=objname

Deletes a Tool Definition (BASE)

DELETEDWORD WORD=string [DICTFILE=objname]

Deletes a word from the dictionary (Plume)

DESPOOL LINES=99

Converts a text file to a spool file
(Modular Tools) [P]

DETAB [TABS=string]

Expands the tab characters with blanks
(Modular Tools) [P]

DIFF FILE=filename

Lists the differences between two files
(Modular Tools) [P]

DISCARD TXT=(string...)

Copies lines, omitting a specified string
(Modular Tools) [P]

DUP LEND=99 REND=99 TGT=99

Duplicates a specified field in each line
(Modular Tools) [P]

ECHO TXT=(string...)

Writes strings to a file (Modular Tools) [P]

EXPAND

Restores a compressed file to its original
form (Modular Tools) [P]

EXTRACT PAT=pattern [FIRST=99 LINES=99]

Extracts lines with a specified pattern
(Modular Tools) [P]

FIND PAT=pattern [FIRST=99 LINES=99]

Finds a pattern in a file (Modular Tools) [P]

FIX [LL=99]

Makes all the lines the same length
(Modular Tools) [P]

GAP [LEND=99 WIDTH=99]

Inserts a gap at a specified point in a record
(Modular Tools) [P]

GATHER

Reads filenames and combines the contents of
the files (Modular Tools) [P]

GEN TXT=(string...) [DUP=99]

Generates a file from strings (Modular Tools) [P]

GENDATA FILE=filename DDEF=ddefname

COUNT=99 [GENTYPE=x]

Generates random test data for a file or entry
point definition (STARS)

GETBLOCK STARTPAT=pattern ENDPAT=pattern

Copies specified blocks of data (Modular Tools) [P]

GETELE [ELEMENT=libname LOCK=x]

Retrieves the contents (data) of a library element (SOLIAC) [P]

GETVER [VERSION=string CCNUM=x MODULEID=x]

Gets a version of a (non-formatted) file (SOLIAC) [P]

GROUP LEND=99 REND=99 [GRP=x]

Inserts or removes group headings (Modular Tools) [P]

HELP [SUBJECT=subject MORE=x]

Obtains user information on a subject area (BASE)

INDENT LEND=99

Indents each line in a file (Modular Tools) [P]

INSERT TXT=(string,...) [COPY=99]

Inserts lines into a file (Modular Tools) [P]

JOIN [CHAR=x MWRD=99 CWRD=99 WIDTH=99]

Combines lines which agree in a specified field (Modular Tools) [P]

KWIC [TITLE=string]

(Key Word In Context) Writes each word on a separate line, along with the remainder of the line (Modular Tools) [P]

LINEDIT [WFSIZE=99 CMDFILE=filename]

Invokes LINEDIT editor. See LINEDIT Command List (Plume) [P]

LISTACTL OBJECT=objname

Lists Access Control information (BASE) [P]

LISTALLACTL [PASSWORD=string]

Lists all Access Control information (BASE) [P]

LISTALLHELP [PASSWORD=string]

Lists all Help Definitions (BASE) [P]

LISTALLMSG [PASSWORD=string]

Lists all Message Definitions (BASE) [P]

LISTALLOBJ [PASSWORD=string]

Lists all Object Definitions (BASE) [P]

LISTALLTOOL [PASSWORD=string]

List all Tool Definitions (BASE) [P]

LISTALLWORD DICTFILE=objname

Lists the contents of the dictionary to Standard Output (Plume) [P]

LISTCID CHANGEID=string

Lists a change set (SOLIAC) [P]

LISTDD DDEF=ddefname [DDESC=ddescid QUALIFY=x EXPLODE=x]

Displays the description specified by the Data Description identifier (STARS) [P]

LISTHELP SUBJECT=subject

Lists a Help Definition (BASE) [P]

LISTLIB [LIBRARY=libname]

Lists information on library objects (SOLIAC) [P]

LISTMSG SUBJECT=subject NUMBER=99

Lists a Message Definition (BASE) [P]

LISTOBJ OBJECT=objname

Lists an Object Definition (BASE) [P]

LISTTOOL TOOL=objname

Lists a Tool Definition (BASE) [P]

LISTVER

Lists version definitions (SOLIAC) [P]

LLNGTH

Linlength writes the length of each line
(Modular Tools) [P]

MERGE FILE=filename [ORDER=x]

Merges two files (Modular Tools) [P]

MERGELIBS MINOR=libname**[MAJOR=libname]**

Merges two libraries (SOLIAC)

MERGEWORD OBJECT=objname**[DICTFILE=objname]**

Merges a text file with the dictionary (Plume)

MESH FILE=filename [COPY1=99 COPY2=99]

Interlaces two files (Modular Tools) [P]

NUMBER [FIRST=99 INCR=99]

Inserts a line number at the beginning of
each line (Modular Tools) [P]

OVERLAY TXT=string [LEND=99]

Replaces part of each line in a file
(Modular Tools) [P]

OWL

Writes one word per line (Modular Tools) [P]

PAD [LL=99]

Makes all lines not less than a specified length
(Modular Tools) [P]

PILOT

Interprets PILOT programs. See PILOT Command
List (PILOT) [P]

PRINT [TITLE=string LMRGN=99 LINES=99**PAGENO=99 NUM=99]**

Creates a spool file from a text file
(Modular Tools) [P]

PULL TXT=(string,...)

Copies lines containing specified strings
(Modular Tools) [P]

PURGE FILE=filename

Deletes a file (Modular Tools) [P]

PUTELE [ELEMENT=libname REASON=string]

Replaces the contents of an element
(SOLIAC) [P]

QUERY

Displays assumed session attributes
(BASE) [P]

QUIT

Exits SDE (BASE)

RECOVERDD DDEF=ddefname

Clears a Data Definition locked during an
abnormal termination (STARS)

RECOVERHELP SUBJECT=subject

Recovers a Help Definition (BASE)

RECOVERLIB LIBRARY=libname

Unlocks a library (SOLIAC)

RECOVERMSG SUBJECT=string NUMBER=99

Recovers a Message Definition (BASE)

RECOVEROBJ OBJECT=objname

Recovers access to an Object Definition (BASE)

REMOVE CHARS=cclass

Removes specified characters from a file
(Modular Tools) [P]

REPORTTEST [LIST=string]

Prints a formatted test report (STARS) [P]

REVERSE

Writes the line backwards (Modular Tools) [P]

ROTATE

Rotates the words in a line (Modular Tools) [P]

SAVECC CCFILE=filename VERSION=string
CHANGEID=string [REASON=string MAXWORK=99]
 Creates a change set (SOLIAC) [P]

SET

Sets assumed session attributes (BASE)

SHOW [BASE=99 NUM=x CHAR=x]
 Displays the contents of a file (Modular Tools) [P]

SORT [ORDER=x]
 Sorts a file (Modular Tools) [P]

SPELL [DICTFILE=objname]
 Checks each word in the file against
 the dictionary (Plume) [P]

STARTACTL OBJECT=objname [ULIST=(string,...)]
 Starts Access Control (BASE)

STARTCC NAME=string
 Puts a file under Change Control (SOLIAC) [P]

SUB OLD=string NEW=string [FIRST=99 LINES=99]
 Substitutes one string for another
 (Modular Tools) [P]

TAIL [LINES=99]
 Copies only the last part of a file
 (Modular Tools) [P]

TEE FILE=filename
 Creates an extra copy of a file
 (Modular Tools) [P]

< test program tool > [DATA=string COUNT=99]
 User-defined tool that runs the test that was
 built using BUILDTEST (user-defined) (STARS) [P]

TEXTFORM

Formats text files containing text formatting
 commands. See TEXTFORM Command List
 (Plume) [P]

TRANSLIT OLD=pattern NEW=spattern
[FIRST=99 LINES=99]
 Transliterates (substitutes) one text pattern
 for another (Modular Tools) [P]

TRIM [CHARS=cclass]
 Removes specified trailing characters
 (Modular Tools) [P]

TRUNC [CHARS=cclass]
 Truncates a line at specified characters
 (Modular Tools) [P]

UNIQUE [REND=99]
 Deletes any line that is identical to the
 previous line (Modular Tools) [P]

UNROT [MIDPT=99]
 Unrotates the lines produced by ROTATE
 (Modular Tools) [P]

XREF [TITLE=string]
 Creates an index file (Modular Tools) [P]

LINEDIT COMMANDS

APPEND—Adds or inserts lines from a file or keyboard
[single-line-reference]A[L] [(skip|.copy)] [=filename]

ASSUME—Assumed command when none is specified
(pseudo-command)
range

CANCEL—Cancels the LINEDIT session
C

DELETE—Removes lines from the workfile
[range]D[L][delimited-string]

GAUGE—Displays a gauge line on the screen
G

HELP—Provides on-line operation information

```

| H | [subject]
| HELP |

```

LIST—Lists lines from the workfile to the user terminal
[range]L[delimited-string]

MOVE—Moves lines within the range
[range]M single-line-reference

NULL—What LINEDIT does when only a carriage return is entered as a command (CR)

OPTION—Allows you to set or examine LINEDIT options

```

O [option-letter] | { Y
                    | N
                    | Setting } ||

```

Option		Possible Setting	Assumed Value
B	Bound Checking	Y or N	N
L	List Command	Y or N	N
M	Margin Set	Y or N	N
ML	Left Margin	1 to 80	1
MR	Right Margin	1 to 80	1
P	Patterns	Y or N	N
S	Page Size	integer	20
V	Verbose	Y or N	N
W	Wrap Around	Y or N	Y

QUIT—Terminates LINEDIT and writes the contents of the workfile to Standard Output

```

| Q
| QUIT |

```

REPLACE—Replaces lines one by one
[range]R[delimited-string]

SUBSTITUTE—Replaces an existing pattern or string with another pattern or string
[range]S[L][occurrence]substitute-string

USER—Performs a user-defined command
U[L]=filename

WRITE—Writes specified lines to a new version of the named file
[range]W[L]=filename

SPECIAL CHARACTERS IN LINEDIT

Symbol	Name	Description
.	dot	Current line; also terminates APPEND and REPLACE
\$	dollar sign	Last line; also represents the last occurrence in SUBSTITUTE
*	asterisk	Entire workfile; also all occurrences in SUBSTITUTE; also, a repetition character in patterns. See Pattern Symbols
=	equal sign	Displays the number of the last line in a range
%	percent sign	Signifies the beginning of a line in patterns
&	ampersand	Displays a logical page of text
@	at sign	Escape character. Used in strings and patterns to enter characters that normally have special meaning
+	plus	Scrolls toward the end of the workfile
-	minus	Scrolls toward the beginning of the workfile
:	colon	Separates commands on the same line
	blank	Normally ignored by LINEDIT, except within strings, filenames, and multi-digit numbers.
,	comma	Separates line numbers in a range

Symbol	Name	Description
?	question mark	A wild card character in patterns
! " ' ~ # / \ ? < >		Delimiters that can be used in strings and patterns in LINEDIT

In addition to the above special characters and their meanings, the pattern matching symbols valid in SDE are also valid in LINEDIT. Note that some symbols, such as *, have more than one meaning, depending on how they are used.

