

Deutsch



FUJITSU Software BS2000

SORT V8.0A

Benutzerhandbuch

Ausgabe November 2014

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions GmbH GmbH 2014.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	11
1.1	Zielsetzung und Zielgruppen des Handbuchs	11
1.2	Konzept des Handbuchs	12
1.3	Änderungen gegenüber dem Vorgänger-Handbuch	14
1.4	Darstellungsmittel	15
2	Funktionen und Definitionen des SORT	17
2.1	Kommando- und Anweisungsfolgen bei Sortier- und Mischläufen	20
2.2	Sortierarten	24
2.2.1	Vollsortieren	25
2.2.2	Auswahlsortieren	27
2.2.3	Adresslistensortieren	29
2.2.4	Mischen	35
2.3	Steuerfelder für Sortier-/Mischläufe	37
2.3.1	Sortierfelder	38
2.3.2	Restfelder	54
2.3.3	Konstantenfelder	56
2.3.4	Summenfelder	58
2.3.5	Maskenfelder	60
2.3.6	Vergleichsfelder und Vergleichskonstanten	66
2.3.7	Symbolische Namen	69
2.3.8	Tabelle für Überschneidungen verschiedener Feldtypen	71
2.4	Satzbearbeitung und Satzveränderung mit SORT	73
2.5	SORT in XS-Umgebung (31-Bit-Adressierung)	80

2.6	Verwendung erweiterter Zeichensätze mit SORT	81
2.6.1	SORT-spezifische Anwendung erweiterter Zeichensätze	82
2.6.1.1	Sortieren mit erweiterten Codes	82
2.6.1.2	CCSN-Angabe in den Dateien von SORT	83
2.6.1.3	Explizite CCSN-Angabe für Datensätze und Ausgabedatei	84
2.6.1.4	Umsetzung von Zeichenkonstanten in den Code der Datensätze	84
2.6.1.5	Randbedingungen für Maskenfelder	85
2.6.1.6	Verwendung erweiterter Zeichensätze in Benutzerausgängen	85
2.7	Verwendung von Unicode-Zeichensätzen mit SORT	86
2.7.1	Normalisierung	87
2.7.2	Zeichen mit spezieller Verarbeitung	88
2.7.3	Zeichen die nicht unterstützt werden	88
3	Dateien des Sortier-/Mischprogramms SORT	91
3.1	Eingabedateien	95
3.1.1	Eingabedateien für Sortierläufe	95
3.1.2	Eingabedateien für Mischläufe	99
3.1.3	Pamkey-Eliminierung bei Eingabedateien	100
3.2	Ausgabedatei für Sortier-/Mischläufe	101
3.2.1	Pamkey-Eliminierung bei der Ausgabedatei	105
3.2.2	POSIX-Ausgabedatei	106
3.3	Arbeitsdateien	107
3.3.1	Pamkey-Eliminierung bei Arbeitsdateien	109
3.4	Hilfsdateien	110
3.4.1	Pamkey-Eliminierung bei Hilfsdateien	113
3.5	Fixpunktdatei	114
3.5.1	Pamkey-Eliminierung bei Fixpunktdateien	115
3.6	Bindemodulbibliothek SORTMODS	116
3.7	Anweisungsdateien	116
3.8	Endebehandlung der SORT-Dateien	117
3.9	Verarbeitung von POSIX-Dateien mit SORT	119
3.9.1	POSIX im BS2000	119
3.9.2	Sortieren von POSIX-Dateien mit SORT	120
3.10	SORT und ACS	121

3.11	Arbeiten mit Dateien größer 32 GByte	122
3.11.1	Anlegen von Dateien größer 32 GByte	122
3.11.2	Adresslistensortieren	123
3.11.3	Empfehlungen zur Arbeit mit großen Dateien	124
4	Anweisungen von SORT	127
<hr/>		
4.1	Eingabequellen	127
4.2	SDF-Syntaxdarstellung	128
4.3	Fehlerbehandlung	128
4.4	Kurzübersicht der SORT-Anweisungen	129
	ADD-SYMBOLIC-NAMES	130
	ASSIGN-EXITS	134
	ASSIGN-FILES	141
	ASSIGN-RESOURCES	144
	END	146
	MERGE-RECORDS	147
	MODIFY-CODE	153
	MODIFY-SORT-DEFAULTS	154
	SELECT-INPUT-RECORDS	158
	SET-RECORD-ATTRIBUTES	162
	SET-SORT-OPTIONS	169
	SHOW-SORT-DEFAULTS	178
	SORT-RECORDS	179
	SUM-RECORDS	191
5	Aufruf von SORT	195
<hr/>		
5.1	Aufruf von SORT als autonomes Programm	195
	START-SORT	196
	SORT-FILE	198
	Kommandoreturncodes	206
5.2	Aufruf von SORT als Unterprogramm	210
5.2.1	Übergabe der Steuerinformation an SORT	213
5.2.1.1	Ebene 0	214
5.2.1.2	Ebene 1	215
5.2.2	Makroaufrufe für SORT	216
5.2.2.1	SRT0 – SORT-Aufruf über Ebene 0	216
5.2.2.2	SRT1 – SORT-Aufruf über Ebene 1	219

5.3	SORT-Zugriffsmethode SORTZM	224
5.3.1	Funktion der SORT-Zugriffsmethode SORTZM	224
5.3.2	Makroaufrufe der SORT-Zugriffsmethode SORTZM	227
5.3.3	SRTOPEN – Sortierung eröffnen	227
5.3.4	SRTPUT – Satz an SORT übergeben	229
5.3.5	SRTGET – Satz von SORT übernehmen	231
5.3.6	SRTCLSE – Sortierung schließen	233
5.3.7	Beispiel	234
6	Benutzerausgänge von SORT	237
<hr/>		
6.1	PLANNING – Planung beendet	242
6.2	INPUT – Eingabe-Satzbearbeitung	243
6.3	OUTPUT – Ausgabe-Satzbearbeitung	248
6.4	WORK-FILE-OVERFLOW – Arbeitsdatei-Überlauf	254
6.5	EXLST-FOR-INPUT – EXLST-Ausgang für Eingabedateien	256
6.6	EXLST-FOR-OUTPUT – EXLST-Ausgang für Ausgabedateien	258
6.7	PHYSICAL-TRANSLATE – Sonderzeichen-Umsetzungstabelle	260
6.8	VIRTUAL-TRANSLATE – Sonderzeichen-Übersetzungstabelle	261
6.9	EXTERNAL-COMPARE – Reihenfolge durch Benutzerroutine	263
6.10	TRANSLATE-CHARACTER - Reihenfolge durch Gleichsetzungstabellen und codierten Zeichensatz	266
6.11	INT – Unterbrechung des Sortier-/Mischlaufs	267
7	Fixpunktverarbeitung	269
<hr/>		
8	Optimierung von Sortierläufen	271
<hr/>		
8.1	Geeignete CORE-Zuweisung	271
8.2	Virtuelles Mischen	273

8.3	Wahl des Sortierverfahrens	274
8.3.1	Umcodierung	274
8.3.2	Zyklensortieren	274
8.3.3	Multitasksortieren	277
8.4	Geeignete Wahl der Dateieigenschaften	284
8.5	Satzsummierung	284
8.6	SORT als Subsystem	285
8.7	Angabe des OPTIMIZATION-Operanden in der SET-SORT-OPTIONS-Anweisung	285
8.8	Ändern der Voreinstellungen für SORT	285
9	Installation	287
<hr/>		
10	Beispiele	293
<hr/>		
10.1	Einführung	293
10.1.1	Aufruf von SORT	294
10.1.2	Zuweisen der Dateien	294
10.1.3	Definition der Sortierkriterien	295
10.1.4	Eingabe der Anweisungen beenden und Sortierlauf starten	297
10.2	Beispielsortierung einer Datei mit fester Satzlänge	298
10.3	Beispielsortierung einer SAM-Datei mit variablem Satzformat	301
10.4	Übersicht über die Anwendungsfälle	304
10.4.1	SORT als Hauptprogramm	304
10.4.2	Anschluss von Benutzerroutinen	305
10.4.3	SORT als Unterprogramm	305
10.4.4	Sortieren nach Unicode	305
10.5	Beispiele	306
10.5.1	Beispiel 1: Vollsordieren von Sätzen mit festem Satzformat	307
10.5.2	Beispiel 2: Vollsordieren von Sätzen mit variablem Satzformat	309
10.5.3	Beispiel 3: Vollsordieren ISAM-Eingabedatei in SAM-Ausgabedatei	311
10.5.4	Beispiel 4: Vollsordieren von ISAM-Dateien mit variablem Satzformat	314
10.5.5	Beispiel 5: Vollsordieren von mehreren Dateien mit variablem Satzformat	317
10.5.6	Beispiel 6: Vollsordieren (Eingabedatei=Ausgabedatei)	320
10.5.7	Beispiel 7: Vollsordieren EBCDIC nach DIN-Norm-Ordnung	322
10.5.8	Beispiel 8: Vollsordieren FORMAT=*MODIFY-CODE	324

10.5.9	Beispiel 9: Vollsordieren FORMAT=*EXTENDED-CHARACTER und FORMAT=*TRANSLATE-CHARACTER	326
10.5.10	Beispiel 10: Vollsordieren mit Summenbildung und SELECT-INPUT-RECORDS	331
10.5.11	Beispiel 11: Auswahlordieren von Sätzen mit variablem Satzformat	335
10.5.12	Beispiel 12: Auswahlordieren (binär) von Sätzen mit festem Satzformat	338
10.5.13	Beispiel 13: Auswahlordieren einer POSIX-Datei	340
10.5.14	Beispiel 14: Adresslistensordieren von Sätzen mit festem Satzformat	343
10.5.15	Beispiel 15: Mischen von Dateien	345
10.5.16	Beispiel 16: Benutzerausgang INPUT	347
10.5.17	Beispiel 17: Benutzerausgang OUTPUT	351
10.5.18	Beispiel 18: Benutzerausgang PHYSICAL-TRANSLATE	355
10.5.19	Beispiel 19: Benutzerausgang VIRTUAL-TRANSLATE	359
10.5.20	Beispiel 20: SORT als Unterprogramm (Ebene 0)	363
10.5.21	Beispiel 21: SORT als Unterprogramm (Ebene 1)	367
10.5.22	Beispiel 22: SORT-Zugriffsmethode	372
10.5.23	Beispiel 23: SORT-Zugriffsmethode (Mehrfachsordieren)	377
10.5.24	Beispiel 24: Vollsordieren nach Daten im Unicode	386
10.6	Verzeichnis der Beispieldateien	389
10.6.1	Vorbemerkung	389
10.6.2	Inhalt der Datei RESTAURANT	390
10.6.3	Inhalt der Datei LITERATURE	390
10.6.4	Inhalt der Datei MUSEUM	391
10.6.5	Inhalt der Datei CULTURE.1	392
10.6.6	Inhalt der Datei CULTURE.2	392
10.6.7	Inhalt der Datei CULTURE.3	393
11	Meldungen des Sortier-/Mischprogramms	395
11.1	Ausgabe von Meldungen auf SYSOUT	395
11.2	Ausgabe von Meldungen in S-Variablen	398
11.3	SORT-/MERGE-Meldungen	403
12	Anhang	405
12.1	SORT-Fehlerbehandlung	405
12.1.1	Behandlung von SORT-internen Fehlern	405
12.1.2	Fehlerinformation beim Aufruf von SORT als autonomes Programm	406
12.1.3	Fehlerinformation beim Aufruf von SORT als Unterprogramm	408
12.1.4	Aufbau des RCF-Bereiches	409

12.2 Aufbau der SORT-Versorgungstabellen 410

12.2.1 Tabellenübersicht 410

12.2.2 Versorgungsblock SVB 412

12.2.3 Übergabebereich 414

12.2.4 SORT-Anweisungstabellen 416

12.3 Sortiertabelle UTF-16 417

Literatur 443

Stichwörter 445

1 Einleitung

Das vorliegende Handbuch beschreibt die Verwendungsmöglichkeiten und die Handhabung des Softwareprodukts SORT mit dem SDF-Kommandoformat im BS2000.

SORT V8.0 ist ablauffähig ab BS2000/OSD-BC[®] V8.0.

SORT V8.0 kann mit einem Startermodul SRT80 der Versionen SORT V7.3 und neuer aufgerufen werden. Dagegen wird die Abauffähigkeit alter SORT-Versionen mit einem Startermodul SRT80 der Version SORT V8.0 nicht garantiert.

Das Programm SORT sortiert und mischt Datensätze nach bestimmten Kriterien. Der Inhalt und das Format von Datensätzen können verändert werden. SORT bietet Benutzerausgänge zum Anschluss von Benutzerrouinen, sowie die Möglichkeit, SORT als Unterprogramm aufzurufen. Außerdem kann SORT auch über eine eigene Zugriffsmethode SORTZM aufgerufen werden.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Das Handbuch wendet sich an den Systemanwender des BS2000.

Er sollte über BS2000-Kenntnisse verfügen und mit den wichtigsten Kommandos vertraut sein. Die BS2000-Handbücher „Kommandos“ [1] und „Datenverwaltungssystem“ [2] und [3] dienen dabei als Grundlage.

Zusätzlich sind BS2000-Assembler- und Systemkenntnisse erforderlich, wenn SORT als Unterprogramm aufgerufen werden soll und die Benutzerausgänge des SORT verwendet werden. Die Handbücher „ASSEMBLER-Befehle (BS2000)“ [4], „ASSEMBH“ [5] und „Makroaufrufe an den Ablaufteil“ [6] sind dafür die Grundlage.

Ergänzend zu dem hier vorliegenden Handbuch gibt es das Taschenbuch zu SORT [12]. Es enthält eine Zusammenstellung aller Anweisungen und Makros von SORT einschließlich ihrer Syntax. Das Taschenbuch ist als Nachschlagewerk für den mit SORT bereits vertrauten Benutzer gedacht.

1.2 Konzept des Handbuchs

Kapitel 2 gibt einen kurzen Überblick über SORT, erklärt die unterschiedlichen Sortierarten und Steuerfelder, die mögliche Satzbearbeitung und Satzverarbeitung, die Umstellung auf XS-Umgebung, die Benutzung erweiterter Zeichensätze und die Besonderheiten der Unicode-Unterstützung in SORT.

Kapitel 3 informiert über die unterschiedlichen Dateien mit denen SORT arbeitet.

Kapitel 4 beschreibt die Eingabequellen für SORT-Anweisungen, sowie die Syntax der Anweisungen im SDF-Format.

Kapitel 5 stellt die Aufrufmöglichkeiten von SORT und die Übergabe der Steuerinformationen an SORT vor.

Kapitel 6 gibt einen Überblick über die verschiedenen Benutzerausgänge und die zugehörigen Maßnahmen, die der Benutzer veranlassen kann.

Kapitel 7 beschreibt die Verarbeitung von Fixpunkten zur Protokollierung und Kontrolle eines Sortierlaufes.

Kapitel 8 gibt Hinweise für die Optimierung von Sortierläufen.

Kapitel 9 führt alle wichtigen Informationen auf, die beim Installieren von SORT beachtet werden müssen.

Kapitel 10 eignet sich am besten für den Einstieg des Erstanwenders. Dieses Kapitel enthält neben einer Reihe von Anwendungsbeispielen einen Einführungsteil. Dieser ist insbesondere für jene Benutzer gedacht, die mit SORT zum ersten Mal zu tun haben. Dem Erstanwender wird hier auf einfache Weise die Definition und der Ablauf von SORT-Läufen verständlich gemacht.

Kapitel 11 listet sämtliche Meldungen auf, die beim SORT-/MERGE-Lauf auftreten können.

Der Anhang beschreibt die SORT-Fehlerbehandlung, erklärt den Aufbau der SORT-Versorgungstabellen und enthält die Sortiertabelle UTF-16.

Daran anschließend finden Sie die Verzeichnisse für Literatur und Stichwörter.

Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Dieses Handbuch SORT V8.0 enthält folgende Funktionserweiterungen und Änderungen gegenüber dem Vorgängerhandbuch zu SORT V7.9:

Kapitel / Abschnitt	Erweiterungen / Neuheiten
alle Kapitel	SORT V8.0A ist ablauffähig ab BS2000/OSD-BC V8.0.
2.6.1.2	Abhängig von der Art der Datei wird die CCSN-Angabe von SORT unterschiedlich behandelt
4.4	Anweisung SET-SORT-OPTIONS : neuer Operand IGNORE-CHARACTER Angabe von Zeichen, die bei der Sortierreihenfolge nicht berücksichtigt werden sollen
4.4	Anweisung ASSIGN-EXITS : Erweiterung des Operanden WORK-FILE-OVERFLOW= *MODULE(...) um INTERFACE-VERSION Steuert die Übergabe des Satzzählers auf 4 oder 8 Bytes.
4.4	Anweisung SORT-RECORDS : Erweiterung der Operanden ESTIMATED-RECORDS, FROM-RECORD und NUMBER-OF-RECORDS um <alphanum-name 1..19> Erweitert die Angabe der Satzanzahl auf mehr als 2.147.483.639.
5.2.2	Makroaufrufe über SRT0 und SRT1: MULTI=NOIMON: Die Parameterdatei SYSPAR wird beim Start gelesen.
6.4	Benutzerausgang WORK-FILE-OVERFLOW: Erweiterung von MODULE um INTERFACE-VERSION=1/2

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

- Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.
- In den Beispielen sind Benutzereingaben in **halbfetter Schreibmaschinenschrift** und Systemausgaben in *Schreibmaschinenschrift* wiedergegeben.
- Zur Hervorhebung wichtiger Informationen wird folgendes Piktogramm verwendet:



für Hinweise auf besonders wichtige Informationen

- Die Syntax der Anweisungen ist im SDF-Kommandoformat.

2 Funktionen und Definitionen des SORT

SORT ist ein Programm, das Datensätze von maximal 99 Eingabedateien in eine Ausgabedatei sortiert bzw. mischt.

SORT ist im Dialog- und Stapelbetrieb im BS2000 ablauffähig und verarbeitet SAM-, ISAM-, BTAM-, PAM- und POSIX-Dateien.

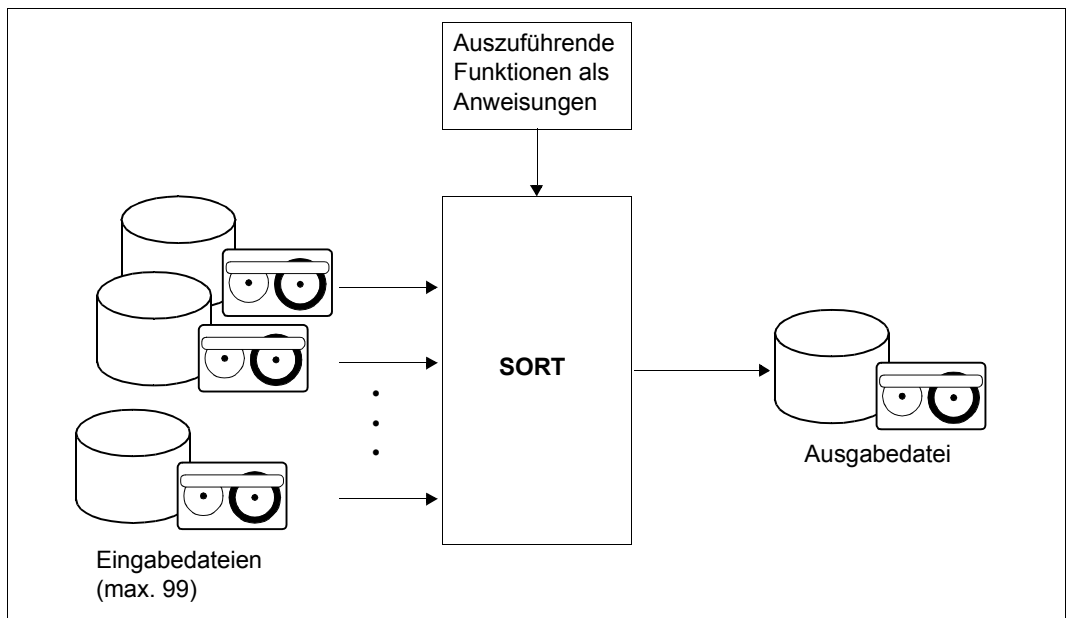


Bild 1: Sortieren und mischen mit SORT

Unter Sortieren versteht man das Ordnen von Datensätzen nach vom Benutzer festgelegten Kriterien.

Beim Mischen werden zwei oder mehr Eingabedateien in eine Ausgabedatei zusammen gemischt. Die Eingabedateien müssen hierfür bereits nach den für den Mischlauf gültigen Kriterien sortiert sein.

Der Benutzer definiert anhand eines oder mehrerer Felder des Eingabesatzes (siehe [Abschnitt „Sortierfelder“ auf Seite 38](#)) die Reihenfolge der Ausgabesätze beim Sortieren und Mischen. Die Ausgabesätze werden auf folgende Arten sortiert:

- aufsteigend
- absteigend oder
- über eine Routine des Benutzers
(siehe [Kapitel „Benutzerausgänge von SORT“ auf Seite 237](#)).

Neben den Funktionen Sortieren und Mischen bietet SORT noch folgende Möglichkeiten:

- Angabe von Auswahlkriterien für die zu sortierenden Sätze:
 - Logische Auswahl von Sätzen für die Sortierung
 - Auswahl einer Satzfolge als Ausschnitt aus den angegebenen Eingabedateien
 - Zusammenfassen von Sätzen mit gleichen Sortierbegriffen ohne und mit Summierung von Summenfeldern im jeweils verbleibenden Satz
 - Eingabereihenfolge bei Sätzen mit gleichen Ordnungskriterien beibehalten
- Sätze und Satzbereiche vom Format und Inhalt her verändern:
 - Änderung des Satzformates
 - Änderung der Satzlängen
 - Umcodierung der Eingabedaten
 - Satzgestaltung durch Auswahlsortieren
- Sätze und Satzteile einfügen, löschen oder verändern:
 - Aufbereiten der Ausgabesätze über Masken für die Druckausgabe
 - Eliminieren von Sortierfeldern
 - Einfügen von frei wählbaren Konstanten
- Protokollierung und Kontrolle des Laufes beeinflussen:
 - Steuerung der Protokollierung und der Laufkontrolle durch den Benutzer
 - Setzen von Fixpunkten
 - Unterbrechen des SORT-Laufes mit anschließendem Dialog mit SORT an der Datensichtstation

- Unterprogramme mit Zusatzfunktionen anschließen:
 - Benutzerausgänge für Anschluss von Benutzerrouinen.

Diese Funktionen sowie der Aufruf von SORT als Unterprogramm und die Zugriffsmethode SRTZM werden in den folgenden Kapiteln beschrieben und erläutert.

Interner Ablauf von Sortierläufen

Ein Sortierlauf lässt sich in folgende Abschnitte gliedern:

1. Vorbereitungsphase

In der Vorbereitungsphase entschlüsselt SORT alle Anweisungen.

2. Planungsphase

SORT legt die Strategie fest und fordert entsprechende Betriebsmittel, wie z.B. Speicher, an.

3. Eingabe- und Vorsortierungsphase

SORT liest die zu sortierenden Sätze ein und erzeugt über eine Vorsortierung geordnete Folgen von Blöcken.

4. DOMINO-Phase

In dieser Phase versucht SORT die vorsortierten Blöcke optimal zu Blockfolgen zu verketteten. Dadurch können in der anschließenden internen Mischphase Datenbewegungen zwischen dem internen Arbeitsspeicher und der Arbeitsdatei eingespart und die interne Mischung verkürzt werden.

5. Interne Mischphase

In der internen Mischphase werden die Blockfolgen durch eine oder mehrere interne Mischungen auf eine für die Endmischung geeignete Anzahl gebracht. Beim Zyklen- und Multitasksortieren (siehe [Kapitel „Optimierung von Sortierläufen“ auf Seite 271](#)) werden diese Blockfolgen jeweils in einer Hilfsdatei zwischengespeichert.

6. Endmisch- und Ausgabephase

In dieser Phase mischt SORT die verbliebenen Folgen (zum Teil auch auf Hilfsdateien) zu einer einzigen Blockfolge und gibt diese in die Ausgabedatei aus.

Die Phasen Eingabe und Vorsortierung sowie die interne Mischphase können bei einer Zyklen- oder Multitasksortierung mehrmals nacheinander durchlaufen werden.

2.1 Kommando- und Anweisungsfolgen bei Sortier- und Mischläufen

Wird SORT als autonomes Programm aufgerufen, sind die folgenden Definitionsschritte nötig:

- Zuweisung der SORT-Dateien über CREATE-FILE- und ADD-FILE-LINK-Kommandos, sofern die Dateizuweisung nicht im SORT-Lauf mit der ASSIGN-FILES-Anweisung des SORT erfolgt
- Aufruf von SORT
- Zuweisen der SORT-Dateien mit der ASSIGN-FILES-Anweisung, sofern die Dateizuweisung nicht schon mittels ADD-FILE-LINK-Kommandos erfolgte. Ein-/Ausgabedateien müssen für jeden Lauf zugewiesen werden, wenn für die Ein-/Ausgabe nicht die Benutzerausgänge INPUT bzw. OUTPUT angegeben wurden. Näheres dazu bei der Beschreibung der ASSIGN-FILES- bzw. ASSIGN-EXITS-Anweisung und im [Kapitel „Dateien des Sortier-/Mischprogramms SORT“ auf Seite 91](#).
- Anweisungen an SORT
Für jeden Lauf muss entweder eine SORT-RECORDS- oder MERGE-RECORDS-Anweisung angegeben werden. Weitere wahlfreie Anweisungen können folgen. Zur Anweisungseingabe können auch Anweisungsdateien verwendet werden. Näheres wird bei der Anweisung ASSIGN-FILES bzw. im [Kapitel „Dateien des Sortier-/Mischprogramms SORT“ auf Seite 91](#) beschrieben.
- END-Anweisung als Abschluss der Anweisungseingabe

Allgemeiner Überblick zur Kommando- und Anweisungsfolge bei Sortier-/Mischläufen

Die Anweisungen und Kommandos müssen in der Reihenfolge eingegeben werden, die durch die nummerierten Abschnitte festgelegt ist. Innerhalb dieser Abschnitte können die Anweisungen in beliebiger Reihenfolge eingegeben werden.

1. Dateizuweisung mit ADD-FILE-LINK-Kommandos (kann entfallen, falls die Dateizuweisung mit der ASSIGN-FILES-Anweisung vorgenommen wird, siehe Punkt 3 in dieser Übersicht)

– Eingabedatei(en)

```
/ADD-FILE-LINK LINK-NAME=SORTIN[,...], -      Sortieren mit einer Eingabedatei
/          FILE-NAME=eingabedateiname
```

oder

```
/ADD-FILE-LINK LINK-NAME=SORTIN01[,...], -    Sortieren mit mehreren Eingabedateien
/          FILE-NAME=eingabedateiname1
/ADD-FILE-LINK LINK-NAME=SORTIN02[,...], -
/          FILE-NAME=eingabedateiname2
/ ...
/ADD-FILE-LINK LINK-NAME=SORTIN99[,...], -
/          FILE-NAME=eingabedateiname99
```

oder

```
/ADD-FILE-LINK LINK-NAME=MERGE01[,...], -     Mischen
/          FILE-NAME=mischdateiname1
/ADD-FILE-LINK LINK-NAME=MERGE02[,...], -
/          FILE-NAME=mischdateiname2
/ ...
/ADD-FILE-LINK LINK-NAME=MERGE99[,...], -
/          FILE-NAME=mischdateiname99
```

– Ausgabedatei

```
/CREATE-FILE FILE-NAME=ausgabedateiname
/ADD-FILE-LINK LINK-NAME=SORTOUT[,...],FILE-NAME=ausgabedateiname
```

– Arbeitsdateien (optional)

```

/CREATE-FILE FILE-NAME=arbeitsdateiname
/ADD-FILE-LINK LINK-NAME=SORTWK[,...], -
/          FILE-NAME=arbeitsdateiname

```

Zuweisung einer Arbeitsdatei

oder

```

/CREATE-FILE FILE-NAME=arbeitsdateiname1
/ADD-FILE-LINK LINK-NAME=SORTWK1[,...], -
/          FILE-NAME=arbeitsdateiname1
/CREATE-FILE FILE-NAME=arbeitsdateiname1
/ADD-FILE-LINK LINK-NAME=SORTWK2[,...], -
/          FILE-NAME=arbeitsdateiname2
/ ...
/CREATE-FILE FILE-NAME=arbeitsdateiname9
/ADD-FILE-LINK LINK-NAME=SORTWK9[,...], -
/          FILE-NAME=arbeitsdateiname9

```

Zuweisung mehrerer Arbeitsdateien

– Hilfsdateien (optional)

```

/CREATE-FILE FILE-NAME=hilfsdateiname1
/ADD-FILE-LINK LINK-NAME=SORTWK01[,...], FILE-NAME=hilfsdateiname1
/CREATE-FILE FILE-NAME=hilfsdateiname2
/ADD-FILE-LINK LINK-NAME=SORTWK02[,...], FILE-NAME=hilfsdateiname2
/ ...
/CREATE-FILE FILE-NAME=hilfsdateiname99
/ADD-FILE-LINK LINK-NAME=SORTWK99[,...], FILE-NAME=hilfsdateiname99

```

– Bindemodulbibliothek (optional)

```

/ADD-FILE-LINK LINK-NAME=SORTMODS, FILE-NAME=bindemodulbibliotheksname

```

– Fixpunktdatei (optional)

```

/CREATE-FILE FILE-NAME=fixpunktdateiname
/ADD-FILE-LINK LINK-NAME=SORTCKPT, FILE-NAME=fixpunktdateiname

```

2. Aufruf von SORT

```

/START-SORT

```

3. Steueranweisungen an SORT

- Dateizuweisung mittels ASSIGN-FILES-Anweisung
(nur notwendig für Dateien, die nicht bereits mit ADD-FILE-LINK-Kommandos zugewiesen wurden, siehe Punkt 1 in dieser Übersicht)

```
//ASSIGN-FILES -
//   INPUT-FILES=eingabedateiname(n), -      Eingabedatei(en)
//                                           (max. 99)
//   OUTPUT-FILES=ausgabedateiname, -        Ausgabedatei
//   WORK-FILES=arbeitsdateiname(n), -       Arbeitsdatei(en)
//                                           (max. 9, optional)
//   AUXILIARY-FILES=hilfsdateiname(n), -    Hilfsdatei(en)
//                                           (max. 99, optional)
//   CHECKPOINT-FILE=fixpunktdateiname, -    Fixpunktdatei (optional)
//   MODULE-LIBRARY=bindemodulbibliotheksname, - Bindemodulbibliothek
//                                           (optional)
//   STATEMENT-FILES=anweisungsdateiname(n)  Anweisungsdatei(en)
//                                           (max. 10, optional)
```

- Angabe der Sortierbedingungen und Definition des Aufbaus der Ausgabedatei

```
//SORT-RECORDS sortierbedingungen          Sortieren
//                                           oder
//MERGE-RECORDS sortierbedingungen        Mischen
```

- Weitere (optionale) Anweisungen

```
//SET-RECORD-ATTRIBUTES ...
//SET-SORT-OPTIONS ...
//SELECT-INPUT-RECORDS ...
```

4. Abschluss der Eingabe und Start des Sortier-/Mischlaufs

```
//END
```

2.2 Sortierarten

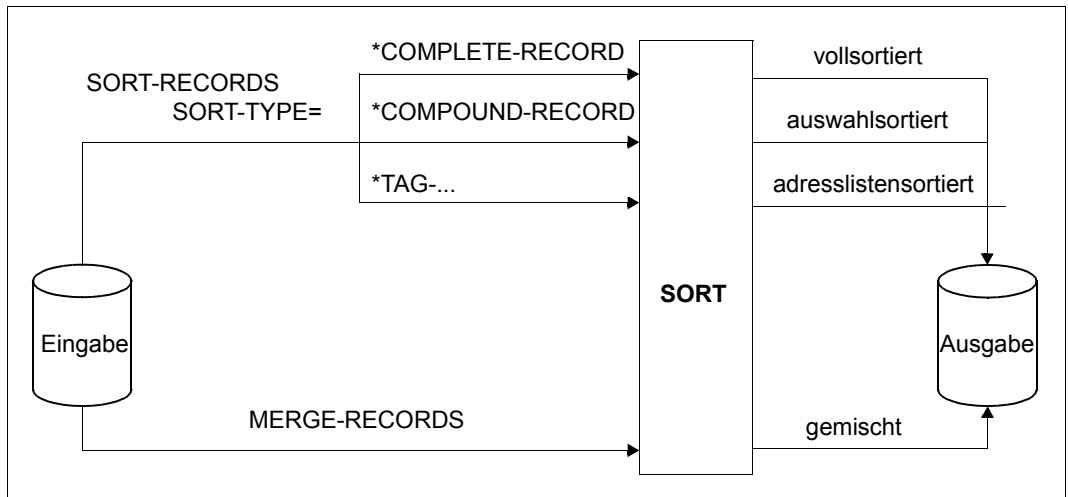


Bild 2: Sortierarten des Sortier-/Mischprogramms SORT

SORT unterstützt folgende Sortierarten:

- **Vollsortieren**
Der gesamte Eingabesatz wird gemäß den Sortierkriterien in die Ausgabedatei geschrieben.
- **Auswahlsortieren**
Ausgewählte Teile des Eingabesatzes werden zu einem neuen Ausgabesatz zusammengestellt. Diese Ausgabesätze werden gemäß den Sortierkriterien in eine Ausgabedatei geschrieben.
- **Adresslistensortieren**
Die gemäß den Sortierkriterien sortierten Sätze erhalten eine von SORT hinzugefügte Adresse, die die Position des ursprünglichen Satzes (Wiedergewinnungsadresse) in der Eingabedatei angibt.
- **Mischen**
Bis zu 99 nach gleichen Sortierkriterien sortierte Eingabedateien gleichen Satzformates (bei fixer Satzlänge auch gleicher Satzlänge) können, entsprechend den Sortierkriterien, in eine Ausgabedatei gemischt werden.

2.2.1 Vollsortieren

Beim Vollsortieren wird der ganze Eingabesatz von SORT übernommen und entsprechend den Sortierfeldern (vgl. [Abschnitt „Sortierfelder“ auf Seite 38](#)) auf- oder absteigend sortiert. Es können sowohl Sätze mit festem wie auch mit variablem Satzformat verarbeitet werden.

Die Ausgabesätze können unter definierten Bedingungen in Satzlänge und/oder Satzformat vom Eingabesatz abweichen (vgl. [Abschnitt „Satzbearbeitung und Satzveränderung mit SORT“ auf Seite 73](#)). Diese Sortierart wird über die Angabe SORT-TYPE=*COMPLETE-RECORD in der SORT-RECORDS-Anweisung ausgewählt. Diese Angabe kann auch weggelassen werden, da sie die Standardeinstellung ist.

Da beim Vollsortieren der ganze Eingabesatz verarbeitet wird, sind keine Aufbereitungsmasken erlaubt.

Beispiel

Die Eingabedatei ADRESSEN enthält folgende Sätze variabler Länge:

s1	MILLER	ANDREW	LONDON	PETER-HOOGANFORD-DRIVE 5
1	5	14	24	39 64
s1	BECKETT	SAM	LIVERPOOL	SEASIDE 37
1	5	14	24	39 50
s1	SOLOW	BRIGITTE	LEEDS	DEARSBY STREET 49
1	5	14	24	39 57
s1	BECKETT	FRANK	MANCHESTER	LOREAN STREET 2
1	5	14	24	39 55

Die Eingabesätze sollen von SORT nach Nachname und Vorname sortiert und ansonsten unverändert in die Ausgabedatei ADRESSEN.SORT geschrieben werden.

```

/start-sort _____ (1)
//assign-files input-files=adressen,output-file=adressen.sort _____ (2)
//sort-records fields=(*field-explicit(position=5,length=9), - _____ (3)
//                      *field-explicit(position=14,length=10))
//set-record-attributes input=*variable(maximum-record-size=64) _____ (4)
//end _____ (5)

```

- (1) Aufruf von SORT
- (2) Zuweisen der Ein- und Ausgabedateien
- (3) Definition der Sortierfelder „Nachname“ und „Vorname“; die Sortierart Vollsortieren sowie aufsteigende Sortierreihenfolge sind voreingestellt
- (4) Angabe der maximalen Länge der Eingabesätze
- (5) Abschluss der Anweisungseingabe und Start des Sortierlaufs

Ausgabedatei ADRESSEN.SORT:

s1	BECKETT	FRANK	MANCHESTER	LOREAN STREET 2
1	5	14	24	39 50 55

s1	BECKETT	SAM	LIVERPOOL	SEASIDE 37
1	5	14	24	39 50

s1	MILLER	ANDREW	LONDON	PETER-HOOGANFORD-DRIVE 5
1	5	14	24	39 64

s1	SOLOW	BRIGITTE	LEEDS	DEARSBY STREET 49
1	5	14	24	39 57

2.2.2 Auswahlsortieren

Beim Auswahlsortieren setzt sich der zu sortierende Satz aus den Teilen des Eingabesatzes zusammen, die im FIELDS-Operanden der SORT-RECORDS-Anweisung als Sortier-, Rest- und Konstantenfelder angegeben wurden. Im Ausgabesatz entspricht die Reihenfolge dieser Teile der Reihenfolge, in der diese Felder definiert wurden.

Standardmäßig erzeugt SORT beim Auswahlsortieren Sätze fester Länge, und zwar unabhängig vom Satzformat der Eingabesätze. Die Sortierart „Auswahlsortieren“ muss über SORT-TYPE=*COMPOUND-RECORD in der SORT-RECORDS-Anweisung angegeben werden.

Besonderheiten bei Eingabesätzen mit variabler Länge:

- Für die Ausgabe von Sätzen fester Länge müssen die angegebenen Sortier- und Restfelder in dem Bereich liegen, der bei allen Sätzen der Eingabedatei vorhanden ist (minimale Satzlänge).
- Sollen Auswahlsätze variabler Länge erzeugt werden, so muss in der SET-RECORD-ATTRIBUTES-Anweisung OUTPUT=*VARIABLE angegeben werden. Für das letzte angegebene Feld gilt:
Ist das Feld ein Restfeld, so darf es ganz oder teilweise im „variablen“ Teil des Eingabesatzes liegen. Der „variable“ Teil des Satzes ist der fiktive Satzbereich, der außerhalb der minimalen Satzlänge liegt und bis zur maximalen Satzlänge reicht.
Ist das Feld ein Sortierfeld vom Format CHARACTER, EBCDIC-DIN oder EBCDIC-INTERNATIONAL, so muss mindestens das erste Zeichen im festen Teil des Eingabesatzes liegen. SORT berechnet die neue Satzlänge automatisch und stellt sie dem neuen Auswahlsatz als Satzlängenfeld voran. Für dieses Satzlängenfeld wird kein Restfeld benötigt.

Die Ausgabesätze können unter bestimmten Bedingungen oder auf Anforderung bezüglich Satzlänge und Satzformat vom Auswahlsatz abweichen, bzw. gegenüber dem Eingabesatz gespreizt oder verkürzt werden (siehe [Abschnitt „Satzbearbeitung und Satzveränderung mit SORT“ auf Seite 73](#)).

Beim Auswahlsortieren ist es möglich, Sortier-, Rest- und Summenfelder im Format BINARY, FIXED-POINT, PACKED-DECIMAL und ZONED-DECIMAL zum Druck aufzubereiten. Dazu wird eine Aufbereitungsmaske benutzt, die der Maske des Assemblerbefehls ED entspricht (siehe [Abschnitt „Maskenfelder“ auf Seite 60](#)).

Beispiel

Aus den Sätzen der Eingabedatei ADRESSEN mit variabler Satzlänge soll eine Ausgabedatei ADRESSEN.SORT erzeugt werden. Diese soll nur die Bereiche Byte 5 - 13 (Familienname) und Byte 24 - 38 (Wohnort) der Eingabesätze enthalten und nach diesen Feldern sortiert sein.

Eingabedatei ADRESSEN:

s1	MILLER	ANDREW	LONDON	PETER-HOOGANFORD-DRIVE 5
1	5	14	24	39 64
s1	BECKETT	SAM	LIVERPOOL	SEASIDE 37
1	5	14	24	39 50
s1	SOLOW	BRIGITTE	LEEDS	DEARSBY STREET 49
1	5	14	24	39 57
s1	BECKETT	FRANK	MANCHESTER	LOREAN STREET 2
1	5	14	24	39 55

```

/start-sort _____ (1)
//assign-files input-files=adressen, - _____ (2)
//          output-file=adressen.sort
//sort-records fields>(*field-explicit(position=5,length=9), - _____ (3)
//          *field-explicit(position=24,length=15)), -
//          sort-type=*compound-record
//set-record-attributes input=*variable(maximum-record-size=64) _____ (4)
//end _____ (5)

```

- (1) Aufruf von SORT
- (2) Zuweisen der Ein- und Ausgabedateien
- (3) Definition der Sortierfelder, Angabe der Sortierart Auswahlsortieren
- (4) Angabe des Datensatztyps und der Datensatzlänge
- (5) Abschluss der Anweisungseingabe

Ausgabedatei ADRESSEN.SORT:

BECKETT	LIVERPOOL
BECKETT	MANCHESTER
MILLER	LONDON
SOLOW	LEEDS
1	10 24

2.2.3 Adresslistensortieren

Wie beim Auswahlsortieren werden beim Adresslistensortieren Auswahlsätze aus den einzelnen Sortierfeldern erzeugt. Zusätzlich fügt SORT jedem Satz der Ausgabedatei die Satzadresse des jeweiligen Satzes der Eingabedatei an (vgl. SORT-TYPE-Operand in der SORT-RECORDS-Anweisung). Da Adresse und Sortierbegriff beim Auswahlsortieren immer feste Länge haben, haben auch die Ausgabesätze beim Adresslistensortieren grundsätzlich eine feste Länge. Das Satzformat der Eingabedatei, ihre DVS-Eigenschaften oder Angaben, die mit dem ADD-FILE-LINK-Kommando gemacht wurden, haben darauf keinen Einfluss. Diese Standardvorgaben von SORT können mit der SET-RECORD-ATTRIBUTES-Anweisung geändert werden.

Besonderheiten bei Eingabesätzen variabler Länge gelten wie beim Auswahlsortieren (siehe [Abschnitt „Auswahlsortieren“ auf Seite 27](#)).

Die Sortierart „Adresslistensortieren“ muss in der SORT-RECORDS-Anweisung mit dem Operanden SORT-TYPE wie folgt angegeben werden:

- SORT-TYPE = *TAG-TRAILER (angefügte Adresse)
- SORT-TYPE = *TAG-HEADER (vorangestellte Adresse)
- SORT-TYPE = *TAG-COMPOUND (vorangestellte erweiterte Adresse)

Bedingungen für das Adresslistensortieren

- Die Eingabedatei muss eine Plattendatei sein.
- Die Eingabedatei muss eine SAM-, ISAM- oder PAM-Datei sein.
- Die Eingabedatei darf keine POSIX-Datei sein.
- Als Eingabe dürfen auch mehrere Eingabedateien gleichen Typs verwendet werden. Da die Wiedergewinnungsadresse jedoch keine Dateikennung enthält, muss der Benutzer selbst sicherstellen, dass die Position der Eingabesätze (und damit die Adresse) unterscheidbar ist.
- Ausgabedatei und Eingabedatei müssen verschieden sein.
- Aufbereitungsmasken sind nicht erlaubt.
- Die Ausgabedatei sollte keine POSIX-Datei sein, da nicht sichergestellt werden kann, dass die Wiedergewinnungsadresse keine Zeichen enthält, die von POSIX als Satztrennzeichen interpretiert werden (siehe [Abschnitt „Sortieren von POSIX-Dateien mit SORT“ auf Seite 120](#)).

Vorteile des Adresslistensortierens

Wenn aus einer Eingabedatei mehrere Dateien erzeugt werden sollen, die nach unterschiedlichen Kriterien sortiert sind, oder wenn die (unsortierte) Eingabedatei erhalten bleiben soll, dann ergeben sich beim Adresslistensortieren folgende Vorteile:

- Speicherplatz für Ausgabedateien kann eingespart werden, da diese nur den Adressverweis auf den Satz der Eingabedatei und ggf. den Sortierbegriff (durch den Operanden ELIMINATE unterdrückbar), enthalten.
- Änderungen im Datenbestand müssen nur in einer Datei, der Eingabedatei (SORTIN-Datei), vorgenommen werden.
 Falls eine solche Änderung nur Teile des Satzes betrifft, die keinen Sortierbegriff darstellen, ist keine weitere Aktion erforderlich. Denn jeder Satz der Eingabedatei kann direkt über die Wiedergewinnungsadresse angesprochen werden, die im Adressfeld der Sätze in den (sortierten) Ausgabedateien steht.
 Auch eine Änderung an einem Sortierbegriff macht lediglich die Wiederholung des Sortierlaufs erforderlich, dessen Sortierbegriff von der Änderung betroffen ist.

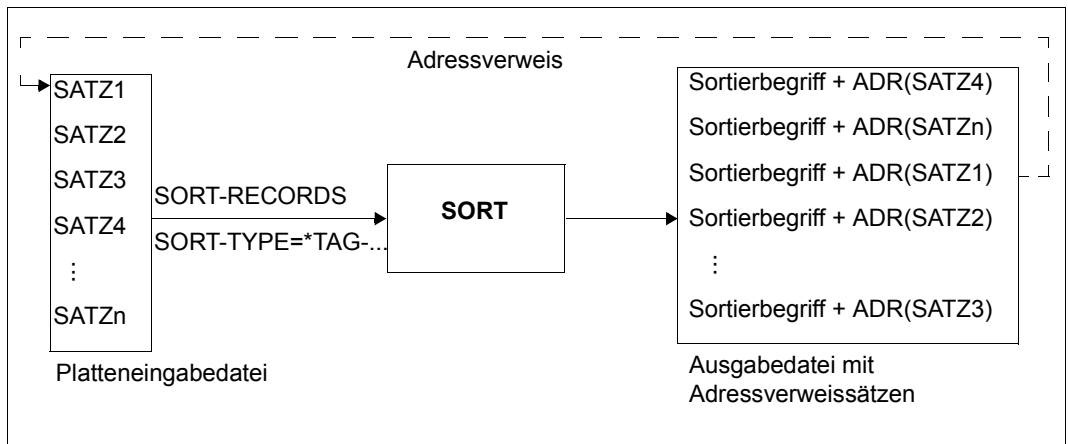


Bild 3: Erstellen von Dateien mit Adressverweissätzen

Soll eine Datei (auf Platte) nach einem Sortierbegriff über Adresslisten sortiert werden, so muss der Benutzer in einem Sortierlauf mit SORT-TYPE=*TAG-COMPOUND oder SORT-TYPE=*TAG-HEADER bzw. SORT-TYPE=*TAG-TRAILER eine Datei mit Adressverweissätzen erstellen. Diese Adressverweissätze bestehen aus Sortier- und Restfeldern und einem Adressfeld (am Anfang bei *TAG-COMPOUND oder *TAG-HEADER bzw. am Ende bei *TAG-TRAILER). Sortierfelder können über den Operanden ELIMINATE auch eliminiert werden.

Adressfeld

Das Format des Adressfelds, das den Anfang (bei *TAG-COMPOUND bzw. *TAG-HEADER) oder das Ende (bei *TAG-TRAILER) des Adressverweissatzes bildet, hängt von der Zugriffsmethode der Eingabedatei ab und sieht folgendermaßen aus:

- **SAM**

Abhängig von der Sortierart ist die Wiedergewinnungsadresse vier oder sechs Byte lang:

- Bei *TAG-HEADER und *TAG-TRAILER wird eine vier Byte lange Adresse im Format bbbbbbrr erzeugt.

bbbbbb Nummer des zugehörigen Dateiblocks, in dem der Satz steht.

rr relative Satznummer im Dateiblock ($1 \leq rr \leq 255$, d.h. der Dateiblock darf nicht mehr als 255 Sätze enthalten).

Beispiel

Adressfeld für den 1. Satz einer Datei (in sedezimaler Darstellung):

000001	01
	1. Satz
1. Dateiblock	

- Bei *TAG-COMPOUND wird eine sechs Byte lange Adresse im Format bbbbbbrrrr erzeugt.

bbbbbbb Nummer des zugehörigen Dateiblocks in dem der Satz steht.

rrrr relative Satznummer im Dateiblock ($1 \leq rrrr \leq 65535$).

Beispiel

Adressfeld für den 1. Satz einer Datei (in sedezimaler Darstellung):

00000001	0001
	1. Satz
1. Dateiblock	

- **ISAM**

Das Adressfeld enthält den ISAM-Schlüssel des entsprechenden Eingabesatzes. Die Länge entspricht dem Dateimerkmal KEY-LENGTH aus der Eingabedatei und kann 1 bis 255 Byte lang sein.

- **PAM**

Das Adressfeld besteht aus den Byte 4 - 7 des PAM-Schlüssels im Format vppp.

v Versionsnummer der Datei (Byte 4)

ppp logische Nummer des PAM-Blocks (Byte 5 - 7)

Beispiel

Adressfeld (in sedezimaler Darstellung):

01	000003
	3. PAM-Block
	1. Version

Beispiel

Die Datei PERSONALDATEI ist eine SAM-Datei und besteht aus Sätzen fester Länge von 59 Byte mit folgendem Aufbau und Inhalt:

Name	Vorname	Straße	Abt.	Pers.Nr.
1	14	26	46	51 59
Miller	Andrew	Poplar Avenue 47	KT25	544507
Allan	Hillary	High Street 101	AY4	345670
Smith	Albert	Gardener Street 14	PX453	047913
Majors	Christine	Railway Cuttings 12	PX23	987650
Smythe	Brenda	Thomas Square 1	BT34	965471
Kennedy	George	Edgeware Road 62	NY211	873250
Stevens	Henry	Market Square 13	NY12	987234
Baker	Fred	Scott Street 34	KT23	765921
Johnston	Annette	Richmond Street 98	BT342	345678
Mellors	Ingrid	Salford Drive 4	TI34	456372
Brown	Tony	Skyview Terrace 9	UB81	786534
Charles	Ernest	Millhouse Street 23	TI32	537892
Walthers	Claudia	Millford Crescent 31	ZY21	342108
Richards	Bernard	Illsley Square 3	UB12	518376
Drever	James	Rose Drive 31	PX3	875211

Diese Datei soll mit der Adresslistensortierung jeweils nach Name, Vorname, Straße, Abt. und Pers.Nr. sortiert werden. Insgesamt sind hierfür 5 Sortierläufe mit SORT-TYPE=*TAG-TRAILER, *TAG-COMPOUND oder *TAG-HEADER erforderlich. Nachfolgend wird der Sortierlauf für den Satzbereich „Name“ beschrieben. Für die übrigen Satzbereiche gilt das hier gesagte in analoger Weise.


```

/start-sort
//assign-files input-files=personaldatei, -
//                output-file=name
//sort-records  fields=(*field-explicit(position=1,length=13)), -
//                sort-type=*tag-trailer
//end

```

Nach diesem Sortierlauf enthält die Ausgabedatei NAME folgende Sätze der Länge 17 Byte (13 Datenbyte, 4 Byte lange Wiedergewinnungsadresse):

Name	Adressfeld*	
	Blk-Nr.	A.
1	14	16 17
A l l a n	000001	02
B a k e r	000001	08
B r o w n	000001	0B
C h a r l e s	000001	0C
D r e v e r	000001	0F
J o h n s t o n	000001	09
K e n n e d y	000001	06
M a j o r s	000001	04
M e l l o r s	000001	0A
M i l l e r	000001	01
R i c h a r d s	000001	0E
S m i t h	000001	03
S m y t h e	000001	05
S t e v e n s	000001	07
W a l t e r s	000001	0D

* Der Inhalt des Adressfelds ist zur besseren Lesbarkeit sedezimal dargestellt

Blk-Nr.: Nummer des Dateiblocks

A.: logische Satzadresse im Block,
z.B. ist „0D“ der 13. Satz der Eingabedatei PERSONALDATEI

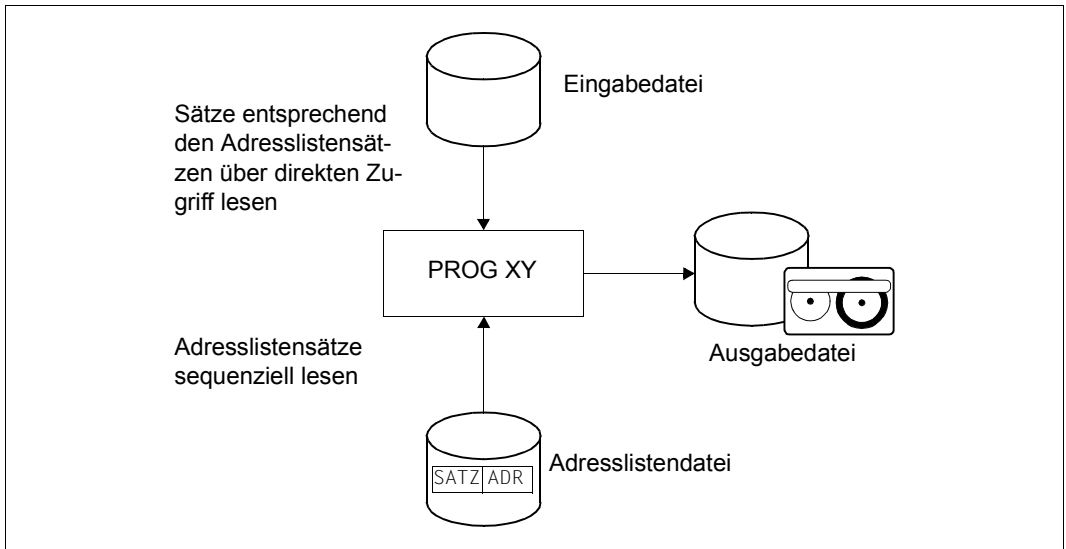


Bild 4: Verarbeitung einer Datei mit Adressverweissätzen

Mit einem Benutzerprogramm kann über das Adressfeld (Wiedergewinnungsadresse, siehe „DVS-Makros“ [3]) der Sätze aus der Datei NAME auf den jeweils zugehörigen gesamten Satz der Datei PERSONALDATEI zugegriffen werden. Der Datei NAME kann entnommen werden, dass z.B. der Satz mit dem Datenfeld „Kennedy“ die logische Satznummer 06 und die Blocknummer 1 hat. Über dieses Adressfeld kann auf den gesamten Satz „Kennedy George Edgeware Road 62 NY211 873250“ aus der Datei PERSONALDATEI zugegriffen werden.

Ändert sich z.B. die Anschrift einer Person in der Datei PERSONALDATEI, so ist nur ein neuer Sortierlauf für die erneute Erstellung der Datei STRASSE erforderlich. Alle anderen Dateien (NAME, VORNAME, ABTEILUNG usw.) bleiben unverändert.

Hinweis

Durch Eliminieren der Sortierfelder können die Ausgabesätze beim Adresslistensortieren zur Platzersparnis auf die Wiedergewinnungsadresse reduziert werden (siehe auch ELIMINATE-Operand in der SORT-RECORDS-Anweisung).

2.2.4 Mischen

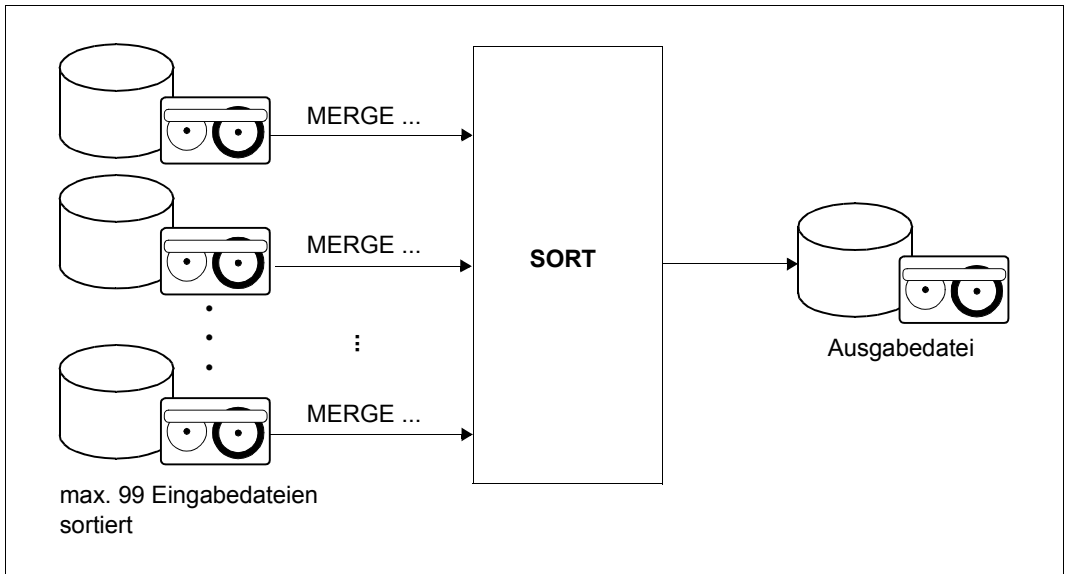


Bild 5: Mischen von Dateien

Beim Mischen von Dateien können bis zu 99 nach gleichen Sortierkriterien sortierte Eingabedateien gleichen Satzformats (bei fixer Satzlänge zusätzlich gleicher Satzlänge) in eine Ausgabedatei entsprechend den Sortierkriterien gemischt werden.

Beispiel

Die beiden Eingabedateien, die aufsteigend nach dem Namen sortiert sind, sollen in eine Ausgabedatei PERSONAL.TOTAL gemischt werden:

Eingabedateien:

Datei PERSONAL1

Name	Vorname
1	10 20

Allan Hilary
Brown Tony
Drever James
Kennedy George

Datei PERSONAL2

Name	Vorname
1	10 20

Baker Fred
Charles Ernest
Johnston Annette
Majors Christine

```

/start-sort _____ (1)
//assign-files input-files=(personal1,personal2), - _____ (2)
//                output-file=personal.total
//merge-records fields=*field-explicit(position=1,length=9) _____ (3)
//end _____ (4)

```

- (1) Aufruf von SORT
- (2) Zuweisen der Ein- und Ausgabedateien
- (3) Mischanweisung
- (4) Ende der Anweisungseingabe

Ausgabedatei PERSONAL.TOTAL:

Name	Vorname
1	10 20

Allan Hilary
Baker Fred
Brown Tony
Charles Ernest
Drever James
Johnston Annette
Kennedy George
Majors Christine

2.3 Steuerfelder für Sortier-/Mischläufe

Unter Steuerfeldern versteht man jene Satzbereiche der zu sortierenden Sätze, die SORT für den jeweiligen Sortier-/Mischlauf auswertet.

Steuerfelder müssen in den Anweisungen zum Sortier-/Mischlauf angegeben werden und können sein:

- Sortierfelder
- Restfelder
- Konstantenfelder
- Summenfelder
- Maskenfelder
- Vergleichsfelder
- Vergleichskonstanten

Für alle Steuerfelder können mit der Anweisung ADD-SYMBOLIC-NAMES symbolische Namen vereinbart werden, die in nachfolgenden Anweisungen verwendet werden können.

SORT generiert verschiedene Verarbeitungsroutinen auf Grund der Angaben für die Steuerfelder. Jede einzelne Routine darf max. 4096 Byte Arbeitsspeicherplatz belegen. Falls diese Angaben zu komplex sind, kann es vorkommen, dass interne Adressierungsprobleme auftreten (Fehlermeldung SRT1250). In diesem Fall sind die Anweisungen zu vereinfachen und der Lauf zu wiederholen.

2.3.1 Sortierfelder

Als Sortierfelder werden jene Bytefolgen eines Satzes bezeichnet, die SORT auswertet, um die Reihenfolge der Ausgabesätze festzulegen. Alle Sortierfelder mit definierter Reihenfolge (vgl. Operand PRIORITY in den Anweisungen SORT-RECORDS- und MERGE-RECORDS) bilden zusammen einen Sortierbegriff. Der Wert eines Sortierfelds, der für die Sortiervergleiche maßgebend ist, und die zulässige Länge sind formatabhängig.

Folgende Formate sind im SORT erlaubt:

Format	Formatbeschreibung	Länge in Byte
BINARY	binär	1 Bit - 256 Byte
CHARACTER	Zeichen	1 - max. Satzlänge
FIXED-POINT	Festpunkt	1 - 256
FLOATING-POINT	Gleitpunkt	1 - 256
PACKED-DECIMAL	dezimal gepackt	1 - 16
ZONED-DECIMAL	dezimal entpackt	1 - 16
EBCDIC-DIN	EBCDIC nach DIN-Norm-Textordnung	1 - 256
EBCDIC-INTERNATIONAL	EBCDIC nach internationaler Textordnung	1 - 256
VIRTUAL-TRANSLATE	Ordnung durch Umcodierungstabelle bestimmt (ASSIGN-EXITS)	1 - 256
PHYSICAL-TRANSLATE	Ordnung durch Umcodierungstabelle bestimmt (ASSIGN-EXITS)	1 - 256
MODIFY-CODE	Ordnung durch MODIFY-CODE-Anweisung bestimmt	1 - 256
EBCDIC-ISO- EBCDIC	Eingabe im EBCDIC-Code, Sortierung im ISO-Code, Ausgabe im EBCDIC-Code	1 - 256
EXTENDED-CHARACTER	Ordnung durch codierten Zeichensatz bestimmt	1 - 256
TRANSLATE-CHARACTER	Ordnung durch Gleichsetzungstabellen und codierten Zeichensatz bestimmt (ASSIGN-EXITS/Bindemodul)	1 - 256
UNICODE-CHARACTER	Ordnung durch die von XHCS gelieferte Unicode Default Collation Table bestimmt	2 - 256

Sortierfelder können in der SORT-RECORDS und MERGE-RECORDS-Anweisung angegeben werden. Die Art und Weise, wie Sortierfelder definiert werden, ist bei der SORT-RECORDS und MERGE-RECORDS-Anweisung beschrieben.

Die folgende Tabelle zeigt im Überblick, welche Angaben für Sortierfelder gemacht werden müssen bzw. können:

Angabe zum Sortierfeld	wahlfrei/ obligatorisch	Bedeutung
POSITION	obligatorisch	Anfang des Sortierfelds
LENGTH	obligatorisch	Länge des Sortierfelds
FORMAT	wahlfrei	Format des Sortierfelds
PRIORITY	wahlfrei	Ordnungsnummer des Sortierfelds
SORTING-ORDER	wahlfrei	Reihenfolge der Sortierung
ELIMINATE	wahlfrei	Zu eliminierendes Sortierfeld
PRINT-MASK	wahlfrei	Zu verwendende Aufbereitungsmaske

Beispiel

Aufbau der Eingabesätze:

Name	Vorname	Straße	Telefon
1	14	26	48 57

Variante 1

Der Sortierbegriff enthält nur ein einziges Sortierfeld, das Feld „Name“. Es beginnt bei Position 1 und ist 13 Byte lang. Diese beiden Werte sind in der Sortier-Anweisung anzugeben.

```
/start-sort
//sort-records fields=*field-explicit(position=1,length=13)
//end
```

Um nach dem 10 Byte langen Feld „Telefon“ ab Position 48 zu sortieren, sind folgende Anweisungen erforderlich:

```
/start-sort
//sort-records fields=*field-explicit(position=48,length=10)
//end
```

Variante 2

Es soll nach dem Feld „Name“ und (bei Sätzen mit gleichem Inhalt des Felds „Name“) nach „Straße“ sortiert werden. Der Sortierbegriff enthält deshalb mehrere Sortierfelder.

```
/start-sort
//sort-records fields=(*field-explicit(position=1,length=13), -
//                      *field-explicit(position=26,length=22))
//end
```

Sortierfelder bei Sätzen mit variabler Satzlänge

Bei allen Formaten mit Ausnahme der Formate CHARACTER, EBCDIC-DIN und EBCDIC-INTERNATIONAL dürfen bei variabler Satzlänge für Sortierfelder nur Satzbereiche angegeben werden, die nicht im „variablen Teil“ des Satzes liegen. Der „variable Teil“ des Satzes ist der fiktive Satzbereich, der außerhalb der minimalen Satzlänge liegt und bis zur maximalen Satzlänge reicht.

Dies bedeutet, dass ein Sortierfeld (ausgenommen die obigen Formate) innerhalb der erlaubten Längen (formatabhängig) nur so lang sein darf wie der kürzeste zu verarbeitende Satz.

Bei den Formaten CHARACTER, EBCDIC-DIN und EBCDIC-INTERNATIONAL dürfen sich die Sortierfelder auch auf den „variablen Teil“ des Satzes erstrecken.

Der nicht vorhandene fiktive Satzbereich wird bei den Sortiervergleichen so behandelt, als ob er mit dem Füllzeichen (FILLER-Operand in der SET-RECORD-ATTRIBUTES-Anweisung) aufgefüllt wäre. Ist kein Füllzeichen angegeben, wird X'00' angenommen.

Beim Auswahl-sortieren muss ein solches Sortierfeld auch das letzte Auswahlfeld sein. Es wird dann nur der vorhandene Teil des Feldes übernommen und die Satzlänge neu berechnet (siehe auch [Abschnitt „Restfelder“ auf Seite 54](#)).

Beispiel

SORT soll Sätze variablen Satzformats mit folgendem Aufbau nach dem Sortierfeld „Abteilung“ (Format CHARACTER) sortieren:

SL	Name	Vorname	Straße	Abteilung
1	5	19	32	54 61
minimale Satzlänge				var. Teil
maximale Satzlänge				

Inhalt der Datei

SL	Miller	Andrew	Platerton Road 47	DV8
1	5	19	32	54 56

SL	Summer	Brenda	Thomas Circuit 1	D
1	5	19	32	54

SL	Becker	Frank	Solomon Hill 34	DSTQM217
1	5	19	32	54 61

Das Feld „Abteilung“ liegt im variablen Satzbereich der zu sortierenden Sätze. Soll das Feld „Abteilung“ als Sortierbegriff genommen werden, muss im FIELDS-Operanden *FIELD-EXPLICIT(POSITION=54,LENGTH=8) angegeben werden, da das erste Byte des Feldes (Byte 54) noch innerhalb der minimalen Satzlänge liegen muss, und die maximale Satzlänge bis Byte 61 reicht.

Das Satzlängefeld als Sortierfeld

Bei Sätzen mit variabler Satzlänge kann auch das Satzlängefeld als Sortierfeld verwendet werden, um z.B. die Sätze nach ihrer Länge zu sortieren.

Das Format des Sortierfeldes muss dabei BINARY, CHARACTER oder FIXED-POINT sein. Bei anderen Formaten erfolgt eine Fehlermeldung mit Abbruch des SORT.

Sortierfeldüberschneidungen

Sortierfelder dürfen sich überschneiden, wenn ihre Formate miteinander kombiniert werden dürfen. Die erlaubten Kombinationen können der folgenden Tabelle entnommen werden.

	BI	CH	VT	ED	EI	PD	ZD	FI	FL	PT	MC	EE	EC	TC	UC
BINARY (BI)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
CHARACTER (CH)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
VIRTUAL-TRANSLATE (VT)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
EBCDIC-DIN (ED)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
EBCDIC-INTERNATIONAL (EI)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
PACKED-DECIMAL (PD)	+	+	+	+	+	-	-	-	-	-	-	-	-	+	+
ZONED-DECIMAL (ZD)	0	0	0	0	0	-	-	-	-	-	-	-	-	0	0
FIXED-POINT (FI)	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0
FLOATING-POINT (FL)	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
PHYSICAL-TRANSLATE (PT)	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
MODIFY-CODE (MC)	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
EBCDIC-ISO-EBCDIC (EE)	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
EXTENDED-CHARACTER (EC)	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
TRANSLATE-CHARACTER (TC)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+
UNICODE-CHARACTER (UC)	+	+	+	+	+	+	0	0	0	0	0	0	0	+	+

Erläuterung:

- + Die Formatkombination ist erlaubt.
- Die Formatkombination ist nicht erlaubt, da beim Vergleich Datenfehler entstehen können. Enthält der FIELDS-Operand eine solche Kombination, gibt SORT eine Fehlermeldung aus.
- o Die Formatkombination kann beim Vergleich zu einem undefinierten Ergebnis führen. Enthält der FIELDS-Operand eine solche Kombination, gibt SORT eine Warnungsmeldung aus.

Beispiel

Im folgenden Beispiel wird gezeigt, wie sich zwei Sortierfelder überschneiden können. Das erste Feld beginnt bei Byte 5 und ist 10 Byte lang. Das Format des Sortierfeldes ist CHARACTER, die Sortierfolge aufsteigend. Diese beiden Angaben werden weggelassen, da sie Standardwerte sind.

Das zweite Sortierfeld beginnt bei Byte 10 und hat eine Länge von 7 Byte. Das Feldformat ist BINARY, die Sortierfolge ist absteigend.

Die beiden Sortierfelder überschneiden sich demnach auf den Bytes 10 bis 14.

```
/start-sort
//sort-records fields>(*field-explicit(position=5,length=10), -
//                               *field-explicit(position=10,length=7, -
//                               format=*binary, -
//                               sorting-order=*descending))
//end
```

Umcodieren von Sortierfeldern

Die Formate EBCDIC-DIN, EBCDIC-INTERNATIONAL, EBCDIC-ISO-EBCDIC, VIRTUAL-TRANSLATE, MODIFY-CODE, PHYSICAL-TRANSLATE, EXTENDED-CHARACTER, TRANSLATE-CHARACTER und UNICODE-CHARACTER beinhalten Umcodierungen. Diese Umcodierungen gelten für die Vergleiche während des Sortierens bzw. Mischens und teilweise auch für die Ausgabe von SORT.

Für die Formate VIRTUAL-TRANSLATE, PHYSICAL-TRANSLATE und MODIFY-CODE muss der Benutzer Code-Tabellen selbst definieren, um so SORT optimal an seine Bedürfnisse anpassen zu können. Für das Format TRANSLATE-CHARACTER werden dem Benutzer zwei grundlegende Code-Tabellen bereitgestellt, für das Format UNICODE-CHARACTER wird dem Benutzer die Unicode Default Collation Table (UTF-16) bereitgestellt; weitere Code-Tabellen können anhand von Muster-Quellen leicht erstellt werden (siehe [Seite 50](#)).

Format EBCDIC-DIN

Sortierfelder im EBCDIC-DIN-Format ermöglichen eine DIN-Norm-gemäße Textsortierung. Ziel dieser Textordnung ist eine Reihenfolge, die dem ISO-7-Bit-Code (Deutsche Referenz-Version nach DIN 66003) entspricht (Ziffern sind dabei in der Sortierreihenfolge vor Buchstaben), jedoch die Kleinbuchstaben den entsprechenden Großbuchstaben gleichsetzt. Zusätzlich führt SORT für diese Sortierfelder noch folgende Gleichsetzungen durch:

- X'00' wird wie X'40' (Leerzeichen) behandelt
- 'ä' bzw. 'Ä' werden wie 'AE' behandelt
- 'ö' bzw. 'Ö' werden wie 'OE' behandelt
- 'ü' bzw. 'Ü' werden wie 'UE' behandelt
- 'ß' wird wie 'SS' behandelt

SORT verändert die Sortierfelder im EBCDIC-DIN-Format in den Sätzen nicht, sondern benutzt Hilfsfelder.

SORT erwartet im EBCDIC-DIN-Format die Sortierfelder in den Codes

- EBCDIC.SRV.10 (Referenz-Version 10 des 8-Bit-Codes) oder
- EBCDIC.DF.03 (Internationale/Deutsche DF-Version 03).

Für beide Codes benutzt SORT eine gemeinsame Umsetztabelle.

Format EBCDIC-INTERNATIONAL

Sortierfelder im EBCDIC-INTERNATIONAL-Format ermöglichen eine von der DIN-Norm abgeleitete internationale Textsortierung. Ziel dieser Textordnung ist eine Reihenfolge, die dem ISO-7-Bit-Code (Deutsche Referenz-Version nach DIN 66003) entspricht (Ziffern sind dabei in der Sortierreihenfolge vor Buchstaben), jedoch Kleinbuchstaben den entsprechenden Großbuchstaben gleichsetzt. Umlaute werden *nicht* gleichgesetzt, sondern als Sonderzeichen behandelt.

Format VIRTUAL-TRANSLATE

Bei Sortier-/Mischläufen, für die Sortierfelder im VIRTUAL-TRANSLATE-Format angegeben werden, muss der Benutzer dazu in der ASSIGN-EXITS-Anweisung den Benutzerausgang VIRTUAL-TRANSLATE mit dem Namen eines Moduls, der eine Umcodierungstabelle enthält, angeben. Die Zeichenumsetzung muss nicht eindeutig sein, d.h. es dürfen mehrere Zeichen auf den gleichen Wert umgesetzt werden.

Beim Format VIRTUAL-TRANSLATE werden die Sortierfelder in jeweils ein Hilfsfeld umcodiert, diese Hilfsfelder miteinander verglichen und so die Sortierreihenfolge ermittelt. Die Sortierfelder werden nicht verändert.

Format PHYSICAL-TRANSLATE

Werden für einen Sortier-/Mischlauf Sortierfelder im PHYSICAL-TRANSLATE-Format angegeben, muss der Benutzer dazu in der ASSIGN-EXITS-Anweisung den Benutzerausgang PHYSICAL-TRANSLATE mit dem Namen eines Moduls, der zwei vom Benutzer zu erstellende Umcodierungstabellen enthält, angeben. Die Umcodierungstabellen müssen die Zeichenreihen für die Umcodierung am Anfang und Ende der Sortierung/Mischung enthalten. Die zweite Codiertabelle ist notwendig, um am Ende das Sortierfeld wieder zurückzucodieren. Im Gegensatz zum VIRTUAL-TRANSLATE-Format werden die Sortierfelder umcodiert, verglichen und danach rückcodiert. Dies bedeutet, dass der Benutzer auf die Eindeutigkeit der Zeichen über die Umcodierung hinweg achten muss und mehrere Zeichen nicht die gleiche Wertigkeit haben dürfen.

Format MODIFY-CODE

Sortierfelder im MODIFY-CODE-Format beziehen sich auf eine MODIFY-CODE-Anweisung. SORT erstellt auf Grund einer MODIFY-CODE-Anweisung zwei Umcodierungstabellen mit Zeichenzuordnungen. Diese Art der Umcodierung hat gegenüber dem Format PHYSICAL-TRANSLATE den Vorteil, dass SORT selbst für eine eindeutige Zeichenzuordnung sorgt. SORT codiert die Zeichenreihen von Sortierfeldern im MODIFY-CODE-Format am Anfang und Ende der Sortierung/Mischung über die zwei Umcodierungstabellen im Satz selbst um.

Format EBCDIC-ISO-EBCDIC

Die Zeichenreihen von Sortierfeldern des Formates EBCDIC-ISO-EBCDIC codiert SORT auf Grund der Formatangabe der einzelnen Sortierfelder vom Eingabecode zum internen Vergleichscode und dann wieder in den Ausgabecode um.

Code für die SORT-Eingabe	Code für die interne Mischung	Code für die SORT-Ausgabe
EBCDIC	ASCII	EBCDIC

Die dafür maßgebenden Umsetztabelle sind nachfolgend dargestellt.

Umsetztabelle von EBCDIC in erweiterten ASCII-Code

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	21	24	25	03	09	0E	FF	26	27	2A	0B	0C	31	34	35
1	36	11	37	60	15	0D	08	10	61	6A	1A	6B	1C	1D	1E	1F
2	20	12	22	23	1B	0A	17	01	28	29	06	2B	2C	2D	2E	2F
3	30	13	32	33	02	18	0F	04	38	39	3A	3B	3C	3D	3E	3F
4	40	14	6C	6D	6E	6F	71	7A	7B	7C	E0	4E	5C	48	4B	FC
5	46	16	7D	7E	7F	81	82	83	84	85	41	44	4A	49	5B	19
6	4D	4F	62	63	64	65	66	67	68	69	BE	4C	45	BF	5E	5F
7	70	05	72	73	74	75	76	77	78	79	5A	43	A0	47	5D	42
8	80	E1	E2	E3	E4	E5	E6	E7	E8	E9	8A	8B	8C	8D	8E	8F
9	90	EA	EB	EC	ED	EE	EF	F0	F1	F2	9A	9B	9C	9D	9E	9F
A	86	87	F3	F4	F5	F6	F7	F8	F9	FA	88	89	91	92	93	94
B	95	96	97	98	99	C1	C2	C3	C4	C5	C6	BB	BC	BD	C7	C8
C	C0	A1	A2	A3	A4	A5	A6	A7	A8	A9	CA	CB	CC	CD	CE	CF
D	D0	AA	AB	AC	AD	AE	AF	B0	B1	B2	DA	DB	DC	DD	DE	DF
E	07	C9	B3	B4	B5	B6	B7	B8	B9	BA	D1	D2	D3	D4	D5	D6
F	50	51	52	53	54	55	56	57	58	59	D7	FB	D8	FD	D9	FE

Umsetztabelle vom erweiterten ASCII-Code in EBCDIC

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	27	34	04	37	71	2A	E0	16	05	25	0B	0C	15	06	36
1	17	11	21	31	41	14	51	26	35	5F	1A	24	1C	1D	1E	1F
2	20	01	22	23	02	03	08	09	28	29	0A	2B	2C	2D	2E	2F
3	30	0D	32	33	0E	0F	10	12	38	39	3A	3B	3C	3D	3E	3F
4	40	5A	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
5	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
6	13	18	62	63	64	65	66	67	68	69	19	1B	42	43	44	45
7	70	46	72	73	74	75	76	77	78	79	47	48	49	52	53	54
8	80	55	56	57	58	59	A0	A1	AA	AB	8A	8B	8C	8D	8E	8F
9	90	AC	AD	AE	AF	B0	B1	B2	B3	B4	9A	9B	9C	9D	9E	9F
A	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
B	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	BB	BC	BD	6A	6D
C	C0	B5	B6	B7	B8	B9	BA	BE	BF	E1	CA	CB	CC	CD	CE	CF
D	D0	EA	EB	EC	ED	EE	EF	FA	FC	FE	DA	DB	DC	DD	DE	DF
E	4A	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
F	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	FB	4F	FD	FF	07

Format EXTENDED-CHARACTER

Sortierfelder im EXTENDED-CHARACTER-Format ermöglichen eine Textsortierung in der für den codierten Zeichensatz vorgesehenen Reihenfolge. Die Sortierung erfolgt wie bei PHYSICAL-TRANSLATE durch Umcodieren, Vergleichen und abschließendem Rückcodieren. Die interne Codiertabelle ist abhängig vom benutzten codierten Zeichensatz (coded character set, CCS).

SORT versucht, das CCS zur Sortierung der Datensätze durch Auswertung der folgenden Quellen zu ermitteln:

- Katalogeintrag der Eingabedatei
- SET-RECORD-ATTRIBUTES-Anweisung des Sortierlaufes
(keine Eingabedatei vorhanden)
- Ist keine Eingabedatei vorhanden und wird auch kein CCSN über die SET-RECORD-ATTRIBUTES-Anweisung angegeben, so verwendet SORT den Code EDF03IRV.

Wird der CCSN eines Codes ermittelt, der nicht in XHCS definiert ist, wird der Sortierlauf mit der Meldung SRT1258 abgebrochen.

Bei einer ISAM-Ausgabedatei darf das Format EXTENDED-CHARACTER nicht für Sortierfelder verwendet werden, die den ISAM-Schlüssel bilden. Die ISAM-Schlüssel müssen im Sinne des Host Codes aufsteigend sortiert sein, eine Sortierung in einem anderen Code ist nicht zulässig. Im Fehlerfall wird der Sortierlauf mit der Meldung SRT1261 abgebrochen.

Format TRANSLATE-CHARACTER

Das Format TRANSLATE-CHARACTER setzt zwei Umsetztabelle voraus. Aus diesen Tabellen generiert SORT den Code des zu sortierenden Zeichens für den Sortiervorgang. In der ersten Tabelle steht der Code, mit dem das zu sortierende Zeichen gleichgesetzt wird. In der zweiten Tabelle steht der Code des Zeichens, um den das zu sortierende Zeichen erweitert wird. Die Unterscheidung, ob das zu sortierende Zeichen nur gleichgesetzt oder auch erweitert wird, trifft SORT über den Eintrag am Code-Platz des Zeichens in der zweiten Tabelle. Ist dieser X'00', so wird gleichgesetzt, sonst wird das gleichgesetzte Zeichen um das Zeichen aus der zweiten Tabelle erweitert. SORT ordnet die Sortierfelder nach dem Gleichsetzen und Erweitern in der Sortierreihenfolge des CCS.

Je nach Aussehen der beiden Tabellen können die Zeichen des Sortierfeldes für die Sortierung auf zwei Weisen beeinflusst werden:

– **Gleichsetzen von Zeichen**

Zum Gleichsetzen wird in der ersten Tabelle auf dem Code-Platz des zu sortierenden Zeichens der Code des Zeichens eingetragen mit dem das zu sortierende Zeichen gleichgesetzt werden soll. In der zweiten Tabelle muss dann auf dem Code-Platz des Zeichens der Wert X'00' eingetragen sein.

Beispiel: 'a' soll mit 'A' gleichgesetzt werden.

Die erste Tabelle muss am Code-Platz von 'a' (=X'81') den Code des Zeichens 'A' (=X'C1') enthalten. Der Code-Platz X'81' muss in der zweiten Tabelle mit X'00' belegt sein (siehe Tabellen auf [Seite 51f](#)).

– **Erweitern eines Zeichens auf zwei Ersatzzeichen**

Zum Erweitern wird in der ersten Tabelle auf dem Code-Platz des Zeichens der Code des *ersten* Ersatzzeichens und in der zweiten Tabelle der Code des *zweiten* Ersatzzeichens eingetragen.

Beispiel: 'ä' soll auf 'AE' umgesetzt werden.

Die erste Tabelle muss am Code-Platz von 'ä' (=X'AB' für EBCDIC.SRV.10 bzw. X'FB' für EBCDIC.DF.03) den Code des ersten Ersatzzeichens 'A' (=X'C1') enthalten. Der gleiche Code-Platz muss in der zweiten Tabelle mit dem Code des zweiten Ersatzzeichens 'E' (=X'C5') belegt sein (siehe Tabellen auf [Seite 51f](#)).

Die Sortierung erfolgt wie bei VIRTUAL-TRANSLATE, d.h. die Sortierfelder werden jeweils in ein Hilfsfeld umcodiert, diese Hilfsfelder miteinander verglichen und so die Sortierreihenfolge ermittelt. Die interne Codiertabelle ist abhängig vom benutzten codierten Zeichensatz. Das CCS zur Sortierung der Datensätze wird wie bei EXTENDED-CHARACTER bestimmt.

Die Sortierfelder werden nicht verändert.

Die Zeichenumsetzung muss nicht eindeutig sein, d.h. es dürfen mehrere Zeichen auf den gleichen Wert abgebildet werden.

Bei einer ISAM-Ausgabedatei darf das Format TRANSLATE-CHARACTER nicht für Sortierfelder verwendet werden, die den ISAM-Schlüssel bilden. Die ISAM-Schlüssel müssen im Sinne des Host Codes aufsteigend sortiert sein, eine Sortierung in einem anderen Code ist nicht zulässig. Im Fehlerfall wird der Sortierlauf mit der Meldung SRT1261 abgebrochen.

Hinweise für das Erstellen der TRANSLATE-CHARACTER-Tabellen:

Die beiden Tabellen je codiertem Zeichensatz werden als ein Bindemodul in der Tabellen-Bibliothek mit dem logischen Namen SYSLNK.TAB abgelegt.

Der Name des Bindemoduls entspricht dem verwendeten Namen des CCS (z.B. „EDF03DRV“).

Die Bibliothek enthält neben den standardmäßig ausgelieferten Bindemodulen „EDF03DRV“ und „EDF03IRV“ auch die zugehörigen Quellen sowie eine Muster-Source, die dem Anwender als Grundlage für die Erstellung von weiteren Bindemodulen mit Umsetztabelle für das TRANSLATE-CHARACTER-Format dienen kann.

Die Tabellen können auch über den Benutzerausgang TRANSLATE-CHARACTER (siehe [Seite 134](#)) bereitgestellt werden. Diese Tabellen haben dann Vorrang vor den Standard-Tabellen die in der Bibliothek mit dem logischen Namen SYSLNK.TAB abgelegt sind.

Folgende Randbedingungen sind beim Erstellen neuer Tabellen zu beachten:

- Der Tabellenwert X'00' ist kein gültiges Ersatzzeichen, sondern die Anzeige, dass das betreffende Zeichen nicht umgesetzt werden soll.
- Ist in der ersten Tabelle X'00' für einen Tabellenplatz eingetragen, so muss in der zweiten Tabelle auf demselben Tabellenplatz ebenfalls X'00' eingetragen sein.
- Am Code-Platz eines Zeichens, das als Ersatzzeichen verwendet wird, muss in der zweiten Tabelle X'00' eingetragen sein. Verkettungen sind nicht zugelassen. Folgende Gleichsetzungen sind daher z.B. nicht möglich:

'ä' = 'ae' **und** 'a' = 'A' **und** 'e' = 'E' oder

'ä' = 'a' **und** 'a' = 'A'.

Umsetzen mit TRANSLATE-CHARACTER

Mit den nachfolgenden Tabellen, sie werden standardmäßig für den CCS „EDF03DRV“ ausgeliefert, werden folgende Gleichsetzungen erreicht:

- Kleinbuchstaben = Großbuchstaben (Gleichsetzen von Zeichen),
- 'ä', 'Ä' = 'AE'; 'ö', 'Ö' = 'OE'; 'ü', 'Ü' = 'UE'; 'ß' = 'SS' (Umsetzen eines Zeichens auf zwei Ersatzzeichen).

Erste Tabelle für TRANSLATE-CHARACTER (EDF03DRV):

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1	00	00	00	00	00	00	00	00	C1	D1	00	00	00	00	00	00
2	00	00	00	00	00	00	00	00	C2	D2	E2	00	00	00	00	00
3	00	00	00	00	00	00	00	00	C3	D3	E3	00	00	00	00	00
4	00	00	00	00	00	00	00	00	C4	D4	E4	00	00	00	00	00
5	00	00	00	00	00	00	00	00	C5	D5	E5	00	00	00	00	00
6	00	00	00	00	00	00	00	00	C6	D6	E6	00	00	00	00	00
7	00	00	00	00	00	00	E2	00	C7	D7	E7	00	00	00	00	00
8	00	00	00	00	00	00	00	00	C8	D8	E8	00	00	00	00	00
9	00	00	00	00	00	00	00	00	C9	D9	E9	00	00	00	00	00
A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B	00	00	00	00	00	00	00	00	C1	00	C1	C1	00	00	00	C1
C	00	00	00	00	00	00	00	00	D6	00	D6	D6	00	00	00	00
D	00	00	00	00	00	00	00	00	E4	00	E4	E4	00	00	00	E4
E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F	00	00	00	00	D6	00	00	00	00	00	00	00	00	00	00	E2

Zweite Tabelle für TRANSLATE-CHARACTER (EDF03DRV):

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7	00	00	00	00	00	00	E2	00	00	00	00	00	00	00	00	00
8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B	00	00	00	00	00	00	00	00	C5	00	C5	C5	00	00	00	C5
C	00	00	00	00	00	00	00	00	C5	00	C5	C5	00	00	00	00
D	00	00	00	00	00	00	00	00	C5	00	C5	C5	00	00	00	C5
E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F	00	00	00	00	C5	00	00	00	00	00	00	00	00	00	00	E2

Format UNICODE-CHARACTER

Sortierfelder im UNICODE-CHARACTER-Format ermöglichen eine Textsortierung nach der Unicode Default Collation Table. SORT unterstützt derzeit den Unicode-Zeichensatz UTF-16.

Die Länge des Feldes wird in Byte angegeben.

Die maximale Länge des Feldes beträgt 256 Byte (128 Unicode-Zeichen).

Eine ungerade Länge und eine Länge größer als 256 Byte wird mit SRT1157 quittiert.

Bei Sätzen variabler Länge kann das Sortierfeld teilweise außerhalb des Satzes liegen.

Endet im Fall von Sätzen mit variabler Länge der Satz innerhalb eines Unicodezeichens (d.h. nach dem ersten Byte), so wird das unvollständige Unicode-Zeichen ganz ignoriert und mit dem Füllzeichen weiter verglichen.



Die Werte für die Sortierelemente (Unicode Default Collation Table), die unter der Web-Seite des Unicode-Konsortiums

<http://www.unicode.org/Public/UCA/4.0/allkeys-4.0.0.txt>

veröffentlicht sind, können sich ändern.

Nähere Informationen zu Unicode entnehmen Sie dem Handbuch „[Unicode im BS2000 \[17\]](#)“ oder dem [Abschnitt „Verwendung von Unicode-Zeichensätzen mit SORT“](#) auf [Seite 86](#).

2.3.2 Restfelder

Restfelder sind die formatneutralen Teile eines Eingabesatzes, die zwar in den Ausgabesatz übernommen werden, für die Sortierung aber nicht entscheidend sind. Restfelder spielen nur beim Auswahlsortieren (und somit auch beim Adresslistensortieren) eine Rolle.

Bei einer Auswahlortierung werden nur die Teile des Eingabesatzes übernommen, die in der SORT-RECORDS-Anweisung angegeben werden. Sollen nun Teile des Eingabesatzes zwar in den Ausgabesatz übernommen werden, jedoch nicht durch die Sortierung bewertet werden, definiert man sie als Restfeld.

Mit Hilfe der Restfelder ist es möglich, in der Ausgabedatei die Reihenfolge der Felder im Vergleich zur Eingabedatei zu verändern. Die Aneinanderreihung der einzelnen Felder ist durch die Reihenfolge der Angaben im FIELDS-Operanden der SORT-RECORDS-Anweisung festgelegt.

Restfelder werden durch die Angabe von *REMAINDER-EXPLICIT oder *REMAINDER-SYMBOLIC im FIELDS-Operanden definiert. Für ein Restfeld darf keine Formatangabe gemacht werden, wenn für das Restfeld keine Aufbereitungsmaske angegeben wurde. SORT gibt in diesem Fall eine Warnung aus und ignoriert die Angaben.

- Restfelder bei Sätzen mit variabler Länge

Bei Eingabesätzen variabler Länge darf ein Restfeld innerhalb des festen Teils mit einer beliebigen Position beginnen und enden. Soll der Ausgabesatz variable Länge haben (INTERNAL=*VARIABLE in der SET-RECORD-ATTRIBUTES-Anweisung), so darf sich das letzte Restfeld (letztes Feld des FIELDS-Operanden) als einziges Satzfeld auch auf den variablen Teil des Eingabesatzes beziehen. Alle anderen Satzfelder müssen sich auf den nicht variablen Teil des Satzes beziehen. SORT berechnet von sich aus die neue Satzlänge einschließlich des variablen Teils und fügt ein entsprechendes Satzlängenfeld ein, ohne dass der Benutzer dafür ein Restfeld angeben muss.

Beispiel

Aus einer Datei mit folgendem Satzaufbau soll eine Ausgabedatei erzeugt werden, die nur die Felder „PLZ“, „Prod.Nr.“ und „Menge“ enthält und nach „PLZ“ sortiert ist:

SL	PLZ	Lieferort	Prod.Nr.	Lieferdatum	Menge
1	5	10	18	27	36 40

Dazu ist „PLZ“ als Sortierfeld zu definieren und die Felder „Prod.Nr.“ sowie „Menge“ als Restfelder. Dies geschieht mit folgenden Anweisungen:

```

/start-sort
//sort-records -
//   fields=(*field-explicit(position=5,length=5), -      Sortierfeld „PLZ“
//             *remainder-explicit(position=18,length=9), - Restfeld „Prod.Nr.“
//             *remainder-explicit(position=36,length=5)), - Restfeld „Menge“
//   sort-type=*compound-record
//end

```

Aufbau der Ausgabesätze:

PLZ	Prod.Nr.	Menge
1	6	15 19

2.3.3 Konstantenfelder

Konstantenfelder werden beim Auswahl- und Adresslistensortieren ähnlich den Restfeldern für den Aufbau eines neuen Auswahlsatzes benutzt. Es sind Felder gleich bleibenden Inhaltes, die in den Ausgabesatz eingebracht werden. Sie können besonders beim Aufbau von Tabellen verwendet werden:

Durch Verwendung von jeweils nur ein Zeichen langen Konstantenfeldern, die den Trennstrich '|' enthalten, können Listen oder Tabellen optisch besser aufbereitet werden.

Wie Restfelder haben sie jedoch keinen Einfluss auf die Sortierung. Sie werden im FIELDS-Operanden der SORT-RECORDS-Anweisung vorgegeben. Folgende Arten von Konstanten sind möglich:

Dezimalzahl $\left\{ \begin{array}{l} +nn\dots n \\ -nn\dots n \end{array} \right\}$

Sedezimalstring X'ss...s'

Zeichenstring [C]'zz...z'

mit

n Ziffer 0 bis 9

s Sedezimalziffer 0 bis F

z alle darstellbaren Zeichen. Ein Hochkomma in einem Zeichenstring muss durch zwei aufeinander folgende Hochkommata wiedergegeben werden.

Dezimalzahlen müssen innerhalb der Grenzen -2^{31} bis $+2^{31}-1$ liegen. SORT verwandelt die Dezimalzahlen in eine Festpunktzahl und legt sie als Konstantenfeld mit einer Länge von 4 Byte im neuen Auswahlatz ab. Dezimalzahlen, die kein Vorzeichen haben, werden als positive Zahlen behandelt.

Sedezimal- und Zeichenstrings können eine Maximallänge von 256 Byte haben. Diese Strings setzt SORT in Konstantenfelder um, die durch das angegebene Zeichenmuster und deren Länge bestimmt werden.

Die zulässige gesamte Länge aller Konstantenfelder darf max. 4000 Byte sein.

Beispiel

Eingabesätze sollen von SORT über das Auswahlsortieren nach „Name“ sortiert werden. Als Restfeld soll das Feld „Telefon“ angegeben werden. In die Auswahlsätze sollen zur optischen Aufbereitung die Konstanten C'|_', C'|_ |' und C'|_ |' eingefügt werden.

Aufbau der Eingabesätze:

Vorname	Name	Telefon
1	14	23 32
ANTON	SCHERER	789543
ELVIRA	MAUS	453214
KLAUS	EPPLER	814300

Die dazugehörigen Anweisungen lauten:

```
/start-sort
//sort-records fields>(*constant-explicit(constant=c'|_'), -
//                      *field-explicit(position=14,length=9), -
//                      *constant-explicit(constant=c'|_ |'), -
//                      *remainder-explicit(position=23,length=10), -
//                      *constant-explicit(constant=c'|_ |')), -
//                      sort-type=*compound-record
//end
```

Ausgabe:

Aufbau der Ausgabesätze:

	Name	Telefon
1 3	12 15	25 26
	EPPLER	814300
	MAUS	453214
	SCHERER	789543

2.3.4 Summenfelder

Summenfelder werden in der SUM-RECORDS-Anweisung definiert und sind Satzfelder mit addierbaren Werten. **Summenfelder dürfen sich nicht mit anderen Summenfeldern oder mit Sortierfeldern überschneiden.**

Die Werte dieser Felder addiert SORT auf, wenn Sätze mit gleichen Sortierfeldern zusammengefasst werden sollen. Führt dabei eine Addition zu einem Überlauf, unterdrückt SORT alle Summenfeldadditionen für die zwei betroffenen Sätze und fasst die zwei beteiligten Sätze nicht zusammen. Deshalb sollte der Benutzer bei Gefahr eines Überlaufes die Summenfelder um eine angebbare Zahl von Byte nach vorn verlängern.

Formate und Längen von Summenfeldern:

Format	Formatbezeichnung	Länge in Byte
BINARY	binär	2, 4, 8
FIXED-POINT	Festpunkt	2, 4, 8
PACKED-DECIMAL	dezimal gepackt	1 - 16
ZONED-DECIMAL	dezimal entpackt	1 - 16

Die Felder werden dabei formatspezifisch entsprechend der folgenden Tabelle aufgefüllt:

Format	Art der Auffüllung	Füllzeichen
BINARY	linksbündig	Null
FIXED-POINT	linksbündig	Vorzeichen
PACKED-DECIMAL	linksbündig	Null
ZONED-DECIMAL	linksbündig	X'F0' (Null EBCDIC)

Beim Format ZONED-DECIMAL werden Leerzeichen (X'40') automatisch in Nullen (X'F0') umgewandelt. Außerdem wird bei diesem Format bei positiven Zahlen die Vorzeichenzone der letzten Ziffernstelle auf X'Fx' gesetzt ($0 \leq x \leq 9$).

Hinweise

- Beim Auswahlsortieren beziehen sich die Summenfeldpositionen auf den Auswahlsatz, wobei bei einer Satzformatwandlung auch das einzufügende oder wegzulassende Satzlängenfeld mit berücksichtigt werden muss. Summenfelder dürfen nur Rest- und Konstantenfelder sein. Summenfelder mit Verlängerung müssen sich jeweils auf die erste Position eines Rest- oder Konstantenfeldes beziehen.
- Die beim Adresslistensortieren mit *TAG-HEADER bzw. *TAG-COMPOUND von SORT zu erzeugenden Satzadressfelder (Wiedergewinnungsadressen) sind bei der Summenfeldpositionsangabe *nicht* zu berücksichtigen.

Beispiel

Die Datei LIEFERUNG enthält für jede ausgeführte Lieferung einen Satz mit folgendem Aufbau. Sie möchten eine Ausgabedatei erzeugen, die für jeden Lieferort einen Satz mit der Anzahl und dem Gesamtwert der Lieferungen enthält. Diese Datei soll nach „Lieferort“ sortiert sein.

SL	PLZ	Lieferort	Prod.Nr.	Lieferdatum	Auftragswert
1	5	10	28	37	47 56

Zu diesem Zweck definieren Sie „Lieferort“ als Sortierfeld und das Feld „Auftragswert“, das zum Gesamtwert der Lieferungen aufsummiert werden soll, als Restfeld. Weiterhin benötigen Sie im Auswahlatz ein Konstantenfeld, das den Wert 1 hat, um die Lieferungen je Ort zu zählen.

Sie geben also folgende Anweisungen ein:

```
/start-sort
//sort-records -
//   fields>(*field-explicit(position=10,length=18), -   Sortierfeld „Lieferort“
//           *remainder-explicit(position=47,length=10), - Restfeld
//                                           „Auftragswert“
//           *constant-explicit(constant=1)), -       Konstantenfeld
//   sort-type=*compound-record
```

Der Auswahlatz, auf den sich die Angaben in der SUM-RECORDS-Anweisung bei Auswahlortierung beziehen müssen, hat dann folgenden Aufbau:

Lieferort	Auftragswert	1
1	19	29 32

Das Konstantenfeld belegt dabei 4 Byte, da Dezimalzahlen von SORT standardmäßig als Festpunktzahl der Länge 4 Byte abgelegt werden. Mit der SUM-RECORDS-Anweisung definieren Sie nun die Bereiche, die SORT bei gleichem Sortierbegriff aufsummieren soll. Um Überläufe zu vermeiden, verlängern Sie das Summenfeld „Gesamtwert“ mit dem Operanden FIELD-EXTENSION. Das Summenfeld „Anzahl“ muss nicht verlängert werden, da es mit 4 Byte bereits groß genug ist.

```
//sum-records -
//   fields>(*field-explicit(position=19,length=10, -
//           format=*zoned-decimal,field-extension=3), -
//           *field-explicit(position=29,length=4))
//end
```

Aufbau der Ausgabesätze:

Lieferort	Gesamtwert	Anzahl
1	19	32 35

2.3.5 Maskenfelder

Beim Auswahlsortieren ist es möglich, Sortier-, Rest- und Summenfelder im Format BINARY, FIXED-POINT, PACKED-DECIMAL und ZONED-DECIMAL zum Druck aufzubereiten. Dazu wird eine Aufbereitungsmaske benutzt, die der Maske des Assemblerbefehls ED entspricht.

Die Aufbereitungsmaske hat folgendes Format:

C'xxx...x'

Als Maskenzeichen xxx...x sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Aufbereitungsmaske. Ersatzzeichen werden als Füllzeichen nicht umgesetzt.
- Steuerzeichen für die Aufbereitung

Die Steuerzeichen (Ziffernzeichen) des Assembler-ED-Befehls werden in der Aufbereitungsmaske durch Ersatzzeichen dargestellt. SORT setzt diese dann in die richtigen Steuerzeichen um.

Folgende Ersatzzeichen sind in der Aufbereitungsmaske anzugeben:

Steuerzeichen	Ersatzzeichen in Maske	Entspricht im ED-Befehl
Ziffernauswahl	# (Nummernzeichen)	X'20'
Geltender Ziffernbeginn	^ (Circumflex)	X'21'

- Einzufügende Zeichen

Alle Zeichen außer '#' und '^' werden als einzufügende Zeichen behandelt. Soll ein Hochkomma (') eingefügt werden, sind in der Aufbereitungsmaske zwei aufeinander folgende Hochkommata (") anzugeben.

Hinweis

Enthält die Aufbereitungsmaske eine *gerade* Anzahl von Steuerzeichen ('#' und/oder '^'), so fügt SORT unmittelbar vor dem ersten Steuerzeichen ein zusätzliches Steuerzeichen '#' ein. Dies ist bedingt durch die Arbeitsweise des Assembler-ED-Befehls.

Beispiel

Angegebene Maske: C' _#^#.#'

_	#	^	#	.	#
---	---	---	---	---	---

Von SORT erweiterte Maske:

_	#	#	^	#	.	#
---	---	---	---	---	---	---



eingefügtes Steuerzeichen

Einschränkungen

- Maskenfelder sind nur beim Auswahlsortieren erlaubt. Die Länge der Ausgabesätze ergibt sich als Summe der Länge der Maskenfelder und der übrigen Auswahlfelder.
- Maskenfelder sind nur für die Formate BINARY, FIXED-POINT, PACKED-DECIMAL und ZONED-DECIMAL erlaubt.
- Beim Format BINARY sind keine Bit-Positionsangaben bzw. Bit-Längen erlaubt.
- Bei Feldern mit Format BINARY oder FIXED-POINT, die länger als 4 Byte sind, werden nur die 4 niederwertigen Byte bei der Druckaufbereitung berücksichtigt und es wird eine Warnungsmeldung ausgegeben.
- Der Parameter ELIMINATE ist bei Maskenfeldern nicht erlaubt. Ist er angegeben, wird er ignoriert und es wird eine Warnungsmeldung ausgegeben.
- Sind Restfelder und Summenfelder als Maskenfelder angegeben und überschneiden sie sich, wird das Maskenfeld beim Restfeld ignoriert und eine Warnungsmeldung ausgegeben.

Wird SORT als Unterprogramm aufgerufen und werden die Anweisungen per Programmschnittstelle übergeben, so müssen die Steuerzeichen '#' und '^' der Druckaufbereitungsmasken stets den Code X'76' bzw. X'6A' besitzen, um als Steuerzeichen erkannt zu werden. Sie werden nicht anhand des CCS umgesetzt, sondern erhalten die Darstellung X'20' bzw. X'21' für den ED-Befehl des Assembler.

Regeln für die Abarbeitung der Aufbereitungsmaske

Die Aufbereitungsmaske eines Maskenfeldes wird nach denselben Regeln wie die Aufbereitungsmaske des Assembler-ED-Befehls abgearbeitet.

Es gelten folgende Regeln:

1. Das aufzubereitende Feld (Sendefeld) und die Aufbereitungsmaske werden von links nach rechts abgearbeitet.
2. Das Füllzeichen (erstes Zeichen der Aufbereitungsmaske) wird unverändert als erstes Zeichen in das Ausgabefeld übernommen.
3. Anschließend werden die Ziffern des Sendefeldes abhängig von den Zeichen der Aufbereitungsmaske (Steuerzeichen und einzufügende Zeichen) wie folgt in das Ausgabefeld übertragen:
 - Jede Ziffer des Sendefeldes wird an Stelle des zugehörigen Steuerzeichens in entpackter Form in das Ausgabefeld übertragen. Führende Nullen werden dabei so lange durch das Füllzeichen ersetzt, bis im Sendefeld die erste Ziffer ungleich Null erkannt wird oder in der Aufbereitungsmaske das Steuerzeichen '^' (geltender Ziffernbeginn) auftritt.
 - Einfügende Zeichen werden im Ausgabefeld solange durch das Füllzeichen ersetzt, bis im Sendefeld die erste Ziffer ungleich Null erkannt wird oder in der Aufbereitungsmaske das Steuerzeichen '^' auftritt. In weiterer Folge werden sie unverändert übernommen.
 - Bei positivem Vorzeichen im Sendefeld wird der Rest der Aufbereitungsmaske durch das Füllzeichen ersetzt. Bei negativem Vorzeichen wird er unverändert übertragen.

Sonderfälle

Enthält das Sendefeld mehr Ziffern als Steuerzeichen in der Aufbereitungsmaske, kürzt SORT das Sendefeld von links beginnend. Werden dabei Ziffern ungleich Null abgeschnitten, bricht SORT mit einer Fehlermeldung ab.

Enthält das Sendefeld weniger Ziffern als Steuerzeichen in der Aufbereitungsmaske, weist SORT führende Nullen zu.

Die folgenden Beispiele 1 - 4 zeigen die Abarbeitung der Aufbereitungsmaske.

Beispiel 1

Sendefeld (Format PACKED-DECIMAL):

0	0	1	2	0	2	0	D
---	---	---	---	---	---	---	---

Angegebene Maske:

*	#	#	#	#	#	.	#	#	-
---	---	---	---	---	---	---	---	---	---

Das Sendefeld wird wie folgt aufbereitet:

1. Als erstes Zeichen wird das Füllzeichen '*' in das linksbündige Byte des Ausgabefelds übertragen.
2. Führende Nullen werden durch das Füllzeichen ersetzt.
3. Die erste Ziffer ungleich Null im Sendefeld bewirkt, dass alle Ziffern in entpackter Form übertragen werden.
4. Das Satzzeichen '.' wird eingefügt.
5. Das negative Vorzeichen im Sendefeld wird als Minuszeichen in das Ausgabefeld übertragen.

1.	2.	3.	4.	3.	5.				
↓	↓	↓	↓	↓	↓				
*	*	*	1	2	0	.	2	0	-

Somit erhält man dieses Ausgabefeld:

Beispiel 2

Sendefeld (Format PACKED-DECIMAL):

0	0	0	0	0	6	8	C
---	---	---	---	---	---	---	---

Angegebene Maske:

_	#	#	.	#	^	#	,	#	#	-
---	---	---	---	---	---	---	---	---	---	---

Das Sendefeld wird wie folgt aufbereitet:

1. Als erstes Zeichen wird das Füllzeichen '_' in das linksbündige Byte des Ausgabefelds übertragen.
2. Führende Nullen werden so lange durch das Füllzeichen ersetzt, bis in der Maske das Steuerzeichen '^' auftritt. Das Satzzeichen '.' und das Steuerzeichen '^' werden ebenfalls durch das Füllzeichen ersetzt.
3. Die rechts des Steuerzeichens '^' stehende Null wird übertragen.
4. Das Satzzeichen ',' wird eingefügt.
5. Die Ziffern 6 und 8 werden übertragen.
6. Da die gepackte Zahl positiv ist, wird das Minuszeichen in der Aufbereitungsmaske durch das Füllzeichen ersetzt.

1.	2.					3.	4.	5.	6.	
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
_	_	_	_	_	_	0	,	6	8	_

Das Ausgabefeld hat dann den Inhalt:

Beispiel 3

Sendefeld (Format PACKED-DECIMAL):

0	0	0	0	0	0	3	8	C
---	---	---	---	---	---	---	---	---

Angegebene Maske:

_	#	^	#	.	#	#	#	-
---	---	---	---	---	---	---	---	---

Da in der angegebenen Maske 6 Steuerzeichen enthalten sind, ihre Anzahl also gerade ist, erweitert SORT die angegebene Maske um ein Ziffernauwahlzeichen '#'.

Von SORT erweiterte Maske:

_	#	#	^	#	.	#	#	#	-
---	---	---	---	---	---	---	---	---	---

Das Sendefeld wird wie folgt aufbereitet:

Die Anzahl der Ziffern im Sendefeld ist um 2 größer als die Anzahl der Steuerzeichen. Daher werden die beiden Nullen im ersten Byte des Sendefeldes abgeschnitten. Falls Ziffern ungleich Null im ersten Byte vorhanden wären, würde SORT mit einer Fehlermeldung abbrechen.

Ausgabefeld:

_	_	_	_	0	.	0	3	8	_
---	---	---	---	---	---	---	---	---	---

Beispiel 4Sendefeld
(Format FIXED-POINT):

F	F	F	6	8	F	A	D
---	---	---	---	---	---	---	---

Entspricht dezimal:

-618579

Angegebene Maske:

*	#	#	#	_	#	^	#	_	#	#	#	-
---	---	---	---	---	---	---	---	---	---	---	---	---

Die Anzahl der Steuerzeichen in der Maske ist um 3 größer als die Anzahl der Ziffern im Sendefeld (9 Steuerzeichen für 6 Ziffern). SORT füllt daher das Sendefeld linksbündig mit 3 führenden Nullen auf.

Erweitertes Sendefeld:

0	0	0	6	1	8	5	7	9	D
---	---	---	---	---	---	---	---	---	---

Da die gepackte Zahl negativ ist, wird das Minuszeichen aus der Aufbereitungsmaske in das Ausgabefeld übertragen.

Ausgabefeld:

*	*	*	*	*	6	1	8	_	5	7	9	-
---	---	---	---	---	---	---	---	---	---	---	---	---

2.3.6 Vergleichsfelder und Vergleichskonstanten

Vergleichsfelder und -konstanten dienen zur Definition von Bedingungen in der SELECT-INPUT-RECORDS-Anweisung. Damit wird gesteuert, welche Sätze in die Sortierung einbezogen oder von ihr ausgeschlossen werden sollen. In diesen Bedingungen kann ein Bereich des Eingabesatzes (Vergleichsfeld) entweder mit einem zweiten Vergleichsfeld oder mit einem festen Wert, einer Vergleichskonstanten, verglichen werden.

Vergleichsfelder

Für Vergleichsfelder sind die folgenden Formate und Längen zugelassen:

Format	Formatbezeichnung	Länge in Byte
BI	binär	1 - 256
CH	Zeichen	1 - 256
FI	Festpunkt	1 - 256
PD	dezimal gepackt	1 - 16
ZD	dezimal entpackt	1 - 16

Vergleichsfelder dürfen sich innerhalb eines Satzes grundsätzlich überschneiden. Der Benutzer muss allerdings dafür sorgen, dass dabei die Formatdarstellung nicht verletzt wird und keine Datenfehler durch die Vergleichsoperationen auftreten.

Vergleichsfelder dürfen sich auch uneingeschränkt mit Sortierfeldern oder Summenfeldern überschneiden. Relationen zwischen Vergleichsfeldern mit verschiedenen Formaten sind hingegen nur beschränkt möglich.

Vergleichskonstanten

Vergleichskonstanten haben die gleiche Form wie Konstantenfelder.

Dezimalzahl $\left\{ \begin{array}{l} +nn\dots n \\ -nn\dots n \end{array} \right\}$

Sedezimalstring X'ss...s'

Zeichenstring [C]'zz...z'

mit

n Ziffer 0 bis 9

s Sedezimalziffer 0 bis F

z alle darstellbaren Zeichen. Ein Hochkomma (') in einem Zeichenstring muss durch zwei aufeinander folgende Hochkommata (") dargestellt werden.

Dezimalzahlen müssen innerhalb der Grenzen -2^{31} bis $+2^{31}-1$ liegen.

Die gesamte Länge aller Vergleichskonstanten darf max. 4000 Byte sein.

Vergleichsrelationen

Bei Vergleichsrelationen muss das Format des ersten (oder einzigen) Vergleichsfeldes mit dem des zweiten bzw. mit der Art der Vergleichskonstanten verträglich sein. Die erlaubten Kombinationen sind in folgender Tabelle beschrieben.

Zulässige Formatkombinationen

1. Vergleichsfeld	2. Vergleichsfeld					Vergleichskonstante		
	BI	CH	FI	PD	ZD	Dezimalzahl	Sedezimalstring	Zeichenstring
BI	+	+	-	-	-	-	+	+
CH	+	+	-	-	-	-	+	+
FI	-	-	+	-	-	+	-	-
PD	-	-	-	+	+	+	-	-
ZD	-	-	-	+	+	+	-	-

Erläuterung:

- + Die Formatkombination ist zulässig.
- Die Formatkombination ist nicht zulässig. SORT meldet einen Fehler.

Sind bei Vergleichsrelationen die beteiligten Vergleichsfelder nicht gleich lang, so gleicht SORT das kürzere Feld gemäß nachfolgender Tabelle an das längere an. Die Vergleichsfelder mit numerischen Formaten (FI, PD, ZD) werden grundsätzlich nach links aufgefüllt und die Vergleichsfelder mit Stringformaten (BI, CH) entsprechend nach rechts.

Format	Ausrichtung	Auffüllung	Füllzeichen
FI	rechtsbündig	nach links	Vorzeichen
PD	rechtsbündig	nach links	Null
ZD	rechtsbündig	nach links	X'F0'(Null)
BI	linksbündig	nach rechts	Null
CH	linksbündig	nach rechts	X'40'(Leerzeichen)

Bei Sätzen variabler Länge dürfen sich die Vergleichsfelder bzw. Teile davon auch auf den „variablen Teil“ des Satzes erstrecken. Diese fiktiven Vergleichsfelder werden von SORT abhängig vom Format entsprechend obiger Tabelle aufgefüllt.

Auch Vergleichskonstanten werden ggf. in der Länge an das zugehörige erste Vergleichsfeld angeglichen. Die Verlängerung oder Verkürzung erfolgt bei numerischen Konstanten nach bzw. von links, bei Zeichenkonstanten nach bzw. von rechts.

Beispiel

In der Ausgabe dieses Sortierlaufs sollen nur die Sätze der Eingabedatei enthalten sein, die in den ersten 5 Byte C'@@@@@' enthalten, und die im Feld ab Byte 7 in der Länge 8 und im Feld ab Byte 22 in gleicher Länge nicht den gleichen Inhalt haben.

```
/start-sort
/sort-records ...
//select-input-records condition=(1,5),eq,c'@@@@@',and,not((7,8),eq,(22,8))
//end
```

2.3.7 Symbolische Namen

Sortier-, Rest-, Summen- und Vergleichsfeldern, sowie Einfüge-, Vergleichskonstanten und Druckaufbereitungsmasken sind über die Anweisung ADD-SYMBOLIC-NAMES symbolische Namen zuordenbar, die in nachfolgenden Anweisungen verwendet werden können.

Symbolische Namen ermöglichen es, den angesprochenen Objekten eine Bedeutung zuzuordnen und somit die Bedienung zu erleichtern und die Sicherheit einer Anwendung zu erhöhen.

Die Zuordnung erfolgt bei Feldern über die Angabe von Position, Länge und Format, bei Konstanten und Masken über deren Wert.

Maximal können 255 verschiedene symbolische Namen verwendet werden. Kleinbuchstaben werden in Großbuchstaben umgesetzt. Die Namenslänge ist mit 20 Zeichen begrenzt.

Ein symbolischer Name muss mit der Anweisung ADD-SYMBOLIC-NAMES entsprechend seiner späteren Verwendung als Feld, Konstante oder Druckaufbereitungsmaske vereinbart werden, bevor er verwendet werden kann.

Jeder symbolische Name darf in der Anweisung ADD-SYMBOLIC-NAMES nur einmal verwendet werden.

Beispiel

Inhalt der Datei RESTAURANT.SAM.FIX

Orlando's	Thompson Street 62	220061	Italian
Java	Hope Street 51	522221	Indonesian
Golden Fleece	Arran Street 44	242437	Yugoslavian
Le Gourmet	Lime Street 46	505397	French
Palenque Mexico	Millwood Drive 2	980149	Mexican
Strawberry	Sauchiehall Street 8	595521	Vegetarian
Persepolis	Salford Square 20	597004	Persian
Vietnam	Thurston Street 47	522518	Vietnamese
Chayota's	Thurston Street 60	292742	Japanese
Willi's Bar	Westland Street 113	748293	German
1	21	48	56

Mit der ADD-SYMBOLIC-NAMES-Anweisung werden symbolische Namen definiert und Feldern zugeordnet. In der SORT-RECORDS-Anweisung werden die Sortierfelder über diese symbolischen Namen angesprochen.

```
/start-sort
//assign-files input-files=restaurant.sam.fix,output-file=restaurant.sort
//add-symbolic-names fields=(restaurantname(position=1,length=20), -
//                               adresse(position=21,length=27))
//sort-records fields>(*field-symbolic(sort-field-name=restaurantname), -
//                               *field-symbolic(sort-field-name=adresse))
//end
```

Inhalt der Ausgabedatei RESTAURANT.SORT

Chayota's	Thurston Street 60
Golden Fleece	Arran Street 44
Java	Hope Street 51
Le Gourmet	Lime Street 46
Orlando's	Thompson Street 62
Palenque Mexico	Millwood Drive 2
Persepolis	Salford Square 20
Strawberry	Sauchiehall Street 8
Vietnam	Thurston Street 47
Willi's Bar	Westland Street 113

2.3.8 Tabelle für Überschneidungen verschiedener Feldtypen

	Sortierfeld	Summenfeld	Restfeld	Vergleichs- feld	Konstanten- feld
Sortierfeld	*	-	+	+	-
Summenfeld	-	-	+	+	+
Restfeld	+	+	+	+	-
Vergleichsfeld	+	+	+	+	-
Konstantenfeld	-	+	-	-	-

Erläuterung:

- * erlaubte Überschneidungen sind beim jeweiligen Feldtyp formatspezifisch angegeben
- + Überschneidungen sind erlaubt
- Überschneidungen sind nicht erlaubt

Beispiel für Überschneidungen von Feldtypen

Die Eingabe an SORT sollen Sätze mit folgendem Aufbau und Inhalt sein:

Name	Vorname	Straße	Telefon
1	10	19	38 45
HILLER	PETER	CHURCH ROAD 2	700781
BECKER	THEODORE	MARKET SQUARE 3	378543

SORT soll diese Eingabesätze auf folgende Weise bearbeiten:

Durch Auswahlsortieren sollen Auswahlsätze entstehen, die nach dem Sortierbegriff „Name – Vorname“ sortiert sind und das Feld „Telefon“ als Restfeld enthalten. Zusätzlich soll dem Satz ein weiteres Feld „IT“ vorangestellt werden, das die Initialen der Namen enthält. Es setzt sich aus zwei Restfeldern zusammen, die jeweils die Länge 1 haben und deren Position mit den Positionen der beiden Sortierfelder übereinstimmt.

Die dazugehörigen Anweisungen lauten:

```
/start-sort
//sort-records fields>(*remainder-explicit(position=1,length=1), -
//                      *remainder-explicit(position=10,length=1), -
//                      *field-explicit(position=1,length=18), -
//                      *remainder-explicit(position=38,length=8)), -
//                      sort-type=*compound-record
//end
```

Ausgabe:

IT	Name	Vorname	Telefon
1	3	12	21 28

BT BECKER THEODORE 378543

HP HILLER PETER 700781

Dieses Beispiel enthält folgende Feldüberschneidungen:

Das Sortierfeld überschneidet sich am ersten Byte mit dem an gleicher Stelle befindlichen Restfeld. Eine Überschneidung erfolgt auch ab Byte 1 bis Byte 9 mit dem Vergleichsfeld. Eine weitere Überschneidung ergibt sich mit dem auf Byte 10 befindlichen zweiten Restfeld.

2.4 Satzbearbeitung und Satzveränderung mit SORT

Während eines Sortier-/Mischlaufs können die eingelesenen Sätze intern auf folgende Weise bearbeitet oder verändert werden (siehe nächste Seite, Bild 6):

- Satzauswahl bei der Eingabe
- Satzformat- und Satzlängenänderung aus Angaben der SET-RECORD-ATTRIBUTES-Anweisung
- Satzlängenänderung aus Angaben der SORT-RECORDS-Anweisung beim Auswahl-sortieren
- Satzlängenänderung aus Angaben der SUM-RECORDS-Anweisung bei Verlängerung der Summenfelder
- Inhaltliche Änderung
- Summierung von Sätzen (SUM-RECORDS-Anweisung)

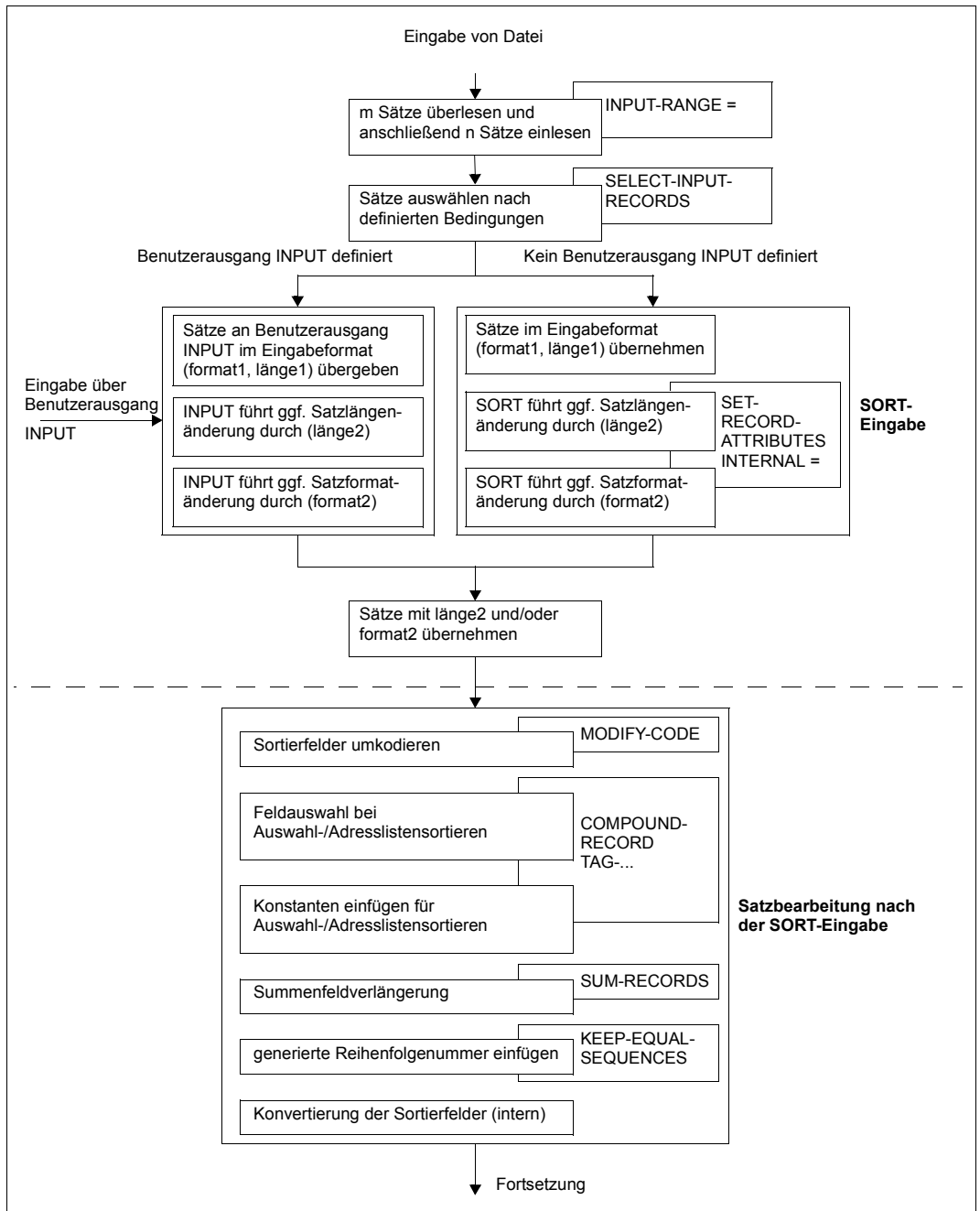


Bild 6: Satzbearbeitung und Satzänderung im SORT

(Teil 1 von 2)

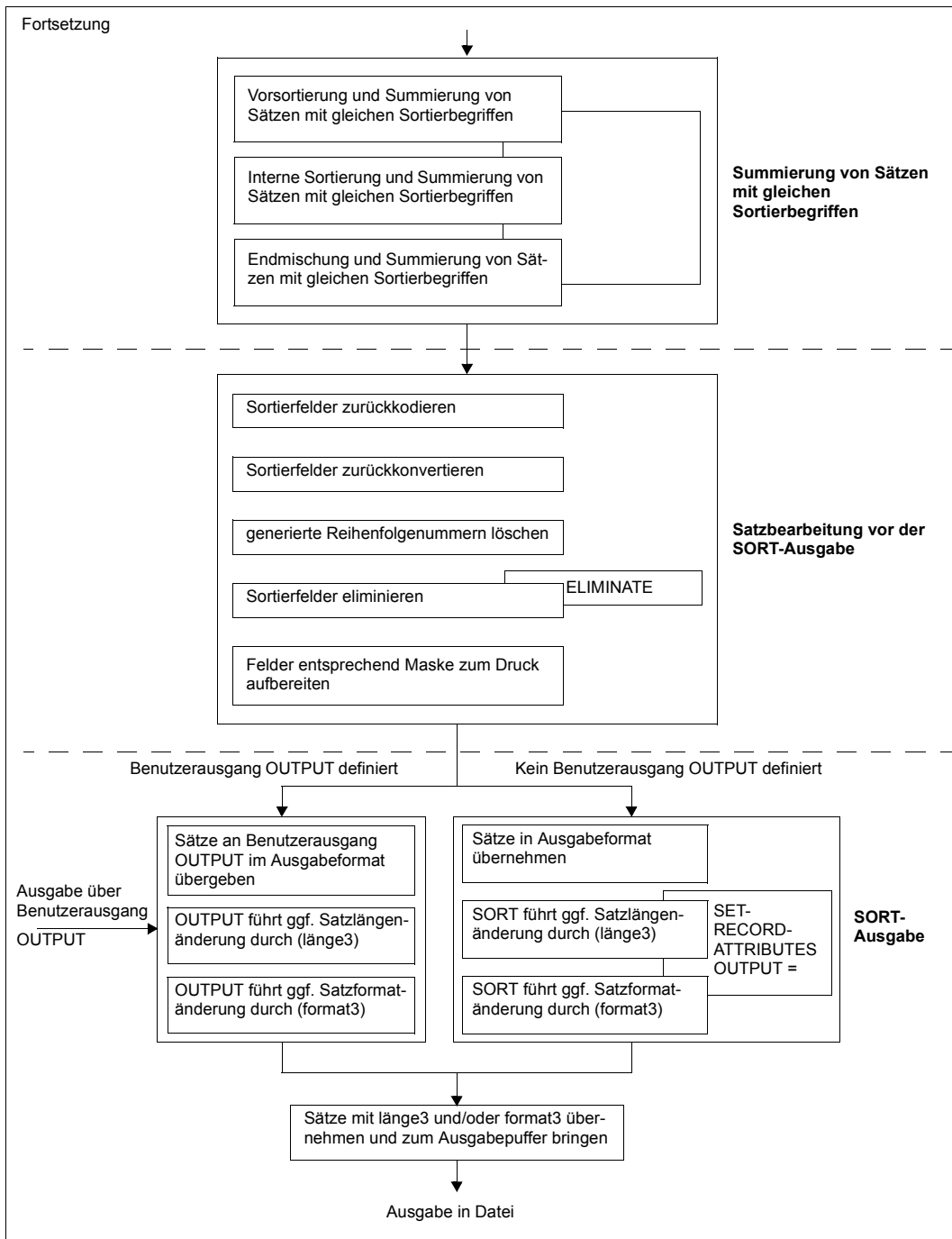


Bild 6: Satzbearbeitung und Satzänderung im SORT

(Teil 2 von 2)

Satzauswahl bei der Eingabe

Der Benutzer kann Sätze der Eingabe von der Sortierung/Mischung ausschließen bzw. einbeziehen über

- SORT-RECORDS-Anweisung, Operand INPUT-RANGE
Mit dem INPUT-RANGE-Operanden wird bei der Sortierung angegeben, wie viele Sätze vom Beginn der Eingabedatei an überlesen und wie viele Sätze anschließend sortiert werden (siehe SORT-RECORDS-Anweisung, [Seite 179](#)).
- SELECT-INPUT-RECORDS-Anweisung
Sätze der Eingabedatei, die die Bedingung der SELECT-INPUT-RECORDS-Anweisung erfüllen, werden in die Sortierung/Mischung einbezogen (siehe SELECT-INPUT-RECORDS-Anweisung, [Seite 158](#)).
- Benutzerausgang INPUT
Über eine Benutzerroutine für den Benutzerausgang INPUT (ASSIGN-EXITS-Anweisung) können Sätze übergeben, eingefügt und von der Sortierung/Mischung ausgeschlossen werden (siehe [Kapitel „Benutzerausgänge von SORT“ auf Seite 237](#)).

Satzformat- und Satzlängenänderung aus Angaben der SET-RECORD-ATTRIBUTES-Anweisung

Satzformat und Satzlänge der Sätze der Eingabedatei (Ist-Zustand) können von SORT nicht geändert werden. Änderungen sind nur möglich beim Übergang von

- der SORT-Eingabe zur internen SORT-Verarbeitung (internes Satzformat und interne Satzlänge).
- der internen SORT-Verarbeitung zur SORT-Ausgabe (Satzformat und Satzlänge der Ausgabesätze).

Diese Satzformat- und Satzlängenänderungen werden mit der SET-RECORD-ATTRIBUTES-Anweisung festgelegt (siehe [Seite 162](#)). Es gelten folgende Punkte:

- Ist-Zustand
Satzformat und Länge der Sätze der Eingabedatei sind primär durch das ADD-FILE-LINK-Kommando bzw. den Katalogeintrag (RECORD-FORMAT, RECORD-SIZE) für die Eingabedatei festgelegt. Sind diese Angaben nicht vorhanden, (z.B. Eingaben über den Benutzerausgang INPUT) müssen Satzformat und Länge der Eingabesätze mit einer SET-RECORD-ATTRIBUTES-Anweisung definiert werden.

Wird eine SET-RECORD-ATTRIBUTES-Anweisung angegeben, so bestimmt

- die Formatangabe *VARIABLE() bzw. *FIXED() im INPUT-Operanden das Satzformat der Eingabesätze.
- die Längenangabe im Operanden MAXIMUM-RECORD-SIZE bzw. RECORD-SIZE die Satzlänge der Eingabesätze.

Wurden trotz des Vorhandenseins eines Katalogeintrages Angaben in der SET-RECORD-ATTRIBUTES-Anweisung gemacht und stimmen diese nicht mit den Merkmalen der Eingabedatei überein, so übernimmt SORT die Merkmale der Eingabedatei und gibt eine Warnungsmeldung aus.

- Eingabesätze variabler Länge
Bei Sätzen variabler Länge benützt SORT als Satzlänge der Eingabesätze den **Maximalwert** der RECORD-SIZE-Angaben aller Eingabedateien und Angabe im MAXIMUM-RECORD-SIZE-Operanden der SET-RECORD-ATTRIBUTES-Anweisung. Fehlen diese Angaben, wird als Satzlänge der Eingabesätze die BUFFER-LENGTH-Angabe übernommen.
- Dateimerkmale der Ausgabedatei
Enthält das ADD-FILE-LINK-Kommando bzw. der Katalogeintrag bereits Satzformat und Satzlänge für die Ausgabedatei, müssen diese bei fester Satzlänge mit den in der SET-RECORD-ATTRIBUTES-Anweisung angegebenen bzw. den von SORT berechneten Werten übereinstimmen. Sind die Werte verschieden, wird eine Fehlermeldung ausgegeben und SORT fehlerhaft beendet. Bei variabler Satzlänge wird die von SORT ermittelte maximale Satzlänge in den Katalogeintrag für die Ausgabedatei eingetragen.
- Satzformat-/Satzlängenänderungen mittels SET-RECORD-ATTRIBUTES-Anweisung
Interne Satzformatänderungen werden mit dem Operanden INTERNAL=*VARIABLE() bzw. *FIXED() der SET-RECORD-ATTRIBUTES-Anweisung festgelegt.
Satzformat-/Satzlängenänderungen der Ausgabesätze werden mit dem Operanden OUTPUT=*VARIABLE() bzw. *FIXED() der SET-RECORD-ATTRIBUTES-Anweisung festgelegt.
- Satzformat-/Satzlängenänderungen bei definierten Benutzerausgängen
Sind die Benutzerausgänge INPUT und/oder OUTPUT definiert, so führt SORT **keine** Satzformat-/Satzlängenänderung durch. Die an die Benutzerausgänge angeschlossenen Routinen müssen diese Änderungen selbst durchführen.
Zuständig ist:
 - Benutzerausgang INPUT für das interne Satzformat und die interne Satzlänge.
 - Benutzerausgang OUTPUT für Satzformat und Satzlänge der Ausgabesätze.

- Satzformat-/Satzlängenänderungen bei undefinierten Benutzerausgängen
Sind keine Benutzerausgänge INPUT und/oder OUTPUT definiert, so führt SORT selbst die Satzformat-/Satzlängenänderungen wie folgt durch:
 - Satzformatänderung
Bei Formatänderung von fixer auf variable Satzlänge berechnet SORT die Satzlänge neu und stellt sie im Satzlängengeld (4 Byte) dem umgewandelten Satz voran. Bei Formatänderung von variabler auf fixe Satzlänge schneidet SORT das Satzlängengeld (4 Byte) am Anfang des Satzes ab. Die umgewandelten Sätze werden bis zur maximalen Satzlänge (Interne bzw. Ausgabesätze) entsprechend den Angaben im FILLER-Operanden der SET-RECORD-ATTRIBUTES-Anweisung aufgefüllt.
 - Satzlängenänderung
Satzlängenänderungen, die SORT durchführt, bewirken Folgendes:
Sätze fester Länge werden beim Verkürzen abgeschnitten, beim Verlängern entsprechend den Angaben im FILLER-Operanden der SET-RECORD-ATTRIBUTES-Anweisung aufgefüllt. Sätze variabler Länge werden nur dann verkürzt (abgeschnitten und Satzlängengeld korrigiert), wenn sie die neue Länge überschreiten. Wird dagegen bei Sätzen variabler Länge die maximale Satzlänge nicht überschritten, bleibt die vorhandene Satzlänge unverändert. Die Anzahl gekürzter Sätze wird mit der Meldung SRT1054 angezeigt.

Satzlängenänderung aus Angaben der SORT-RECORDS-Anweisung

- ELIMINATE-Operand
Wird der ELIMINATE-Operand in der Sortierfeldbeschreibung angegeben, so verkürzt SORT den Satz um die Länge dieses Sortierfeldes.
- SORT-TYPE-Operand
Beim Auswahlsortieren (SORT-TYPE=*COMPOUND-RECORD) und Adresslisten-sortieren (SORT-TYPE=*TAG-COMPOUND / *TAG-TRAILER / *TAG-HEADER) wird die Satzlänge verändert. Die neue Satzlänge des Auswahlsatzes ist die Summe aus den Längen der Sortierfelder, Konstantenfelder, Restfelder und des Adressfeldes.

Satzlängenänderung aus Angaben der SUM-RECORDS-Anweisung

In der SUM-RECORDS-Anweisung kann bei der Feldbeschreibung eine Feldverlängerung angegeben werden. Der Satz wird dann um die angegebene Länge verlängert.

Inhaltliche Veränderung

Inhaltliche Änderungen können zurückgeführt werden auf:

- Auswahlsatzangaben in der SORT-RECORDS-Anweisung.
- Umcodierungen von Sortierfeldern des Formates PHYSICAL-TRANSLATE.
- Satzbearbeitung durch die Benutzerausgänge INPUT und/oder OUTPUT.
- ELIMINATE-Angabe in der Sortierfeldbeschreibung der Anweisung SORT-RECORDS bzw. MERGE-RECORDS.
- Auffüllen von Sätzen entsprechend den Angaben des FILLER-Operanden der Anweisung SET-RECORD-ATTRIBUTES bei Satzformat- und Satzlängenänderung.
- Druckaufbereitung mit Aufbereitungsmasken.

Summierung von Sätzen

Auf Grund der Anweisung SUM-RECORDS können Sätze mit gleichen Sortierbegriffen bei identischem Inhalt der Sortierfelder zusammengezogen werden, wobei die angegebenen Summenfelder addiert werden. Diese Satzverdichtung erstreckt sich über die ganze Sortierung und Mischung hinweg, d.h. während der Vorsortierungsphase, internen Mischung und Endmischung (Siehe [Abschnitt „Summenfelder“ auf Seite 58](#)).

2.5 SORT in XS-Umgebung (31-Bit-Adressierung)

SORT ist im 24- und 31-Bit-Adressierungsmodus ablauffähig. Maßgebend ist jeweils der Adressierungsmodus, mit dem SORT über das Kommando „START-SORT“ gestartet oder von einem Hauptprogramm als Unterprogramm aufgerufen wird. SORT selbst schaltet den Hardware-Adressierungsmodus nicht um.

Für den Adressierungsmodus ist Folgendes zu beachten:

- im 24-Bit-Adressierungsmodus

Benutzerrountinen, die von SORT aufgerufen werden, dürfen noch im alten (nicht umgestellten) Zustand benutzt werden (entspricht AMODE/RMODE=24, PARMOD=24). Sie dürfen aber auch ganz oder teilweise umgestellt sein (entspricht AMODE/RMODE=ANY, PARMOD=31). SORT stellt sich bei der PARMOD-relevanten Schnittstelle der Benutzerroutine EXLST-FOR-INPUT/EXLST-FOR-OUTPUT auf den angegebenen EXLST-Makro ein und generiert die zugehörigen FCBs mit passendem PARMOD.

Die Funktionen von SORT dürfen uneingeschränkt benutzt werden.

- im 31-Bit-Adressierungsmodus

Die Benutzerausgänge INPUT, OUTPUT und EXTERNAL-COMPARE müssen mit PARAMETER-MODE=ANY verwendet werden.

Für alle SORT-Schnittstellen zu den Benutzerausgängen gilt die 31-Bit-Adressierung.

Bei den Benutzerausgängen EXLST-FOR-INPUT/EXLST-FOR-OUTPUT müssen die EXLST-Makros mit PARMOD=31 übersetzt sein.

Im 31-Bit-Adressierungsmodus dürfen FCB-Verweistabellen nicht mehr verwendet werden.

Sind die Konventionen für die SORT-XS-Umstellung nicht erfüllt, geben SORT oder der dynamische Bindelader Fehlermeldungen aus.

2.6 Verwendung erweiterter Zeichensätze mit SORT

Rechenanlagen (Hosts) und Datensichtstationen arbeiten mit je einem Satz von Buchstaben, Ziffern und Zeichen, aus denen Wörter und andere elementare Bestandteile einer Sprache aufgebaut sind, dem so genannten **Zeichensatz (character set)**.

Durch die Erweiterung dieser Zeichensätze können landesspezifische Zeichendarstellungen, wie z.B. Umlaute (deutsch) oder Akzente (französisch), innerhalb eines Zeichensatzes gleichzeitig angeboten werden.

Ein **codierter Zeichensatz (coded character set, CCS)** ist die eindeutige Darstellung der Zeichen eines Zeichensatzes in binärer Form. Der Inhalt eines codierten Zeichensatzes und seine Regeln, wie z.B. die Sortierreihenfolge und Konvertierungsvorschriften, sind durch internationale Normen festgelegt.

Beispiel: Das Zeichen 'ä' ist im codierten Zeichensatz EBCDIC.DF.03 (deutsche Referenzversion) durch das Byte X'FB', in EBCDIC.DF.04-1 durch X'43' dargestellt.

Jeder codierte Zeichensatz (auch kurz: Code) wird durch seinen eindeutigen Namen (**coded character set name, CCSN**) bestimmt.

Beispiel: der Code EBCDIC.DF.03 (Internationale Referenzversion) hat den Namen „EDF03IRV“.

Eine Liste der existierenden Codes befindet sich im Anhang des Handbuchs „XHCS“ ([10]).

Erweiterte Codes ergänzen die bestehenden

7-bit-Codes EBCDIC.DF.03 für Hosts und
ISO646 für Datensichtstationen
mit etwa 90 genutzten Zeichen

um

8-bit-Codes EBCDIC.DF.04-x für Hosts und
ISO8859-x für Datensichtstationen
mit etwa 190 genutzten Zeichen.

Für Anweisungsdateien wird die Angabe eines 8-bit-Codes ISO8859-x nicht unterstützt.

Voraussetzungen

Zur Erstellung erweiterter Codes im Host und zur Datenübertragung zwischen Host und Datensichtstation wird das Softwareprodukt **XHCS (eXtended Host Code Support)** benötigt. Im Handbuch zu XHCS [10] befindet sich eine detaillierte Beschreibung der Prinzipien und Funktionen von XHCS sowie eine Auflistung der Code-Tabellen und der Namen von Standard-Codes.

Als Hardware für die Ein-/Ausgabe erweiterter Zeichensätze auf Datensichtstationen werden so genannte 8-bit-Terminals benötigt.

Die Prüfung der 8-bit-Fähigkeit von Datensichtstationen erfolgt mit dem Softwareprodukt VTSU.

SORT benötigt und verwendet XHCS-Funktionen, wenn

- Sortierfelder das Format EXTENDED-CHARACTER oder TRANSLATE-CHARACTER besitzen oder
- in den Anweisungen Zeichenkonstanten enthalten sind und sich der CCSN der Datensätze vom CCSN der Anweisungen unterscheidet.

Benötigt SORT XHCS-Funktionen und ist XHCS nicht verfügbar, wird der Sortierlauf mit der Meldung SRT1257 abgebrochen.

2.6.1 SORT-spezifische Anwendung erweiterter Zeichensätze

Die Sortierung von Datensätzen kann unter Verwendung erweiterter Codes durchgeführt werden. Die Verfahren und Randbedingungen dafür werden im Folgenden beschrieben.

2.6.1.1 Sortieren mit erweiterten Codes

SORT bietet dazu die Formate EXTENDED-CHARACTER und TRANSLATE-CHARACTER. Mit diesen Formaten ist eine Sortierung von Datensätzen in der Sortierreihenfolge des betreffenden Codes möglich.

Sie können in den SORT-Anweisungen ADD-SYMBOLIC-NAMES, MERGE-RECORDS, SORT-RECORDS sowie dem Kommando SORT-FILE als Formate für Sortierfelder angegeben werden.

Das Beispiel auf [Seite 326](#) zeigt eine Anwendung der beiden Formate.

Für die Überschneidung von Sortierfeldern gelten die Einschränkungen der Tabelle „Sortierfeldüberschneidungen“, [Seite 42](#). Die Länge des Sortierfeldes beträgt 1-256 Byte.

SORT versucht, den CCSN zur Sortierung der Datensätze durch Auswertung der folgenden Quellen zu ermitteln:

- Katalogeintrag der Eingabedatei
- SET-RECORD-ATTRIBUTES-Anweisung des Sortierlaufes (keine Eingabedatei vorhanden)
- Ist keine Eingabedatei vorhanden und wird auch kein CCSN über die SET-RECORD-ATTRIBUTES-Anweisung angegeben, so verwendet SORT den Code EDF03IRV.

Wird der CCSN eines Codes ermittelt, der nicht in XHCS definiert ist, so führt dies zum Abbruch des Sortierlaufes mit der Meldung SRT1258, wenn SORT Funktionen von XHCS benötigt.

Werden keine XHCS-Funktionen benötigt, wird der ermittelte CCSN ungeprüft als CCS-Attribut in den Katalogeintrag der Ausgabedatei übernommen.

Bei einer ISAM-Ausgabedatei dürfen die Formate EXTENDED-CHARACTER und TRANSLATE-CHARACTER nicht für Sortierfelder verwendet werden, die den ISAM-Schlüssel bilden. Die ISAM-Schlüssel müssen im Sinne des Host Codes aufsteigend sortiert sein, eine Sortierung in einem anderen Code ist nicht zulässig. Im Fehlerfall wird der Sortierlauf mit der Meldung SRT1261 abgebrochen.

2.6.1.2 CCSN-Angabe in den Dateien von SORT

Abhängig von der Art der Datei wird die CCSN-Angabe von SORT unterschiedlich behandelt:

– Eingabedateien:

Erfolgt die Eingabe der Datensätze aus einer Eingabedatei, so übernimmt SORT den CCSN der Eingabedatei zur Sortierung der Datensätze.

Bei Verwendung mehrerer Eingabedateien müssen sie denselben CCSN besitzen, da SORT keine Datenkonvertierung zur Anpassung unterschiedlicher Codes in den Eingabedateien durchführt. Dummy-Eingabedateien verhalten sich dabei neutral. Im Fehlerfall wird der Sortierlauf mit Ausgabe der Meldung SRT1254 abgebrochen.

Anmerkung: Unschädlich ist die gemischte Verwendung von „EDF03IRV“ und „kein CCSN definiert“. Der CCSN der Datensätze ist in diesem Fall undefiniert.

Existieren Eingabedatei(en) und eine CCSN-Angabe in der SET-RECORD-ATTRIBUTES-Anweisung, so wird der CCSN der Eingabedatei(en) zur Sortierung verwendet, die CCSN-Angabe in der SET-RECORD-ATTRIBUTES-Anweisung wird ignoriert, die Meldung SRT1256 wird ausgegeben.

– Ausgabedatei:

Der CCSN der Ausgabedatei wird von SORT vergeben, eine Benutzerangabe für die Ausgabedatei wird ignoriert. Der CCSN der Ausgabedatei wird wie der CCSN der Datensätze bestimmt.

Liegt eine Eingabedatei vor, wird der CCSN dieser Datei benutzt.

Liegen mehrere Eingabedateien vor, wird der CCSN der ersten Datei benutzt.

Ist die erste Datei eine Dummy-Datei, so wird der CCSN der ersten Nicht-Dummy-Datei benutzt.

Sind alle Eingabedateien Dummy führt dies zu keinem CCSN für die Ausgabedatei.

– Anweisungsdatei:

Der CCSN der Anweisungsdatei wird von SORT beim Einlesen der Anweisungen ausgewertet (siehe [Abschnitt „Umsetzung von Zeichenkonstanten in den Code der Datensätze“ auf Seite 84](#)).

- Hilfs-, Arbeits- und Fixpunktdateien:
Für diese Dateien wird von SORT kein CCS-Attribut vergeben, eine Benutzerangabe wird ignoriert.

2.6.1.3 Explizite CCSN-Angabe für Datensätze und Ausgabedatei

Ist keine Eingabedatei vorhanden, d.h. die Eingabe der Datensätze erfolgt über den Benutzerausgang INPUT oder über die Zugriffsmethode SORTZM, kann der Name des CCS der Datensätze über die Anweisung SET-RECORD-ATTRIBUTES, Operand CODED-CHARACTER-SET=<name 1..8>, eingegeben werden.

Der CCSN wird für die Ausgabedatei übernommen. Ist keine Eingabedatei vorhanden und wird auch kein CCSN über die SET-RECORD-ATTRIBUTES-Anweisung angegeben, so verwendet SORT den Code EDF03IRV.

2.6.1.4 Umsetzung von Zeichenkonstanten in den Code der Datensätze

In den nachfolgenden SORT-Anweisungen können Konstanten als Zeichenfolge definiert werden (z.B. SET-RECORD-ATTRIBUTES FILLER='#'). Diese werden durch SORT vom CCS der Konstanten in das CCS der Datensätze konvertiert, bevor diese Konstanten in den Ausgabesatz eingefügt oder mit Feldern des Eingabesatzes verglichen werden. Haben die Datensätze ein Unicode-CCSN, so wird die Konvertierung mit SRT1260 abgewiesen.

Bei folgenden Anweisungen können Konstanten als Zeichenfolge angegeben werden:

ADD-SYMBOLIC-NAMES	CONSTANTS=list-poss(256):<name(VALUE=<c-string>)>
MODIFY-CODE	SEQUENCES=list-poss(256):<c-string>
SELECT-INPUT-RECORDS	CONDITION=<text>
SET-RECORD-ATTRIBUTES	FILLER=<c-string>
SET-SORT-OPTIONS	IGNORE-CHARACTER=<c-string 1..1 >
SORT-RECORDS	FIELDS=*CONSTANT-EXPLICIT(CONSTANT=<c-string>)
SORT-RECORDS	FIELDS=*FIELD-EXPLICIT(PRINT-MASK=<c-string>)
SUM-RECORDS	FIELDS=*FIELD-EXPLICIT(PRINT-MASK=<c-string>).

Bei Eingabe der Anweisungen (mit Zeichenkonstanten) an SORT über SYSDDTA wird der CCSN der Zeichenkonstanten von SORT ermittelt.

Wird SORT als Unterprogramm aufgerufen und werden die Anweisungen (mit Zeichenkonstanten) per Programmschnittstelle übergeben, so kann der CCSN der Zeichenkonstanten in den Anweisungen durch die SORT-Anweisung SET-SORT-OPTIONS, Operand STATEMENT-CCSN, vereinbart werden.

Fehlt die CCSN-Angabe in der SET-SORT-OPTIONS-Anweisung, verwendet SORT den Code EDF03IRV.

2.6.1.5 Randbedingungen für Maskenfelder

Wird SORT als Unterprogramm aufgerufen und werden die Anweisungen per Programmschnittstelle übergeben, so müssen die Steuerzeichen '#' und '^' der Druckaufbereitungsmasken stets den Code X'76' bzw. X'6A' besitzen, um als Steuerzeichen erkannt zu werden. Sie werden nicht anhand des CCS umgesetzt, sondern erhalten die Darstellung X'20' bzw. X'21' für den ED-Befehl des Assembler.

2.6.1.6 Verwendung erweiterter Zeichensätze in Benutzerausgängen

An den Benutzerausgängen INPUT, OUTPUT und EXTERNAL-COMPARE wird dem Benutzer der CCSN der Datensätze übergeben (31-Bit-Schnittstelle). Der CCSN dient nur zur Benutzerinformation, bei EXTERNAL-COMPARE ist ein Sortierfeldvergleich nur unter Berücksichtigung des übergebenen CCSN's sinnvoll.

2.7 Verwendung von Unicode-Zeichensätzen mit SORT

Unicode ist ein genormter alphanumerischer Zeichensatz und fasst alle weltweit bekannten Textzeichen in einem einzigen Zeichensatz zusammen. SORT unterstützt derzeit die Unicode-Variante UTF-16 (UCS Transformation Format 16 Bit), bei dem jedes Zeichen durch zwei Byte dargestellt wird. Bei UTF-16 entsprechen die ersten 256 der maximal 65.536 Zeichen den Zeichen des Zeichensatzes ISO Latin-1 (ISO 8859-1).

Die Unicode-Norm beschreibt einen linguistischen Sortieralgorithmus. Jedem Unicode-Zeichen wird ein Sortierelement (Collation-Element) zugeteilt. Die Reihenfolge nach der die Unicode-Zeichen sortiert werden, wird mit Hilfe dieser Sortierelemente festgelegt. Die Sortierelemente werden mittels einer von XHCS gelieferten Tabelle (Unicode Default Collation Table) festgelegt. Diese Tabelle enthält eine Wertigkeit des Zeichens auf verschiedenen Ebenen (Leveln). SORT unterscheidet drei Ebenen, die in der nachfolgenden Tabelle dargestellt sind. Der Vergleich der einzelnen Zeichen findet jeweils von links nach rechts statt. Der erste Unterschied bestimmt das Vergleichsergebnis.

Im BS2000 können Sie sich das Sortierelement über XHCS besorgen, siehe auch Handbuch „XHCS“ [10].

Vergleichsebene	Beschreibung	Beispiel
Ebene 1	Grundzeichen (base character)	a < b role < roles < rule
Ebene 2	Akzente (diakritische Zeichen)	A < Å role < rôle < roles
Ebene 3	Groß-/Kleinschreibung	a < A role < Role < rôle

Ebene 1 : Jedem Grundzeichen (a,b,c...) ist eine feste Wertigkeit in der Unicode Default Collation Table zugeordnet. Nachfolgende Zeichen oder diakritische Zusätze haben keinen Einfluß auf die Sortierreihenfolge der Zeichen.

Ebene 2 : Zum Grundzeichen gehört ein diakritisches Zeichen. Ein diakritisches Zeichen ist ein Zusatzzeichen (z.B. Akzent, Strich, Punkt, Häkchen, Tilde), das in, über oder unter einen Buchstaben gesetzt wird, um dessen Aussprache oder Betonung näher zu bezeichnen. Grundzeichen mit diakritischem Zeichen haben auf Ebene 1 die gleiche Wertigkeit wie das zugehörige Grundzeichen ohne diakritisches Zeichen. Auf Ebene 2 haben Zeichen mit diakritischem Zeichen eine höhere Wertigkeit als dasselbe Zeichen ohne diakritisches Zeichen. Ist der gesamte Sortierbegriff ansonsten gleich, wird mit diesem diakritischen Zeichen die Sortierreihnefolge (siehe Beispiel Ebene 2: role < rôle) festgelegt.

Ebene 3: Die Sortierreihenfolge wird durch die Unterscheidung zwischen Groß- und Kleinbuchstaben festgelegt. Großbuchstaben haben eine höhere Wertigkeit als Kleinbuchstaben. Ebene 3 wird nur berücksichtigt, wenn Ebene 1 und Ebene 2 für den gesamten Sortierbegriff gleich sind (siehe Beispiel Ebene 3: role < Role).



Folgen zwei Unicode-Sortierfelder direkt aufeinander, so kann es einen Unterschied in der Sortierreihenfolge ergeben, ob man in der SORT-RECORDS- oder MERGE-RECORDS-Anweisung die Sortierfelder als ein oder als zwei Felder angibt. In Ebene 2 und Ebene 3 wird die Sortierreihenfolge nur innerhalb eines Sortierfeldes festgelegt.

Einem Sortierelement kann ein Zeiger auf ein zweites Sortierlement zugeordnet werden. Zum Beispiel, wenn zu einem Grundzeichen ein diakritisches Zeichen gehört und beide Zeichen als ein Zeichen verschlüsselt sind.

Zeigt einen Zeiger auf ein zweites Sortierelement mit fehlerhaftem Inhalt (z.B. Zeiger zeigt aus der Tabelle), so wird der Sortierlauf mit SRT1089 abgebrochen.

2.7.1 Normalisierung

Bei Grundzeichen mit diakritischen Zeichen (Ebene 2) kann die Codierung eines Zeichens in Unicode unterschiedlich sein. D.h. es kann für ein Zeichen mehrere Codierungen geben. So kann z.B. das Zeichen „Ä“ auch als Zeichenfolge von „A“ und „ö“ geschrieben werden. Beim Vergleich werden diese Zeichen als gleichwertig behandelt. Um eine Datei zu vereinheitlichen, gibt es die Normalisierungsfunktionen DECOMPOSE und COMPOSE. Die Normalisierung ordnet gleichen Zeichen mit unterschiedlicher Codierung ein einheitliches Format zu. Grundlage der Normalisierung ist die Normalisierungstabelle von XHCS, die aus der Unicode-Sortiertabelle abgeleitet wird, siehe auch Handbuch „XHCS“ [10].

Die Normalisierungsfunktion erfolgt über den XHCS-Makroaufruf:

```
NLSCNV ACTION = COMPOSE oder
```

```
NLSCNV ACTION = DECOMPOSE.
```

COMPOSE fügt ein Grundzeichen mit den zugehörigen diakritischen Zeichen zu einem einzigen Zeichen zusammen.

DECOMPOSE zerlegt ein zusammengesetztes Zeichen in das Grundzeichen und die zugehörigen diakritischen Zeichen. Die Reihenfolge der verknüpften diakritischen Zeichen ist dabei streng festgelegt.

Liegt keine der beiden Normalisierungsformen vor, so kann es zu Folgefehlern in der Sortierreihenfolge kommen.

2.7.2 Zeichen mit spezieller Verarbeitung

Über den Operanden IGNORE-UNICODE-BLANK in der SET-SORT-OPTIONS-Anweisung ist es möglich bestimmte Zeichen im Text zu berücksichtigen oder zu ignorieren. Werden Zeichen ignoriert, wird der Sortierbegriff zusammengezogen und am Ende gegebenenfalls mit dem Unicode-Füllzeichen aufgefüllt. Dadurch wird die Sortierreihenfolge verändert. Zeichen die ignoriert werden können sind Leerzeichen, Bruchstrich, Bindestrich und variable Sortierelemente.

Die nachfolgende Tabelle zeigt in einem Beispiel wie sich die Sortierreihenfolge ändert:

Zeichen	berücksichtigt	ignoriert
Leerzeichen	zu Hause bleiben zu Hause sein zuhause bleiben	zuhause bleiben zu Hause bleiben zu Hause sein
Bruchstrich z.B. $\frac{1}{4}$	drei Zeichen: 1 (eins), - (Bruchstrich), 4 (vier)	zwei Zeichen: 14

Besonders behandelt werden folgende Zeichen die aus mehreren Grundzeichen (nicht diakritischen Zeichen) zusammengesetzt sind:

œ (hex: 0153)	entspricht:	oe (hex: 006F 0065)
Œ (hex: 0152)	entspricht:	OE (hex: 004F 0045)
№ (hex: 2116)	entspricht:	No (hex: 004E 0366)
ß (hex: 00DF)	entspricht:	ss (hex: 0073 0073)

2.7.3 Zeichen die nicht unterstützt werden

Nicht alle zwei Byte langen Verschlüsselungen gehören zu einem unterstützten Unicode-Zeichen. Alle Zeichen, die nicht unterstützt werden, werden ignoriert. Zu diesen Zeichen gehören auch die zwei Byte langen Zeichen, die außerhalb des unterstützten Bereichs liegen. Unterstützt wird momentan der Bereich zwischen X'0000' und X'2FFF'. Werden Zeichen nicht unterstützt, wird der Sortierbegriff zusammengezogen und am Ende gegebenenfalls mit dem Unicode-Füllzeichen aufgefüllt.

Zu den nicht unterstützten Zeichen gehört auch das Zeichen NIL (X'0000'). Besteht ein Sortierbegriff nur aus NIL-Zeichen, so wird nur mit Unicode-Füllzeichen verglichen. Das Unicode-Füllzeichen wird in der SET-RECORD-ATTRIBUTES-Anweisung angegeben.



Ändert sich die von XHCS gelieferte Sortiertabelle (z.B. durch Aufnahme neuer Zeichen), so kann sich auch die Sortierreihenfolge ändern. Zeichen die bisher ignoriert wurden, werden dann nach der Aufnahme in die Unicode-Tabelle ausgewertet.