Refer to the Pattern Symbols in the Symbols Chart at the beginning of this Quick Reference Guide.

TEXTFORM COMMANDS

Command Name and Description	Command	Parameters
AS IS—Prints the text without formatting	A	text
BLANKS—Inserts blanks in the text	BLANK	nnn
BOLDFACE—Overprints for a boldface effect	BOLD	text
BOLDFACE OFF—Turns off BOLD	BOFF	
BOX COMMANDS:		
BOX, BEGIN—Draws a box	B	xx,yy,...zz
BOX, END—Draws the lower edge of the box	E	
BOX, ENLARGE—Makes the box larger	BB	xx,yy,...zz
BOX LINE—Draws a horizontal line between the outside columns of a box	L	
CENTER—Centers text	C	text
COMMENT—Allows for comments in the source text	COMMENT	text
DATE COMMANDS:		
DATE—Inserts a specified date	DATE	text
TODAY—Inserts the current date	TODAY	

Command Name and Description	Command	Parameters
EXCEPTION—Enters text into a supplemental list	X	n, text
FOLIO—Prints a standard footer on the page	FOLIO	text
FOOTNOTE—Begins the footnote	F	text
FOOTNOTE, END—Ends the footnote	Z	
FORMAT—Defines the format of a page	FORMAT	aa,bb,nn
HEADINGS:		
HEADING STYLE—Selects the style of the heading	HEAD	c
HEADING LEVEL—Automatically numbers the headings	H	nn, text
INDENT—Indents the text	I	nn
INDENT, HANGING—Indents the text after placing the first line at the margin	J	nn
JUSTIFICATION—Justifies the right margin	JUST	
JUSTIFICATION, OFF—Turns off JUST	NOJUST	
KEEP—Prevents text from being split by a page break	K	nn, text

Command Name and Description	Command	Parameters
KEEP, RELEASE— Terminates K	R	
LENGTH—Establishes the length of a line of text	LENGTH	xxx
LINE COMMANDS:		
LINES, HORIZONTAL—Draws horizontal lines	G	aa,bb,...,yy,zz
LINES, VERTICAL— Draws vertical lines	V	xx,yy,...zz
LINES, VERTICAL, ADD—Adds verti- cal lines	VV	xx,yy,...zz
LINES, VERTICAL, END—Terminates V or VV	Q	
LIST—Creates supple- mental lists	LIST	n,n,...nn,text
MARGIN—Sets the left margin of the text	MARGIN	nn
MARGIN NOTE— Inserts a mark in the margin	M	
NEW PAGE—Starts a new page before printing subsequent text	N	nnn
NULL—Delimits a command	\ (CR)	
OVERPRINT—Over- prints text	O	
PARAGRAPH— Formats text in paragraph style	P	
SEQUENCE—Prints line numbers	SEQ	

Command Name and Description	Command	Parameters
SEQUENCE, END— Terminates SEQ	NOSEQ	
SET COMMANDS:		
SET APPENDIX— Establishes the names of the Appendixes	SETAPP	text
SETE—Sets the delimiter	SETE	c
SET SECTION—Sets the section numbers	SETSEC	nnn
SET TABS—Sets the tab stops	SETTAB	nnxxx.nnyyy...
SKIP—Prints blank lines	S	nn
SPACE—Controls the spacing between the lines	SPACE	n
TABULATION COMMANDS:		
TAB—Tabulates text	T	nn
TAB, REMOVE— Returns character position to the left margin	U	
PAD AND TAB—Pad lines with dots between the last character and position nn	.T	nn
TITLE—Selects title line	TITLE	n, text
UNDERLINE— Underlines the text	UNDER	text
UNDERLINE OFF— Terminates UNDER	UOFF	

TEXTFORM PARAMETERS

Parameter Representations	Description
aa,bb,xx,yy,zz xxx,yyy,zzz	Indicates horizontal numeric character positions.
n,nn,nnn	A numeric count of character positions or number of lines, or a designation of type, such as heading selection: n has a value of 1-9; nn represents 1-99; nnn represents 1-999.
c	An ASCII character is requested as a parameter.
text	"Text" in the parameter column indicates that a line or lines of text may follow the command starting on the next line. These text lines are treated separately, according to the processing of the command.

ASSUMED VALUES OF DOCUMENT
FORMATTING COMMANDS

Command	Assumed Value
MARGIN	10
LENGTH	75 (Effective line length=LENGTH minus MARGIN)
SPACE	Single spacing
JUST/NOJUST	Justification on (JUST)
HEAD	Style A
NOSEQ/SEQ	Non-sequenced document (NOSEQ)
FORMAT	Page form margin=7, length=75, depth=60
TODAY	Today's date
SETTAB	105,210,315,420,525,630,735,840,945, 1050,1155,1260,1365,1470,1575
SETE	\

PILOT COMMANDS

Command Syntax:

*label opcode[modifier][conditioner][relational expression]:
text-field
:continue text line

Command	Syntax	Description
ACCEPT	A:	Accepts one line of input from the student
	Modifiers:	
	S	ACCEPT a single keystroke for the answer
	X	ACCEPT the answer exactly as entered
COMPUTE	C:	Performs numeric computations and alphanumeric string manipulations
	Edit Options:	
	U	Translate the string to upper case
	C	Capitalize the first letter in the string
	\ xy	Replace all instances of x by y
	\ x \	Remove all occurrences of x
DIMENSION	D:	Reserves space for string-variables and numeric arrays
END	E:	Ends a subroutine or course
ESCAPE	@	Executes a user-written subroutine called SYSX
EXECUTE INDIRECT:	XI:	Executes a program-generated command

Command	Syntax	Description
FILE CLOSE	FX:	Closes the file that is currently open
FILE DELETE	FD:	Deletes a specified file
FILE INPUT	FI:	Places data from a specified record into the string variable
FILE OPEN	F:	Opens a file
Modes:		
	I	Read only file
	O	Write only file
FILE OUTPUT	FO:	Writes data to a specified record in the file
GOTO		Jumps to a specified destination (Initiated by the student)
JUMP	J:	Jumps to a specified destination
Destination:		
	@A	JUMP to the last ACCEPT
	@M	JUMP to the next MATCH
	@P	JUMP to the next PROBLEM statement
	@%P:	JUMP to the last PROBLEM statement
	label	JUMP to a label or to a label represented by a string variable
LINK	L:	Chains from one program to another
Modifier:		
	X	Resets all variables to zero and releases all arrays and strings

Command	Syntax	Description
MATCH	M:	Tests student responses
Modifiers:		
	S	Ignore one or two wrong letters
	J	Jump to a destination
Features:		
	&	Match A and B
	%	Delimit the answer with spaces
	!	Match A or B
	*	Match any character in this position
PROBLEM	PR:	Marks the beginning of a section
Options:		
	U	Convert all student input to upper case
	L	Convert all student input to lower case
	S	Remove all blanks from student input
	G	Enable GOTO
	W	Discard the current label table
	E	Enable @ (Escape)
	M	Suppress the display of SDE messages
	N	None of the above options are desired
QUIT	QUIT:	Terminates the PILOT course
@QUIT	@QUIT:	Terminates the PILOT course, saving the current state of variables
REMARK	R:	Inserts comments into the program
RESTORE	RS:	Restores string and integer variables

Command	Syntax	Description
RUN TIME COMMANDS (Initiated by the student)		
GOTO @ (Escape) QUIT @QUIT		See individual tool names
TYPE	T:	Causes the text-field to be displayed at the terminal
Modifiers:	H	Suppress the line feed and (CR)
	S	Clears the screen, and places cursor at (1,1)
USE	U:	Transfers control to a subroutine

CONDITIONERS

Y	Execute if yes match
N	Execute if no match
n	Execute if there is a match with the answer counter
E	Execute if error condition is set
C	Execute if last relational expression is true

SYSTEM VARIABLES

%A	Answer counter
%B	Student answer buffer
%R	Status of last J, U, F, FI, FO, FX or RS: executed
%P	Statement number of last PR: executed
%L	Last label encountered or referred to
%U	Logon user-id

Computational Features Valid on C: or on Relational Expressions:

numeric variables, arrays, varying length strings
 + - * / ** !! = <> < > <= >= \$ ~ !
 ABS STR ASC CHR LEN INS

9 TABELLEN + CODES

ASCII-Code	9-20
Binär-Hex-Dezimal-Umrechnung	9-30
Monitor Flags bei NEXTDO	9-40
Symbolic Unit Designator, SUD	9-50
Simulated Option Switch (SIMOPTSW)	9-50
File Typen	9-50
Daten der verschiedenen Disc-Typen	9-60
Peripheral Type Codes	9-70
Dump-Adress-"Masstab"	9-100

Left Hex	Right Hex	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0010	2	␣	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
1000	8																
1001	9																
1010	A																
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

BINÄR-HEX-DEZIMAL-UMRECHNUNG

BINÄR	HEX	5	4	3	2	1
0000	0	0	0	0	0	0
0001	1	65536	4096	256	16	1
0010	2	131072	8192	512	32	2
0011	3	196608	12288	768	48	3
0100	4	262144	16384	1024	64	4
0101	5	327680	20480	1280	80	5
0110	6	393216	24576	1536	96	6
0111	7	458752	28672	1792	112	7
1000	8	524288	32768	2048	128	8
1001	9	589824	36864	2304	144	9
1010	A	655360	40960	2560	160	10
1011	B	720896	45056	2816	176	11
1100	C	786432	49152	3072	192	12
1101	D	851968	53248	3328	208	13
1110	E	917504	57344	3584	224	14
1111	F	983040	61440	3840	240	15

MONITOR FLAGS bei NEXTDO

MF	CC	Eintrag	Beschreibung
MF01	51	C	alle 30 MONITOR FLAGS auf ASCII-Null setzen
		<input type="checkbox"/>	die alten MONITOR FLAGS unverändert übernehmen
MF02	52	0	LOG-Eintrag normal vornehmen
		7	gewisse Logeintragungen unterdrücken
MF03	53	0	kein Programm-Halt
		1	Halt vor Programm-Start
		2	Halt nach Programm-Ende
		3	Halt vor Programm-Start und nach Programm-Ende
MF04	54	0	kein Memory Dump
		1	Memory Dump vor Programm-Start
		2	Memory Dump nach Programm-Ende
		3	Memory Dump vor Programm-Start und nach Programm-Ende
MF05	55	0	kein Memory Clear (des Virtuellen Memorys!)
		1	Memory Clear
		5	Memory Clear und B-Serie auf Simulated System Disc kopieren
MF06	56	0	keine Testhilfen
		C	COBOL DEBUG (COBUG)
		S	SYMBOLIC DEBUG (NEATUS)
MF07	57	0	Programm-Name beim Start auf den System Console Printer ausgeben
		1	Programm-Name nicht ausgeben
MF08	58	0	Programm normal beenden
		2	Aus ...AB wird ...ABJ
MF09	59	0	COBOL-Programm ohne CARTUNE laufenlassen
		C	COBOL-Programm mit CARTUNE laufenlassen
MF10	60		zurzeit noch frei

Die weiteren 20 MONITOR FLAGS sind frei für den Anwender.

SYMBOLIC UNIT DESIGNATOR (SUD)

D00	System-Disc
D01-D89	User-Discs
D90-D99	CPL-Discs
I01-I99	Console I-O-Devices
M01-M99	Magnetic Tapes
R01-R99	Remote Peripherals
P01-P09	Printers
P11-P19	Card Readers
P21-P29	Card Punchs
P31-P39	Paper Tape Readers
P41-P49	Paper Tape Punchs
P51-P59	Optical Character Readers
P61-P69	MICR
A01-AFF	Adapters

SIMULATED OPTION SWITCH (SIMOPTSW)

Bit	Hex Eintrag	Funktion
1	01	nicht verwendet
2	02	Halt nach laufendem Programm
3	04	Memory ausdrucken (Virt. Memory)
4	08	nicht verwendet
5	10	Ausdruck des PAL Status Report
6	20	Control-String Abbruch
7	40	Close aller eröffneten BAT-Files
8	80	nicht verwendet

FILE TYPEN

DATA	0	Daten File
SRCE	1	Source Programm
OBJT	2	Object Programm
CNTL	6	Control String
	7	Catalog Files

DATEN DER VERSCHIEDENEN DISC-TYPEN

DATEN DER VERSCHIEDENEN DISC - TYPEN

Disc-Typ	Zyl. pro Disc	Spur. pro Zyl.	Spuren pro Disc	Sekt. pro Spur	Sekt. pro Zyl.	Sektor. pro Disc	Bytes pro Sekt.	Bytes pro Spur	Bytes pro Zylind.	Bytes pro Disc (User)	Disc Pack Code	Höchs. EDA-Modul	ØZugr. Zeit
6591-0101	348	12	4 176	5	60	20 880	1322	6 610	79 320	27 603 360	81	00	35.1
6591-0101	348	12	4 176	12	144	50 112	512	6 144	73 728	25 657 344	22	00	35.1
6591-0201	696	12	8 352	5	60	41 760	1322	6 610	79 320	55 206 720	81	00	35.1
6591-0201	696	12	8 352	12	144	100 224	512	6 144	73 728	51 314 688	22	01	35.1
6580	404	19	7 676	8	152	61 408	1322	10 576	200 944	81 181 376	81	00	38.3
6580/EDA	404	19	7 676	9	171	69 084	1322	11 898	226 062	91 329 048	81	01	38.3
6580/EDA/DD	808	19	15 352	9	171	138 168	1322	11 898	226 062	182 658 096	81	02	38.3
6580/EDA	404	19	7 676	20	380	153 520	512	10 240	194 560	78 602 240	22	02	38.3
6580/EDA/DD	808	19	15 352	20	380	307 040	512	10 240	194 560	157 204 480	22	04	38.3
656	398	2	796	12	24	9 552	512	6 144	12 288	4 890 624	22	00	47.5
6531	817	1	817	32	32	26 144	512	16 384	16 384	13 385 728	22	00	38.3
6530-0201	817	3	2 451	32	96	78 432	512	16 384	49 152	40 157 184	22	01	38.3
6530-0301	817	5	4 085	32	160	130 730	512	16 384	81 920	66 928 640	22	01	38.3
6540	817	10	8 170	32	320	261 440	512	16 384	163 840	133 857 280	22	03	38.3
6550	833	40	33 320	16	640	533 120	1024	16 384	655 360	545 914 880	40	08	33.3

Alle Werte: User - Grössen (excl. Alternate Tracks)

PERIPHERAL TYPE CODES

Unit Description	Unit Type Code	File Spec. Type Code
Punched Card Units		
684-101 Reader/Punch	07	01
684-301 Punch	09	03
688-102/111 Reader/Punch	01	01
686-201 Reader	02	02
686-302/311 Punch	03	03
6831-0201/0301 Reader (integrated)	04	02
Punched Paper Tape Units		
660-101 Reader	11	11
665-101 Punch	12	12
6640-0101 Reader/Punch	13	10
Disk Units		
655-201 4 MB	32	32
656-102 5 MB Fixed Disk	35	36
656-102 5 MB Removable Disk	36	36
657-101/102 30/56 MB	37	37
658-201 100 MB	3C	37
658-201 200 MB	3A	37
658-201 100 MB IDC	3E	36
658-201 200 MB IDC	38	38
658-201 35/70 MB IDC	39	38
6590-0101 Removable Disk	81	37
6530 Fixed Disk	81	37
6530 Fixed Disk	82	37
6540 Fixed Disk	83	37
6550 Fixed Disk	85	37
6580 100 MB	86	37
6580 200 MB	87	37
Magnetic Tape — 9 Channels		
633-111 50 ips Phase	47	40
633-119 50 ips NRZI	45	45
633-121 50 ips Phase	43	43
633-211 90 ips Phase	41	41
90 ips Phase	44	44
633-271 150 ips Phase	44	44
1-11-111	42	42

PERIPHERAL TYPE CODES

Unit Description		Unit Type Code	File Spec. Type Code
Magnetic Tape — 9 Channel (Continued)			
634-100/110	25 ips NRZI	48	45
634-109/119	25 ips Phase w/o Dual Mode	48	40
634-109/119	25 ips Phase with Dual Mode	4A	40
634-109/119	25 ips NRZI with Dual Mode	4A	45
634-209/219	50 ips Phase w/o Dual Mode	49	40
634-209/219	50 ips Phase with Dual Mode	48	40
634-209/219	50 ips NRZI with Dual Mode	48	45
635-109	100 ips Phase with Dual Mod	4C	40
635-109	100 ips NRZI with Dual Mode	4C	45
635-209	200 ips Phase with Dual Mod	4D	40
635-209	200 ips NRZI with Dual Mode	4D	45
6325-0101/0201	45 ips Phase and NRZI	50	40
6340-0205/0215	50 ips Phase	49	40
6370-0401	75 ips Phase and GCR	51	40
6370-0601	125 ips Phase and GCR	51	40
6370-0801	200 ips Phase and GCR	52	40
6373	25-75 ips Phase and GCR	51	40
Magnetic Tape — 7 Channel			
633-117	50 ips NRZI	4E	48
634-107/117	25 ips NRZI	4F	48
Printers — Standard Fonts			
640-200	Single-Numeric, 132 Column	62	62
640-200	Double-Numeric, 132 Column	63	63
640-210	Single-Numeric, 160 Column	64	64
640-210	Double-Numeric, 160 Column	65	65
640-300	128 Characters, 132 Column	66	66
645-100	(International only)	67	67
646-201	1200 lpm Train, 132 Column	68	68
647-201	2000 lpm Train, 132 Column	69	69
649-300	Low-Speed, 132 Column	68	62
6420 ALL	Band, 132 Column	6A	62
6471	2000 lpm Band, 132 Column	6A	62
Channel I/O Link (CIOL) Device			
6480	Laser Printer	AO AE	N A

PERIPHERAL TYPE CODES

Unit Description		Unit Type Code	File Spec. Type Code
MICR Units			
670	MICR Sorter	85	N/A
675	MICR Sorter	85	N/A
6770	MICR Sorter	87	N/A
6780	MICR Sorter	88	N/A
Tape Cassettes/Flexible Disks			
636-201/301	Cassette Unit	91	45
658	Controller Log Cassette	83	37
7649	Cassette/Flexible Disk	92	45
7649	General Purpose Peripheral Controller	90	45

DUMP-«MASSTAB»:

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
0
1
2
3

10-05

10 IVS

SCL	10-10
USC	10-100
Interactive Utilities	10-120
EDIT	10-190
NOTE	10-260
IDS	10-310
VRX Protection Data Base	10-370

Ausführliche und massgebende Information in:

-RM-237-V11 IVS Terminal User Information
-RM-238-V11 VRX System Administration

Stand: V11/07.85

VRX/IVS SYSTEM COMMAND LANGUAGE

BEFEHLSARTEN

Definitionen

DEfine
LOcal
REdefine

Control

BYe
GOto
MCibegin
POp
PUsh
RESUme
RETurn
SET
SIGNAL
STop

File Control

CHange
CREatecat
DELEte
FILECreation
FILELocation
FILENode
Llist
NEWFilenode
REMove

Interactive

ACcept
DEPot
NEWPassword
SENd

Job Control

ABort
DEBug
DISplay
EXecute
KILL
QUEue
SAve
SPool
SUBmit

Zuordnen von Arbeitsmitteln

ASSign
CANcel
FRee
REConfigure

Zusammensetzung

BEgin
ENd

IF
 Then
 ELse
IFEnd

Procedure Definitionen

PRocedure
PROCENd

EINFACHE BEFEHLE

Befehl	Bedeutung					
$\text{ABort } \left\{ \begin{array}{l} \text{job-number} \\ \text{job-descriptor-reference} \\ \text{MYJob} \end{array} \right\} \left\{ \begin{array}{l} \text{'string-expression'} \\ \end{array} \right\};$	Abnormales Abbrechen eines Job's					
$\text{ACcept string-variable};$	Dateneingabe über ein IVS-Terminal in ein mit STRING definiertes Feld.					
$\text{ASSign } \left\{ \begin{array}{l} \text{object-descriptor-reference} \\ \text{object-type [object-id] (parameter,...)} \end{array} \right\} \text{ ,...};$						
where <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td rowspan="3" style="padding: 0 5px;">object-type</td> <td style="padding: 0 5px;">File</td> <td rowspan="3" style="padding: 0 5px;">}</td> </tr> <tr> <td style="padding: 0 5px;">UNit</td> </tr> <tr> <td style="padding: 0 5px;">PRInter FLags</td> </tr> </table>	object-type	File	}	UNit	PRInter FLags	Zuordnen von Dateien Peripherie und Flags für eigene Session oder für Batch Job's die submitted werden.
object-type		File		}		
		UNit				
	PRInter FLags					
BYE [:message];	Job Beendigung.					
CANcel:	Aufheben von Pending Object-Assigns					
$\text{CHange old file-locator TO new file-locator;}$						
where <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td rowspan="3" style="padding: 0 5px;">file-locator</td> <td style="padding: 0 5px;">catalog-filename</td> <td rowspan="3" style="padding: 0 5px;">]</td> </tr> <tr> <td style="padding: 0 5px;"># file-reference</td> </tr> <tr> <td style="padding: 0 5px;">file-descriptor</td> </tr> </table>	file-locator	catalog-filename]	# file-reference	file-descriptor	Ändern des Dateinamens einer katalogisierten oder nicht katalogisierten Datei.
file-locator		catalog-filename]		
		# file-reference				
	file-descriptor					