3 Dateien des Sortier-/Mischprogramms SORT

Das Sortier-/Mischprogramm SORT arbeitet mit Dateien, die entweder von SORT oder vom Benutzer eingerichtet werden können und mit Dateien, die vom Benutzer selbst eingerichtet werden müssen. DUMMY-Dateien können verarbeitet werden. Im Einzelnen arbeitet SORT mit den folgenden Dateien (siehe [Bild 7](#)):

- Sortiereingabedateien
- Mischeingabedateien
- Ausgabedatei
- Hilfsdateien (nicht bei Mischanwendung)
- Arbeitsdateien (nicht bei Mischanwendung)
- Fixpunktdatei
- Bindemodulbibliothek SORTMODS
- Anweisungsdateien

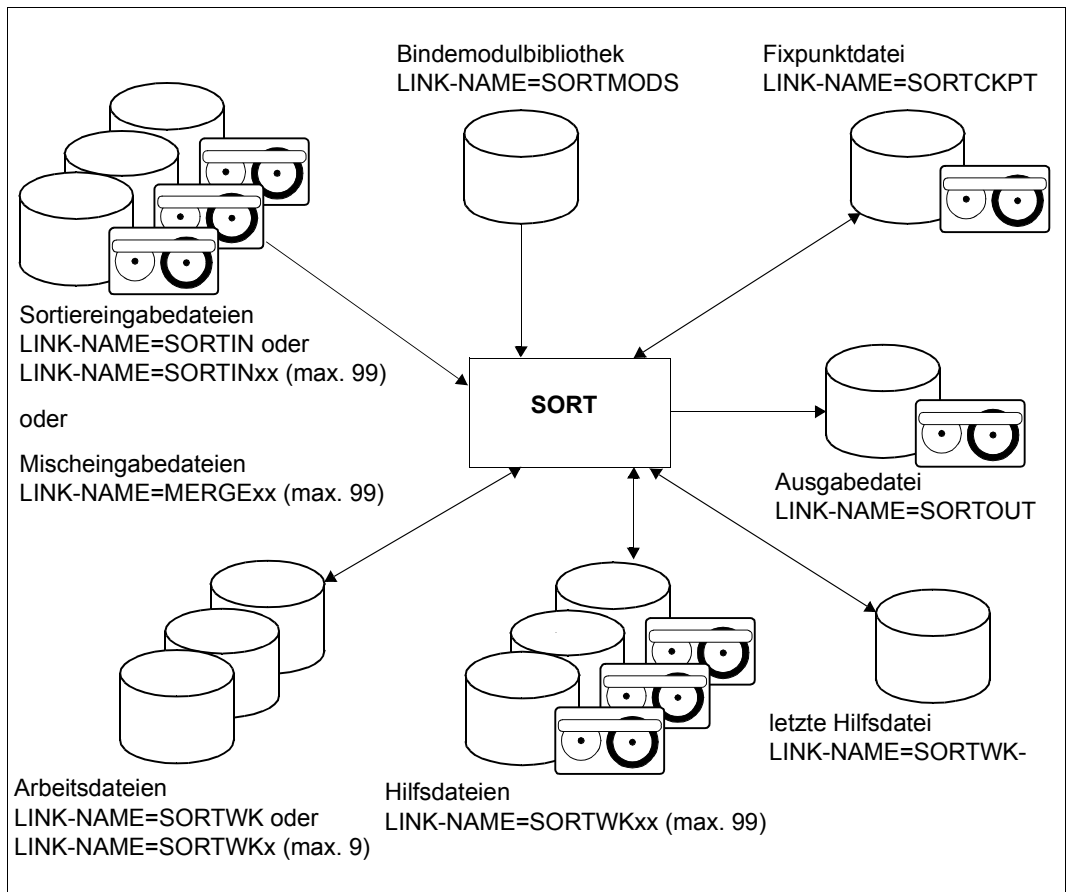


Bild 7: Dateien des SORT

Die Zuordnung der benötigten Dateien erfolgt entweder mit der ASSIGN-FILES-Anweisung oder mit Dateikettungenamen in einem ADD-FILE-LINK-Kommando oder einem FILE-Makroaufruf. Im Fall der ASSIGN-FILES-Anweisung werden von SORT intern ADD-FILE-LINK-Kommandos mit den entsprechenden Dateikettungenamen (SORTIN oder SORTINxx, MERGExx, SORTOUT, SORTWK oder SORTWKx, SORTWKxx, SORTCKPT und SORTMODS) abgesetzt. Existieren für diese Dateikettungenamen bereits TFT-Einträge, so werden sie überschrieben.

Wenn Dateien gleichzeitig über die ASSIGN-FILES-Anweisung und über das ADD-FILE-LINK-Kommando zugewiesen werden, so werden die über das Kommando zugewiesenen Dateien ignoriert. Wenn der Operand LINK-PREFIX-CHANGE in der Anweisung SET-SORT-OPTIONS angegeben ist, wird in den verwendeten Dateikettungenamen das Präfix „SORT“ bzw. „MERGE“ durch das angegebene Präfix ersetzt.

Der Operand IGNORE-INOUT-FILE der SET-SORT-OPTIONS-Anweisung ist für Dateien, die über die ASSIGN-FILES-Anweisung zugewiesen wurden, nicht wirksam.

Hinweise

- Werden Eingabe-, Ausgabe-, Arbeits- und Hilfsdateien vom Benutzer mit CREATE-FILE und ADD-FILE-LINK eingerichtet, so können dafür Angaben zur Benutzung schneller Speichermedien gemacht werden (Operand PERFORMANCE). Werden diese Dateien von SORT eingerichtet, so wird der Wert PERFORMANCE=*STD verwendet, d.h. das Standardspeichermedium wird verwendet.
- Ein- oder Ausgabedateien können im DIV(Data-in-Virtual)-Format vorliegen (PAM-Datei mit BLOCK-CONTROL-INFO=*NO). SORT bearbeitet diese Dateien mit der Zugriffsmethode UPAM.
- Alle SORT-Banddateien müssen auf getrennten Bändern eingerichtet sein. Eine Ausnahme ist nur bei Sortierläufen zulässig, wenn Sortiereingabedateien und die Sortierausgabedatei auf einem gemeinsamen Band liegen.

Behandlung der von SORT eingerichteten temporären Dateien

Von SORT eingerichtete Dateien, z.B. Hilfs- und Arbeitsdateien, werden am Ende einer Sortierung automatisch gelöscht, wenn der Lauf normal oder fehlerhaft ohne RESTART-Möglichkeit endet. Endet ein Lauf fehlerhaft mit der Möglichkeit eines RESTART, löscht SORT diese Dateien nicht. Unterbleibt der RESTART, so muss der Benutzer die temporären Dateien selbst löschen.

Zuordnung und Erstellung der SORT-Dateien

Dateiart	Zuordnung über		Erstellung	
	/ADD-FILE-LINK	//ASSIGN-FILES	auf Grund von Angaben	bei Bedarf von SORT eingerichtet
Sortiereingabedatei Sortiereingabedateien	SORTIN SORTINxx	INPUT-FILES= INPUT-FILES=(...)	-	-
Mischeingabedateien	MERGExx	INPUT-FILES=(...)	-	-
Sortier-/ Mischausgabedatei	SORTOUT	OUTPUT-FILE=	-	-
Sortierarbeitsdatei Sortierarbeitsdateien	SORTWK SORTWKx	WORK-FILES= WORK-FILES=(...)	-	1 Platten- Datei
Sortierhilfsdateien	SORTWKxx	AUXILIARY-FILES=(...)	bis zu 99 Dateien	1 Platten- Datei als Reserve
Fixpunktdatei	SORTCKPT	CHECKPOINT-FILE=	1 Datei	1 Platten- Datei
Bindemodulbibliothek	SORTMODS	MODULE-LIBRARY=	-	-
Anweisungsdateien	-	STATEMENT-FILES=(...)	-	-

Erläuterung:

- x steht für Werte von 1 bis 9
- xx steht für Werte von 01 bis 99
- nicht erlaubt

Hinweise

- Der Namensteil „SORT“ bzw. „MERGE“ des Dateikettungsnamens wird bei Angabe des Operanden LINK-PREFIX-CHANGE=<name 1..4> der SET-SORT-OPTIONS-Anweisung durch das angegebene Präfix <name 1..4> ersetzt. Bei den folgenden Datei-beschreibungen gilt dies für alle angegebenen Dateikettungsamen mit den Namens-teilen „SORT“ und „MERGE“.
- Die Dateifolgen x, xx müssen lückenlos durchnummeriert sein. Bei einer Lücke werden nur die Dateien bis zur Lücke verarbeitet.
- SORTIN und SORTWK dürfen nur in Verbindung mit **einer** Eingabedatei bzw. Arbeits-datei angegeben werden. Die Dateikettungsamen SORTIN und SORTINxx sowie SORTWK und SORTWKx dürfen nicht kombiniert werden.

3.1 Eingabedateien

3.1.1 Eingabedateien für Sortierläufe

SORT kann bis zu 99 Eingabedateien in eine Ausgabedatei sortieren. Die Verbindung zum Sortier-/Mischprogramm stellt der Operand INPUT-FILES der ASSIGN-FILES-Anweisung mit den intern vergebenen Dateikettungsnamen SORTIN bzw. SORTINxx her.

Dateikettungsnamen für Eingabedateien

Die Dateikettungsnamen müssen in ADD-FILE-LINK-Kommandos bzw. FILE-Makros angegeben werden oder werden von SORT bei einer ASSIGN-FILES-Anweisung erzeugt.

- Eine Eingabedatei (Eindateiensortierung)

```
/ADD-FILE-LINK LINK-NAME=SORTIN,FILE-NAME=dateiname,... €
```

oder

```
//ASSIGN-FILES INPUT-FILES=dateiname,... €
```

- Mehrere Eingabedateien (Mehrdateiensortierung)

Soll SORT mehrere Eingabedateien in eine Ausgabedatei sortieren, müssen die einzelnen Eingabedateien entweder über ADD-FILE-LINK-Kommandos oder die ASSIGN-FILES-Anweisung zugeordnet werden.

ADD-FILE-LINK-Kommando

```
ADD-FILE-LINK LINK-NAME=SORTINxx,....
```

Für „xx“ ist aufsteigend und lückenlos 01, 02, ..., 99 anzugeben, je nach Anzahl der Eingabedateien. Bei 4 Eingabedateien sind die Dateikettungsnamen wie folgt anzugeben:

```
/ADD-FILE-LINK LINK-NAME=SORTIN01,FILE-NAME=dateiname1 ... 1. Eingabedatei
/ADD-FILE-LINK LINK-NAME=SORTIN02,FILE-NAME=dateiname2 ... 2. Eingabedatei
/ADD-FILE-LINK LINK-NAME=SORTIN03,FILE-NAME=dateiname3 ... 3. Eingabedatei
/ADD-FILE-LINK LINK-NAME=SORTIN04,FILE-NAME=dateiname4 ... 4. Eingabedatei
```

ASSIGN-FILES-Anweisung

```
//ASSIGN-FILES -
//      INPUT-FILES=(dateiname1, dateiname2, dateiname3, dateiname4)
```

Hinweis

Wurden weder eine ASSIGN-FILES-Anweisung noch ADD-FILE-LINK-Kommandos oder FILE-Makros zur Zuordnung von Eingabedateien angegeben, dann muss die Sortiereingabe über den Benutzerausgang INPUT definiert sein. Ist keine Eingabe definiert, so gibt SORT eine Fehlermeldung aus.

Eingabedatei = Ausgabedatei

Beim Adresslistensortieren darf die Eingabedatei nicht als Ausgabedatei verwendet werden. Dies gilt auch für jene Fälle, in denen sich Dateimerkmale (Satzlänge, Satzformat) zwischen Ein- und Ausgabe unterscheiden.

Dateimerkmale bei mehreren Eingabedateien

Bei mehreren Eingabedateien müssen alle Dateien vom gleichen Satzformat sein (RECORD-FORMAT=*FIXED oder RECORD-FORMAT=*VARIABLE). Haben die Eingabedateien festes Satzformat, so müssen alle Eingabedateien außerdem die gleiche Satzlänge haben. Dateityp (ACCESS-METHOD) und Blockgröße (BUFFER-LENGTH) dürfen beim Sortieren von mehreren Eingabedateien unterschiedlich sein.

Bei Verwendung der Formate EXTENDED-CHARACTER und TRANSLATE-CHARACTER müssen die Eingabedateien denselben CCSN besitzen, da SORT keine Datenkonvertierung zur Anpassung unterschiedlicher Codes in den Eingabedateien durchführt. Im Fehlerfall wird der Sortierlauf mit Ausgabe der Meldung SRT1254 abgebrochen.

Ist eine Eingabedatei eine POSIX-Datei, müssen alle Eingabedateien POSIX-Dateien sein.

Die Dateimerkmale überprüft SORT bei katalogisierten Dateien bereits bei Beginn des Sortierlaufs. Bei Fremddateien (Dateien, für die noch kein Systemkatalogeintrag existiert) überprüft SORT die Dateimerkmale erst unmittelbar bevor die Datei eröffnet wird. Dies ist von Bedeutung, wenn die Eingabe aus mehreren Dateien besteht.

Zugriffsrechte für POSIX-Eingabedateien

POSIX-Eingabedateien werden nur dann sortiert, wenn das Zugriffsrecht Lesen vorhanden ist. Zugriffsrechte für POSIX-Eingabedateien werden durch SORT nicht verändert.

Eröffnung der Eingabedateien

Eingabedateien eröffnet SORT standardmäßig mit OPEN=INPUT. Bei BTAM- und SAM-Magnetbanddateien kann der Benutzer den OPEN-Modus im ADD-FILE-LINK-Kommando auch auf OPEN-MODE=*REVERSE ändern und dadurch Bandeingabedateien rückwärts einlesen lassen.

Gesperrte Dateien versucht SORT in Abständen von einer Sekunde zu eröffnen (maximal 100 Versuche). Kann SORT die Datei dann immer noch nicht eröffnen, so gibt SORT eine Fehlermeldung aus. Dies wirkt sich, wenn nur eine Eingabedatei vorhanden ist, anders aus als bei mehreren Eingabedateien:

- Eine Eingabedatei
Beim Eindateiensort bricht SORT den Sortierlauf ab, wenn die Eingabedatei nicht eröffnet werden kann.
- Mehrere Eingabedateien
Mittels des Operanden INPUT-OPEN-ERROR der SET-SORT-OPTIONS-Anweisung kann man festlegen, wie SORT reagieren soll, wenn eine der Eingabedateien nicht eröffnet werden kann. Es gibt folgende Möglichkeiten:
 - *CONTINUE-NEXT-FILE(TERMINATE=*NORMAL)
SORT meldet den Fehler, übergeht die betroffene Datei und bearbeitet die übrigen Dateien weiter.
 - *FINISH-INPUT
SORT meldet den Fehler, beendet die Eingabe und sortiert die eingelesenen Sätze.
 - *TERMINATE-ABNORMAL
SORT meldet den Fehler und beendet fehlerhaft.
 - *CONTINUE-NEXT-FILE (TERMINATE=*ABNORMAL)
SORT meldet den Fehler, übergeht die betroffene Datei, bearbeitet die übrigen Dateien weiter und beendet wie folgt
 - a) in einem autonomen Sortierlauf mit TERM UNIT=STEP,MODE=ABNORMAL.
 - b) in einem Sortierunterprogrammablauf mit Abspeichern des Rücksprungkennzeichens X'FF' im niedrigstwertigen Byte von Register 15. In den ersten beiden Bytes von Register 15 steht die Meldungsnummer (so hat z.B. bei einem Eröffnungsfehler SRT1035 Register 15 den Inhalt X'103500FF').

Hinweis

Bei sonstigen DVS-Fehlern wird der SORT-Lauf mit Fehler abgebrochen.

Schlussbehandlung der Eingabedateien nach Beendigung des Sortierlaufs

Die Eingabedatei(en) werden von SORT nach dem Einlesen geschlossen (CLOSE) und der zugehörige Dateikettungsname freigegeben (RELEASE). SORT steuert die Freigabe der Eingabedateien (bzw. Geräte) auf Grund der

- Art des Eingabegeräts (Band/Platte)
- Gerätebelegung vor dem SORT-Lauf (PREMOUNT-LIST=0 im Kommando CREATE-FILE)
- Lage von Banddateien (Multifile/Multivolume-Sets).

Aus diesen Angaben wählt SORT die optimale RELEASE-Art für die Eingabedateien aus. Es wird dadurch eine möglichst frühzeitige Gerätefreigabe erreicht. Bei Banddateien wird auch das Entladen des Bandes optimiert.

Mit dem Operanden KEEP-INPUT-TAPES der SET-SORT-OPTIONS-Anweisung kann bestimmt werden, ob Eingabebänder nach dem Einlesen zurückgespult und entladen werden sollen, oder ob nur das Rückspulen erfolgen soll. Bei Multi-Volume-Dateien wird allerdings nur das letzte Band des Multi-Volume-Sets nicht entladen.

Hinweis

Werden bei Multi-Volume-Eingabedateien, für die noch kein Katalogeintrag existiert, nicht alle Bänder im ADD-FILE-LINK-Kommando angegeben, fordert SORT die fehlenden Bänder nach. Wird die Bandanforderung vom Operateur abgelehnt, wird Dateieinde angenommen und der Sortierlauf fortgesetzt. Die bis dahin eingelesenen Sätze werden sortiert. Wird bei einer katalogisierten Multi-Volume-Eingabedatei eine Bandanforderung vom Operateur abgelehnt, wird der Sortierlauf abgebrochen.

- Bei Bandeingabedateien von Multifile/Multivolume-Sets sollte die Lage der Dateien auf den Bändern (auf dem Band) mit der Reihenfolge der Verarbeitung übereinstimmen, um unnötige Positions- und Entladevorgänge zu vermeiden (1. Datei = SORTIN01, 2. Datei = SORTIN02).
- Bei gemischter Eingabe von Band- und Plattendateien sollte man die Eingabedateien in Gerätegruppen ordnen, d.h. zuerst alle Banddateien, danach alle Plattendateien (oder umgekehrt) verarbeiten. Mit einem LOCK-FILE-LINK-Kommando kann der Benutzer verhindern, dass die ADD-FILE-LINK-Zuordnung freigegeben wird. LOCK-FILE-LINK wirkt solange, bis ein UNLOCK-FILE-LINK-Kommando abgesetzt wird.

Bei Banddateien werden am Ende des Sortierlaufs alle Bandgeräte freigegeben.

3.1.2 Eingabedateien für Mischläufe

SORT kann bis zu 99 Eingabedateien in eine Ausgabedatei mischen. Alle Eingabedateien müssen beim Mischen bereits nach denselben Sortierkriterien sortiert sein. Die Dateikettungsnamen MERGE01, MERGE02, ..., MERGE xx stellen die Verbindung zum Sortier-/Mischprogramm her. Für „ xx “ muss aufsteigend und lückenlos, je nach Anzahl der Eingabedateien, 01, 02, ..., 99 angegeben werden. Diese Dateikettungsnamen werden bei Verwendung des ASSIGN-FILES-Kommandos intern vergeben. Sie können aber auch in einem ADD-FILE-LINK-Kommando bzw. FILE-Makro stehen.

```
/START-SORT
.
.
//ASSIGN-FILES INPUT-FILES=(Liste von Eingabedateien)
//MERGE-RECORDS FIELDS=...
.
//END €
```

Bei Verwendung der Dateikettungsnamen müssen die dazugehörigen ADD-FILE-LINK-Kommandos bei 4 Eingabedateien wie folgt lauten:

```
/ADD-FILE-LINK LINK-NAME=MERGE01,FILE-NAME=dateiname1 ...      1. Eingabedatei
/ADD-FILE-LINK LINK-NAME=MERGE02,FILE-NAME=dateiname2 ...      2. Eingabedatei
/ADD-FILE-LINK LINK-NAME=MERGE03,FILE-NAME=dateiname3 ...      3. Eingabedatei
/ADD-FILE-LINK LINK-NAME=MERGE04,FILE-NAME=dateiname4 ...      4. Eingabedatei
```

Hinweis

Wenn die Eingabe für den Mischlauf über den Benutzerausgang INPUT definiert ist, muss sie zur Gänze über diesen Benutzerausgang definiert sein.

Dateimerkmale der Mischeingabedateien

Alle Mischeingabedateien müssen gleiches Satzformat (RECORD-FORMAT=*VARIABLE oder RECORD-FORMAT=*FIXED) haben. Haben die Mischeingabedateien festes Satzformat, muss auch die Satzlänge gleich sein. Bei Sätzen mit variabler Satzlänge muss die Satzstruktur für jene Satzteile gleich sein, auf die sich die Sortierfeldbeschreibung bezieht. Bei Verwendung der Formate EXTENDED-CHARACTER und TRANSLATE-CHARACTER müssen die Eingabedateien denselben CCSN besitzen, da SORT keine Datenkonvertierung zur Anpassung unterschiedlicher Codes in den Eingabedateien durchführt. Im Fehlerfall wird der Sortierlauf mit Ausgabe der Meldung SRT1254 abgebrochen.

Ist eine der Mischeingabedateien eine POSIX-Datei, müssen alle Mischeingabedateien POSIX-Dateien sein und das Zugriffsrecht Lesen besitzen.

Eröffnung der Mischeingabedateien

Standardmäßig werden die Eingabedateien für einen Mischlauf mit OPEN=INPUT eröffnet. BTAM-Dateien und SAM-Magnetbanddateien können auch rückwärts eingelesen werden. Sie werden dann mit OPEN-MODE=*REVERSE eröffnet.

Tritt bei der Eröffnung einer Eingabedatei ein Fehler auf, wird die SORT-Meldung SRT1035 ausgegeben. Bei sonstigen DVS-Fehlern gibt SORT die Meldung SRT1036 aus. Mischläufe werden danach abgebrochen.

Schlussbehandlung der Mischeingabedateien

Bei Beendigung des Mischlaufs schließt SORT die Eingabedateien (CLOSE) und gibt den dazugehörigen Dateikettungsnamen frei (RELEASE). Mit einem LOCK-FILE-LINK-Kommando kann der Benutzer verhindern, dass die ADD-FILE-LINK-Zuordnung freigegeben wird. LOCK-FILE-LINK wirkt solange bis ein UNLOCK-FILE-LINK-Kommando abgesetzt wird.

Bei Banddateien werden am Ende des Mischlaufs alle Bandgeräte freigegeben.

3.1.3 Pamkey-Eliminierung bei Eingabedateien

Die Pamkey-Eliminierung hat auf Eingabedateien nur dann einen Einfluss, wenn es sich um PAM-Dateien handelt.

Für Eingabedateien mit BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK beginnt der Eingabesatz nach dem Control-Field (Blockanfang + 12).

Bei Banddateien mit BLOCK-CONTROL-INFO=*NO wird jeder Block als Eingabesatz interpretiert, ggf. nachdem er mit X'00' auf die maximale Blocklänge aufgefüllt wurde.

Die Überprüfung, ob ein gelesener Block zur Eingabedatei gehört und damit weiterverarbeitet werden soll (Lückentest), erfolgt bei Dateien mit BLOCK-CONTROL-INFO=*PAMKEY bzw. *WITHIN-DATA-BLOCK durch Vergleich der Coded-File-ID im PAM-Schlüssel bzw. Control-Field mit der im Katalog eingetragenen. Bei PAM-Dateien mit BLOCK-CONTROL-INFO=*NO werden alle Blöcke übernommen.

3.2 Ausgabedatei für Sortier-/Mischläufe

Für die Ausgabe kann unabhängig von der Anzahl der Eingabedateien immer nur eine Ausgabedatei zugewiesen werden. Es ist notwendig, dass der Ausgabedatei genügend Speicherplatz über den SPACE-Operanden im CREATE-FILE-Kommando zugewiesen wird. Unnötige Sekundärzuweisungen vermeidet der Benutzer, wenn er für die Ausgabedatei die Anzahl der einzulesenden PAM-Seiten der Eingabedateien als Primärzuweisung angibt. Satzlängenänderungen und Auswahlkriterien können bei der Größenangabe für die Primärzuweisung bereits berücksichtigt werden.

Der Dateikettungsname SORTOUT stellt die Verbindung zum Sortier-/Mischprogramm her. Er gilt für Sortier-/Mischläufe gleichermaßen und wird entweder intern beim ASSIGN-FILES-Kommando vergeben oder im ADD-FILE-LINK-Kommando bzw. FILE-Makro angegeben.

Hinweis

- Wurde weder ein ADD-FILE-LINK-Kommando bzw. FILE-Makro mit LINK-NAME= SORTOUT noch eine ASSIGN-FILES-Anweisung angegeben, muss die Sortierausgabe über den Benutzerausgang OUTPUT definiert sein. Ist dies ebenfalls nicht der Fall, gibt SORT eine Fehlermeldung aus.
- Kann SORT die angegebene Ausgabedatei nicht eröffnen, gibt SORT die Meldung SRT1035 aus und beendet den Lauf fehlerhaft.

Dateiattribute der Ausgabedatei

ACCESS-METHOD (FILE-STRUCTURE)

Ausgabedateien können PAM-, BTAM-, SAM- oder ISAM-Dateien sein. Wurde für die Ausgabedatei keine ACCESS-METHOD angegeben, übernimmt SORT die ACCESS-METHOD der Eingabedatei bzw. der ersten Eingabedatei, wenn mehrere Eingabedateien angegeben werden. Ist dies nicht möglich (z.B. Eingabe über Benutzereingang INPUT), setzt SORT standardmäßig ACCESS-METHOD=*SAM.

RECORD-FORMAT

Fehlt diese Angabe für die Ausgabedatei, übernimmt SORT standardmäßig den Wert aus dem OUTPUT-Operanden der SET-RECORD-ATTRIBUTES-Anweisung. Ist dies nicht möglich, übernimmt SORT den Wert der Eingabedatei.

Ausnahme

Bei Auswahl- und Adresslistensortieren (siehe [Seite 27](#) bzw. [Seite 29](#)) legt SORT die Ausgabedatei standardmäßig mit RECORD-FORMAT=*FIXED an.

RECORD-SIZE

Wurde keine Längenangabe für die Ausgabedatei angegeben, entnimmt SORT diese der ersten (oder einzigen) Eingabedatei. Ist dies nicht möglich, nimmt SORT hierfür die Längenangabe, die im OUTPUT-Operanden der Anweisung SET-RECORD-ATTRIBUTES angegeben bzw. von SORT selbst berechnet wurde.

Falls für die erste (oder einzige) Eingabedatei RECORD-SIZE=0 festgelegt ist, wird dieser Wert nicht für die Ausgabedatei übernommen. Ist für die Ausgabedatei in diesem Fall dennoch RECORD-SIZE=0 erwünscht, so muss dies mit einem entsprechenden ADD-FILE-LINK-Kommando erzwungen werden.

BUFFER-LENGTH

Ist die Blockgröße der Ausgabedatei nicht definiert, wird sie aus der Eingabedatei bzw. bei mehreren Eingabedateien aus der ersten Eingabedatei übernommen. Ist diese Blockgröße kleiner als die Ausgabesatzlänge oder ist keine Eingabedatei vorhanden (die zu sortierenden Sätze werden durch den Benutzerausgang INPUT übergeben), wird die Blockgröße der Ausgabedatei gleich der Ausgabesatzlänge gesetzt, aufgerundet auf ein Vielfaches von STD(1)-Blöcken (2048 Byte, K-/NK2-Platten) bzw. STD(2)-Blöcken (4096 Byte, NK4-Platten). Die Ausgabesatzlänge entnimmt SORT entweder der SET-RECORD-ATTRIBUTES-Anweisung oder SORT berechnet sie.

Ist die Blockgröße der Ausgabedatei definiert, wird geprüft, ob sie größer oder gleich der Ausgabesatzlänge ist. Falls die Blockgröße der Ausgabedatei kleiner ist als die Ausgabesatzlänge, wird der Lauf fehlerhaft beendet. Für NK4-Platten muss die Blockgröße ein Vielfaches von STD(2) sein, andernfalls wird der Sortierlauf mit der Meldung SRT1252 abgebrochen.

CODED-CHARACTER-SET

Der CCSN der Ausgabedatei wird von SORT vergeben, eine Benutzerangabe für die Ausgabedatei wird ignoriert. Der CCSN der Ausgabedatei wird wie der CCSN der Datensätze bestimmt. Wird „EDF03IRV“ ermittelt, wird für die Ausgabedatei kein CCS-Attribut definiert (siehe [Abschnitt „Sortieren mit erweiterten Codes“ auf Seite 82](#)).

BLOCK-CONTROL-INFO (BLKCTRL)

Ist das Datenformat der Ausgabedatei nicht definiert, wird es aus der Eingabedatei bzw. bei mehreren Eingabedateien aus der ersten Eingabedatei übernommen. Nötigenfalls korrigiert SORT das Datenformat in Abhängigkeit von ACCESS-METHOD, Blockgröße und Class2-Option BLKCTRL.

Ist das Datenformat der Ausgabedatei definiert, versucht SORT die Ausgabedatei anzulegen. Im Fehlerfall wird die Meldung SRT1253 ausgegeben und der Sortierlauf beendet.

Besonderheiten bei ISAM-Ausgabedateien

- ISAM-Schlüssel
SORT benutzt bei ISAM-Ausgabedateien mit nicht definiertem ISAM-Schlüssel grundsätzlich das höchstwertige Sortierfeld (Feld mit der Ordnungsnummer 1) als ISAM-Schlüssel. Das höchstwertige Sortierfeld und weitere für den ISAM-Schlüssel benutzte Sortierfelder dürfen daher nur in aufsteigender Sortierfolge sortiert werden. Wird ein Wert für KEY-POSITION explizit angegeben, muss dieser mit dem Anfang des höchstwertigen Sortierfeldes übereinstimmen.
- Länge des ISAM-Schlüssels
SORT nimmt bei ISAM-Ausgabedateien als Schlüssellänge die Länge des höchstwertigen Sortierfeldes. Ist KEY-LENGTH bereits für die Ausgabedatei definiert (ADD-FILE-LINK-Kommando/FILE-Makroaufruf bzw. Katalogeintrag), so darf sich die Schlüssellänge auch auf die Folgesortierfelder erstrecken. Diese Folgesortierfelder müssen hintereinander liegen und dem höchstwertigen Sortierfeld unmittelbar folgen. Eine Überschneidung der Sortierfelder ist nicht zulässig. Der zulässige Wert von KEY-LENGTH ergibt sich aus der Summe der Längen des höchstwertigen Sortierfeldes und der unmittelbar folgenden Sortierfelder. Der ISAM-Schlüssel muss bei Sätzen mit variabler Satzlänge mit seiner gesamten Länge im festen Teil des Satzes liegen.
- Datenformate bei ISAM-Ausgabedateien
Das höchstwertige Sortierfeld der Eingabedatei darf nur das Datenformat BINARY und CHARACTER haben. Alle übrigen Datenformate sind *nicht* erlaubt, da SORT dieses Feld zum ISAM-Schlüssel der Ausgabedatei macht. Beim Datenformat BINARY müssen die Felder auf Byte-Grenze beginnen.
- ISAM-Dateien mit Satzzeichenfeldern (FLAGGED ISAM FILES)
Standardmäßig nimmt SORT für LOGICAL-FLAG-LENGTH und VALUE-FLAG-LENGTH „0“ an. Werden im CREATE-FILE-Kommando/FILE-Makro Werte für LOGICAL-FLAG-LENGTH und VALUE-FLAG-LENGTH angegeben, darf die Summe aus KEY-LENGTH + VALUE-FLAG-LENGTH + LOGICAL-FLAG-LENGTH 255 nicht überschreiten. SORT überprüft die Bedingung und bricht bei Fehler den Lauf ab.
- Gleicher Sortierbegriff
Bei Sätzen mit gleichen Sortierbegriffen ist die Reihenfolge der Ausgabe nicht definiert. Dies gilt auch für den Fall, dass KEEP-EQUAL-SEQUENCES=*YES angegeben wurde.

- **Doppelter ISAM-Schlüssel**
SORT nimmt standardmäßig DUPLICATE-KEY=*YES an, d.h. in der Ausgabedatei dürfen Sätze mit gleichem ISAM-Schlüssel vorhanden sein. In diesem Fall gibt SORT die Warnmeldung SRT1070 aus und setzt den Sortier-/Mischlauf normal fort. Wenn DUPLICATE-KEY=*NO im ADD-FILE-LINK-Kommando gewählt wird, und es treten während der Sortierung doppelte ISAM-Schlüssel auf, bricht SORT den Sortier-/Mischlauf mit der Fehlermeldung SRT1036 ab.

Bei Verwendung doppelter ISAM-Schlüssel können keine Sekundärschlüssel gebildet werden (siehe Kommando CREATE-ALTERNATE-INDEX im Handbuch „Kommandos“ [1]).

Um das Auftreten von Sätzen mit gleichem ISAM-Schlüssel in der Ausgabedatei zu vermeiden, muss das Sortierfeld verlängert oder bei definiertem ISAM-Schlüssel die Schlüssellänge vergrößert werden

- Der Prozentsatz der freien Pufferlänge (PADDING-FACTOR) ist vom DVS standardmäßig auf 15% gesetzt. Die (max.) Satzlänge darf daher die Puffergröße BUFFER-LENGTH - PADDING-FACTOR nicht überschreiten, da sonst das DVS einen Fehler meldet.

Eröffnung der Ausgabedatei

Die Ausgabedateien können – der ACCESS-METHOD entsprechend – auf folgende Weise eröffnet werden:

- PAM-Dateien: mit OUTIN
- BTAM-Dateien: mit OUTPUT oder OUTIN
- SAM- oder ISAM-Dateien: mit OUTPUT oder EXTEND

Wird der Eröffnungsmodus nicht angegeben, eröffnet SORT standardmäßig PAM-Dateien mit OPEN=OUTIN. BTAM-Dateien, SAM-Dateien und ISAM-Dateien werden standardmäßig mit OPEN=OUTPUT eröffnet.

Hinweis

Kann SORT die angegebene Ausgabedatei nicht eröffnen, gibt SORT die Meldung SRT1035 aus und beendet den Lauf fehlerhaft.

Schlussbehandlung der Ausgabedatei

Bei Beendigung schließt SORT die Ausgabedatei (CLOSE). Der Dateikettungsname wird nicht freigegeben.

3.2.1 Pamkey-Eliminierung bei der Ausgabedatei

Das Dateimerkmal BLOCK-CONTROL-INFO für Ausgabedateien bestimmt SORT auf folgende Weise:

1. Falls der Benutzer ein ADD-FILE-LINK-Kommando mit dem Operanden BLOCK-CONTROL-INFO angegeben hat, gilt dieser Wert.
2. Wenn für die Datei bereits ein Wert für BLOCK-CONTROL-INFO im Katalog eingetragen ist, gilt dieser Wert.
3. Treffen beide Fälle nicht zu, so versucht SORT den Wert für BLOCK-CONTROL-INFO von der Eingabedatei zu übernehmen. Dies ist jedoch nur möglich, wenn das Ausgabe-medium das zulässt; z.B. kann auf einer NONKEY-Platte keine Datei mit PAMKEY eingerichtet werden.
4. Trifft keiner der voranstehenden Fälle zu, so tritt folgende Default-Regelung in Kraft:

Für Banddateien ist BLOCK-CONTROL-INFO=*DATA (aus Performancegründen).

Bei Plattendateien hängt BLOCK-CONTROL-INFO gemäß folgender Tabelle von ACCESS-METHOD und dem Präformat (SM-Pubsets) bzw. der Class2-Option BLKCTRL ab (siehe auch Handbuch „Einführung in das DVS“ [2]):

ACCESS-METHOD	Class2-Option BLKCTRL / Präformat		
	PAMKEY	NONKEY / NK2	NONKEY / NK4
*SAM	*PAMKEY	*DATA	*DATA
*ISAM	*PAMKEY	*DATA2K	*DATA4K
*PAM	*PAMKEY	*NO	*NO

SAM-Ausgabedatei

Bei SAM-Ausgabedateien besteht abhängig vom BLOCK-CONTROL-INFO-Wert und Satzformat folgender Zusammenhang zwischen Blocklänge und maximaler Satzlänge:

BLOCK-CONTROL-INFO	Satzformat	max. Satzlänge
*PAMKEY	*FIXED	BUFFER-LENGTH
	*VARIABLE	BUFFER-LENGTH-4
*WITHIN-DATA-BLOCK	*FIXED	BUFFER-LENGTH-16
	*VARIABLE	BUFFER-LENGTH-16
*NO *)	*FIXED	BUFFER-LENGTH
	*VARIABLE	BUFFER-LENGTH-4

*) Nur für Banddateien

PAM-Ausgabedatei

Bei BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK wird der Ausgabesatz hinter dem Control-Field (Blockanfang + 12) eingetragen.

Auf eine Banddatei mit BLOCK-CONTROL-INFO=*NO wird jeder einzelne Ausgabesatz, nach eventueller Auffüllung mit X'00', in Blocklänge ausgegeben.

3.2.2 POSIX-Ausgabedatei

Zugriffsrechte einer bereits bestehenden POSIX-Ausgabedatei werden von SORT nicht verändert. Eine POSIX-Ausgabedatei, die von SORT angelegt wird, erhält als Zugriffsrechte Lese- und Schreibrecht für den Benutzer.

3.3 Arbeitsdateien

Bei Sortierläufen, bei denen die zu sortierende Datenmenge größer ist als der verfügbare CORE-Speicher, benötigt SORT zum Zwischenspeichern und internen Mischen eine Arbeitsdatei auf Magnetplatte. Wird mit dem Multitaskverfahren sortiert, muss der Benutzer mindestens 2, aber höchstens 9 solcher Arbeitsdateien auf Platte einrichten. Arbeitsdateien müssen PAM-Dateien sein.

Bei reinen Mischläufen sind keine Arbeitsdateien erforderlich.

Einrichten von Arbeitsdateien

Der Benutzer kann Arbeitsdateien über die ASSIGN-FILES-Anweisung bzw. CREATE-FILE-/ADD-FILE-LINK-Kommando oder FILE-Makroaufruf einrichten. Arbeitsdateien können aber auch von SORT selbst eingerichtet werden.

Die für die Arbeitsdatei berechnete Blockgröße wird dabei stets auf ein Vielfaches von STD(2) aufgerundet (eine vom Benutzer vorgegebene Blockgröße wird ignoriert).

Für Arbeitsdateien ist das coded character set ohne Bedeutung. Eine Benutzervorgabe wird ignoriert.

Richtet der Benutzer Arbeitsdateien ein, muss er im ADD-FILE-LINK-Kommando bzw. FILE-Makroaufruf bei einer Arbeitsdatei den Dateikettungsnamen SORTWK und bei mehreren Arbeitsdateien den Dateikettungsnamen SORTWKx angeben. 'x' steht für einen Wert von 1-9, lückenlos aufsteigend. Arbeitsdateien müssen während des Sortier-/ Mischlaufes existent sein, sie dürfen erst nach dem Sortier-/Mischlauf vom Benutzer gelöscht werden. Erfolgt die Dateizuweisung über die ASSIGN-FILES-Anweisung, werden die Dateikettungsnamen von SORT vergeben.

SORT berechnet die Primär- und Sekundärzuweisung für Arbeitsdateien auf Grund einer der folgenden Angaben in der angegebenen Rangfolge:

- Größe der Eingabedateien (bei Plattendateien).
- DISK-SPACE-Angabe (ASSIGN-RESOURCES-Anweisung).
Als Mindestgröße wird die achtfache interne Blocklänge angenommen.
- RECORDS-PER-CYCLE-Angabe (SORT-RECORDS-Anweisung).
- ESTIMATED-RECORDS-Angaben (SORT-RECORDS-Anweisung)
dividiert durch die Anzahl der Hilfsdateien.
- NUMBER-OF-RECORDS-Angabe des INPUT-RANGE-Operanden
(SORT-RECORDS-Anweisung) dividiert durch die Anzahl der Hilfsdateien.
- MEMORY-SIZE-Angabe * 16.

Bei vom Benutzer eingerichteten Arbeitsdateien werden deren Primär- und Sekundärzuweisungen angehoben, wenn diese unter den von SORT errechneten Werten liegen und können über den Benutzerausgang PLANNING ausgegeben werden. SORT nimmt keine Korrektur der vom Benutzer angegebenen Werte vor, wenn für die Sekundärzuweisung der Wert Null angegeben wurde.

Eine Arbeitsdatei, die das Sortierprogramm einrichtet, wird katalogisiert unter dem Dateikettungsnamen SORTWK und dem Dateinamen

```
SORTWORK.tsn.jjmmtt.hhmmss
```

Dabei bedeuten:

tsn	Prozessfolgennummer (TSN) des Sortierlaufs
jj	Jahresangabe
mm	Monatsangabe
tt	Tagesangabe
hhmmss	sechsstellige Uhrzeitangabe

Bei Angabe des Operanden LINK-PREFIX-CHANGE=<name 1..4> in der SET-SORT-OPTIONS-Anweisung wird der Namensteil „SORTWORK“ durch „<name 1..4>WORK“ ersetzt.

Zuweisen von Speicherplatz für Arbeitsdateien

Die Größe von Arbeitsdateien (in PAM-Seiten), die SORT benötigt, lässt sich wie folgt berechnen:

$$\text{Dateigröße} = \frac{1,1 * \text{PAM-Seiten der Sortierdatenmenge}}{(\text{Anzahl der Hilfsdateien} + 1)}$$

Hinweise

- Wird eine Arbeitsdatei mit Sekundärzuweisung = 0 eingerichtet, sollte aus Sicherheitsgründen in obiger Formel statt des Faktors 1,1 der Faktor 1,2 gewählt werden.

Bei sehr großen Dateien (z.B. größer 32 GByte) sollte ebenfalls der Faktor 1,2 gewählt werden, da die Sekundärzuweisung (max. 32767 PAM-Seiten) im Vergleich zur Dateigröße vernachlässigbar ist.

- Arbeitet der Benutzer mit dem Benutzerausgang PLANNING und der Maßnahme DIALOG, gibt SORT die von ihm errechnete Größe über die SORT-Meldung SRT1031 aus.

Die Sortierzeit und der Durchsatz können verbessert werden, wenn der Benutzer

- genügend Speicherplatz zuweist oder genaue Werte für ESTIMATED-RECORDS- bzw. RECORDS-PER-CYCLE-Operanden angibt, damit häufige Sekundärzuweisungen vermieden werden können
- die Arbeitsdateien auf getrennten privaten Datenträger einrichtet
- die Arbeitsdateien nicht auf den Datenträgern der Ein-/Ausgabedateien legt.

Endebehandlung von Arbeitsdateien

- Von SORT eingerichtete Arbeitsdateien werden, wenn der Sortierlauf normal endet, geschlossen, freigegeben und gelöscht. Endet ein Sortierlauf fehlerhaft und wurde mindestens ein Fixpunkt geschrieben, gibt SORT diese Dateien weder frei noch löscht er sie, da sonst kein RESTART-PROGRAM mehr möglich wäre. Unterbleibt ein RESTART-PROGRAM, so muss der Benutzer die Datei löschen. Wird ein RESTART-PROGRAM durchgeführt und endet der Lauf normal, meldet SORT die Datei ab, gibt sie frei und löscht sie.
- Arbeitsdateien, die der Benutzer einrichtet, werden am Ende des Sortierlaufs geschlossen. Der Dateikettungsname wird jedoch nicht freigegeben. Mit der Angabe DELETE-WORK-FILES=*YES der Anweisung SET-SORT-OPTIONS kann der Benutzer festlegen, dass SORT auch die Arbeitsdateien löscht, die vom Benutzer oder vom aufrufenden Programm angelegt worden sind. Standardmäßig löscht SORT diese Dateien nicht.

Hinweise

Arbeitsdateien mit einem Dateikettungsname SORTWKx ('x' steht für einen Wert von 1-9) können ausschließlich nur beim Multitasksortieren (siehe [Seite 277](#)) verwendet werden. Verschiedene SORTWKx können nicht dazu verwendet werden, parallele Sortierläufe bei Nutzung der SORT-Zugriffsmethode SORTZM (siehe [Seite 224](#)) mit verschiedenen Arbeitsdateien zu versorgen.

3.3.1 Pamkey-Eliminierung bei Arbeitsdateien

Arbeitsdateien legt SORT als PAM-Dateien mit BLOCK-CONTROL-INFO=*NO an, auch wenn der Anwender *WITHIN-DATA-BLOCK angibt.

3.4 Hilfsdateien

Hilfsdateien benötigt SORT immer dann, wenn große Datenmengen zyklisch sortiert werden sollen, d.h. die Datenmengen in zu sortierende Teilmengen aufgeteilt werden. Dies ist beim Zyklen- und Multitasksortieren der Fall. Für jede Teilmenge mit Ausnahme der letzten, die in der Arbeitsdatei verbleibt, benötigt SORT eine Hilfsdatei. Daraus ergibt sich:

$$\text{Anzahl der Hilfsdateien} = \text{Zyklenanzahl} - 1$$

Hilfsdateien dürfen sowohl Platten- als auch Banddateien sein (Ausnahme Multitasksortieren: nur Plattendateien). SORT beschreibt und liest diese Dateien sequenziell (SAM). Stehen diese Dateien auf getrennten Datenträgern, kann daher der Durchsatz eventuell beschleunigt werden. Maximal sind 99 solcher Hilfsdateien zulässig.

Bei reinen Mischläufen sind keine Hilfsdateien erforderlich.

Einrichten von Hilfsdateien

Der Benutzer kann Hilfsdateien auf Platte über die ASSIGN-FILES-Anweisung bzw. CREATE-FILE-/ADD-FILE-LINK-Kommando oder FILE-Makroaufruf einrichten. Hilfsdateien auf Bändern müssen über das ADD-FILE-LINK-Kommando zugewiesen werden. Für Hilfsdateien ist das coded character set ohne Bedeutung. Eine Benutzervorgabe wird ignoriert.

Dem Sortier-/Mischprogramm werden diese Dateien über den Dateikettungsnamen SORT-WKxx zugeordnet. „xx“ muss eine lückenlos aufsteigende Reihenfolge von 01 bis max. 99 bilden. Der Benutzer sollte Hilfsdateien immer dann selbst einrichten, wenn große Datenmengen sortiert und Fixpunkte geschrieben werden sollen, bzw. mit dem Multitasksortieren sortiert wird.

Hat der Anwender Angaben gemacht, aus denen SORT die Anzahl der Zyklen ermitteln kann (z.B. ESTIMATED-RECORDS und RECORDS-PER-CYCLE in der SORT-RECORDS-Anweisung), so legt SORT bei Bedarf die erforderliche Anzahl von Hilfsdateien selbst an.

Fehlen diese Angaben, so legt SORT bei Bedarf 8 Hilfsdateien an. Hat der Anwender schon einen Teil dieser Hilfsdateien angelegt, dann werden von SORT nur noch die restlichen Hilfsdateien bereitgestellt. Reichen die Hilfsdateien nicht aus oder kommt es bei einer Arbeitsdatei zu einem Überlauf, richtet SORT einmalig eine zusätzliche letzte Plattenhilfsdatei ein.

Das Einrichten der letzten Hilfsdatei erfolgt unabhängig davon, ob die anderen Hilfsdateien von SORT oder vom Anwender eingerichtet worden sind. Ist allerdings die Anzahl 99 bereits erreicht, wird die letzte Hilfsdatei nicht angelegt. Auch die letzte Hilfsdatei kann wieder genau einen Zyklus aufnehmen. Alle übrigen Sätze verbleiben in der Arbeitsdatei und müssen dort bearbeitet werden.

Die letzte Hilfsdatei belegt standardmäßig PUBLIC-Plattenplatz.

Der Anwender kann den (u.U. ungewollten) Verbrauch an PUBLIC-Plattenplatz auf folgende Weise verhindern:

- Die letzte Hilfsdatei kann vom Anwender selbst angelegt werden, indem der Dateikettungsname SORTWKEX vergeben wird.
- Diese Datei kann sich auch auf einer Privatplatte befinden.
- Der Platz für diese Datei kann durch eine kleine Primärzuweisung und die Angabe 0 als Sekundärzuweisung begrenzt werden. In diesem Fall führt eine zu große Anzahl von Sätzen zu einem SORT-Abbruch. Dadurch wird verhindert, dass der ganze PUBLIC-Plattenplatz von der Arbeits- und Hilfsdatei verbraucht wird.

Wenn der Speicherplatz der letzten Hilfsdatei erschöpft ist, wird der Sortierlauf mit der Meldung SRT1060 abgebrochen.

Hilfsdateien werden unter dem Dateikettungsnamen SORTWKxx katalogisiert. SORT generiert den Dateinamen wie folgt:

```
SORTWKxx.tsn.jjmmtt.hhmmss
```

Dabei bedeuten:

xx	laufende Nummer des Dateikettungsnamens
tsn	Prozessfolgenummer (TSN) des Sortierlaufs
jj	Jahresangabe
mm	Monatsangabe
tt	Tagesangabe
hhmmss	sechstellige Uhrzeitangabe

Bei Angabe des Operanden LINK-PREFIX-CHANGE=<name 1..4> in der SET-SORT-OPTIONS-Anweisung wird der Namensteil „SORTWKxx“ durch „<name 1..4>WKxx“ ersetzt.

Bandhilfsdateien, von SORT eingerichtet

Banddateien können nur dann über ASSIGN-FILES-Anweisungen zugewiesen werden, wenn bereits ein Katalogeintrag existiert. Hat der Benutzer bereits über ADD-FILE-LINK-Kommando/FILE-Makroaufruf Bandhilfsdateien eingerichtet, erweitert SORT diese bis auf die Anzahl, die im TAPE-UNITS-Operanden der ASSIGN-RESOURCES-Anweisung angegeben wurde.

Hinweis

Bei Bandhilfsdateien sollte die RECORDS-PER-CYCLE-Satzmenge die Kapazität des kürzesten Bandes nicht übersteigen (ggf. auch der kürzesten Bandreihe).

Plattenhilfsdateien, von SORT eingerichtet

Richtet SORT Plattenhilfsdateien ein, wird als Wert für die Primärzuweisung der von der Arbeitsdatei belegte Speicher benutzt.

Bei vom Benutzer eingerichteten Hilfsdateien werden deren Primär- und Sekundärzuweisungen angehoben, wenn diese unter den von SORT errechneten Werten liegen. SORT nimmt keine Korrektur vor, wenn der Wert für die Sekundärzuweisung Null ist.

Endebehandlung von Hilfsdateien

- Von SORT eingerichtete Hilfsdateien werden, wenn der Sortierlauf normal endet, abgemeldet, freigegeben und gelöscht. Endet der Sortierlauf fehlerhaft und wurden bereits Fixpunkte geschrieben, gibt SORT diese Dateien weder frei noch löscht er sie, da ansonsten kein RESTART-PROGRAM mehr möglich ist. Führt der Benutzer kein RESTART-PROGRAM durch, muss er die Dateien selbst löschen. Wird ein RESTART-PROGRAM durchgeführt und endet der Lauf normal, schließt SORT die Dateien, gibt sie frei und löscht sie.
- Hilfsdateien, die der Benutzer eingerichtet hat, werden am Ende des Sortierlaufs geschlossen, aber der Dateikettungsname wird nicht freigegeben. Bei Bandhilfsdateien werden die Bänder zurückgespult, aber nicht entladen.

Mit der Angabe DELETE-WORK-FILES=*YES der Anweisung SET-SORT-OPTIONS kann der Benutzer festlegen, dass SORT auch die Hilfsdateien löscht, die vom Benutzer oder vom aufrufenden Programm angelegt worden sind. Standardmäßig löscht SORT diese Dateien nicht.

Hinweis

Die Sortierzeit und der Durchsatz können verbessert werden, wenn der Benutzer bei der Verwendung von Plattenarbeitsdateien

- genügend Speicherplatz zuweist oder genaue ESTIMATED-RECORDS- bzw. RECORDS-PER-CYCLE-Operanden angibt, und so häufige Sekundärzuweisungen vermeidet;
- die Hilfsdateien auf getrennten privaten Datenträgern einrichtet. SORT kann die Dateien dann sequenziell bearbeiten, ohne dass durch die Positionierung der Plattenarme für mehrere Dateien die Bearbeitung verzögert wird;
- die Hilfsdateien nicht auf den Datenträger der Ein-/Ausgabedateien legt.

3.4.1 Pamkey-Eliminierung bei Hilfsdateien

Hilfsdateien sind SAM-Dateien auf Platte oder Band.

Hilfsdateien auf Platte legt SORT abhängig von der CLASS2-OPTION BLKCTRL=PAMKEY/NONKEY mit BLOCK-CONTROL-INFO=*PAMKEY/*WITHIN-DATA-BLOCK an. Bei BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK ist zu beachten, dass die maximale Länge der von SORT verarbeitbaren Sätze um 8 Byte auf 32751 reduziert wird.

Hilfsdateien auf Band legt SORT mit BLOCK-CONTROL-INFO=*NO an, auch wenn der Anwender *WITHIN-DATA-BLOCK angibt.

3.5 Fixpunktdatei

SORT benötigt für die Ausgabe der Fixpunkte eine Fixpunktdatei. Diese kann sowohl Platten-(nicht NK4-Platten) als auch Banddatei sein. Sie kann vom Benutzer oder von SORT eingerichtet werden. Treten beim Schreiben des Fixpunktes Fehler auf, wird die Meldung SRT1042 ausgegeben, die Verarbeitung wird fortgesetzt.

Von einem Fixpunkt aus kann ein abgebrochener Sortier-/Mischlauf mit dem RESTART-PROGRAM-Kommando erneut gestartet werden.

- Der Benutzer kann eine Fixpunktdatei über ASSIGN-FILES-Anweisung bzw. CREATE-FILE-/ADD-FILE-LINK-Kommando oder FILE-Makroaufruf mit dem Dateikettungsnamen SORTCKPT einrichten.

Für Fixpunktdateien ist das coded character set ohne Bedeutung. Eine Benutzervorgabe wird ignoriert.

Durch Angabe von OPEN-MODE=*INOUT im ADD-FILE-LINK-Kommando bzw. OPEN=INOUT im FILE-Makroaufruf kann eine bestehende Fixpunktdatei fortgeschrieben werden. Bei Plattendateien sollte die Speicherplatzzuweisung genügend groß sein. Als Mindestgröße für die Primärzuweisung gilt:

$$((\text{MEMORY-SIZE-Wert} * 4) + 80)$$

Der optimale Wert für die Primärzuweisung errechnet sich nach:

$$((\text{MEMORY-SIZE-Wert} * 4) + 80) * \text{Fixpunktzahl}$$

Die Sekundärzuweisung sollte einem Fixpunkt entsprechen, also:

$$((\text{MEMORY-SIZE-Wert} * 4) + 80)$$

Das Einrichten einer Fixpunktdatei durch den Benutzer ist nur bei privaten Datenträgern sinnvoll, da SORT von sich aus auf gemeinschaftlichem Datenträger eine Fixpunktdatei einrichtet.

- Hat der Benutzer keine Fixpunktdatei angegeben, richtet SORT eine Plattendatei mit dem Dateikettungsnamen SORTCKPT ein und vergibt den Dateinamen

`SORTCKPT.tsn.jjmmtt.hhmmss`

Dabei bedeuten:

tsn	Prozessfolgennummer (TSN) des Sortierlaufs
jj	Jahresangabe
mm	Monatsangabe
tt	Tagesangabe
hhmmss	sechsstellige Uhrzeitangabe

Bei Angabe des Operanden LINK-PREFIX-CHANGE=<name 1..4> in der SET-SORT-OPTIONS-Anweisung wird der Namensteil „SORTCKPT“ durch „<name 1..4>CKPT“ ersetzt.

Endebehandlung von Fixpunktdateien

- Fixpunktdateien, die der Benutzer eingerichtet hat, werden am Ende des Sortierlaufs geschlossen, aber der Dateikettungsname wird nicht freigegeben. Bänder werden zurückgespult aber nicht entladen.
- Von SORT eingerichtete Fixpunktdateien werden, wenn der Sortier-/ Mischlauf normal endet, geschlossen, freigegeben und gelöscht. Dies gilt auch für den Fall, dass der Sortier-/Mischlauf fehlerhaft endet und keine Fixpunkte ausgegeben wurden. Endet ein Lauf fehlerhaft und gibt SORT Fixpunkte aus, wird die Fixpunktdatei weder freigegeben noch gelöscht, da nur so ein RESTART-PROGRAM möglich ist. Führt der Benutzer keinen RESTART-PROGRAM durch, muss er die Fixpunktdatei selbst löschen.

Hinweis

Bei der Verarbeitung von POSIX-Dateien als Eingabedateien können keine Fixpunkte eingerichtet werden.

3.5.1 Pamkey-Eliminierung bei Fixpunktdateien

Fixpunktdateien legt SORT als PAM-Dateien BLOCK-CONTROL-INFO=*NO an.

Für Fixpunktdateien darf nur BLOCK-CONTROL-INFO=*PAMKEY/*NO angegeben werden. Bei BLKCTRL=*WITHIN-DATA-BLOCK liefert SORT beim Fixpunktschreiben den Return-Code X'48' und bricht den Lauf ab.

3.6 Bindemodulbibliothek SORTMODS

Eine zusätzliche Bindemodulbibliothek kann mit dem Dateikettungsnamen SORTMODS definiert werden um eigene Benutzerroutinen, die SORT-Benutzerausgänge verwenden, zu unterstützen. Die Benutzerroutinen werden dann vorrangig aus der so definierten Bindemodulbibliothek geladen. Diese Bindemodulbibliothek kann auch mit der ASSIGN-FILES-Anweisung zugewiesen werden.

Der Namensteil „SORTMODS“ kann mit der SET-SORT-OPTIONS-Anweisung durch „<name 1..4>MODS“ ersetzt werden.

3.7 Anweisungsdateien

Bei der Anweisungseingabe können an beliebiger Stelle Anweisungen, die in so genannten Anweisungsdateien zusammengefasst sind, eingefügt werden. Bei den Anweisungsdateien ist all das zu beachten, was auch für BS2000-Prozedurdateien gilt.

Über die Anweisung ASSIGN-FILES (Operand STATEMENT-FILES) können bis zu 10 Anweisungsdateien zugeordnet werden. Sie werden in der Reihenfolge ihres Auftretens unmittelbar nach der ASSIGN-FILES-Anweisung abgearbeitet. Voraussetzung dafür ist aber, dass die ASSIGN-FILES-Anweisung von SYSCMD gelesen wurde.

Wird die ASSIGN-FILES-Anweisung mehrfach angegeben, so bleiben die aus Anweisungsdateien vorheriger ASSIGN-FILES-Anweisungen stammenden Anweisungen gültig, solange sie nicht durch gleichnamige überschrieben werden. Anweisungsdateien können nicht geschachtelt werden, d.h. innerhalb einer Anweisungsdatei wird die Zuweisung weiterer ignoriert.

Jede Anweisungsdatei muss mit einer END-Anweisung abgeschlossen werden. Durch die END-Anweisung wird SYSDTA der nächsten Anweisungsdatei oder wieder SYSCMD zugewiesen.

Tritt beim Zuweisen von SYSDTA zu einer Anweisungsdatei ein Fehler auf (z.B. Anweisungsdatei ist eine PAM-Datei), so wird der Lauf mit einer Fehlermeldung abgebrochen.

Wenn in einer Anweisungsdatei ein Syntaxfehler auftritt, der nicht korrigiert wird (z.B. PROCEDURE-DIALOG=*NO), werden die noch vorhandenen Anweisungsdateien verarbeitet, danach wird der Lauf abgebrochen.

Der CCSN der Anweisungsdateien wird beim Einlesen der Anweisungen ausgewertet. Zeichenfolgen als Konstanten in den SORT-Anweisungen werden durch SORT in das CCS der Datensätze konvertiert, bevor diese Konstanten in den Ausgabesatz eingefügt oder mit Feldern des Eingabesatzes verglichen werden.

Anweisungsdateien mit einem ISO-CCSN werden mit SRT1149 und Anweisungsdateien mit einem Unicode-CCSN werden mit SRT1150 zurückgewiesen.

3.8 Endebehandlung der SORT-Dateien

Tabellarische Übersicht der Endebehandlung der SORT-Dateien bei normaler Beendigung des Sortier-/Mischlaufs:

Funktion der Dateien	Dateikettungsname	Erstellung der Datei durch	Dateizustände nach dem Sortier-/Mischlauf				
			Dateien sind		Dateikettungsname ist freigegeben	Dateien auf Magnetbändern: Bänder werden	
			geschlossen	gelöscht		zurückgespult	entladen
Sortier-eingabedatei(en)	SORTIN SORTINxx	Benutzer	ja	nein	ja	ja	ja ¹⁾ nein ²⁾
Misch-eingabedateien	MERGExx	Benutzer	ja	nein	ja	ja	ja ¹⁾ nein ²⁾
Ausgabedatei	SORTOUT	Benutzer	ja	nein	nein	ja	nein
Arbeitsdatei(en)	SORTWK SORTWKx	Benutzer	ja	nein	nein	-	-
		SORT	ja	ja	ja	-	-
Hilfsdateien	SORTWKxx	Benutzer	ja	nein	nein	ja	nein
		SORT	ja	ja	ja	ja	nein
Fixpunktdatei	SORTCKPT	Benutzer	ja	nein	nein	ja	nein
		SORT	ja	ja	ja	nein	nein
Bindemodulbibliothek	SORTMODS	Benutzer	-	-	nein	-	-

¹⁾ Wenn in der SET-SORT-OPTIONS-Anweisung KEEP-INPUT-TAPE=*NO angegeben wurde

²⁾ Wenn in der SET-SORT-OPTIONS-Anweisung KEEP-INPUT-TAPE=*YES angegeben wurde

Bei normaler Beendigung des Sortier-/Mischlaufs schließt SORT alle SORT-Dateien (CLOSE). Die Dateikettungsname der Eingabedateien und der von SORT angelegten Arbeits-, Hilfs- und Fixpunktdateien gibt SORT automatisch frei (RELEASE). Nach einem normalen Ende des Sortier-/Mischlaufs werden die von SORT angelegten Dateien gelöscht (ERASE).

Bandeingabedateien werden zurückgespult und entladen, wenn der Operand KEEP-INPUT-TAPE auf NO gesetzt wurde (UNLOAD). Bandausgabedateien, Bandhilfsdateien und vom Benutzer erstellte Fixpunktdateien werden zurückgespult, aber nicht entladen.

Tabellarische Übersicht der Endebehandlung der SORT-Dateien bei fehlerhafter Beendigung des Sortier-/Mischlaufs:

Funktion der Dateien	Dateikettungsname	Erstellung der Datei durch	Dateizustände nach dem Sortier-/Mischlauf				
			Dateien sind		Dateikettungsname ist freigegeben	Dateien auf Magnetbändern: Bänder werden	
			geschlossen	gelöscht		zurückgespult	entladen
Sortier-eingabedatei(en)	SORTIN SORTINxx	Benutzer	ja	nein	ja	ja	ja ¹⁾ nein ²⁾
Misch-eingabedateien	MERGExx	Benutzer	ja	nein	ja	ja	ja ¹⁾ nein ²⁾
Ausgabedatei	SORTOUT	Benutzer	ja	nein	nein	ja	nein
Arbeitsdatei(en)	SORTWKx	Benutzer	ja	nein	nein	-	-
		SORT	ja	ja [*]	ja [*]	-	-
Hilfsdateien	SORTWKxx	Benutzer	ja	nein	nein	ja	nein
		SORT	ja	ja [*]	ja [*]	ja	nein
Fixpunktdatei	SORTCKPT	Benutzer	ja	nein	nein	ja	nein
		SORT	ja	ja [*]	ja [*]	nein	nein
Bindemodul-bibliothek	SORTMODS	Benutzer	-	-	nein	-	-

^{*}) Nur wenn keine Fixpunkte geschrieben wurden.

¹⁾ Wenn in der SET-SORT-OPTIONS-Anweisung KEEP-INPUT-TAPE=*NO angegeben wurde

²⁾ Wenn in der SET-SORT-OPTIONS-Anweisung KEEP-INPUT-TAPE=*YES angegeben wurde

Bei fehlerhafter Beendigung eines Sortier-/Mischlaufs schließt SORT alle Dateien (CLOSE). Die Dateikettungsname der von SORT angelegten Arbeits-, Hilfs- und Fixpunktdateien werden freigegeben (RELEASE), wenn keine Fixpunkte geschrieben wurden. In diesem Fall werden diese Dateien auch automatisch gelöscht.

Bandeingabedateien werden zurückgespult und entladen wenn der Operand KEEP-INPUT-TAPES auf *NO gesetzt wurde (UNLOAD). Bandausgabedateien, Bandhilfsdateien und vom Benutzer erstellte Fixpunktdateien werden zurückgespult, aber nicht entladen.

3.9 Verarbeitung von POSIX-Dateien mit SORT

Um die Funktionen zur Dateibearbeitung von POSIX-Dateien im BS2000 nutzen zu können, muss das Subsystem POSIX aktiviert sein.

3.9.1 POSIX im BS2000

Die zunehmende Vernetzung unterschiedlicher Rechnersysteme und die verteilte Verarbeitung in diesen Netzen erfordert die Standardisierung und Offenheit der Rechnersysteme und deren Schnittstellen. Diese Schnittstellen müssen den POSIX-/XPG4-Standards entsprechen. Das Betriebssystem BS2000 unterstützt die POSIX-/XPG4-Standards mit dem Softwareprodukt „POSIX“.

Unter POSIX (**P**ortable **O**pen **S**ystem Interface for **U**NIX) bzw. XPG4 (**X**/Open **P**ortability **G**uide Version **4**) versteht man eine Reihe von Standards auf UNIX-Basis. POSIX bezeichnet sowohl diese Standards als auch das Softwareprodukt.

Durch das Softwareprodukt POSIX wird das BS2000 zu einem offenen System. Anwendungen, die dem Standard entsprechen, sind portabel zwischen dem BS2000 und anderen Systemen, die POSIX-Schnittstellen unterstützen, besonders UNIX.

Das POSIX-Dateisystem ist ein Dateisystem im BS2000 mit der Struktur eines UNIX-Dateisystems (UFS). Es ist hierarchisch aufgebaut und besteht aus Dateien (POSIX-Dateien) und Dateiverzeichnissen. POSIX-Benutzer können POSIX-Dateien erzeugen und bearbeiten. POSIX-Benutzer können vom POSIX-Dateisystem aus auf ferne UNIX-Dateisysteme zugreifen. Umgekehrt kann von einem fernen UNIX-Rechner auf das lokale POSIX-Dateisystem zugegriffen werden.

Der Zugang zu POSIX ist BS2000-Benutzern mit entsprechender Berechtigung möglich. Auch von einem UNIX-Rechner aus (über rlogin oder Emulation) kann der Zugang zu POSIX auf einem BS2000-Rechner erfolgen. Die Zugangskontrolle wird vollständig über das BS2000 abgewickelt.

Weitere Informationen zu POSIX im BS2000 finden Sie in den Handbüchern „POSIX Grundlagen für Benutzer und Systemverwalter“ [14] und „POSIX Kommandos“ [15].

3.9.2 Sortieren von POSIX-Dateien mit SORT

Die Daten in den POSIX-Dateien liegen im Text-Format vor, das von SORT nicht unmittelbar verarbeitet werden kann. Die einzelnen Sätze sind bei diesem Format durch Satzende-Kennzeichen begrenzt, die in den folgenden Grafiken durch das Symbol ▼ dargestellt werden.

Text 1 ▼ Text 2 ▼ ... ▼ Text n ▼

Vor der Bearbeitung durch die Sortier Routinen werden diese Daten von SORT in Sätze variabler Länge umgewandelt, denen jeweils ein Satzlängenfeld (slf) vorangestellt wird.

slf Text 1

slf Text 2

:

slf Text n

Nach dem Sortiervorgang wandelt SORT die sortierte Ausgabedatei wieder in das Text-Format um, wenn sie im POSIX-Dateisystem gespeichert werden soll.

Ausgabertext 1 ▼ Ausgabertext 2 ▼ Ausgabertext 3 ▼ ... ▼ Ausgabertext n ▼

Die interne Verwendung von Sätzen variabler Satzlänge bewirkt, dass sich die Position der Benutzerdaten im Satz um das Satzlängenfeld verschiebt. Für den Anwender von POSIX-Dateien hat das im Normalfall jedoch keine Auswirkungen. SORT berechnet die Feldpositionen bei Sätzen aus POSIX-Dateien standardmäßig relativ zum Anfang der Benutzerdaten.

Will der Anwender dennoch auf das interne Satzlängenfeld zugreifen, z.B. um die Sätze nach ihrer Länge zu sortieren, steht der Operand IGNORE-LENGTH-FIELD in der Anweisung SET-SORT-OPTIONS (siehe Seite 169) und im Kommando SORT-FILE (siehe Seite 198) zur Verfügung.

Die Angabe IGNORE-LENGTH-FIELD=*NO bewirkt, dass sowohl bei variabel langen Sätzen in BS2000-Dateien, als auch bei Sätzen in POSIX-Dateien die Positionen innerhalb des Satzes ab Satzanfang berechnet werden. Damit beginnen die Benutzerdaten an Position 5 im Satz.

Die Verschlüsselung des Satzende-Kennzeichens wird durch den Operanden CODE in der Anweisung ASSIGN-FILES (siehe Seite 141) und im Kommando SORT-FILE (siehe Seite 198) bestimmt. Bei CODE=*EBCDIC wird das Satzende-Kennzeichen als X'15' verschlüsselt, bei CODE=*ASCII als X'0A'.

Bei der Verwendung von POSIX-Dateien als Ausgabedatei ist darauf zu achten, dass die Ausgabesätze keine Zeichen enthalten, die als Satzende-Kennzeichen interpretiert werden. Das bedeutet im Einzelnen:

- In der Anweisung SORT-RECORDS oder im Kommando SORT-FILE dürfen keine Konstantenfelder angegeben werden, die Satzende-Kennzeichen enthalten.
- Die Sätze einer BS2000-Eingabedatei dürfen keine Satzende-Kennzeichen enthalten, wenn die Ausgabedatei eine POSIX-Datei sein soll.
- Die Sortierart Adresslistensortieren darf nicht verwendet werden, da nicht sichergestellt werden kann, dass die Adressfelder keine als Satzende-Kennzeichen interpretierbaren Zeichen enthalten.

Hinweis

Das Kommando „sort“, welches in einer POSIX-Shell aufgerufen werden kann, ist nicht identisch mit einem Aufruf des Produkts SORT.

3.10 SORT und ACS

Wenn das Subsystem ACS (Alias Catalog Service) eingesetzt ist, sind folgende Auswirkungen zu beachten:

1. Existiert ein vom Benutzer angelegter TFT-Eintrag, so kann nicht festgestellt werden, ob eine Ersetzung des Dateinamens vorgenommen wurde oder nicht. Deshalb wird bei Verwendung des dort entnommenen Dateinamens (z.B. bei SHOW-FILE-ATTRIBUTES) ACS unterbrochen (HOLD-ALIAS-SUBSTITUTION).
2. Legt SORT selbst einen TFT-Eintrag mit dem FILE-Makro an, so bleibt ACS in dem Zustand wie beim Start von SORT, d.h. es können Ersetzungen von Dateinamen durchgeführt werden. Damit besteht kein Unterschied bezüglich der Dateinamen zwischen den TFT-Einträgen des Benutzers und denen von SORT.
3. Gibt ein Benutzer in einer ASSIGN-FILES-Anweisung einen Dateinamen an, so muss er, falls bereits ein TFT-Eintrag vorhanden ist, bedenken, ob eine Ersetzung des Dateinamens durchgeführt wurde oder nicht. Der Dateiname in der Anweisung wird zum Dateinamen-Vergleich in jedem Fall ohne irgendwelche Ersetzungen herangezogen.
4. Ändert der Benutzer zwischen dem Anlegen der TFT-Einträge und dem Start von SORT den Zustand von ACS (z.B. durch verspätetes Laden, Unterbrechen usw.), so kann es zu Unklarheiten bei den letztendlich verwendeten Dateinamen kommen.

3.11 Arbeiten mit Dateien größer 32 GByte

Dem stetigen Wachstum der Plattenspeicherkapazität und von online bereitzuhaltenden Datenbeständen wurde in BS2000/OSD-BC V5.0 durch die Erweiterung der bisher verfügbaren Platten- und Dateigrößen von ca. 32 GByte Rechnung getragen.

Ab BS2000/OSD-BC V5.0 beträgt:

- die maximale Kapazität einer einzelnen Platte ca. 4 TByte (2.147.483.647 PAM-Seiten)
- die maximale Dateigröße ebenfalls ca. 4 TByte (2.147.483.647 PAM-Seiten)

SORT V8.0 unterstützt diese Dateigröße von mehr als 32 Gbyte bei Eingabe-, Ausgabe-, Arbeits- und Hilfsdateien.

Wegen der zahlreichen unterschiedlichen Konstellationen beim Sortieren ist es möglich, dass nur eine, mehrere oder alle beteiligten Dateien die Grenze von 32 GByte überschreiten. SORT erlaubt für die verwendeten Dateien prinzipiell Dateigrößen von mehr als 32 GByte. Allerdings muss der Anwender dafür sorgen, dass Dateien größer 32 GByte auf einem Pubset angelegt werden, das für solche Dateien geeignet ist und auf dem genügend Platz zur Verfügung steht.

Weitere Informationen zu Dateien größer 32 GByte finden Sie im Handbuch „Einführung in das DVS“ [2].

3.11.1 Anlegen von Dateien größer 32 GByte

Dateien, deren Größe 32 GByte überschreiten kann, sollten vor dem Sortierlauf mit CREATE-FILE/ADD-FILE-LINK angelegt werden. Damit ist gewährleistet, dass eine genügend große Datei auf einem geeigneten Pubset bereitgestellt wird.

Das Anlegen von Ausgabe-, Arbeits- und Hilfsdateien kann jedoch auch dem SORT überlassen werden. Hierzu müssen in der ASSIGN-FILES-Anweisung die Namen der anzulegenden Dateien inklusive Katalogkennung (cat-id) angegeben werden. Diese Katalogkennung muss sich auf ein Pubset beziehen, das für Dateien ≥ 32 GByte geeignet ist. Die Größe der anzulegenden Datei kann jedoch nur für die Arbeitsdatei mit der Anweisung //ASSIGN-RESOURCES ...,DISK-SPACE=<integer 1..2147483647>... explizit vorgegeben werden.

Für Arbeits- und Hilfsdateien kann die Angabe des Dateinamens entfallen, wenn das Default-Pubset Dateien ≥ 32 GByte erlaubt. In diesem Fall legt SORT die Dateien bei Bedarf unter einem selbst generierten Namen auf dem Default-Pubset an.

Erzwingen von Dateien größer 32 GByte

Soll bereits vor dem Sortierlauf geprüft werden, ob für die einzelnen Dateien genügend Speicherplatz zur Verfügung steht, so muss die jeweilige Datei mit entsprechender SPACE-Angabe bereitgestellt werden. Hierbei ist insbesondere der Wert für die Primärzuweisung von Bedeutung, da der Wert der Sekundärzuweisung 32767 PAM-Seiten nicht überschreiten kann. Dies gilt auch für den automatisch errechneten Wert für die Sekundärzuweisung, wenn mit der Angabe DISK-SPACE der ASSIGN-RESOURCES-Anweisung ein größerer Wert für die Primärzuweisung festgelegt wird.

Bei Dateien ≥ 32 GByte muss also mehr als bei kleineren Dateien auf eine ausreichende Primärzuweisung geachtet werden, weil der Speicherplatz, der mittels Sekundärzuweisung zur Verfügung gestellt wird, einen geringeren Anteil an der Gesamtmenge hat.

3.11.2 Adresslistensortieren

Beim Adresslistensortieren gibt es für Eingabedateien ≥ 32 GByte abhängig vom Dateityp folgende Einschränkungen:

- SAM-Dateien

Bei den Varianten *TAG-HEADER und *TAG-TRAILER werden vier Byte lange Wiedergewinnungs-Adressfelder aufgebaut (drei Byte für die Nummer des Datenblocks und ein Byte für die relative Satznummer im Block). Bei Eingabedateien ≥ 32 GByte kann der Fall eintreten, dass die Nummer des Datenblocks in drei Byte nicht mehr dargestellt werden kann. Dann wird die Sortierung mit der Meldung SRT1264 abgelehnt.

Bei der Variante *TAG-COMPOUND gibt es keine Einschränkungen.

- PAM-Dateien

Alle Wiedergewinnungsadressen sind vier Byte lang. Bei Eingabedateien ≥ 32 GByte werden alle Varianten des Adresslistensortierens zurückgewiesen.

- ISAM-Dateien

Für Eingabedateien ≥ 32 GByte gibt es keine Einschränkungen gegenüber Dateien < 32 GByte.

3.11.3 Empfehlungen zur Arbeit mit großen Dateien

Wegen der großen Laufzeiten beim Sortieren von großen Dateien kann es sinnvoll sein, Prüfpunkte zu setzen. Das setzt natürlich voraus, dass mit dem Zyklen- oder dem Multitasksortieren gearbeitet wird. Der erhöhte Zeitaufwand wird mit einer größeren Sicherheit honoriert. Sortierläufe dieser Größe können durchaus 24 Stunden und länger dauern.

Um möglichst lange Folgen in der Vorsortierphase (und damit kurze Sortierzeiten) zu erlangen, ist es unbedingt zu empfehlen, in der Anweisung ASSIGN-RESOURCES einen großen Wert für MEMORY-SIZE anzugeben (z.B. 10000 und größer). Das Optimum für diesen Wert hängt von folgenden Faktoren ab:

- Größe des Realspeicher des verwendeten Rechners,
- Maximalwert für den virtuellen Speicher der Kennung,
- andere Benutzer, die parallel zu SORT arbeiten.

Durch unzureichend großen Realspeicher und Parallelarbeit kann es zu verstärktem Paging kommen. Dadurch wird der Vorteil der langen Folgen wieder aufgehoben. Bei sehr großen Sortierläufen ist anzustreben, dass der Rechner exklusiv für SORT zur Verfügung steht.



Wird für MEMORY-SIZE ein wesentlich zu großer Wert angegeben, so kann sich die Laufzeit des SORT stark vergrößern, weil die Seiten des virtuellen Speichers zwischenzeitlich aus- und eingelagert werden.

- Die Größe des Realspeichers (in MByte) ermitteln Sie mit
`/SHOW-SYSTEM-INFORMATION INFORMATION=MEMORY-SIZE`
- Die maximale Größe des virtuellen Speichers der Kennung (in MByte) ermitteln Sie mit
`/SHOW-USER-ATTRIBUTES (Feld ADDRESS-SPACE-LIMIT)`

Den Wert für die maximale Größe des virtuellen Speichers kann die Systembetreuung gegebenenfalls erhöhen.

Als MEMORY-SIZE-Wert sind etwa 70% des kleineren der beiden Werte zu empfehlen. MEMORY-SIZE gibt den Wert des zu verwendenden Arbeitsspeichers in CORE-Seiten (4096 Byte) an.

Ist der Realspeicher sehr klein, so kann der Wert für MEMORY-SIZE den Wert des Realspeichers auch überschreiten. Die Größe des virtuellen Speichers kann nicht überschritten werden.

Beispiel:

Größe Realspeicher	120 MByte
Größe virtueller Speicher der Kennung	128 MByte
Wert für MEMORY-SIZE in ASSIGN-RESOURCES	20480 (entspricht 80 MByte)

Der Wert MEMORY-SIZE in ASSIGN-RESOURCES wird u.U. durch den Grenzwert CORE-MAXIMUM des Systems für SORT begrenzt. In diesem Fall kann dieser Wert mit der Angabe CORE-MAXIMUM der Anweisung MODIFY-SORT-DEFAULTS erhöht werden.

Der Wert für CORE-MAXIMUM muss mindestens 1/16 des Wertes von MEMORY-SIZE betragen, sonst wird MEMORY-SIZE entsprechend verringert.

Ein ausreichend großer Wert für MEMORY-SIZE liegt dann vor, wenn es nur einen Vorsortierlauf und einen Endmischlauf, aber keinen Zwischenmischlauf gibt (erkennbar bei MIN-MSG-WEIGHT=*ALL).

Damit große MEMORY-Werte von SORT auch tatsächlich genutzt werden können, muss SORT im oberen Adressraum ablaufen. Dies erreichen Sie mit der Angabe PROG-MODE=*ANY beim Kommando START-SORT oder indem Sie SORT mit dem Kommando SORT-FILE starten.

4 Anweisungen von SORT

4.1 Eingabequellen

Die Anweisungen können bei Verwendung von SORT als autonomes Programm oder Unterprogramm auf Ebene 0 von einer Datensichtstation, Prozedur- oder Enterdatei eingegeben werden. Wird SORT als Unterprogramm auf Ebene 1 genutzt, müssen Anweisungen im virtuellen Speicher bereitgestellt werden.

Für jeden Sortier-/Mischlauf muss entweder eine SORT-RECORDS-Anweisung oder eine MERGE-RECORDS-Anweisung angegeben werden. Jede Definition eines Sortier-/Mischlaufes muss mit der END-Anweisung abgeschlossen werden.

Mit Ausnahme der END-Anweisung und der ADD-SYMBOLIC-NAME-Anweisung können alle Anweisungen eines Sortier-/Mischlaufes in beliebiger Reihenfolge angegeben werden. Die END-Anweisung muss immer als letzte Anweisung gegeben werden und darf nur einmal vorkommen. Die ADD-SYMBOLIC-NAME-Anweisung muss immer vor der Anweisung stehen, in der der zugewiesene symbolische Name genutzt wird. Im Dialogbetrieb können Anweisungen eines Typs mehrfach vorkommen. Im Prozedur-, Stapel- und Ebene-1-Betrieb wird der SORT-Lauf abgebrochen, wenn Anweisungen eines Typs mehrfach vorkommen.

SORT-Anweisungen sind der Form nach Sätze variabler Länge. Die Längenangabe steht im ersten Halbwort des Satzlängenfeldes, das dem eigentlichen Satz vorangestellt ist. Die SORT-Anweisungen werden über die Systemdatei SYSDTA eingelesen. Werden die Anweisungen direkt an der Datensichtstation eingegeben, wird die Länge durch das System ergänzt.

Beim Aufruf von SORT als Unterprogramm können die SORT-Anweisungen auch direkt im Hauptspeicher an das Sortier-/Mischprogramm übergeben werden.