Befehl	Bedeutung
CR reatecat subcatalog-name;	Erstellen eines neuen Subkataloges.
DEB ug [VRC-module-name] [TAsk=task-name] ;	Aktivieren des Interaktiven Debug Systems.
DE fine { variable-id: data-type reference-id: SCL-object-type } ,...;	
where data-type { INTEger STRing BOolean NAme }	Definieren für alle Procedure Levels, die in dieser Session aktuell sind.
and SCL-object-type { FLie FLags JOB OPtion PRInter PROCEs STation UNit }	
DE lete { file-locator catalog-identifier ALI } ;	Löschen einer Datei, eines Subkatalogs, oder einer ganzen Platte.
where catalog-identifier { catalog-name [= [level]] * [= [level]] }	
and unit-locator { volume-serial-number unit-descriptor unit-descriptor-reference }	

Befehl	Bedeutung
DEP ot { * () (station-attribue = expression,...) } ;	Anzeigen und Ändern der MYSTATION-Charakteristiks.
Dis play [job-locator] { [ALI] system-info } ,... { session-info }	
TO { JD CT DAyfile TErminAl } ,...;	Anzeigen von Informationen über das System, die Session oder den Job.
where job-locator { job-number job-descriptor-reference MYJob }	
and system-info { ACTive HelD JOBS MEssages PENding PERipherals SPOoled TAskS }	
and session-info { :SCL-object-type :SYMBOLs :object-descriptor-reference :expression }	

Befehl	Bedeutung
[EXecute] process-name [(parameter,...)]	
$\left[\begin{array}{l} \text{file-descriptor} \\ \text{flags-descriptor} \\ \text{option-descriptor} \\ \text{MCL-object-descriptor} \\ \text{SPEC = string} \end{array} \right];$	Starten von Prozeduren, VRX-Load-Modulen und MCL-Control-Strings.
where $\left. \begin{array}{l} \text{procedure-name} \\ \text{control-string} \\ \text{SCL-procedure-file} \\ \text{VRX-load-module} \end{array} \right\}$	
FILECreation $\left\{ \begin{array}{l} \text{CAT} \\ \text{NOCat} \end{array} \right\};$	Festlegen des Defaultwertes, ob eine Datei im Katalog oder außerhalb des Kataloges erstellt werden soll.
FILELocation $\left\{ \begin{array}{l} \text{CAT} \\ \text{NOCat} \\ \text{NOCat/CAT} \\ \text{CAT/NOCat} \end{array} \right\};$	Festlegen der Suchreihenfolge für eine bestehende Datei.
FILENode $\left\{ \begin{array}{l} \$ \\ \text{USER} \\ \text{subcatalog-name} \end{array} \right\};$	Angabe oder Anzeige des aktuellen Subkatalog-Begriffes.
FR ee identifier,...;	Aufheben von Zuteilungen.
GO to label;	Verzweigen an einen Ansprungspunkt innerhalb einer Prozedur.

Befehl	Bedeutung
KILL job-locator [:message];	Abbruch eines Job's alle Benutzerspooldateien des Job's werden gelöscht.
LIS t $\left\{ \begin{array}{l} \text{file-locator} \\ \text{catalog-identifier ALI} \\ \text{unit-locator ALI} \end{array} \right\}$ DETA il = $\left\{ \begin{array}{l} \text{Abbr} \\ \text{Cat} \\ \text{Full} \\ \text{Lm} \\ \text{Term} \end{array} \right\} \dots;$	Auflisten des Inhalts eines Kataloges einer Backup Datei oder des Directories.
LO cal $\left\{ \begin{array}{l} \text{variable-id: data-type} \\ \text{reference-id: SCL-object-type} \end{array} \right\} \dots;$	Definieren von Feldern oder Deskriptoren innerhalb einer Prozedur.
MCI begin;	Ab dem Erscheinen dieses SCL-Befehls erwartet VRX MCL-Parameter.
NEW Filenode $\left\{ \begin{array}{l} * \\ \$ \\ \text{subcatalog-name} \end{array} \right\};$	Anzeigen des Filenode-Wertes im aktiven User-Profil-Satz bzw. Ändern dieses Wertes im User-Profil.
NEW Password $\left\{ \begin{array}{l} \text{B, no-of-uses} \\ \text{I} \end{array} \right\};$	Ändern des Interaktive-Passwort oder/und das Batch-Passwort im User-Profil.
PO p;	Wiedereinsetzen der mit PUSH sichergestellten Session-Parameter.

Befehl	Bedeutung
PU sh;	Stellt die durch die SET-Anweisung veränderbaren Session-Parameter sicher.
QUE ue [printer-reference] ;	Schließt eine Druckdatei und überträgt sie in die Liste der zu druckenden Dateien.
REC onfigure [(memory-parameter,...)] ;	Anzeigen oder Ändern von Memory-Parametern für einen Job.
RE define { variable-id:data-type reference-id: SCL-object-type } ,...;	Ändern einer vorher mit DEFINE definierten Variablen.
REM ove subcatalog-name;	Löschen eines Subkataloges ohne gültige Eintragungen aus dem FC.
RESU me;	Weiterfahren, wo vorher mit BREAK unterbrochen wurde.
RET urn (Status = { ABorted ACTive Completed Null }) [:message] ;	Verlassen einer aufgerufenen Prozedur und zurückkehren zur aufrufenden Prozedur.

Befehl	Bedeutung
SA ve [printer-reference TO] file-locator;	Abschließen einer Spool-Datei und übertragen des Inhalts in eine CAM-Datei.
SEN d { user-id [user-id] /terminal /SYSTOPerator } :message;	Senden einer Meldung zum System-Operator oder einem anderen IVS-Terminal.
SET [([session-parameter-value = expression,...])];	
where session-parameter-value { AUToterm LEft LIst RIght }	
SI gnal;	Anzeigen oder Verändern der aktuellen Session-Parameter oder Zurücksetzen auf ihren Default-Wert.
SI gnal;	Weiterfahren wo vorher mit BREAK unterbrochen wurde und setzen eines Signals.
SP ool file-locator [TO printer-reference] ;	Übertragen des Inhalts einer CAM-Datei in eine Spool-Datei.

Befehl	Bedeutung
STop ;	Zurückgehen zur SCL nach einem BREAK und Abschließen des unterbrochenen Prozesses.
SU bit { BEGIN-group process-name [(parameter,...)] file-locater	
[W ith [job-Reference job-attribute-list] [PO st post-count]	
[AF ter wait-id [/T] ,...] ;	Starten von CATJOBS und Prozeduren außerhalb der Session.

ZUSAMMENGESetzte BEFEHLE

BE gin	BE und EN werden benutzt, mehrere SCL-Befehle als Gruppe zu kennzeichnen.
[\$label] ... statement-1 [statement-2] • • [statement-N]	
EN d;	
IF conditon TH en [statement] ...	Abfrage nach Konditionen. Aufgrund des Ergebnisses können unterschiedliche SCL-Befehle durchlaufen werden.
[EL se [statement] ...]	
IF End;	

Befehl Bedeutung

PROCEDURE DEFINITION

PR ocedure [proc-name] [(proc-argument,...)];	
statement-1	
[statement-2]	
•	
•	
[statement-N]	
PROC end [proc-name] ;	Prozeduren sind eine Folge von SCL-Anweisungen, die in einer Datei (Type = 6) enthalten sind. Bei Bedarf können beim Start Variable angegeben werden.

USC BEFEHLE

:AB ort { job-number MY Job }	Abnormales Abbrechen der Terminal-Session oder eines Job's mit gleicher USERID.
:AN swer message-number [.response]	Beantworten einer Meldung des Systems, dieser Session, oder eines Batch Job's.
:BR eak	Unterbrechung der Session.
:CO nverse job-id, { OP TE rm }	Ändern der Zuteilung von pending oder späteren Nachrichten zur System-Konsole oder dem IVS-Terminal.
:DI splay [job-number] [ALL] system-info,...	Anzeigen von Meldungen der Session, oder zugeordneter Jobs, ebenso von Job-Status-Meldungen, Peripheriezuteilungen und Spoolfile Informationen.
[TO { JD CT DA yfile TE rminal }]	
:RE Init	Beenden der laufenden Session und automatisches Starten einer neuen.
:RO ute { job-id, TE rm message-number, OP }	Ändern der Zuteilung einer einzelnen Nachricht zur System-Konsole oder dem IVS-Terminal.

SYSTEM MERKMALE

MYFlags	MYJob	MYStation
----------------	--------------	------------------

Liste der möglichen JOB-Parameter

Acct	JOBName	PROJect
B2recsize	JOBNUmber	PROTectdsa
CLass	JPTy	Size
CPty	Localdsa	STACksize
DELetedigits	Maxt	STATus
DEST	Name	Userid
INteractive	OptS	Vmaxmem
IOct	PAssword	Wsest
JOBIId	PROGOvlys	

Liste der möglichen PRINT-Parameter

Copy	NAME	Sud
Dest	NOheaders	TApe
Format	Overflow	TRainno
Hold	Pages	TYpe
Keep		

Liste der möglichen STATION-Parameter

BACKspace	Length	Termtype
BReark	Ovmode	UNSolicited
Delete	Prompt	WAit
Experience	Routechar	Width
Foldmark		

MNEMONIC-Definitionen

Drop	NOTify
Expert	NOVice
Fold	SCreen
HCopy	SHow
HOLD	TRunc

**INTERACTIVE
UTILITIES****INTERACTIVE UTILITY Reservierte Worte**

All	In
BLocks	New
BY	Out
Cc	Possible
EXecute	Records
FORmat	To
FRom	Unit

FILE CATALOG Symbole

\$ Dateien im ersten Level des Katalogs

* Dateien im ersten Level des Home-Katalogs

\$ = Alle Dateien auf allen Stufen des Katalogs

† = Alle Dateien auf allen Stufen des Home-Katalogs

Allgemeines IUP-Format

backup-file-locator := $\left\{ \begin{array}{l} \text{file-descriptor-parameters} \\ \# \text{file-descriptor-reference} \end{array} \right\}$

catalog-identifier := $\left\{ \begin{array}{l} \text{subcatalog-name} \\ \$ \\ * \end{array} \right\} \left[= \left\{ \begin{array}{l} \text{space} \\ \text{integer} \end{array} \right\} \right]$

file-locator := $\left\{ \begin{array}{l} \text{cataloged-file-name} \\ \text{file-descriptor-parameters} \\ \# \text{file-descriptor-reference} \end{array} \right\}$

file-name := $\left\{ \begin{array}{l} \text{new-file-name} \\ (\text{NAME}=\text{new-file-name}) \\ \# \text{file-descriptor-reference} \end{array} \right\}$

format-option := $\left[\text{FORmat-} \left\{ \begin{array}{l} \text{B} \text{Locks} \\ \text{C} \text{c} \end{array} \right\} \right]$

list-options := $\left[\text{DET} \text{ail} = \left\{ \begin{array}{l} \text{Lm} \\ \text{Cat} \\ \text{Full} \end{array} \right\} \dots \right]$

location-attribute-list := $\left\{ \begin{array}{l} \text{six-digit-disk-volume-number} \\ \text{physical-location-attribute-list} \\ \text{logical-location-attribute-list} \end{array} \right\}$

object-limiter := $\left[\left\{ \begin{array}{l} \text{R} \text{ecords} \\ \text{B} \text{Locks} \end{array} \right\} = \left\{ \begin{array}{l} n \\ n/n \\ n/* \end{array} \right\} \right]$

physical-location-attribute-list :=

$\left(\left(\left[\begin{array}{l} \text{S} \text{XN} \\ \text{S} \text{E} \text{C} \text{T} \text{I} \text{O} \text{N} \\ \text{A} \text{S} \text{S} \text{O} \text{C} \text{I} \text{A} \text{T} \text{I} \text{O} \text{N} \end{array} \right] \left\{ \begin{array}{l} \text{S} \text{U} \text{D} \\ \text{V} \text{O} \text{L} \text{U} \text{M} \text{E} \text{N} \text{O} \end{array} \right\} \left[\text{A} \text{L} \text{L} \text{O} \text{C} \text{B} \text{O} \text{U} \text{N} \text{D} \text{S} \right] \right) \right)$

logical-location-attribute-list :=

$\left\{ \begin{array}{l} \$ \\ \text{catalog name} \end{array} \right\} \left[\text{physical-location-attribute-list} \right]$

unit-locator := $\left\{ \begin{array}{l} \text{disk-volume-number} \\ \# \text{unit-descriptor-reference} \\ \text{unit-descriptor} \end{array} \right\}$

INTERACTIVE UTILITY LADEMODULE

(Abrufbar mit EXECUTE, können während eines BREAK nicht abgerufen werden.)