4.2 SDF-Syntaxdarstellung

Details zur allgemeinen SDF-Metasyntax, Datentypen sowie zur Eingabe von Kommandos bzw. Anweisungen finden Sie im Handbuch „Dialogschnittstelle SDF“ [7] bzw. „Kommandos“ [1].

4.3 Fehlerbehandlung

Syntaxfehler werden von SDF selbst behandelt. Semantische Fehler innerhalb einer Anweisung (z.B. widersprüchliche Operanden) werden folgendermaßen behandelt:

- Im Dialog- und Prozedurbetrieb mit der SDF-Option `PROCEDURE-DIALOGUE=*YES` wird ein Korrekturdiallog geführt.
- Im Prozedurbetrieb mit der SDF-Option `PROCEDURE-DIALOGUE=*NO` und bei Verwendung von `SORT` als Unterprogramm auf Ebene 1 wird auf die nächste `STEP-` oder `END-`Anweisung verzweigt.
- Anweisungsdateien werden wie Prozedurdateien behandelt.

Ablauf des Korrekturdiallogs

Nachdem über Meldungen auf die Art der Fehler hingewiesen wurde, wird die Korrektur der fehlerhaften Operanden über die Datensichtstation angefordert. Dieser Vorgang wird solange wiederholt, bis die Anweisung fehlerfrei vorliegt.

4.4 Kurzübersicht der SORT-Anweisungen

Operation	Anwendungsbereich	Alte ISP-Anweisung
ADD-SYMBOLIC-NAMES	Zuordnung symbolischer Namen zu Sortier-, Summier-, Vergleichs- und Restfeldern, sowie zu Konstanten und Masken.	
ASSIGN-EXITS	Ermöglicht den Anschluss von Benutzerrouninen an definierten SORT-Ausgängen.	MODS
ASSIGN-FILES	Zuweisen der Eingabe-, Ausgabe-, Arbeits-, Hilfs-, Checkpoint- und Anweisungsdateien.	
ASSIGN-RESOURCES	Größenangabe von Haupt- und Plattenspeichern und der Anzahl von Bandgeräten.	ALLOC
END	Beenden der Anweisungseingabe und Starten des Sortier-/Mischlaufes.	END
MERGE-RECORDS	Anweisung für eine Mischung, beschreibt die Sortierfelder sowie deren wesentliche Kenndaten.	MERGE
MODIFY-CODE	Festlegen und Ändern der Code-Sortierreihenfolge für MODIFY-CODE-Sortierfelder.	NEWCOL
MODIFY-SORT-DEFAULTS	Ändern von Voreinstellungen.	
SELECT-INPUT-RECORDS	Sätze der Eingabedatei(en) können mit Hilfe eines logischen Ausdruckes ausgewählt werden.	OMIT, INCLUDE
SET-RECORD-ATTRIBUTES	Beschreibung der Eingabe-, Internen- und Ausgabe-sätze bezüglich Format, Länge und eventueller Auffüllung.	RECORD
SET-SORT-OPTIONS	Angabe von Optionen für die Meldungs Ausgabe, der Behandlung der Link-Namen, der Fehlerbehandlung usw.	OPTION
SHOW-SORT-DEFAULTS	Anzeigen von Voreinstellungen.	
SORT-RECORDS	Anweisung zur Sortierung, beschreibt die Sortierfelder und deren wesentliche Kenndaten.	SORT
SUM-RECORDS	Sätze mit gleichen Sortierbegriffen können zu einem Satz zusammengefasst werden. Summenfelder werden addiert.	SUM

Die Standardanweisungen von SDF können zusätzlich gegeben werden. Sie werden nicht in diesem Handbuch beschrieben. Eine Beschreibung findet sich im Handbuch „Dialog-schnittstelle SDF“ [7].

ADD-SYMBOLIC-NAMES

Funktion: Zuweisen von symbolischen Namen für

- Sortierfelder
- Summenfelder
- Vergleichsfelder
- Restfelder
- Konstanten
- Druckaufbereitungsmasken.

Symbolische Namen können in folgenden Anweisungen verwendet werden:

- MERGE-RECORDS
- SELECT-INPUT-RECORDS
- SORT-RECORDS
- SUM-RECORDS

Insgesamt können 255 symbolische Namen vereinbart werden.

ADD-SYMBOLIC-NAMES

```

FIELDS = *NO / list-poss(255): <name 1..20>(…)
  <name 1..20>(…)
    |
    | POSITION = <integer 1..32759>(…)
    |   <integer 1..32759>(…)
    |     |
    |     | BIT-POSITION = 0 / <integer 0..7>
    |     |
    |     | LENGTH = <integer 0..32759>(…)
    |     |   <integer 0..32759>(…)
    |     |     |
    |     |     | NUMBER-OF-BITS = 0 / <integer 0..7>
    |     |     |
    |     |     | FORMAT = *CHARACTER / *NO / *BINARY / *FIXED-POINT / *FLOATING-POINT /
    |     |     |   *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN /
    |     |     |   *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE /
    |     |     |   *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER /
    |     |     |   *TRANSLATE-CHARACTER / *UNICODE-CHARACTER
    |
    | CONSTANTS = *NO / list-poss(255): <name 1..20>(…)
    |   <name 1..20>(…)
    |     |
    |     | VALUE = <integer -2147483639..2147483639> / <c-string 1..256 with-low> /
    |     |   <x-string 1..512>
    |
    | PRINT-MASKS = *NO / list-poss(255): <name 1..20>(…)
    |   <name 1..20>(…)
    |     |
    |     | FORM = <c-string 1..254 with-low>

```

FIELDS =

Es dürfen maximal 255 symbolische Namen für Sortier-, Summen-, Vergleichs- und Restfelder bezüglich Position, Länge und Format für die Anweisungen SORT-RECORDS, MERGE-RECORDS, SUM-RECORDS und SELECT-INPUT-RECORDS vereinbart werden.

FIELDS = *NO

Es werden keine symbolischen Namen für Felder vereinbart.

FIELDS = list-poss(255): <name 1..20(...)>

Symbolischer Name des Feldes.

POSITION = <integer 1..32759(...)>

Position des Feldes relativ zum Satzanfang. Die Positionsangabe muss innerhalb der erlaubten Grenzen liegen. Diese sind beim Operanden POSITION der Anweisung beschrieben, für die das Feld vereinbart wird.

BIT-POSITION = 0 / <integer_0..7>

Position des binären Feldes relativ zum Feldanfang. Die Angabe von BIT-POSITION ist nur für Sortierfelder mit Format BINARY erlaubt.

LENGTH = <integer 0..32759(...)>

Länge des Feldes. Die Längenangabe muss innerhalb der erlaubten Grenzen liegen. Diese sind bei der Anweisung, für die das Feld vereinbart wird, beim Operanden LENGTH beschrieben.

NUMBER-OF-BITS = 0 / <integer 0..7>

Längenangabe in Bits zusätzlich zur Längenangabe in Bytes. Die Angabe von NUMBER-OF-BITS ist nur für Sortierfelder mit Format BINARY erlaubt.

FORMAT = *CHARACTER / *NO / *BINARY / *FIXED-POINT / *FLOATING-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN / *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE / *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER / *TRANSLATE-CHARACTER / *UNICODE-CHARACTER

Format des Sortierfeldes (Eigenschaften siehe [Seite 38](#)). Die Formatangabe muss bei der Anweisung, für die das Feld vereinbart wird, möglich sein. Die Angabe von FORMAT=*NO ist nur in Zusammenhang mit Restfeldern sinnvoll (siehe Anweisung SORT-RECORDS, [Seite 186](#)).

CONSTANTS =

Symbolischer Name der Konstanten für die Anweisungen SORT-RECORDS und SELECT-INPUT-RECORDS. Die Länge aller angegebenen Konstanten und Druckaufbereitungsmasken darf 5000 Byte nicht überschreiten.

CONSTANTS = *NO

Es werden keine symbolischen Namen für Konstanten vereinbart.

CONSTANTS = list-poss(255): <name 1..20(...)>

Symbolischer Name der Konstanten.

VALUE =

Wert der Konstanten. Die erlaubten Werte sind bei der Anweisung, für die diese Konstante vereinbart wird, angegeben.

VALUE = <integer -2147483639..2147483639>

Dezimalkonstante. Wird von SORT in eine Festpunktzahl in der Länge von 4 Byte umgewandelt.

VALUE = <c-string 1..256 with-low>

Zeichenkonstante.

VALUE = <x-string 1..512>

Sedezimalkonstante.

PRINT-MASKS =

Zuordnung von symbolischen Namen zu Druckaufbereitungsmasken für die Anweisungen SORT-RECORDS und SUM-RECORDS. Die Länge aller angegebenen Konstanten und Druckaufbereitungsmasken darf 5000 Byte nicht überschreiten, wobei für jede Maske zusätzlich 1 Byte für ein Längenfeld zu berücksichtigen ist.

PRINT-MASKS = *NO

Es werden keine symbolischen Namen für Druckaufbereitungsmasken vereinbart.

PRINT-MASKS = list-poss(255): <name 1..20(...)>

Name der Druckaufbereitungsmaske.

FORM = <c-string 1..254 with-low>

Form der Druckaufbereitungsmaske. Es sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Maske.
- Die Steuerzeichen '#' (Nummernzeichen) und '^' (Circumflex).
- Einzufügende Zeichen (ungleich den Steuerzeichen).

ASSIGN-EXITS

Funktion: Verbinden der von SORT zur Verfügung gestellten Benutzerausgänge mit den jeweiligen Benutzer-Routinen.

SORT verwendet Module aus Bibliotheken. Diese Bibliotheken müssen zuvor auf eine der folgenden Arten zugewiesen worden sein:

- //ASSIGN-FILES MODULE-LIBRARY=bibliotheksname
- /ADD-FILE-LINK LINK-NAME=SORTMODS
- /ADD-FILE-LINK LINK-NAME=<BLSLIB00..BLSLIB99>
- /SET-TASKLIB LIBRARY=bibliotheksname

SORT sucht die Benutzer-Routinen folgendermaßen:

- Wenn eine Bibliothek mit //ASSIGN-FILES MODULE-LIBRARY=... (siehe [Seite 141](#)) oder mit dem Dateikettungsnamen SORTMODS zugewiesen ist, versucht SORT, die Routine aus dieser Bibliothek zu laden.

Wenn SORT die geforderte Benutzer-Routine nicht in dieser Bibliothek findet, dann sucht er sie in den Bibliotheken mit den Dateikettungsnamen BLSLIB00 - BLSLIB99 in aufsteigender Reihenfolge (00, 01, .., 99). Die Nummern müssen dabei nicht lückenlos vorhanden sein.

- Ist keine Bibliothek mit dem Dateikettungsnamen SORTMODS oder mit //ASSIGN-FILES MODULE-LIBRARY=... zugewiesen, so wird eine mit dem Kommando SET-TASKLIB bereitgestellte Bibliothek durchsucht.

ASSIGN-EXITS

```

PLANNING = *NO / *DIALOG / *TERMINATE-ABNORMAL
, INPUT = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
        |
        | , PARAMETER-MODE = 24 / *ANY
, OUTPUT = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
        |
        | , PARAMETER-MODE = 24 / *ANY
, EXLST-FOR-INPUT = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
, EXLST-FOR-OUTPUT = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
, WORK-FILE-OVERFLOW = *NO / *DIALOG / *FINISH-INPUT / *TERMINATE-ABNORMAL / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
        |
        | , INTERFACE-VERSION = 1 / <integer 1..2>
, PHYSICAL-TRANSLATE = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
, VIRTUAL-TRANSLATE = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
, EXTERNAL-COMPARE = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>
        |
        | , PARAMETER-MODE = 24 / *ANY
, TRANSLATE-CHARACTER = *NO / *MODULE(...)
    *MODULE(...)
        |
        | NAME = <name 1..8>

```

PLANNING =

Wird aktiviert, nachdem die Planung abgeschlossen und die Sortierstrategie ausgewählt wurde.

PLANNING = *NO

Der Exit wird nicht aktiviert.

PLANNING = *DIALOG

SORT gibt die geschätzte Größe der Arbeitsdatei (in PAM-Seiten) und die Anzahl der angeforderten CORE-Seiten aus. Danach sind die folgenden Maßnahmen zulässig:

CONTINUE Die Verarbeitung wird fortgesetzt.

START Der SORT-Lauf wird neu gestartet. Nur die geänderten Anweisungen müssen neu eingegeben werden. START ist nur bei autonomen SORT-Läufen und beim Aufruf von SORT als Unterprogramm über Ebene 0 zulässig.

TERMINATE Der SORT-Lauf wird abgebrochen.

Wird im Prozedur- oder Stapelbetrieb keine Maßnahme angegeben, so wird mit CONTINUE fortgesetzt.

PLANNING = *TERMINATE-ABNORMAL

Der SORT-Lauf wird abgebrochen.

INPUT =

Wird aktiviert, wenn SORT einen Satz von der Eingabe übernimmt. Der Satz kann überprüft, geändert oder gelöscht werden. Es ist auch möglich, neue Sätze einzufügen. Die Benutzerroutine kann auch die gesamte Eingabe übernehmen.

INPUT = *NO

Der Exit wird nicht aktiviert.

INPUT = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden und aufgerufen wird.

NAME = <name 1..8>

Name des Benutzermoduls.

PARAMETER-MODE =

Beschreibt die Schnittstelle zwischen SORT und Benutzermodul und den Adressierungsmodus in dem der Benutzermodul abläuft. Für einen Benutzermodul, der nur im 24-Bit-Adressierungsmodus ablauffähig ist, muss PARAMETER-MODE=24 gesetzt sein und die entsprechende Schnittstelle verwendet werden.

PARAMETER-MODE = 24

Der Benutzermodul kann nur im 24-Bit-Adressierungsmodus ablaufen.

PARAMETER-MODE = *ANY

Der Benutzermodul kann im 24- oder 31-Bit-Adressierungsmodus ablaufen.

OUTPUT =

Wird aktiviert, bevor SORT in die Ausgabedatei schreibt. Der Satz kann überprüft, geändert oder gelöscht werden. Es ist auch möglich, neue Sätze einzufügen. Die Benutzerroutine kann auch die gesamte Ausgabe übernehmen.

OUTPUT = *NO

Der Exit wird nicht aktiviert.

OUTPUT = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden und aufgerufen wird.

NAME = <name 1..8>

Name des Benutzermoduls.

PARAMETER-MODE =

Beschreibt die Schnittstelle zwischen SORT und Benutzermodul und den Adressierungsmodus in dem der Benutzermodul abläuft. Für einen Benutzermodul, der nur im 24-Bit-Adressierungsmodus ablauffähig ist, muss PARAMETER-MODE=24 gesetzt sein und die entsprechende Schnittstelle verwendet werden.

PARAMETER-MODE = 24

Der Benutzermodul kann nur im 24-Bit-Adressierungsmodus ablaufen.

PARAMETER-MODE = *ANY

Der Benutzermodul kann im 24- oder 31-Bit-Adressierungsmodus ablaufen.

EXLST-FOR-INPUT =

Angabe von EXLST-Ausgängen für die Eingabedatei(en).

EXLST-FOR-INPUT = *NO

Der Exit wird nicht aktiviert.

EXLST-FOR-INPUT = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

EXLST-FOR-OUTPUT =

Angabe von EXLST-Ausgängen für die Ausgabedatei.

EXLST-FOR-OUTPUT = *NO

Der Exit wird nicht aktiviert.

EXLST-FOR-OUTPUT = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

WORK-FILE-OVERFLOW =

Wird aktiviert, wenn der Überlauf einer Plattenarbeitsdatei mit Sekundärzuweisung = 0 unmittelbar bevorsteht, SORT den Engpass nicht beseitigen kann und keine Hilfsdatei für einen weiteren Zyklus zur Verfügung steht.

WORK-FILE-OVERFLOW = *NO

Der Exit wird nicht aktiviert.

WORK-FILE-OVERFLOW = *DIALOG

Die Anzahl der bis dahin von SORT übernommenen Sätze wird angezeigt und eine der folgenden Maßnahmen erwartet:

CONTINUE SORT versucht, mit einer geringen Reserve den SORT-Lauf durchzuführen.

FINISH SORT beendet die Satzeingabe und bearbeitet die bereits übernommenen Sätze.

TERMINATE SORT bricht den Lauf ab.

WORK-FILE-OVERFLOW = *FINISH-INPUT

Die Satzeingabe wird beendet und SORT bearbeitet die bisher eingelesenen Sätze.

WORK-FILE-OVERFLOW = *TERMINATE-ABNORMAL

Der Sortier-/Mischlauf wird abgebrochen.

WORK-FILE-OVERFLOW = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden und aufgerufen wird.

NAME = <name 1..8>

Name des Benutzermoduls.

INTERFACE-VERSION =

Version der Schnittstelle, die SORT bei Aufruf des Exit-Moduls verwenden soll (siehe [Abschnitt „WORK-FILE-OVERFLOW – Arbeitsdatei-Überlauf“ auf Seite 254](#)).

INTERFACE-VERSION = 1

SORT verwendet die alte Schnittstelle mit 4-Byte Satzzähler, es können maximal 2.14.483.639 Sätze verarbeitet werden.

INTERFACE-VERSION = 2

SORT verwendet die neue Schnittstelle mit 8-Byte Satzzähler, es können maximal bis zu 9.223.372.036.854.775.807 - 8 Sätze verarbeitet werden.

PHYSICAL-TRANSLATE =

Anschluss einer Benutzerroutine, die aus zwei Code-Tabellen (je 256 Zeichen) besteht. Durch die Code-Tabellen wird die Sortierreihenfolge für Felder mit dem Format PHYSICAL-TRANSLATE festgelegt. Mit der ersten Tabelle werden die Sortierfelder nach der Eingabe umcodiert, mit der zweiten Tabelle vor der Ausgabe wieder rückcodiert (siehe [Kapitel „Benutzerausgänge von SORT“ auf Seite 237](#)).

PHYSICAL-TRANSLATE = *NO

Der Exit wird nicht aktiviert.

PHYSICAL-TRANSLATE = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

VIRTUAL-TRANSLATE =

Anschluss einer Benutzerroutine, die aus einer Code-Tabelle (256 Zeichen) besteht. Durch die Code-Tabelle wird die Sortierreihenfolge für Felder mit dem Format VIRTUAL-TRANSLATE festgelegt. Vor den Vergleichen wird das Sortierfeld in ein Hilfsfeld umcodiert. Das Sortierfeld wird nicht verändert (siehe [Kapitel „Benutzerausgänge von SORT“ auf Seite 237](#)).

VIRTUAL-TRANSLATE = *NO

Der Exit wird nicht aktiviert.

VIRTUAL-TRANSLATE = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

EXTERNAL-COMPARE =

Wird bei jedem Satzvergleich für Felder mit der Sortierreihenfolge EXTERNAL-COMPARE aktiviert. Der Benutzermodul kann dann die Reihenfolge bestimmen.

EXTERNAL-COMPARE = *NO

Der Exit wird nicht aktiviert.

EXTERNAL-COMPARE = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

PARAMETER-MODE =

Beschreibt die Schnittstelle zwischen SORT und Benutzermodul und den Adressierungsmodus in dem der Benutzermodul abläuft. Für einen Benutzermodul, der nur im 24-Bit-Adressierungsmodus ablauffähig ist, muss PARAMETER-MODE=24 gesetzt sein und die entsprechende Schnittstelle verwendet werden.

PARAMETER-MODE = 24

Der Benutzermodul kann nur im 24-Bit-Adressierungsmodus ablaufen.

PARAMETER-MODE = *ANY

Der Benutzermodul kann im 24- oder 31-Bit-Adressierungsmodus ablaufen.

TRANSLATE-CHARACTER =

Anschluss einer Benutzerroutine, die aus zwei Code-Tabellen (je 256 Zeichen) besteht. Durch die Code-Tabellen wird die Sortierreihenfolge für Felder mit dem Format TRANSLATE-CHARACTER bestimmt, wenn der CCSN der Eingabedatei dem Namen des angegebenen Moduls entspricht (siehe [Kapitel „Benutzerausgänge von SORT“ auf Seite 237](#)).

TRANSLATE-CHARACTER = *NO

Der Exit wird nicht aktiviert.

TRANSLATE-CHARACTER = *MODULE(...)

Angabe eines Benutzermoduls, der von SORT eingebunden wird.

NAME = <name 1..8>

Name des Benutzermoduls.

ASSIGN-FILES

Funktion: Zuweisen der Eingabe-, Ausgabe-, Arbeits-, Hilfs-, Fixpunkt- und Anweisungsdateien, sowie einer Bibliothek mit Benutzermodulen.

ASSIGN-FILES

```

INPUT-FILES = *LINK / list-poss(99): <filename 1..54> / <posix-pathname 1..1023>
, OUTPUT-FILE = *LINK / <filename 1..54> / <posix-pathname 1..1023>
, WORK-FILES = *STD / list-poss(9): <filename 1..54>
, AUXILIARY-FILES = *STD / list-poss(99): <filename 1..54>
, CHECKPOINT-FILE = *STD / <filename 1..54>
, MODULE-LIBRARY = *STD / <filename 1..54>
, STATEMENT-FILES = *NONE / list-poss(10): <filename 1..54>
, CODE = *EBCDIC / *ASCII

```

INPUT-FILES =

Zuordnung der Eingabedatei(en) (max. 99).

INPUT-FILES = *LINK

Die Eingabedateien sind über ADD-FILE-LINK-Kommandos zugewiesen (Dateikettungsname SORTIN, SORTINxx oder MERGExx).

Standardmäßig wird der Dateikettungsname bei Beendigung von SORT wieder freigegeben. Bei Angabe des Kommandos LOCK-FILE-LINK vor dem Aufruf von SORT bleibt der Dateikettungsname erhalten.

INPUT-FILES = list-poss(99): <filename 1..54> / <posix-pathname 1..1023>

Name(n) der Eingabedatei(en). Eingabedateien, die über ADD-FILE-LINK-Kommando oder in einer vorangegangenen ASSIGN-FILES-Anweisung zugewiesen wurden, werden ignoriert.

POSIX-Dateinamen müssen zur Unterscheidung von BS2000-Dateinamen in Hochkomma angegeben werden.

POSIX-Eingabedateien und BS2000-Eingabedateien dürfen in einem Sortierlauf nicht gleichzeitig verwendet werden. Falls eine Eingabedatei eine POSIX-Datei ist, müssen alle anderen Eingabedateien ebenfalls POSIX-Dateien sein.

Einschränkung:

Die Summe aus der Anzahl und den Längen aller angegebenen POSIX-Eingabedateinamen darf nicht größer als 5100 sein. Darüber hinausgehende Angaben führen zu einer Fehlermeldung mit anschließendem Abbruch des SORT.

OUTPUT-FILE =

Zuordnung der Ausgabedatei.

OUTPUT-FILE = *LINK

Die Ausgabedatei ist über ADD-FILE-LINK-Kommando zugewiesen (Dateikettungsname SORTOUT).

OUTPUT-FILE = <filename 1..54> / <posix-pathname 1..1023>

Name der Ausgabedatei. Ausgabedateien, die über ADD-FILE-LINK-Kommando oder in einer vorangegangenen ASSIGN-FILES-Anweisung zugewiesen wurden, werden ignoriert. Eine Veränderung der Dateiattribute gegenüber den aus der Eingabedatei übernommenen oder von SORT festgelegten kann nur durch Zuordnung über ADD-FILE-LINK-Kommando erfolgen.

Der Dateikettungsname SORTOUT wird verwendet. Dieser bleibt nach dem Ende des Sortierlaufs erhalten.

Ein POSIX-Dateiname muss zur Unterscheidung von einem BS2000-Dateinamen in Hochkommata angegeben werden.

WORK-FILES =

Zuordnung der Arbeitsdatei(en) (max. 9).

WORK-FILES = *STD

Die Arbeitsdateien sind über ADD-FILE-LINK-Kommando zugewiesen (Dateikettungsname SORTWK oder SORTWKx) oder werden von SORT angelegt (Dateiname: SORTWORK.tsn.jjmmmtt.hhmmss).

WORK-FILES = list-poss(9): <filename 1..54>

Name(n) der Arbeitsdatei(en). Über ADD-FILE-LINK-Kommandos oder eine zuvor angegebene ASSIGN-FILES-Anweisung zugewiesene Arbeitsdateien werden ignoriert. Die Dateien und Dateikettungsnamen bleiben nach dem Ende des Sortierlaufes erhalten.

AUXILIARY-FILES =

Zuordnung der Hilfsdatei(en) (max. 99).

AUXILIARY-FILES = *STD

Die Hilfsdateien sind über ADD-FILE-LINK-Kommandos zugewiesen (Dateikettungsname SORTWKxx) oder werden von SORT angelegt (Dateiname: SORTWKxx.tsn.jjmmmtt.hhmmss).

AUXILIARY-FILES = list-poss(99): <filename 1..54>

Name(n) der Hilfsdatei(en). Über ADD-FILE-LINK-Kommandos oder eine zuvor angegebene ASSIGN-FILES-Anweisung zugewiesene Hilfsdateien werden ignoriert. Die Dateien und Dateikettungsnamen bleiben nach dem Ende des Sortierlaufes erhalten.

CHECKPOINT-FILE =

Zuordnung einer Fixpunktdatei.

CHECKPOINT-FILE = *STD

Die Fixpunktdatei wird über ADD-FILE-LINK-Kommando zugewiesen (Dateikettungsname SORTCKPT) oder wird von SORT angelegt (Dateiname: SORTCKPT.tsn.jjmmmtt.hhmmss).

CHECKPOINT-FILE = <filename 1..54>

Name der Fixpunktdatei. Eine über ADD-FILE-LINK-Kommando oder eine zuvor angegebene ASSIGN-FILES-Anweisung zugewiesene Fixpunktdatei wird ignoriert. Die Datei und der Dateikettungsname bleiben nach dem Ende des Sortierlaufes erhalten.

MODULE-LIBRARY =

Zuordnung einer Bindemodul-Bibliothek mit Benutzermodulen. SORT lädt Benutzermodule aus dieser Bibliothek nach.

MODULE-LIBRARY = *STD

Die Bindemodul-Bibliothek ist über ADD-FILE-LINK-Kommando zugewiesen (Dateikettungsname SORTMODS). Der Dateikettungsname bleibt nach dem Ende des Sortierlaufes erhalten.

MODULE-LIBRARY = <filename 1..54>

Name der Bindemodul-Bibliothek. Eine über ADD-FILE-LINK-Kommando oder in einer zuvor angegebenen ASSIGN-FILES-Anweisung zugewiesene Bibliothek wird ignoriert.

STATEMENT-FILES =

Zuweisung einer oder mehrerer Dateien (max. 10) mit SORT-Anweisungen (siehe [Abschnitt „Anweisungsdateien“ auf Seite 116](#)).

STATEMENT-FILES = *NONE

Keine Angabe von Anweisungsdateien.

STATEMENT-FILES = list-poss(10): <filename 1..54>

Name(n) der Anweisungsdatei(en).

CODE =

Angabe des Codes von POSIX-Dateien. Er gilt sowohl für die Eingabedatei als auch für die Ausgabedatei und legt lediglich die Verschlüsselung des Satzende-Kennzeichens fest. Dieser Operand hat insbesondere **keinen** Einfluss auf die Sortierreihenfolge. Für BS2000-Dateien hat der Operand keine Bedeutung und wird ignoriert.

CODE = *EBCDIC

Das Satzende-Kennzeichen wird nach EBCDIC verschlüsselt und hat den Wert X'15'.

CODE = *ASCII

Das Satzende-Kennzeichen wird nach ASCII verschlüsselt und hat den Wert X'0A'.

ASSIGN-RESOURCES

Funktion: Vereinbaren der Größe des von SORT zu benutzenden Haupt- und Plattenspeichers, sowie die Anzahl der Bandgeräte für Hilfsdateien. Die Angaben bei ASSIGN-RESOURCES haben Vorrang vor den Werten, die SORT ermittelt.

ASSIGN-RESOURCES

MEMORY-SIZE = *STD / *MIN / *SMALL / *LARGE / *MAX / <integer 1..500000>

,**DISK-SPACE** = *BY-CALCULATION / <integer 1..2147483647>

,**TAPE-UNITS** = *NONE / <integer 1..99>

MEMORY-SIZE =

Gibt die Größe des virtuellen Speichers für die Vorsortierungsbereiche und die internen Eingabe-/Ausgabepuffer an. Wird dieser Operand nicht angegeben, berechnet SORT einen Wert aus sonstigen direkten oder indirekten Angaben zum Sortier-/Mischlauf. Sind keine Angaben für eine Abschätzung vorhanden, wird der Wert verwendet, der mit dem Operanden CORE-DEFAULT der Anweisung MODIFY-SORT-DEFAULTS als Standardwert festlegbar ist. Ober- und Untergrenze können ebenfalls mit der Anweisung MODIFY-SORT-DEFAULTS festgelegt werden.

MEMORY-SIZE = *STD

Von SORT aus den Angaben zum Sortier-/Mischlauf berechneter Wert. Kann SORT diesen Wert nicht berechnen, da die Anhaltspunkte dafür fehlen, wird der voreingestellte Standardwert angenommen. Dieser Standardwert kann mit dem Operanden CORE-DEFAULT der Anweisung MODIFY-SORT-DEFAULTS kennungs- oder anlagenspezifisch festgelegt werden (Voreinstellung bei der Auslieferung von SORT = 40).

MEMORY-SIZE = *MIN

Minimalwert. Entspricht dem Wert, der mit dem Operanden CORE-MINIMUM der Anweisung MODIFY-SORT-DEFAULTS kennungs- oder anlagenspezifisch festgelegt werden kann (Voreinstellung bei der Auslieferung von SORT = 24). Kann SORT mit diesem CORE-Wert nicht arbeiten (z.B. Satzlänge = 32759), wird der Wert von SORT vergrößert.

MEMORY-SIZE = *SMALL

Unterer Wert, entspricht dem größeren der folgenden Werte:

- von SORT berechneter Wert minus 33% oder
- Angabe im Operanden CORE-MINIMUM der Anweisung MODIFY-SORT-DEFAULTS.

MEMORY-SIZE = *LARGE

Oberer Wert, entspricht dem kleineren der folgenden Werte:

- von SORT berechneter Wert plus 33% oder
- Angabe im Operanden CORE-MAXIMUM der Anweisung MODIFY-SORT-DEFAULTS.

MEMORY-SIZE = *MAX

Maximalwert. Wert, der mit dem Operanden CORE-LIMIT der Anweisung MODIFY-SORT-DEFAULTS kennungs- oder anlagenspezifisch festgelegt werden kann (Voreinstellung bei der Auslieferung von SORT = 96).

MEMORY-SIZE = <integer 1..500000>

Größe des gewünschten Speicherbereiches in CORE-Seiten (4K). Wird ein Wert größer als 32767 angegeben, so wird dieser Wert in Megabyte umgewandelt (Division durch 256), sodass der tatsächliche CORE-Wert um bis zu 255 CORE-Seiten kleiner sein kann als der angegebene Wert.

DISK-SPACE =

Größe einer Plattenarbeitsdatei, die SORT anlegen soll, in PAM-Seiten.

DISK-SPACE = *BY-CALCULATION

Die Größe der Arbeitsdatei wird von SORT aus den Angaben für ESTIMATED-RECORDS und RECORDS-PER-CYCLE (bei der SORT-RECORDS-Anweisung) errechnet. Ist dies nicht möglich, verwendet SORT den Wert für MEMORY-SIZE multipliziert mit 16.

DISK-SPACE = <integer 1..2147483647>

Größe der Arbeitsdatei in PAM-Seiten. Hat der Benutzer eine oder mehrere Arbeitsdateien selbst eingerichtet, so wird dieser Wert ignoriert.

TAPE-UNITS =

Angabe von Hilfsbandgeräten, die für die Zyklusausgabe beim Zyklensortieren verwendet werden sollen. Ergeben sich auf Grund der Angaben für ESTIMATED-RECORDS und RECORDS-PER-CYCLE (bei der SORT-RECORDS-Anweisung) mehr Zyklen als Hilfsdateien über Dateikettungsamen zugeordnet sind, richtet SORT weitere Hilfsdateien ein. Die Anzahl der Bandhilfsdateien wird dann maximal bis auf den angegebenen Wert ergänzt und der Rest als Plattenhilfsdatei eingerichtet.

TAPE-UNITS = *NONE

Es werden keine Bandgeräte für Hilfsdateien zur Verfügung gestellt.

TAPE-UNITS = <integer 1..99>

Anzahl der Bandgeräte für Hilfsdateien.

END

Funktion: Abschluss der Eingabe von Anweisungen für SORT und Starten des Sortier-/Mischlaufs. Die END-Anweisung muss immer als letzte Anweisung gegeben werden und darf nur einmal vorkommen.

END

Diese Anweisung hat keine Operanden.

MERGE-RECORDS

Funktion: Mit MERGE-RECORDS legt der Benutzer Sortierfelder und deren wesentliche Kenndaten (Position, Länge,...) für einen Mischlauf fest. Ein Mischlauf kann nur erfolgen, wenn alle Eingabedateien bereits nach den angegebenen Sortierbegriffen sortiert sind.

Ein Mischlauf mit MERGE-RECORDS sollte immer dann erfolgen, wenn Dateien bereits nach einem gemeinsamen Kriterium sortiert worden sind, da die rechenzeitintensive Sortierphase wegfällt.

MERGE-RECORDS

```

FIELDS = *COMPLETE-RECORD / list-poss(64): *FIELD-EXPLICIT(...) / *FIELD-SYMBOLIC(...)

*FIELD-EXPLICIT(...)
  |
  | POSITION = <integer 1..32759>(…)
  |   <integer 1..32759>(…)
  |   | BIT-POSITION = 0 / <integer 0..7>
  |   |
  |   | LENGTH = <integer 0..32759>(…)
  |   |   <integer 0..32759>(…)
  |   |   | NUMBER-OF-BITS = 0 / <integer 0..7>
  |   |
  |   | FORMAT = *CHARACTER / *BINARY / *FIXED-POINT / *FLOATING-POINT /
  |   |   *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN /
  |   |   *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE /
  |   |   *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER /
  |   |   *TRANSLATE-CHARACTER / *UNICODE-CHARACTER
  |   |
  |   | SORTING-ORDER = *ASCENDING / *DESCENDING / *EXTERNAL-COMPARE
  |   |
  |   | PRIORITY = *CURRENT-NUMBER / <integer 1..64>
  |   |
  |   | ELIMINATE = *NO / *YES
  |   |
  |   | TWO-DIGIT-YEAR = *NO / *YES
  |
  |
  | *FIELD-SYMBOLIC(…)
  |   | SORT-FIELD-NAME = list-poss(64): <name 1..20>
  |   |
  |   | SORTING-ORDER = *ASCENDING / *DESCENDING / *EXTERNAL-COMPARE
  |   |
  |   | PRIORITY = *CURRENT-NUMBER / <integer 1..64>
  |   |
  |   | ELIMINATE = *NO / *YES
  |   |
  |   | TWO-DIGIT-YEAR = *NO / *YES
  |
  |
  | RECORDS-PER-CYCLE = *NO / <integer 1..2147483639>
  |
  | CHECKPOINT = *NO / *YES

```

FIELDS =

Angabe der Sortierfelder, die die Mischreihenfolge festlegen. Es muss mindestens 1, maximal dürfen 64 Sortierfelder angegeben werden.

FIELDS = *COMPLETE-RECORD

Bei fixem Satzformat wird der ganze Satz als Sortierfeld verwendet. Bei variablem Satzformat wird das Satzlängengeld nicht in den Satzvergleich einbezogen. Der variable Teil des Satzes wird für den Satzvergleich bis zur maximalen Satzlänge mit dem Füllzeichen des FILLER-Operanden der SET-RECORD-ATTRIBUTES-Anweisung (Standardwert X'00') aufgefüllt.

FIELDS = *FIELD-EXPLICIT(...)

Angabe von Sortierfeldern durch Position, Länge und Format.

POSITION = <integer 1..32759(...)>

Position des Sortierfeldes relativ zum Satzanfang.

Bei Sätzen mit festem Satzformat hat das erste Datenfeld die Position 1. Bei Sätzen mit variablem Satzformat hängt die Position des ersten Datenfelds von der Dateart und dem Operanden IGNORE-LENGTH-FIELD der Anweisung SET-SORT-OPTIONS ab. Diese Abhängigkeit wird in folgender Tabelle dargestellt:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Bei allen Formaten außer CHARACTER, EBCDIC-DIN, EBCDIC-INTERNATIONAL, EXTENDED-CHARACTER und TRANSLATE-CHARACTER müssen die Positionsangaben für Sortierfelder im Bereich 1 bis 4096 liegen.

BIT-POSITION = 0 / <integer 0..7>

Positionsangabe in Bit zusätzlich zur Positionsangabe in Byte. Die Angabe von BIT-POSITION ist nur für Sortierfelder mit Format BINARY erlaubt.

LENGTH = <integer 0..32759(...)>

Länge des Sortierfeldes. Die Länge muss innerhalb der für das Format erlaubten Grenzen liegen. Durch die Angabe von BIT-POSITION und NUMBER-OF-BITS vergrößert sich die Länge des Sortierfeldes um 1 Byte ($\text{BIT-POSITION} + \text{NUMBER-OF-BITS} \leq 7$) oder um 2 Byte ($\text{BIT-POSITION} + \text{NUMBER-OF-BITS} > 7$). Dies ist bei der maximalen Länge zu berücksichtigen. Bei Sätzen variabler Länge dürfen sich Sortierfelder vom Format CHARACTER, EBCDIC-DIN und EBCDIC-INTERNATIONAL auch in den variablen Teil des Satzes hinein fortsetzen.

NUMBER-OF-BITS = 0 / <integer 0..7>

Längenangabe in Bits zusätzlich zur Längenangabe in Byte. Die Angabe von NUMBER-OF-BITS ist nur für Sortierfelder mit Format BINARY erlaubt.

FORMAT = *CHARACTER / *BINARY / *FIXED-POINT / *FLOATING-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN / *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE / *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER / *TRANSLATE-CHARACTER / *UNICODE-CHARACTER

Format des Sortierfeldes (Eigenschaften siehe [Seite 38](#)).

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

SORTING-ORDER = *EXTERNAL-COMPARE

Die Reihenfolge für dieses Sortierfeld wird vom Benutzer über den Benutzerausgang EXTERNAL-COMPARE festgelegt. Dieses Sortierfeld darf max. 255 Byte lang sein.

PRIORITY =

Wertigkeit des Sortierfeldes. Sind mehrere Sortierfelder angegeben, werden zuerst die Sortierfelder mit der Wertigkeit 1 verglichen. Sind sie gleich, werden die Sortierfelder mit der Wertigkeit 2 verglichen. Dieser Vorgang wird solange fortgesetzt, bis die Sortierfelder ungleich sind oder die Sortierfelder mit der höchsten Wertigkeit verglichen wurden.

PRIORITY = *CURRENT-NUMBER

Die Wertigkeit wird durch die Reihenfolge, in der die Sortierfelder angegeben werden, bestimmt (das erste Sortierfeld hat die Wertigkeit 1).

PRIORITY = <integer 1..64>

Angabe der Ordnungsziffer. Wird die Ordnungsziffer bei einem Sortierfeld angegeben, so muss sie auch bei allen anderen Sortierfeldern angegeben werden. Alle Ordnungsziffern müssen eine mit 1 beginnende, lückenlos aufsteigende Reihenfolge ergeben.

ELIMINATE =

Angabe, ob das Sortierfeld eliminiert (nicht in die Ausgabe übernommen) wird.

ELIMINATE = *NO

Das Sortierfeld wird nicht eliminiert.

ELIMINATE = *YES

Das Sortierfeld wird eliminiert. Es dient nur zur Feststellung der Sortierreihenfolge. Für binäre Felder mit Bit-Angaben ist diese Angabe nicht zulässig. SORT gibt in diesem Fall eine Warnung aus und ignoriert die Angabe. Beim Benutzerausgang OUTPUT (mit PARAMETER-MODE = 24) werden gleiche Sätze nicht angezeigt.

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT der Anweisung SET-SORT-OPTIONS ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jiv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fvj'

Dabei sind: j = eine Stelle der zweistelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

FIELDS = *FIELD-SYMBOLIC(...)

Angabe von Sortierfeldern durch symbolische Namen, die vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Felder vereinbart werden müssen.

SORT-FIELD-NAME = list-poss(64): <name 1..20>

Name eines Sortierfeldes (max 64 Felder). Wird eine Liste mit symbolischen Namen angegeben, so gelten die bei SORTING-ORDER und ELIMINATE gemachten Angaben für alle Namen, der Wert für PRIORITY wird für den ersten angegebenen Namen übernommen, für die restlichen Namen wird ein jeweils um 1 erhöhter Wert eingetragen.

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

SORTING-ORDER = *EXTERNAL-COMPARE

Die Reihenfolge für dieses Sortierfeld wird vom Benutzer über den Benutzerausgang EXTERNAL-COMPARE festgelegt. Dieses Sortierfeld darf max. 255 Byte lang sein.

PRIORITY =

Wertigkeit des Sortierfeldes. Sind mehrere Sortierfelder angegeben, werden zuerst die Sortierfelder mit der Wertigkeit 1 verglichen. Sind sie gleich, werden die Sortierfelder mit der Wertigkeit 2 verglichen. Dieser Vorgang wird solange fortgesetzt, bis die Sortierfelder ungleich sind oder die Sortierfelder mit der höchsten Wertigkeit verglichen wurden.

PRIORITY = *CURRENT-NUMBER

Die Wertigkeit wird durch die Reihenfolge, in der die Sortierfelder angegeben werden, bestimmt (das erste Sortierfeld hat die Wertigkeit 1).

PRIORITY = <integer 1..64>

Angabe der Ordnungsziffer. Wird die Ordnungsziffer bei einem Sortierfeld angegeben, so muss sie auch bei allen anderen Sortierfeldern angegeben werden. Alle Ordnungsziffern müssen eine mit 1 beginnende, lückenlos aufsteigende Reihenfolge ergeben.

ELIMINATE =

Angabe, ob das Sortierfeld eliminiert (nicht in die Ausgabe übernommen) wird.

ELIMINATE = *NO

Das Sortierfeld wird nicht eliminiert.

ELIMINATE = *YES

Das Sortierfeld wird eliminiert. Es dient nur zur Feststellung der Sortierreihenfolge. Für binäre Felder mit Bit-Angaben ist diese Angabe nicht zulässig. SORT gibt in diesem Fall eine Warnung aus und ignoriert die Angabe. Beim Benutzerausgang OUTPUT (mit PARAMETER-MODE=24) werden gleiche Sätze nicht angezeigt.

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT der Anweisung SET-SORT-OPTIONS ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fvj'

Dabei sind: j = eine Stelle der zweistelligen Jahrestelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

RECORDS-PER-CYCLE =

Anzahl der Sätze pro Fixpunktzyklus. Der angegebene Wert bestimmt, nach wie vielen Sätzen SORT jeweils einen Fixpunkt ausgeben soll. Diese Angabe ist nur sinnvoll, wenn gleichzeitig CHECKPOINT=*YES angegeben wird.

RECORDS-PER-CYCLE = *NO

Es werden keine Fixpunkte ausgegeben.

RECORDS-PER-CYCLE = <integer 1..2147483639>

Anzahl der Sätze, nach der ein Fixpunkt ausgegeben wird.

CHECKPOINT =

Steuerung der Ausgabe von Fixpunkten.

CHECKPOINT = *NO

SORT schreibt keine Fixpunkte.

CHECKPOINT = *YES

SORT schreibt nach der im RECORDS-PER-CYCLE-Operanden angegebenen Satzanzahl einen Fixpunkt.

MODIFY-CODE

Funktion: Ändern der durch den EBCDIC-Code festgelegten Sortierreihenfolge in eine vom Benutzer festgelegte Reihenfolge. Diese Anweisung muss angegeben werden, wenn ein Feld mit FORMAT=*MODIFY-CODE vereinbart wurde.

MODIFY-CODE
SEQUENCES = list-poss(256): <c-string 2..2 with-low> / <x-string 3..4>

SEQUENCES =

Gibt alle Zeichen an, die in einer anderen Reihenfolge sortiert werden sollen, als sie durch den EBCDIC-Code vorgesehen ist. Über eine Liste von Zeichenpaaren, in der Form 'zq' (Zeichen) oder X'zq' (Sedezimal) wird veranlasst, dass jeweils das Zeichen 'q' in der Sortierreihenfolge unmittelbar hinter das Zeichen 'z' verschoben wird. Maximal können 256 Zeichenpaare angegeben werden.

SEQUENCES = list-poss(256): <c-string 2..2 with-low>

Angabe eines Zeichenpaares in Character-Darstellung. Beispiel: MODIFY-CODE SEQUENCES = ('aA','bB'). Diese Angabe bewirkt, dass in der Sortierreihenfolge 'A' unmittelbar hinter 'a' und 'B' unmittelbar hinter 'b' eingeordnet wird.

SEQUENCES = list-poss(256): <x-string 3..4>

Angabe eines Zeichenpaares in Sedezimal-Darstellung. Beispiel: MODIFY-CODE SEQUENCES = (X'81C1'). Diese Angabe bewirkt, dass in der Sortierreihenfolge X'C1' (= 'A') unmittelbar hinter X'81' (= 'a') eingeordnet wird.

Soll ein Zeichen in der Sortierreihenfolge hinter ein Zeichen gebracht werden, das bereits umcodiert wurde, so gilt nicht dessen ursprüngliche EBCDIC-Position, sondern die neue.

Soll ein Zeichen (z.B. X'F8') an den Beginn der Sortierreihenfolge gebracht werden, so sind dafür 2 Angaben notwendig:

MODIFY-CODE SEQUENCES = (X'00F8',X'F800')

Mit dem ersten Zeichenpaar wird X'F8' unmittelbar hinter X'00' gebracht, mit dem zweiten Zeichenpaar wird X'00' unmittelbar hinter X'F8' gebracht, sodass X'F8' am Beginn der Sortierfolge steht.

MODIFY-SORT-DEFAULTS

Funktion: Festlegen bzw. Ändern von Voreinstellwerten für SORT-Parameter.
Die SORT-Parameter können anlagen- oder kennungsspezifisch voreingestellt werden.
Die Gültigkeitsdauer der Voreinstellwerte kann auf den aktuellen SORT-Lauf beschränkt werden oder auf alle folgenden SORT-Läufe ausgedehnt werden.

MODIFY-SORT-DEFAULTS

```

MIN-MSG-WEIGHT = *UNCHANGED / *STD / *ALL / *NORMAL / *CRITICAL / *NONE
, CORE-MINIMUM = *UNCHANGED / *STD / <integer 24..32767>
, CORE-MAXIMUM = *UNCHANGED / *STD / <integer 24..32767>
, CORE-LIMIT = *UNCHANGED / *STD / <integer 24..32767>
, CORE-DEFAULT = *UNCHANGED / *STD / <integer 24..32767>
, SUBTASK-JOB-CLASS = *UNCHANGED / *STD / <name 1..8>
, SAVE-DEFAULTS = *NO / *YES

```

MIN-MSG-WEIGHT =

Voreinstellung für das minimale auszugebende Meldungsgewicht.
Angabe der Priorität, ab der Meldungen ausgegeben werden sollen. Die Meldung wird dann ausgegeben, wenn ihre Priorität größer oder gleich der angegebenen ist.

MIN-MSG-WEIGHT = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

MIN-MSG-WEIGHT = *STD

Der Standardwert wird verwendet (*NORMAL).

MIN-MSG-WEIGHT = *NORMAL

Bei autonomen Sortier-/Mischläufen werden Meldungen ab Priorität 2 ausgegeben. Bei SORT als Unterprogramm werden Meldungen ab Priorität 3 ausgegeben.

MIN-MSG-WEIGHT = *ALL

Alle Meldungen werden ausgegeben (ab Priorität 0).

MIN-MSG-WEIGHT = *CRITICAL

Meldungen ab Priorität 3 werden ausgegeben.

MIN-MSG-WEIGHT = *NONE

Meldungen ab Priorität 7 werden ausgegeben (nur Meldungen über interne Fehler).

Hinweis

Existiert in dem Sortierlauf eine SET-SORT-OPTIONS-Anweisung, so hat die Angabe bzw. der Standardwert für MIN-MSG-WEIGHT in der SET-SORT-OPTIONS-Anweisung Vorrang.

CORE-MINIMUM =

Untergrenze für die CORE-Wert-Berechnung durch SORT.

CORE-MINIMUM = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

CORE-MINIMUM = *STD

Der Standard-Wert wird verwendet (24).

CORE-MINIMUM = <integer 24...32767>

Der angegebene Wert wird verwendet.

CORE-MAXIMUM =

Maximalwert für den intensiv genutzten virtuellen Speicher. Wird in der ASSIGN-RESOURCES-Anweisung ein Wert größer als 400 angegeben, so wird nur ein Sechzehntel dieses Speichers intensiv genutzt. Wenn der intensiv genutzte Speicher den hier eingestellten Wert überschreitet, wird der in der ASSIGN-RESOURCES-Anweisung angegebene Wert reduziert.

CORE-MAXIMUM = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

CORE-MAXIMUM = *STD

Der Standard-Wert wird verwendet (4096).

CORE-MAXIMUM = <integer 24...32767>

Der angegebene Wert wird verwendet.

Hinweis

Falls der angegebene Wert den mit dem Operanden CORE-MINIMUM festgelegten Minimalwert unterschreitet, ist der Maximalwert gleich dem Minimalwert.

CORE-LIMIT =

Obergrenze für die CORE-Wert-Berechnung durch SORT.

CORE-LIMIT = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

CORE-LIMIT = *STD

Der Standard-Wert wird verwendet (96).

CORE-LIMIT = <integer 24...32767>

Der angegebene Wert wird verwendet.

Hinweis

Der angegebene Wert darf den mit dem Operanden CORE-MINIMUM festgelegten Wert nicht unterschreiten und den mit dem Operanden CORE-MAXIMUM festgelegten Wert nicht überschreiten. Trifft einer dieser Fälle zu, wird als Grenzwert der im entsprechenden Operanden festgelegte Wert verwendet.

CORE-DEFAULT =

CORE-Wert beim Fehlen jeglicher Angaben.

CORE-DEFAULT = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

CORE-DEFAULT = *STD

Der Standard-Wert wird verwendet (40).

CORE-DEFAULT = <integer 24...32767>

Der angegebene Wert wird verwendet.

Hinweis

Der angegebene Wert darf den mit dem Operanden CORE-MINIMUM festgelegten Wert nicht unterschreiten und den mit dem Operanden CORE-MAXIMUM festgelegten Wert nicht überschreiten. Trifft einer dieser Fälle zu, wird als Grenzwert der im entsprechenden Operanden festgelegte Wert verwendet.

SUBTASK-JOB-CLASS =

Jobklasse, die für den Start einer Nebentask verwendet werden soll.

SUBTASK-JOB-CLASS = *UNCHANGED

Der Wert bleibt unverändert (siehe Hinweis auf [Seite 157](#)).

SUBTASK-JOB-CLASS = *STD

Der Standard-Wert soll verwendet werden (*STD).

SUBTASK-JOB-CLASS = <name 1...8>

Die angegebene Jobklasse soll verwendet werden.

SAVE-DEFAULTS =

Festlegung der Gültigkeitsdauer für die veränderten Voreinstellwerte.

SAVE-DEFAULTS = *NO

Änderung nur für diesen SORT-Lauf.

SAVE-DEFAULTS = *YES

Änderung für alle weiteren SORT-Läufe.

Das Ergebnis wird in der Parameterdatei abgelegt.

Falls die Benutzerkennung, unter der das Kommando eingegeben wird, mit derjenigen übereinstimmt, unter der die zentrale Parameterdatei installiert ist, gelten die neuen Voreinstellungen anlagenweit. Andernfalls gelten sie kennungsspezifisch.

Hinweis

Der Operandenwert *UNCHANGED wird folgendermaßen ausgewertet:

Falls in einer vorangegangenen MODIFY-SORT-DEFAULTS-Anweisung desselben SORT-Laufs ein Wert ungleich *UNCHANGED angegeben wurde, ist dieser gültig. Andernfalls wird der Wert der kennungsspezifischen Parameterdatei oder, wenn diese noch nicht existiert, der zentralen Parameterdatei entnommen.

Die kennungsspezifische Parameterdatei wird bei der Ausführung der ersten MODIFY-SORT-DEFAULTS-Anweisung mit SAVE-DEFAULTS=*YES in einer Kennung angelegt, und bei jeder weiteren Ausführung dieser Anweisung geändert.

Die zentrale Parameterdatei wird bei der Installation des Produktes SORT angelegt.

Die anlagenspezifische Änderung der Voreinstellung ist nur von der Kennung aus möglich, auf der SORT installiert ist, da dort auch die zentrale Parameterdatei abgelegt ist.

Über die aktuell gültigen Voreinstellungen können Sie sich mit der Anweisung SHOW-SORT-DEFAULTS informieren.

SELECT-INPUT-RECORDS

Funktion: Mit SELECT-INPUT-RECORDS wählt der Benutzer die Sätze aus den Eingabedateien aus, die sortiert werden sollen. Die Auswahl erfolgt über einen logischen Ausdruck (logisch kombinierte Vergleiche).

SELECT-INPUT-RECORDS
CONDITION = <text 0..1800 with-low>

CONDITION = <text 0..1800 with-low>

Im CONDITION-Operanden muss die Bedingung angegeben werden, unter der ein Eingabesatz in die Sortierung aufgenommen wird. Die Bedingung kann aus einer oder mehreren durch AND oder OR logisch verknüpften Vergleichsrelationen bestehen, die achtfach über Klammern verschachtelt werden dürfen. Es dürfen maximal 64 Relationen angegeben werden.

Angabe von Vergleichsrelationen in folgender Form:

$$[\text{NOT}] (\text{rel1}, \left\{ \begin{array}{c} \text{AND} \\ \text{OR} \end{array} \right\}, \text{rel2} \dots)$$

Durch die Angabe von **NOT** werden alle Sätze, die die Bedingung **nicht** erfüllen, in die Sortierung aufgenommen.

Die **Vergleichsrelationen** rel1, rel2, ... haben folgenden formalen Aufbau:

$$\text{rel} = \text{feld1}, \text{rop}, \left\{ \begin{array}{c} \text{feld2} \\ \text{konstante} \end{array} \right\}$$

Die **Vergleichsfelder** feld1, feld2 müssen in folgender Form angegeben werden:

$$\text{feld} = \left\{ \begin{array}{c} (\text{beginn}, \text{länge}, \text{format}) \\ \text{symbolischer Name} \end{array} \right\}$$

Beginn:

Hier muss der Anfang des Vergleichsfeldes angegeben werden. Die Beginnangaben können auch bei binärem Datenformat nur in Byte gemacht werden. Bei fixem Satzformat hat das erste Datenbyte die Position 1, bei Sätzen mit variablem Satzformat hängt diese Position von der Dateart und dem Operanden IGNORE-LENGTH-FIELD der Anweisung SET-SORT-OPTIONS ab. Die Position des ersten Datenfelds in diesen Fällen können Sie folgender Tabelle entnehmen:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Länge:

Hier muss die Länge des Vergleichsfeldes in Byte angegeben werden. Die jeweiligen Maximallängen sind vom Datenformat abhängig. Siehe nachfolgende Tabelle.

Format:

Für Vergleichsfelder sind folgende Formate und Längen zugelassen:

Format	Formatbezeichnung	Länge in Byte
BI	binär	1 - 256
CH	Zeichen	1 - 256
FI	Festpunkt	1 - 256
PD	dezimal gepackt	1 - 16
ZD	dezimal entpackt	1 - 16

Symbolischer Name:

Dieser Name muss zuvor mit der ADD-SYMBOLIC-NAMES-Anweisung bezüglich Position, Länge und Format als Feld vereinbart worden sein. Wurde dieser Name noch nicht vereinbart, gibt SORT eine Fehlermeldung aus. Ein symbolischer Name darf maximal 20 Zeichen lang sein.

Konstanten, die an Stelle des zweiten Vergleichsfeldes `feld2` angegeben werden können, haben folgende Form:

$$\text{konstante} = \left\{ \begin{array}{ll} \text{Dezimalzahl} & \left. \begin{array}{l} +nn\dots n \\ -nn\dots n \end{array} \right\} \\ \text{Sedezimalzahlstring} & \text{X'ss\dots s'} \\ \text{Zeichenstring} & \text{C'zz\dots z'} \\ \text{Symbolischer Name} & \text{name} \end{array} \right\}$$

Die Länge aller Konstanten darf 4000 Byte nicht überschreiten.

Dezimalzahlen müssen mit einem Vorzeichen angegeben werden. Hat das erste Vergleichsfeld das Format `FIXED-POINT`, so darf die Konstante die Werte `+2147483647` nicht überschreiten bzw. `-2147483648` nicht unterschreiten.

Bei **Sedezimal-** und **Zeichenkonstanten** ist die Maximallänge 256 Byte.

Symbolischer Name:

Dieser Name muss zuvor mit der `ADD-SYMBOLIC-NAMES`-Anweisung als Konstante vereinbart worden sein. Wurde dieser Name noch nicht vereinbart, gibt `SORT` eine Fehlermeldung aus. Ein symbolischer Name darf maximal 20 Zeichen lang sein. Bezüglich Maximallänge, Anpassung von Konstanten und Formatverträglichkeit gilt das Gleiche wie bei direkter Angabe des Konstantenwertes.

Vergleichsoperator rop:

Folgende Vergleichsoperatoren sind zulässig:

Vergleichsoperator	Ersatzzeichen	Bedeutung	
EQ	=	gleich	(equal)
LT	<	kleiner	(less than)
GT	>	größer	(greater than)
NE	<>	ungleich	(not equal)
GE	>=	größer gleich	(greater or equal)
LE	<=	kleiner gleich	(less or equal)

Die Vergleichsrelationen können durch `AND` oder `OR` miteinander verknüpft werden, wobei der normale Vorrang `AND` vor `OR` durch Klammern verändert werden kann.

Zulässige Kombinationen von Vergleichsfeldern und -konstanten:

Das Format des ersten (oder einzigen) Vergleichsfeldes muss mit dem des zweiten bzw. mit der Art der Vergleichskonstanten verträglich sein. Die erlaubten Kombinationen sind in folgender Tabelle beschrieben.

1. Vergleichsfeld	2. Vergleichsfeld					Vergleichskonstante		
	BI	CH	FI	PD	ZD	Dezimalzahl	Sedezimalstring	Zeichenstring
BI	+	+	-	-	-	-	+	+
CH	+	+	-	-	-	-	+	+
FI	-	-	+	-	-	+	-	-
PD	-	-	-	+	+	+	-	-
ZD	-	-	-	+	+	+	-	-

Erläuterung:

- + Die Formatkombination ist erlaubt.
- Die Formatkombination ist nicht erlaubt. SORT meldet einen Fehler.

Beim Vergleich unterschiedlich langer Vergleichsfelder passt SORT das kürzere Vergleichsfeld an das längere an. Folgende Tabelle gibt eine Übersicht über die Art der Auffüllung des kürzeren Feldes:

Format	Ausrichtung	Auffüllung	Füllzeichen
BI	linksbündig	nach rechts	Null
CH	linksbündig	nach rechts	X'40'(Leerzeichen)
FI	rechtsbündig	nach links	Vorzeichen
PD	rechtsbündig	nach links	Null
ZD	rechtsbündig	nach links	X'F0'(Null)

Beim Vergleich eines Vergleichsfeldes mit einer Konstanten wird die Konstante an die Länge des Vergleichsfeldes angepasst. Numerische Konstanten werden linksbündig aufgefüllt oder abgeschnitten, Zeichenkonstanten werden rechtsbündig aufgefüllt oder abgeschnitten. Wird bei dieser Anpassung ein signifikanter Teil der Vergleichskonstanten abgeschnitten, d.h. ein Teil, der ungleich der entsprechenden Auffüllung ist, dann gibt SORT eine Warnungsmeldung aus.

SET-RECORD-ATTRIBUTES

Funktion: Mit SET-RECORD-ATTRIBUTES legt der Benutzer das Format für Ein- und Ausgabesätze sowie für interne Sätze fest. Diese Anweisung muss angegeben werden, wenn SORT diese Angaben nicht über andere Anweisungen erhält oder errechnen kann (z.B. keine Eingabedatei, wenn die Satzeingabe nur über den Benutzeranschluss INPUT erfolgt) oder Satzlängen bzw. -formate verändert werden sollen.

SET-RECORD-ATTRIBUTES

INPUT = *STD / *VARIABLE(...) / *FIXED(...)

*VARIABLE(...)

 | **MAXIMUM-RECORD-SIZE** = *STD / <integer 1..32759>

*FIXED(...)

 | **RECORD-SIZE** = *STD / <integer 1..32759>

,**INTERNAL** = *BY-CALCULATION / *VARIABLE(...) / *FIXED(...)

*VARIABLE(...)

 | **MAXIMUM-RECORD-SIZE** = *BY-CALCULATION / <integer 1..32759>

 | **MINIMUM-RECORD-SIZE** = *BY-CALCULATION / <integer 1..32759>

 | **AVERAGE-RECORD-SIZE** = *BY-CALCULATION / <integer 1..32759>

*FIXED(...)

 | **RECORD-SIZE** = *BY-CALCULATION / <integer 1..32759>

,**OUTPUT** = *STD / *VARIABLE(...) / *FIXED(...)

*VARIABLE(...)

 | **MAXIMUM-RECORD-SIZE** = *STD / <integer 1..32759>

*FIXED(...)

 | **RECORD-SIZE** = *STD / <integer 1..32759>

,**FILLER** = *NIL / <c-string 1..1 with-low> / <x-string 1..2>

,**CODED-CHARACTER-SET** = *NOT-SPECIFIED / <name 1..8>

,**UNICODE-FILLER** = *BLANK / <x-string 4..4>

INPUT =

Beschreibung der Eingabesätze bezüglich Satzformat und Satzlänge.

INPUT = *STD

SORT versucht das Satzformat und die Satzlänge aus einem vorher angegebenen ADD-FILE-LINK-Kommando oder aus dem Katalogeintrag zu erhalten. Kann das Satzformat nicht festgestellt werden, wird das Format des internen Satzes übernommen. Wenn das nicht möglich ist, wird festes Satzformat angenommen. Kann die Satzlänge nicht festgestellt werden, wird als Ersatzwert die auf Grund der BUFFER-LENGTH-Angabe maximal mögliche Satzlänge übernommen. Wenn das nicht möglich ist, wird 2048 angenommen.

INPUT = *VARIABLE(...)

Die Eingabesätze haben variables Satzformat. Stimmen der einem vorangehenden ADD-FILE-LINK-Kommando oder dem Katalogeintrag entnommene und der angegebene Wert nicht überein, gibt SORT eine Warnung aus und ignoriert den angegebenen Wert.

MAXIMUM-RECORD-SIZE =

Maximale Länge der Eingabesätze.

MAXIMUM-RECORD-SIZE = *STD

Die Satzlänge wird einem vorher angegebenen ADD-FILE-LINK-Kommando oder dem Katalogeintrag entnommen. Ist das nicht möglich, wird als Ersatzwert die auf Grund der BUFFER-LENGTH-Angabe maximal mögliche Satzlänge übernommen. Wenn das nicht möglich ist, wird 2048 angenommen.

MAXIMUM-RECORD-SIZE = <integer 1..32759>

Angabe der maximalen Eingabesatzlänge. SORT verwendet den Maximalwert aus den Satzlängen aller Eingabedateien und der hier angegebenen Länge. Ist der angegebene Wert kleiner als der Maximalwert aus den Satzlängen aller Eingabedateien, gibt SORT eine Warnung aus.

INPUT = *FIXED(...)

Die Eingabesätze haben festes Satzformat. Stimmen der einem vorangehenden ADD-FILE-LINK-Kommando oder dem Katalogeintrag entnommene und der angegebene Wert nicht überein, gibt SORT eine Warnung aus und ignoriert den angegebenen Wert.

RECORD-SIZE =

Länge der Eingabesätze.

RECORD-SIZE = *STD

SORT versucht die Satzlänge aus einem vorher angegebenem ADD-FILE-LINK-Kommando oder aus dem Katalogeintrag zu erhalten. Kann die Satzlänge nicht festgestellt werden, wird die BUFFER-LENGTH übernommen. Wenn das nicht möglich ist, wird 2048 angenommen.

RECORD-SIZE = <integer 1..32759>

Länge der Eingabesätze. Stimmen der einem vorangehenden ADD-FILE-LINK-Kommando oder dem Katalogeintrag entnommene und der angegebene Wert nicht überein, gibt SORT eine Warnung aus und ignoriert den angegebenen Wert.

INTERNAL =

Beschreibung der internen Sätze. Muss nur angegeben werden, wenn das Satzformat oder die Satzlänge gegenüber den Eingabesätzen im Benutzerausgang INPUT oder durch SORT geändert werden soll. Ist der Benutzerausgang INPUT angegeben, so ist dieser für die Änderungen zuständig, andernfalls werden Satzformat oder Satzlänge durch SORT angepasst.

INTERNAL = *BY-CALCULATION

Satzformat und Satzlänge werden von den Eingabesätzen übernommen. Ist das Eingabesatzformat nicht angegeben, wird als Ersatzwert festes Satzformat genommen.

INTERNAL = *VARIABLE(...)

Die internen Sätze erhalten variables Satzformat. Haben die Eingabesätze festes Satzformat und wird die Anpassung von SORT durchgeführt, so wird den internen Sätzen ein Satzlängenfeld (4 Byte) vorangestellt.

MAXIMUM-RECORD-SIZE =

Maximale Länge der internen Sätze.

MAXIMUM-RECORD-SIZE = *BY-CALCULATION

Bei Eingabesätzen mit variablem Satzformat ist die maximale interne Satzlänge gleich der maximalen Eingabesatzlänge. Bei Eingabesätzen mit festem Satzformat ist die maximale interne Satzlänge gleich der Eingabesatzlänge + 4.

MAXIMUM-RECORD-SIZE = <integer 1..32759>

Angabe der maximalen Länge der internen Sätze. Eingabesätze fester Länge werden verkürzt, wenn ihre Länge größer ist als interne Länge minus 4, Eingabesätze variabler Länge werden verkürzt, wenn sie die angegebene Länge überschreiten.

MINIMUM-RECORD-SIZE =

Minimale Länge der internen Sätze. Wird für Kapazitätsberechnungen verwendet, sollte nur angegeben werden, wenn bekannt.

MINIMUM-RECORD-SIZE = *BY-CALCULATION

Wird von SORT berechnet.

MINIMUM-RECORD-SIZE = <integer 1..32759>

Angabe der minimalen Satzlänge.

AVERAGE-RECORD-SIZE =

Durchschnittliche Länge der internen Sätze. Wird für Kapazitätsberechnungen verwendet, sollte nur angegeben werden, wenn bekannt.

AVERAGE-RECORD-SIZE = *BY-CALCULATION

Wird von SORT berechnet.

AVERAGE-RECORD-SIZE = <integer 1..32759>

Angabe der durchschnittlichen Satzlänge.

INTERNAL = *FIXED(...)

Die internen Sätze erhalten festes Satzformat. Haben die Eingabesätze variables Satzformat und wird die Anpassung von SORT durchgeführt, so wird das Satzlängenfeld (4 Byte) der Eingabesätze abgeschnitten.

RECORD-SIZE =

Länge der internen Sätze.

RECORD-SIZE = *BY-CALCULATION

Bei Eingabesätzen mit festem Satzformat ist die interne Satzlänge gleich der Eingabesatzlänge. Bei Eingabesätzen mit variablem Satzformat ist die interne Satzlänge gleich der maximalen Eingabesatzlänge minus 4.

RECORD-SIZE = <integer 1..32759>

Angabe der internen Satzlänge. Wird die Anpassung von SORT durchgeführt, so werden die Eingabesätze, wenn sie die angegebene Länge überragen, abgeschnitten, andernfalls mit dem im FILLER-Operanden angegebenen Zeichen aufgefüllt.

OUTPUT =

Beschreibung der Ausgabesätze. Muss nur angegeben werden, wenn das Satzformat oder die Satzlänge gegenüber den internen Sätzen im Benutzeranfang OUTPUT oder durch SORT geändert werden soll. Ist der Benutzeranfang OUTPUT angegeben, so ist dieser für die Änderungen zuständig, andernfalls werden Satzformat oder Satzlänge durch SORT angepasst. Stimmt RECORD-FORMAT aus einem ADD-FILE-LINK-Kommando oder dem Katalogeintrag der Ausgabedatei nicht mit dem angegebenen Satzformat überein, wird der Sortierlauf fehlerhaft beendet. Eine fehlende RECORD-FORMAT-Angabe wird von SORT entsprechend ergänzt.

OUTPUT = *STD

Satzformat und Satzlänge werden von den internen Sätzen übernommen, wobei bei der Länge die eliminierten Sortierfelder und die Druckaufbereitungsmasken berücksichtigt werden.

OUTPUT = *VARIABLE(...)

Die Ausgabesätze erhalten variables Satzformat. Haben die internen Sätze festes Satzformat und wird die Anpassung von SORT durchgeführt, so wird den internen Sätzen ein Satzlängenfeld (4 Byte) vorangestellt.

MAXIMUM-RECORD-SIZE =

Maximale Satzlänge der Ausgabesätze, wird im Katalogeintrag der Ausgabedatei für RECORD-SIZE eingesetzt.

MAXIMUM-RECORD-SIZE = *STD

Bei internen Sätzen mit variablem Satzformat ist die maximale Ausgabesatzlänge gleich der maximalen Länge der internen Sätze. Bei internen Sätzen mit festem Satzformat ist die maximale Ausgabesatzlänge gleich der Länge der internen Sätze + 4.

MAXIMUM-RECORD-SIZE = <integer 1..32759>

Angabe der maximalen Ausgabesatzlänge. Interne Sätze fester Länge werden verkürzt, wenn ihre Länge größer ist als die angegebene Länge minus 4, interne Sätze variabler Länge werden verkürzt, wenn sie die angegebene Länge überschreiten.

OUTPUT = *FIXED(...)

Die Ausgabesätze erhalten festes Satzformat. Haben die internen Sätze variables Satzformat und wird die Anpassung von SORT durchgeführt, so wird das Satzlängengebiet (4 Byte) der internen Sätze abgeschnitten.

RECORD-SIZE =

Länge der Ausgabesätze. Stimmt die RECORD-SIZE aus einem ADD-FILE-LINK-Kommando oder dem Katalogeintrag der Ausgabedatei mit der Angabe nicht überein, so wird der Sortierlauf fehlerhaft beendet.

RECORD-SIZE = *STD

Bei internen Sätzen mit festem Satzformat ist die Ausgabesatzlänge gleich der Länge der internen Sätze. Bei internen Sätzen mit variablem Satzformat ist die Ausgabesatzlänge gleich der maximalen Länge der internen Sätze - 4. Zusätzlich werden zu eliminierende Sortierfelder und Druckaufbereitungsmasken berücksichtigt.

RECORD-SIZE = <integer 1..32759>

Angabe der Ausgabesatzlänge. Wird die Anpassung von SORT durchgeführt, so werden die internen Sätze, wenn sie die angegebene Länge überragen, abgeschnitten, andernfalls mit dem im FILLER-Operanden angegebenen Zeichen aufgefüllt.

FILLER =

Angabe eines 1 Byte langen Füllzeichens, mit dem Sätze bei Formatänderung (V → F) oder bei Satzlängenänderung (interne Satzlänge > Eingabesatzlänge oder Ausgabesatzlänge > interne Satzlänge) aufgefüllt werden.

FILLER = *NIL

Als Füllzeichen wird NIL (X'00') verwendet.

FILLER = <c-string 1..1 with-low>

Zeichenkonstante (1 Byte).

FILLER = <x-string 2..2>

Sedezimalkonstante (1 Byte).

CODED-CHARACTER-SET =

Angabe des Namens des coded character sets zur Sortierung der Datensätze. Dieser Operand wird von SORT nur ausgewertet bei Eingabe der Datensätze über den Benutzerausgang INPUT oder über die SORT-Zugriffsmethode. Ist eine Eingabedatei vorhanden, wird dieser Operand ignoriert und die Warnmeldung SRT1256 ausgegeben.

CODED-CHARACTER-SET = *NOT-SPECIFIED

Keine Angabe eines coded character sets. Der Code EDF03IRV wird verwendet.

CODED-CHARACTER-SET = <name 1..8>

Angabe des Namens des coded character sets.

UNICODE-FILLER =

Angabe eines zwei Byte langen Zeichens, mit dem ein Sortierfeld mit dem Format UNICODE-CHARACTER bei Bedarf aufgefüllt wird. Das ist z.B. nötig, wenn das Feld wegen variablen Sätzen nur unvollständig ist, oder wenn es wegen IGNORE-UNICODE-BLANK verkürzt wird.

UNICODE-FILLER = *BLANK

Es wird X'0020' (Unicode-Leerzeichen) verwendet.

UNICODE-FILLER = <x-string 4..4>

Das angegebene Zeichen wird verwendet.

SET-SORT-OPTIONS

Mit SET-SORT-OPTIONS legt der Benutzer fest

- den Einfluss des Satzlängenfelds auf die Positionsberechnung für Steuerfelder,
- die Ausgabe von Meldungen,
- die Überprüfung der Sortierfolge,
- Abweichungen von vorgegebenen Dateikettungsnamen,
- das Verhalten bei Fehlern,
- die Optimierung von Sortier- und Mischläufen,
- die Behandlung von benutzten Dateien und Bändern,
- die Ausgabe von Diagnoseunterlagen,
- ob Leerzeichen oder andere variable Sortierelemente (Collation-Elemente) beim Festlegen der Sortierreihenfolge berücksichtigt werden.

SET-SORT-OPTIONS

```

MIN-MSG-WEIGHT = *NORMAL / *ALL / *CRITICAL / *NONE
, SEQUENCE-CHECK = *YES / *NO
, LINK-PREFIX-CHANGE = *NO / <name 1..4>
, IGNORE-INOUT-FILE = *NO / *YES
, INPUT-OPEN-ERROR = *CONTINUE-NEXT-FILE(...) / *FINISH-INPUT / *TERMINATE-ABNORMAL
  *CONTINUE-NEXT-FILE(...)
    | TERMINATION = *ABNORMAL / *NORMAL
, OPTIMIZATION = *RUN-TIME / *CPU-TIME / *VIRTUAL-MEMORY
, DESTROY-WORK-FILES = *NO / *YES
, KEEP-INPUT-TAPES = *NO / *YES
, DUMP = *YES / *NO
, STATEMENT-CCSN = *NOT-SPECIFIED / <name 1..8>
, IGNORE-LENGTH-FIELD = *STD / *YES / *NO
, DELETE-WORK-FILES = *NO / *YES
, CENTURY-WINDOW-SHIFT = 50(...) / <integer 0..99> (...)
  <integer 0..99>(…)
    | WINDOW-BOUNDARY = *SUBTRACT-FROM-CURRENT-YEAR / *ADD-TO-CURRENT-YEAR
, IGNORE-UNICODE-BLANK = *NO / *YES
, IGNORE-CHARACTER = *NONE / < list-poss(4):<c-string 1..1 with low> / <x-string 1..2>

```

MIN-MSG-WEIGHT =

Angabe der Priorität, ab der Meldungen ausgegeben werden sollen. Die Meldung wird dann ausgegeben, wenn deren Priorität größer oder gleich der angegebenen ist.

MIN-MSG-WEIGHT = *NORMAL

Bei autonomen Sortier-/Mischläufen werden Meldungen ab Priorität 2 oder ab der voreingestellten Priorität ausgegeben. Bei SORT als Unterprogramm werden Meldungen ab Priorität 3 ausgegeben.

MIN-MSG-WEIGHT = *ALL

Alle Meldungen werden ausgegeben (ab Priorität 0).

MIN-MSG-WEIGHT = *CRITICAL

Meldungen ab Priorität 3 werden ausgegeben.

MIN-MSG-WEIGHT = *NONE

Meldungen ab Priorität 7 werden ausgegeben (nur Meldungen über interne Fehler).

SEQUENCE-CHECK =

Mit diesem Operanden kann man die Überprüfung der auf- oder absteigenden Sortierfolge vor der Endausgabe steuern. Bei Verwendung der SUM-RECORDS-Anweisung wird die Sortierfolge auf jeden Fall überprüft, SEQUENCE-CHECK=*NO bleibt in diesem Fall unwirksam. Die Folgekontrolle erfolgt vor einem eventuell angegebenen Benutzerausgang OUTPUT. Die mit diesem Benutzerausgang eingefügten oder geänderten Sätze können daher die Sortierfolge in Unordnung bringen. Die Angabe des SEQUENCE-CHECK-Operanden hat Vorrang vor dem entsprechenden Kennzeichenbyte im Benutzerausgang OUTPUT (nur bei PARAMETER-MODE=24).

SEQUENCE-CHECK = *YES

SORT überprüft vor der Endausgabe die auf- oder absteigende Sortierfolge.

SEQUENCE-CHECK = *NO

Die Sortierfolge wird nicht überprüft.

LINK-PREFIX-CHANGE =

Vereinbarung alternativer Präfixe für folgende Dateikettungsamen: SORTIN / SORTINxx / MERGExx / SORTOUT / SORTWK / SORTWKx / SORTWKxx / SORTCKPT / SORTMODS. In diesen Dateikettungsamen wird die Zeichenfolge „SORT“ bzw. „MERGE“ durch die angegebene Zeichenfolge (1 bis 4 Zeichen) ersetzt. Damit können mehrere SORT-Läufe von einem Hauptprogramm aus gleichzeitig durchgeführt werden. Das alternative Präfix wird auch bei der Generierung der Namen der von SORT angelegten Dateien (Arbeits-, Hilfs- und Fixpunktdateien) verwendet.

LINK-PREFIX-CHANGE = *NO

Das Präfix wird nicht verändert.

LINK-PREFIX-CHANGE = <name 1..4>

Angabe eines Präfix.

IGNORE-INOUT-FILE =

Mit diesem Operanden kann man angeben, ob SORT die Dateikettungsamen SORTIN/ SORTINxx für die SORT-Eingabedatei und SORTOUT für die SORT-Ausgabedatei ignorieren soll. Dies ist anzuwenden, wenn die Eingabe und die Ausgabe über die Benutzer- ausgänge INPUT und OUTPUT erfolgt.

IGNORE-INOUT-FILE = *NO

SORT berücksichtigt die Dateikettungsamen.

IGNORE-INOUT-FILE = *YES

Die Dateikettungsamen werden nicht berücksichtigt.

INPUT-OPEN-ERROR =

Mit diesem Operanden kann man festlegen, wie sich SORT verhalten soll, wenn bei mehreren Eingabedateien eine der Eingabedateien nicht eröffnet werden kann.

INPUT-OPEN-ERROR = *CONTINUE-NEXT-FILE(...)

SORT meldet den Fehler, übergeht die betroffene Datei und bearbeitet die übrigen Dateien.

TERMINATION =**TERMINATION = *ABNORMAL**

Der Sortier-/Mischlauf wird fehlerhaft beendet. In einem autonomen Sortier-/Mischlauf wird mit TERM UNIT=STEP,MODE=ABNORMAL beendet. Bei SORT als Unterprogramm enthält Register 15 im niederwertigsten Byte X'FF' und in den ersten beiden Byte die Meldungsnummer (z.B. bei Eröffnungsfehler SRT1035 hat Register 15 den Inhalt X'103500FF').

TERMINATION = *NORMAL

Der Sortier-/Mischlauf wird normal beendet.

INPUT-OPEN-ERROR = *FINISH-INPUT

SORT meldet den Fehler, beendet die Eingabe und sortiert die eingelesenen Sätze.

INPUT-OPEN-ERROR = *TERMINATE-ABNORMAL

SORT meldet den Fehler und beendet abnormal.

OPTIMIZATION =

Optimierung des Sortier-/Mischlaufes.

OPTIMIZATION = *RUN-TIME

Es wird eine Optimierung der Laufzeit gefordert.

OPTIMIZATION = *CPU-TIME

Es wird eine Optimierung der CPU-Zeit gefordert. Zwischen den Angaben *RUN-TIME und *CPU-TIME besteht zurzeit kein Unterschied.