Befehl	Bedeutung
BACKUP	Sicherheitskopie von Disk-File(s) auf Band.
Einzelne Datei:	
BACKUP;	
file-locator IN backup-file-locator;	
\$.THOMAS.SPECIAL IN (NAME=MYBACKUP, SUD=M01);	
Dateiengruppe:	
BACKUP;	
unit-locator	
catalog-identifier	All In backup-file-locator;
500103 ALL IN (NAME=BU1000, SUD=M01);	
\$.USER ALL IN (NAME=BACKUP, SUD=M01);	
COPY	Kopieren von Disk-Files CAT und NOCAT.
Einzelne Datei:	
COPY;	
file-locator [Out] To location-attribute-list	
$\left[\text{New} \left(\left(\left(\text{NAME}=\text{x} \dots \text{x} \right) \right) \left(\text{BLOCKSIZE}=\text{x} \dots \text{x} \right) \right) \right];$	
"CAT TO CAT"	\$.ROI.PROC TO \$.STP;
"CAT TO NOCAT"	\$.ROI.PROC OUT TO 500100;
"NOCAT TO CAT"	(NAME=PROC, SUD=D02) TO *;
"NOCAT TO NOCAT"	(NAME=PROC, SUD=D02) TO 500100;

Befehl

Bedeutung

COPY

Dateiengruppe:

```

COPY;
{ unit-locator
  catalog-identifier } All [Out] To location-attribute-list;
"CAT TO CAT"      $.ROI ALL TO $.STP;
"CAT TO NOCAT"    $.ROI ALL OUT TO 500100;
"NOCAT TO CAT"    500100 ALL TO *;
"NOCAT TO NOCAT" 500100 ALL TO 500522;

```

COMPARE

Vergleichen (Byte um Byte) zwischen dem Inhalt zweier Dateien, Anzeigen von Unterschieden.

```

COMPARE;
file-locator [object-limiter] To

```

```

file-locator [object-limiter] [BY BLocks];

```

```

(NAME=$.SUB.WA,AGE=3) TO (NAME=$.SUB.WB,AGE=NEW);
(NAME=WA,SUD=D01) RECORDS=210/* TO
(NAME=WB,SUD=D01) Records=210/*;

```

PRINT

Ausgeben des Inhalts einer Plattendatei (Dump).

```

EXECUTE PRINT;
file-locator [object-limiter] [format-option];

```

```

(NAME=$.STP.PROC,AGE=NEW);

```

Befehl

Bedeutung

RETRIEVE

Zurückkopieren von Band-Dateien auf Platte (nach BACKUP).

Einzeldatei, nicht im Katalog:

```

RETRIEVE;
file-locator FRom backup-file-locator

```

```

New { (physical-location-attribute-list) } ...;
      { (NAME=new-file-name) }

```

```

(NAME=P1) FROM (NAME=BU,SUD=M01)

```

```

NEW (NAME=P2,SUD=D02);

```

Einzeldatei, katalogisiert:

```

RETRIEVE;
file-locator FRom backup-file-locator

```

```

[New (physical-location-attribute-list)
 (NAME=unqualified-file-name)] ;

```

```

$.STP.PROC FROM #BACKUP NEW (NAME=PROCA);

```

Dateiengruppe, nicht im Katalog:

```

RETRIEVE;
unit-locator All FRom backup-file-locator

```

```

[New (ALLOCBOUNDS=n/n/n)];

```

```

500100 ALL FROM (NAME=BACKUP, SUD=M01);

```

Dateiengruppe, katalogisiert:

```

RETRIEVE;
catalog-identifier All FRom backup-file-locator

```

```

[New physical-location-attribute-list];

```

```

USER ALL FROM (NAME=BU,SUD=M02)

```

```

NEW (SUD=D02, ALLOCBOUNDS=/ /2/2);

```

SCL INTERACTIVE UTILITY BEFEHLE**

(Direkt aufrufbar; auch während BREAK)

Befehl	Bedeutung
CHANGE	Ändern eines Dateinamens
CH ange file-locator TO file-name;	
CHANGE (NAME=YOURFILE,SUD=D02) TO NEWNAME;	
CHANGE WORKA TO WORKB;	
CHANGE \$.SUBCAT.WORK TO \$.SUB.WORK;	
DELETE	Löschen von einem oder mehreren File(s), CAT oder NOCAT.
Einzeldatei:	
DE lete file-locator;	
DELETE \$.STP.PROC;	
Dateiengruppe nicht im Katalog:	
DE lete unit-locator ALL ;	
DELETE UNIT(SUD=D02) ALL;	
Dateiengruppe katalogisiert:	
DE lete	$\left\{ \begin{array}{l} \$ \\ \text{subcatalog-name} \\ * \end{array} \right\}$ ALL ;
DELETE \$.SUBCAT ALL;	

Befehl	Bedeutung
LIST	Liste aller Dateien aus Disk-Directory, Datei-Katalog oder Backup-Band-Datei
Auflisten eines Directories:	
L ist unit-locator [ALL] [list-options];	
LIST 500103 DETAIL=FULL;	
Auflisten Datei-Katalog:	
L ist catalog-identifier [(ALL)] [list-options]	
[with Volumeno disk-volume-number];	
LIST \$.STP ALL WITH VOLUMENO 500147;	
Auflisten Backup-Band-Datei:	
L ist backup-file-locator [ALL];	
LIST (NAME=BACKUP,SUD=M01);	
REMOVE	Löschen des Basis-Kataloges \$ oder eines Sub-Kataloges. Der FC bzw. Sub-Katalog muß leer sein.
RE Move subcatalog-name;	
REMOVE \$.STP;	

VRX/IVS EDIT BEFEHLS-TYPEN

SYSTEM-FUNKTIONEN

Create
Edit
Finish
Print
Recover

Editor

Satz auffinden

Programmierhilfe

Down
First
Last
Locate
Pageline
Query
Reference
Up

Exit
Gauge
Id
Noprint
Tabs
Version

Satz ändern

Löschen oder ein-
fügen von Sätzen

Modify
Zape

Add
Copy
Join
Omit
Sequence
Xomit

VRX/IVS EDIT

Unterstützte Sprachen

Angabe

Bedeutung

BASIC	(VRX Basic)
COBOL	(COBOL Compiler)
COBOL 1	(Century '68 stage one COBOL)
COBOL 2	(Century '68 stage two COBOL)
COBOL 3	(Century '68 stage three COBOL)
COBOL 4	(VRX '74 and Century '74 COBOL Compiler)
DATA	(keine Sprache)
FORT/B	(Fortran Basic)
FORT/F	(Fortran Full)
FORT/I	(Fortran Intermediate)
FORT/V	(VRX Fortran)
HEADCS	(Control String)
NCRL	(VRX ISWL)
NEAT/3	(Compiler)
NEAT/C	(Assembler)
NEATVS	(Compiler)
PROC	(Procedure file)

BEFEHLE †

† Wenn keine weiteren Parameter vorhanden sind, werden keine Begrenzungszeichen benötigt.

Befehl	Bedeutung
Add [image], $\left[\begin{array}{c} \{D\} \\ \{R\} \end{array} \right]$	Einfügen eines Satzes Ersetzen eines Satzes Einfügen mehrerer Sätze
Copy, filename [version],[sud], [range-begin], $\left[\begin{array}{c} \{range-end\} \\ * \end{array} \right]$	Kopiert Teile eines Programms in das Workfile
Create, filename, language, [id],[R]	Erstellen eines neuen Ursprungsprogramms, Prozedur, Controlstring oder Datendatei
Down, [integer], $\left[\begin{array}{c} \{integer\} \\ * \end{array} \right]$	Positioniert und zeigt einen oder mehrere Sätze nach dem laufenden
Edit, filename [version],[sud],[id], [R], $\left[\begin{array}{c} \{S\} \\ \{P\} \\ \{C\} \\ \{D\} \end{array} \right]$	Editieren eines bestehenden Ursprungsprogramms, Prozedur, Controlstring oder Datendatei
Exit, $\left\{ \begin{array}{l} \text{Save} \\ \text{Keep} \\ \text{Update} \\ \text{FORGET} \end{array} \right\}$ [sud], $\left[\begin{array}{c} \{ANY\} \\ * \end{array} \right]$, [D]	Abschluß des Editierens
Finish	Abschluß von EDIT
First $\left[\begin{array}{c} \{integer\} \\ * \end{array} \right]$	Aufzeigen des ersten Satzes und weiterer

Befehl	Bedeutung
Gauge	Zeigt einen Raster am Bildschirm
Id [,value]	Anzeigen oder Ändern der Identifikation
Join, range-begin $\left[\begin{array}{c} \{range-end\} \\ * \end{array} \right]$	Verschieben von Sätzen innerhalb der Datei
Last	Positionieren und zeigen des letzten Satzes einer Datei
†† Locate,/string/ $\left[\begin{array}{c} \{integer\} \\ * \end{array} \right]$	Sucht alle Sätze nach dem Current, bis er die Zeichenfolge /string/ findet und zeigt die Sätze mit diesem Inhalt
[Modify,]/string-A/[string-B] / [occurrences] $\left[\begin{array}{c} \{integer\} \\ * \end{array} \right]$ [I/D]	Ändern der Zeichenfolge /string-A/ in /string-B/ in einem oder mehreren Sätzen
Noprint, $\left\{ \begin{array}{l} \{P\} \\ \{I\} \end{array} \right\}$, $\left\{ \begin{array}{l} \{+\} \\ \{-\} \end{array} \right\}$	Spezifiziert, daß die Seiten-Zeilen oder die Identifikation gezeigt oder nicht gezeigt werden sollen.
Omit, [range-begin],[range-end]	Löschen des laufenden Satzes oder einer Satzgruppe
Pageline, line number $\left[\begin{array}{c} \{integer\} \\ * \end{array} \right]$	Aufzeigen von Sätzen auf Grund der Seiten-Zeilenummer
Print, filename [version],[sud], $\left[\begin{array}{c} \{S\} \\ \{P\} \\ \{C\} \\ \{D\} \end{array} \right]$	Ausdruck einer bestehenden Datei oder eines Teiles davon