OPTIMIZATION = *VIRTUAL-MEMORY

Es wird eine Optimierung des Speicherplatzes gefordert (z.B. nicht mehr benötigte Lademodule werden freigegeben). Bei SORT als Unterprogramm werden nach Beendigung des Sortierlaufes alle dynamisch nachgeladenen Module wieder entladen (SRTXGEN, SRTXKERN und SRTXKRN1).

DESTROY-WORK-FILES =

Art des Löschens der von SORT angelegten Dateien.

DESTROY-WORK-FILES = *NO

Die von SORT eingerichteten Dateien werden logisch gelöscht, d.h. es wird nur der Katalogeintrag gelöscht.

DESTROY-WORK-FILES = *YES

Die von SORT eingerichteten Dateien werden logisch und physikalisch gelöscht, d.h. es wird der Katalogeintrag gelöscht und die Datei mit binären Nullen überschrieben. Kann aus Datenschutzgründen angegeben werden.

KEEP-INPUT-TAPES =

Steuert das Entladen von Eingabebändern.

KEEP-INPUT-TAPES = *NO

Eingabebänder werden nach dem Einlesen zurückgespult und entladen, um das Bandgerät für weitere Dateien freizumachen.

KEEP-INPUT-TAPES = *YES

Eingabebänder werden nach dem Einlesen nur zurückgespult und nicht entladen, sie können von einem anderen Programm weiterverwendet werden. Bei Multi-Volume-Dateien wird allerdings nur das letzte Band nicht entladen.

DUMP =

Steuert die Ausgabe von Speicherinhalten und fehlerhaften Daten nach SYSOUT und SYSLST bei Auftreten von internen SORT-Fehlern bzw. Anwendungsfehlern.

DUMP = *YES

Diagnosehilfen werden ausgegeben.

DUMP = *NO

Diagnosehilfen werden unterdrückt. Dieser Wert sollte nur angegeben werden, wenn schwerwiegende Datenschutzgründe dafür sprechen. Es besteht die Gefahr, dass SORT- oder Anwendungsfehler nicht diagnostiziert werden können.

STATEMENT-CCSN =

Angabe des coded character sets der Zeichenkonstanten in den Anweisungen. Die Angabe gilt für die Konstanten aller Anweisungen des Programmlaufes. Dieser Operand wird von SORT nur ausgewertet bei Aufruf von SORT als Unterprogramm.

STATEMENT-CCSN = *NOT-SPECIFIED

Keine Angabe eines coded character sets. Der Code EDF03IRV wird verwendet.

STATEMENT-CCSN = <name 1..8>

Angabe des Namens des coded character sets.

IGNORE-LENGTH-FIELD =

Angabe, ob bei variabler Satzlänge das Satzlängenfeld bei Positionsangaben für Steuerfelder berücksichtigt werden muss.

Die Angabe gilt für alle Steuerfelder im aktuellen Sortierlauf. Auf Positionsangaben in Sätzen fester Länge hat sie keinen Einfluss.

IGNORE-LENGTH-FIELD = *STD

Bei BS2000-Dateien wirkt *STD wie *NO, d.h. das Satzlängenfeld muss bei Positionsangaben berücksichtigt werden. Das Verhalten ist demnach wie in den Vorversionen.

Bei POSIX-Dateien wirkt *STD wie *YES. Das Satzlängenfeld, das von SORT intern angelegt wird, wird also bei der Positionsberechnung ignoriert.

IGNORE-LENGTH-FIELD = *YES

Das Satzlängenfeld wird bei der Positionsberechnung ignoriert. Das erste Datenfeld eines Satzes beginnt auf Position 1.

IGNORE-LENGTH-FIELD = *NO

Das Satzlängenfeld wird bei der Berechnung berücksichtigt. Das erste Datenfeld eines Satzes beginnt auf Position 5.

DELETE-WORK-FILES =

Gibt an, ob Arbeits- und Hilfsdateien (SORTWK, SORTWKx, SORTWKxx), die vom Benutzer angelegt worden sind, am Ende des Sortierlaufs von SORT gelöscht werden.

DELETE-WORK-FILES = *NO

Die Arbeits- und Hilfsdateien werden am Ende des Sortierlaufs nur dann gelöscht, wenn sie von SORT angelegt worden sind.

DELETE-WORK-FILES = *YES

Die Arbeits- und Hilfsdateien werden am Ende des Sortierlaufs auch dann gelöscht, wenn sie vom Benutzer angelegt worden sind.

Hinweis

Erfolgt ein SORT-Abbruch bei einem SORT-Lauf, bei dem Prüfpunkt/Wiederanlauf genutzt wird, bleiben die Arbeits- und Hilfsdateien unabhängig vom Wert des Operanden DELETE-WORK-FILES erhalten, um einen Wiederanlauf zu ermöglichen.

CENTURY-WINDOW-SHIFT = 50(...) / <integer 0..99>(…)

Legt fest, wie Felder, die zweistellige Jahreszahlen enthalten (FORMAT=*PACKED-DECIMAL oder *ZONED-DECIMAL und TWO-DIGIT-YEAR=*YES) von SORT interpretiert und sortiert werden. Der Operand CENTURY-WINDOW-SHIFT legt ein Jahrhundertfenster fest, innerhalb dessen auf- oder absteigend sortiert wird. Abhängig vom Operanden WINDOW-BOUNDARY wird dieses Jahrhundertfenster durch seinen Anfang oder durch sein Ende bestimmt.

WINDOW-BOUNDARY =

Legt fest, auf welche Weise Beginn und Ende des Jahrhundertfensters berechnet werden.

WINDOW-BOUNDARY = *SUBTRACT-FROM-CURRENT-YEAR

Der **Beginn** des Jahrhundertfensters wird folgendermaßen berechnet:

<beginn-jh-fenster> = <aktuelles-jahr> - <integer-in-century-window-shift>

Da das Jahrhundertfenster 100 Jahre lang ist, ist sein **Ende** 99 Jahre später, also:

<ende-jh-fenster> = <beginn-jh-fenster> + 99

WINDOW-BOUNDARY = *ADD-TO-CURRENT-YEAR

Das **Ende** des Jahrhundertfensters wird folgendermaßen berechnet:

$$\langle \text{ende-jh-fenster} \rangle = \langle \text{aktuelles-jahr} \rangle + \langle \text{integer-in-century-window-shift} \rangle$$

Da das Jahrhundertfenster 100 Jahre lang ist, ist sein **Beginn** 99 Jahre früher, also:

$$\langle \text{beginn-jh-fenster} \rangle = \langle \text{ende-jh-fenster} \rangle - 99.$$

Beispiel:

Das aktuelle Jahr sei 2002 und in der Anweisung SET-SORT-OPTIONS ist CENTURY-WINDOW-SHIFT=55 angegeben. Das Jahrhundertfenster geht demnach von 1947 (=2002-55) bis 2046 (=1947+99).

Für die Sortierung bedeutet das:

Die zweistelligen Jahreszahlen 00 bis 46 werden als 2000 bis 2046 interpretiert und sind somit größer als die Jahreszahlen 47 bis 99, da diese als 1947 bis 1999 interpretiert werden.

Die Sortierreihenfolge ist in diesem Fall (bei aufsteigender Sortierung):

47, 48, ..., 98, 99, 00, 01, ..., 45, 46.

IGNORE-UNICODE-BLANK =

Mit diesem Operanden wird festgelegt, ob Leerzeichen oder andere durch variable Sortierelemente (Collation-Elemente) vertretene Zeichen beim Festlegen der Sortierreihenfolge berücksichtigt werden.

IGNORE-UNICODE-BLANK = *NO

Alle Zeichen werden berücksichtigt.

IGNORE-UNICODE-BLANK = *YES

Leerzeichen und andere durch variable Collation-Elemente vertretene Zeichen werden nicht zur Festlegung der Sortierreihenfolge herangezogen. Der Sortierbegriff wird zusammengezogen und am Ende gegebenenfalls mit dem Unicode-Füllzeichen aufgefüllt.

IGNORE-CHARACTER =

Mit diesem Operanden wird festgelegt, ob, und wenn ja, welche Zeichen bei der Sortierreihenfolge ignoriert werden sollen. Das Ignorieren der Zeichen erfolgt für die Formate *CHARACTER und *TRANSLATE-CHARACTER. Die Zeichen werden zwar beim Sortierfeldvergleich ignoriert, bleiben aber im Satz erhalten.

IGNORE-CHARACTER = *NONE

Es sind keine Zeichen zu ignorieren.

IGNORE-CHARACTER = list-poss(4): <c-string 1..1 with-low>

Zeichenkonstante (1 Byte)

IGNORE-CHARACTER = list-poss(4): <x-string 1..2>

Sedezimalkonstante (1 Byte)

Beispiele:

```
SET-SORT-OPTIONS IGNORE-CHARACTER=C' '  
SET-SORT-OPTIONS IGNORE-CHARACTER=X'40'  
SET-SORT-OPTIONS IGNORE-CHARACTER=(C' ',C'_)  
SET-SORT-OPTIONS IGNORE-CHARACTER=(C' ',X'6D')  
SET-SORT-OPTIONS IGNORE-CHARACTER=(C'(',C)'),C'<',C'>'  
SET-SORT-OPTIONS IGNORE-CHARACTER=*NONE
```

SHOW-SORT-DEFAULTS

Funktion: Anzeigen von Voreinstellwerten.

Die aktuell gültigen Voreinstellungen für folgende SORT-Parameter werden ausgegeben:

- minimales Meldungsgewicht, das die Ausgabe von Meldungen bewirkt (MIN-MSG-WEIGHT)
- Maximalwert für intensiv genutzten Speicher (CORE-MAXIMUM)
- Untergrenze für die CORE-Wert-Berechnung (CORE-MINIMUM)
- Obergrenze für die CORE-Wert-Berechnung (CORE-LIMIT)
- Standardwert für den CORE-Wert (CORE-DEFAULT)
- Jobklasse für Nebentasks (SUBTASK-JOB-CLASS)

SHOW-SORT-DEFAULTS

Diese Anweisung hat keine Operanden.


```

*FIELD-SYMBOLIC(...)
  |
  | SORT-FIELD-NAME = list-poss(64): <name 1..20>
  |
  | ,SORTING-ORDER = *ASCENDING / *DESCENDING / *EXTERNAL-COMPARE
  |
  | ,PRIORITY = *CURRENT-NUMBER / <integer 1..64>
  |
  | ,ELIMINATE = *NO / *YES
  |
  | ,PRINT-MASK-NAME = *NO / <name 1..20>
  |
  | ,TWO-DIGIT-YEAR = *NO / *YES
  |
*REMAINDER-EXPLICIT(...)
  |
  | POSITION = <integer 1..32759>
  |
  | ,LENGTH = <integer 1..32759>
  |
  | ,FORMAT = *NO / *BINARY / *FIXED-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL
  |
  | ,PRINT-MASK = *NO / <c-string 1..254 with-low>
  |
*REMAINDER-SYMBOLIC(...)
  |
  | FIELD-NAME = list-poss(64): <name 1..20>
  |
  | ,PRINT-MASK-NAME = *NO / <name 1..20>
  |
*CONSTANT-EXPLICIT(...)
  |
  | CONSTANT = <integer -2147483639..2147483639> / <c-string 1..256 with-low> /
  |                 <x-string 1..512>
  |
*CONSTANT-SYMBOLIC(...)
  |
  | CONSTANT-NAME = list-poss(64): <name 1..20>
  |
,SORT-TYPE = *COMPLETE-RECORD / *COMPOUND-RECORD / *TAG-COMPOUND / *TAG-HEADER /
             *TAG-TRAILER
,ESTIMATED-RECORDS = *BY-CALCULATION / <integer 1..2147483639> / <alphanumeric-name 1..19>
,RECORDS-PER-CYCLE = *BY-CALCULATION / <integer 1..2147483639>
,INPUT-RANGE = *ALL / *PARAMETER(...)
  |
  | *PARAMETER(...)
  |   |
  |   | FROM-RECORD = 0 / <integer 0..2147483639> / <alphanumeric-name 1..19>
  |   |
  |   | ,NUMBER-OF-RECORDS = *REST-INPUT / <integer 1..2147483639> / <alphanumeric-name 1..19>
  |   |
,CHECKPOINT = *NO / *YES
,KEEP-EQUAL-SEQUENCES = *NO / *YES

```

(Teil 2 von 2)

FIELDS =

Im FIELDS-Operanden werden Sortier-, Rest- und Konstantenfelder definiert. Mindestens ein Feld muss, maximal 64 Felder dürfen angegeben werden.

FIELDS = *COMPLETE-RECORD

Bei festem Satzformat wird der ganze Satz als Sortierfeld verwendet. Bei variablem Satzformat wird das Satzlängenfeld nicht in den Satzvergleich einbezogen. Der variable Teil des Satzes wird für den Satzvergleich bis zur maximalen Satzlänge mit dem Füllzeichen des FILLER-Operanden der SET-RECORD-ATTRIBUTES-Anweisung aufgefüllt. Die Angabe von *COMPLETE-RECORD ist nur beim Vollsordieren (SORT-TYPE=*COMPLETE-RECORD) erlaubt.

FIELDS = *FIELD-EXPLICIT(...)

Angabe von Sortierfeldern durch Position, Länge und Format.

POSITION = <integer 1..32759(...)>

Position des Sortierfeldes relativ zum Satzanfang.

Bei Sätzen mit festem Satzformat hat das erste Datenfeld die Position 1. Bei Sätzen mit variablem Satzformat hängt die Position des ersten Datenfelds von der Dateiart und dem Operanden IGNORE-LENGTH-FIELD der Anweisung SET-SORT-OPTIONS ab. Diese Abhängigkeit wird in folgender Tabelle dargestellt:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Bei allen Formaten außer CHARACTER, EBCDIC-DIN, EBCDIC-INTERNATIONAL, EXTENDED-CHARACTER und TRANSLATE-CHARACTER müssen die Positionsangaben für Sortierfelder im Bereich 1 bis 4096 liegen.

BIT-POSITION = 0 / <integer 0..7>

Positionsangabe in Bit zusätzlich zur Positionsangabe in Byte. Die Angabe ist nur für Sortierfelder mit Format BINARY erlaubt, wenn der Operand PRINT-MASK nicht angegeben ist.

LENGTH = <integer 0..32759(...)>

Länge des Sortierfeldes. Die Länge muss innerhalb der für das Format erlaubten Grenzen liegen. Durch die Angabe von BIT-POSITION und NUMBER-OF-BITS vergrößert sich die Länge des Sortierfeldes um 1 Byte ($\text{BIT-POSITION} + \text{NUMBER-OF-BITS} \leq 7$) oder um 2 Byte ($\text{BIT-POSITION} + \text{NUMBER-OF-BITS} > 7$). Dies ist bei der maximalen Länge zu berücksichtigen. Bei Sätzen variabler Länge dürfen sich Sortierfelder vom Format CHARACTER, EBCDIC-DIN und EBCDIC-INTERNATIONAL auch in den variablen Teil des Satzes hinein fortsetzen.

NUMBER-OF-BITS = 0 / <integer 0..7>

Längenangabe in Bits zusätzlich zur Längenangabe in Byte. Die Angabe ist nur für Sortierfelder mit Format BINARY erlaubt, wenn der Operand PRINT-MASK nicht angegeben ist.

FORMAT = *CHARACTER / *BINARY / *FIXED-POINT / *FLOATING-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN / *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE / *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER / *TRANSLATE-CHARACTER / *UNICODE-CHARACTER

Format des Sortierfeldes (Eigenschaften siehe [Seite 38](#)).

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

SORTING-ORDER = *EXTERNAL-COMPARE

Die Reihenfolge für dieses Sortierfeld wird über den Benutzerausgang EXTERNAL-COMPARE festgelegt. Dieses Sortierfeld darf max. 255 Byte lang sein.

PRIORITY =

Wertigkeit des Sortierfeldes. Sind mehrere Sortierfelder angegeben, werden zuerst die Sortierfelder mit der Wertigkeit 1 verglichen. Sind sie gleich, werden die Sortierfelder mit der Wertigkeit 2 verglichen. Dieser Vorgang wird solange fortgesetzt, bis die Sortierfelder ungleich sind oder die Sortierfelder mit der höchsten Wertigkeit verglichen wurden.

PRIORITY = *CURRENT-NUMBER

Die Wertigkeit wird durch die Reihenfolge, in der die Sortierfelder angegeben werden, bestimmt (das erste Sortierfeld hat die Wertigkeit 1).

PRIORITY = <integer 1..64>

Angabe der Ordnungsziffer. Wird die Ordnungsziffer bei einem Sortierfeld angegeben, so muss sie auch bei allen anderen Sortierfeldern angegeben werden. Alle Ordnungsziffern müssen eine mit 1 beginnende, lückenlos aufsteigende Reihenfolge ergeben.

ELIMINATE =

Angabe, ob das Sortierfeld eliminiert (nicht in die Ausgabe übernommen) wird.

ELIMINATE = *NO

Das Sortierfeld wird nicht eliminiert.

ELIMINATE = *YES

Das Sortierfeld wird eliminiert. Es dient nur zur Feststellung der Sortierreihenfolge. Für binäre Felder mit Bit-Angaben und Maskenfelder ist diese Angabe nicht zulässig. SORT gibt in diesem Fall eine Warnung aus und ignoriert die Angabe. Beim Benutzerausgang OUTPUT (mit PARAMETER-MODE=24) werden gleiche Sätze nicht angezeigt.

PRINT-MASK =

Aufbereitungsmaske, mit der das Feld zum Druck aufbereitet wird. Eine Aufbereitungsmaske ist nur beim Auswahlsortieren und nur bei den Formaten BINARY, FIXED-POINT, PACKED-DECIMAL und ZONED-DECIMAL erlaubt. Die Länge aller Konstanten und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich ein Byte für ein Längenfeld berücksichtigt werden muss.

PRINT-MASK = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK = <c-string 1..254>

Format der Aufbereitungsmaske. Als Maskenzeichen sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Maske.
- Die Steuerzeichen '#' (Nummernzeichen) und '^' (Circumflex).
- Einzufügende Zeichen (ungleich den Steuerzeichen).

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT der Anweisung SET-SORT-OPTIONS ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fjv'

Dabei sind: j = eine Stelle der zweistelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

FIELDS = *FIELD-SYMBOLIC(...)

Angabe von Sortierfeldern durch symbolische Namen, die vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Felder vereinbart werden müssen.

SORT-FIELD-NAME = list-poss(64): <name 1..20>

Name eines Sortierfeldes (max. 64 Felder). Wird eine Liste von symbolischen Namen angegeben, so gelten die bei SORTING-ORDER, ELIMINATE und PRINT-MASK-NAME gemachten Angaben für alle Namen, der Wert für PRIORITY wird für den ersten angegebenen Namen übernommen, für die restlichen Namen wird ein jeweils um 1 erhöhter Wert eingetragen.

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

SORTING-ORDER = *EXTERNAL-COMPARE

Die Reihenfolge für dieses Sortierfeld wird über den Benutzerausgang EXTERNAL-COMPARE festgelegt. Dieses Sortierfeld darf max. 255 Byte lang sein.

PRIORITY =

Wertigkeit des Sortierfeldes. Sind mehrere Sortierfelder angegeben, werden zuerst die Sortierfelder mit der Wertigkeit 1 verglichen. Sind sie gleich, werden die Sortierfelder mit der Wertigkeit 2 verglichen. Dieser Vorgang wird solange fortgesetzt, bis die Sortierfelder ungleich sind oder die Sortierfelder mit der höchsten Wertigkeit verglichen wurden.

PRIORITY = *CURRENT-NUMBER

Die Wertigkeit wird durch die Reihenfolge, in der die Sortierfelder angegeben werden, bestimmt (das erste Sortierfeld hat die Wertigkeit 1).

PRIORITY = <integer 1..64>

Angabe der Ordnungsziffer. Wird die Ordnungsziffer bei einem Sortierfeld angegeben, so muss sie auch bei allen anderen Sortierfeldern angegeben werden. Alle Ordnungsziffern müssen eine mit 1 beginnende, lückenlos aufsteigende Reihenfolge ergeben.

ELIMINATE =

Angabe, ob das Sortierfeld eliminiert (nicht in die Ausgabe übernommen) wird.

ELIMINATE = *NO

Das Sortierfeld wird nicht eliminiert.

ELIMINATE = *YES

Das Sortierfeld wird eliminiert. Es dient nur zur Feststellung der Sortierreihenfolge. Für binäre Felder mit Bit-Angaben und Maskenfelder ist diese Angabe nicht zulässig. SORT gibt in diesem Fall eine Warnung aus und ignoriert die Angabe. Beim Benutzerausgang OUTPUT (mit PARAMETER-MODE=24) werden gleiche Sätze nicht angezeigt.

PRINT-MASK-NAME =

Angabe einer Druckaufbereitungsmaske durch einen symbolischen Namen. Eine Aufbereitungsmaske ist nur beim Auswahl-sortieren und nur bei den Formaten BINARY, FIXED-POINT, PACKED-DECIMAL und ZONED-DECIMAL erlaubt. Die Länge aller Konstanten und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich ein Byte für ein Längenfeld berücksichtigt werden muss.

PRINT-MASK-NAME = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK-NAME = <c-string 1..254>

Format der Aufbereitungsmaske. Als Maskenzeichen sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Maske.
- Die Steuerzeichen '#' (Nummernzeichen) und '^' (Circumflex).
- Einzufügende Zeichen (ungleich den Steuerzeichen).

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT der Anweisung SET-SORT-OPTIONS ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fvj'

Dabei sind: j = eine Stelle der zweistelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

FIELDS = *REMAINDER-EXPLICIT(...)

Angabe von Restfeldern durch Position und Länge. Restfelder werden beim Auswahl- oder Adresslistensortieren für den Aufbau des Ausgabesatzes verwendet, sie haben keinen Einfluss auf die Sortierreihenfolge.

POSITION = <integer 1..32759>

Position des Restfeldes relativ zum Satzanfang.

Bei Sätzen mit festem Satzformat hat das erste Datenfeld die Position 1. Bei Sätzen mit variablem Satzformat hängt die Position des ersten Datenfelds von der Dateiart und dem Operanden IGNORE-LENGTH-FIELD der Anweisung SET-SORT-OPTIONS ab. Diese Abhängigkeit wird in folgender Tabelle dargestellt:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Restfelder dürfen an beliebiger Stelle des Satzes beginnen. Bei Sätzen variabler Länge darf beim Auswahl-sortieren ein Restfeld, wenn es das letzte angegebene Feld ist, auch ganz oder teilweise im variablen Teil des Satzes liegen. Bei Sätzen mit fester Länge muss das Restfeld ganz im Satz liegen.

LENGTH = <integer 1..32759>

Länge des Restfeldes. Restfelder sind innerhalb der Satzlänge nicht begrenzt.

FORMAT = *NO / *BINARY / *FIXED-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL

Format des Restfeldes. Die Angabe eines Formats ungleich *NO ist nur zusammen mit einer Aufbereitungsmaske erlaubt.

Format	Formatbezeichnung	Länge in Byte
BINARY	binär	1 Bit - 256 Byte
FIXED-POINT	Festpunkt	1 - 256
PACKED-DECIMAL	Dezimal gepackt	1 - 16
ZONED-DECIMAL	Dezimal entpackt	1 - 16
NO		1 - max. Satzlänge

PRINT-MASK =

Aufbereitungsmaske, mit der das Feld zum Druck aufbereitet wird. Das Restfeld muss eines der Formate BINARY, FIXED-POINT, PACKED-DECIMAL oder ZONED-DECIMAL haben.

Die Länge aller Konstanten und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich 1 Byte für ein Längenfeld zu berücksichtigen ist.

PRINT-MASK = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK = <c-string 1..254>

Format der Aufbereitungsmaske. Als Maskenzeichen sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Maske.
- Die Steuerzeichen '#' (Nummernzeichen) und '^' (Circumflex).
- einzufügende Zeichen (ungleich den Steuerzeichen).

FIELDS = *REMAINDER-SYMBOLIC(...)

Angabe von Restfeldern durch symbolische Namen, die vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Felder vereinbart werden müssen. Restfelder werden beim Auswahl- oder Adresslistensortieren für den Aufbau des Ausgabesatzes verwendet, sie haben keinen Einfluss auf die Sortierreihenfolge.

FIELD-NAME = list-poss(64): <name 1..20>

Name des Restfeldes (max 64 Felder). Wird eine Liste von symbolischen Namen angegeben, gelten die bei PRINT-MASK-NAME gemachten Angaben für alle Namen.

PRINT-MASK-NAME =

Angabe einer Druckaufbereitungsmaske durch einen symbolischen Namen, der vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Maske vereinbart werden muss. Das Restfeld muss eines der Formate BINARY, FIXED-POINT, PACKED-DECIMAL oder ZONED-DECIMAL haben. Die Länge aller Konstanten und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich 1 Byte für ein Längenfeld zu berücksichtigen ist.

PRINT-MASK-NAME = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK-NAME = <name 1..20>

Name der Aufbereitungsmaske.

FIELDS = *CONSTANT-EXPLICIT(...)

Angabe von Konstanten, die beim Auswahl- oder Adresslistensortieren für den Aufbau des Ausgabesatzes verwendet werden.

CONSTANT =

Wert der Konstanten. Die gesamte Länge aller Konstanten und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich 1 Byte für ein Längenfeld zu berücksichtigen ist.

CONSTANT = <integer-2147483639..2147483639>

Dezimalzahlen werden von SORT in Festpunktzahlen in der Länge von 4 Byte umgewandelt.

CONSTANT = <c-string 1..256 with-low>

Zeichenstring. Ein Hochkomma in einem Zeichenstring muss durch 2 aufeinander folgende Hochkommata angegeben werden.

CONSTANT = <x-string 1..512>

Sedezimalstring.

FIELDS = *CONSTANT-SYMBOLIC(...)

Angabe von Konstanten durch einen symbolischen Namen, der vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Konstante vereinbart werden muss. Die Konstanten werden beim Auswahl- oder Adresslistensortieren für den Aufbau des Ausgabesatzes verwendet.

CONSTANT-NAME = list-poss(64): <name 1..20>

Angabe eines symbolischen Namens. Die gesamte Länge aller Konstantenfelder und Druckaufbereitungsmasken darf 4000 Byte nicht überschreiten, wobei für jede Maske zusätzlich ein Byte für ein Längenfeld zu berücksichtigen ist.

SORT-TYPE =

Sortierart, nach der sortiert werden soll.

SORT-TYPE = *COMPLETE-RECORD

Vollsortierung. Der ganze Eingabesatz wird zur Sortierung übernommen.

SORT-TYPE = *COMPOUND-RECORD

Auswahlsortierung. Der zu sortierende Satz setzt sich aus den Teilen des Eingabesatzes zusammen, die im FIELDS-Operanden als Sortier-, Restfelder und Konstanten angegeben wurden. Die Reihenfolge, in der diese Felder zum Ausgabesatz zusammengestellt werden, wird durch die Reihenfolge, in der die Felder definiert wurden, bestimmt. Beim Auswahl-sortieren werden, unabhängig vom Satzformat der Eingabesätze, standardmäßig Sätze fester Länge erzeugt.

SORT-TYPE = *TAG-COMPOUND

Adresslistensortierung. SORT stellt die Satzadresse, bei SAM-Dateien eine erweiterte Satzadresse (sechsstellig) an den Anfang des Auswahl-satzes.

SORT-TYPE = *TAG-HEADER

Adresslistensortierung. SORT stellt die Satzadresse an den Anfang des Auswahlsatzes.

SORT-TYPE = *TAG-TRAILER

Adresslistensortierung. SORT fügt die Satzadresse am Ende des Auswahlsatzes an.

ESTIMATED-RECORDS =

Ungefähre Anzahl der zu bearbeitenden Sätze. SORT kann dadurch die Sortierstrategie und die Speicheranforderung genauer festlegen.

ESTIMATED-RECORDS = *BY-CALCULATION

Die ungefähre Satzanzahl wird nicht angegeben.

ESTIMATED-RECORDS = <integer 1..2147483639>

Geschätzte Anzahl der zu sortierenden Sätze. Sätze, die über Benutzerroutrinen eingefügt oder weggelassen werden, sollten bei dieser Angabe berücksichtigt werden.

ESTIMATED-RECORDS = <alphanum-name 1..19>

Geschätzte Anzahl der zu sortierenden Sätze, sofern sie die Größe von 2.14.483.639 überschreitet.

RECORDS-PER-CYCLE =

Größe eines Sortier-Zyklus. SORT soll jeweils die angegebene Anzahl von Sätzen vorsortieren, intern mischen und die so sortierte Folge von Sätzen auf eine Hilfsdatei ausgeben. Wurde beim Operanden ESTIMATED-RECORDS eine Angabe gemacht, ergibt sich die Anzahl der Hilfsdateien aus der Angabe bei ESTIMATED-RECORDS dividiert durch die Angabe bei RECORDS-PER-CYCLE. SORT richtet die errechnete Anzahl von Hilfsdateien ein bzw. ergänzt die Anzahl der bereits vorhandenen Hilfsdateien auf die berechnete Anzahl.

RECORDS-PER-CYCLE = *BY-CALCULATION

SORT führt keine Zyklussortierung durch.

RECORDS-PER-CYCLE = <integer 1..2147483639>

Anzahl der Sätze, die in einem Zyklus sortiert werden sollen.

INPUT-RANGE =

Ausschluss von Sätzen am Beginn oder Ende der Eingabedatei von der Sortierung.

INPUT-RANGE = *ALL

Alle Sätze werden in die Sortierung aufgenommen.

INPUT-RANGE = *PARAMETER(...)

Angabe eines Satzbereiches, der in die Sortierung aufgenommen werden soll.

FROM-RECORD =

Anzahl der Sätze, die SORT vom Beginn der Eingabedatei an überlesen soll.

FROM-RECORD = 0

Es sollen keine Sätze überlesen werden.

FROM-RECORD = <integer 1..2147483639>

Anzahl der Sätze, die SORT vom Beginn der Eingabedatei an überlesen soll.

FROM-RECORD = <alphanum-name 1..19>

Anzahl der Sätze, sofern sie die Größe von 2.14.483.639 überschreitet, die SORT vom Beginn der Eingabedatei an überlesen soll.

NUMBER-OF-RECORDS =

Anzahl der Sätze, die SORT einlesen soll.

NUMBER-OF-RECORDS = *REST-INPUT

SORT soll alle Sätze bis zum Dateiende einlesen.

NUMBER-OF-RECORDS = <integer 1..2147483639>

Angabe der Satzanzahl.

NUMBER-OF-RECORDS = <alphanum-name 1..19>

Angabe der Satzanzahl, sofern sie die Größe von 2.14.483.639 überschreitet.

CHECKPOINT =

Steuerung der Ausgabe von Fixpunkten.

CHECKPOINT = *NO

SORT schreibt keine Fixpunkte.

CHECKPOINT = *YES

SORT soll nach jeweils so vielen Sätzen, wie im RECORDS-PER-CYCLE-Operanden angegeben wurden, jeweils einen Fixpunkt schreiben.

KEEP-EQUAL-SEQUENCES =

Beibehaltung der Reihenfolge von Sätzen mit gleichem Sortierbegriff.

KEEP-EQUAL-SEQUENCES = *NO

Die Eingabereihenfolge wird nicht beibehalten. Die Reihenfolge, in der Sätze mit gleichen Sortierbegriffen in die Ausgabe übernommen werden, ist zufällig.

KEEP-EQUAL-SEQUENCES = *YES

Die Eingabereihenfolge von Sätzen mit gleichen Sortierbegriffen wird beibehalten.

SUM-RECORDS

Funktion: Mit SUM-RECORDS legt der Benutzer Summenfelder fest. Dadurch werden Sätze mit demselben Sortierbegriff zu einem Satz zusammengefasst und die mit SUM-RECORDS ausgewählten Summenfelder addiert. Die Summierung unterbleibt, wenn dabei ein arithmetischer Überlauf auftreten würde. SORT gibt in diesem Fall eine Warnung aus. Sätze mit nur einmal vorkommendem Sortierbegriff werden nicht verändert. Sätze mit nicht addierbaren Summenfeldern bleiben bestehen.

SUM-RECORDS

```

FIELDS = *NONE / list-poss(64): *FIELD-EXPLICIT(...) / *FIELD-SYMBOLIC(...)

*FIELD-EXPLICIT(...)
  | POSITION = <integer 1..4096>
  | ,LENGTH = <integer 1..16>
  | ,FORMAT = *FIXED-POINT / *BINARY / *PACKED-DECIMAL / *ZONED-DECIMAL
  | ,FIELD-EXTENSION = 0 / <integer 0..16>
  | ,PRINT-MASK = *NO / <c-string 1..254 with-low>

*FIELD-SYMBOLIC(...)
  | SUM-FIELD-NAME = list-poss(64): <name 1..20>
  | ,FIELD-EXTENSION = 0 / <integer 0..16>
  | ,PRINT-MASK-NAME = *NO / <name 1..20>

```

FIELDS =

Im FIELDS-Operanden können bis zu 64 Summenfelder angegeben werden. Bei zwei Sätzen mit gleichen Sortierbegriffen werden die Summenfelder addiert und einer der beiden Sätze eliminiert. Summenfelder dürfen sich nicht mit anderen Summenfeldern oder Sortierfeldern überschneiden.

FIELDS = *NONE

Von allen Sätzen mit gleichen Sortierbegriffen wird ein Satz in die Ausgabe übernommen, die anderen werden eliminiert. Es erfolgt keine Summierung. Die Auswahl, welcher Satz übernommen wird, ist rein zufällig.

FIELDS = *FIELD-EXPLICIT(...)

Angabe eines Summenfeldes durch Position, Länge und Format.

POSITION = <integer 1..4096>

Position des Summenfeldes relativ zum Satzanfang.

Bei Sätzen mit festem Satzformat hat das erste Datenfeld die Position 1. Bei Sätzen mit variablem Satzformat hängt die Position des ersten Datenfelds von der Dateiarart und dem Operanden IGNORE-LENGTH-FIELD der Anweisung SET-SORT-OPTIONS ab. Diese Abhängigkeit wird in folgender Tabelle dargestellt:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Das Satzlängenfeld ist nicht als Summenfeld erlaubt. Beim Auswahl- und Adresslisten-sortieren beziehen sich die Positionsangaben auf den neu gebildeten Auswahl Satz (einschließlich eventueller Formatänderungen).

LENGTH = <integer 1..16>

Länge des Summenfeldes in Byte. Die zulässige bzw. Maximallänge ist durch das Format bestimmt.

FORMAT = *FIXED-POINT / *BINARY / *PACKED-DECIMAL / *ZONED-DECIMAL

Format des Summenfeldes. Beim Format ZONED-DECIMAL werden Leerzeichen (X'40') automatisch in Nullen (X'F0') umgewandelt. Außerdem wird bei positiven Zahlen die Vorzeichenzone der letzten Ziffernstelle auf X'Fx' gesetzt ($0 \leq x \leq 9$).

Format	Formatbezeichnung	Länge in Byte
BINARY	binär	2, 4, 8
FIXED-POINT	Festpunkt	2, 4, 8
PACKED-DECIMAL	dezimal gepackt	1 - 16
ZONED-DECIMAL	dezimal entpackt	1 - 16

FIELD-EXTENSION = Q / <integer 0..16>

Angabe einer Summenfeldverlängerung. Das Summenfeld wird um die angegebene Anzahl von Byte nach vorn (linksbündig) verlängert, sodass auch Sätze, bei denen die Addition der Summenfelder zu einem Überlauf führen würde, zusammengefasst werden können. Die Auffüllung der verlängerten Summenfelder erfolgt in Abhängigkeit vom Format.

Auch für verlängerte Summenfelder sind nur die formatspezifischen Längen bzw. Maximallängen zugelassen.

Format	Auffüllung	Füllzeichen
BINARY	linksbündig	X'00' (Null)
FIXED-POINT	linksbündig	Vorzeichen
PACKED-DECIMAL	linksbündig	Null
ZONED-DECIMAL	linksbündig	X'F0' (Null)

PRINT-MASK =

Aufbereitungsmaske, mit der das Summenfeld zum Druck aufbereitet wird. Eine Aufbereitungsmaske ist nur beim Auswahlsortieren erlaubt. Die Länge aller Druckaufbereitungsmasken darf 2000 nicht überschreiten, wobei für jede Maske zusätzlich ein Byte für ein Längenfeld zu berücksichtigen ist.

PRINT-MASK = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK = <c-string 1..254_with-low>

Format der Aufbereitungsmaske. Als Maskenzeichen sind folgende Zeichen möglich:

- Ein frei wählbares Füllzeichen als erstes Zeichen der Maske.
- Die Steuerzeichen '#' (Nummernzeichen) für Ziffernauswahl und '^' (Circumflex) für Ziffernbeginn.
- Einzufügende Zeichen (ungleich den Steuerzeichen).

FIELDS = *FIELD-SYMBOLIC(...)

Angabe von Summenfeldern durch symbolische Namen, die vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Felder vereinbart werden müssen.

SUM-FIELD-NAME = list-poss(64): <name 1..20>

Name des Summenfeldes. Es dürfen maximal 64 Summenfelder angegeben werden. Wird eine Liste von symbolischen Namen angegeben, so gelten die bei FIELD-EXTENSION und PRINT-MASK-NAME gemachten Angaben für alle Namen.

FIELD-EXTENSION = 0 / <integer 0..16>

Angabe einer Summenfeldverlängerung. Das Summenfeld wird um die angegebene Anzahl von Byte nach vorn (linksbündig) verlängert, sodass auch Sätze, bei denen die Addition der Summenfelder zu einem Überlauf führen würde, zusammengefasst werden können. Die Auffüllung der verlängerten Summenfelder erfolgt in Abhängigkeit vom Format.

PRINT-MASK-NAME =

Angabe einer Druckaufbereitungsmaske durch einen symbolischen Namen, der vorher mit der ADD-SYMBOLIC-NAMES-Anweisung als Maske vereinbart werden muss. Eine Aufbereitungsmaske ist nur beim Auswahlsortieren erlaubt.

PRINT-MASK-NAME = *NO

Es wird keine Aufbereitungsmaske angegeben.

PRINT-MASK-NAME = <name 1..20>

Name der Aufbereitungsmaske.

5 Aufruf von SORT

5.1 Aufruf von SORT als autonomes Programm

Der Aufruf von SORT kann durch eine der folgenden Anweisungen erfolgen:

```
/START-SORT  
/SORT-FILE
```

Im Dialogbetrieb wird das Start-Kommando direkt über die Datensichtstation eingegeben oder steht in einer Prozedur.

Im Stapelbetrieb steht das Kommando in einer ENTER-Prozedur.

Die SORT-Steueranweisungen werden von der Systemdatei SYSDDTA gelesen.

START-SORT

Anwendungsbereich: FILE, UTILITIES

Mit dem Kommando START-SORT wird SORT als autonomes Programm aufgerufen. Zur weiteren Bearbeitung sind SORT-Steueranweisungen nötig.

SORT liefert nach Beendigung einen Kommandoreturncode (siehe [Abschnitt „Kommando-returncodes“ auf Seite 206](#)).

START-SORT

```
VERSION = *STD / <product-version 3..8 without-man>  
,CPU-LIMIT = *JOB-REST / <integer 1..32767>  
,MONJV = *NONE / <filename 1..54 without-gen-vers>  
,PROGRAM-MODE = 24 / *ANY / *DBL-DEFAULT
```

VERSION =

Produktversion des SORT, die gestartet werden soll (siehe auch Abschnitt [„Koexistenz“ auf Seite 289](#)).

VERSION = *STD

Keine explizite Angabe der Produktversion. In diesem Fall wird die Produktversion folgendermaßen ausgewählt:

1. Die mit dem Kommando /SELECT-PRODUCT-VERSION vorgegebene Version.
2. Die höchste mit IMON installierte SORT-Version.

VERSION = <product-version 3..8 without-man>

Explizite Angabe der Produktversion in der Form n.n oder nn.n (wobei n für eine Ziffer steht, z.B. 07.9). Das Voranstellen des Zeichens V ist möglich, ebenso das Einschließen in Hochkommata (ggf. mit vorangestelltem Zeichen C, z.B. C'V07.9').

Aus Kompatibilitätsgründen ist auch die Angabe des Freigabe- und Korrekturstands erlaubt. Diese Angabe wird jedoch nicht ausgewertet, da Koexistenz und Austauschbarkeit nur für Hauptversionen zugelassen sind.

CPU-LIMIT =

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf. Überschreitet das Programm beim Ablauf diese Zeit, wird der Sortier-/Mischlauf abgebrochen (Meldung SRT1038).

CPU-LIMIT = *JOB-REST

Ist im SET-LOGON-PARAMETERS-Kommando der Operand CPU-LIMIT=*NO angegeben, so gibt es keine Zeitbeschränkung für das Programm. Ist im SET-LOGON-PARAMETERS-Kommando der Operand CPU-LIMIT=<integer 1..32767> angegeben, so wird als Zeitbeschränkung für das Programm der bei der Systemgenerierung festgelegte Wert verwendet.

CPU-LIMIT = <integer 1..32767>

Angabe der CPU-Zeit in Sekunden.

MONJV =

Angabe einer Jobvariablen. Dieser Operand ist nur mit dem Software-Produkt JV möglich.

MONJV = *NONE

Keine Angabe einer Jobvariablen.

MONJV = <filename 1..54>

Name der Jobvariablen, die das Programm überwachen soll. Ist diese Jobvariable noch nicht vorhanden, wird sie neu angelegt. Während des Programmablaufes setzt dann das System die Jobvariable auf die entsprechenden Werte:

\$R Programm läuft

\$T Programm beendet

\$A Programm abnormal beendet.

Nach Beendigung des SORT-Laufs enthält die Jobvariable zusätzlich Information über den Grund der Beendigung. Den genauen Aufbau der Jobvariablen finden Sie im [Abschnitt „Fehlerinformation beim Aufruf von SORT als autonomes Programm“ auf Seite 406](#).

PROGRAM-MODE =

Dieser Operand bestimmt, ob SORT in den unteren oder in den oberen Adressraum (> 16 MB) geladen wird.

PROGRAM-MODE = 24

SORT wird unterhalb der 16-MB-Grenze geladen. Die Programmausführung erfolgt im 24-Bit-Adressierungsmodus. Extern-Verweise werden als 24-Bit-Adressen interpretiert.

PROGRAM-MODE = *ANY

SORT kann sowohl unterhalb als auch oberhalb der 16-MB-Grenze liegen.

PROGRAM-MODE = *DBL-DEFAULT

Es gilt die Einstellung des letzten MODIFY-DBL-DEFAULT-Kommandos. Bei Task-Beginn ist *ANY voreingestellt.

SORT-FILE

Anwendungsbereich: FILE, UTILITIES

Mit dem Kommando SORT-FILE ruft der Benutzer SORT auf. Gleichzeitig vereinbart er für einen Sortierlauf Ein- und Ausgabedateien, sowie Sortierfelder und Anweisungsdateien. Deshalb werden keine SORT-Steueranweisungen angefordert. Die Sortierung erfolgt mit den bei SORT-FILE festgelegten Angaben. Werden zusätzliche Anweisungen benötigt, müssen diese in Anweisungsdateien zusammengefasst und mit dem Operanden STATEMENT-FILES zugewiesen werden.

Hinweis

Dieses Kommando wird durch eine Prozedur abgearbeitet. Die Operanden des Kommandos werden jeweils als ein Parameter übergeben. Der Operand

```
FIELDS=FIELD-EXPLICIT(POSITION=100,LENGTH=10,FORMAT=CHARACTER, -  
                        SORTING-ORDER=ASCENDING)
```

wird beispielsweise wie folgt an die Prozedur weitergegeben:

```
FIELDS=F(100,10,CH,A,,,,*NO)
```

Ein Parameter darf 254 Zeichen lang sein. Wird diese Beschränkung überschritten (z.B. durch Angabe von zu vielen Sortierfeldern oder zu vielen Anweisungsdateien), wird die Meldung

```
SSM2054 SYMBOLIC OPERAND ERROR IN COMMAND. COMMAND IGNORED
```

ausgegeben. Das Kommando ist zu vereinfachen, oder es ist das Kommando START-SORT zu verwenden.

SORT-FILE wird immer mit
PROGRAM-MODE=*ANY,
MONJV=*NONE und
CPU-LIMIT=*JOB-REST
ausgeführt.

SORT-FILE

```

INPUT-FILES = *LINK / list-poss(99): <filename 1..54> / <posix-pathname 1..1023>
, OUTPUT-FILE = *LINK / <filename 1..54> / <posix-pathname 1..1023>
, CODE = *EBCDIC / *ASCII
, FIELDS = *COMPLETE-RECORD / list-poss(64): *FIELD-EXPLICIT(...) / *FIELD-SYMBOLIC(...)
  *FIELD-EXPLICIT(...)
    |
    | POSITION = <integer 1..32759>
    | , LENGTH = <integer 1..32759>
    | , FORMAT = *CHARACTER / *FIXED-POINT / *FLOATING-POINT / *PACKED-DECIMAL /
    |   *ZONED-DECIMAL / *EBCDIC-DIN / *EBCDIC-INTERNATIONAL /
    |   *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE / *MODIFY-CODE /
    |   *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER / *TRANSLATE-CHARACTER /
    |   *UNICODE-CHARACTER
    | , SORTING-ORDER = *ASCENDING / *DESCENDING
    | , TWO-DIGIT-YEAR = *NO / *YES
  *FIELD-SYMBOLIC(...)
    |
    | SORT-FIELD-NAME = list-poss(64): <name 1..20>
    | , SORTING-ORDER = *ASCENDING / *DESCENDING
    | , TWO-DIGIT-YEAR = *NO / *YES
, IGNORE-LENGTH-FIELD = *STD / *YES / *NO
, STATEMENT-FILES = *NONE / list-poss(10): <filename 1..54>
, VERSION = *STD / <product-version 3..8 without-man>
, CENTURY-WINDOW-SHIFT = 50(...) / <integer 0..99> (...)
  <integer 0..99>(…)
    |
    | WINDOW-BOUNDARY = *SUBTRACT-FROM-CURRENT-YEAR / *ADD-TO-CURRENT-YEAR

```

INPUT-FILES =

Zuordnung der Eingabedatei(en) (max. 99).

INPUT-FILES = *LINK

Die Eingabedateien sind über ADD-FILE-LINK-Kommandos zugewiesen (Dateikettungsname SORTIN oder SORTINxx).

Standardmäßig wird der Dateikettungsname bei Beendigung von SORT wieder freigegeben. Bei Angabe des Kommandos LOCK-FILE-LINK vor dem Aufruf von SORT bleibt der Dateikettungsname erhalten.

INPUT-FILES = list-poss(99): <filename 1..54> / <posix-pathname 1..1023>

Name(n) der Eingabedatei(en). Über ADD-FILE-LINK-Kommando zugewiesene Eingabedateien werden ignoriert. Maximal dürfen 99 Eingabedateien angegeben werden.

POSIX-Dateinamen müssen zur Unterscheidung von BS2000-Dateinamen in Hochkommata angegeben werden.

POSIX-Eingabedateien und BS2000-Eingabedateien dürfen in einem Sortierlauf nicht gleichzeitig verwendet werden. Falls eine Eingabedatei eine POSIX-Datei ist, müssen alle anderen Eingabedateien ebenfalls POSIX-Dateien sein.

Einschränkung:

Die Summe aus der Anzahl und den Längen aller angegebenen POSIX-Eingabedateinamen darf nicht größer als 5100 sein. Darüber hinausgehende Angaben führen zu einer Fehlermeldung mit anschließendem Abbruch des SORT.

OUTPUT-FILE =

Zuordnung der Ausgabedatei.

OUTPUT-FILE = *LINK

Die Ausgabedatei ist über ADD-FILE-LINK-Kommando zugewiesen (Dateikettungsname SORTOUT).

OUTPUT-FILE = <filename 1..54> / <posix-pathname 1..1023>

Name der Ausgabedatei. Eine über ADD-FILE-LINK-Kommando zugewiesene Ausgabedatei wird ignoriert. Eine Veränderung der Dateiattribute gegenüber den aus der Eingabedatei übernommenen oder von SORT festgelegten kann nur durch Zuordnung über ADD-FILE-LINK-Kommando erfolgen.

Der Dateikettungsname SORTOUT wird verwendet. Dieser bleibt nach dem Ende des Sortierlaufs erhalten.

Ein POSIX-Dateiname muss zur Unterscheidung von einem BS2000-Dateinamen in Hochkommata angegeben werden.

CODE =

Angabe des Codes von POSIX-Dateien. Er gilt sowohl für die Eingabedatei als auch für die Ausgabedatei und legt lediglich die Verschlüsselung des Satzende-Kennzeichens fest.

Dieser Operand hat insbesondere **keinen** Einfluss auf die Sortierreihenfolge. Für BS2000-Dateien hat der Operand keine Bedeutung und wird ignoriert.

CODE = *EBCDIC

Das Satzende-Kennzeichen wird nach EBCDIC verschlüsselt und hat den Wert X'15'.

CODE = *ASCII

Das Satzende-Kennzeichen wird nach ASCII verschlüsselt und hat den Wert X'0A'.

FIELDS =

Angabe, ob die Sortierung nach dem gesamten Eingabesatz oder nach einzelnen Sortierfeldern erfolgen soll. Beschreibung von Sortierfeldern bezüglich Position, Länge, Format und Sortierfolge. Mindestens ein Feld muss, maximal 64 Felder dürfen explizit oder über ihre symbolischen Namen angegeben werden.

FIELDS = *COMPLETE-RECORD

Bei Eingabesätzen fester Länge wird der ganze Satz als Sortierfeld genommen. Bei variablen Sätzen ist das Satzlängelfeld nicht Teil des Sortierfeldes. Beim Vergleich unterschiedlich langer Sätze wird der kürzere Satz bis zur Länge des längeren Satzes durch Auffüllen mit dem in der SET-RECORD-ATTRIBUTES-Anweisung, Operand FILLER, angegebenen Wert (Standard X'00') aufgefüllt.

FIELDS = *FIELD-EXPLICIT(...)

Angabe von Sortierfeldern durch Position, Länge, Format und Sortierreihenfolge.

POSITION = <integer 1..32759>

Position des Sortierfeldes relativ zum Satzanfang.

Bei Sätzen mit festem Satzformat hat das erste Datenfeld die Position 1. Bei Sätzen mit variablem Satzformat hängt diese Position von der Dateiart und dem Operanden IGNORE-LENGTH-FIELD ab. Die Position des ersten Datenfelds in diesen Fällen können Sie folgender Tabelle entnehmen:

IGNORE-LENGTH-FIELD=	*STD	*YES	*NO
BS2000-Datei	5	1	5
POSIX-Datei	1	1	5

Bei allen Formaten außer CHARACTER, EBCDIC-DIN, EBCDIC-INTERNATIONAL, EXTENDED-CHARACTER und TRANSLATE-CHARACTER müssen die Positionsangaben für Sortierfelder im Bereich 1 bis 4096 liegen.

LENGTH = <integer 1..32759>

Länge des Sortierfeldes in Byte. Die Länge muss innerhalb der für das Format erlaubten Länge liegen.

FORMAT = *CHARACTER / *FIXED-POINT / *FLOATING-POINT / *PACKED-DECIMAL / *ZONED-DECIMAL / *EBCDIC-DIN / *EBCDIC-INTERNATIONAL / *PHYSICAL-TRANSLATE / *VIRTUAL-TRANSLATE / *MODIFY-CODE / *EBCDIC-ISO-EBCDIC / *EXTENDED-CHARACTER / *TRANSLATE-CHARACTER / *UNICODE-CHARACTER

Format des Sortierfeldes (Eigenschaften siehe [Abschnitt „Sortierfelder“ auf Seite 38](#)).

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fvj'

Dabei sind: j = eine Stelle der zweistelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

FIELDS = *FIELD-SYMBOLIC(...)

Angabe von Sortierfeldern durch symbolische Namen, die mit der ADD-SYMBOLIC-NAMES-Anweisung (in einer Anweisungsdatei) als Felder vereinbart werden müssen.

SORT-FIELD-NAME = list-poss(64): <name 1..20>

Name des Sortierfeldes. Wird eine Liste von symbolischen Namen angegeben, so gelten die bei SORTING-ORDER gemachten Angaben für alle Namen.

SORTING-ORDER =

Reihenfolge, in der SORT die Sätze ordnen soll.

SORTING-ORDER = *ASCENDING

Sortierreihenfolge aufsteigend.

SORTING-ORDER = *DESCENDING

Sortierreihenfolge absteigend. Diese Angabe ist bei ISAM-Ausgabedateien nicht zulässig.

TWO-DIGIT-YEAR =

Legt fest, ob das Sortierfeld eine zweistellige Jahreszahl enthält.

TWO-DIGIT-YEAR = *NO

Das Sortierfeld enthält keine zweistellige Jahreszahl.

TWO-DIGIT-YEAR = *YES

Das Sortierfeld enthält eine zweistellige Jahreszahl. Diese Angabe ist nur zusammen mit FORMAT=*PACKED-DECIMAL oder FORMAT=*ZONED-DECIMAL erlaubt. Die Sortierreihenfolge hängt in diesem Fall vom Operanden CENTURY-WINDOW-SHIFT ab.

Folgende Sortierfeldformate sind möglich:

FORMAT=	Sortierfeldformat
*PACKED-DECIMAL	X'0jv' und X'0jjmmtv'
*ZONED-DECIMAL	X'Fvj'

Dabei sind: j = eine Stelle der zweistelligen Jahreszahl, mm = Monat, tt = Tag, v = Vorzeichen (zwingend positiv)

IGNORE-LENGTH-FIELD =

Angabe, ob bei variabler Satzlänge das Satzlängenfeld bei Positionsangaben für Steuerfelder berücksichtigt werden muss.

Die Angabe gilt für alle Steuerfelder im aktuellen Sortierlauf. Auf Positionsangaben in Sätzen fester Länge hat sie keinen Einfluss.

IGNORE-LENGTH-FIELD = *STD

Bei BS2000-Dateien wirkt *STD wie *NO, d.h. das Satzlängenfeld muss bei Positionsangaben berücksichtigt werden. Das Verhalten ist demnach wie in den Vorversionen.

Bei POSIX-Dateien wirkt *STD wie *YES. Das Satzlängenfeld, das von SORT intern angelegt wird, wird also bei der Positionsberechnung ignoriert.

IGNORE-LENGTH-FIELD = *YES

Das Satzlängenfeld wird bei der Positionsberechnung ignoriert. Das erste Datenfeld eines Satzes beginnt auf Position 1.

IGNORE-LENGTH-FIELD = *NO

Das Satzlängenfeld wird bei der Berechnung berücksichtigt. Das erste Datenfeld eines Satzes beginnt auf Position 5.

STATEMENT-FILES =

Zuweisung einer oder mehrerer Dateien (max. 10), die SORT-Anweisungen enthalten. Die Anweisungsdateien werden vor der Sortieranweisung abgearbeitet.

STATEMENT-FILES = *NONE

Keine Angabe von Anweisungsdateien.

STATEMENT-FILES = list-poss:(10)<filename 1..54>

Namen der Anweisungsdateien.

VERSION =

Produktversion des SORT, die gestartet werden soll (siehe auch Abschnitt „Koexistenz“ auf [Seite 289](#)).

VERSION = *STD

Keine explizite Angabe der Produktversion. In diesem Fall wird die Produktversion folgendermaßen ausgewählt:

1. Die mit dem Kommando /SELECT-PRODUCT-VERSION vorgegebene Version.
2. Die höchste mit IMON installierte SORT-Version.

VERSION = <product-version 3..8 without-man>

Explizite Angabe der Produktversion in der Form n.n oder nn.n (wobei n für eine Ziffer steht, z.B. 07.9). Das Voranstellen des Zeichens V ist möglich, ebenso das Einschließen in Hochkommata (ggf. mit vorangestelltem Zeichen C, z.B. C'V07.9').

Aus Kompatibilitätsgründen ist auch die Angabe des Freigabe- und Korrekturstands erlaubt. Diese Angabe wird jedoch nicht ausgewertet, da Koexistenz und Austauschbarkeit nur für Hauptversionen zugelassen sind.

CENTURY-WINDOW-SHIFT = 50(...) / <integer 0..99>(…)

Legt fest, wie Felder, die zweistellige Jahreszahlen enthalten (FORMAT=*PACKED-DECIMAL oder *ZONED-DECIMAL und TWO-DIGIT-YEAR=*YES) von SORT interpretiert und sortiert werden. Der Operand CENTURY-WINDOW-SHIFT legt ein Jahrhundertfenster fest, innerhalb dessen auf- oder absteigend sortiert wird. Abhängig vom Operanden WINDOW-BOUNDARY wird dieses Jahrhundertfenster durch seinen Anfang oder durch sein Ende bestimmt.

WINDOW-BOUNDARY =

Legt fest, auf welche Weise Beginn und Ende des Jahrhundertfensters berechnet werden.

WINDOW-BOUNDARY = *SUBTRACT-FROM-CURRENT-YEAR

Der **Beginn** des Jahrhundertfensters wird folgendermaßen berechnet:

<beginn-jh-fenster> = <aktuelles-jahr> - <integer-in-century-window-shift>

Da das Jahrhundertfenster 100 Jahre lang ist, ist sein **Ende** 99 Jahre später, also:

<ende-jh-fenster> = <beginn-jh-fenster> + 99

WINDOW-BOUNDARY = *ADD-TO-CURRENT-YEAR

Das **Ende** des Jahrhundertfensters wird folgendermaßen berechnet:

$\langle \text{ende-jh-fenster} \rangle = \langle \text{aktuelles-jahr} \rangle + \langle \text{integer-in-century-window-shift} \rangle$

Da das Jahrhundertfenster 100 Jahre lang ist, ist sein **Beginn** 99 Jahre früher, also:

$\langle \text{beginn-jh-fenster} \rangle = \langle \text{ende-jh-fenster} \rangle - 99.$

Beispiel:

Das aktuelle Jahr sei 2002 und in der Anweisung SET-SORT-OPTIONS ist CENTURY-WINDOW-SHIFT=55 angegeben. Das Jahrhundertfenster geht demnach von 1947 (=2002-55) bis 2046 (=1947+99).

Für die Sortierung bedeutet das:

Die zweistelligen Jahreszahlen 00 bis 46 werden als 2000 bis 2046 interpretiert und sind somit größer als die Jahreszahlen 47 bis 99, da diese als 1947 bis 1999 interpretiert werden.

Die Sortierreihenfolge ist in diesem Fall (bei aufsteigender Sortierung):

47, 48, ..., 98, 99, 00, 01, ..., 45, 46.

Kommandoreturncodes

SORT liefert einen Kommando-Returncode, der mit SDF-P-Anweisungen zur Steuerung in S-Prozeduren verwendet werden kann. Durch den Kommando-Returncode besteht die Möglichkeit, auf bestimmte Fehlersituationen gezielt zu reagieren.

Der Kommando-Returncode besteht aus drei Teilen:

- dem Maincode, der einem Meldungsschlüssel entspricht, über den mit dem Kommando HELP-MSG-INFORMATION detaillierte Informationen abgefragt werden können
- dem Subcode1 (SC1), der die aufgetretene Fehlersituation in eine Fehlerklasse einordnet, aus der abgeleitet werden kann, wie schwerwiegend ein Fehler ist
- dem Subcode2 (SC2), der Zusatzinformationen (Wert ungleich Null) enthalten kann

Falls SORT als Unterprogramm aufgerufen wurde, wird kein Kommandoreturncode geliefert.

Treten mehrere Warnungen auf, so wird der Kommandoreturncode nur für die letzte Warnung geliefert.

Treten sowohl Warnungen als auch Fehler auf, dann wird der Kommandoreturncode für den Fehler geliefert.

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	0	CMD0001	Normale Beendigung des SORT-Laufs. Es traten weder Fehler noch Warnungen auf.
2	0	SRTxxxx	Normale Beendigung des SORT-Laufs. Es traten keine Fehler, aber Warnungen auf. Der Maincode enthält die Nummer der letzten aufgetretenen Warnungsmeldung. Betroffene SRT-Meldungen (xxxx =): 1034, 1042, 1054, 1055, 1058, 1060, 1065, 1066, 1070, 1082, 1086, 1129, 1133, 1159, 1161, 1162, 1163, 1166, 1167, 1173, 1174, 1179, 1183, 1186, 1189, 1190, 1191, 1192, 1196, 1207, 1219, 1221, 1222, 1227, 1228, 1229, 1230, 1256, 1310

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	1	SRTxxxx	<p>Abnormale Beendigung des SORT-Laufs mit Fehler. Der Maincode enthält die Nummer der Abbruchmeldung. Betroffene SRT-Meldungen (xxxx =): 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1079, 1079, 1080, 1081, 1085, 1088, 1099, 1103*, 1104*, 1105*, 1106*, 1107*, 1108*, 1109*, 1110*, 1111*, 1112*, 1113*, 1114*, 1115*, 1116*, 1117*, 1118*, 1119*, 1120*, 1121*, 1122*, 1123*, 1124*, 1125*, 1126*, 1131*, 1134, 1135, 1136, 1138, 1139, 1140*, 1141*, 1142*, 1143*, 1144*, 1145*, 1146*, 1147*, 1148*, 1155*, 1156*, 1157*, 1158*, 1168*, 1169*, 1170*, 1171, 1172, 1178, 1180, 1181, 1182, 1184*, 1185, 1187*, 1188*, 1193*, 1194*, 1195, 1199*, 1201, 1203, 1204, 1205, 1206, 1209, 1211, 1212, 1213, 1215, 1218, 1220, 1223, 1224, 1225, 1226, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1239, 1240, 1250, 1251, 1253, 1254, 1260, 1261, 1301, 1302, 1303, 1304, 1305, 1307, 1308, 1309, 1311, 1317</p> <p>* Die so gekennzeichneten Meldungen beziehen sich auf fehlerhafte SORT-Steueranweisungen. Falls diese im Dialog eingegeben wurden und durch eine nachfolgende Anweisung korrigiert wurden, wird der Returncode nicht versorgt.</p>
0	32	SRTxxxx	<p>Abnormale Beendigung des SORT-Laufs mit internem Fehler. Der Maincode enthält die Nummer der Abbruchmeldung. Betroffene SRT-Meldungen (xxxx =): 1039, 1069, 1127</p>
0	64	SRTxxxx	<p>Abnormale Beendigung des SORT-Laufs mit sonstigem Fehler. Der Maincode enthält die Nummer der Abbruchmeldung. Betroffene SRT-Meldungen (xxxx =): 1032, 1035, 1036, 1040, 1043, 1044, 1045, 1047, 1048, 1051, 1056, 1057, 1064, 1067, 1068, 1083, 1084, 1087, 1089, 1132, 1137, 1149, 1150, 1197, 1202, 1208, 1214, 1216, 1244, 1252, 1255, 1257, 1258, 1259, 1262, 1263, 1264, 1266, 1306, 1314, 1315, 1316</p>
0	255	SRT1312	<p>Abnormale Beendigung des SORT-Laufs, da die geforderte SORT-Version nicht verfügbar ist.</p>

Im Fehlerfall können die Komponenten des Returncodes mit den SDF-P-Funktionen SUBCODE1(), SUBCODE2() und MAINCODE() abgefragt werden.

Nach einer fehlerfreien Durchführung (SC1 = 0) kann mit dem Kommando SAVE-RETURN-CODE der Returncode sichergestellt und ebenfalls ausgewertet werden. (Nähere Information zu Kommando-Returncodes und zum Abfragen von Returncodes siehe Handbuch „SDF-P“ [13]).

Beispiel zum Abfragen von Returncodes

```

/SORT-FILE EINGABE,AUSGABE _____ (1)
/SAVE-RETURNCODE _____ (2)
/IF-BLOCK-ERROR _____ (3)
/ WRITE-TEXT '-----' _____ (4)
/ WRITE-TEXT '--- SORT-Lauf mit Fehler abgebrochen ---'
/ WRITE-TEXT '-----'
/ELSE
/ IF &SUBCODE2 > 0 _____ (5)
/ WRITE-TEXT '-----' _____ (6)
/ WRITE-TEXT '--- SORT-Lauf mit Warnung beendet ---'
/ WRITE-TEXT '-----'
/ ELSE
/ WRITE-TEXT '-----' _____ (7)
/ WRITE-TEXT '--- SORT-Lauf ohne Warnung oder Fehler beendet ---'
/ WRITE-TEXT '-----'
/ END-IF
/END-IF
/IF &SUBCODE1 > 0 OR &SUBCODE2 > 0 _____ (8)
/ WRITE-TEXT 'Subcode 1: &SUBCODE1'
/ WRITE-TEXT 'Subcode 2: &SUBCODE2'
/ WRITE-TEXT 'Maincode : &MAINCODE'
/ HELP-MSG-INFORMATION &MAINCODE
/END-IF

```


- (1) Start-Kommando für SORT
- (2) Sicherstellen des Returncodes von SORT
(wird nur bei normaler Beendigung des SORT durchlaufen, d.h. wenn der Spin-Off-Mechanismus nicht ausgelöst wurde)
- (3) Prüfung auf SORT-Abbruch (gleichzeitig Wiederaufsetzpunkt in der Prozedur, falls Spin-Off-Mechanismus ausgelöst wurde)
- (4) Ausgabe nach Abbruch des SORT
- (5) Falls der SORT-Lauf nicht abgebrochen wurde, wird geprüft, ob es eine Warnung gab (Subcode 2 größer Null)
- (6) Ausgabe, wenn im SORT-Lauf mindestens eine Warnung, aber kein Fehler aufgetreten ist
- (7) Ausgabe, wenn im SORT-Lauf weder Warnungen noch Fehler aufgetreten sind
- (8) Wenn eine Warnung oder ein Fehler aufgetreten ist (mindestens einer der beiden Subcodes ist größer als 0), werden die drei Komponenten des Returncodes und die entsprechende Meldung ausgegeben.

5.2 Aufruf von SORT als Unterprogramm

Als Unterprogramm wird SORT durch einen Unterprogrammssprung nach SORTU oder ILSORT aufgerufen. Dazu wird im rufenden Modul eine entsprechende V-Adresse verwendet, die auf den SORT-Startermodul SRT80 zeigt. Die Rücksprungadresse muss in Register 14 stehen.

Hinweis

SORT V8.0 kann mit einem Startermodul SRT80 der Versionen SORT V7.3 und neuer aufgerufen werden. Dagegen wird die Ablauffähigkeit alter SORT-Versionen mit einem Startermodul SRT80 der Version SORT V8.0 nicht garantiert.

Beispiel

```
L      15,=V(SORTU)      oder   L      15,=V(ILSORT)
BALR  14,15              BALR  14,15
```

Vor der Ausführung des Programms muss die ungelöste externe Referenz auf den SORT-Startermodul SRT80 befriedigt werden. Das erfolgt im einfachsten Fall mit dem DBL (Dynamischer Bindelader).

Damit der DBL den Startermodul SRT80 finden kann, muss ihm der Name der zugehörigen Bibliothek in einem ADD-FILE-LINK-Kommando mit den LINK-Namen BLSLIBxx mitgeteilt werden.

Wenn SORT unter Einsatz von IMON-GPN installiert wurde, dann ist der Name der Sortierbibliothek frei wählbar. Er muss dann aus dem IMON-SCI (Software-Configuration-Inventary), der zentralen Datenbasis von IMON ermittelt werden. Dazu bietet sich die Nutzung von SDF-P-Prozeduren an.

Mit dem Kommando

```
/SET-VARIABLE SORTLIB=INSTALLATION-PATH -
/              (LOGICAL-ID           = 'SYSLNK' -
/              ,INSTALLATION-UNIT = 'SORT' -
/              ,VERSION           = '*STD' -
/              ,DEFAULT-PATH-NAME = '$.SORTLIB')
```

wird der Variablen SORTLIB der Name der aktuellen Sortierbibliothek zugewiesen.

Im anschließenden Kommando

```
/ADD-FILE-LINK LINK-NAME=BLSLIBxx, FILE-NAME=&SORTLIB
```

wird der auf diese Weise ermittelte Name verwendet.

Der folgende Ausschnitt aus einer Beispielprozedur zeigt, wie nach dem erfolgreichen Übersetzen des Hauptprogramms die erzeugte *OMF-Datei aufgerufen werden kann.

```

      :
      Hauptprogramm übersetzen
      :
/SET-VARIABLE SORTLIB = '$.SORTLIB'
/IF (SDF-P-VERSION >= 'V02.0A00')
/ SET-VARIABLE SORTLIB = INSTALLATION-PATH -
/           (LOGICAL-ID           = 'SYSLNK' -
/           ,INSTALLATION-UNIT = 'SORT' -
/           ,VERSION             = '*STD' -
/           ,DEFAULT-PATH-NAME = '&SORTLIB')
/END-IF
/ADD-FILE-LINK LINK-NAME=BLSLIB01,FILE-NAME=&SORTLIB
/START-EXE-PROG FROM-FILE=*OMF
      :

```

Prozeduren für das statische Binden mit BINDER müssen in entsprechender Weise aufgebaut sein.

Für die Angabe der Makro-Bibliothek (bisher \$.SORTMACLIB) in Übersetzungsläufen bietet sich eine analoge Vorgehensweise an.

Wird das Programm mit /START-PROGRAM FROM-FILE=*MODULE(*OMF) oder /EXEC * aufgerufen, gilt RUN-MODE=*STD. In diesem Fall muss die Bibliothek zum Befriedigen von Externverweisen mit /SET-TASKLIB zugewiesen werden.

Registerkonventionen

- Reg 1 Enthält die Adresse des Versorgungsblocks SVB.
- Reg 13 Enthält die Adresse des 18 Wort großen Sicherungsbereichs, in den die Registerinhalte des rufenden Programms gesichert werden. Dieser Sicherungsbereich muss vom Benutzer bereitgestellt werden.
- Reg 14 Rücksprungadresse des aufrufenden Programms.
- Reg 15 Enthält im niedrigstwertigen Byte das Rücksprungkennzeichen, das bei Beendigung des Sortier-/Mischlaufs übergeben wird.

X'00' der Sortier-/Mischlauf endete normal.

X'FF' der Sortier-/Mischlauf endete fehlerhaft. Die Fehler werden über eine Meldung angezeigt.

Im Fehlerfall enthalten die beiden höchstwertigen Byte zusätzlich die letzten vier Stellen des SORT-Meldungsschlüssels. Der SORT-Meldungsschlüssel wird dabei als gepackte Dezimalzahl ohne Vorzeichenzone abgespeichert. Wird beispielsweise ein SORT-Lauf mit der Fehlermeldung SRT1035 fehlerhaft beendet, hat Register 15 den Inhalt X'103500FF'.

Die Übergabe der Daten an SORT muss aus dem Programm-Adressraum erfolgen. Ist im zugehörigen SORT-Makro (SRT0, SRT1) der Operand STXIT=YES angegeben, so prüft SORT die übergebenen Daten und Adressen auf Gültigkeit. Erkannte Programmfehler führen zur Ausgabe einer Meldung (SRT1071 bis SRT1077) und zur Beendigung des Programmlaufes. In diesem Fall muss der Programmfehler beseitigt und der Programmlauf wiederholt werden.

Gültige, aber nicht korrekte Übergabedaten können zu unvorhersehbarem Programmverhalten führen.

5.2.1 Übergabe der Steuerinformation an SORT

Ruft ein Programm SORT als Unterprogramm auf, müssen die Angaben für den Sortier-/Mischlauf mit

- SORT-Anweisungen (siehe [Kapitel „Anweisungen von SORT“ auf Seite 127](#)) oder
- SORT-Makroaufrufen (siehe [Abschnitt „Makroaufrufe für SORT“ auf Seite 216](#)).

übergeben werden. Für die Übergabe an SORT stehen 2 verschiedene Übergabeebenen zur Verfügung.

Ebene 0 SORT erwartet Anweisungen über SYSDTA. Den zugehörigen Versorgungsblock SVB erstellt der Benutzer mit einem SRT0-Makroaufruf (siehe [Abschnitt „Makroaufrufe für SORT“ auf Seite 216](#)). Register 1 verweist dann auf die Adresse des SRT0-Makroaufrufs.

Ebene 1 SORT erwartet Anweisungen im rufenden Programm als Sätze variabler Länge. Die Anweisungen und den zugehörigen Versorgungsblock SVB erstellt der Benutzer mit SRT1-Makroaufrufen (siehe [Abschnitt „Makroaufrufe für SORT“ auf Seite 216](#)). Register 1 verweist auf die Folge der SRT1-Makroaufrufe.

5.2.1.1 Ebene 0

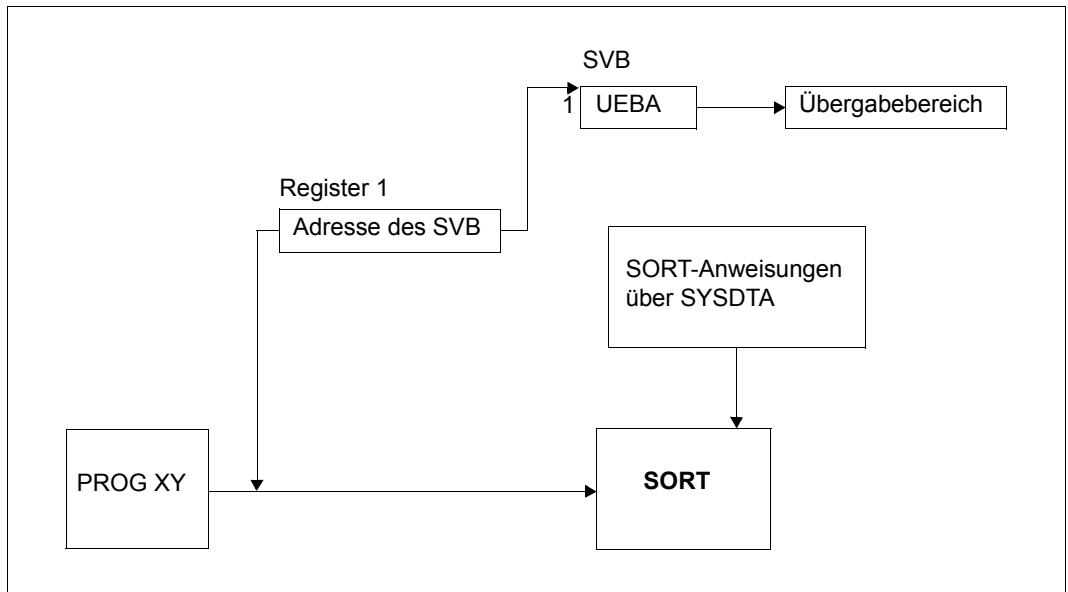


Bild 8: Aufruf von SORT als Unterprogramm über Ebene 0

Über Ebene 0 erwartet SORT die SORT-Anweisungen von SYSDTA. Register 1 muss auf den Versorgungsblock (SVB) verweisen. Den Versorgungsblock SVB kann der Benutzer mit dem SRT0-Makroaufruf erstellen (siehe [Abschnitt „Makroaufrufe für SORT“ auf Seite 216](#)). Der Aufbau des SVB ist im Anhang auf [Seite 412](#) beschrieben.

5.2.1.2 Ebene 1

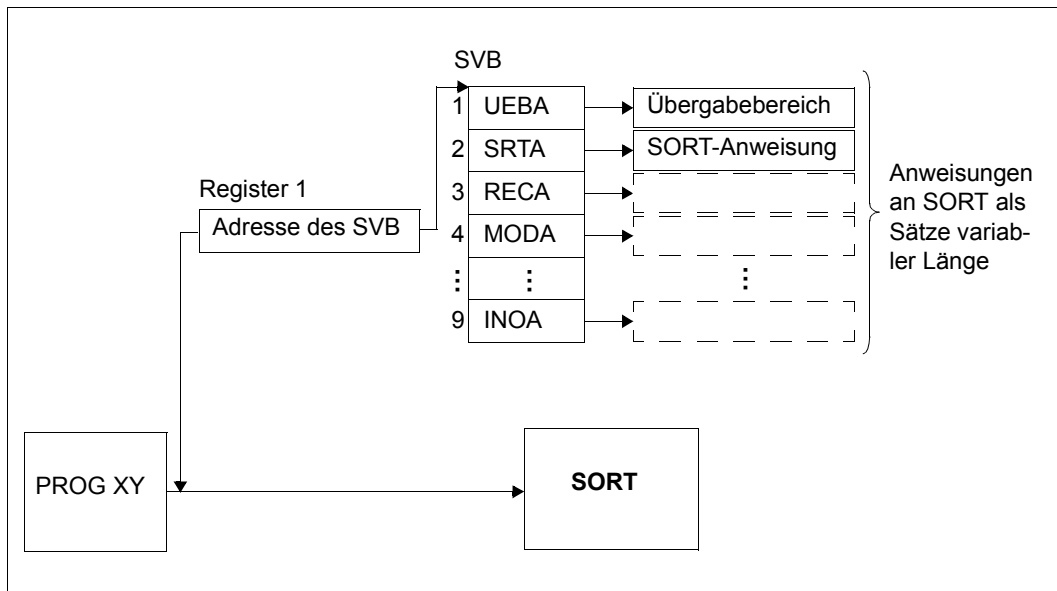


Bild 9: Aufruf von SORT als Unterprogramm über Ebene 1

Werden die SORT-Anweisungen über Ebene 1 übergeben, erwartet SORT diese als Sätze variabler Länge im rufenden Programm. Register 1 muss auf den Versorgungsblock (SVB) verweisen. Die Anweisungen der Ebene 1 erstellt der Benutzer über Makroaufrufe SRT1 (siehe [Abschnitt „Makroaufrufe für SORT“ auf Seite 216](#)). Der Aufbau des Versorgungsblocks SVB ist im Anhang auf [Seite 412](#) beschrieben.

5.2.2 Makroaufrufe für SORT

Für den Aufruf von SORT als Unterprogramm über die Ebenen 0 und 1 stehen folgende Makroaufrufe zur Verfügung:

SRT0 SORT-Aufruf über Ebene 0
SRT1 SORT-Aufruf über Ebene 1

Diese Makroaufrufe erstellen auch den Versorgungsblock SVB einschließlich des Übergabesteuerfeldes.

5.2.2.1 SRT0 – SORT-Aufruf über Ebene 0

Das Sortier-/Mischprogramm SORT verfügt über einen Makro SRT0, mit dem der Benutzer SORT über Ebene 0 versorgen kann.

Name	Operation	Operanden
[name]	SRT0	[SDF = <u>NO</u> / YES] [,STXIT = <u>YES</u> / NO] [,RDONLY = <u>NO</u> / YES] [,RCF = <u>NO</u> / aadr / sadr] [,RCFL = länge] [,MSGPROT = <u>OUT</u> / LST / BOTH / NO] [,VERSION = * <u>STD</u> / version] [,MULTI = <u>STD</u> / OPT / NOIMON]

name	Maximal vier Zeichen langer symbolischer Kurzname. Dieser Name adressiert den Versorgungsblock SVB für die Versorgung über Ebene 0. Die Adresse muss nach Register 1 geladen werden. Der Kurzname wird allen mit dem Makro SRT0 generierten symbolischen Namen als Präfix vorangestellt.
SDF = <u>NO</u> / YES	Damit SDF verwendet wird, ist SDF=YES explizit anzugeben. Wird nichts oder NO angegeben, müssen nachfolgende Eingaben im ISP-Format getätigt werden
STXIT = <u>YES</u> / NO	Angabe zur STXIT-Benutzung (siehe „Makroaufrufe an den Ablaufteil“ [6]).
<u>YES</u>	SORT verwendet die STXIT-Eingänge selbst.
NO	SORT verwendet keine STXIT-Eingänge. STXIT=NO bewirkt Folgendes: <ul style="list-style-type: none"> – der Benutzer kann nicht mit dem Kommando SEND-MSG in den SORT-Lauf eingreifen – SORT gibt im Fehlerfall keinen Spezial-DUMP aus

- für Benutzerdaten und -adressen findet keine Prüfung auf Gültigkeit statt.
- RDONLY = YES / NO** Angabe, ob die SRT0-Makros in einem schreibgeschützten oder nicht schreibgeschützten Modul stehen.
- YES** SRT0-Makros stehen in einem schreibgeschützten READONLY-Modul (z.B. gemeinsam benutzbar).
- NO** SRT0-Makros stehen in einem nicht schreibgeschützten Modul.
- RCF = aadr / sadr / NO**
 Angabe eines Bereichs für die Ablage des Rückkehrcodes und der SORT-Meldungen. In den ersten vier Byte des RCF-Bereichs ist das Rückkehrkennzeichen RC abgelegt (siehe Anhang, [Seite 409](#)). Anschließend sind die SORT-Meldungen, die durch den Operanden MIN-MSG-WEIGHT der SET-SORT-OPTIONS-Anweisung ausgegeben werden, als Sätze variabler Länge hintereinander abgelegt (entsprechend einer Ausgabe auf SYSOUT/SYSLST).
- aadr** Symbolische Adresse des Bereichs.
Beispiel
 RCF=BEREICH
- sadr** Symbolische Adresse im S-Adressenformat des Bereichs.
 „sadr“ ist wie bei Assemblerschreibweise in runde Klammern zu setzen, jedoch ohne vorangestelltes S.
Beispiel
 RCF=(BEREICH)
 RCF=(DISTANZ(REG))
 RCF=(0(5))
- NO** Es wird kein Bereich für die Ablage des Rückkehrcodes und der SORT-Meldungen benutzt.
- RCFL = länge** Legt die Länge des angegebenen RCF-Bereichs fest. Die maximale Länge ist 32767 Byte. Ist kein Operand RCFL bzw. für RCFL ein Wert kleiner als 4 angegeben, wird eine Länge von vier Byte angenommen.
- MSGPROT = OUT / LST / BOTH / NO**
 Angabe, ob SORT-Meldungen auf SYSOUT und/oder SYSLST ausgegeben werden. Ist der Operand RCF angegeben, werden die SORT-Meldungen zusätzlich in den RCF-Bereich ausgegeben.