Befehl	Bedeutung
Query $\left[\left\{ \begin{array}{c} \text{integer} \\ * \end{array} \right\} \right]$	Zeigen des laufenden Satzes und weiterer
Recover, filename [,sud]	Kopiert eine Datei von der FILE RECOVERY AREA in die Workfile zum editieren
Reference, tag, $\left[\left(\begin{array}{c} U \\ D \end{array} \right) \right], \left[\left\{ \begin{array}{c} \text{integer} \\ * \end{array} \right\} \right]$	Sucht einen Satz mit dem Reference Namen-tag und stellt ihn zur Verfügung
Sequence, range-begin, $\left\{ \begin{array}{c} \text{range-end} \\ * \end{array} \right\}$ [increment],[newbase]	Ordnet einer Gruppe Ursprungszeilen neue Seiten-Zeilen-Nummern zu
$\left\{ \begin{array}{l} \text{Tabs [, setting [, setting...]] } \\ \text{Tabs [, character] } \end{array} \right\}$	Zeigen und Ändern der Tabulatorpositionen und des Tabulatorzeichens
Up, [integer], $\left[\left\{ \begin{array}{c} \text{integer} \\ * \end{array} \right\} \right]$	Positioniert und zeigt mehrere Sätze vor dem laufenden
Version [, number]	Anzeigen und verändern der Versionsnummer der gegenwärtig editierten Datei
Xomit, range-begin [, [range-end]]	Reaktivieren der vorher durch OMIT gelöschten Sätze im gleichen Lauf
†† Zapp, column-number, / string / $\left[\left\{ \begin{array}{c} \text{integer} \\ * \end{array} \right\} \right]$	Ersetzen von Zeichen in einem oder mehreren Sätzen

†† Muß am Anfang und Ende die Begrenzungszeichen für den String-Parameter haben.

EDIT PARAMETER WERTE

Parameter	Bedeutung
/	: Vom Benutzer vergebenes Begrenzungszeichen, 1-stellig
*	: Vom laufenden Satz bis zum Ende
character	: 1-stelliges Tabulatorzeichen (darf keine Ziffer, Punkt, Semicolon, Schrägstrich oder Komma sein)
image	: Einzufügender Satz (Default ist Automatischer Text Add Mode)
increment	: $\left\{ \begin{array}{c} <\text{leer}> \\ 1...999 \end{array} \right\}$
integer	: $\left\{ \begin{array}{c} <\text{leer}> \\ 1...999 \end{array} \right\}$
newbase	: $\left\{ \begin{array}{c} <\text{leer}> \\ 1...999999 \end{array} \right\}$
occurrences	: $\left\{ \begin{array}{c} <\text{leer}> \\ 1...25 [, 1...25,...] \end{array} \right\}$
range-begin and range-end	: Müssen Seiten-Zeilenummer sein
setting	: $\left\{ \begin{array}{c} <\text{leer}> \\ 1...80 [, 1...80,...] \end{array} \right\}$ Achtung: Die Werte müssen in steigender Folge sein

EDIT PARAMETER WERTE

Parameter	Bedeutung
string	: N-stellige Zeichenfolge
string-A	: Zeichenfolge die ersetzt werden muß
string-B	: A-String ersetzende Zeichenfolge
sud	: { <leer> D01...D99 CAT }
tag	: N-stelliges Merkmal
value	: { 1 to 8 Zeichenfolge 1-8stellig Leerzeichen <leer> }

NOTE ATTRIBUTE MENU

Menu-Wort & Default Wert	Möglichkeiten
A. AGE = NEW	O oder N oder R
B. FN =	Dateiname
C. CAT = SESSION DEFAULT	Y or N
D. VOL = D01	volume or sud
E. FORMAT = SPUR	SPUR or TEXT
F. LANG = COBOL	B, C, C0, C6, C7, CS, DATA, DOC, I, J, N, NC, NF, VF, P
G. DATING = GEN	GEN or DATE
H. CDATE = "current"	date in DDMMYY form
I. SEQ = Y	Y or N
J. WFLN = 80	höchstens 240
K. WFIZE = 5000	höchstens 99999 Sätze
L. LINEOP = S	S
M. DISPLAY MENU	
N. CONTINUE	

Mögliche Sprachen

Sprache	Eintrag	Seiten-Zeilen- Nummer
VRX BASIC	B	1 bis 5
VRX/NCS COBOL-74	C	1 bis 6
NCS COBOL-68	C0, C6, C7	1 bis 6
CONTROL STRING	CS	1 bis 6
DATA (Unsequenced Text)	DATA	keine
DOCUMENT	DOC	1 bis 8
JOB	J	1 bis 6
NCS FORTRAN	NF	75 bis 80
VRX FORTRAN	VF	75 bis 80
ISWL or NCRL	I	1 bis 6
NEAT/3 or NEAT/VS	N	1 bis 6
PROCEDURE	P	1 bis 6
NEAT/C	NC	1 bis 6

Syntax zum Auffinden eines Satzes

Einzelner Satz

$$\left[\begin{array}{l} \langle \text{leer} \rangle \\ * \\ + \text{ crp rel \#} \\ - \text{ crp rel \#} \\ \text{line \#} \\ \text{First} \\ \text{Last} \end{array} \right]$$

Mehrere Sätze

$$\left[\begin{array}{l} \text{ALL} \\ \left[\begin{array}{l} \langle \text{leer} \rangle \\ * \\ + \text{ crp rel \#} \\ - \text{ crp rel \#} \\ \text{line \#} \\ \text{First} \end{array} \right] / \left[\begin{array}{l} \langle \text{leer} \rangle \\ * \\ + \text{ crp rel \#} \\ - \text{ crp rel \#} \\ \text{line \#} \\ \text{Last} \end{array} \right] \end{array} \right]$$

Mehrere Sätze einer anderen Datei

$$\left[\begin{array}{l} \text{ALL} \\ \left\{ \begin{array}{l} \text{line \#} \\ \text{First} \end{array} \right\} / \left\{ \begin{array}{l} \text{line \#} \\ \text{Last} \end{array} \right\} \end{array} \right]$$

NOTE BEFEHLE

Befehl	Bedeutung
Add [single-record] [NO]	Neueinfügen von Sätzen
Change $\left\{ \begin{array}{l} \text{string-1} \\ \text{Lead} \\ \text{Trail} \end{array} \right\}$ [TO] string-2	Ersetzen einer Zeichenfolge durch eine andere, in einem oder mehreren Sätzen
$\left[\begin{array}{l} \text{single-record} \\ \text{range} \end{array} \right]$ [P]	
DEBUG $\left\{ \begin{array}{l} \text{ALL} \\ \text{PROC} \\ \text{D} \\ \text{PAR} \end{array} \right\}$ $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$	Generieren einer Diagnostikliste
Delete [single-record] [range [string]] [P]	Löschen eines oder mehrerer Sätze
EDIT [single-record-reference]	Inhaltliches Verändern eines Satzes. Es können Zeichen ersetzt, eingesetzt oder verändert werden
END	Beendigung des Editierens
Find string $\left[\begin{array}{l} \text{single-record} \\ \text{range} \end{array} \right]$	Absuchen des Workfiles nach Sätzen mit einem bestimmten Inhalt
[Every] [P]	
Gauge	Zeigen eines Rasters am Schirm
Import file [external-range]	Kopieren von Sätzen aus einer externen Datei in das Workfile
$\left[\begin{array}{l} \text{single-record} \\ \text{range} \end{array} \right]$ [Over]	
List [single-record] [range] [P]	Anzeigen von Sätzen am Bildschirm oder Drucker
MACRO macro-identifizier	Abrufen eines oder mehrerer NOTE Befehle unter einem Buchstaben
MOVE $\left[\begin{array}{l} \text{single-record-reference} \\ \text{range-of-records-reference} \end{array} \right]$	Verschieben von einem oder mehreren Sätzen im Workfile
[TO [single-record-reference]] [NO] [D]	
Open $\left\{ \begin{array}{l} \text{print-file} \\ \text{subject-file} \end{array} \right\}$ P[integer]	Eröffnen eines Druckers oder einer Workfile

	Befehl	Bedeutung
PAge		Setzt eine eröffnete Druckdatei auf Seitenanfang
POsition [single-record]		Verändern des CRP
RENumber [single-record] [range]		Neunummerierung eines Workfile-Teiles
	[base, increment]	
REStore [single-record] [range]		Reaktivieren von durch Delete gelöschten Sätzen
SAVE [file-name [SAME [NOTSAME]		Inhalt des Workfiles auf ein Diskfile hinausschreiben
[RENUMBER [base-value , increment-value]]		

SET	}	ALL	} Setzen von Werten für Texteditorfunktionen
		AUTO { OFF recording-frequency }	
		BASE base-value	
		ID [identification-value]	
		INC increment-value	
		MARGINR [margin-start-position]	
		NEWLINE [newline-character]	
		QUIET { ON OFF }	
		TAB [tabulation-character]	
		XTAB [tab-stop-position [, tab-stop-position] . . .]	

TELL	}	ALL	} Anzeigen von Texteditor-Funktionswerten
		AUTO	
		BASE	
		FIRST	
		ID	
		INC	
		LAST	
		MARGINR	
		NEWLINE	
		QUIET	
TAB			
WORK			

NOTE "REGISTERS"

BASE	Basiswert für Numerierung
CRP	Current Record Pointer. Zeigt auf den sich in Bearbeitung befindlichen Satz
INC	Erhöhungswert für Numerierung
QUIET	Bei CHANGE, FIND usw. Satz anzeigen?
MARGINR	Rechter Rand. Verhindert, daß dahinter Stellen durch STRING-Operationen verändert werden

ZEICHENFOLGEN

1. Bis zu 80 Stellen sind erlaubt
2. Eine Zeichenfolge muß mit einem nicht in der Folge selbst vorkommenden Zeichen limitiert werden.