<u>OUT</u>	SORT-Meldungen auf SYSOUT und in den RCF-Bereich ausgeben.
LST	SORT-Meldungen auf SYSLST und in den RCF-Bereich ausgeben.
BOTH	SORT-Meldungen auf SYSOUT und SYSLST und in den RCF-Bereich ausgeben.
NO	Keine Ausgabe der SORT-Meldungen auf SYSOUT und SYSLST. Ausgabe nur in den RCF-Bereich.
VERSION = <u>*STD</u> / version	Angabe der Version von SORT, die als Unterprogramm nachgeladen werden soll. Die Auswertung der Versionsangabe entspricht der beim Hauptprogrammaufruf.
<u>*STD</u>	Es wird keine spezielle SORT-Version angefordert.
version	Versionsbezeichnung im Format [v]v.v[a[nn]] [v]v.v : Produktversion a : Freigabestand nn : Korrekturstand – Die zusätzliche Angabe von 'V' und Hochkommata wie beim Start-Kommando ist nicht möglich. – Freigabe- und Korrekturstand können zwar angegeben werden, sie werden aber nicht ausgewertet, da Koexistenz und Austauschbarkeit nur für Hauptversionen zugelassen sind.
MULTI = <u>STD</u> / OPT / NOIMON	Angabe, ob bei jedem Sortierlauf die Dateinamen ermittelt und die Parameterdatei SYSPAR gelesen werden sollen. Wird SORT häufig (öfters als 1000 mal) als Unterprogramm aufgerufen, führt eine Reduzierung der auszuführenden Operationen zu einer Optimierung der Laufzeit.
<u>STD</u>	Alle vorbereitenden Operationen werden ausgeführt. Die Parameterdatei SYSPAR wird zu Beginn jedes Sortierlaufs gelesen.
OPT	Die Dateinamen werden mittels IMON nur beim ersten Unterprogrammaufruf von SORT ermittelt. Die Parameterdatei SYSPAR wird nur zu Beginn des ersten Sortierlaufs gelesen.

NOIMON Die Dateinamen werden nicht mittels IMON ermittelt. Es gelten die Standardnamen (z.B. SYSLNK.SORT.ver). Die Parameterdatei SYSPAR wird einmalig beim ersten Aufruf gelesen. Deren Werte werden dann zur Generierung der SORT-Standard-Parameter eingesetzt.

Hinweis

Wird während eines Sortierlaufs bei der Einstellung MULTI= OPT oder MULTI=NOIMON die Anweisung MODIFY-SORT-DEFAULTS angegeben, so gelten für diesen Sortierlauf die dort eingestellten Parameter. Diese Parameter können mit SAVE-DEFAULTS=*YES in die Datei SYSPAR gespeichert werden. Innerhalb des gleichen Hauptprogrammaufrufs werden die Änderungen jedoch für die nächsten Aufrufe eines Sortierunterprogramms nicht wirksam.

5.2.2.2 SRT1 – SORT-Aufruf über Ebene 1

Das Sortier-/Mischprogramm verfügt über einen Makro SRT1, über den der Benutzer die vollständige Versorgung von SORT über Ebene 1, einschließlich des Versorgungsblockes SVB und des Übergabesteuerfeldes erstellen kann.

Für jede Anweisung ist ein SRT1-Makroaufruf erforderlich. SORT generiert für den Versorgungsblock und für jede angegebene Anweisung symbolische Namen, die sich aus dem Kurznamen und den ersten zwei oder drei Zeichen der Anweisung zusammensetzen. Sie erscheinen in der Makroauflistung, falls „PRINT GEN“ angegeben wurde. Soll der Inhalt einer Anweisung geändert werden, kann dies über ihren symbolischen Namen durchgeführt werden.

Name	Operation	Operanden
[name]	SRT1	(anweisungstext) [SDF = <u>NO</u> / YES] [,STXIT = <u>YES</u> / NO] [,RDONLY = <u>NO</u> / YES] [,RCF = <u>NO</u> / aadr / sadr] [,RCFL = länge] [,MSGPROT = <u>OUT</u> / LST / BOTH / NO] [,VERSION=* <u>STD</u> / version] [,MULTI = <u>STD</u> / OPT / NOIMON]

name Maximal vier Zeichen langer symbolischer Kurzname. Er kennzeichnet eine Gruppe von SRT1-Makroaufrufen, über die die Anweisungen für einen SORT-Lauf erstellt werden, und muss beim ersten SRT1-Makroaufruf angegeben werden. Für alle weiteren SRT1-Makroaufrufe des betreffenden SORT-Laufs darf „name“ nicht mehr angegeben werden.

	Über diesen Namen wird der Versorgungsblock SVB für die Versorgung über Ebene 1 adressiert und damit der Bezug zur Anweisung hergestellt. Die Adresse muss nach Register 1 geladen werden. Der Kurzname wird allen von den Makros SRT1 generierten symbolischen Namen als Präfix vorangestellt.
anweisungstext	Hier muss die betreffende SORT-Anweisung in Klammern eingeschlossen angegeben werden. Das Format ist das gleiche wie bei Versorgung über Ebene 0. Verkürzte Schlüsselwörter und Stellungsooperanden sind zulässig.
SDF = YES / <u>NO</u>	Damit SDF verwendet wird, ist SDF=YES explizit anzugeben. Wird nichts oder NO eingegeben, müssen die Anweisungen im ISP-Format angegeben werden.
STXIT = <u>YES</u> / NO	Angabe zur STXIT-Benutzung (siehe „Makroaufrufe an den Ablaufteil“ [6]). Der Operand darf nur beim ersten SRT1-Makroaufruf, der mit „name“ gekennzeichnet ist, angegeben werden.
<u>YES</u>	SORT verwendet die STXIT-Eingänge selbst.
NO	SORT verwendet keine STXIT-Eingänge. STXIT=NO bewirkt Folgendes: <ul style="list-style-type: none"> – der Benutzer kann nicht mit dem Kommando SEND-MSG in den SORT-Lauf eingreifen – SORT gibt im Fehlerfall keinen Spezial-DUMP aus – für Benutzerdaten und -adressen findet keine Prüfung auf Gültigkeit statt.
RONLY = YES / <u>NO</u>	Angabe, ob die SRT1-Makros in einem schreibgeschützten oder nicht schreibgeschützten Modul stehen.
YES	SRT1-Makros stehen in einem schreibgeschützten READONLY-Modul (z.B. gemeinsam benutzbar).
<u>NO</u>	SRT1-Makros stehen in einem nicht schreibgeschützten Modul.
RCF = aadr / sadr / <u>NO</u>	Angabe eines Bereichs für die Ablage des Rückkehrcodes und der SORT-Meldungen. In den ersten vier Byte des RCF-Bereichs ist das Rückkehrzeichen RC abgelegt (siehe Anhang, Seite 409). Anschließend sind die SORT-Meldungen, die durch den Operanden MIN-MSG-WEIGHT der SET-SORT-OPTIONS-Anweisung ausgegeben werden, als Sätze variabler Länge hintereinander abgelegt (entsprechend einer Ausgabe auf SYSOUT/SYSLST).

aadr	Symbolische Adresse des Bereichs. <i>Beispiel</i> RCF=BEREICH
sadr	Symbolische Adresse im S-Adressenformat des Bereichs. „sadr“ ist wie bei Assemblerschreibweise in runde Klammern zu setzen, jedoch ohne vorangestelltes S. <i>Beispiel</i> RCF=(BEREICH) RCF=(DISTANZ(REG)) RCF=(0(5))
<u>NO</u>	Es wird kein Bereich für die Ablage des Rückkehrcodes und der SORT-Meldungen benutzt.
RCFL = länge	Legt die Länge des angegebenen RCF-Bereichs fest. Die maximale Länge ist 32767 Byte. Ist kein Operand RCFL bzw. für RCFL ein Wert kleiner als 4 angegeben, wird eine Länge von vier Byte angenommen.
MSGPROT = <u>OUT</u> / LST / BOTH / NO	Angabe, ob SORT-Meldungen auf SYSOUT und/oder SYSLST ausgegeben werden sollen. Ist der Operand RCF angegeben, werden die SORT-Meldungen zusätzlich in den RCF-Bereich ausgegeben.
<u>OUT</u>	SORT-Meldungen auf SYSOUT und in den RCF-Bereich ausgegeben.
LST	SORT-Meldungen auf SYSLST und in den RCF-Bereich ausgegeben.
BOTH	SORT-Meldungen auf SYSOUT und SYSLST und in den RCF-Bereich ausgegeben.
NO	Keine Ausgabe der SORT-Meldungen auf SYSOUT und SYSLST. Ausgabe nur in den RCF-Bereich.
VERSION = <u>*STD</u> / version	Angabe der Version von SORT, die als Unterprogramm nachgeladen werden soll. Die Auswertung der Versionsangabe entspricht der beim Hauptprogrammaufruf.
<u>*STD</u>	Es wird keine spezielle SORT-Version angefordert.

version	<p>Versionsbezeichnung im Format [v]v.v[a[nn]]</p> <p>[v]v.v : Produktversion a : Freigabestand nn : Korrekturstand</p> <ul style="list-style-type: none"> – Die zusätzliche Angabe von 'V' und Hochkommata wie beim Start-Kommando ist nicht möglich. – Freigabe- und Korrekturstand können zwar angegeben werden, sie werden aber nicht ausgewertet, da Koexistenz und Austauschbarkeit nur für Hauptversionen zugelassen sind.
MULTI = <u>STD</u> / OPT / NOIMON	<p>Angabe, ob bei jedem Sortierlauf die Dateinamen ermittelt und die Parameterdatei SYSPAR gelesen werden soll.</p> <p>Wird SORT häufig (öfters als 1000 mal) als Unterprogramm aufgerufen, führt eine Reduzierung der auszuführenden Operationen zu einer Optimierung der Laufzeit.</p>
<u>STD</u>	Alle vorbereitenden Operationen werden ausgeführt. Die Parameterdatei SYSPAR wird zu Beginn jedes Sortierlaufs gelesen.
OPT	Die Dateinamen werden mittels IMON nur beim ersten Unterprogrammaufruf von SORT ermittelt. Die Parameterdatei SYSPAR wird nur zu Beginn des ersten Sortierlaufs gelesen.
NOIMON	Die Dateinamen werden nicht mittels IMON ermittelt. Es gelten die Standardnamen (z.B. SYSLNK.SORT.ver). Die Parameterdatei SYSPAR wird beim Start gelesen. Für die Parameter gelten die generierten Standards.

Hinweis

Wird während eines Sortierlaufs bei der Einstellung MULTI= OPT oder MULTI=NOIMON die Anweisung MODIFY-SORT-DEFAULTS angegeben, so gelten für diesen Sortierlauf die dort eingestellten Parameter. Diese Parameter können mit SAVE-DEFAULTS=*YES in die Datei SYSPAR gespeichert werden. Innerhalb des gleichen Hauptprogrammaufrufs werden die Änderungen jedoch für die nächsten Aufrufe eines Sortierunterprogramms nicht wirksam.

Konventionen für SRT1-Makroaufrufe

- Jeder Anweisung entspricht ein SRT1-Makroaufruf.
- Der Anweisungstext ist in runde Klammern einzuschließen.
- Verkürzte Schlüsselwörter und Stellungsoperanden sind zulässig.
- Beim ersten SRT1-Makroaufruf muss ein 4 Zeichen langer Name angegeben werden. Die restlichen Makroaufrufe einschließlich SRT1 (END) dürfen nicht mehr mit einem symbolischen Namen gekennzeichnet sein.
- Im ersten SRT1-Makro muss am Ende, außerhalb der den Anweisungstext umschließenden Klammer, der Zusatz `SDF=YES` enthalten sein.
- Für die SRT1-Makroaufrufe müssen ansonsten alle Assemblerkonventionen für Makroaufrufe beachtet werden wie z.B. Fortsetzungszeilen, Längenbegrenzungen pro Operand=127.

Der gesamte Anweisungstext bildet dabei einen Operanden der Makroanweisung. Bei der `SELECT-INPUT-RECORDS`-Anweisung darf vor und nach `AND/OR` das Komma nicht weggelassen werden.

- Bei der Umsetzung des SRT1-Makroaufrufs in DC-Assembleranweisungen wird jedes einzeln stehende Hochkomma (bei Konstantendarstellung) entsprechend Assemblerkonvention verdoppelt. Sollen jedoch auch innerhalb einer Konstanten Hochkommata verwendet werden, sind alle Hochkommata (auch die einschließenden) anzugeben (z.B. `FILLER="''''''`).

Beispiel

```

START
.
.
.
LA      13,SAVE
LA      1,VS1
L       15,=V(SORTU)
BALR   14,15
.
.
.
SAVE   DS      18F
VS1    SRT1    (SORT-RECORDS FIELDS=*FIELD-EXPLICIT(POSITION=30,
                LENGTH=8)),SDF=YES,STXIT=NO
        SRT1    (ASSIGN-EXITS PLANNING=*DIALOG)
        SRT1    (SET-SORT-OPTIONS MIN-MSG-WEIGHT=*ALL)
        SRT1    (END)

```

5.3 SORT-Zugriffsmethode SORTZM

Beim Voll- und Auswahlsortieren kann der Benutzer SORT nicht nur als autonomes Programm (mit dem Kommando /START-SORT) oder als Unterprogramm (über SORTU bzw. ILSORT) aufrufen, sondern auch über eine eigene Zugriffsmethode SORTZM.

5.3.1 Funktion der SORT-Zugriffsmethode SORTZM

Bild 10 zeigt die Wirkungsweise von SORTZM. Die Zugriffsmethode SORTZM übergibt die zu sortierenden Sätze an SORT und übernimmt die sortierten Sätze wieder von SORT. Das Hauptprogramm muss die zu sortierenden Sätze in einem eigenen Satzpuffer bereitstellen. Die Auswahl der Sätze für die Sortierung bleibt dabei dem Benutzer überlassen. Eine Satzauswahl über die Anweisung SELECT-INPUT-RECORDS wird von SORT nicht ausgewertet. Eine Angabe von INPUT-RANGE in der SORT-RECORDS-Anweisung wird ebenso ignoriert wie der Benutzerausgang INPUT.

Die sortierten Sätze gibt SORT in einem eigenen Satzpuffer zurück.

Die Verbindung zwischen Hauptprogramm und SORT wird mit Makroaufrufen hergestellt, die in folgender Reihenfolge anzugeben sind:

SRTOPEN	Sortierung eröffnen
SRTPUT	Satz an SORT übergeben (für jeden Satz anzugeben)
SRTGET	Satz von SORT übernehmen (für jeden Satz anzugeben)
SRTCLSE	Sortierung schließen.

SRTCLSE ist auch bei einer Leersortierung erforderlich. SRTPUT und SRTGET dürfen auch entfallen (Leersortierung). Bei der SRTGET-Schleife darf auch vorzeitig mit SRTCLSE beendet werden.

Die Zugriffsmethode SORTZM kann SORT über Ebene 0 oder Ebene 1 mit den SORT-Anweisungen versorgen (siehe [Seite 213](#)).

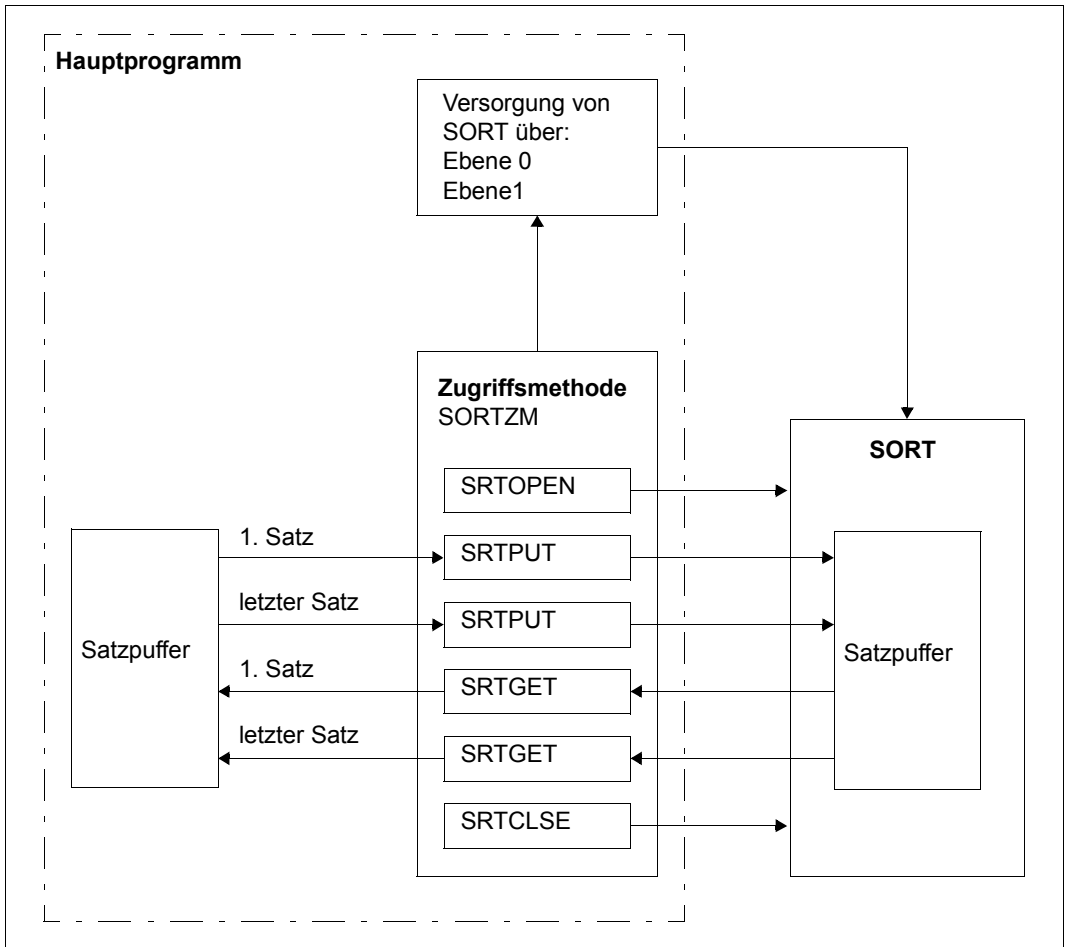


Bild 10: Schnittstellen der Zugriffsmethode SORTZM

Bedingungen zur Anwendung der SORTZM

- Das Mischen von Dateien ist nicht zugelassen.
- Angegebene Ein- und Ausgabedateien und die Benutzerausgänge INPUT und OUTPUT werden ignoriert.
- Mit SORTZM ist nur Voll- und Auswahl-sortieren möglich, kein Adresslistensortieren.
- In der Versorgung muss die SET-RECORD-ATTRIBUTES-Anweisung angegeben werden.
- Die Register 0, 1, 14 und 15 werden von SORTZM verändert.
- Ist im zugehörigen SORT-Makro (SRT0, SRT1) der Operand STXIT=YES angegeben, so prüft SORT die übergebene Satzadresse und -länge auf Gültigkeit. Im Fehlerfall wird die Meldung SRT1079 ausgegeben und der Programmablauf mit Fehler beendet (siehe ERROR-Operand bei SRTOPEN). In diesem Fall muss der Programmfehler beseitigt und der Programmablauf wiederholt werden.
- Die Makros von SORTZM können sowohl in schreibgeschützten Speicherseiten (RDONLY=YES) als auch in überschreibbaren Speicherseiten (RDONLY=NO) ablaufen.
 - Bei RDONLY=YES (siehe Makroaufrufe SRT0 und SRT1) verändert SORT auch das Register 13. Dieses Register darf daher zwischen den Makroaufrufen der SORTZM nicht verändert werden (ggf. vom Benutzer sicherzustellen).
 - Bei RDONLY=NO (Standard) legt SORTZM im Übergabebereich (siehe Anhang, [Seite 414](#)) einen 4 Byte großen Sicherstellungsbereich an und führt das Sichern und Wiederherstellen von Register 13 selbst durch.

Mehrfachsordieren mit SORTZM

SORTZM erlaubt es, Eingabesätze an verschiedene ineinander geschachtelte einzelne Sortierläufe zu übergeben. Die einzelnen Makroaufrufe dürfen dabei im Hauptprogramm in beliebiger Reihenfolge angegeben werden.

Die einzige Einschränkung ist, dass die zu einem Sortierlauf gehörenden Makroaufrufe in der Reihenfolge SRTOPEN, SRTPUT, SRTGET und SRTCLSE angegeben werden müssen. Bei fehlerhafter Reihenfolge der Makroaufrufe (z.B. doppeltes SRTOPEN, SRTPUT nach SRTGET) gibt SORT die Meldung SRT1308 aus und beendet den Programmablauf mit Fehler (siehe ERROR-Operand bei SRTOPEN).

Für jeden Sortierlauf ist ein eigener Versorgungsbereich anzugeben (siehe auch die Beispiele zu SORTZM, [Seite 372](#) ff). Bei den Makroaufrufen der SORTZM können die Operanden sowohl als Stellungs- als auch als Schlüsselwortoperanden angegeben werden.

5.3.2 Makroaufrufe der SORT-Zugriffsmethode SORTZM

Makros früherer SORT-Versionen (V7.0 bis V7.3) sind ohne Neuübersetzung mit SORT V8.0 ablauffähig, es findet jedoch keine Prüfung auf korrekte Reihenfolge der Makros statt. Makros ab Version 7.4 können ohne Einschränkungen unter SORT V8.0 verwendet werden (außer Operand LINKADR).

Bei Makros der Version 8.0 wird die Abauffähigkeit mit älteren SORT-Versionen nicht garantiert. Insbesondere treten beim Ablauf mit SORT der Version V7.3 oder kleiner unbefriedigte Externverweise beim Binden auf.

5.3.3 SRTOPEN – Sortierung eröffnen

Der Makroaufruf SRTOPEN ruft SORT auf.

Name	Operation	Operanden
[name]	SRTOPEN	[[SCB =] adr1 / (r1)] [, [ERROR =] adr2 / (r2)] [, LINKADR = linkadr / (r3)]

name	Symbolischer Name, der den Makroaufruf adressiert (z.B. als Sprungziel).
SCB = adr1 / (r1)	bezeichnet den Versorgungsblock SVB.
adr1	symbolische Adresse des Versorgungsblocks SVB.
(r1)	Register, das die Adresse des Versorgungsblocks SVB enthält. $0 \leq r1 \leq 15$ Ist kein Operand SCB angegeben, wird als Standardwert $r1=0$ angenommen.
ERROR = adr2 / (r2)	Bezeichnet die Fehlerroutine, zu der bei einem fehlerhaften SORT-Lauf verzweigt wird. Bei fehlerhaftem Ablauf mit Verzweigung zur ERROR-Adresse darf danach kein SRTCLSE-Makro für diese Sortierung aufgerufen werden.
adr2	symbolische Adresse der Fehlerroutine.

- (r2) Register, das die Adresse der Fehlerroutine enthält.
 $2 \leq r2 \leq 12$
Der Inhalt des Registers r2 darf während des gesamten SORT-Laufes nicht verändert werden.

Ist kein Operand ERROR angegeben, bricht SORT einen fehlerhaften Lauf mit TERM MODE=ABNORMAL,UNIT=STEP ab.
- LINKADR = linkadr / (r3)
Angabe der Adresse des Eintrittspunktes für die Zugriffsmethode.
Es handelt sich dabei um den Eintrittspunkt SORTZM1.
- linkadr symbolische Adresse eines Wortes, in dem der Anwender vor der Ausführung des Makros die Adresse des Eintrittspunktes abgelegt hat.
- (r3) Register, das die Adresse eines Wortes enthält, das auf die Adresse des Eintrittspunktes verweist.
 $2 \leq r3 \leq 12$

Ist kein Operand LINKADR angegeben, wird als Standardwert V(SORTZM1) angenommen.
- Hinweis*
- Falls der Operand LINKADR verwendet wird, muss er bei **allen** Zugriffsmethoden-Makros angegeben werden.

5.3.4 SRTPUT – Satz an SORT übergeben

Der Makroaufruf SRTPUT übergibt einen Satz aus der Eingabe an SORT. Für jeden Satz der Eingabe muss ein SRTPUT-Aufruf abgesetzt werden. SRTPUT-Aufrufe sind nur bis zum ersten SRTGET-Aufruf erlaubt.

Name	Operation	Operanden
[name]	SRTPUT	[[SCB =] adr1 / (r1)] [, [RECORD =] adr2 / (r2)] [, LINKADR = linkadr / (r3)]

name	Symbolischer Name, der den Makroaufruf adressiert (z.B. als Sprungziel).
SCB = adr1 / (r1)	bezeichnet den Versorgungsblock SVB.
adr1	symbolische Adresse des Versorgungsblocks SVB.
(r1)	Register, das die Adresse des Versorgungsblocks SVB enthält. $0 \leq r1 \leq 15$ Ist kein Operand SCB angegeben, wird als Standardwert r1=0 angenommen.
RECORD = adr2 / (r2)	Bezeichnet den Satzpuffer im Hauptprogramm, aus dem der Satz an SORT übergeben wird.
adr2	symbolische Adresse des Satzpuffers.
(r2)	Register, das die Adresse des Satzpuffers enthält. $0 \leq r2 \leq 12$ Ist kein Operand RECORD angegeben, wird als Standardwert r2=1 angenommen.
LINKADR = linkadr / (r3)	Angabe der Adresse des Eintrittspunktes für die Zugriffsmethode. Es handelt sich dabei um den Eintrittspunkt SORTZM1.
linkadr	symbolische Adresse eines Wortes, in dem der Anwender vor der Ausführung des Makros die Adresse des Eintrittspunktes abgelegt hat.

(r3) Register, das die Adresse eines Wortes enthält, das auf die Adresse des Eintrittspunktes verweist.

$$2 \leq r3 \leq 12$$

Ist kein Operand LINKADR angegeben, wird als Standardwert V(SORTZM1) angenommen.

Hinweis

Falls der Operand LINKADR verwendet wird, muss er bei **allen** Zugriffsmethoden-Makros angegeben werden.

5.3.5 SRTGET – Satz von SORT übernehmen

Der Makroaufruf SRTGET übernimmt einen Satz von SORT nach der Sortierung. Für jeden Satz der SORT-Ausgabe muss ein SRTGET-Aufruf abgesetzt werden.

Name	Operation	Operanden
[name]	SRTGET	[[SCB =] adr1 / (r1)] [, [RECORD =] adr2 / (r2)] [, [EOS =] adr3 / (r3)] [, LINKADR = linkadr / (r4)]

name	Symbolischer Name, der den Makroaufruf adressiert (z.B. als Sprungziel).
SCB = adr1 / (r1)	bezeichnet den Versorgungsblock SVB.
adr1	symbolische Adresse des Versorgungsblocks SVB.
(r1)	Register, das die Adresse des Versorgungsblocks SVB enthält. $0 \leq r1 \leq 15$ Ist kein Operand SCB angegeben, wird als Standardwert $r1=0$ angenommen.
RECORD = adr2 / (r2)	bezeichnet den Satzpuffer im Hauptprogramm, in den der Satz von der SORT-Ausgabe übernommen wird (Übertragungsbetrieb).
adr2	symbolische Adresse des Satzpuffers.
(r2)	Register, das die Adresse des Satzpuffers enthält. $0 \leq r2 \leq 12$ Ist kein Operand RECORD angegeben, wird der Satz von der SORT-Ausgabe in einem SORT-eigenen Satzpuffer zur Verfügung gestellt, dessen Adresse in Register 1 geliefert wird (Ortungsbetrieb).
EOS = adr3 / (r3)	bezeichnet die Routine zu der am Ende der Übergabe verzweigt wird.
adr3	symbolische Adresse der Routine.
(r3)	Register, das die Adresse der Routine enthält. $2 < r3 \leq 12$ Ist kein Operand EOS angegeben, wird dem Hauptprogramm das Ende der Übergabe durch die Adresse Null in Register 1 angezeigt.

LINKADR = linkadr / (r4)

Angabe der Adresse des Eintrittspunktes für die Zugriffsmethode.
Es handelt sich dabei um den Eintrittspunkt SORTZM1.

linkadr

symbolische Adresse eines Wortes, in dem der Anwender vor der Ausführung des Makros die Adresse des Eintrittspunktes abgelegt hat.

(r3)

Register, das die Adresse eines Wortes enthält, das auf die Adresse des Eintrittspunktes verweist.

$2 \leq r4 \leq 12$

Ist kein Operand LINKADR angegeben, wird als Standardwert V(SORTZM1) angenommen.

Hinweis

Falls der Operand LINKADR verwendet wird, muss er bei **allen** Zugriffsmethoden-Makros angegeben werden.

5.3.6 SRTCLSE – Sortierung schließen

Der Makroaufruf SRTCLSE schließt die Sortierung. SRTCLSE darf nicht mehr aufgerufen werden, wenn SORT über den Fehlerausgang ERROR ins Hauptprogramm zurückgekehrt ist.

Name	Operation	Operanden
[name]	SRTCLSE	[[SCB =] adr / (r1)] [,LINKADR = linkadr / (r2)]

name	Symbolischer Name, der den Makroaufruf adressiert (z.B. als Sprungziel).
SCB = adr / (r1)	bezeichnet den Versorgungsblock SVB.
adr	symbolische Adresse des Versorgungsblocks SVB.
(r1)	Register, das die Adresse des Versorgungsblocks SVB enthält. $0 \leq r1 \leq 15$ Ist kein Operand SCB angegeben, wird als Standardwert $r=0$ angenommen.
LINKADR = linkadr / (r2)	Angabe der Adresse des Eintrittspunktes für die Zugriffsmethode. Es handelt sich dabei um den Eintrittspunkt SORTZM1.
linkadr	symbolische Adresse eines Wortes, in dem der Anwender vor der Ausführung des Makros die Adresse des Eintrittspunktes abgelegt hat.
(r2)	Register, das die Adresse eines Wortes enthält, das auf die Adresse des Eintrittspunktes verweist. $2 \leq r2 \leq 12$ Ist kein Operand LINKADR angegeben, wird als Standardwert V(SORTZM1) angenommen.

Hinweis

Falls der Operand LINKADR verwendet wird, muss er bei **allen** Zugriffsmethoden-Makros angegeben werden.

5.3.7 Beispiel

```

SRTZM   START
        .
        .
        .
SRTOPEN SCB=B1,ERROR=FEHL _____ (1)
        .
        .
        .
EINLES  RDATA   EINB,FEHL _____ (2)
        SRTPUT  SCB=B1,RECORD=EINB
        B       EINLES
        .
        .
        .
AUSGABE SRTGET   SCB=B1,RECORD=AUSB,EOS=CLOSE _____ (3)
        B       AUSGABE
CLOSE   SRTCLSE SCB=B1 _____ (4)
        .
        .
        .
FEHL    TERM
        TERM MODE=ABNORMAL,UNIT=STEP
        .
        .
        .
EINB    DS      CL80 _____ (5)
        .
        .
        .
AUSB    DS      CL80 _____ (6)
        .
        .
        .
B1      SRT1    (SORT-RECORDS FIELDS=FIELD-EXPLICIT(POSITION=5,- _____ (7)
          SRT1    (LENGTH=5)),SDF=YES
          SRT1    (SET-RECORD-ATTRIBUTES INPUT=VARIABLE(80))
          SRT1    (END)
        END      SRTZM

```

- (1) Mit dem Makroaufruf SRTOPEN wird SORT als Unterprogramm aufgerufen. Über die Adresse B1 wird SORT mitgeteilt, wo die Makroaufrufe stehen, die die Anweisungen für diesen Lauf des Sortier-/Mischprogramms enthalten. FEHL ist die Adresse, zu der bei vorzeitigem Abbruch des SORT-Laufs verzweigt werden soll.
- (2) Hier beginnt die Leseschleife. Die zu sortierenden Sätze werden in den Eingabebereich EINB eingelesen (z.B. über RDATA). Anschließend wird der Satz aus der Eingabe über den Makroaufruf SRTPUT an SORT übergeben. Dieser SRTPUT-Aufruf wird so oft wiederholt, wie Sätze an SORT zu übergeben sind.
- (3) Anfang der Ausgabeschleife. Über den Makroaufruf SRTGET werden die sortierten Sätze von SORT in den Ausgabebereich AUSB übernommen. Dieser SRTGET-Aufruf wird so oft wiederholt, wie Sätze übergeben werden. Sind alle Sätze übernommen (Endebedingung EOS), wird zur Beendigung von SORT zum Aufruf SRTCLSE verzweigt.
- (4) Der Makroaufruf SRTCLSE B1 beendet den unter B1 beschriebenen Sortierlauf.
- (5) Eingabebereich für die zu sortierenden Sätze.
- (6) Ausgabebereich für die sortierten Sätze.
- (7) SRT1-Makroaufrufe, die die Anweisungen für den SORT-Lauf an Adresse B1 enthalten.

6 Benutzerausgänge von SORT

Das Sortier-/Mischprogramm verfügt über mehrere Benutzerausgänge, an denen der Benutzer bestimmte Maßnahmen veranlassen kann.

Folgende Benutzerausgänge sind vorhanden:

Ausgang	Funktion (Kurzbeschreibung)	Bearbeitungsphase
PLANNING	wird aktiviert, wenn die Planung abgeschlossen ist und SORT die Sortierstrategie festgelegt hat. (Beschreibung ab Seite 242)	Planungsphase
INPUT	wird aktiviert, wenn SORT einen Satz von der Eingabe übernimmt. Der Satz kann dann geändert, gelöscht oder eingefügt werden. Der Aufbau der Schnittstelle ist abhängig vom Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung. Der Funktionsumfang ist für 24- oder 31-Bit-Adressierungsmodus nahezu gleich. (Beschreibung ab Seite 243)	Ersteingabe
OUTPUT	wird unmittelbar, bevor SORT in die Ausgabe-datei schreibt, aktiviert. Die Ausgabesätze können so kontrolliert, geändert, eingefügt oder gelöscht werden. Der Aufbau der Schnittstelle ist abhängig vom Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung. Der Funktionsumfang ist für 24- oder 31-Bit-Adressierungsmodus nahezu gleich. (Beschreibung ab Seite 248)	Endausgabe
WORK-FILE-OVERFLOW	wird aktiviert, wenn die Kapazität der Arbeitsdatei erschöpft ist (SORT). (Beschreibung ab Seite 254)	Ersteingabe
EXLST-FOR-INPUT	Über diesen Benutzerausgang wird für die Eingabe ein DVS-EXLST-Makro angeschlossen. (Beschreibung ab Seite 256)	Vorbereitungsphase
EXLST-FOR-OUTPUT	Über diesen Benutzerausgang wird für die Ausgabe ein DVS-EXLST-Makro angeschlossen. (Beschreibung ab Seite 258)	Ausgabedatei-bearbeitung

Ausgang	Funktion (Kurzbeschreibung)	Bearbeitungsphase
PHYSICAL-TRANSLATE	Angabe von 2 Codetabellen zur Umcodierung der PHYSICAL-TRANSLATE-Formatfelder zur Festlegung einer anderen Sortierfolge. (Beschreibung ab Seite 260)	Ersteingabe, internes Mischen, Endausgabe, MERGE-Lauf
VIRTUAL-TRANSLATE	Angabe einer Codetabelle für VIRTUAL-TRANSLATE-Formatfelder zur Festlegung einer anderen Sortierfolge. (Beschreibung ab Seite 261)	Ersteingabe, internes Mischen, Endausgabe, MERGE-Lauf
EXTERNAL-COMPARE	Dieser Benutzerausgang wird bei jedem Satz für Sortierfelder mit der Sortierreihenfolge EXTERNAL-COMPARE aktiviert, sodass die Benutzerroutine die Reihenfolge festlegen kann. Der Aufbau der Schnittstelle ist abhängig vom Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung. Der Funktionsumfang ist für 24- oder 31-Bit-Adressierungsmodus gleich. (Beschreibung ab Seite 263)	Ersteingabe, internes Mischen, Endausgabe, MERGE-Lauf
TRANSLATE-CHARACTER	Angabe von 2 Codetabellen zur Umcodierung der TRANSLATE-CHARACTER-Formatfelder zur Festlegung einer anderen Sortierfolge. (Beschreibung ab Seite 266)	Ersteingabe, internes Mischen, Endausgabe, MERGE-Lauf
INT	wird aktiviert, wenn der Benutzer über SEND-MESSAGE-Kommando Auskunft über den Stand der Sortierung/Mischung wünscht oder eine sonstige erlaubte Maßnahme ausführen lassen will. Das Kommando kann bei Stapelverarbeitung auch vom Operateur an der Konsole abgesetzt werden. Er muss dabei die Task-Sequence-Nummer angeben. (Beschreibung ab Seite 267)	alle Phasen

Mit Ausnahme des Benutzerausgangs INT kann der Benutzer für alle Ausgänge in der ASSIGN-EXITS-Anweisung Maßnahmen angeben. Der Benutzer kann über die Ausgänge eigene Routinen anschließen, in Dialog mit dem Sortier-/Mischprogramm treten oder den SORT-Lauf an bestimmten festgelegten Ablaufpunkten beeinflussen.

Beim Benutzerausgang muss eine der folgenden Maßnahmen, sofern sie der Benutzer-
ausgang zulässt, angegeben werden..

DIALOG	<p>SORT soll über SYSOUT den augenblicklichen Ablaufzustand melden und eine weitere Maßnahme über SYSDTA anfordern. Wird ein Benutzerausgang erreicht, bewirkt die Maßnahme DIALOG, dass SORT mit der zugehörigen Meldung die Maßnahmen ausgibt, die als Antwort zulässig sind. Die Angabe DIALOG ist nur im Dialogbetrieb möglich. Je nach Benutzerausgang ist jeweils eine der folgenden Antworten möglich. Beim INT-Ausgang können auch mehrere Maßnahmen nacheinander ausgeführt werden.</p> <p>C[ONTINUE] Die Verarbeitung soll fortgesetzt werden.</p> <p>F[INISH] SORT soll die Satzeingabe beenden und die bereits übernommenen Sätze bearbeiten.</p> <p>S[TART] Der SORT-Lauf soll mit den verbesserten Anweisungen neu gestartet werden. Nur die geänderten Anweisungen müssen eingegeben werden.</p> <p>T[ERMINATE] Der SORT-Lauf soll beendet werden.</p> <p>Beim INT-Ausgang (nicht in der ASSIGN-EXITS-Anweisung angebar) ist zusätzlich noch möglich:</p> <p>D[ISPLAY] SORT soll den Stand der Verarbeitung melden, d.h. die augenblickliche aktive Phase des SORT-Laufs, verbrauchte Zeiten und die laufenden Bearbeitungszähler.</p> <p>CK[PT] SORT soll einen Fixpunkt ausgeben, soweit möglich.</p> <p><i>Hinweise</i></p> <ul style="list-style-type: none"> – Gibt der Benutzer bei DIALOG in einer Prozedur keine Maßnahme an, wird dies wie eine CONTINUE-Maßnahme behandelt. – Die Angaben für die Maßnahmen bei DIALOG können von hinten her bis zur Eindeutigkeit verkürzt werden. – Die Angabe von DIALOG in einer Batch-Prozedur wird ignoriert, die Verarbeitung wird fortgesetzt.
FINISH-INPUT	<p>SORT soll die Satzeingabe beenden und die bereits übernommenen Sätze bearbeiten.</p>

MODULE (NAME=<name 1..8>) SORT soll eine Benutzerroutine einbinden und beim angegebenen Ausgang aufrufen. Diese Benutzerroutine muss als Bindemodul in der verwendeten TASKLIB liegen oder in einer Bibliothek, die mit dem Dateikettungsnamen SORTMODS zugeordnet wurde.

Registerkonventionen

- Alle Register, die in einer Benutzerroutine verwendet werden, müssen vorher sichergestellt werden (z.B. durch SAVE (14,12) oder STM 14,12,12(13)).
- Vor dem Rücksprung von der Benutzerroutine ins Sortier-/Mischprogramm müssen die sichergestellten Register mit Ausnahme von Register 1 und 15 wieder geladen werden, z.B. durch RETRN (14,12),RC=8.
- Folgende Register haben eine feste Verwendung:

Reg1 Wird eine Benutzerroutine angesprungen, enthält das Register 1 die Adresse einer Operandenliste mit Satzadressen und ggf. Anzeigen. Bei der Rückkehr ins Sortier-/Mischprogramm muss Register 1 folgende Angaben enthalten:

- Beim Benutzerausgang INPUT die Adresse des verarbeiteten Satzes.
- Bei den Benutzerausgängen OUTPUT und EXTERNAL-COMPARE die Adresse des entsprechenden Versorgungsbereichs.

Reg 13 verweist auf einen 18 Worte großen Sicherstellungsbereich im Sortier-/Mischprogramm.

Reg 14 enthält die Rücksprungadresse ins Sortier-/Mischprogramm. Die Benutzerroutine wird durch BALR 14,15 angesprungen.

Reg 15 Wird eine Benutzerroutine angesprungen, enthält das Register 15 die Adresse des Einsprungpunkts in die Routine.
Wird PARAMETER-MODE=24 verwendet, so enthält Register 15 bei der Rückkehr ins Sortier-/Mischprogramm bei den Ausgängen INPUT, OUTPUT, und EXTERNAL-COMPARE im rechtsbündigen Byte das Rücksprungkennzeichen. Die restlichen Byte müssen auf Null gesetzt werden. Bei der Verwendung von PARAMETER-MODE=ANY zeigt Register 1 auf den Parameterbereich. Der vierte Pointer in diesem Bereich zeigt auf das Aktionswort, in dem das Rücksprungkennzeichen steht.

Die Übergabe der Daten an SORT muss aus dem Programm-Adressraum erfolgen.

Ist im zugehörigen SORT-Makro (SRT0, SRT1) der Operand STXIT=YES angegeben, so prüft SORT die von der Benutzerroutine zurückgegebenen Daten und Adressen auf Gültigkeit. Erkannte Programmfehler führen zur Ausgabe einer entsprechenden Meldung und zur Beendigung des Sortierlaufes.

In diesem Fall muss der Programmfehler beseitigt und der Sortierlauf wiederholt werden.

TERMINATE-
ABNORMAL

Der SORT-Lauf soll beendet werden.

6.1 PLANNING – Planung beendet

Der Ausgang PLANNING wird aktiviert, wenn alle Informationen für den Sortier-/Mischlauf ausgewertet sind und eine Strategie für die Ausführung festgelegt wurde. Über den Ausgang PLANNING kann der Benutzer einen SORT-Lauf optimieren, wenn er als Maßnahme DIALOG angibt. Der Lauf kann dann, nachdem SORT die zugewiesenen Betriebsmittel überprüft hat, mit den verbesserten Anweisungen erneut gestartet werden.

Der Ausgang PLANNING lässt für die ASSIGN-EXITS-Anweisung eine der folgenden Maßnahmen zu.

DIALOG SORT gibt folgende Meldungen aus (nur bei MIN-MSG-WEIGHT=*ALL):

SRT1031 geschätzte Größe der Arbeitsdatei
 SRT1033 intensiv genutzter Speicher
 SRT1050 Blockgröße von Arbeits- und Hilfsdatei
 SRT1061 Maximale Anzahl der Mischwege

Danach wird im Stapelbetrieb mit CONTINUE als Maßnahme fortgesetzt. Im Dialogbetrieb sind die folgenden Maßnahmen zulässig:

C[ONTINUE] Die Verarbeitung soll fortgesetzt werden.

S[TART] Der SORT-Lauf soll mit den verbesserten Anweisungen neu gestartet werden. Nur die geänderten Anweisungen müssen neu eingegeben werden. START ist nur bei autonomen SORT-Läufen und beim Aufruf von SORT als Unterprogramm über Ebene 0 zulässig.

T[ERMINATE] Der SORT-Lauf soll abgebrochen werden.

TERMINATE-
 ABNORMAL Der SORT-Lauf soll beendet werden.

6.2 INPUT – Eingabe-Satzbearbeitung

Über den Benutzerausgang INPUT kann der Benutzer die Satzeingabe des Sortier-/Mischprogramms kontrollieren und beeinflussen. Sätze können überprüft, geändert, eingefügt oder gelöscht werden. Der Benutzer kann auch die gesamte Eingabe über INPUT vornehmen. INPUT wird jedes Mal angesprungen, wenn ein Eingabesatz an das Sortier-/Mischprogramm übergeben wird.

Der Aufbau der Schnittstelle ist vom Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung abhängig.

Schnittstelle zur Benutzerroutine bei PARAMETER-MODE=ANY

Der Benutzerausgang bietet bei 31-Bit-Adressierung eine Erweiterung um die Benutzerkonstante und den Namen des coded character sets. Wenn das Sortier-/Mischprogramm die Steuerung an die Benutzerroutine übergibt, enthält Register 1 die Adresse eines 20 Byte großen Versorgungsbereichs mit folgendem Aufbau:

- Byte 0 - 3: Adresse des nächsten Eingabesatzes
Beim Einfügen oder Ändern eines Eingabesatzes muss hier die Satzadresse vom Benutzer übergeben werden. Die Adresse des nächsten Eingabesatzes wird auf Null gesetzt bei:
- Ende der Eingabedatei
 - Ende einer Dateifolge (Mehrdateiensort)
 - Fehlen der Eingabedatei
- Byte 4 - 7: Adresse der Benutzerkonstante
Die Adresse der Benutzerkonstante verweist auf einen 4 Byte großen Bereich, über den der Benutzer Informationen an den Benutzerausgang OUTPUT weitergeben kann (z.B. die Adresse eines dynamisch angeforderten Speichers).
- Byte 8 - 11: Adresse des Dateikennzeichens
Die Adresse des Dateikennzeichens verweist auf einen 4 Byte großen Bereich. Wird ein Satz eingefügt, liefert SORT in diesem Bereich rechtsbündig das Kennzeichen der aktuellen Datei.
Das Dateikennzeichen der Eingabedatei wird binär angegeben und entspricht der laufenden Nummer von SORTINxx bzw. MERGExx.
Das Dateikennzeichen dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Byte 12 - 15: Adresse des Aktionswortes

Wenn die Benutzerroutine die Steuerung an SORT zurückgibt, muss im rechtsbündigen Byte des Aktionswortes eines der folgenden Rücksprungkennzeichen mitgegeben werden:

- X'00' SORT soll den Eingabesatz übernehmen. Byte 0 - 3 des Versorgungsbereichs muss dann die Adresse des Eingabesatzes enthalten. Dieser Satz kann der von der Versorgung übernommene oder ein geänderter Satz sein.
Soll ein Satz verlängert werden, muss der Benutzer dafür einen eigenen Bereich zur Verfügung stellen.
Bei Sätzen variabler Länge darf das Längenfeld (die ersten vier Byte des Satzes) nicht im Originalbereich verändert werden. Soll die Länge variabler Sätze verändert werden, muss auch bei Verkürzung ein eigener Satzbereich bereitgestellt werden.
- X'04' Der Satz, dessen Adresse in Byte 0 - 3 des Versorgungsbereichs steht, ist zu löschen. Nicht zulässig, wenn im Versorgungsbereich die Adresse auf Null gesetzt ist, d.h. die Eingabe abgeschlossen oder keine Eingabedatei vorhanden ist.
- X'08' Dieses Rücksprungkennzeichen muss der Benutzer angeben, wenn nicht mehr zur Benutzerroutine verzweigt werden soll. Das Kennzeichen X'08' ist beim Ende der Gesamteingabe erforderlich, wenn nicht mit X'14' ein vorzeitiges Ende oder mit X'10' ein Abbruch des SORT-Laufs angezeigt wird.
Wird das Kennzeichen X'08' angegeben, bevor die Eingabe beendet ist, liest SORT die restlichen Eingabesätze ein, ohne jedoch die Benutzerroutine aufzurufen.
- X'0C' Der Satz, dessen Adresse in Byte 0 - 3 des Versorgungsbereichs steht, soll eingefügt werden. Sätze können auch dann eingefügt werden, wenn bei Aufruf der Benutzerroutine im Versorgungsbereich die Adresse Null steht, d.h. die Eingabe abgeschlossen ist oder keine Eingabedateien vorhanden sind. X'0C' muss auf jeden Fall gesetzt werden, wenn der Benutzer die Eingabe selbst besorgt.
- X'10' bewirkt, dass der SORT-Lauf abgebrochen wird. Wurde SORT als Unterprogramm aufgerufen, wird mit dem Fehlerkennzeichen X'FF' ins rufende Programm zurückgekehrt.

- X'14' Die Eingabe wird vorzeitig beendet. Bei einem Sortierlauf bezieht sich dies auf die gesamte Dateifolge, d.h. X'14' schließt das Rücksprungzeichen X'08' ein.
In einem Mischlauf wird nur die Mischeingabedatei geschlossen, die im Dateikennzeichenbyte angegeben ist. Der Mischlauf wird dann mit den verbliebenen Mischeingabedateien fortgesetzt. Bei der letzten Mischeingabedatei ist mit X'14' auch X'08' eingeschlossen.

Byte 16 - 19: Adresse des CCSN

Die Adresse verweist auf einen 8 Byte großen Bereich, in dem SORT den Namen des coded character sets der Daten ablegt.
Der Name dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Hinweise

- Bei der Rückkehr ins Sortier-/Mischprogramm mit gesetztem Rücksprungkennzeichen X'00' oder X'0C' muss in Byte 0 - 3 des Versorgungsbereichs die Adresse des zu übergebenden Satzes oder die Adresse Null stehen.
Steht im Versorgungsbereich bei X'00' und X'0C' eine Adresse Null, wird bei Sortierläufen die aktuelle Eingabedatei abgeschlossen. Bei Dateifolgen wird die aktuelle Eingabedatei abgeschlossen und mit der nächsten fortgefahren. Bei Mischläufen führt eine Adresse Null zum Abschluss der Eingabedatei, auf die das von SORT gelieferte Dateikennzeichen verweist (gleiche Wirkung wie Rücksprungkennzeichen X'14'). Der Mischlauf wird mit den verbliebenen Mischeingabedateien fortgesetzt.
- Beim Adresslistensortieren darf dieser Benutzerausgang nicht zum Einfügen von Sätzen benutzt werden.
- Der übergebene Satz wird bereits im Satzübergabebereich des Benutzers bearbeitet und dabei zum Teil verändert (konvertiert).
- Für einzufügende und zu verlängernde Sätze muss der Benutzer einen eigenen Satzbereich zur Verfügung stellen.
- Bei Sätzen variabler Länge darf das Längenfeld (die ersten vier Byte des Satzes) nicht im Originalbereich verändert werden. Soll die Länge variabler Sätze verändert werden, muss auch bei Verkürzung ein eigener Satzbereich bereitgestellt werden.

Schnittstelle zur Benutzerroutine bei PARAMETER-MODE=24

Wenn das Sortier-/Mischprogramm die Steuerung an die Benutzerroutine übergibt, weist Register 1 auf einen 4 Byte großen Bereich mit folgendem Aufbau:

Byte 0: Dateikennzeichen

Dieses Byte enthält das Dateikennzeichen der Eingabedatei, von der der Eingabesatz gelesen wurde. Wird ein Satz eingefügt, liefert SORT in diesem Byte das Kennzeichen der aktuellen Datei. Das Dateikennzeichen der Eingabedatei wird in binärer Schreibweise angegeben und entspricht der laufenden Nummer von SORTINxx bzw. MERGExx.

Das Dateikennzeichen dient nur zur Information für den Benutzer.

Beim Rücksprung zum Sortierprogramm wird dieses Byte nicht ausgewertet.

Byte 1 - 3: Adresse des nächsten Eingabesatzes

Beim Einfügen oder Ändern eines Eingabesatzes muss hier die Satzadresse vom Benutzer übergeben werden. Die Adresse des nächsten Eingabesatzes wird auf Null gesetzt bei:

- Ende der Eingabedatei
- Ende einer Dateifolge (Mehrdateiensort)
- Fehlen der Eingabedatei

Wenn die Benutzerroutine die Steuerung an SORT zurückgibt, muss im rechtsbündigen Byte von Register 15 eines der folgenden Rücksprungkennzeichen mitgegeben werden:

X'00' SORT soll den Eingabesatz übernehmen. Die Adresse des Eingabesatzes muss dann nach Register 1 geladen werden. Dieser Satz kann ein von der Versorgung übernommener oder ein geänderter sein.

Soll ein Satz verlängert werden, muss der Benutzer hierfür einen eigenen Bereich zur Verfügung stellen.

Bei Sätzen variabler Länge darf das Längengebiet (die ersten vier Byte des Satzes) nicht im Originalbereich verändert werden. Soll die Länge variabler Sätze verändert werden, muss auch bei Verkürzung ein eigener Bereich bereitgestellt werden.

X'04' veranlasst, dass der im Versorgungsbereich angegebene Satz gelöscht wird.

X'08' Dieses Rücksprungkennzeichen muss der Benutzer angeben, wenn nicht mehr zur Benutzerroutine verzweigt werden soll.

Das Kennzeichen X'08' ist beim Ende der Gesamteingabe erforderlich, wenn nicht mit X'14' ein vorzeitiges Ende oder mit X'10' ein Abbruch des SORT-Laufs angezeigt wird.

Wird das Kennzeichen X'08' gegeben, bevor die Eingabe beendet ist, liest SORT die restlichen Eingabesätze ein, ohne jedoch die Benutzerroutine aufzurufen.

- X'0C' Der Satz, dessen Adresse in Register 1 steht, soll eingefügt werden. Sätze können auch dann eingefügt werden, wenn im Versorgungsbereich die Adresse Null steht, d.h. die Eingabe abgeschlossen ist oder keine Eingabedateien vorhanden sind. X'0C' muss auf jeden Fall gesetzt werden, wenn der Benutzer die Eingabe selbst besorgt.
- X'10' bewirkt, dass der SORT-Lauf abgebrochen wird. Wurde SORT als Unterprogramm aufgerufen, wird mit dem Fehlerkennzeichen X'FF' ins rufende Programm zurückgekehrt.
- X'14' Die Eingabe wird vorzeitig beendet. Bei einem Sortierlauf bezieht sich dies auf die gesamte Dateifolge, d.h. X'14' schließt das Rücksprungkennzeichen X'08' ein. In einem Mischlauf wird nur die Mischeingabedatei abgeschlossen, auf die das von SORT gelieferte Dateikennzeichen verweist. Der Mischlauf wird dann mit den verbliebenen Mischeingabedateien fortgesetzt. Bei der letzten Mischeingabedatei ist mit X'14' auch X'08' eingeschlossen.

Hinweise

- Steht im Register 1 bei X'00' und X'0C' eine Adresse Null, wird bei Sortierläufen die aktuelle Eingabedatei abgeschlossen. Bei Dateifolgen wird die aktuelle Eingabedatei abgeschlossen und mit der nächsten fortgefahren. Bei Mischläufen führt eine Adresse Null zum Abschluss der Eingabedatei, auf die das Dateikennzeichenbyte verweist (gleiche Wirkung wie Rücksprungkennzeichen X'14'). Der Mischlauf wird mit den verbliebenen Mischeingabedateien fortgesetzt.
- Beim Adresslistensortieren darf dieser Benutzerausgang nicht zum Einfügen von Sätzen benutzt werden.
- Der übergebene Satz wird bereits im Satzübergabebereich des Benutzers bearbeitet und dabei zum Teil verändert (konvertiert).
- Für einzufügende und zu verlängernde Sätze muss der Benutzer einen eigenen Satzbereich zur Verfügung stellen.
- Bei Sätzen variabler Länge darf das Längenfeld (die ersten vier Byte des Satzes) nicht im Originalbereich verändert werden. Soll die Länge variabler Sätze verändert werden, muss auch bei Verkürzung ein eigener Satzbereich bereitgestellt werden.

6.3 OUTPUT – Ausgabe-Satzbearbeitung

Über den Ausgang OUTPUT kann der Benutzer die Satzausgabe des Sortier-/Mischprogramms kontrollieren und beeinflussen. Sätze können auf diese Weise von einer Benutzerroutine geprüft, geändert, eingefügt oder gelöscht werden. Außerdem ist es möglich, Sätze mit gleichen Sortierfeldern zusammenzufassen (Verdichten, Summensatzbildung). Die über OUTPUT angeschlossene Benutzerroutine kann auch die gesamte Ausgabe übernehmen.

Der Ausgang OUTPUT wird bei PARAMETER-MODE=24 zum ersten Mal vor einer Ausgabe des Sortier-/Mischprogramms mit einer Nullversorgung aktiviert. Auf diese Weise kann über ein Kennzeichenbyte die gewünschte Satzverarbeitung angezeigt werden. Danach wird OUTPUT vor jedem Schreiben eines Satzes in die Ausgabedatei aktiviert.

Bei PARAMETER-MODE=ANY wird der Ausgang vor jedem Schreiben eines Satzes in die Ausgabedatei aktiviert.

Der Aufbau der Schnittstelle ist vom Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung abhängig.

Schnittstelle zur Benutzerroutine bei PARAMETER-MODE=ANY

Der Benutzerausgang bietet bei 31-Bit-Adressierung eine Erweiterung um die Benutzerkonstante und den Namen des coded character sets.

Folgende Funktionen können am Benutzerausgang nicht ausgeführt werden:

- Steuerung der Folgekontrolle.
Diese Funktion kann durch den Parameter SEQUENCE-CHECK in der SET-SORT-OPTIONS-Anweisung ausgeführt werden.
- Anzeige von gleichen Sätzen.
Diese Funktion ist in der SUM-RECORDS-Anweisung enthalten.

Wenn das Sortier-/Mischprogramm die Steuerung an die Benutzerroutine übergibt, enthält Register 1 die Adresse eines 20 Byte großen Versorgungsbereichs mit folgendem Aufbau:

Byte 0 - 3: Adresse des nächsten Ausgabesatzes

In diesen Bytes steht die Adresse des Satzes, den SORT als nächsten vom SORT-Satzpuffer ausgeben soll. Steht der letzte Ausgabesatz im DVS-Ausgabepuffer, dann ist die Adresse Null.

Bei Rückkehr mit Rücksprungkennzeichen X'00', und X'0C' im Aktionswort muss die Benutzerroutine hier die Satzadresse eintragen.

- Byte 4 - 7: Adresse des aktuellen Ausgabesatzes
- Diese Bytes enthalten die Adresse des aktuellen Ausgabesatzes im DVS-Ausgabepuffer.
- Die Adresse ist Null, wenn der erste Satz verarbeitet ist. Die hier angegebene Adresse darf nicht verändert werden.
- Byte 8 - 11: Adresse der Benutzerkonstante
- Die Adresse der Benutzerkonstante verweist auf einen 4 Byte großen Bereich, aus dem der Benutzer Informationen übernehmen kann, die der Benutzerausgang INPUT evtl. übergeben hat.
- Byte 12 - 15: Adresse des Aktionswortes
- Wenn die Benutzerroutine die Steuerung an SORT zurückgibt, muss im rechtsbündigen Byte des Aktionswortes eines der folgenden Rücksprungkennzeichen mitgegeben werden:
- X'00' Der Satz, dessen Adresse in Byte 0 - 3 des Versorgungsbereichs steht, soll in die Ausgabe übernommen werden. Es kann dies der übernommene, ein veränderter oder ein ersetzter Satz sein. Soll der Satz verlängert werden, muss der Benutzer den Satz in einem eigenen Bereich zur Verfügung stellen.
- X'04' Der Satz, dessen Adresse in Byte 0 - 3 des Versorgungsbereichs steht, soll gelöscht werden. Steht dort die Adresse Null, ist das Rücksprungzeichen X'04' nicht erlaubt. X'04' muss auf jeden Fall gesetzt werden, wenn der Benutzer die Ausgabe selbst besorgt.
- X'08' Die Benutzerroutine soll nicht mehr aufgerufen werden. Dieses Kennzeichen muss bei Ausgabeende gesetzt werden, soweit nicht über X'10' ein Abbruch der Verarbeitung angezeigt wird. Wird X'08' vor dem Ausgabeende gesetzt, gibt SORT die restlichen Sätze aus, ohne dass die Benutzerroutine nochmals aufgerufen wird.
- X'0C' Der Satz, dessen Adresse in Byte 0 - 3 des Versorgungsbereichs steht, soll eingefügt werden. Sätze können auch dann eingefügt werden, wenn in Byte 0 - 3 des Versorgungsbereichs die Adresse Null steht.
- X'10' Der SORT-Lauf soll abgebrochen werden. Wurde SORT als Unterprogramm aufgerufen, wird bei der Rückkehr ins rufende Programm das Fehlerkennzeichen X'FF' gesetzt.

Byte 16 - 19: Adresse des CCSN

Die Adresse verweist auf einen 8 Byte großen Bereich, in dem SORT den Namen des coded character sets der Daten ablegt.

Der Name dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Hinweise

- Wird bei der Rückkehr ins Sortier-/Mischprogramm das Rücksprungkennzeichen X'00' oder X'0C' gesetzt, muss in Byte 0 - 3 des Versorgungsbereichs die Adresse des zu übergebenden Satzes oder die Adresse Null stehen. Die Adresse Null führt zum Abschluss der Sortierung (evtl. auch mit vorzeitiger Beendigung der Ausgabe). Hierbei ist die Wirkung des Rücksprungkennzeichens X'08' mit eingeschlossen.
- Die beiden in der Versorgung angegebenen Sätze stehen der Benutzerroutine im Ausgabesatzformat zur Verfügung.
- Für einzufügende oder zu verlängernde Sätze muss der Benutzer einen eigenen Bereich zur Verfügung stellen.

Schnittstelle zur Benutzerroutine bei PARAMETER-MODE=24

Übergibt SORT die Steuerung an die Benutzerroutine, so enthält Register 1 die Adresse eines 8 Byte großen Versorgungsbereichs mit folgendem Aufbau:

Byte 0: Folgekennzeichen

Dieses Byte zeigt an, ob die Sortierfelder der beiden angegebenen Sätze gleich sind. Ist dies der Fall, kann z.B. ein Summensatz gebildet werden. Sollen Sortierfelder über ELIMINATE-Operanden eliminiert werden, unterbleibt eine Anzeige. Das Folgekennzeichen wird nur dann gesetzt, wenn eine Ausgabedatei vorhanden ist.

X'00' wird gesetzt, wenn die Sortierfelder nicht gleich sind.

X'04' wird gesetzt, wenn die Sortierfelder zweier aufeinander folgender Sätze gleich sind.

Byte 1 - 3: Adresse des nächsten Ausgabesatzes

In diesen Bytes steht die Adresse des Satzes, den SORT als nächsten vom SORT-Satzpuffer ausgeben soll. Steht der letzte Ausgabesatz im DVS-Ausgabepuffer, dann ist die Adresse Null.

Gibt die Benutzerroutine die Steuerung an SORT zurück, muss im Register 1 bei Rücksprungkennzeichen X'00' und X'0C' diese Adresse, eine sonstige Satzadresse oder „Null“ stehen.

Byte 4: Kennzeichenbyte

Beim ersten Aufruf der OUTPUT-Benutzerroutine muss in diesem Byte die Satzverarbeitung angegeben werden.

X'00' Der Benutzerausgang soll für jeden Ausgabesatz angesprungen werden, ohne die Reihenfolge zu überprüfen.

X'04' Der Ausgang soll für jeden Ausgabesatz angesprungen und die Reihenfolge überprüft werden.

X'10' Der Ausgang soll für jeden Ausgabesatz angesprungen und die Reihenfolge überprüft werden. Außerdem soll im Folgekennzeichen angezeigt werden, ob die Sätze gleiche Sortierfelder haben, ausgenommen bei Sortierfeldern mit ELIMINATE-Operanden.

Hinweis

Angaben zur Reihenfolgeprüfung in der SET-SORT-OPTIONS-Anweisung (SEQUENCE-CHECK-Operand) haben Vorrang vor Angaben im Kennzeichenbyte.

Byte 5-7: Adresse des aktuellen Ausgabesatzes

Diese Bytes enthalten die Adresse des aktuellen Ausgabesatzes im DVS-Ausgabepuffer.

Die Adresse ist Null, wenn der erste Satz verarbeitet ist oder keine Ausgabedatei vorhanden ist. Die hier angegebene Adresse darf nicht verändert werden.

Konventionen für den Rücksprung ins Sortier-/Mischprogramm bei PARAMETER-MODE=24

- Erster Aufruf des Benutzerausgangs OUTPUT
Wird die OUTPUT-Benutzerroutine zum ersten Mal aufgerufen, muss im Kennzeichenbyte des Versorgungsbereichs die gewünschte Satzverarbeitung angegeben werden. Die übrigen Bytes des Versorgungsbereichs müssen Null sein. Das Register 15 wird auf das Rücksprungkennzeichen X'08' überprüft. Das Register 1 hat hier keine Bedeutung. Das Kennzeichenbyte darf während der gesamten Verarbeitung nicht verändert werden.

- Weitere Aufrufe des Benutzerausgangs OUTPUT
Gibt die Benutzerroutine die Steuerung an SORT zurück, müssen die Register 1 und 15 versorgt werden.
- Reg 15 Enthält im rechtsbündigen Byte eines der folgenden Rücksprungkennzeichen:
- X'00' Der Satz mit der im Register 1 stehenden Adresse soll in die Ausgabe übernommen werden. Es kann dies der übernommene, ein veränderter oder ein ersetzter Satz sein. Soll der Satz verlängert werden, muss der Benutzer den Satz in einem eigenen Bereich zur Verfügung stellen.
 - X'04' Der Satz, dessen Adresse in Byte 1 - 3 im Versorgungsbereich steht, soll gelöscht werden. Steht dort die Adresse Null, ist das Rücksprungkennzeichen X'04' nicht erlaubt.
X'04' muss auf jeden Fall gesetzt werden, wenn der Benutzer die Ausgabe selbst besorgt.
 - X'08' Die Benutzerroutine soll nicht mehr aufgerufen werden. Dieses Kennzeichen muss bei Ausgabeende gesetzt werden, soweit nicht über X'10' ein Abbruch der Verarbeitung angezeigt wird. Wird X'08' vor dem Ausgabeende gesetzt, gibt SORT die restlichen Sätze aus, ohne dass die Benutzerroutine nochmals aufgerufen wird.
 - X'0C' Der Satz, dessen Adresse im Register 1 steht, soll eingefügt werden. Sätze können auch dann eingefügt werden, wenn in Byte 1 - 3 des Versorgungsbereichs die Adresse Null steht.
 - X'10' Der SORT-Lauf soll abgebrochen werden. Wurde SORT als Unterprogramm aufgerufen, wird bei der Rückkehr ins rufende Programm das Fehlerkennzeichen X'FF' gesetzt.
- Reg 1 Wird bei der Rückkehr ins Sortier-/Mischprogramm das Rücksprungkennzeichen X'00' oder X'0C' gesetzt, muss im Register 1 die Adresse des zu übergebenden Satzes oder die Nulladresse stehen.
Steht im Register 1 die Adresse Null, führt das zum Abschluss der Sortierung/Mischung (evtl. auch zur vorzeitigen Beendigung der Ausgabe). Hierbei ist die Wirkung von Rücksprungkennzeichen X'08' mit eingeschlossen.

Hinweise

- Wurde das Kennzeichenbyte mit X'10' besetzt und sollen Sätze mit gleichen Sortierfeldern zusammengefasst werden (Kumulierung der Summenfelder), verfährt der Benutzer am einfachsten wie folgt:

Die Summenfelder des 1. Satzes (Adresse in Byte 1 - 3 des Versorgungsbereichs) werden zum 2. Satz (Adresse in Byte 5 - 7 des Versorgungsbereichs) addiert. Der Satz kann dann weiter bearbeitet werden, z.B. durch eine Überlaufkontrolle. Anschließend muss das Rücksprungkennzeichen X'04' (Satz löschen) gesetzt und ins Sortier-/Mischprogramm zurückgekehrt werden. Das Register 1 hat in diesem Fall keine Bedeutung.
- Die Sätze, die der Versorgungsbereich angibt (Satzadresse Byte 1 - 3 und Satzadresse Byte 5 - 7), stehen der Benutzerroutine im Ausgabeformat zur Verfügung.
- X'04' (Satz löschen) bezieht sich auf die Satzadresse in Byte 1 - 3 des Versorgungsbereichs.
- X'00' bzw. X'0C' (Satz übernehmen oder einfügen) bezieht sich auf die Satzadresse im Register 1.
- Sätze, die eingefügt oder verlängert werden sollen, müssen in einem gesonderten Speicherbereich zur Verfügung gestellt werden.

6.4 WORK-FILE-OVERFLOW – Arbeitsdatei-Überlauf

Der Ausgang WORK-FILE-OVERFLOW wird aktiviert, wenn der Überlauf einer Plattenarbeitsdatei mit Sekundärzuweisung Null unmittelbar bevorsteht, SORT den Engpass nicht mehr beheben kann und keine Hilfsdatei für einen weiteren Zyklus mehr verfügbar ist. Der Benutzer kann den Lauf abbrechen oder die Sortierung auf die von SORT übernommenen Sätze beschränken.

DIALOG

Die Anzahl der bis dahin von SORT übernommenen Sätze wird mit der Meldung „SRT1017 RECORDS TO BE SORTED/MERGED: ...n“ angezeigt und eine der folgenden Maßnahmen erwartet:

C[ONTINUE] SORT versucht, mit einer geringen Reserve den SORT-Lauf durchzuführen.

F[INISH] SORT beendet die Satzeingabe und bearbeitet die bereits übernommenen Sätze.

T[ERMINATE] SORT bricht den Lauf ab.

FINISH-INPUT

Die Satzeingabe wird beendet und SORT bearbeitet die bisher eingelesenen Sätze.

MODULE Eine Benutzerroutine wird angeschlossen. Im Register 1 steht die
(NAME=<name 1..8>, Adresse eines 8 Byte großen Versorgungsbereichs mit folgendem
INTERFACE- Aufbau:
VERSION=1/2)

Byte 0 - 3: INTERFACE-VERSION=1:
Anzahl der bisher übernommenen Sätze.
Ist diese Anzahl größer als 2.147.483.647, so wird
die Anzahl auf diese Grenzzahl gesetzt, was zu
Fehlern im Exit führen kann.

INTERFACE-VERSION=2:
Adresse eines 8-Byte-Feldes, das die Anzahl der
bisher übernommenen Sätze enthält.

Byte 4: Dateikennzeichen
Dieses Byte gibt Auskunft über die Dateart.
X'01' Arbeitsdatei
X'02' Hilfsdatei

Byte 5: Dateifolgenummer
Dieses Byte enthält die laufende Nummer des
Dateikettungsnamens SORTWKx bzw.
SORTWKxx.

Byte 6 - 7: Größe der Arbeitsdatei

Registerkonventionen

Gibt die Benutzerroutine die Steuerung an SORT zurück, steht im
rechtsbündigen Byte von Register 15 eines der folgenden Kennzei-
chen:

X'00' SORT soll die Verarbeitung fortsetzen (kann zum
Abbruch wegen Ein-/Ausgabefehler führen).

X'04' Die Eingabe wird beendet und die eingegebenen
Sätze werden sortiert (FINISH-Maßnahme).

X'08' Der Sortierlauf wird wegen eines Fehlers abgebro-
chen (TERMINATE-Maßnahme).

TERMINATE- Der Sortier-/Mischlauf soll abnormal beendet werden.
ABNORMAL

6.5 EXLST-FOR-INPUT – EXLST-Ausgang für Eingabedateien

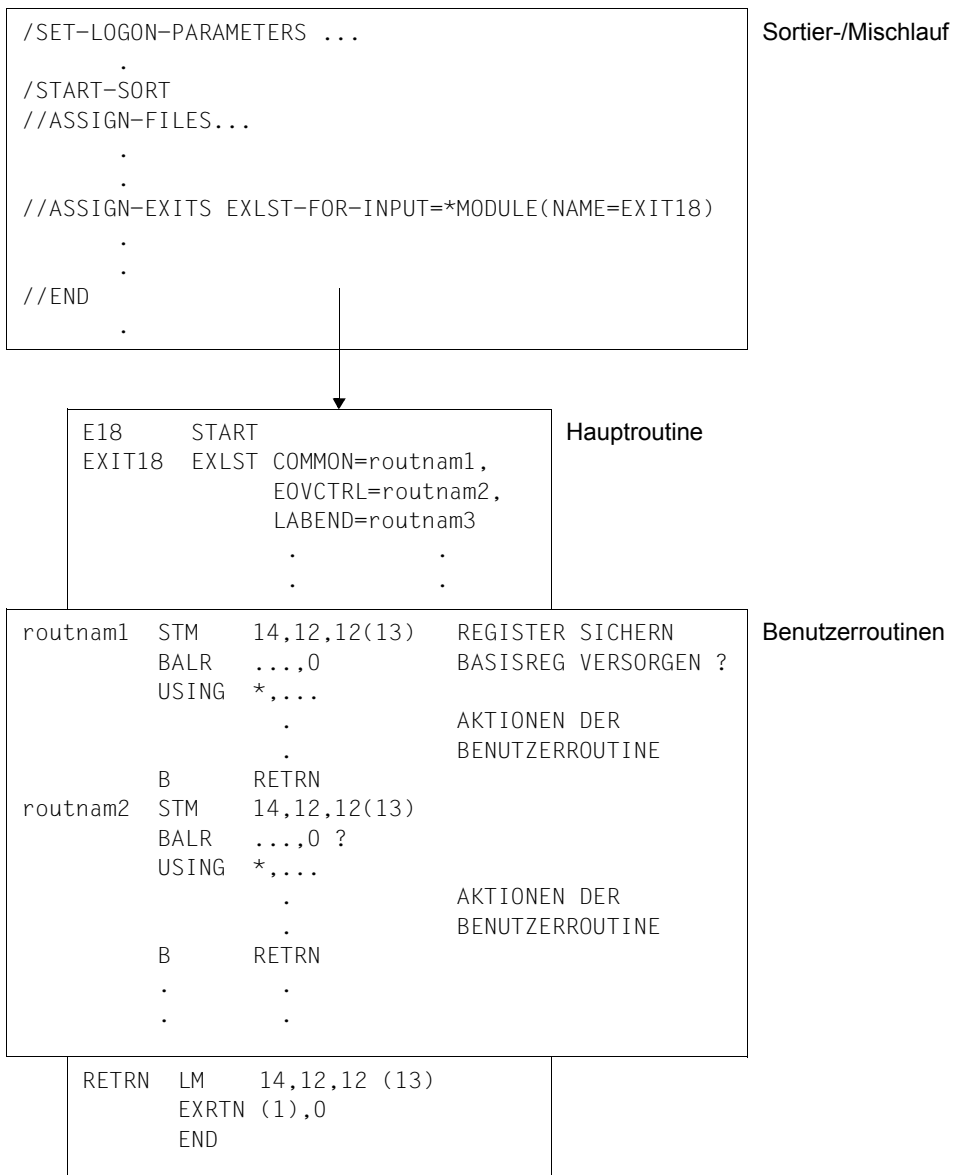
Über den Benutzerausgang EXLST-FOR-INPUT kann der Benutzer für die Eingabedatei bzw. Eingabedateien EXLST-Ausgänge angeben und die darin aufgeführten Routinen entsprechend den DVS-Konventionen SORT zur Verfügung stellen.

Am Anfang dieses Moduls steht ein EXLST-Makro mit den Verweisen auf die Benutzerrou-tinen für die einzelnen benutzten DVS-Ausgänge. Die Operanden des EXLST-Makros, die an das DVS weitergereicht werden, sind in der nachfolgenden Übersicht zusammengestellt (ausführliche Beschreibung siehe „DVS-Makros“ [3]).

DVS-Ausgang	Datenträger	Funktion
COMMON	Band/Platte	Es können alle Ausgänge angegeben werden, die auch mit COMMON abgedeckt sind, außer EOFADDR und USERERR.
EOVCTRL	Band	Der Ausgang schließt nach Bereitstellung eines neuen Datenträgers die Kennsatzverarbeitung ab.
ERRADR	Band/Platte	Die Kontrolle wird an diesen Ausgang übergeben, wenn ein Hardwarefehler auftrat oder eine Ein-/Ausgabe fehlerhaft beendet wurde.
ERROPT	Band/Platte	Der Ausgang schließt für SAM-Dateien Routinen an, die beim Auftreten eines fehlerhaften Blocks ablaufen.
LABEND	Band	Der Ausgang prüft Benutzerkennsätze auf Dateiende (EOF).
LABEOV	Band	Der Ausgang wird zur Prüfung von Benutzer- und Bandenkennsätzen verwendet.
LABGN	Band	Der Ausgang wird verwendet für die Prüfung von Benutzerkennsätzen, die den Daten der Eingabedateien vorausgehen.
OPENV	Band	Der Ausgang gilt für Datenträger mit Nicht-Standardkennsätzen.
PGLOCK	Platte	Die Kontrolle wird an diesen Ausgang übergeben, wenn ein anderer Auftrag Sperrungen veranlasst hat.

Hinweise

- Wird die Steuerung an die Benutzerroutine übergeben, enthält Register 1 die FCB-Adresse der Datei.
- Ein EXLST-Makro darf beim Aufruf von SORT im 24-Bit-Adressierungsmodus sowohl mit PARMOD=24 als auch mit PARMOD=31 übersetzt sein. Beim Aufruf im 31-Bit-Adressierungsmodus ist PARMOD=31 allerdings zwingend notwendig.

Beispiel*Hinweis*

Der Rücksprung mittels EXRTN ist nur für einen Teil der EXLST-Routinen zugelassen (siehe Benutzerhandbuch „DVS-Makros“ [3], Makro EXRTN).