Programmiersprachen, die IDS benutzen

BASIC
COBOL
FORTRAN

Erstellung von IDS-Tabellen

BASIC und FORTRAN

Monitor flag 16 setzen auf $\left\{ \begin{array}{l} \square = \text{keine Tabellen (Default)} \\ D = \text{alle außer Ursprungs-} \\ \quad \text{zeilen} \\ F = \text{alle} \end{array} \right.$

oder $\left\{ \begin{array}{l} \square = \text{keine Tabellen} \\ \text{SYMBUG} = \text{alle außer Ur-} \\ \quad \text{sprungszeilen} \\ \text{FULL SYMBUG} = \text{alle} \end{array} \right.$

COBOL

Monitor flag 17 setzen auf $\left\{ \begin{array}{l} N = \text{keine Tabellen} \\ Y = \text{alle außer Ursprungs-} \\ \quad \text{zeilen (Default)} \\ I = \text{alle} \end{array} \right.$

IDS Abruf

SCL verb **DEBUG**: ? **DEBUG**; program;

SCL verb **EXECUTE**: ? **EX**ecute program,
 OPTION (DEBUG);

Qualifikatoren

PROGram program-name

SECTion section-name [OF program-name]

SUB subroutine-name

DATA data-name-1 [IN data-name-2] . . . [OF program-name]

IDS-Regeln – Allgemein

1. IDS-Tabellen müssen beim Compilieren erstellt werden.
2. Qualifikations-Anweisungen beeinflussen die Merkmalsauflösung der folgenden IDS-Anweisungen.
3. Die Eingabe von nur NEWLINE wiederholt die zuletzt eingegebene Zeile.
4. Sofern eine IDS-Anweisung mehr als eine Zeile umfaßt, muß am Ende der fortzusetzenden Zeile ein % stehen.
5. Mehrere IDS-Anweisungen auf einer Zeile werden durch Semicolons getrennt.
6. IDS kann nicht bei Lademodulen benutzt werden, die Parallel multiple Tasks enthalten.
7. IDS kann während eines BREAK's aktiviert werden.

Allgemeine IDS-SYNTAX-Regeln

AT-clause := AT $\left\{ \begin{array}{l} \text{ALL} \\ \text{symbolic-code-address} \end{array} \right.$

condition-parameter := $\left[\begin{array}{l} \text{[ON integer [TIME]]} \\ \text{[FOR integer [TIMES]]} \\ \text{[WHEN condition]} \dots \end{array} \right]$

debug acitons := **ACT**ions(statement-list . . .)

flow-limits := $\left\{ \begin{array}{l} \text{ALL} \\ \text{starting-address [TO ending-address]} \end{array} \right.$

statement-list :=
IDS-debug-statement
[; IDS-debug-statement] . . .

IDS-ANWEISUNGEN

symbolic-code-address := $\left\{ \begin{array}{l} \text{BASIC-statement-reference} \\ \text{COBOL-statement-reference} \\ \text{FORTRAN-statement-reference} \end{array} \right\}$

where

BASIC-statement-reference (and) FORTRAN-statement-reference :=

$\left\{ \begin{array}{l} \text{statement-label-number} \\ \text{LINE line-number} \end{array} \right\} \left[\begin{array}{l} \text{IN subroutine-name : OF} \\ \text{program-name} \end{array} \right]$

COBOL-statement-reference :=

$\left\{ \begin{array}{l} \text{paragraph-name [IN section-name]} \\ \text{LINE line-number} \end{array} \right\} \left[\text{OF program-name} \right]$

symbolic-data-address :=

$\left\{ \begin{array}{l} \text{BASIC-symbolic-data-address} \left[\begin{array}{l} \text{IN} \\ \text{OF} \end{array} \right] \text{subroutine-name} \\ \text{COBOL-symbolic-data-address} \left[\begin{array}{l} \text{IN data-name} \\ \text{OF program-name} \end{array} \right] \\ \text{FORTRAN-symbolic-data-address} \left[\begin{array}{l} \text{IN} \\ \text{OF} \end{array} \right] \text{subroutine-name} \end{array} \right\}$

where

BASIC-symbolic-data-address (and)
FORTRAN-symbolic-data-address :=

$\left\{ \begin{array}{l} \text{variable-name} \\ \text{array-name} \left[\left(\text{array-index-value} \left[\text{, array-index-value} \right] \dots \right) \right] \\ \text{string-name} \left[\left(\text{lower-limit-value} : \text{upper-limit-value} \right) \right] \end{array} \right\}$

COBOL-symbolic-data-address :=

data-name $\left[\left(\text{table-index-value} \left[\text{, table-index-value} \right] \dots \right) \right]$

trace-limits := $\left\{ \begin{array}{l} \text{ALL} \\ \text{starting-address [TO ending-address]} \end{array} \right\}$

CLear $\left\{ \begin{array}{l} \text{ALI} \\ \text{symbolic-code-address} \\ \text{symbolic-data-address} \end{array} \right\}$ Löscht früher eingegebene IDS-Anweisungen

DUmp $\left\{ \begin{array}{l} \text{ALI} \\ \text{symbolic-data-address} \end{array} \right\}$ Zeigt Inhalte von Datenfeldern an

ENquire $\left[\begin{array}{l} \text{ALI} \\ \text{Flows} \\ \text{Location} \\ \text{Pauses} \\ \text{SOURCE} \\ \text{STACK} \\ \text{Traces} \\ \text{Watch} \end{array} \right] \dots$ Informiert, welche IDS-Funktionen im Moment aktiv sind und liefert Details über diese Funktionen

EQUATE symbolic-address TO identifier

Teilt einem langen Feldnamen eine Kurzform zu

FLow (flow-limits) [condition-parameter] ... [debug-actions]

Zeigt den Ausführungsweg eines Programms an

GO [TO symbolic-code-address]

Setzt ein Programm nach einem Breakpoint fort

HEIp [IDS-statement-name]

Zeigt das Format einer IDS-Anweisung

IDS-ANWEISUNGEN

IF condition [THEn] statement-list [ELse statement-list] [IFEnd]	Kontrolliert den Ablauf von IDS-Anweisungen
NExt [integer] $\left[\begin{array}{l} \text{LInes} \\ \text{Vmi} \end{array} \right]$	Wird angewendet, um eine bestimmte Anzahl Source- zeilen bzw. Virtual Machine Instructions durchzuführen
OUt = $\left\{ \begin{array}{l} \text{TErmi} \\ \text{PRIn} \\ \text{file-name} \end{array} \right\}$...	Angabe, wohin der IDS-Out- put versorgt werden soll
Pause [AT-clause] [condition-parameter]... (statement-list)	Setzt Breakpoints
REcall file-name	Aufrufen einer Plattendatei, die IDS-Befehle beinhaltet
RU n	Beendet den IDS-Modus
SEt symbolic-data-address-1 to $\left\{ \begin{array}{l} \text{symbolic-data-address-2} \\ \text{literal} \end{array} \right\}$	Ändert den Inhalt eines Datenfeldes.

IDS-ANWEISUNGEN

Source	$\left[\begin{array}{l} * \\ * \text{VMI} \\ \text{statement-line-number...} \\ \text{data-name} \\ \text{paragraph-name} \\ \text{subroutine-name} \\ \text{statement-line-number : statement-line-number} \end{array} \right]$	
		Zeigt Ursprungszeilen bzw. Virtual Machine Instructions an
STOP		Beendigung des IDS-Mo- dus und des zu testenden Programmes
SYsdump	$\left\{ \begin{array}{l} \text{LOCAL} \\ \text{STACK} \end{array} \right\}$...	Ausgabe eines Dumps in das Jobdetailfile
TRace [trace-limits] [condition-parameter]... [debug-actions]		Verfolgen des Programm- ablaufes innerhalb einer be- stimmten Routine
WAtch symbolic-data-address (statement-list)		Ein so definiertes Feld wird nach jeder ausgeführten Anweisung überprüft.

VRX PROTECTION DATA BASE

* Obligatorische User Profiles:

- USERID = PUBLIC
- USERID = SYSTEM
- USERID = SYSADMIN

* Profile-Start (ohne Template-File):

- Batch: NEXTDO PROFUTIL
 FILE REFERENCE = PRINTFILE
 SUD = P0n, NOTCAM
 FILE REFERENCE = COMMANDS
 SUD = P11, NOTCAM
 STOPRD

- Profutil-Parameters -
ENDS

- Interactive: PROFUTIL,
 FILE (REFERENCE = COMMANDS,
 SUD = P11, NOTCAM),
 FILE (REFERENCE = PRINTFILE,
 SUD = P0n, NOTCAM);

* PROFUTIL-Anweisungen:

- | | |
|---|--------------------------------|
| ADD USERID=x, ACCT=x,...; | Neuen Profile-Record erstellen |
| DELETE USERID=x, ACCT=x; | Profile-Record löschen |
| DISPLAY USERID={x
ALL}, ACCT={x
ALL}; | Anzeigen von Profile-Records |
| END; | Ende PROFUTIL |
| MAKETEMP... | «Default-Record» erstellen |
| MODIFY USERID=x, ACCT=x,...; | Profile-Record mutieren |
| OPTIONS REPORT=x...; | Profile-Report-Möglichkeiten |

VRX PROTECTION DATA BASE

* Protected Object Records möglich für:

- jedes zu schützende Cat.-File
- jeden zu schützenden Subcatalog
- für den ganzen Catalog

* PDB File Recovery:

RUNS, PDBRECOVER (an System-Konsole)

STICHWORTE

Abkürzungen	10-110
ALPHADIR	2-30
ASCII-Code	9-20
Band-Band kopieren	2-80
Band(-Teil) kopieren	2-70
Binär-Dezimal-Umrechnung	9-30
CALGEN	2-230
CAM: Aufbau indexed File (ICAM)	5-520
CAM: Aufbau relative File	5-500
CAM: Aufbau sequential File	5-500
CAMBAS-Befehle	6-150
CAMBAS-Status	6-160
CAM-Discfile auf Band kopieren	2-100
CAMSAVE	2-100
CARTUNE	4-90
CBEXTOBJ	4-105
CDPRNT	2-170
COBOL: vgl. "VRYCOBOL"	
COBUG	4-70
Compilieren (VRXCOBOL)	4-20
Control String ausdrucken	2-150
CPYALT	2-80
CSPUL	4-110

DATA DIVISION	4-230
DATBAS-BEFEHLE	5-20
Datenträger(-Teile) vergleichen	2-500
Detail MCL Statement	1-140
Dezimal-Binär-Umrechnung	9-30
Dezimal-Hex-Umrechnung	9-30
DINIT	2-180
DIRDOC	2-30
Directory ab Magnetband drucken	2-40
Directory des Discs drucken	2-30
Directory des Magnetbandes drucken	2-40
Disc-Band kopieren	2-60
Disc-Controller Operationen	1-30
Disc Directory drucken	2-30
Disc-Disc kopieren	2-50
Disc-File-Angaben modifizieren	2-520
Disc-File ausdrucken	2-150
Discfile(s)-Discfile(s) kopieren	2-90
Disc(-File) locken/entlocken	2-300
Disc-File(s) löschen	2-120
Disc-File MERGE	2-240
Disc-File SORT	2-240
Disc initialisieren	2-180
Disc löschen	2-110
Disc-Records selektiv ausdrucken	2-150
Disc reinitialisieren	2-180
Disc-Sektoren ändern	2-510
Disc-Teil-Band kopieren	2-90
Disc-Teilbereich ausdrucken	2-130
Disc-Teile vergleichen	2-500
Disc-Typen-Tabelle	9-60
DML-Befehle (TOTAL)	5-20
DML Status Codes (TOTAL)	5-30
DRECT	2-30 / 2-110 /
	2-120 / 2-300 /
	2-520
Dump-Massstab	9-100