6.6 EXLST-FOR-OUTPUT – EXLST-Ausgang für Ausgabedateien

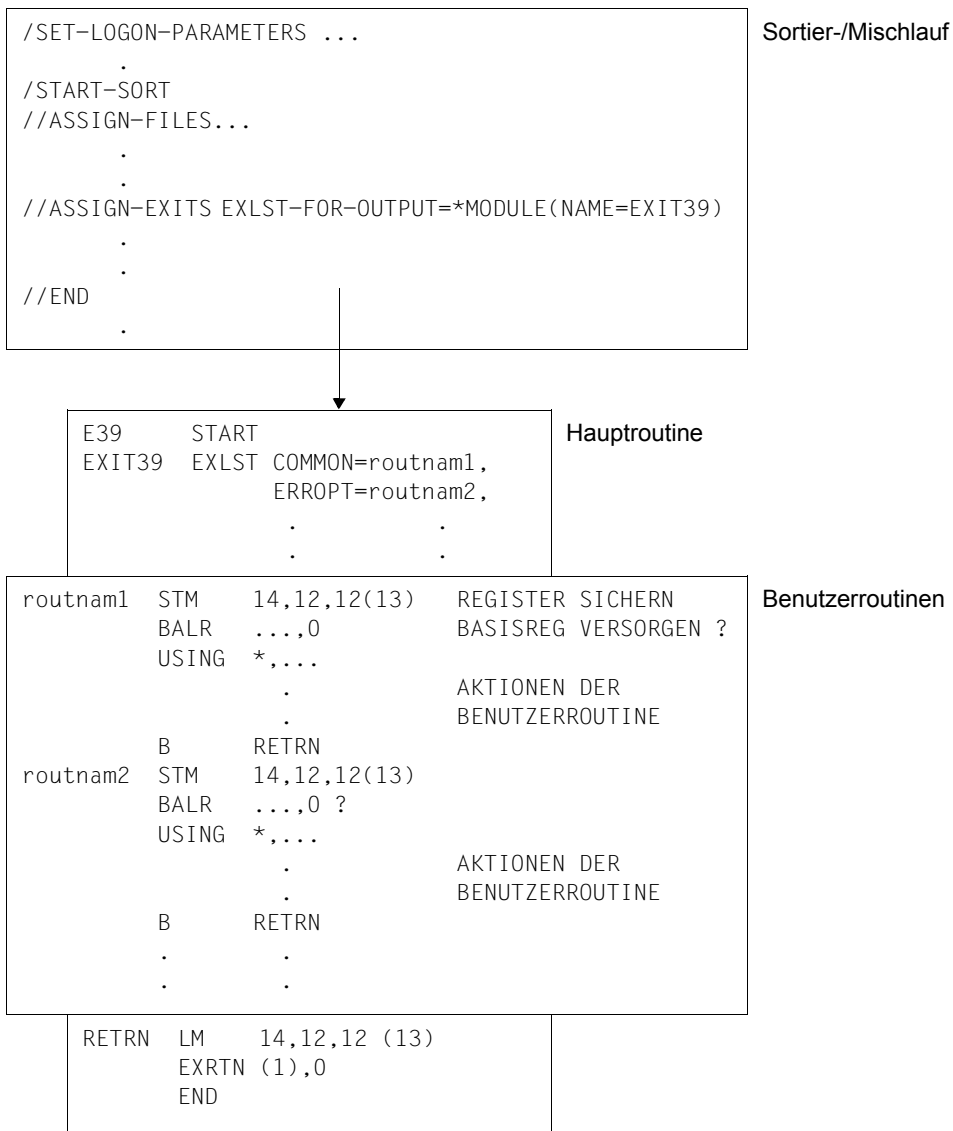
Über den Benutzerausgang EXLST-FOR-OUTPUT kann der Benutzer für die Ausgabedatei EXLST-Ausgänge angeben und die darin angeführten Routinen entsprechend den DVS-Konventionen SORT zur Verfügung stellen.

Am Anfang eines Moduls steht ein EXLST-Makro mit den Verweisen auf die Benutzerroutinen für die einzelnen DVS-Ausgänge. Die Operanden des EXLST-Makros, die für die benutzte Ausgabedatei an das DVS weitergereicht werden, sind in der nachfolgenden Übersicht zusammengestellt (ausführliche Beschreibung siehe „DVS-Makros“ [3]).

DVS-Ausgang	Datenträger	Funktion
COMMON	Band/Platte	Es können alle Ausgänge angegeben werden, die auch mit COMMON abgedeckt sind.
EOVCTRL	Band	Der Ausgang schließt nach Bereitstellung eines neuen Datenträgers die Kennsatzverarbeitung ab.
ERRADR	Band/Platte	Die Kontrolle wird an diesen Ausgang übergeben, wenn ein Hardwarefehler auftrat oder eine Ein-/Ausgabe fehlerhaft beendet wurde.
ERROPT	Band/Platte	Der Ausgang schließt für SAM-Dateien Routinen an, die beim Auftreten eines fehlerhaften Blocks ablaufen.
LABEND	Band	Der Ausgang prüft Benutzerkennsätze auf Dateiende (EOF).
LABEOV	Band	Der Ausgang wird zur Prüfung von Anfangkennsätzen, die bei Ausgabedateien am Ende der Bandspule auftreten, verwendet.
LABGN	Band	Der Ausgang wird für die Prüfung von Benutzerkennsätzen verwendet, die den Daten der Ausgabedateien vorausgehen oder um Benutzeranfangskennsätze (UHL) bei Ausgabedateien zu erstellen.
OPENV	Band	Der Ausgang gilt für Datenträger mit Nicht-Standardkennsätzen.

Hinweise

- Wird die Steuerung an die Benutzerroutine übergeben, enthält Register 1 die FCB-Adresse der Datei.
- Ein EXLST-Makro darf beim Aufruf von SORT im 24-Bit-Adressierungsmodus sowohl mit PARMOD=24 als auch mit PARMOD=31 übersetzt sein. Beim Aufruf im 31-Bit-Adressierungsmodus ist PARMOD=31 allerdings zwingend notwendig.

Beispiel*Hinweis*

Der Rücksprung mittels EXRTN ist nur für einen Teil der EXLST-Routinen zugelassen (siehe Benutzerhandbuch „DVS-Makros“ [3], Makro EXRTN).

6.7 PHYSICAL-TRANSLATE – Sonderzeichen-Umsetzungstabelle

Über diesen Ausgang kann eine Benutzerroutine angeschlossen werden, die die Sortierfelder im PHYSICAL-TRANSLATE-Format umcodiert.

Die Benutzerroutine wird nicht aktiv aufgerufen; sie besteht nur aus zwei Umcodierungstabellen mit je 256 Codezeichen. Die erste Codetabelle, relativ 0 zum Anfang, dient der Hincodierung und die zweite Codetabelle, relativ 256 zum Anfang, wird zur Rückcodierung benutzt.

1. Tabelle Mit dieser Tabelle werden die PHYSICAL-TRANSLATE Formatfelder vor den Vergleichen umcodiert.
2. Tabelle Mit dieser Tabelle werden die Felder, die mit der Tabelle 1 umcodiert wurden, wieder rückcodiert. Auf diese Weise werden die alten Inhalte nach der Verarbeitung wieder hergestellt.

Die Position in den Tabellen bestimmt die Zuordnung von Argument und Funktionswert der Umsetzung.

1. Tabelle: Benutzertabelle

	0	1	2	...	E	F
0	xx	xx	xx	...	xx	xx
1	xx	E2	xx	...	xx	xx
2	xx	xx	xx	...	xx	xx
⋮	⋮	⋮	⋮	⋮	⋮	⋮
E	xx	xx	xx	...	xx	xx
F	xx	xx	xx	...	xx	xx

2. Tabelle: EBCDIC-Tabelle

	0	1	2	...	E	F
0	xx	xx	xx	...	xx	xx
1	xx	xx	xx	...	xx	xx
2	xx	xx	xx	...	11	xx
⋮	⋮	⋮	⋮	⋮	⋮	⋮
E	xx	xx	xx	...	xx	xx
F	xx	xx	xx	...	xx	xx

xx: Zeichen in sedezipalischer Schreibweise

Das Zeichen sedezipal 11 erhält in der Benutzertabelle die Position E2. Alle Bytes der PHYSICAL-TRANSLATE-Felder, die X'11' enthalten, werden in X'E2' übersetzt. Nach der Bearbeitung durch SORT wird X'E2' wieder in X'11' rückcodiert.

Hinweis

Die Umcodierungstabellen müssen sehr sorgfältig erstellt werden, wenn die gewünschte Sortierreihenfolge erreicht werden soll. Dies gilt insbesondere, wenn bei der Rückcodierung der alte Code eindeutig wieder hergestellt werden soll. Einfacher kann eine Reihenfolgeänderung über Umcodierungsformate erreicht werden. Weiterhin gibt es eine einfache Umcodierung über die MODIFY-CODE-Anweisung, oder für spezielle Umcodierungen (mehrere Codezeichen zusammenlegen) über das VIRTUAL-TRANSLATE-Format mit der VIRTUAL-TRANSLATE-Codetabelle.

6.8 VIRTUAL-TRANSLATE – Sonderzeichen-Übersetzungstabelle

Über diesen Ausgang werden Sortierfelder im VIRTUAL-TRANSLATE-Format nach einer umcodierten Wertigkeit sortiert. Im Gegensatz zum PHYSICAL-TRANSLATE-Ausgang werden die Sortierfelder selbst nicht verändert.

Als Benutzerroutine wird eine 256 Byte große Codetabelle angegeben, mit der VIRTUAL-TRANSLATE-Formatfelder für jeden Vergleich auf ein Hilfsfeld umcodiert werden. Von Vorteil ist, dass hierbei keine Rückcodierung nötig ist, und mehrere Codezeichen zusammen auf ein neues umgesetzt werden können.

Die Benutzerroutine wird nicht aktiv aufgerufen und besteht nur aus einer 256 Byte großen Codetabelle, relativ 0 zum Anfang der Routine (Bindemodul).

Beispiel für eine Benutzertabelle:

	0	...	8	9	A	B	C	D	E	F
0	00	...	00	00	00	00	00	00	00	00
1	00	...	0B	14	00	00	0B	14	00	00
2	00	...	0C	15	1D	00	0C	15	1D	00
3	00	...	0D	16	1E	00	0D	16	1E	00
4	00	...	0E	17	1F	00	0E	17	1F	00
5	00	...	0F	18	20	00	0F	18	20	00
6	00	...	10	19	21	00	10	19	21	00
7	00	...	11	1A	22	00	11	1A	22	00
8	00	...	12	1B	23	00	12	1B	23	00
9	00	...	13	1C	24	00	13	1C	24	00
A	00	...	00	00	00	00	00	00	00	00
B	00	...	00	00	00	00	00	00	00	00
C	00	...	00	00	00	00	00	00	00	00
D	00	...	00	00	00	00	00	00	00	00
E	00	...	00	00	00	00	00	00	00	00
F	00	...	00	00	00	00	00	00	00	00

Diese Benutzertabelle bewirkt, dass alle Groß- und Kleinbuchstaben bei der Sortierung gleichwertig behandelt werden. So haben z.B. 'A' (X'C1') und 'a' (X'81') die gleiche Codierung X'0B' in der Benutzertabelle und werden daher bei der Sortierung gleich behandelt (vergleiche Beispiel für VIRTUAL-TRANSLATE, [Seite 359](#)). Aufgebaut werden kann eine solche Tabelle über eine Assembler-CSECT, die dann in der Anweisung ASSIGN-EXITS als Maßnahme MODULE(NAME=<name=1..8>) anzugeben ist.

6.9 EXTERNAL-COMPARE – Reihenfolge durch Benutzerroutine

Über den Ausgang EXTERNAL-COMPARE kann der Benutzer bei jedem Vergleich von Sortierfeldern die auf- oder absteigende Ordnung bestimmen, wenn für sie die Reihenfolge EXTERNAL-COMPARE im FIELDS-Operanden der SORT-RECORDS-Anweisung angegeben wurde.

Der Benutzerausgang EXTERNAL-COMPARE ist für 24- und 31-Bit-Adressierung gültig. Der Aufbau der Schnittstelle richtet sich nach dem Operanden PARAMETER-MODE der ASSIGN-EXITS-Anweisung.

Schnittstelle zur Benutzerroutine bei PARAMETER-MODE=ANY

Die Benutzerroutine wird bei jedem Vergleich von zwei EXTERNAL-COMPARE-Sortierfeldern aufgerufen. Der Benutzerausgang bietet bei 31-Bit-Adressierung eine Erweiterung um den Namen des coded character set, die Adresse des internen Satzformates und die Adresse der internen Sortierfeld-Position.

Wenn das Sortier-/Mischprogramm die Steuerung an die Benutzerroutine übergibt, enthält Register 1 die Adresse eines 32 Byte großen Versorgungsbereichs mit folgendem Aufbau:

Byte 0 - 3: Adresse des EXTERNAL-COMPARE-Sortierfeldes des ersten Vergleichssatzes

Byte 4 - 7: Adresse des EXTERNAL-COMPARE-Sortierfeldes des zweiten Vergleichssatzes

Byte 8 - 11: Adresse der Ordnungsziffer des Sortierfeldes

Die Adresse verweist auf einen 4 Byte langen Datenbereich, in dessen rechtsbündigem Byte die entsprechende Information bereitgestellt wird.

Byte 12 - 15: Adresse der Sortierfeldlänge

Die Adresse verweist auf einen 4 Byte langen Datenbereich, in dessen rechtsbündigem Byte die entsprechende Information bereitgestellt wird.

Byte 16 - 19: Adresse des Aktionswortes

Wenn die Benutzerroutine die Steuerung an SORT zurückgibt, muss im rechtsbündigen Byte des Aktionswortes eines der folgenden Rücksprungkennzeichen mitgegeben werden:

X'00' Das Sortierfeld des 1. Vergleichssatzes hat Vorrang.

X'04' Die beiden Sortierfelder sind gleichrangig.

X'08' Das Sortierfeld des 2. Vergleichssatzes hat Vorrang.

Byte 20 - 23: Adresse des internen Satzformates

Die Adresse verweist auf einen 4 Byte langen Datenbereich, in dessen rechtsbündigem Byte die Information bereitgestellt wird:

X'02' variables Satzformat

X'04' festes Satzformat

Das Satzformat dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Byte 24 - 27: Adresse der internen Sortierfeld-Position

Die Adresse verweist auf einen 4 Byte langen Datenbereich, in dessen beiden rechtsbündigen Bytes die Distanz des Sortierfeldes zum Satzanfang bereitgestellt wird. Die Adresse des Sortierfeldes wird im Versorgungsbereich in Byte 0 - 3 bereitgestellt. Die Sortierfeld-Position dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Byte 28 - 31: Adresse des CCSN

Die Adresse verweist auf einen 8 Byte großen Bereich, in dem SORT den Namen des coded character sets ablegt, mit dem die Datensätze bearbeitet werden.

Der Name dient nur zur Information für den Benutzer. Beim Rücksprung zum Sortierprogramm wird dieses Feld nicht ausgewertet.

Hinweis

Die Benutzeroutine darf die Sortierfelder nicht verändern.

Schnittstelle zur Benutzeroutine bei PARAMETER-MODE=24

Die Benutzeroutine wird bei jedem Vergleich von zwei EXTERNAL-COMPARE-Sortierfeldern aufgerufen.

Register 1 verweist auf den folgenden 8 Byte großen Versorgungsbereich:

Byte 0: Ordnungsziffer des Sortierfeldes

Byte 1 - 3: Adresse des EXTERNAL-COMPARE-Sortierfeldes des ersten Vergleichssatzes

Byte 4: Sortierfeldlänge

Byte 5 - 7: Adresse des EXTERNAL-COMPARE-Sortierfeldes des zweiten Vergleichssatzes

Die Benutzeroutine gibt die Steuerung an SORT zurück, nachdem sie die beiden EXTERNAL-COMPARE-Sortierfelder verglichen hat. Sie muss im rechtsbündigen Byte von Register 15 eines der folgenden Rücksprungkennzeichen übergeben:

X'00' Das Sortierfeld des 1. Vergleichssatzes hat Vorrang.

X'04' Die beiden Sortierfelder sind gleichrangig.

X'08' Das Sortierfeld des 2. Vergleichssatzes hat Vorrang.

Hinweis

Die Benutzeroutine darf die Sortierfelder nicht verändern.

6.10 TRANSLATE-CHARACTER - Reihenfolge durch Gleichsetzungstabellen und codierten Zeichensatz

Über diesen Ausgang kann eine Benutzeroutine angeschlossen werden, die die Sortierfelder im TRANSLATE-CHARACTER-Format umcodiert, wenn der CCSN der Eingabedatei dem Namen des angegebenen Moduls entspricht. Die Benutzeroutine besteht aus zwei Umcodierungstabellen mit je 256 Codezeichen. Aus diesen Tabellen generiert SORT den Code des zu sortierenden Zeichens für den Sortiervorgang.

Diese Tabellen haben Vorrang vor den von SORT gelieferten Standard-Tabellen in der Bibliothek mit dem logischen Namen SYSLNK.TAB. Ansonsten werden die Tabellen wie Standardtabellen behandelt. Es ist nicht erforderlich, dass eine Standard-Tabelle mit gleichem Namen existiert. Der CCSN muss aber XHCS bekannt sein.

Hat die zu sortierende Datei einen CCSN der nicht in der Anweisung steht, aber in der Bibliothek mit dem logischen Namen SYSLNK.TAB vorhanden ist, wird auf die Datei in dieser Bibliothek zugegriffen.

6.11 INT – Unterbrechung des Sortier-/Mischlaufs

Der Ausgang INT ermöglicht dem Benutzer von einer Datenstation aus den Dialog mit dem Sortier-/Mischprogramm SORT. Dieser Ausgang kann nicht in der ASSIGN-EXITS-Anweisung angegeben werden. Dies bedeutet, dass keine Benutzeroutine angeschlossen werden kann.

Die Aktivierung des Ausganges INT erfolgt

- im Dialogbetrieb durch Wechsel vom Programmmodus (SORT) in den Systemmodus durch Betätigung der K2-Taste an der Datensichtstation, von der aus SORT gestartet wurde. Nachdem das System den Schrägstrich ausgegeben hat, kann das Kommando /INFORM-PROGRAM gegeben werden.
- im Batchbetrieb durch Eingabe von /INFORM-PROGRAM JOB-ID=*TSN(TSN=tsn) von der Konsole (tsn ist die Nummer des Prozesses, in dem SORT läuft).

Danach gibt eine Meldung auf der Datenstation Auskunft über die verbrauchte Lauf- und CPU-Zeit und verlangt eine der folgenden Maßnahmen:

- D**[ISPLAY] Der Stand der Verarbeitung soll ausgegeben werden, z.B. Anzahl der bisher bearbeiteten Sätze.
- CK**[PT] Zum nächstmöglichen Zeitpunkt soll ein Fixpunkt ausgegeben werden. Wird z.B. anschließend die Maßnahme TERMINATE getroffen, schreibt SORT noch vor dem Abbruch des Sortier-/Mischlaufs einen Fixpunkt.
- C**[ONTINUE] Die Verarbeitung wird fortgesetzt.
- F**[INISH] SORT soll die Satzeingabe beenden und die bereits übernommenen Sätze bearbeiten (sortieren/mischen).
- T**[ERMINATE] Der SORT-Lauf soll beendet werden.

Hinweise

- Wurde eine DISPLAY- oder CKPT-Maßnahme ausgeführt, wird eine weitere Maßnahme angefordert. Konnte nach den Maßnahmen CKPT und TERMINATE kein Fixpunkt geschrieben werden, gibt SORT eine Fehlermeldung aus und setzt die Verarbeitung fort.
- SORT ist über den Benutzerausgang INT nicht ansprechbar, wenn bei der SORTZM oder beim Aufruf von SORT als Unterprogramm beim *ersten* Makroaufruf SRT0 oder SRT1 der Operand STXIT=NO angegeben wird.

Beispiel

```
/SET-LOGON-PARAMETERS ...
```

```
.
```

```
.
```

```
.
```

```
/START-SORT
```

```
.
```

```
.
```

```
.
```

```
//SORT-RECORDS...
```

```
.
```

```
.
```

```
//END
```

```
.
```

```
.
```

```
.
```

← K2-Taste bzw. ESCAPE/BREAK-Taste

```
/SEND-MESSAGE TO=PROGRAM
```

Eingabe der Maßnahmen, z.B. CKPT

```
.
```

```
.
```

```
.
```

7 Fixpunktverarbeitung

Fixpunkte werden von SORT bei Sortier-/Mischläufen ausgegeben, wenn entweder

- der Operand CHECKPOINT in der SORT-RECORDS- bzw. MERGE-RECORDS-Anweisung angegeben wurde oder
- im Dialogbetrieb zum nächstmöglichen Zeitpunkt, wenn ein SEND-MESSAGE-Aufruf erfolgte und eine CKPT-Maßnahme verlangt wurde.

Keine Fixpunkte werden geschrieben, wenn SORT als Unterprogramm aufgerufen wurde und die Einschränkungen des CHKPT-Makros nicht beachtet werden. (Benutzung eines Stacks, Speicherpools oder der Interprozesskommunikation, siehe „Makroaufrufe an den Ablaufteil“ [6], WRCPT-Makro.)

Bei der Verarbeitung von POSIX-Dateien als Eingabedateien können ebenfalls keine Fixpunkte eingerichtet werden.

Beim Multitasksortieren sind Fixpunkte nur vor dem letzten Mischdurchlauf mit der Endausgabe möglich.

Sortierläufe

Während eines Sortierlaufs werden Fixpunkte nur am Ende eines Zyklus geschrieben. Ein Zyklus ist beendet, nachdem SORT eine sortierte Teilmenge auf eine Hilfsdatei ausgegeben hat. Beim letzten Zyklus können unmittelbar vor dem Mischdurchlauf mit der Endausgabe Fixpunkte geschrieben werden.

Unabhängig vom Ablaufzustand kann ein Fixpunkt ausgegeben werden, wenn ein SEND-MESSAGE-Aufruf erfolgte und nach der Eingabe einer CKPT-Maßnahme der SORT-Lauf mit TERMINATE beendet wurde. Dies gilt jedoch nicht für das Multitasksortieren.

Mischläufe

Während eines Mischlaufs können Fixpunkte in beliebig dichter Reihenfolge geschrieben werden. Wurde in der MERGE-RECORDS-Anweisung der Operand CHECKPOINT angegeben, kann über den RECORDS-PER-CYCLE-Operanden der MERGE-RECORDS-Anweisung festgelegt werden, nach welcher Anzahl von Mischeingabesätzen ein Fixpunkt ausgegeben werden soll.

Im Dialogbetrieb kann über ein SEND-MESSAGE-Kommando und eine CKPT-Maßnahme jederzeit veranlasst werden, dass zum nächstmöglichen Zeitpunkt ein Fixpunkt geschrieben wird. Die Fixpunktdatei muss allerdings genügend groß sein.

Pamkey-Eliminierung

Für Fixpunktdateien darf der Wert BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK nicht angegeben werden.

RESTART-PROGRAM

Durch das RESTART-PROGRAM-Kommando kann ein abgebrochener Sortier-/Mischlauf an dem zuletzt geschriebenen Fixpunkt wieder fortgesetzt werden. Hierbei ist jeweils der Name der Fixpunktdatei und die PAM-Seite anzugeben, die bei jedem Fixpunkt protokolliert wird.

8 Optimierung von Sortierläufen

Sortierläufe können optimiert werden durch

- geeignete CORE-Zuweisung
- virtuelles Mischen
- Wahl des Sortierverfahrens (Umcodierung, Zyklen-, Multitasksortieren)
- geeignete Wahl der Dateieigenschaften
- Satzsummierung
- SORT als Subsystem laden
- Angabe des OPTIMIZATION-Operanden in der SET-SORT-OPTIONS-Anweisung
- Ändern der Voreinstellungen für SORT.

8.1 Geeignete CORE-Zuweisung

Der CORE-Wert bestimmt die Größe des intensiv genutzten virtuellen Speichers.

Bei normaler Auslastung des Systems sollte die CORE-Zuweisung SORT überlassen werden. SORT berechnet den CORE-Wert über die Sortierzyklussatzmenge. Die entsprechenden Informationen über die Berechnung einer Sortierzyklussatzmenge müssen SORT zur Verfügung stehen (RECORDS-PER-CYCLE- oder ESTIMATED-RECORDS-Angabe in Verbindung mit der Definition der Hilfsdateien über ADD-FILE-LINK-Zuordnung und/oder TAPE-UNITS-Angabe).

Berechnung des CORE-Werts durch SORT:

$$\text{CORE} = \frac{\text{Zyklusdatenmenge in Byte}}{2^{**} 20} + 16$$

SORT gibt im Dialog- und Stapelbetrieb den errechneten Wert für CORE über die SORT-Meldungen SRT1033 aus, wenn in der ASSIGN-EXITS-Anweisung PLANNING=*DIALOG angegeben wurde.

In einem System mit geringer Auslastung oder einer Anlage mit sehr großem Speicherausbau kann die Sortierung beschleunigt werden, wenn über die ASSIGN-RESOURCES-Anweisung ein CORE-Wert zugewiesen wird, der größer als der von SORT berechnete ist.

Bei sehr stark belasteten Systemen kann eventuell eine Verbesserung erreicht werden, wenn über die ASSIGN-RESOURCES-Anweisung ein CORE-Wert zugewiesen wird, der niedriger ist als der von SORT berechnete.

Den von SORT berechneten CORE-Wert kann man durch Angabe des Operanden MEMORY-SIZE mit der ASSIGN-RESOURCES-Anweisung modifizieren. Folgende Prioritätsklassen sind möglich:

MEMORY-SIZE =	Bedeutung
MIN	Minimalwert. Entspricht dem CORE-Wert, der mit der Anweisung MODIFY-SORT-DEFAULTS und dem Operanden CORE-MINIMUM kennungs- oder anlagenspezifisch voreingestellt werden kann (Voreinstellung bei Auslieferung von SORT: 24).
SMALL	Unterer Wert. Entspricht dem von SORT berechneten CORE-Wert - 33%.
STD	Von SORT berechneter Wert. SORT berechnet diesen Wert aus den Angaben zum Sortier-/Mischlauf.
LARGE	Oberer Wert. Entspricht dem von SORT berechneten CORE-Wert + 33%.
MAX	Maximalwert. Entspricht dem CORE-Grenzwert, der mit der Anweisung MODIFY-SORT-DEFAULTS und dem Operanden CORE-LIMIT kennungs- oder anlagenspezifisch voreingestellt werden kann (Voreinstellung bei Auslieferung von SORT: 96).

Mit der Angabe MIN-MSG-WEIGHT=*ALL der Anweisung SET-SORT-OPTIONS erhält der Anwender Informationen darüber, wie sein Sortierlauf ausgelegt ist:

- Bei kleinen Eingabedateien (z.B. 1000 Sätze mit je 50 Byte Länge) wird angestrebt, alle Sätze sofort im Hauptspeicher zu sortieren, ohne dass eine Zwischenspeicherung in einer SORTWK-Datei erfolgt. Darüber informiert die Meldung SRT1013.
- Für größere Eingabedateien wird versucht, die beim Vorsortieren entstandenen Folgen in einem Endmischlauf zusammen zu führen. Dies wird dem Benutzer mit der Meldung SRT1012 mitgeteilt. Erst bei großen Sortierläufen sind auch interne Zwischenmischläufe erforderlich.
- Bei Nutzung des Benutzerausgangs „PLANNING“ mit DIALOG gibt die Meldung SRT1061 an, wie viele Mischwege im Endmischen maximal vereinigt werden können.
- Nach Beendigung der DOMINO-Phase gibt die Meldung SRT1028 an, wie viele Folgen tatsächlich entstanden sind. Durch Ändern von MEMORY-SIZE kann eventuell eine günstigere Konstellation erreicht werden. Allerdings ändert sich die Zahl der tatsächlich erzeugten Folgen normalerweise von Sortierlauf zu Sortierlauf, weil sich die Eingabedaten ändern.

8.2 Virtuelles Mischen

Die zu sortierenden Datensätze müssen zur Durchführung von Vergleichen in einen virtuellen Speicherbereich gebracht werden, dessen Größe vom Anwender im Operanden MEMORY-SIZE der ASSIGN-RESOURCES-Anweisung spezifiziert bzw. von SORT aus Angaben zur Datenmenge berechnet wird. Auf diesen Bereich wird in sehr stark streuender Form zugegriffen, sodass es mit zunehmender Größe dieses Speicherbereiches zu einer Verstärkung des Pagings kommen kann. Dadurch können konkurrierende Task-Abläufe betroffen werden, was auch die Ablaufzeit des SORT-Laufs beeinflussen kann.

Beim Vorsortieren mit virtueller Mischung wird der Speicherbereich im Verhältnis 1 zu 15 in einen Vorsortierungsbereich und in einen Mischbereich aufgeteilt. Dadurch wird der streuende Zugriff auf 1/16 beschränkt, der Rest (Mischbereich) wird in sequenzieller Form angesprochen.

Voraussetzung für eine virtuelle Mischung ist die Angabe eines MEMORY-SIZE-Wertes ≥ 400 in der ASSIGN-RESOURCES-Anweisung und die Mindestverfügbarkeit von 400 Seiten Klasse-6-Speicher. Außerdem darf der intensiv genutzte Speicher (1/16 des angegebenen Wertes) den Maximalwert, der mit dem Operanden CORE-MAXIMUM der Anweisung MODIFY-SORT-DEFAULTS eingestellt werden kann, nicht überschreiten. Ist dies der Fall, wird die Größe des intensiv genutzten Speichers auf diesen Maximalwert begrenzt und die gesamte Speicheranforderung neu berechnet (durch Multiplikation mit 16). Nur wenn dieser neu berechnete Wert ≥ 400 ist, kann eine virtuelle Mischung durchgeführt werden.

Beispiel

CORE-Wert = 1600, CORE-MAXIMUM = 50

Der intensiv zu nutzende Speicherbereich wird mit $1600/16 = 100$ ermittelt. Da die voreingestellte Begrenzung kleiner ist (50), wird deren Wert verwendet. Der endgültige CORE-Wert und der damit angeforderte virtuelle Speicher wird durch die Rückrechnung (Multiplikation der intensiv genutzten Speichergröße mit 16) auf 800 Seiten ($= 50 * 16$) reduziert.

Mit folgenden Meldungen werden Informationen über die virtuelle Mischung ausgegeben:

Mit der Meldung SRT1033 wird die Größe des angeforderten Speichers angegeben.

Die Meldung SRT1062 wird ausgegeben, wenn eine virtuelle Mischung möglich ist.

Die beiden Meldungen werden nur ausgegeben, wenn in der ASSIGN-EXITS-Anweisung beim Benutzerausgang PLANNING die Maßnahme DIALOG angegeben wurde und in der SET-SORT-OPTIONS-Anweisung die entsprechende Meldungspriorität gesetzt wurde.

8.3 Wahl des Sortierverfahrens

Bei Sortierungen kann die Laufzeit und der Systemdurchsatz beeinflusst werden durch Verwendung von

- Umcodierung
- Zyklensortieren
- Multitasksortieren.

8.3.1 Umcodierung

Durch Umcodierungen entsteht ein erhöhter CPU-Zeitbedarf. Die folgende Aufstellung soll, soweit eine Wahlmöglichkeit besteht, eine Hilfestellung bei der Wahl des Umcodierungsformates geben.

- Geringer Aufwand bei Hin- und Rückcodierung (Formate MODIFY-CODE, PHYSICAL-TRANSLATE und EBCDIC-ISO-EBCDIC)
- Großer Aufwand bei Umcodierung pro Satzvergleich auf Hilfsspeicher (Formate VIRTUAL-TRANSLATE und EBCDIC-INTERNATIONAL)
- Sehr großer Aufwand bei DIN-orientierter Umcodierung mit spezieller Behandlung der Umlaute (Formate EBCDIC-DIN).

8.3.2 Zyklensortieren

Beim Zyklensortieren wird die zu sortierende Datenmenge in Teilmengen (Zyklen) aufgeteilt. SORT sortiert jede Teilmenge für sich auf der Arbeitsdatei (SORTWKx) und lagert sie anschließend auf eine Hilfsdatei (SORTWKxx) aus. Die Hilfsdateien können Band- oder Plattendateien sein.

Das Zyklensortieren dient im Wesentlichen zum gesteuerten Erzeugen von Fixpunkten (RESTART-Fähigkeit). Soweit die Hilfsdateien auf getrennten (eigenen) Datenträgern liegen, kann über verminderte Positionierzeiten eventuell etwas Laufzeit eingespart werden. Die CPU-Zeiten werden jedoch immer größer sein als bei einer einfachen Sortierung mit einem Zyklus.

Eine Zyklensortierung wird nur durchgeführt, wenn der Operand RECORDS-PER-CYCLE angegeben wird und dieser kleiner ist als die Anzahl der Eingabesätze.

Für das Zyklensortieren benötigt SORT als Betriebsmittel:

- 1 Plattenarbeitsdatei der Größe $1,1 \cdot \text{Zyklusdatenmenge}$.
- n Hilfsdateien (Platte und/oder Band), wobei $n = \text{Zyklusanzahl} - 1$ ist. Jede der Hilfsdateien hat die Größe $1,1 \cdot \text{Zyklusdatenmenge}$.

Die Anzahl der Hilfsdateien ergibt sich aus dem Maximum von

Anzahl der über ADD-FILE-LINK-Kommandos mit LINK-NAME=SORTWKxx zugeordneten Dateien, ergänzt durch den TAPE-UNITS-Operanden (ASSIGN-RESOURCES-Anweisung)

und

$$\frac{\text{ESTIMATED-RECORDS}}{\text{RECORDS-PER-CYCLE-Wert}} - 1$$

Hierbei haben die Angaben im FILE-Kommando Vorrang.

Hinweis

Entsprechen die durch ADD-FILE-LINK-Kommando mit LINK-NAME=SORTWKxx und TAPE-UNITS-Operanden definierten Hilfsdateien nicht der sich aus ESTIMATED-RECORDS-Wert/RECORDS-PER-CYCLE-Wert ermittelten Zyklusanzahl-1, richtet SORT die fehlenden Dateien als Plattendateien ein.

Reichen diese Hilfsdateien nicht aus, richtet SORT von sich aus eine zusätzliche Plattenhilfsdatei ein. Genügt diese zusätzliche Plattenhilfsdatei nicht, versucht SORT über eine Erweiterung der Arbeitsdatei (Sekundärzuweisung) den Rest der Eingabe zu sortieren.

Die Satzanzahl pro Zyklus wird auf eine der folgenden Arten errechnet:

$$\text{Zyklussatzanzahl} = \text{RECORDS-PER-CYCLE-Wert}$$

oder ersatzweise gilt auch:

$$\text{Zyklussatzanzahl} = \frac{\text{INPUT-RANGE, NUMBER-OF-RECORDS-Angabe}}{\text{Anzahl der Hilfsdateien} + 1}$$

Beispiel

```

/add-file-link file-name=input.1,link-name=sortin
/add-file-link file-name=output.1,link-name=sortout
/start-sort
% SRT1001 2014-10-12/15:41:58/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit(position=5,length=8, -
//                               sorting-order=*descending), -
//                               records-per-cycle=90
//set-sort-options min-msg-weight=*all
//end
% SRT1046 2014-10-12/15:42:27/000000.25 END OF PREPARATORY PHASE
% SRT1010 2014-10-12/15:42:27/000000.27 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1015 2014-10-12/15:42:28/000000.30 END OF CYCLE .....1
% SRT1010 2014-10-12/15:42:28/000000.30 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1015 2014-10-12/15:42:28/000000.33 END OF CYCLE .....2
% SRT1010 2014-10-12/15:42:28/000000.33 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1015 2014-10-12/15:42:28/000000.36 END OF CYCLE .....3
% SRT1010 2014-10-12/15:42:28/000000.36 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1016 SORT/MERGE INPUT RECORDS:.....300 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....300
% SRT1025 PAM INPUTS:.....8
% SRT1026 PAM OUTPUTS:.....8
% SRT1002 2014-10-12/15:47:26/000000.48 SORT/MERGE COMPLETED

```

Die Hilfsdateien werden von SORT angelegt und am Ende des Laufes wieder gelöscht.

8.3.3 Multitasksortieren

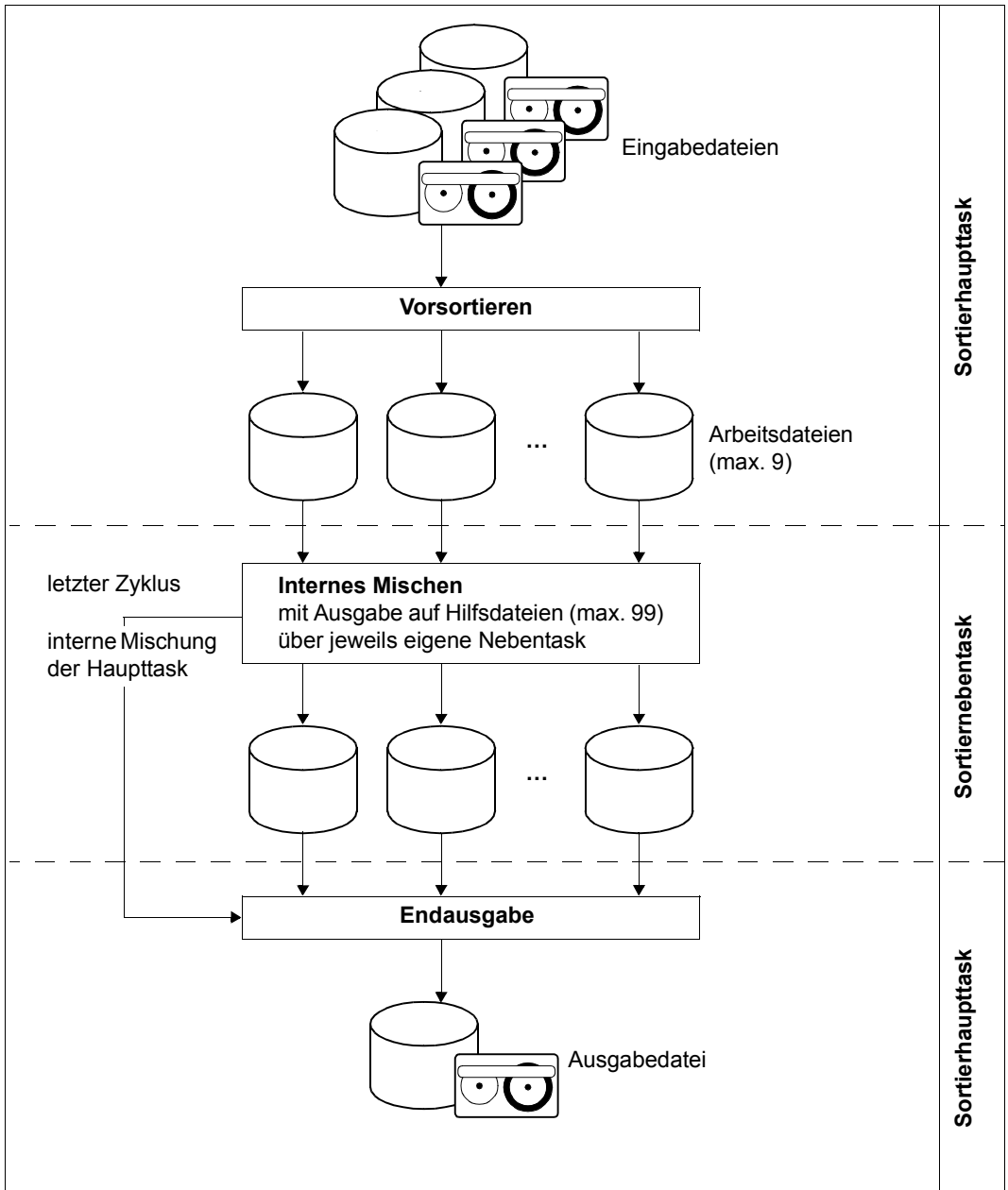


Bild 11: Multitasksortieren

Beim Multitasksortieren wird mit mehreren Arbeitsdateien (2 bis max. 9) gearbeitet. Dieses Sortierverfahren ist ein Zyklensortieren mit mehreren Prozessabläufen, und zwar einer Sortierhaupttask und bis zu 8 Sortiernebantasks. Die Anzahl der Tasks (eine Haupttask und mehrere Nebentasks), die gleichzeitig laufen können, ist gleich der Anzahl der Arbeitsdateien. Sie bestimmt den Grad der Parallelarbeit von SORT. Die Anzahl der insgesamt gestarteten Nebentasks wird vorrangig durch das Resultat von (ESTIMATED-RECORDS/RECORDS-PER-CYCLE) bestimmt. Fehlen Werte dazu, verwendet SORT die Anzahl der Hilfsdateien.

Das Multitasksortieren führt bei richtiger Anwendung zu teilweise erheblichen Laufzeiteinsparungen. Da aber der CPU-Zeitbedarf gegenüber dem einfachen Sortieren mit einem Zyklus und auch noch gegenüber dem Zyklensortieren stets größer ist, kann eine Durchsatzsteigerung nur in einem gering belasteten System mit genügender CPU-Zeitreserve erzielt werden.

Sortierhaupttask

Die Sortierhaupttask

- sortiert die Ersteingabe vor,
- gibt Teilmenge auf Arbeitsdateien aus,
- koordiniert die Nebentaskmischläufe,
- mischt beim letzten Sortierzyklus die Hilfsdateien ein, die von den Nebentasks beschrieben wurden.

In der Sortierhaupttask wird die Ersteingabe vorsortiert. Die Arbeitsdateien werden danach zum internen Mischen mit Ausgabe auf eine Hilfsdatei jeweils an eine Sortiernebantask abgegeben. Die Sortierhaupttask mischt die so erzeugten Hilfsdateien beim letzten Zyklus, bei dem auch die interne Mischung von der Sortierhaupttask durchgeführt wird, in die Endausgabe ein.

Die Sortierhaupttask und die Sortiernebantasks kommunizieren miteinander und synchronisieren so die Abläufe. So muss z.B. die Sortierhaupttask, wenn nötig, warten, bis eine Arbeitsdatei frei wird oder bis alle Nebentaskabläufe abgeschlossen sind.

Das Vorsortieren in der Haupttask und das interne Mischen in der Nebentask laufen parallel ab.

Sortiernebantask

In den Sortiernebantasks erfolgt eine Optimierung der Sortierfolgen (sortierte Sequenzen) mithilfe von Stellvertretern (sonst in der Sortierhaupttask). Die einzelnen Teilmengen (Sortierfolgen) werden intern gemischt und auf Hilfsdateien ausgegeben.

Als Betriebsmittel erfordert das Multitasksortieren

- Eingabedatei (max. 99)
- Plattenarbeitsdateien (2 bis max. 9)
- Plattenhilfsdateien (max. 99)
- Ausgabedatei.

Hinweis

Der Operand MIN-MSG-WEIGHT der Anweisung SET-SORT-OPTIONS steuert die SYSLST-Ausgabe der Sortiernebantasks. Bei Angabe von MIN-MSG-WEIGHT=*ALL wird das SYSLST-Protokoll stets ausgegeben, sonst nur bei Fehlern in der Sortiernebantask.

Voraussetzungen für das Multitasksortieren

Für das Multitasksortieren sollten die nachfolgenden Bedingungen erfüllt sein. Ist dies nicht der Fall, sollte mit nur einer Arbeitsdatei zyklweise sortiert werden. Bei Unklarheiten bei der Angabe von Werten (z.B. Satzlänge, ESTIMATED-RECORDS) ist es besser, keine Angaben zu machen, da falsche Angaben zu einem Abbruch führen können.

- SORT muss mindestens 2, maximal 9 Arbeitsdateien zugewiesen bekommen. Daraus erkennt SORT, dass eine Multitask-Sortierung gefordert ist.
- Die Benutzerkennung, unter der sortiert wird, muss einen „Expresseintrag“ haben (SHOW-USER-ATTRIBUTES-Kommando, Anzeige START-IMMED=YES).
- Werden ESTIMATED-RECORDS und RECORDS-PER-CYCLE angegeben, berechnet SORT die Anzahl der Hilfsdateien (maximal 99).
- Die Auslastung des Systems sollte so gering sein, dass Nebentasks, die die interne Mischung besorgen, sofort gestartet und ohne Behinderung ablaufen können.
- Hilfsdateien auf Arbeitsbändern dürfen beim Multitasksortieren nicht verwendet werden. Wenn der TAPE-UNITS-Parameter in der ASSIGN-RESOURCES-Steueranweisung ungleich Null angegeben ist, wird automatisch nur ein Zyklensortieren durchgeführt.

- Geräte, auf denen private Platten mit Arbeits- oder Hilfsdateien verwendet werden, müssen mehrfachbenutzbar sein.
- Keine Verwendung der Benutzerausgänge VIRTUAL-TRANSLATE und/oder EXTERNAL-COMPARE.

Wenn weder ein ESTIMATED-RECORDS-Wert noch ein RECORDS-PER-CYCLE-Wert angegeben wurden, berechnet SORT einen RECORDS-PER-CYCLE-Wert aus der Eingabedatenmenge (Größe der Eingabedatei in PAM-Seiten), der durchschnittlichen Satzlänge und der Anzahl der Hilfsdateien. Bei Banddateien kann SORT die Eingabedatenmenge nicht feststellen und somit keinen RECORDS-PER-CYCLE-Wert berechnen. Daher wird in diesem Fall kein Multitasksortieren durchgeführt.

Besondere Fehlersituationen

Die nachfolgend beschriebenen Fehlersituationen werden von SORT durch die Ausgabe von Meldungen behandelt. Nötigenfalls wird der Sortierlauf abgebrochen.

- Der „Expreseintrag“ fehlt in der Benutzerkennung:
Es wird die Warnung SRT1065 ausgegeben; der Sortierlauf wird als Zyklensortierung in der Haupttask normal fortgesetzt.
- Der „Expreseintrag“ wird während des Sortierlaufs entzogen:
Es wird die Warnung SRT1066 ausgegeben; der Sortierlauf wird als Zyklensortierung in der Haupttask normal fortgesetzt.
- Sortiernebtasks werden mit „CANCEL-JOB“ abgebrochen:
Es werden die Meldungen SRT1051, SRT1052 und SRT1053 ausgegeben; der Sortierlauf wird fortgesetzt. Beim letzten Sortierzyklus (Haupttask) werden die Hilfsdateien der abgebrochenen Hilfstasks nicht berücksichtigt. Der Sortier-/Mischlauf wird mit der Meldung SRT1059 regulär beendet.
- Sortiernebtasks werden nicht gestartet:
Nach Ablauf einer Wartezeit von 10 Minuten wird die Meldung SRT1068 ausgegeben; der Sortierlauf wird abgebrochen.
- Sonstige Fehlerzustände von Sortiernebtasks:
Beim Endmischen werden in diesem Fall die Meldungen SRT1067, SRT1052 und SRT1053 ausgegeben; der Sortierlauf wird fortgesetzt. Beim letzten Sortierzyklus (Haupttask) werden die Hilfsdateien der betroffenen Nebentasks nicht berücksichtigt. Der Sortier-/Mischlauf wird mit der Meldung SRT1059 regulär beendet.

Beispiele

Variante 1

```

/add-file-link link-name=sortin,file-name=input.1
/add-file-link link-name=sortout,file-name=output.1
/add-file-link link-name=sortwk1,file-name=work.1
/add-file-link link-name=sortwk2,file-name=work.2
/start-sort
% SRT1001 2014-10-12/13:50:09/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit(position=5,length=10), -
//          estimated-records=1000,records-per-cycle=300
//set-sort-options min-msg-weight=*all
//end
% SRT1046 2014-10-12/13:50:10/000000.24 END OF PREPARATORY PHASE
% SRT1010 2014-10-12/13:50:11/000000.30 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:50, TSN = 8D2R
% SRT1010 2014-10-12/13:50:13/000000.41 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:50, TSN = 8D2U
% SRT1010 2014-10-12/13:51:00/000000.51 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:51, TSN = 8D3B
% SRT1010 2014-10-12/13:51:01/000000.59 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1016 SORT/MERGE INPUT RECORDS:.....2.000 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....2.000
% SRT1025 PAM INPUTS:.....2
% SRT1026 PAM OUTPUTS:.....11
% SRT1002 2014-10-12/13:51:39/000000.72 SORT/MERGE COMPLETED

```

Es werden insgesamt 3 Nebentasks gestartet, die jeweils 600 Sätze auf eine Hilfsdatei sortieren. Die Anzahl der Hilfsdateien wird von SORT berechnet (ESTIMATED-RECORDS/RECORDS-PER-CYCLE). Die letzten 200 Sätze werden von der Haupttask sortiert und zusammen mit den drei Hilfsdateien auf die Ausgabedatei gemischt.

Variante 2

```

/add-file-link link-name=sortin,file-name=input.1
/add-file-link link-name=sortout,file-name=output.1
/add-file-link link-name=sortwk1,file-name=work.1
/add-file-link link-name=sortwk2,file-name=work.2
/add-file-link link-name=sortwk01,file-name=auxiliary.1
/add-file-link link-name=sortwk02,file-name=auxiliary.2
/add-file-link link-name=sortwk03,file-name=auxiliary.3
/start-sort
% SRT1001 2014-10-12/13:46:36/000000.00 SORT/MERGE STARTED,
VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit(position=5,length=10), -
// estimated-records=2000
//set-sort-options min-msg-weight=*all
//end
% SRT1046 2014-10-12/13:46:37/000000.25 END OF PREPARATORY PHASE
% SRT1010 2014-10-12/13:46:37/000000.31 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:46, TSN = 8DZB
% SRT1010 2014-10-12/13:46:38/000000.42 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:46, TSN = 8DZC
% SRT1010 2014-10-12/13:47:35/000000.51 END OF PRESORT PHASE
% JMS0066 JOB ,EXAMPLE' ACCEPTED ON 96-02-05 AT 13:47, TSN = 8DZI
% SRT1010 2014-10-12/13:47:36/000000.60 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....1
% SRT1028 STRINGS AFTER DOMINO:.....1
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1016 SORT/MERGE INPUT RECORDS:.....2.000 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....2.000
% SRT1025 PAM INPUTS:.....3
% SRT1026 PAM OUTPUTS:.....12
% SRT1002 2014-10-12/13:48:27/000000.70 SORT/MERGE COMPLETED

```

SORT berechnet auf Grund der Angabe bei ESTIMATED-RECORDS und der Anzahl der zugewiesenen Hilfsdateien einen RECORDS-PER-CYCLE Wert. Die Anzahl der Nebentasks ist gleich der Anzahl der zugewiesenen Hilfsdateien.

Variante 3

```

/add-file-link file-name=input.1,link-name=sortin
/add-file-link file-name=output.1,link-name=sortout
/add-file-link file-name=work.1,link-name=sortwk1
/add-file-link file-name=work.2,link-name=sortwk2
/start-sort
% SRT1001 2014-10-12/12:10:46/000000.00 SORT/MERGE STARTED, VERSION
08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit(position=5,length=10)
//set-sort-options min-msg-weight=*all
//end
% SRT1046 2014-10-12/12:10:47/000000.24 END OF PREPARATORY PHASE
% SRT1010 2014-10-12/12:10:49/000000.41 END OF PRESORT PHASE
% SRT1027 STRINGS AFTER PRESORTING:.....2
% SRT1028 STRINGS AFTER DOMINO:.....2
% SRT1012 NO INTERNAL MERGE NECESSARY
% SRT1016 SORT/MERGE INPUT RECORDS:.....2.000 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....2.000
% SRT1025 PAM INPUTS:.....11
% SRT1026 PAM OUTPUTS:.....11
% SRT1002 2014-10-12/12:10:51/000000.51 SORT/MERGE COMPLETED

```

Da SORT keinen CYCLE-WERT berechnen kann und auch keine Hilfsdateien zugewiesen sind, wird auch keine Multitask-Sortierung durchgeführt.

Hinweis

Wie beim Zyklensortieren kann auch beim Multitasksortieren die Leistung verbessert werden, wenn Arbeitsdateien und Hilfsdateien auf getrennten Datenträgern mit eigener Zugriffseinrichtung liegen. Dies gilt auch für die Eingabedateien und die Ausgabedateien, die aber untereinander nicht getrennte Zugriffseinrichtungen benötigen.

8.4 Geeignete Wahl der Dateieigenschaften

Eingabedateien (SORTINxx)

Eingabedateien sollten nach Möglichkeit Blockgrößen von mindestens 4096 Byte haben (bei einer Verkettung von 2 PAM-Seiten tritt bereits eine erhebliche CPU- und Laufzeiterparnis ein).

Eingabedateien auf privaten Datenträgern führen in der Regel zu schnellerer Bearbeitung (Laufzeiterparnis).

Ausgabedatei (SORTOUT)

Ausgabedateien sollen Blockgrößen von mindestens 4096 Byte haben. Bei einer Verkettung von 2 PAM-Seiten tritt bereits eine erhebliche Zeitersparnis ein.

Außerdem sollte darauf geachtet werden, dass eine ausreichende Primär- und Sekundärzuweisung definiert wurde. Auf diese Weise kann der Verwaltungsaufwand für die Dateierweiterungen eingeschränkt werden. Die sonstigen Dateimerkmale sollten besser, soweit dies möglich ist, durch SORT von der Eingabedatei übernommen werden, da auf diese Weise Fehler vermieden werden. Ausgabedateien auf privaten Datenträgern können in der Regel zu Laufzeitverbesserungen führen.

Arbeitsdateien (SORTWK oder SORTWKx)

Erhebliche Laufzeiterparnis sind möglich, wenn eine Arbeitsdatei auf privater Platte eingerichtet wird und diese Platte möglichst nicht parallel von anderen Taskabläufen angesprochen wird. Beim Multitasksortieren gilt das für jede einzelne Arbeitsdatei.

Die Dateimerkmale einer Arbeitsdatei werden von SORT voll definiert und eine zu niedrige SPACE-Angabe ggf. korrigiert.

Hilfsdateien (SORTWKxx)

Für Hilfsdateien gilt unter Einbeziehung von Bändern entsprechend, was für die Arbeitsdateien gesagt wurde. Die Einschränkungen für das Sortierverfahren Multitasksortieren müssen hierbei beachtet werden. Bänder sollten grundsätzlich mit hoher Schreiddichte bearbeitet werden.

8.5 Satzsummierung

Für Satzsummierungen sollte grundsätzlich die SUM-RECORDS-Anweisung verwendet werden, die bewirkt, dass Summierungen schon ab der Vorsortierung erfolgen und damit Ein- und Ausgaben von erheblichem Umfang eingespart werden. Beim Satzsummieren werden bei jedem Aufeinandertreffen von 2 Sätzen mit gleichen Sortierbegriffen die in der SUM-RECORDS-Anweisung beschriebenen Summenfelder im 1. Satz addiert und der 2. Satz jeweils gelöscht. Bei Test-Messungen konnten bis zu 60 % (!) Lauf- und CPU-Zeiterparnis registriert werden (abhängig vom Verdichtungsgrad).

8.6 SORT als Subsystem

Teile des SORT können von der Systembetreuung als Subsystem geladen werden. Das erspart einen Teil der Ladevorgänge und macht SORT damit schneller.

Das gilt besonders für das Multitasksortieren, bei dem ansonsten für jede Nebentask ein Teil der Module erneut geladen werden muss.

8.7 Angabe des OPTIMIZATION-Operanden in der SET-SORT-OPTIONS-Anweisung

Mit dem OPTIMIZATION-Operanden in der SET-SORT-OPTIONS-Anweisung kann der Sortierlauf hinsichtlich Laufzeit, CPU-Zeit oder Speicherbedarf optimiert werden. Bei OPTIMIZATION=*VIRTUAL-MEMORY werden nicht mehr benötigte Lademodule (ca. 85% des statischen Programmcodes) freigegeben. Die Laufzeiten bzw. CPU-Zeiten erhöhen sich beim Freigeben der Lademodule sowie beim Wiederladen für einen evtl. nachfolgenden Sortierlauf. Andererseits kann bei Adressraummangel mit OPTIMIZATION=*VIRTUAL-MEMORY erreicht werden, dass der durch das Freigeben der Lademodule frei werdende Speicher für die Sortierarbeitsspeicher benutzt wird und SORT somit effizienter mit dem verfügbaren Speicher umgeht.

8.8 Ändern der Voreinstellungen für SORT

Zum Ändern und Anzeigen von SORT-Parametern stehen die Anweisungen MODIFY-SORT-DEFAULTS (siehe [Seite 154](#)) und SHOW-SORT-DEFAULTS (siehe [Seite 178](#)) zur Verfügung.

9 Installation

SORT V8.0 ist unter dem Betriebssystem BS2000/OSD-BC ab V8.0 ablauffähig.

Aus Gründen der Kompatibilität mit älteren Anwendungen ist weiterhin eine Bibliothek SORTLIB erforderlich.

Die Bibliothek SORTLIB enthält den Startermodul SRT80 (Objektmodul). Ausserdem ist ein Objektmodul SRTXKERN enthalten. Dieser dient dazu, Aufrufe zu befriedigen, die in alten Anwendungen fest einprogrammiert auf diesen Modul in der Bibliothek SORTLIB verweisen. Wenn IMON nicht im Einsatz ist, wird bei Unterprogrammaufrufen noch auf die Bibliothek SORTLIB zugegriffen. Dies trifft ebenfalls auf alte Startermodule aus SORT Versionen V7.4 und kleiner zu, da diese prinzipiell noch nicht mit IMON arbeiten.

Die Installation ist nicht an fest vorgegebene Dateinamen und Kennungen gebunden. Das bedeutet, dass die Produkte den Standort aller Bestandteile einer Version selbst dynamisch ermitteln müssen. Als Werkzeug steht dafür das Subsystem IMON zur Verfügung (siehe „IMON“[\[11\]](#)).

Produktdateien

logischer Name	Bedeutung	Standard-Dateiname
SYSPRG	Phase (Kompatibilität)	SORT
SYSOML	Modulbibliothek	SORTLIB
SYSLNK	Modulbibliothek	SYSLNK.SORT.080
SYSLNK.TAB	Tabellenbibliothek	SYSLNK.SORT.080.TAB
SYSPAR	Parameterdatei	SYSPAR.SORT.080
SYSLIB	Makrobibliothek	SYSLIB.SORT.080
SYSSDF	SDF-Syntaxdatei	SYSSDF.SORT.080
SYSMES	Meldungsdatei (MSGMAKER-Format)	SYSMES.SORT.080
SYSSPR	Prozedurbibliothek	SYSSPR.SORT.080
SYSREP	Rep-Korrekturdatei	SYSREP.SORT.080
SYSSSC	Subsystemdeklarationen oberer Adressraum	SYSSSC.SORT.080
SYSSSC.LOW	Subsystemdeklarationen unterer Adressraum	SYSSSC.SORT.080.LOW
SYSSII	Installationsinformationen für IMON	SYSSII.SORT.080
SYSFGM.D	Freigabemitteilung deutsch	SYSFGM.SORT.080.D
SYSFGM.E	Freigabemitteilung englisch	SYSFGM.SORT.080.E
SYSACF	ACS-Modelldatei	SYSACF.SORT.080
SYSRME.D	Readme-Datei, Deutsch	SYSRME.SORT.080.D
SYSRME.E	Readme-Datei, Englisch	SYSRME.SORT.080.E
SYSNRF		SYSNRF.SORT.080

Anmerkungen

1. Die Dateien SYSSPR.SORT.080 und SYSACF.SORT.080 haben feste Namen und müssen mit diesen Namen in der Standard-Benutzerkennung (DEFLUID) bereitgestellt werden.
2. Die Datei SYSLIB.SORT.080 sollte aus Gründen der Kompatibilität noch als \$.SORT-MACLIB abgespeichert werden oder mit Hilfe von ACS mit diesem Namen versehen werden.
3. Die Bibliothek \$.SORTLIB wird immer ausgeliefert. Sie enthält die Module SRT80 und SRTXKERN. Aus SRTXKERN wird immer in die richtige Bibliothek (SYSLNK.SORT.080) verzweigt.

Frei wählbare Dateinamen

Die Installation von SORT ist nicht an feste Dateinamen und Kennungen gebunden.

Die Namen der einzelnen Dateien des SORT und die Kennungen, in denen sie abgelegt werden, können bei der Installation mit IMON-GPN festgelegt werden. Das erlaubt beispielsweise die Ablage der Sortierbibliothek in einer Kennung SORT.

Voraussetzung dafür ist, dass die Subsysteme DSSM und IMON-GPN installiert sind.

Für alle Produktdateien werden ihre zugehörigen Dateinamen (die vollständigen Pfadnamen) im so genannten IMON-SCI (Software-Configuration-Inventory), der zentralen Datenbasis von IMON, abgelegt und können bei späteren SORT-Anwendungen dort abgerufen werden. Der Nutzer muss diese Dateinamen also nicht kennen. Sie werden ihm vom Betriebssystem ermittelt.

Das IMON-SCI enthält für jede Produktdatei folgende drei Informationen:

1. Logischer Name der Datei (für die Sortierbibliothek ist das z.B. SYSLNK)
2. Name des Produkts (SORT)
3. Produkt-Version (z.B. 08.0A00)

Koexistenz

Da die Namen und Ablagekennungen der Produktdateien nicht fest vorgegeben sind, ist es möglich auf einer Anlage mehrere Versionen eines Produkts bereitzustellen. Dazu müssen die unterschiedlichen Versionen des Produkts nur unter verschiedenen Kennungen oder mit unterschiedlichen Dateinamen installiert werden.

Diese Koexistenz (coexistence) verschiedener Produktversionen auf einer Anlage wird von SORT ab V7.5A unterstützt, d.h. SORT V8.0A kann mit SORT-Versionen ab V7.5A koexistieren. SORT-Versionen einschließlich V7.4A sind nicht koexistenzfähig, da sie noch feste Dateinamen enthalten und auch organisatorische Hilfsmittel (IMON) fehlen.

Koexistenz ist nur für Hauptversionen erlaubt, verschiedene Freigabe- oder Korrekturstände derselben Hauptversion sind untereinander nicht koexistenzfähig.

Beispiel

Die Versionen V7.7A und V8.0A sind untereinander koexistenzfähig. Eine Version V8.0B oder V8.0A10 wäre mit V8.0A00 nicht koexistenzfähig.

In den Kommandos START-SORT und SORT-FILE sowie in den Makros zum Unterprogrammaufruf kann der Benutzer mit dem Operanden VERSION die gewünschte SORT-Version auswählen.

Eine weitere Möglichkeit, die gewünschte SORT-Version anzufordern, liefert das Systemkommando `SELECT-PRODUCT-VERSION`. In diesem Kommando kann der Benutzer vor dem SORT-Aufruf (bei einer SORT-Unterprogrammvariante vor dem Aufruf des zugehörigen Hauptprogramms) die gewünschte SORT-Version festlegen. Diese SORT-Version wird dann ausgewählt, wenn beim Start von SORT keine explizite Versionsangabe erfolgt. Der Operand `SCOPE` des Kommandos `SELECT-PRODUCT-VERSION` steuert die Gültigkeitsdauer dieser Festlegung. Mit `SCOPE=*PROGRAM` wird die SORT-Version für den folgenden Programmaufruf festgelegt, d.h. das Kommando `SELECT-PRODUCT-VERSION` muss vor jedem Hauptprogrammaufruf wiederholt werden. Mit `SCOPE=*TASK` wird die SORT-Version bis zum Ende der Task festgelegt.

Bei vollständiger Versionsangabe im Startkommando oder beim Unterprogrammaufruf wird mit dieser Version gearbeitet (vorausgesetzt, diese Version ist ordnungsgemäß installiert). Bei fehlender (d.h. `*STD`) bzw. zu wenig detaillierter Versionsangabe kommen u.U. mehrere Versionen in Frage. In diesem Fall werden vom Kommandoserver bzw. von SORT beim Unterprogrammaufruf alle passenden Versionen ermittelt. Die Version wird dann unter Beachtung folgender Prioritäten bestimmt:

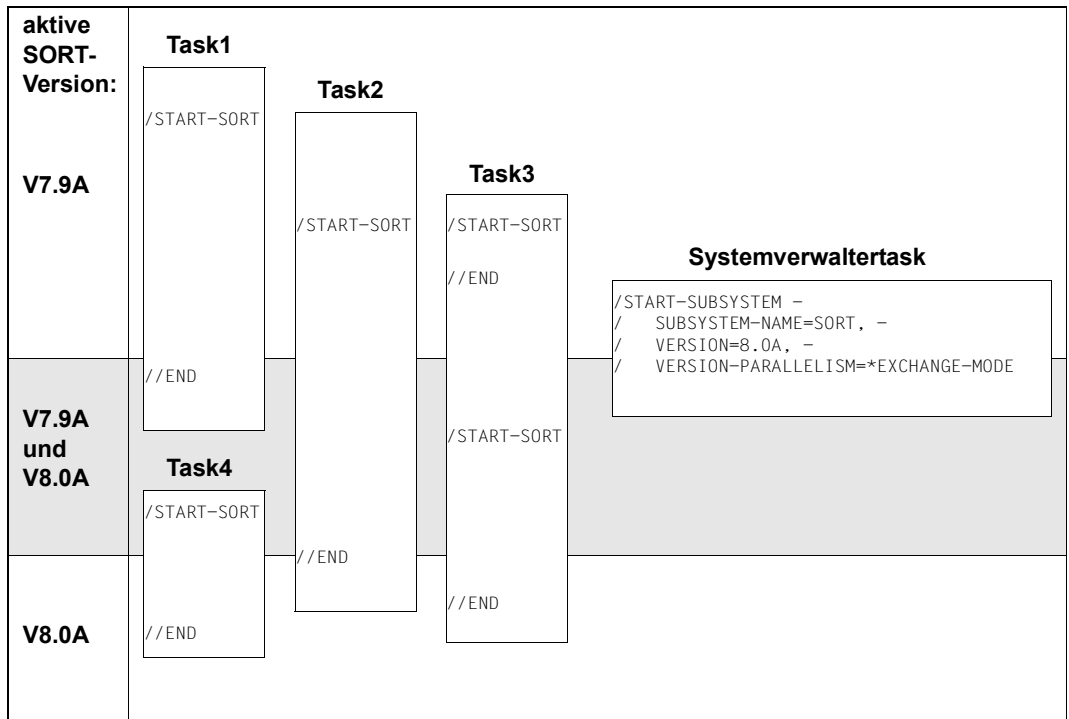
1. Die mit dem Kommando `/SELECT-PRODUCT-VERSION` vorgegebene Version.
2. Die höchste mit IMON installierte SORT-Version.

Ist in einem Betriebssystem das Subsystem IMON-GPN nicht installiert, so wird der Großmodul von SORT aus der Bibliothek `$.SYSLNK.SORT.080` bzw. `$.SORTLIB` nachgeladen. Das gilt auch, wenn das Subsystem IMON-GPN aktiv, das Produkt SORT aber in seiner IMON-SCI nicht enthalten ist.

Wurde eine explizit angeforderte SORT-Version nicht gefunden, erfolgt ein Abbruch.

Austauschbarkeit

Wenn mehrere Versionen von SORT gleichzeitig installiert sind, ist es möglich, diese im laufenden Betrieb auszutauschen. Folgende Grafik stellt an einem Beispiel die zeitliche Abfolge dar, wie eine Version (V7.9A) des Subsystems SORT gegen eine andere (V8.0A) ausgetauscht werden kann:



Zu Beginn des Beispiels ist SORT V7.9A aktiv. Drei Tasks (Task1, Task2 und Task3) starten je einen SORT-Lauf.

Mit dem Kommando START-SUBSYSTEM aktiviert die Systembetreuung die Version V8.0A von SORT, wobei er mit der Angabe VERSION-PARALLELISM=*EXCHANGE-MODE eine temporäre Koexistenz mehrerer Versionen erlaubt. Der SORT-Lauf in Task3 ist zu diesem Zeitpunkt bereits beendet, aber die beiden Tasks Task1 und Task2 arbeiten noch mit SORT. Deshalb bleibt die Version V7.9A von SORT weiterhin aktiv. Alle Tasks, die ab jetzt einen SORT-Lauf starten (Task3 und Task4), arbeiten allerdings mit der neuen Version V8.0A.

Die Beendigung des SORT-Laufs in Task1 hat keinen Einfluss auf die Beendigung der Version V7.9A von SORT. Diese Version des Subsystems SORT wird erst dann deaktiviert, wenn die letzte Task (Task2), die noch mit ihr arbeitet, den SORT-Lauf beendet.

10 Beispiele

Dieses Kapitel enthält Anwendungsbeispiele für die meisten Funktionen des SORT. Die Beispiele wurden ausprobiert und die Ablaufprotokolle in diesem Kapitel abgedruckt. Ausgenommen sind die Assemblerprogramme. Hier ersetzt der Quelltext aus Platzgründen die Assemblerliste.

10.1 Einführung

Diese Einführung hat folgende Struktur:

- Kurze Rekapitulation der Syntax
- Beispiele für die Syntax der SORT-Steueranweisungen

Kurze Rekapitulation der Syntax von SORT-Steueranweisungen

Für die Durchführung eines Sortierlaufs sind in der Regel drei Schritte notwendig:

- Aufruf von SORT
- Zuweisen der von SORT benötigten Ein- und Ausgabedateien
- Definition der Sortierkriterien.

10.1.1 Aufruf von SORT

SORT kann mit einem der folgenden Kommandos aufgerufen werden (siehe [Kapitel „Aufruf von SORT“ auf Seite 195](#)):

START-SORT	Der Aufruf mit diesem Kommando ist notwendig, wenn die gesamte Funktionalität von SORT zur Verfügung stehen soll. Deshalb wird in den weiteren Beispielen ausschließlich diese Art des Aufrufs verwendet.
SORT-FILE	Mit diesem Kommando können einfache Sortierungen durchgeführt werden. Dabei sind keine weiteren Eingaben mehr notwendig, da die Dateizuweisungen und die Definition von Sortierkriterien direkt mit diesem Kommando vorgenommen werden.

Aufruf von SORT mit START-SORT

```
/start-sort
% SRT1001 2014-10-12/12:19:38/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
%/
```

SORT meldet sich mit dem doppelten Promptstrich ('// ') empfangsbereit für SORT-Steueranweisungen.

10.1.2 Zuweisen der Dateien

SORT benötigt für jeden SORT-Lauf (mindestens) eine Eingabedatei und eine Ausgabedatei.

Ein- und Ausgabedatei müssen SORT für jeden SORT-Lauf explizit angegeben werden.

Ein- und Ausgabedateien können auf verschiedene Art und Weise zugewiesen werden:

- Als Anweisung an SORT:

```
//ASSIGN-FILES INPUT-FILES=eingabedatei, OUTPUT-FILE=ausgabedatei
```

- Vor dem Aufruf von SORT:

```
/ADD-FILE-LINK LINK-NAME=SORTIN, FILE-NAME=eingabedatei...
/CREATE-FILE FILE-NAME=ausgabedatei
/ADD-FILE-LINK LINK-NAME=SORTOUT, FILE-NAME=ausgabedatei...
```

- Direkte Angabe der Dateien im SORT-FILE-Kommando:

```
/SORT-FILE ... INPUT-FILES=eingabedatei,OUTPUT-FILE=ausgabedatei...
```

Hinweis

Im Allgemeinen erzeugt SORT die Ausgabedateien mit den Dateimerkmale der Eingabedateien. Ausnahmen hierzu sind Sortierungen, bei denen

- SORT andere Standardvorgaben hat (vgl. [Abschnitt „Auswahlsortieren“ auf Seite 27](#)).
- die Ausgabedatei absichtlich ein anderes Format als die Eingabedatei haben soll.

SORT greift bei einem SORT-Lauf auf diese Dateien zu oder legt sie an:

- Eingabedateien
 - Ausgabedatei
- } müssen bei jedem Sortiervorgang vom Benutzer zugewiesen werden
- Arbeitsdateien
 - Hilfsdateien
 - Fixpunktdatei
- } nur bei großen Eingabedateien von Bedeutung

Kleine Dateien können sofort im Hauptspeicher (CORE) sortiert werden.

10.1.3 Definition der Sortierkriterien

Sortierkriterien im weiteren Sinne sind alle Anweisungen, die SORT gegeben werden. Unter Sortierkriterien im engeren Sinn sind jene Angaben zu verstehen, die sich auf den Sortierbegriff und die Sortierart beziehen. *Sortierkriterien* wird im Folgenden immer im engeren Sinne verwendet.

Syntaxbeispiel für eine einfache Sortierung

Die einfachste Sortierung kann ausgelöst werden durch

```
//sort-records
```

Hier verwendet SORT die durch SDF vorgegebenen Standardeinstellungen. Diese Standardeinstellungen sind:

FIELDS = COMPLETE-RECORD	Der gesamte Satz wird bewertet
FORMAT = CHARACTER	Die Daten werden als alphanumerische Zeichen interpretiert
SORTING-ORDER = ASCENDING	Die niedrigstwertigen Daten werden an den Anfang der Datei sortiert, die höchstwertigen an das Ende.

Beispiel

Würde mit der Anweisung SORT-RECORDS ohne Operanden eine Datei sortiert, die nur aus Großbuchstaben besteht, stünde in der sortierten Datei zuerst das 'A' und zuletzt das 'Z'.)

Mit SORT können Datensätze nicht nur als Ganzes sortiert werden. Die Anweisung SORT-RECORDS bietet folgende weiteren Möglichkeiten:

- Es können Abschnitte eines Satzes bestimmt werden, die für die Sortierung zu berücksichtigen sind. Andere Teile des Satzes haben dann keinen Einfluss auf die Sortierung. Beispielsweise soll eine Namensliste nach Nachnamen und innerhalb desselben Nachnamens nach Vornamen sortiert werden, wobei diese Felder nicht notwendigerweise am Satzanfang stehen müssen.
- Die Felder eines Eingabesatzes können in einer neuen, vom Benutzer definierten Reihenfolge zu einem Ausgabesatz zusammengestellt werden.

Zur Definition von Feldern, die für die Sortierung berücksichtigt oder in den Ausgabesatz übernommen werden sollen, steht in der Anweisung SORT-RECORDS der Operand FIELDS zur Verfügung.

Felddefinitionen

Der Sortierbegriff besteht aus einem Sortierfeld oder mehreren Sortierfeldern. Sortierfelder beschreiben alle Bereiche des Eingabesatzes, die SORT für die Bestimmung der Reihenfolge der Sätze in der Ausgabedatei auswertet. Die Definition der Felder erfolgt im FIELDS-Operanden.

SORT bewertet die Sortierfelder standardmäßig in der im Befehl angegebenen Reihenfolge. Das wichtigste (oder übergeordnete) Sortierkriterium sollte an erster Stelle genannt werden. (Abweichungen hiervon: vgl. PRIORITY bei SORT-RECORDS).

```
//sort-records fields=( -  
//      *field-explicit(position=...,length=...,format=...,sorting-order=...), -  
//      *field-explicit(...),...)
```

Hierbei haben die Angaben nach FIELD-EXPLICIT folgende Funktion:

POSITION	Anfang des Feldes in Byte
LENGTH	Länge des Feldes in Byte
FORMAT	Datenformat
SORTING-ORDER	Sortierrichtung

Beschreibungen der maximalen Größen und möglichen Einträge für die Platzhalter finden sich in den vorderen Kapiteln des Handbuchs.

Syntax-Beispiel: ein Sortierbegriff – zwei Sortierfelder

```
//sort-records fields=( -  
//      *field-explicit(position=1,length=11), -  
//      *field-explicit(position=23,length=10,sorting-order=*descending))
```

Die Anweisung definiert als Sortierbegriff zwei Sortierfelder.

1. Sortierfeld: *FIELD-EXPLICIT(POSITION=1,LENGTH=11)

SORT beginnt bei Byte 1 (POSITION) und bewertet die folgenden 11 Byte (LENGTH) des Satzes. Der Satzbereich soll aufsteigend sortiert werden. Die Daten selbst sind vom Typ CHARACTER (Zeichen). Diese letzten beiden Angaben werden nicht ausdrücklich benötigt, da sie bereits die Standardvorgabe sind.

2. Sortierfeld: *FIELD-EXPLICIT(POSITION=23,LENGTH=10,
 SORTING-ORDER=*DESCENDING)

Das zweite Sortierfeld beginnt bei Position 23 und erstreckt sich über die folgenden 10 Bytes. Dieser Satzbereich soll jedoch absteigend sortiert werden (SORTING-ORDER=*DESCENDING).

10.1.4 Eingabe der Anweisungen beenden und Sortierlauf starten

Jede Definition eines Sortierlaufs muss mit der END-Anweisung abgeschlossen werden. Erst danach beginnt SORT mit der Ausführung der eingegebenen Anweisungen.

10.2 Beispielsortierung einer Datei mit fester Satzlänge

SORT soll die Datei **ADRESSEN** alphabetisch nach dem Straßennamen sortieren. Die Datei **ADRESSEN** enthält Sätze mit folgendem Aufbau und Inhalt:

Name	Vorname	Straße Nach diesem Feld soll sortiert werden (Sortierfeld)	PLZ	Tel.
1	11	23	49	56 62
Miller	Andrew	Poplar Avenue 47	KT25	544507
Allan	Hilary	High Street 101	AY4	345679
Smith	Albert	Gardener Street 14	PX453	047913
Majors	Christine	Railway Cuttings 12	PX23	987650
Smythe	Brenda	Thomas Square 1	BT34	965471
Kennedy	George	Edgware Road 62	NY211	873250
Stevens	Henry	Market Square 13	NY12	987234
Baker	Fred	Scott Street 34	KT23	765921
Johnston	Annette	Richmond Street 98	BT342	345678
Mellors	Ingrid	Salford Drive 4	TI34	456372
Brown	Tony	Skyview Terrace 9	UB81	786534
Charles	Ernest	Millhouse Street 23	TI32	537892
Walters	Claudia	Millford Crescent 31	ZY21	342108
Richards	Bernard	Illsley Square 3	UB12	518376
Drever	James	Rose Drive 31	PX3	875211

Anzeige der Eigenschaften der Datei **ADRESSEN**

```

/show-file-attributes file-name=adressen, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.ADRESSEN
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN      = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (F,N)       REC-SIZE   = 62
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM  = *K          SO-MIGR    = *ALLOWED

```

Über das **SHOW-FILE-ATTRIBUTES**-Kommando werden die Dateieigenschaften der Eingabedatei **ADRESSEN** abgefragt. Das System weist u.a. diese Eigenschaften für die Datei **ADRESSEN** aus:

- SAM-Datei (FILE-STRUC=SAM)
- feste Satzlänge (REC-FORM=(F,N))
- Satzlänge (REC-SIZE=62).

SORT legt die Ausgabedatei mit denselben Merkmalen an.

SORT aufrufen

```
/start-sort
% SRT1001 2014-10-12/16:05:25/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
```

SORT meldet sich betriebsbereit und erwartet die Eingabe der SORT-RECORDS-Anweisung.

Ein- und Ausgabegabedatei zuweisen

```
//assign-files input-files=adressen,output-file=adressen.sort
```

Mit der Anweisung ASSIGN-FILES wird die Datei ADRESSEN als Eingabedatei und die Datei ADRESSEN.SORT als Ausgabedatei zugewiesen.

Die Ausgabedatei ADRESSEN.SORT wird von SORT mit den Dateimerkmalen der Eingabedatei ADRESSEN eingerichtet.

Sortierfeld definieren

```
//sort-records fields=(*field-explicit(position=23,length=26))
```

Die Datei ADRESSEN soll nach Straßennamen sortiert werden. Das Feld „Straße“ beginnt bei Byte 23 und es sollen die folgenden 26 Byte bewertet werden. Im Operanden *FIELD-EXPLICIT der SORT-RECORDS-Anweisung wird daher als Sortierbegriff das Sortierfeld (POSITION=23,LENGTH=26) angegeben.

Anweisungsfolge abschließen und Sortierlauf starten

```
//end
```

Mit der END-Anweisung wird die Eingabe der Anweisungen an SORT abgeschlossen und der Sortierlauf gestartet.

Meldungen von SORT

```
% SRT1016 SORT/MERGE INPUT RECORDS:.....15 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....15
% SRT1002 2014-10-12/16:12:59/000000.13 SORT/MERGE COMPLETED
```

SORT meldet, dass 15 Sätze eingelesen und 15 Sätze ausgegeben wurden. Das Resultat der Sortierung ist zu sehen, wenn man sich die Ausgabedatei ADRESSEN.SORT anzeigen lässt.

Ergebnis der Sortierung in der Ausgabedatei ADRESSEN.SORT

Name	Vorname	Straße (Sortierfeld)	PLZ	Tel.
1	11	23	49	56 62
Kennedy	George	Edgware Road 62	NY211	873250
Smith	Albert	Gardener Street 14	PX453	047913
Allan	Hilary	High Street 101	AY4	345679
Richards	Bernard	Illsley Square 3	UB12	518376
Stevens	Henry	Market Square 13	NY12	987234
Walters	Claudia	Millford Crescent 31	ZY21	342108
Charles	Ernest	Millhouse Street 23	TI32	537892
Miller	Andrew	Poplar Avenue 47	KT25	544507
Majors	Christine	Railway Cuttings 12	PX23	987650
Johnston	Annette	Richmond Street 98	BT342	345678
Drever	James	Rose Drive 31	PX3	875211
Mellors	Ingrid	Salford Drive 4	TI34	456372
Baker	Fred	Scott Street 34	KT23	765921
Brown	Tony	Skyview Terrace 9	UB81	786534
Smythe	Brenda	Thomas Square 1	BT34	965471

Dateimerkmale der Ausgabedatei ADRESSEN.SORT

```

/show-file-attributes file-name=adressen.sort, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.ADRESSEN.SORT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD            DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)       REC-SIZE   = 62
% AVAIL       = *STD
% WORK-FILE   = *NO         F-PREFORM  = *K           SO-MIGR    = *ALLOWED

```

Die Ausgabedatei weist dieselben Dateimerkmale wie die Eingabedatei auf.

10.3 Beispielsortierung einer SAM-Datei mit variablem Satzformat

Die Datei LITERATURE ist eine SAM-Datei mit variabler Satzlänge (REC-FORM=V).