EDIT (IVS)
Editier-Befehle (OLPD)
Entlocken Disc(-File)
ENVIRONMENT DIVISION

10-190
3-30
2-300
4-190

FCL: File Characteristics Parameters 1-150
FCL: Section Characteristics Parameters 1-190
File Status Werte 4-440
File-Typen 9-50
Firmware Control Functions 1-40
Format-Karten: TRANPRO 6-90

GENLCOMP 2-500

Hex-Binär-Umrechnung 9-30
Hex-Dezimal-Umrechnung 9-30

IDS 10-310
IDENTIFICATION DIVISION 4-180
Indexed-CAM-File aufbauen (ICAM) 5-520
Initialisieren Discs:
 6560/6580 an DC-625,PA 2-180
 6530/6531/6540 2-190
 6550/6580 an IDC/6590 2-200
Initialisieren Magnetband 2-210
Interactive Utilities 10-120
IVS-SCL 10-10
IVS-USC 10-100

JCL	1-100
Job Aufbau	1-100
Job Control Language	1-100
JOBDET Parameters	1-120
Job katalogisieren	2-20
JOB Parameters	1-110
Job Specification Language	1-110
JSL	1-110
Kalender Unterhalt	2-230
Katalogisieren eines Jobs	2-20
Konsolen-Befehle (VRX)	1-70
Konsolen-Befehle (TRANPRO)	6-20
Konsolen-Bilden	1-60
Kopieren:	
Band-Band	2-80
Band(-Teil)-Disc	2-70
CAM-File: Disc-Tape	2-100
CAM-File: Tape-Disc	2-100
Discfile(s)-Discfile(s)	2-90
Disc(-Teil)-Band	2-60
L CDPRT	2-170
L CPYALT	2-80
L DINIT	2-180
L DRECT	2-110 / 2-120 / 2-300 / 2-520
Linkage Editor	4-50
LINKEDIT	4-50
LIST80	2-170
LISTCSA4	2-150
L MTPRNT	2-140 / 2-160
Load Module erstellen	4-50
Lochkarten auf disc kopieren	2-290
Lochkarten ausdrucken	2-170

Locken Disc(-File)	2-300
Löschen Disc	2-110
Löschen Disc-File(s)	2-120
L TINIT	2-210
L TRKFX	2-510

Magnetband auf Magnetband kopieren	2-80
Magnetband-Directory drucken	2-40
Magnetbänder vergleichen	2-500
Magnetband-Fils ausdrucken	2-160
Magnetband initialisieren	2-210
Magnetband-Spoolfile ausdrucken	2-220
Magnetband(-Teil) drucken	2-140
Magnetband(-Teil) auf Disc kopieren	2-70
Magnetplatte: vgl. "Disc"	
Master MCL Statements	1-130
MCL	1-130
Menu-Bilder	1-60
MERGE	2-240
Message Communication (OLPD)	3-60
MF 01-10	9-40
Monitor Control Language	1-130
Monitor Flags 01-10	9-40
MODLIBRY	4-150
MSCATALOG	2-20
MTPRINT	2-40 / 2-140 / 2-160

NEXTDO: Monitor Flags	9-40
NOTE	10-260

Object Module binden	4-50
Object Modul Library	4-150
Object Programm Dokumentation	4-105
OLPD: Editing Commands	3-30
OLPD: Message Communications	3-60
OLPD: Spoolfile Access	3-60
OLPD: System Commands	3-20
Operator Befehle: TRANPRO	6-20
Operator Communication: Abfragen	1-60
Operator Communication: Befehle	1-70
Peripheral Type Codes	9-80
PRINT Parameters	1-120
PRINTDISC	2-130 / 2-150
PROCEDURE DIVISION	4-280
Protection Data Base	10-370
QBACKUP	2-60
QDISCMGT	2-60
QDISCOPY	2-50 / 2-90
QMGTDISC	2-70
QMIDRECT	2-40
QRESTORE	2-70
Records selektiv drucken (Disc)	2-150
REFEDIT	2-320
Relative CAM File aufbauen	5-500
Reserved Words (COBOL)	4-420
Runtime Parameters:TRANPRO Job	6-60

SCL	10-10
Sequential CAM File aufbauen	5-500
SFPVRX	2-150
Simulated Option Switch (SIMOPTS)	9-50
SORT/MERGE	2-240
SORTVRX	2-240
Source-Programm ausdrucken	2-150
Source-Programm-Library	4-110
Spoolfile Access (OLPD)	3-60
Spoolfile (ab Tape) ausdrucken	2-220
SPUR	2-290
SPURLIST	2-150
Start: Disc-Controller	1-30
Start: System	1-20
SUD-Spektrum	9-50
SW-1 Dump	1-50
Systemabschluss	1-20
Systemstart	1-20
System Dump	1-50
Tagesstart	1-20
Tapespoolfile ausdrucken	2-220
Text Editor: OLPD	3-20
TINIT	2-210
TOTAL:DML-Befehle	5-20
TOTAL:DML Status Codes	5-30
TRANSPRO	
CAMBAS-Befehle	6-150
CAMBAS-STATUS	6-160
Datenkonvertierung (Input-Data-String)	6-130
Datenkonvertierung (Output-Data-String)	6-120
Datentypen	6-110
Format Karten	6-90
Konsolen Befehle	6-20
Operator Befehle	6-20
Runtime Parameters	6-60
TRKFX	2-510

Umrechnungs-Tabelle	9-30
UNIT Parameter	1-120
USC	10-100
UTILCNTL... vgl. "L..."	
UTIL INITIALIZE	2-200
Vergleichen von Datenträgern	2-500
VRX-Abschluss	1-20
VRXCOBOL:	
Cobol-Merge	4-450
Cobol-Sort	4-450
Compilation	4-20
DATA DIVISION	4-230
Datentypen	4-260
ENVIRONMENT DIVISION	4-190
File Status Werte	4-440
IDENTIFICATION DIVISION	4-180
Konditions-Eintragungen	4-400
Object Dokumentation	4-105
Object Modul Library	4-150
PROCEDURE DIVISION	4-280
Reserved Words	4-420
Source Programm Formate	4-180
Source Programm Library	4-110
Testhilfe COBUG	4-70
Tuninghilfe CARTUNE	
VRX File Status Werte	4-440
VRX Object Modul Library	4-150
VRXOFFLN	2-220
VRX OLPD:	
Editing Commands	3-30
Message Communication	3-60
Spoolfile Access	3-60
System Commands	3-20
VRX Start	1-20
VSn Firmware Control Functions	1-40

Gebräuchliche Abkürzungen

BAT	Basic Access Technique
BO	Boot=Systemstart
CAM	Criterion Access Methode
COACH	Test Firmware
COBUG	COBOL-Debug Testhilfe
CPL	Common Programm Library
CRT	Cathode Ray Tube
CS	Control String
CSD	Common System Disc
DMS	Daten Management System
EDA	Extended Disc Adressing
EDIT	IVS-Editor
FCL	File Control Language
ICAM	Indexed Cam-File
IDPS	Interactiv Direct Processing System
IMOS	Interactiv Multiprogramming Operating System
IRX	Interactiv Resource Executive

IDS	Interactive Deysay System
JCL	Job Control Language
JSL	Job Specification Language
IVS	Interactive virtual System
KB	Kilo-Bytes (1024 Bytes)
LLI	Low Level Interface
LPN	Logical Peripheral Number
MB	Mega-Bytes (1000 KB)
MCL	Monitor Control Language
MCS	Message Control System
MICR	Belegleser
MP	Multi Processor System
NCS	NCR Century Software
NDL	Network Definition Language

NEAT NCR Programmiersprache
NOTE NCR Online Text Editor
NRZI Non Return to Zero (Aufzeichnungsart)

OCTF Operator Communication Transaction File
OLPD Online Program Development

QUERY Abfrage-System

RS Real Storage

SAM System Access Methode
SCL System Command Language
SES System External Storage
SUD Symbolic Unit Designator

TAM Teleprocessing Access Methode
TOICAM Time Ordered Indexed CAM-File
TOTAL Datenbanksystem
TRANPRO Transaction Processing

USC User System Command

VRX Virtual Resource Executive
VS Virtual Storage

9 TABELLEN + CODES

ASCII-Code	9-20
Binär-Hex-Dezimal-Umrechnung	9-30
Monitor Flags bei NEXTDO	9-40
Symbolic Unit Designator, SUD	9-50
Simulated Option Switch (SIMOPTSW)	9-50
File Typen	9-50
Daten der verschiedenen Disc-Typen	9-60
Dump-Adress-«Massstab»	9-70

BINÄR-HEX-DEZIMAL-UMRECHNUNG

BINÄR	HEX	5	4	3	2	1
0000	0	0	0	0	0	0
0001	1	65536	4096	256	16	1
0010	2	131072	8192	512	32	2
0011	3	196608	12288	768	48	3
0100	4	262144	16384	1024	64	4
0101	5	327680	20480	1280	80	5
0110	6	393216	24576	1536	96	6
0111	7	458752	28672	1792	112	7
1000	8	524288	32768	2048	128	8
1001	9	589824	36864	2304	144	9
1010	A	655360	40960	2560	160	10
1011	B	720896	45056	2816	176	11
1100	C	786432	49152	3072	192	12
1101	D	851968	53248	3328	208	13
1110	E	917504	57344	3584	224	14
1111	F	983040	61440	3840	240	15

MONITOR FLAGS bei NEXTDO

MF	CC	Eintrag	Beschreibung
MF01	51	C	alle 30 MONITOR FLAGS auf ASCII-Null setzen
		□	die alten MONITOR FLAGS unverändert übernehmen
MF02	52	0	LOG-Eintrag normal vornehmen
		7	gewisse Logeintragungen unterdrücken
MF03	53	0	kein Programm-Halt
		1	Halt vor Programm-Start
		2	Halt nach Programm-Ende
		3	Halt vor Programm-Start und nach Programm-Ende
MF04	54	0	kein Memory Dump
		1	Memory Dump vor Programm-Start
		2	Memory Dump nach Programm-Ende
		3	Memory Dump vor Programm-Start und nach Programm-Ende
MF05	55	0	kein Memory Clear (des Virtuellen Memorys!)
		1	Memory Clear
		5	Memory Clear und B-Serie auf Simulated System Disc kopieren
MF06	56	0	keine Testhilfen
		C	COBOL DEBUG (COBUG)
		S	SYMBOLIC DEBUG (NEATUS)
MF07	57	0	Programm-Name beim Start auf den System Console Printer ausgeben
		1	Programm-Name nicht ausgeben
MF08	58	0	Programm normal beendet
		2	Aus ...AB wird ...ABJ
MF09	59	0	COBOL-Programm ohne CARTUNE laufenlassen
		C	COBOL-Programm mit CARTUNE laufenlassen
MF10	60		zurzeit noch frei

Die weiteren 20 MONITOR FLAGS sind frei für den Anwender.