Dateimerkmale der Datei LITERATURE.SAM

```
/show-file-attributes file-name=literature, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.LITERATURE
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)       REC-SIZE   = 0
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM  = *K          SO-MIGR   = *ALLOWED
```

Sie enthält Sätze mit folgendem Aufbau und Inhalt und soll nach Titeln sortiert werden.

SL	Name	Vorname	Titel (Sortierfeld)	Gattung
1	5	19	30	63
	Pasternak	Boris	Doctor Zhivago	Novel
	Capote	Truman	In Cold Blood	Novel
	Boyle	Jimmy	A Sense of Freedom	Autobiography
	Arden	John	Sergeant Musgrave's Dance	Theatre
	Milligan	Spike	Puckoon	Novel
	Dahl	Roald	Kiss Kiss	Short Stories
	Shakespeare	William	Romeo and Juliet	Theatre
	Fielding	Henry	Tom Jones	Novel
	Jonson	Ben	Volpone	Theatre
	Dumas	Alexandre	The Three Musketeers	Novel
	Troyat	Henri	Pushkin	Biography
	Shaw	Bernard	Pygmalion	Theatre
	Sharpe	Tom	Riotous Assembly	Novel
	Thomas	Dylan	Fern Hill	Poem
	Gogol	Nikolai	Dead Souls	Novel

Berechnen der Position des ersten Datenbytes bei Sätzen variabler Länge

Jedem Satz variabler Länge ist ein 4 Byte langes Satzlängenfeld vorangestellt. Im Beispiel ist dieses mit SL gekennzeichnet. Obwohl der Benutzer keinen Einfluss auf dieses Feld hat, gehört es mit zum Datensatz. Deshalb muss zu jeder Position eines Feldes die Zahl der Bytes im Satzlängenfeld hinzuaddiert werden.

Ablaufprotokoll der Sortierung nach TITEL

```

/start-sort
% SRT1001 2014-10-12/15:56:27/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=literature,output-file=literature.sort
//sort-records fields=(*field-explicit(position=30,length=33))
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....15 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....15
% SRT1002 2014-10-12/15:56:31/000000.26 SORT/MERGE COMPLETED

```

Ergebnis der Sortierung in der Ausgabedatei LITERATURE.SORT

SL	Name	Vorname	Titel	Gattung
1	5	19	30	63
	Boyle	Jimmy	A Sense of Freedom	Autobiography
	Gogol	Nikolai	Dead Souls	Novel
	Pasternak	Boris	Doctor Zhivago	Novel
	Thomas	Dylan	Fern Hill	Poem
	Capote	Truman	In Cold Blood	Novel
	Dahl	Roald	Kiss Kiss	Short Stories
	Milligan	Spike	Puckoon	Novel
	Troyat	Henri	Pushkin	Biography
	Shaw	Bernard	Pygmalion	Theatre
	Sharpe	Tom	Riotous Assembly	Novel
	Shakespeare	William	Romeo and Juliet	Theatre
	Arden	John	Sergeant Musgrave's Dance	Theatre
	Dumas	Alexandre	The Three Musketeers	Novel
	Fielding	Henry	Tom Jones	Novel
	Jonson	Ben	Volpone	Theatre

Die Datei LITERATURE.SORT ist nun aufsteigend alphabetisch nach Titeln geordnet.

Dateimerkmale der Ausgabedatei LITERATURE.SORT

```
/show-file-attributes file-name=literature.sort, -  
/  
                          information=*parameters(organization=*yes)  
%0000000003 :CTID:$EXAMPLE.LITERATURE.SORT  
% ----- ORGANIZATION -----  
% FILE-STRUC = SAM                  BUF-LEN      = STD(1)          BLK-CONTR = PAMKEY  
% IO(USAGE)  = READ-WRITE  IO(PERF)      = STD              DISK-WRITE = IMMEDIATE  
% REC-FORM   = (V,N)       REC-SIZE      = 2044  
% AVAIL      = *STD  
% WORK-FILE  = *NO          F-PREFORM   = *K              SO-MIGR   = *ALLOWED
```

10.4 Übersicht über die Anwendungsfälle

10.4.1 SORT als Hauptprogramm

Nr.	Sortierart	Eingabedatei: Dateityp und Satzformat	Ausgabedatei: Dateityp und Satzformat	weitere Kriterien
01	Vollsortieren	SAM, fest	SAM, fest	SORT-RECORDS
02	Vollsortieren	SAM, variabel	SAM, variabel	Position des Sortierfeldes mit Satzlängenfeld
03	Vollsortieren	ISAM, variabel	SAM, variabel	Änderung der Dateieigenschaften der Ausgabedatei
04	Vollsortieren	ISAM, variabel	ISAM, variabel	Ausgabedatei mit anderer Schlüsselposition
05	Vollsortieren	ISAM, variabel	SAM, variabel	mehrere Eingabedateien
06	Vollsortieren	SAM, variabel	SAM, variabel	Eingabe- gleich Ausgabedatei, symbolische Bezeichner
07	Vollsortieren	SAM, variabel	SAM, variabel	Unterschied DIN-Sortierung/EBCDIC-Sortierung
08	Vollsortieren	SAM, variabel	SAM, variabel	MODIFY-CODE: eigene Abfolge von Zeichen
09	Vollsortieren	SAM, variabel	SAM, variabel	sortieren mit erweiterten Zeichensätzen
10	Vollsortieren	SAM, variabel	SAM, variabel	Summierung und Ausschluss einiger Sätze
11	Auswahlsortieren	SAM, variabel	SAM, variabel	2 Sortierfelder, 1 Restfeld
12	Auswahlsortieren	SAM, fest	SAM, fest	Festlegen Sortierfeld im binären Format
13	Auswahlsortieren	POSIX	POSIX	1 Sortierfeld, 2 Restfelder
14	Adresslisten- sortieren	SAM, fest	SAM, fest	SORT-TYPE=*TAG-TRAILER
15	Mischen	SAM, variabel	SAM, variabel	3 sortierte Eingabedateien MERGE-RECORDS

10.4.2 Anschluss von Benutzerroutinen

Nr.	Sortierart	Eingabedatei: Dateityp und Satzformat	Ausgabedatei: Dateityp und Satzformat	weitere Kriterien
16	Vollsortieren	SAM, fest	SAM, fest	Benutzerausgang INPUT
17	Vollsortieren	SAM, variabel	SAM, variabel	Benutzerausgang OUTPUT
18	Vollsortieren	SAM, fest	SAM, fest	Benutzerausgang PHYSICAL-TRANSLATE
19	Vollsortieren	SAM, variabel	SAM, variabel	Benutzerausgang VIRTUAL-TRANSLATE

10.4.3 SORT als Unterprogramm

Nr.	Sortierart	Eingabedatei: Dateityp und Satzformat	Ausgabedatei: Dateityp und Satzformat	weitere Kriterien
20	Vollsortieren	SAM, fest	SAM, fest	Übergabe der Steueranweisungen über Ebene 0
21	Vollsortieren	SAM, fest	SAM, fest	Übergabe der Steueranweisungen über Ebene 1
22	Vollsortieren	SAM, variabel	SAM, variabel	SORT-Zugriffsmethode SRTZM
23	Mehrfachsortieren (Voll-/Auswahl- sortieren)	SAM, fest	SAM, fest	SORT-Zugriffsmethode SRTZM

10.4.4 Sortieren nach Unicode

Nr.	Sortierart	Eingabedatei: Dateityp und Satzformat	Ausgabedatei: Dateityp und Satzformat	weitere Kriterien
24	Vollsortieren	SAM, variabel	SAM, variabel	Format Unicode-Character

10.5 Beispiele

In den Beispielen werden verschiedene Dateien zur Demonstration verwendet. Der Inhalt der unsortierten Dateien ist dann gleich, wenn der erste Teil des Dateinamens gleich ist. Die Dateieigenschaften ändern sich jedoch bei verschiedenen Beispielen, da schließlich die Art der Datei die Position der Daten bestimmt. Die Art der Datei ist deshalb im Dateinamen enthalten.

So werden z.B. von der Datei mit dem ersten Namen RESTAURANT diese Dateien verwendet:

Eigenschaften	Dateiname
SAM-Datei mit fester Satzlänge	RESTAURANT.SAM.FIX
SAM-Datei mit variabler Satzlänge	RESTAURANT.SAM
ISAM-Datei mit variabler Satzlänge	RESTAURANT.ISAM

Variable Dateien weisen also im Dateinamen nur den Zugriffstyp auf, Dateien mit fester Satzlänge wurden mit dem Zusatz FIX versehen.

Die Ausgabedateien sind entsprechend gekennzeichnet, führen am Ende jedoch immer den Zusatz SORT.

RESTAURANT.ISAM.SORT wäre die sortierte Datei zu RESTAURANT.ISAM.

Da sich der Inhalt der Ausgangsdatei nicht ändert, findet sich ein Ausdruck am Ende dieses Kapitels.

10.5.1 Beispiel 1: Vollsortieren von Sätzen mit festem Satzformat

Eingabe: SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat

Ausgabe: SAM-Datei RESTAURANT.SAM.FIX.SORT mit festem Satzformat

Übung zu:

- Position des Sortierfeldes ermitteln
- SORT aufrufen
- Sortierbefehl geben: SORT-RECORDS FIELDS=*FIELD-EXPLICIT

Struktur der Eingabedatei RESTAURANT.SAM.FIX

Restaurantname (Sortierfeld)	Straße	Telefon	Art der Küche
1	21	48	56 66

Der vollständige Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“](#) auf [Seite 389](#) dargestellt.

Dateimerkmale der Datei RESTAURANT.SAM.FIX

```
/show-file-attributes file-name=restaurant.sam.fix, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.SAM.FIX
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% REC-FORM   = (F,N)       REC-SIZE   = 66
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM  = *K          SO-MIGR    = *ALLOWED
```

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/14:16:58/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.sam.fix, -
//              output-file=restaurant.sam.fix.sort
//sort-records fields=*field-explicit(position=1,length=20)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/14:16:58/000000.27 SORT/MERGE COMPLETED

```

Struktur und Inhalt der Ausgabedatei RESTAURANT.SAM.FIX.SORT (Auszug)

Restaurantname	Straße	Telefon	Art der Küche
1	21	48	56 66
Chayota's	Thurston Street 60	292742	Japanese
Golden Fleece	Arran Street 44	242437	Yugoslavian
Java	Hope Street 51	522221	Indonesian

10.5.2 Beispiel 2: Vollsortieren von Sätzen mit variablem Satzformat

Eingabe: SAM-Datei RESTAURANT.SAM mit variablem Satzformat

Ausgabe: SAM-Datei RESTAURANT.SAM.SORT mit variablem Satzformat

Übung zu:

- Position des Sortierfeldes ermitteln
- SORT aufrufen
- Sortierbefehl geben: SORT-RECORDS FIELDS=*FIELD-EXPLICIT

Die Sätze der Eingabedatei RESTAURANT.SAM sind wie folgt strukturiert:

SL	Restaurantname	Straße (Sortierfeld)	Telefon	Art der Küche
1	5	25	52	60 70
	Orlando's	Thompson Street 62	220061	Italian

Der vollständige Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“ auf Seite 389](#) dargestellt.

Dateimerkmale der Datei RESTAURANT.SAM

```
/show-file-attributes file-name=restaurant.sam, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.SAM
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)       REC-SIZE  = 0
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM = *K          SO-MIGR   = *ALLOWED
```

RESTAURANT.SAM ist eine Datei mit variabler Satzlänge. Deshalb muss zu der Position des ersten Datenbytes die Länge des Satzlängenfelds addiert werden. Das 'O' von „Orlando's“ steht also an Position 5 im Satz.

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/14:17:05/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.sam,output-file=restaurant.sam.sort
//sort-records fields=*field-explicit(position=25,length=27)
//set-record-attributes output=*variable(maximum-record-size=74)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/14:17:06/000000.29 SORT/MERGE COMPLETED

```

Inhalt der Ausgabedatei RESTAURANT.SAM.SORT (Auszug)

SL	Restaurantname	Straße (Sortierfeld)	Telefon	Art der Küche	
1	5	25	52	60	74
	Golden Fleece	Arran Street 44	242437	Yugoslavian	
	Java	Hope Street 51	522221	Indonesian	
	Le Gourmet	Lime Street 46	505397	French	

```

/show-file-attributes file-name=restaurant.sam.sort, -
/
  information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.SAM.SORT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE   IO(PERF) = STD            DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N)         REC-SIZE = 74
% AVAIL = *STD
% WORK-FILE = *NO          F-PREFORM = *K          SO-MIGR = *ALLOWED

```

Die mit der Anweisung SET-RECORD-ATTRIBUTES festgelegte Satzlänge wird von SORT in den Katalog übernommen.

10.5.3 Beispiel 3: Vollsortieren ISAM-Eingabedatei in SAM-Ausgabedatei

Eingabe: ISAM-Datei RESTAURANT.ISAM mit variablem Satzformat

Ausgabe: SAM-Datei RESTAURANT.ISAM.SORT.SAM mit variablem Satzformat

Übung zu:

- Position des Sortierfeldes ermitteln
- SORT aufrufen
- Sortierbefehl geben: SORT-RECORDS FIELDS=*FIELD-EXPLICIT
- ISAM-Datei in SAM-Datei umwandeln

Kommentar

Bei der folgenden Sortierung müssen zwei Dinge beachtet werden:

1. die Datei-Typen sind unterschiedlich (ISAM-Eingabe- und SAM-Ausgabedatei).
2. Die Position des ersten Datenbytes des Sortierfeldes verschiebt sich um das Satzlängenfeld und den ISAM-Schlüssel nach hinten.

Der erste Punkt hat Auswirkungen auf das ADD-FILE-LINK-Kommando, der zweite Punkt auf die Berechnung der Position des Sortierfeldes.

Standardmäßig legt SORT die als SORTOUT zugewiesene Datei mit denselben Dateimerkmalen an, die auch die SORTIN-Datei aufweist. Soll dies nicht so sein, muss der neue Typ explizit angegeben werden. Dies geschieht im SET-FILE-LINK-Kommando mit dem Operanden ACCESS-METHOD=*SAM.

Inhalt der Eingabedatei RESTAURANT.ISAM

SL	ISAM-Schlüssel	Restaurantname	Straße (Sortierfeld)	Telefon	Art der Küche
1	5	13	33	60	68
	00010000	Orlando's	Thompson Street 62	220061	Italian
	00020000	Java	Hope Street 51	522221	Indonesian
	00030000	Golden Fleece	Arran Street 44	242437	Yugoslavian
	00040000	Le Gourmet	Lime Street 46	505397	French
	00050000	Palenque Mexico	Millwood Drive 2	980149	Mexican
	00060000	Strawberry	Sauchiehall Street 8	595521	Vegetarian
	00070000	Persepolis	Salford Square 20	597004	Persian
	00080000	Vietnam	Thurston Street 47	522518	Vietnamese
	00090000	Chayota's	Thurston Street 60	292742	Japanese
	00100000	Willi's Bar	Westland Street 113	748293	German

Ablaufprotokoll

```

/add-file-link link-name=sortin,file-name=restaurant.isam
/create-file file-name=restaurant.isam.sort.sam
/add-file-link link-name=sortout,file-name=restaurant.isam.sort.sam, -
/          access-method=sam
/start-sort
% SRT1001 2014-10-12/14:17:13/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit(position=13,length=20)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/14:17:14/000000.23 SORT/MERGE COMPLETED

```


Struktur und Inhalt der Ausgabedatei RESTAURANT.ISAM.SORT.SAM

SL	ursprünglicher ISAM-Schlüssel	Restaurantname	Straße (Sortierfeld)	Telefon	Art der Küche
1	5	13	33	60	68
	00090000	Chayota's	Thurston Street 60	292742	Japanese
	00030000	Golden Fleece	Arran Street 44	242437	Yugoslavian
	00020000	Java	Hope Street 51	522221	Indonesian
	00040000	Le Gourmet	Lime Street 46	505397	French
	00010000	Orlando's	Thompson Street 62	220061	Italian
	00050000	Palenque Mexico	Millwood Drive 2	980149	Mexican
	00070000	Persepolis	Salford Square 20	597004	Persian
	00060000	Strawberry	Sauchiehall Street 8	595521	Vegetarian
	00080000	Vietnam	Thurston Street 47	522518	Vietnamese
	00100000	Willi's Bar	Westland Street 113	748293	German

Dateimerkmale der Ausgabedatei RESTAURANT.ISAM.SORT.SAM

```

/show-file-attributes file-name=restaurant.isam.sort.sam, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.ISAM.SORT.SAM
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE  IO(PERF)   = STD            DISK-WRITE = IMMEDIATE
% REC-FORM    = (V,N)       REC-SIZE   = 2044
% AVAIL       = *STD
% WORK-FILE   = *NO         F-PREFORM  = *K            SO-MIGR    = *ALLOWED

```

Kommentar

Der ursprüngliche ISAM-Schlüssel ist Teil des Datensatzes. Bei der Bearbeitung der Datei mit einem Editor (z.B. EDT) ist er normalerweise nicht sichtbar. In SAM-Dateien hat er jedoch keine DVS-Funktion und ist deshalb sichtbar.

Die ISAM-Schlüssel sind ein Indikator für die veränderte (sortierte) Reihenfolge der Datensätze. Entscheidend für die Reihenfolge der Datensätze ist jedoch das Sortierfeld, welches bei Position 13 beginnt.

10.5.4 Beispiel 4: Vollsortieren von ISAM-Dateien mit variablem Satzformat

Eingabe: ISAM-Datei RESTAURANT.ISAM mit variablem Satzformat

Ausgabe: ISAM-Datei RESTAURANT.ISAM.SORT mit variablem Satzformat

Übung zu:

- Registrierung der ISAM-Eigenschaften der Eingabe- und der Ausgabedatei

SORT legt die Ausgabedatei mit neuen Dateimerkmalen an. Der Sortierbegriff bestimmt dabei Position und Länge des ISAM-Schlüssels der Ausgabedatei.

Bei der Verwendung von ISAM-Dateien als Ausgabedateien eines Sortierlaufs ist insbesondere Folgendes zu beachten:

- Es darf nur aufsteigend sortiert werden.
- Die Sortierfelder müssen unmittelbar aneinander anschließen.

Inhalt der Eingabedatei RESTAURANT.ISAM (Auszug)

SL	ISAM-Schlüssel	Restaurantname	Straße (Sortierfeld)	Telefon	Art der Küche
1	5	13	33	60	68
	00010000	Orlando's	Thompson Street 62	220061	Italian
	00020000	Java	Hope Street 51	522221	Indonesian
	00030000	Golden Fleece	Arran Street 44	242437	Yugoslavian
	00040000	Le Gourmet	Lime Street 46	505397	French

Der vollständige Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“](#) auf [Seite 389](#) dargestellt.

Dateimerkmale der Datei RESTAURANT.ISAM

```
/show-file-attributes file-name=restaurant.isam, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.ISAM
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN      = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE  IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM    = (V,N)       REC-SIZE    = 0
% KEY-LEN     = 8           KEY-POS     = 5
% AVAIL       = *STD
% WORK-FILE   = *NO         F-PREFORM   = *K          SO-MIGR    = *ALLOWED
```

Die Datei weist die für den Editor EDT typische Schlüsselposition und Schlüssellänge auf.

Ablaufprotokoll

```

/start-sort
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.isam,output-file=restaurant.isam.sort
//sort-records fields=*field-explicit(position=13,length=20)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/14:17:22/000000.27 SORT/MERGE COMPLETED
    
```

In der Ausgabedatei im ISAM-Format entspricht die Definition des Sortierfeldes dem neuen ISAM-Schlüssel. Je nach Länge und Position des neuen Schlüssels kann die Ausgabedatei nicht mehr mit dem Programm EDT bearbeitet werden. Bei diesem Beispiel entspricht weder Position noch Länge des Schlüssels den von EDT vorausgesetzten Größen.

Struktur und Inhalt der Ausgabedatei RESTAURANT.ISAM.SORT (Auszug)

SL	(ISAM-Schl. der Eingabedatei)	Restaurantname (ISAM-Schlüssel)	Straße	Telefon	Art der Küche
1	5	13	33	60	68
	00090000	Chayota's	Thurston Street 60	292742	Japanese
	00030000	Golden Fleece	Arran Street 44	242437	Yugoslavian
	00020000	Java	Hope Street 51	522221	Indonesian

Dateimerkmale der Ausgabedatei RESTAURANT.ISAM.SORT

```

/show-file-attributes file-name=restaurant.isam.sort, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.ISAM.SORT
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE  IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM    = (V,N)       REC-SIZE   = 2048
% KEY-LEN     = 20          KEY-POS    = 13
% AVAIL       = *STD
% WORK-FILE   = *NO         F-PREFORM  = *K          SO-MIGR    = *ALLOWED

```

ISAM-Schlüssel-Position und ISAM-Schlüssel-Länge entsprechen nun den Definitionen des Sortierfeldes.

10.5.5 Beispiel 5: Vollsortieren von mehreren Dateien mit variablem Satzformat

Eingabe: ISAM-Datei CULTURE.ISAM.1 mit variablem Satzformat
 ISAM-Datei CULTURE.ISAM.2 mit variablem Satzformat
 ISAM-Datei CULTURE.ISAM.3 mit variablem Satzformat

Ausgabe: SAM-Datei CULTURE.SAM.SORT mit variablem Satzformat

Übung zu:

- Zuweisung mehrerer Eingabedateien für Sortier- und Mischlauf
- Feststellen, ob Zuweisung erfolgt ist
- Veränderung der Dateieigenschaften der Ausgabedatei

Vorbemerkung

Es ist möglich, mehrere Dateien in einem Lauf zu sortieren und das Ergebnis in einer Datei sortiert abzulegen. Wenn die Eingabedateien bereits in sich nach dem gemeinsamen Sortierkriterium sortiert sind, sollte ein Mischlauf gestartet werden. Für letzteren Fall bietet SORT den Befehl MERGE-RECORDS.

Inhalt der Eingabedatei CULTURE.ISAM.1 (Auszug)

SL	ISAM-Schlüssel	Restaurantname (Sortierfeld)	Straße	Telefon	Art der Küche
1	5	13	34	56	66
	00010000	Aquitaine	Acacia Avenue 39	284028	French
	00020000	August Gardens	Newton Street 16	2604106	Argentinian
	00030000	Bosna	Freeling Street 11	64115447	Yugoslavian

Der vollständige Inhalt dieser und der übrigen in diesem Beispiel verwendeten Dateien ist im [Abschnitt „Verzeichnis der Beispieldateien“ auf Seite 389](#) dargestellt.

Ablaufprotokoll

```

/add-file-link link-name=sortin01,file-name=culture.1.isam
/add-file-link link-name=sortin02,file-name=culture.2.isam
/add-file-link link-name=sortin03,file-name=culture.3.isam
/create-file file-name=culture.sam.sort
/add-file-link link-name=sortout,file-name=culture.sam.sort, -
/          access-method=*sam
/show-file-link
%-- LINK-NAME ----- FILE-NAME -----
%   SORTIN01           :CTID:$EXAMPLE.CULTURE.1.ISAM
%   SORTIN02           :CTID:$EXAMPLE.CULTURE.2.ISAM
%   SORTIN03           :CTID:$EXAMPLE.CULTURE.3.ISAM
%   SORTOUT            :CTID:$EXAMPLE.CULTURE.SAM.SORT
/start-sort
% SRT1001 2014-10-12/14:17:30/000000.00 SORT/MERGE STARTED,
      VERSION 08.OA00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//sort-records fields=*field-explicit( -
//          position=13,length=20,sorting-order=*descending)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....15 (FROM 01)
% SRT1016 SORT/MERGE INPUT RECORDS:.....16 (FROM 02)
% SRT1016 SORT/MERGE INPUT RECORDS:.....14 (FROM 03)
% SRT1017 RECORDS TO BE SORTED/MERGED:.....45
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....45
% SRT1002 2014-10-12/14:17:31/000000.26 SORT/MERGE COMPLETED

```

SORT meldet für jede Eingabedatei die eingelesenen Sätze, die Summe der zu sortierenden Sätze und die Anzahl der Ausgabesätze.

Inhalt der Ausgabedatei CULTURE.SAM.SORT (Auszug):

00150000	Zung-Hua	Bond Street 33	555320	Chinese
00140000	Zigzag	Taylor Drive 72	226750	Argentinian
00160000	Ziggy's	Leslie Street 72	390092	Argentinian
00150000	Zagreb	Sutherland Drive 8	6515509	Yugoslavian
00130000	Why Not?	Wimbledon Drive 11	399936	French
00120000	Watermill	Leslie Street 33	348000	Swiss
00140000	Veracruz	Landers Street 207	5702520	Mexican
00130000	Venezia	Landers Street 84	847414	Italian
00110000	Torino	Gardener Lane 8	469571	Italian
00100000	Tivoli	Wilberforce Drive 52	221274	Italian
00120000	Tivoli	Wilberforce Drive 52	221274	Italian
00110000	Tai Tung	Acacia Avenue 77	281104	Chinese
00100000	Sultana	Fulton Street 28	332871	Indian
00140000	Spiros	Upman Street 65	366883	Greek
00130000	Slavonia	Allcock Street 16	564906	Yugoslavian
00120000	Siracusa	Polson Street 33	770613	Italian
00090000	Scorpio	Leslie Street 35	399897	Greek
00110000	Saint George's	Upman Street 67	363666	English
00080000	Opatija	Robertson Street 2	268353	Yugoslavian
00100000	Nitaya	Thompson Street 19	197772	Thai
.
.
.
00060000	Don Quixote	Billington Street 6	342318	Spanish
00050000	Datscha	King Street 3	341218	Russian
00040000	China House	May Street 20	531620	Chinese
00030000	Canton	Theresa Street 49	522185	Chinese
00020000	Buenos Aires	Zoo Road 22	779646	Argentinian
00040000	Bouillabaisse	Falcon Street 10	297909	French
00030000	Bosna	Freeling Street 11	64115447	Yugoslavian
00050000	Bologna	Leslie Street 23	393939	Italian
00040000	Baltic Grillhouse	Dartford Road 33	554401	Yugoslavian
00020000	August Gardens	Newton Street 16	2604106	Argentinian
00030000	Aubergine	Penman Street 19	674829	Argentinian
00020000	Auberger	Richmond Drive 15	347577	French
00010000	Asado Steak	Taylor Drive 1	294577	Argentinian
00010000	Aquitaine	Acacia Avenue 39	284028	French
00010000	Alcazar's	Doubleway Drive 39	8111590	Argentinian

Der hier zusätzlich ausgedruckte ehemalige ISAM-Schlüssel ist ein Anzeiger für die geänderte Reihenfolge aus den einzelnen Dateien. Die ISAM-Schlüssel-Verweise treten auch mehrfach auf (vgl. die letzten drei Sätze), da ISAM-Schlüssel für jede Datei getrennt angelegt werden.

10.5.6 Beispiel 6: Vollsortieren (Eingabedatei=Ausgabedatei)

Eingabe: SAM-Datei RESTAURANT.SAM.EINAUS mit variablem Satzformat

Ausgabe: SAM-Datei RESTAURANT.SAM.EINAUS mit variablem Satzformat

Übung zu:

- Zuweisen derselben Datei als Ein- und Ausgabedatei
- Verwendung einer symbolischen Feldbezeichnung

Die Datei RESTAURANT.SAM.EINAUS entspricht inhaltlich der Datei RESTAURANT.SAM. Die Datei für dieses Beispiel wurde erzeugt, indem RESTAURANT.SAM kopiert wurde.

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/14:17:38/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.einaus,output-file=restaurant.einaus
//add-symbolic-name fields=restaurantname(position=5,length=20)
//sort-records fields=*field-symbolic(restaurantname)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/14:17:39/000000.27 SORT/MERGE COMPLETED

```

Inhalt der Ausgabedatei (Auszug)

Chayota's	Thurston Street 60	292742	Japanese
Golden Fleece	Arran Street 44	242437	Yugoslavian
Java	Hope Street 51	522221	Indonesian

Kommentar

In der ASSIGN-FILES-Anweisung werden für die Ein- und Ausgabedatei gleiche Namen verwendet. Die Eingabedatei wird so zur Ausgabedatei.

Zur Definition des Sortierkriteriums wird ein symbolischer Name benutzt. Statt mit *FIELD-EXPLICIT Position und Länge zu definieren, kann auf die bereits erfolgte Definition durch die Anweisung ADD-SYMBOLIC-NAMES zurückgegriffen werden. Dabei wird der vorher definierte symbolische Name verwendet.

Nach der Sortierung wird das Ergebnis der Sortierung unter demselben Namen wie die Eingabedatei gespeichert.

10.5.7 Beispiel 7: Vollsordieren EBCDIC nach DIN-Norm-Ordnung

Eingabe: SAM-Datei DEUTSCH mit variablem Satzformat

Ausgabe: SAM-Datei DEUTSCH.DIN mit variablem Satzformat

Übung zu:

- Änderung der Standardvorgabe für Format
- Vergleich der verschiedenen Ergebnisse

Vorbemerkung

Manche Sortierungen verlangen eine andere Reihenfolge der Sortierung als die durch EBCDIC vorgegebene. Ein Beispiel hierfür ist die alphabetische Sortierung nach DIN-Reihenfolge.

SORT bietet hierfür beim Parameter FORMAT die Möglichkeit, die Reihenfolge der Sortierung u.a. nach DIN vorzugeben. Das folgende Beispiel zeigt die Sortierung nach DIN im Unterschied zur Reihenfolge nach EBCDIC.

Die DIN-Reihenfolge ist dadurch gekennzeichnet, dass

- Groß- und Kleinschreibung gleichgesetzt wird,
- die Umlaute ä, ö, ü wie ae, oe und ue behandelt werden und
- das scharfe S (ß) wie ss behandelt wird.

Diese Sortierung ist z.B. für die Sortierung von Namenslisten mit Umlauten prädestiniert. Das folgende Beispiel zeigt die Auswirkungen der DIN-Sortierung im Gegensatz zur EBCDIC- Sortierreihenfolge.

Ablaufprotokoll

```
/sort-file input-files=deutsch,output-file=deutsch.din, -
/      fields=*field-explicit(position=5,length=8,format=*ebcdic-din)
% SRT1001 2014-10-12/14:17:21/000000.00 SORT/MERGE STARTED,
      VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
% SRT1016 SORT/MERGE INPUT RECORDS:.....34 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....34
% SRT1002 2014-10-12/14:17:22/000000.27 SORT/MERGE COMPLETED
```

Kommentar

Da in diesem Fall eine einfache Sortierung vorgenommen werden soll, wird das Kommando SORT-FILE verwendet. Zu beachten ist hier der Zusatz FORMAT=*EBCDIC-DIN, der die besondere DIN-Sortierung festlegt.

Ergebnis verschiedener Sortierungen mit der Eingabedatei DEUTSCH

Eingabedatei	EBCDIC-Reihenfolge	DIN-Reihenfolge
scherzen	allgemein	Abendland
allgemein	astronomisch	ändern
überlegen	brasilianisch	allgemein
mexikanisch	bretonisch	Allgemeinheit
Mehrzahl	dichterisch	Astronomie
oder	englisch	astronomisch
Trennung	französisch	Biologie
Verhältniswort	Österreich	brasilianisch
Stilkunde	Übertragung	Brasilien
brasilianisch	mechanisch	bretonisch
philosophisch	medizinisch	Deutschland
medizinisch	mexikanisch	dichterisch
ändern	oder	Druckersprache
dichterisch	philosophisch	englisch
Druckersprache	scherzen	französisch
Astronomie	scherzhaft	Franzosen
mechanisch	ändern	Landwirtschaft
Übertragung	öffentlich	Mechanik
scherzhaft	überlegen	mechanisch
bretonisch	Abendland	medizinisch
Biologie	Allgemeinheit	Mehrzahl
Brasilien	Astronomie	mexikanisch
Allgemeinheit	Biologie	oder
öffentlich	Brasilien	öffentlich
astronomisch	Deutschland	Österreich
Mechanik	Druckersprache	philosophisch
Franzosen	Franzosen	Scherz
englisch	Landwirtschaft	scherzen
Landwirtschaft	Mechanik	scherzhaft
französisch	Mehrzahl	Stilkunde
Abendland	Scherz	Trennung
Österreich	Stilkunde	überlegen
Scherz	Trennung	Übertragung
Deutschland	Verhältniswort	Verhältniswort

Bei Sortierungen von Namenslisten ist es nicht sinnvoll, dass alle Kleinbuchstaben vor dem Großbuchstaben 'A' und Umlaute zwischen den Kleinbuchstaben einsortiert werden. Bei der Sortierung nach DIN werden Klein- und Großbuchstaben gleich behandelt und Umlaute an der alphabetisch richtigen Stelle eingeordnet.

10.5.8 Beispiel 8: Vollsortieren FORMAT=*MODIFY-CODE

Eingabe: SAM-Datei MODCOD.INPUT mit variablem Satzformat

Ausgabe: SAM-Datei MODCOD.OUTPUT mit variablem Satzformat

Übung zu:

- Festlegen einer von EBCDIC abweichenden Zeichenreihenfolge
- Eingabe der Zeichendefinition als Buchstabenpaare oder Paare hexadezimaler Zeichen im Kommando MODIFY-CODE

Inhalt der Eingabedatei MODCOD.INPUT

U
o
A
u
O
a
ö
Ä
Ö
ä
Ü

Kommentar

Anwendung findet diese Option, wenn vorgegebene Sortierfolgen nicht der gewünschten entsprechen (z.B. französische accents oder Umlaute).

Diese Datei soll die Möglichkeit illustrieren, dass besondere Zeichen auch besondere Behandlung erfordern können. Die Sortierreihenfolge wird wie folgt definiert:

Das Zeichen 'a' soll unmittelbar hinter 'A' einsortiert werden, 'Ä' (X'8B') hinter 'a' (X'81') und 'ä' (X'AB') hinter 'Ä'. Mit den Zeichen 'O', 'o' (X'96'), 'Ö' (X'8C') und 'ö' (X'AC') sowie den Zeichen 'U', 'u' (X'A4'), 'Ü' (X'8D') und 'ü' (X'AD') soll entsprechend verfahren werden. Den Sedezimalangaben liegt der Code EBCDIC.SRV.10 zu Grunde.

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/14:17:38/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=modcode.input,output-file=modcode.output
//sort-records fields=*field-explicit( -
//          position=5,length=1,format=*modify-code)
//modify-code sequences= -
//          (c'Aa',x'818b',x'8bab',c'0o',x'968c',x'8cac',c'Uu',x'a48d',x'8dad')
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....12 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....12
% SRT1002 2014-10-12/14:17:39/000000.27 SORT/MERGE COMPLETED

```

Inhalt der Ausgabedatei MODCOD.OUTPUT

```

A
a
Ä
ä
O
o
Ö
ö
U
u
Ü
ü

```

10.5.9 Beispiel 9: Vollsortieren FORMAT=*EXTENDED-CHARACTER und FORMAT=*TRANSLATE-CHARACTER

Eingabe: SAM-Datei XHCS.SAM mit variablem Satzformat

Ausgabe: SAM-Datei XHCS.SAM.EXCHAR mit variablem Satzformat
SAM-Datei XHCS.SAM.TRCHAR mit variablem Satzformat

Übung zu:

- SORT aufrufen
- Entscheidung, welches Sortierformat genommen werden soll
- Sortieranweisung geben:
 SORT-RECORDS FIELDS=*FIELD-EXPLICIT(...FORMAT=*EXTENDED-CHARACTER)
 SORT-RECORDS FIELDS=*FIELD-EXPLICIT(...FORMAT=*TRANSLATE-CHARACTER)

Inhalt der Eingabedatei XHCS.SAM

SL	Sor- tier- feld	
1	5	9

orde ntlich
 anbi eten
 unor dentlich
 Ärge r
 Öffe ntlichkeit
 Über fluss
 Unfa ll
 Orga nisation
 über flüssig
 Aber witz
 ärge rlich
 öffe ntlich

Dateiattribute der Eingabedatei

```

/show-file-attributes file-name=xhcs.sam, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.XHCS.SAM
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)  = STD            DISK-WRITE = IMMEDIATE
% REC-FORM    = (V,N)       REC-SIZE  = 0
% COD-CH-SET  = EDF03DRV
% AVAIL       = *STD
% WORK-FILE   = *NO         F-PREFORM = *K          SO-MIGR    = *ALLOWED

```

Anmerkungen zur Sortierung mit verschiedenen Formatangaben

Bei manchen Sortierungen ist es notwendig, ein besonderes Sortierformat anzugeben, um eine den lokalen Regeln entsprechende Reihenfolge zu erhalten. Dies kann gelten, wenn ein CCS verwendet wird.

Die hier gezeigte Datei soll gemäß deutschen Erwartungen sortiert werden; dies ist erkennbar an den Umlauten, die eine besondere Sortierung verlangen können. Vor der Sortierung sollte Folgendes berücksichtigt werden:

- ohne Formatangabe verwendet SORT das im Katalogeintrag der Eingabedatei gefundene CCS. Dies kann bedeuten, dass Umlaute wie Sonderzeichen behandelt und an den Anfang einer Sortierliste gestellt werden;
- die Sortierung nach der Code-Reihenfolge des CCS (hier EDF03DRV) positioniert Umlaute so, dass sie immer nach den Vokalen liegen (FORMAT=*EXTENDED-CHARACTER). In anderen CCS können andere Regeln festgelegt sein.
- die Sortierung mit der Formatangabe *TRANSLATE-CHARACTER ersetzt für die Sortierung Zeichen gemäß den vorgegebenen Gleichsetzungstabellen (hier wurden die Tabellen von [Seite 51f](#) verwendet) und sortiert dann nach der Code-Reihenfolge des CCS (hier EDF03DRV). Die Verwendung vorgegebener oder selbstdefinierter Tabellen gewährleistet eine optimale Sortierung nach eigenen Spezifikationen.
- Die bereits in „[Beispiel 7: Vollsortieren EBCDIC nach DIN-Norm-Ordnung](#)“ auf [Seite 322](#) besprochene Sortierung nach DIN setzt Umlaute um und Groß- und Kleinbuchstaben gleich und sortiert nach der DIN-Vorschrift. Umsetzungstabellen können nicht definiert werden.

Das folgende Beispiel soll dies verdeutlichen.

Die Buchstabenfolgen des Originals werden je nach Angabe des FORMAT-Operanden sortiert:

Original	Nach Sortierung mit EXTENDED-CHARACTER (CCS: EDF03DRV)	Nach Sortierung mit TRANSLATE-CHARACTER (CCS: EDF03DRV)
oooooooo	oooooooo	ööööööö
öööööööö	00000000	ÖÖÖÖÖÖ
00000000	öööööööö	oooooooo
ÖÖÖÖÖÖ	ÖÖÖÖÖÖ	00000000

Die Sortierung mit FORMAT=*EXTENDED-CHARACTER platziert zuerst die Vokale (und dabei Kleinbuchstaben vor Großbuchstaben) und dann die Umlaute.

Bei Verwendung von FORMAT=*TRANSLATE-CHARACTER wird die Zeichenkette „öööööööö“ wie „oeoeoeoeoeoeoe“ behandelt und liegt deshalb im Alphabet weiter vorn als „oooooooo“. In diesem Beispiel liegen die Ö-Umlaute also vor dem Doppel-O. In einer Reihe mit „aaaa“ und „ääää“ lägen die „aaaa“s vor den „ääää“s, da „aa“ vor „ae“ liegt.

Ablauf der Sortierungen

```

/start-sort
% SRT1001 2014-10-12/14:17:38/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=xhcs.sam,output-file=xhcs.sam.exchar
//sort-records fields=*field-explicit( -
//                               position=5,length=4,format=*extended-character)
//set-record-attributes output=*variable(maximum-record-size=40)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....12 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....12
% SRT1002 2014-10-12/14:17:39/000000.31 SORT/MERGE COMPLETED

```



```

/start-sort
% SRT1001 2014-10-12/14:19:07/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=xhcs.sam,output-file=xhcs.sam.trchar
//sort-records fields=*field-explicit( -
//          position=5,length=4,format=*translate-character)
//set-record-attributes output=*variable(maximum-record-size=40)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....12 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....12
% SRT1002 2014-10-12/14:19:26/000000.27 SORT/MERGE COMPLETED
    
```

Ergebnis der Sortierung (ohne Satzlängenfeld)

Eingabedatei XHCS.SAM	Ausgabedatei XHCS.SAM.EXCHAR	Ausgabedatei XHCS.SAM.TRCHAR
ordentlich	anbieten	ärgerlich
anbieten	Aberwitz	Ärger
unordentlich	ärgerlich	Aberwitz
Ärger	Ärger	anbieten
Öffentlichkeit	ordentlich	öffentlich
Überfluss	Organisation	Öffentlichkeit
Unfall	öffentlich	ordentlich
Organisation	Öffentlichkeit	Organisation
überflüssig	unordentlich	Überfluss
Aberwitz	Unfall	überflüssig
ärgerlich	überflüssig	unordentlich
öffentlich	Überfluss	Unfall

Dateiattribute der sortierten Dateien

```

/show-file-attributes file-name=xhcs.sam, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.XHCS.SAM.EXCHAR
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)      REC-SIZE   = 40
% COD-CH-SET = EDF03DRV
% AVAIL      = *STD
% WORK-FILE  = *NO        F-PREFORM  = *K          SO-MIGR   = *ALLOWED
%0000000003 :CTID:$EXAMPLE.XHCS.SAM.TRCHAR
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)      REC-SIZE   = 40
% COD-CH-SET = EDF03DRV
% AVAIL      = *STD
% WORK-FILE  = *NO        F-PREFORM  = *K          SO-MIGR   = *ALLOWED

```

10.5.10 Beispiel 10: Vollsordieren mit Summenbildung und SELECT-INPUT-RECORDS

Eingabe: SAM-Datei CLIENT.SELECT mit variablem Satzformat

Ausgabe: SAM-Datei CLIENT.SELECT.SORT mit variablem Satzformat

Übung zu:

- Sätze sortieren
- Teil eines Satzes als Zahl definieren
- die definierten Zahlen gleicher Sätze summieren
- Sätze zur Summierung und Sortierung auswählen
- Summenfeld erweitern, um Überlauf zu verhindern

Vorbemerkung

Ein Anwendungsgebiet für das hier gewählte Beispiel könnte die Bilanzerstellung (Summierung) nach Kundennummer sein, wobei die Bestellungen eines bestimmten Monats nicht berücksichtigt werden sollen.

Für dieses Beispiel müssen deshalb diese Angaben gemacht werden:

- wonach soll sortiert werden (SORT-RECORDS)
- wo befindet sich das Summenfeld (SUM-RECORDS)
- überschreitet die zu erwartende Summe die bisherige Feldlänge?
- welche Sätze sollen nicht berücksichtigt werden (SELECT-INPUT-RECORDS)

Das Resultat der Summierung ist in der Datei CLIENT.SELECT.SUM zu sehen.

Inhalt der Eingabedatei CLIENT.SELECT

SL	Sortierfeld	Summenfeld		Ver-gleichs-feld	
1	5	14	21	32	38
	CLIENT-A	700000	MOTOR	MAI	EINGABESATZ1
	CLIENT-F	000083	AUTO	JAN	EINGABESATZ2
	CLIENT-E	000700	MOTOR	AUG	EINGABESATZ3
	CLIENT-D	076000	AUTO	JUN	EINGABESATZ4
	CLIENT-B	006900	MOTOR	AUG	EINGABESATZ5
	CLIENT-G	000070	AUTO	JAN	EINGABESATZ6
	CLIENT-J	006700	AUTO	JUN	EINGABESATZ7
	CLIENT-A	800000	MOTOR	AUG	EINGABESATZ8
	CLIENT-K	075600	AUTO	JAN	EINGABESATZ9
	CLIENT-S	000099	AUTO	JUN	EINGABESATZ10
	CLIENT-C	000001	MOTOR	AUG	EINGABESATZ11
	CLIENT-B	010000	AUTO	JAN	EINGABESATZ12
	CLIENT-A	030000	AUTO	JUN	EINGABESATZ13
	CLIENT-W	008700	AUTO	JAN	EINGABESATZ14
	CLIENT-D	024000	MOTOR	AUG	EINGABESATZ15
	CLIENT-E	001350	AUTO	JAN	EINGABESATZ16
	CLIENT-C	999999	AUTO	JAN	EINGABESATZ17
	CLIENT-J	000305	MOTOR	AUG	EINGABESATZ18
	CLIENT-X	123456	AUTO	JAN	EINGABESATZ19
	CLIENT-A	600000	MOTOR	MAI	EINGABESATZ20
	CLIENT-T	000058	AUTO	JUN	EINGABESATZ21
	CLIENT-Y	005000	AUTO	JAN	EINGABESATZ22

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/09:31:48/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=client.select,output-file=client.select.sum
//sort-records fields=*field-explicit(position=5,length=8)
//sum-records fields=*field-explicit( -
//      position=14,length=6,format=*zoned-decimal,field-extension=2)
//select-input-records condition=(32,3,ch<>'JUN')
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....22 (FROM 01)
% SRT1024 DELETED SELECT-INPUT-RECORDS RECORDS:.....5
% SRT1017 RECORDS TO BE SORTED/MERGED:.....17
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....12
% SRT1020 DELETED SUM RECORDS:.....5
% SRT1002 2014-10-12/09:32:00/000000.29 SORT/MERGE COMPLETED
    
```

Inhalt der Datei CLIENT.SELECT.SUM

SL		Summen- feld mit Summen- feldverlän- gerung			
1	5	14	23	34	40
	CLIENT-A	02100000	MOTOR	MAI	EINGABESATZ1
	CLIENT-B	00016900	AUTO	JAN	EINGABESATZ12
	CLIENT-C	01000000	MOTOR	AUG	EINGABESATZ11
	CLIENT-D	00024000	MOTOR	AUG	EINGABESATZ15
	CLIENT-E	00002050	AUTO	JAN	EINGABESATZ16
	CLIENT-F	00000083	AUTO	JAN	EINGABESATZ2
	CLIENT-G	00000070	AUTO	JAN	EINGABESATZ6
	CLIENT-J	00000305	MOTOR	AUG	EINGABESATZ18
	CLIENT-K	00075600	AUTO	JAN	EINGABESATZ9
	CLIENT-W	00008700	AUTO	JAN	EINGABESATZ14
	CLIENT-X	00123456	AUTO	JAN	EINGABESATZ19
	CLIENT-Y	00005000	AUTO	JAN	EINGABESATZ22

Kommentar

Die Ausgabedatei enthält nicht mehr die Sätze, die durch die SELECT-INPUT-RECORDS-Anweisung ausgesondert wurden. Ebenso findet sich für alle Sätze, bei denen das Sortierkriterium übereinstimmt, nur noch ein Satz in der Datei. Dafür wurde für diese identischen Sätze im Summenfeld addiert. Die Summe für das Sortierkriterium Kunde steht nun im Summenfeld.

Beispiel

CLIENT-A weist im Eingabe-Summenfeld die Einträge 700000, 800000, 30000 und 600000 auf. Der Eintrag 30000 erfolgte jedoch im JUN, der generell nicht berücksichtigt werden sollte. Die Summe der übrigen drei Werte ergibt 2100000. Dieses Ergebnis wurde für CLIENT-A im Summenfeld eingetragen.

Durch die Angabe der Summenfeldverlängerung wird vermieden, dass es zu Überläufen kommt.

10.5.11 Beispiel 11: Auswahlsortieren von Sätzen mit variablem Satzformat

Eingabe: SAM-Datei RESTAURANT.SAM mit variablen Satzformat

Ausgabe: SAM-Datei RESTAURANT.SAM.SELECT mit variablem Satzformat

Übung zu:

- Definition zweier Sortierfelder
- Definition eines Restfeldes
- Ausschließen eines Teil des Ausgangssatzes (Auswahlsortieren)
- Änderung der durch SORT vorgegebenen Dateimerkmale

Vorbemerkung

Dieses Beispiel soll Folgendes zeigen:

Die Möglichkeiten von SORT,

- Ausgabesätze aus Teilen des Eingabesatzes zusammensetzen
- die Dateiattribute der Ausgabedatei innerhalb von SORT ändern zu können.

Letztere Möglichkeit ist besonders bei diesem Beispiel wichtig. Beim Auswahlsortieren erzeugt SORT unabhängig vom Typ der Eingabedatei eine Datei mit Sätzen fester Länge. Soll die Ausgabedatei mit einem Editor wie EDT weiterverarbeitet werden, ist in den meisten Fällen eine Datei mit Sätzen variabler Länge gewünscht.

Die Änderung der Dateiattribute erfolgt mit der Anweisung SET-RECORD-ATTRIBUTES.

Struktur der Eingabedatei RESTAURANT.SAM

SL	Restaurantname (2. Sortierfeld)	Straße (Restfeld)	Telefon (soll entfallen)	Art der Küche (1. Sortierfeld)
1	5	25	52	60
	Orlando's	Thompson Street 62	220061	Italian

Der vollständige Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“](#) auf [Seite 389](#) dargestellt.

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/09:36:05/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.sam,output-file=restaurant.sam.select
//set-record-attributes output=*variable(maximum-record-size=*std), -
//      filler=' '
//sort-records fields=( -
//      *field-explicit(position=60,length=13), -
//      *field-explicit(position=5,length=20), -
//      *remainder-explicit(position=25,length=27)), -
//      sort-type=*compound-record
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/09:36:14/000000.27 SORT/MERGE COMPLETED

```

Dateimerkmale der Ausgabedatei RESTAURANT.SAM.SELECT

```

/show-file-attributes file-name=restaurant.sam.select, -
/      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.SAM.SELECT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM      BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)    REC-SIZE  = 64
% AVAIL      = *STD
% WORK-FILE  = *NO      F-PREFORM = *K          SO-MIGR    = *ALLOWED

```

SORT trägt im Katalog die Summe der Länge aller Felder als maximale Satzlänge ein.

Inhalt der Ausgabedatei RESTAURANT.SAM.SELECT

SL	Art der Küche (1. Sortierfeld)	Restaurantname (2. Sortierfeld)	Straße (Restfeld)
1	5	18	38
			68
	French	Le Gourmet	Lime Street 46
	German	Willi's Bar	Westland Street 113
	Indonesian	Java	Hope Street 51
	Italian	Orlando's	Thompson Street 62
	Japanese	Chayota's	Thurston Street 60
	Mexican	Palenque Mexico	Millwood Drive 2
	Persian	Persepolis	Salford Square 20
	Vegetarian	Strawberry	Sauchiehall Street 8
	Vietnamese	Vietnam	Thurston Street 47
	Yugoslavian	Golden Fleece	Arran Street 44

Das in der Eingabedatei vorhandene Feld Telefon wurde durch das Auswahlsortieren nicht in die Ausgabedatei übernommen.

10.5.12 Beispiel 12: Auswahlsortieren (binär) von Sätzen mit festem Satzformat

Eingabe: SAM-Datei SAM.BIN.FIX mit festem Satzformat

Ausgabe: SAM-Datei SAM.BIN.FIX.SORT mit festem Satzformat

Übung zu:

- Festlegen einer Bit-Position
- Änderung des FORMAT-Parameters

Inhalt der Eingabedatei SAM.BIN.FIX

abdruckbar	sedezimal	binär			
		1	2	3	
		01234567	0 1234567	01234 567	Byte Bit
NCK	D5C3D2	11010101	1 1000011	11010 010	Satz 1
TUM	E3E4D4	11100011	1 1100100	11010 100	Satz 2
HEL	C8C5D3	11001000	1 1000101	11010 011	Satz 3
ELK	C5D3D2	11000101	1 1010011	11010 010	Satz 4
			Sortierfeld (Länge 1 Byte + 4 Bit = 12 Bit)		

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/12:21:05/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=sam.bin.fix,output-file=sam.bin.fix.sort
//sort-records fields=*field-explicit( -
//          position=2(bit-position=1),length=1(number-of-bits=4), -
//          format=*binary), -
//          sort-type=*compound-record
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....4 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....4
% SRT1002 2014-10-12/12:21:17/000000.22 SORT/MERGE COMPLETED
    
```

Inhalt der Ausgabedatei SAM.BIN.FIX.SORT

sedezimal	binär		
	1	2	
	0 1234567	01234 567	Byte
			Bit
43D0	0 1000011	11010 000	Satz 1
45D0	0 1000101	11010 000	Satz 3
53D0	0 1010011	11010 000	Satz 4
64D0	0 1100100	11010 000	Satz 2
	*) Sortierfeld	*)	

*) Binäre Nullen, die von SORT eingefügt werden, um den Ausgabesatz auf ganze Bytes aufzufüllen

10.5.13 Beispiel 13: Auswahlsortieren einer POSIX-Datei

Eingabe: POSIX-Datei „restaurant“ mit variablem Satzformat

Ausgabe: POSIX-Datei „sorted/restaurant“ mit variablem Satzformat

Übung zu:

- Zuweisen von POSIX-Dateien
- Definition eines Sortierfeldes
- Definition zweier Restfelder
- Ausschließen eines Teil des Ausgangssatzes (Auswahlsortieren)

Vorbemerkung

Die POSIX-Datei „restaurant“ soll nach dem Feld „Art der Küche“ sortiert werden. Die Ausgabedatei soll im Verzeichnis „sorted“ unter dem Namen „restaurant“ abgelegt werden. Sie soll nur noch die Felder „Art der Küche“, „Restaurantname“ und „Telefon“ aus der Eingabedatei enthalten.

Struktur der Eingabedatei „restaurant“

Restaurantname (1. Restfeld)	Straße (soll entfallen)	Telefon (2. Restf.)	Art der Küche (Sortierfeld)
1	21	48	56
Orlando's	Thompson Street 62	220061	Italian
Java	Hope Street 51	522221	Indonesian
Golden Fleece	Arran Street 44	242437	Yugoslavian
Le Gourmet	Lime Street 46	505397	French
Palenque Mexico	Millwood Drive 2	980149	Mexican
Strawberry	Sauchiehall Street 8	595521	Vegetarian
Persepolis	Salford Square 20	597004	Persian
Vietnam	Thurston Street 47	522518	Vietnamese
Chayota's	Thurston Street 60	292742	Japanese
Willi's Bar	Westland Street 113	748293	German

Dateimerkmale der Eingabedatei „restaurant“

```

/start-posix-shell
POSIX Basissshell 10.0A43 created Dec 17 2012
POSIX Shell 08.0A43 created Jul 13 2012
Copyright (C) Fujitsu Technology Solutions 2009
    All Rights reserved
.
.
.
$ls -l restaurant
-rw-r--r--  1 EXAMPLE  USROTHER      690 Jan 24 10:13 restaurant
$exit

```

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/12:46:01/000000.00 SORT/MERGE STARTED,
    VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files='restaurant',output-file='sorted/restaurant'
//sort-records fields=( -
//      *field-explicit(position=56,length=11), -
//      *remainder-explicit(position=1,length=20), -
//      *remainder-explicit(position=48,length=8)), -
//      sort-type=*compound-record
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/12:50:45/000000.47 SORT/MERGE COMPLETED

```

Dateimerkmale der Ausgabedatei „sorted/restaurant“

```

/start-posix-shell
POSIX Basisshell 10.0A43 created Dec 17 2012
POSIX Shell 08.0A43 created Jul 13 2012
Copyright (C) Fujitsu Technology Solutions 2009
    All Rights reserved
.
.
.
$cd sorted
$ls -l restaurant
-rw----- 1 EXAMPLE USROther      400 Jan 24 12:50 restaurant
$exit

```

Inhalt der Ausgabedatei „sorted/restaurant“

Art der Küche (Sortierfeld)	Restaurantname (1. Restfeld)	Telefon (2. Restf.)
1	12	32
French	Le Gourmet	505397
German	Willi's Bar	748293
Indonesian	Java	522221
Italian	Orlando's	220061
Japanese	Chayota's	292742
Mexican	Palenque Mexico	980149
Persian	Persepolis	597004
Vegetarian	Strawberry	595521
Vietnamese	Vietnam	522518
Yugoslavian	Golden Fleece	242437

Das in der Eingabedatei vorhandene Feld „Straße“ wurde durch das Auswahlsortieren nicht in die Ausgabedatei übernommen.

10.5.14 Beispiel 14: Adresslistensortieren von Sätzen mit festem Satzformat

Eingabe: SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat

Ausgabe: SAM-Datei RESTAURANT.SAM.FIX.ADR mit festem Satzformat

Übung zu:

- Verwendung einer SAM-Datei
- Positionierung der von SORT erzeugten Adressen am Ende des in der Ausgabedatei abgelegten Satzes

Vorbemerkung

Adresslistensortierung bezeichnet ein besonderes Sortierverfahren, welches es ermöglicht, über die Sätze in der sortierten Datei auf die Sätze in der Eingabedatei zuzugreifen. SORT vergibt bei dieser Sortierart Adressen, wo die einzelnen Sätze in der Eingabedatei gefunden wurden. Diese Adresse wird je nach Angabe vor (SORT-TYPE=*TAG-HEADER) oder hinter (SORT-TYPE=*TAG-TRAILER) den Datensatz angefügt. Diese Adresse wird auch Wiedergewinnungsadresse genannt.

Diese Sortierart kann z.B. verwendet werden, um einen Index für eine (längere) SAM-Datei einer Datenbank zu erstellen.

Ablaufprotokoll

```

/start-sort
% SRT1001 2014-10-12/09:23:05/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=restaurant.sam.fix, -
//              output-file=restaurant.sam.fix.adr
//sort-records fields=*field-explicit(position=1,length=13), -
//              sort-type=*tag-trailer
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 2014-10-12/09:23:22/000000.29 SORT/MERGE COMPLETED

```

Inhalt der Ausgabedatei

Datenbytes (Sortierfeld)	SAM-Adresse (in sedezimaler Darstellung)	
	Nr. des Datei- blocks	relative Satz- nummer

1	14	17
Chayota's	000001	09
Golden Fleece	000001	03
Java	000001	02
Le Gourmet	000001	04
Orlando's	000001	01
Palenque Mexico	000001	05
Persepolis	000001	07
Strawberry	000001	06
Vietnam	000001	08
Willi's Bar	000001	0A

10.5.15 Beispiel 15: Mischen von Dateien

Eingabe: SAM-Datei CULTURE.SAM.1 mit variablem Satzformat
 SAM-Datei CULTURE.SAM.2 mit variablem Satzformat
 SAM-Datei CULTURE.SAM.3 mit variablem Satzformat

Ausgabe: SAM-Datei CULTURE.SAM.123 mit variablem Satzformat

Vorbemerkung

MERGE-RECORDS mischt bereits sortierte Dateien ineinander. Die Eingabedateien müssen hinsichtlich des Sortierkriteriums bereits sortiert sein. Auf diese Weise wird der rechenzeitintensive Sortiervorgang eingespart.

Inhalt der Eingabedatei CULTURE.SAM.1 (Auszug)

SL	Restaurantname (Sortierfeld)	Straße	Telefon	Art der Küche
1	5	26	48	58
	Aquitaine	Acacia Avenue 39	284028	French
	August Gardens	Newton Street 16	2604106	Argentinian
	Bosna	Freeling Street 11	64115447	Yugoslavian

Der vollständige Inhalt dieser und der übrigen in diesem Beispiel verwendeten Dateien ist im [Abschnitt „Verzeichnis der Beispieldateien“ auf Seite 389](#) dargestellt.

Ablaufprotokoll

```
/start-sort
% SRT1001 2014-10-12/16:19:30/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=(culture.sam.1,culture.sam.2,culture.sam.3), -
//          output-file=culture.sam.123
//merge-records fields=*field-explicit(position=5,length=21)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....15 (FROM 01)
% SRT1016 SORT/MERGE INPUT RECORDS:.....16 (FROM 02)
% SRT1016 SORT/MERGE INPUT RECORDS:.....14 (FROM 03)
% SRT1017 RECORDS TO BE SORTED/MERGED:.....45
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....45
% SRT1002 2014-10-12/16:19:31/000000.26 SORT/MERGE COMPLETED
```

10.5.16 Beispiel 16: Benutzerausgang INPUT

Vollsortieren von Sätzen mit festem Satzformat

Eingabe: SAM-Datei UE.EXIT.INPUT mit festem Satzformat
Ausgabe: SAM-Datei UE.EXIT.OUTPUT mit festem Satzformat
Bibliothek: SORT.EXAMPLE.LIB (benutzerdefiniert)
Modulname: E15
Benutzerausgang: INPUT (PARAMETER-MODE=*ANY)

Übung zu:

- Einbinden eines Moduls aus einer Bibliothek
- Auswahl des Benutzerausgangs
- Festlegen der Adressierungsmethode

Beschreibung des Benutzermoduls

In diesem Beispiel sollen durch eine Benutzeroutine Sätze gelöscht, eingefügt und geändert werden. Die Routine wird für jeden Eingabesatz aktiviert und soll folgende Maßnahmen durchführen:

- Alle Sätze löschen, die mit der Ziffer 8 beginnen.
- Alle Sätze ändern, die mit der Ziffer 9 beginnen. Als erstes Zeichen soll 'X' stehen.
- Beim ersten Satz, der mit der Ziffer 7 beginnt, soll ein Satz eingefügt werden, bei den weiteren Sätzen nicht mehr.

Merkmal aller Benutzeroutinen für Benutzerausgänge ist, dass sie in Modulbibliotheken gespeichert werden müssen.

Quellprogramm-Datei

```

E15      START
        PRINT NOGEN
        ENTRY E15SEL
        USING E15SEL,15
E15SEL  SAVE (14,12)          REGISTER SICHERN
        L 7,12(1)            ADRESSE DES AKTIONSWORTES
E1       CLC 0(4,1),=XL4'00' ENDE?
        BE ENDE              YES
        L 8,0(1)             LADEN ADR DES SATZES
        CLI 0(8),C'8'        1. STELLE DES SATZES=8?
        BE LOESCHEN          YES
        CLI 0(8),C'9'        1. STELLE DES SATZES=9?
        BE AENDERN           YES
WEICHE   NOP NORMAL
        CLI 0(8),C'7'        1. STELLE DES SATZES=7?
        BE EINFUEG
*        K E I N E  A E N D E R U N G
NORMAL  MVI 3(7),X'00'       RUECKSPRUNGKENNZ. NACH AKTIONSWORT
        RETRN (14,12)        SATZ VERARBEITEN
*        S A T Z  L O E S C H E N
LOESCHEN MVI 3(7),X'04'       RUECKSPRUNGKENNZ. NACH AKTIONSWORT
        RETRN (14,12)        SATZ LOESCHEN
*        S A T Z  A E N D E R N
AENDERN MVI 3(7),X'00'       RUECKSPRUNGKENNZ. NACH AKTIONSWORT
        MVI 0(8),C'X'        1.STELLE IM SATZ AENDERN
        RETRN (14,12),       SATZ VERARBEITEN
*        S A T Z  E I N F U E G E N
EINFUEG LA 8,SATZ            ADR DES EINZUFUEGENDEN SATZES
        ST 8,0(1)            ADR VON SATZ NACH VERSORGBEREICH
        OI WEICHE+1,X'FO'    WEICHE AUF SPRUNG SETZEN
        MVI 3(7),X'0C'       RUECKSPRUNGKENNZ. NACH AKTIONSWORT
        RETRN (14,12)        SATZ EINFUEGEN
*        E N D E
ENDE    MVI 3(7),X'08'       RUECKSPRUNGKENNZ. NACH AKTIONSWORT
        RETRN (14,12)        EINGABE-ENDE FUER DIESEN EXIT
SATZ    DS 0CL17
        DC C'EINGEFUEGTER SATZ'
        END E15

```

Erläuterung zum Quellprogramm

Das Quellprogramm wird mit ASSEMBH übersetzt.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen E15.

Dateimerkmale der Eingabedatei UE.EXIT.INPUT

```

/show-file-attributes file-name=ue.exit.input, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.UE.EXIT.INPUT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (F,N)       REC-SIZE  = 17
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM = *K          SO-MIGR   = *ALLOWED

```

Inhalt der Eingabedatei UE.EXIT.INPUT

	Sortier-	
1	5	11 17

```

9999 999999 9999999
2222 222222 2222222
0000 000000 0000000
8888 888888 8888888
5555 555555 5555555
1111 111111 1111111
7777 777777 7777777
8888 888888 8888888
0000 000000 0000000
9999 999999 9999999

```

Ablaufprotokoll

```

/add-file-link link-name=blslib,file-name=sort.example.lib
/start-sort
% SRT1001 2014-10-12/13:20:33/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=ue.exit.input,output-file=ue.exit.output
//assign-exits input=*module(name=e15,parameter-mode=*any)
//sort-records fields=*field-explicit(position=5,length=6), -
//
      estimated-records=10
//end

```

```

% SRT1016 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1018 INSERTED INPUT RECORDS:.....1
% SRT1019 DELETED INPUT RECORDS:.....2
% SRT1017 RECORDS TO BE SORTED/MERGED:.....9
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....9
% SRT1002 2014-10-12/13:26:47/000000.32 SORT/MERGE COMPLETED

```

Dateimerkmale der Ausgabedatei UE.EXIT.OUTPUT

```

/show-file-attributes file-name=ue.exit.output, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.UE.EXIT.OUTPUT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN      = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (F,N)      REC-SIZE   = 17
% AVAIL      = *STD
% WORK-FILE  = *NO        F-PREFORM   = *K          SO-MIGR    = *ALLOWED

```

Inhalt der Ausgabedatei UE.EXIT.OUTPUT

```

EINGEFUEGTER SATZ   dieser Satz wurde eingefügt
000000000000000000
000000000000000000
111111111111111111
222222222222222222
555555555555555555
777777777777777777
X999999999999999999  dieser Satz wurde geändert
X999999999999999999  dieser Satz wurde geändert

```

10.5.17 Beispiel 17: Benutzerausgang OUTPUT

Vollsortieren von Sätzen mit variablem Satzformat

Eingabe: SAM-Datei E35A.INPUT mit variablem Satzformat
Ausgabe: SAM-Datei E35A.OUTPUT mit variablem Satzformat
Bibliothek: SORT.EXAMPLE.LIB
Modulname: E35B
Benutzerausgang: OUTPUT (PARAMETER-MODE=*ANY)

Übung zu:

- Einbinden eines Moduls aus einer Bibliothek
- Zuweisen der Ein- und Ausgabedatei mittels ASSIGN-FILES
- Auswahl des Benutzerausgangs für Assembler-Ausgaberoutinen

Beschreibung der Benutzerroutine

In diesem Beispiel werden durch eine Benutzerroutine Sätze eingefügt, geändert und gelöscht. Die Routine wird für jeden Ausgabesatz angesprungen und soll folgende Maßnahmen durchführen:

- Alle Sätze ausgeben, die in Byte 5 die Ziffer '3' enthalten.
- Nach jedem Satz, der in Byte 5 die Ziffer '1' enthält, einen Satz einfügen.
- Alle Sätze ändern, die in Byte 5 die Ziffer '5' enthalten. Als erstes Zeichen soll 'A' stehen.
- Alle Sätze löschen, die in Byte 5 keine der Ziffern '1', '3' oder '5' enthalten.

Quellprogramm-Datei

```

E35B      START
          PRINT NOGEN
          ENTRY E35B01
          USING E35B01,15
E35B01    SAVE (14,12)          REGISTER SICHERN
          L 7,12(1)            ADRESSE DES AKTIONSWORTES
          CLC 0(4,1),=XL4'00'  LETZTER AUSGABESATZ?
          BE EOF                JA
          L 8,0(1)             ADRESSE DES SATZES NACH REG. 8
          CLI 4(8),C'3'        1. STELLE=3?
          BE VERARB            1. STELLE=5?
          CLI 4(8),C'5'
          BE AENDERN
SCHALTER  NOP EINFUEG
          CLI 4(8),C'1'        1. STELLE=1?
          BNE LOESCH
          OI SCHALTER+1,X'F0'   SCHALTER SETZEN
          B VERARB+4
*         S A T Z           L O E S C H E N
LOESCH    MVI 3(7),X'04'       RUECKSPRUNGKENNZ. IN AKTIONSWORT
          B URSPRUNG
*         S A T Z           E I N F U E G E N
EINFUEG   MVC 0(4,1),=A(EIN)
          MVI SCHALTER+1,X'00'  SCHALTER RUECKSETZEN
          MVI 3(7),X'0C'       RUECKSPRUNGKENNZ. IN AKTIONSWORT
          B URSPRUNG
*         S A T Z           A E N D E R N
AENDERN   MVI 4(8),C'A'        1. STELLE DES SATZES AENDERN
*         S A T Z           U E B E R N E H M E N
VERARB    MVI SCHALTER+1,X'00'  SCHALTER RUECKSETZEN
          MVI 3(7),X'00'       RUECKSPRUNGKENNZ. IN AKTIONSWORT
          B URSPRUNG
*         E N D E
EOF       MVI 3(7),X'08'       KEIN RUECKSPRUNG MEHR
URSPRUNG  RETRN (14,12)        REGISTER WIEDER HERSTELLEN
EIN       DC Y(ENDE=EIN)      EINZUFUEGENDER SATZ
          DS CL2
          DC C'EINGEFUEGTER SATZ'
ENDE     EQU *
          END E35B

```


Erläuterung zum Quellprogramm

Das Quellprogramm wird mit ASSEMBH übersetzt.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen E35B.

Inhalt der Eingabedatei E35A.INPUT

SL	Sortier- feld		
1	5	11	21
		111111	111111111111
		222222	222222222222
		333333	333333333333
		444444	444444444444
		555555	555555555555
		666666	666666666666
		111111	111111111111
		222222	222222222222
		333333	333333333333
		444444	444444444444
		555555	555555555555
		666666	666666666666
		111111	111111111111
		222222	222222222222
		333333	333333333333
		444444	444444444444
		555555	555555555555
		666666	666666666666

Ablaufprotokoll des Sortierlaufs

```
/add-file-link link-name=blslib,file-name=sort.example.lib
/start-sort
% SRT1001 2014-10-12/13:20:33/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=e35a.input,output-file=e35a.output
//assign-exits output=*module(name=e35b01,parameter-mode=*any)
//sort-records fields=*field-explicit(position=5,length=6)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....18 (FROM 01)
% SRT1021 SORTED/MERGED RECORDS.....18
% SRT1022 INSERTED OUTPUT RECORDS:.....3
% SRT1023 DELETED OUTPUT RECORDS:.....9
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....12
% SRT1002 2014-10-12/13:20:47/000000.32 SORT/MERGE COMPLETED
```

Kommentar

In diesem Beispiel bearbeitet eine Assembler-Routine die auszugebenden Sätze. Deshalb muss das Assemblermodul als OUTPUT zugewiesen werden. Der Name für das Modul entspricht dem Modulnamen in der Bibliothek SORT.EXAMPLE.LIB. Mit dem Operanden PARAMETER-MODE wird der Adressierungsmodus und die Schnittstelle zwischen SORT und dem Benutzermodul festgelegt.

In diesem Beispiel ist PARAMETER-MODE=*ANY richtig, da der von der Assembler-Routine verwendete Benutzerausgang OUTPUT sowohl 31- als auch 24-Bit-Adressierung zulässt.

Benutzerausgänge, die ausschließlich die 24-Bit-Adressierung zulassen, verlangen, dass bei PARAMETER-MODE **unbedingt** der Wert 24 eingetragen wird.

10.5.18 Beispiel 18: Benutzerausgang PHYSICAL-TRANSLATE

Vollsortieren von Sätzen mit festem Satzformat

Eingabe: SAM-Datei PHYSICAL.TRANSLATE.INPUT
Ausgabe: SAM-Datei PHYSICAL.TRANSLATE.OUTPUT
Bibliothek: SORT.EXAMPLE.LIB
Modulname: ETBSORT
Benutzerausgang: PHYSICAL-TRANSLATE

In diesem Beispiel wird eine Datei nach einer selbstdefinierten Reihenfolge (Ziffern, Zeichen, Leerstellen) sortiert, die weder der EBCDIC noch der ASCII-Norm entspricht.

Die vorgefundenen Zeichen werden mit der ersten in der Benutzerroutine definierten Code-Tabelle übersetzt und sortiert. Vor der Ausgabe werden die Zeichen mit der zweiten Code-Tabelle wieder rückcodiert.

Quellprogramm der Benutzerroutine

```

ETBSORT  CSECT
          TITLE 'SPEZIALALPHABET'
*
* KODEUMSETZUNG VON EBCDIC IN FOLGENDE
*
* REIHENFOLGE
* '0123456789ABCDEFGHI'
* 'JKLMNOPQRSTUVWXYZ'
* ANALOGES GILT FUER DIE AENDERUNG DER REIHENFOLGE VON GROSS- UND
* KLEINBUCHSTABEN
*
          DC    XL64'00'
BLANK    DC    X'25'                SPACE HINTER BUCHSTABEN
          DC    XL128'00'
ABISI    DC    XL9'0B0C0D0E0F10111213'    BUCHSTABEN A - I
          DC    XL7'00'
JBISR    DC    XL9'1415161718191A1B1C'    BUCHSTABEN J - R
          DC    XL8'00'
SBISZ    DC    XL8'1D1E1F2021222324'    BUCHSTABEN S - Z
          DC    XL6'00'
NULL9    DC    XL10'0102030405060708090A'  ZAHLEN 0 - 9 VOR BUCHST.
          DC    XL6'00'
*
          DC    X'00'
          DC    '0123456789'    ZAHLEN 0 - 9 RUECKKODIEREN.
          DC    'ABCDEFGHI'    BUCHSTABEN A - I RUECKKOD.
          DC    'JKLMNOPQR'    BUCHSTABEN J - R RUECKKOD.
          DC    'STUVWXYZ'    BUCHSTABEN S - Z RUECKKOD.
          DC    X'40'    SPACE RUECKKODIEREN
          DC    XL127'00'
*
          END

```

Erläuterung zum Quellprogramm

Das Quellprogramm wird mit ASSEMBH übersetzt.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen ETBSORT.

Ablaufprotokoll

```
/add-file-link link-name=blslib,file-name=sort.example.lib
/start-sort
% SRT1001 2014-10-12/16:14:06/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=physical-translate.input, -
//              output-file=physical-translate.output
//assign-exits physical-translate=*module(name=etbsort)
//sort-records fields=*field-explicit(position=1,length=4, -
//              format=*physical-translate)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....25 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....25
% SRT1002 2014-10-12/16:14:22/000000.26 SORT/MERGE COMPLETED
```

Kommentar

Bevor SORT gestartet wird, wird die Bibliotheksdatei SORT.EXAMPLE.LIB zugewiesen. Die Anweisung ASSIGN-EXITS ermöglicht, die zuvor übersetzte Routine ETBSORT zu aktivieren.

Inhalt der Eingabe- und Ausgabedateien

Die Eingabedatei PHYSICAL.TRANSLATE.INPUT ist nach EBCDIC sortiert. Die Ausgabedatei PHYSICAL.TRANSLATE.OUTPUT wurde unter Verwendung der vom Benutzer definierten Code-Tabelle sortiert. Die folgende Gegenüberstellung der Dateien soll den Unterschied zwischen diesen beiden Sortierarten verdeutlichen.

PHYSICAL.TRANSLATE.INPUT

Sor- tier- feld	
-----------------------	--

1 5

```

FGH 906C
ABCD E234
ADTX 0914
ASDF 9082
BKLM 0127
BSDF P093
ERSK 7654
ERU 897X
JKDS 7809
KLMD 9808
SFTD OPLD
0012 3POL
1178 LLOP
1234 OPRS
1234 6793
2349 MNB
2356 ADFG
4456 PPL0
4569 87 N
4578 DFGH
7658 9XYZ
77 9 AAPK
7777 RRRS
9875 DGFK
9999 9999

```

PHYSICAL.TRANSLATE.OUTPUT

Sor- tier- feld	
-----------------------	--

1 5

```

0012 3POL
1178 LLOP
1234 OPRS
1234 6793
2349 MNB
2356 ADFG
4456 PPL0
4569 87 N
4578 DFGH
7658 9XYZ
7777 RRRS
77 9 AAPK
9875 DGFK
9999 9999
ABCD E234
ADTX 0914
ASDF 9082
BKLM 0127
BSDF P093
ERSK 7654
ERU 897X
JKDS 7809
KLMD 9808
SFTD OPLD
FGH 906C

```

10.5.19 Beispiel 19: Benutzerausgang VIRTUAL-TRANSLATE

Vollsortieren von Sätzen nach eigener Sortierreihenfolge

Eingabe: SAM-Datei VIRTRAN.INPUT mit variablem Satzformat
Ausgabe: SAM-Datei VIRTRAN.OUTPUT mit variablem Satzformat
Bibliothek: SORT.EXAMPLE.LIB (benutzerdefiniert)
Modulname: VIRTRAN
Benutzerausgang: VIRTUAL-TRANSLATE

Übung zu:

- Definition einer eigenen Codetabelle mit dem Assembler
- Ablegen dieser Tabelle in eine Bibliothek
- Zuweisen der Bibliothek und Verwendung des Moduls
- Vergleich des Ergebnisses

Quellprogramm

```

VIRTRAN  CSECT
VIRTRAN  AMODE ANY
VIRTRAN  RMODE ANY
          TITLE 'EIGENE SORTIERFOLGE'
*KODEUMSETZUNG VON EBCDIC IN NEUE
*        REIHENFOLGE
*
*REIHENFOLGE
*        'WERDEN GLEICHWERTIG BEHANDELT'
*
          DC    XL64'00'
BLANK    DC    X'00'                SPACE
          DC    XL64'00'
*
* KLEINBUCHSTABEN
KLABISI  DC    XL9'0B0C0D0E0F10111213'    KLEINBUCHSTABEN A-I
          DC    XL7'00'
KLJBISR  DC    XL9'1415161718191A1B1C'    KLEINBUCHSTABEN J-R
          DC    XL8'00'
KLSBISZ  DC    XL8'1D1E1F2021222324'    KLEINBUCHSTABEN S-Z
          DC    XL23'00'
*
* GROSSBUCHSTABEN UND ZIFFERN
*
ABISI    DC    XL9'0B0C0D0E0F10111213'    GROSSBUCHSTABEN A-I
          DC    XL7'00'
JBISR    DC    XL9'1415161718191A1B1C'    GROSSBUCHSTABEN J-R
          DC    XL8'00'
SBISZ    DC    XL8'1D1E1F2021222324'    GROSSBUCHSTABEN S-Z
          DC    XL6'00'
NULL9    DC    XL10'0102030405060708090A'  ZIFFERN 0-9
          DC    XL6'00'
*
          END

```

Erläuterung zum Quellprogramm

Das Quellprogramm wird mit ASSEMBH übersetzt.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen VIRTRAN.

Anmerkung zur Codetabelle

Über den Benutzerausgang VIRTUAL-TRANSLATE wird eine Codetabelle definiert (vgl. hierzu auch Benutzerausgang PHYSICAL-TRANSLATE).

Ablaufprotokoll

```
/add-file-link link-name=blslib,file-name=sort.example.lib
/start-sort
% SRT1001 2014-10-12/17:17:55/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=virtran.input,output-file=virtran.output
//assign-exits virtual-translate=*module(name=virtran)
//sort-records fields=*field-explicit(position=5,length=15, -
//          format=*virtual-translate),estimated-records=29
//set-record-attributes output=*variable(maximum-reocrd-size=45)
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....29 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....29
% SRT1002 2014-10-12/17:18:09/000000.37 SORT/MERGE COMPLETED
```

Erläuterung zum Ablauf

- Zuweisen der Bibliothek mit dem Modul VIRTRAN
- Zuweisen der Ein- und Ausgabedateien
- Zuweisen des Moduls VIRTRAN mit der ASSIGN-EXITS-Anweisung
- Verweis in der SORT-RECORDS-Anweisung auf die Codetabelle
- Ankündigung der Anzahl der Sätze
- Ankündigung des längsten Eintrags (Länge 45)

SORT legt die Ausgabedatei mit der durch die SET-RECORD-ATTRIBUTES-Anweisung festgelegten Länge an und trägt diesen Wert im Katalog ein.

```

/show-file-attributes file-name=virtran.output, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.VIRTRAN.OUTPUT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE) = READ-WRITE  IO(PERF) = STD            DISK-WRITE = IMMEDIATE
% REC-FORM = (V,N)        REC-SIZE = 45
% AVAIL = *STD
% WORK-FILE = *NO          F-PREFORM = *K          SO-MIGR = *ALLOWED

```

VIRTRAN.INPUT	VIRTRAN.OUTPUT
scherzen	Abendland
allgemein	allgemein
mexikanisch	Allgemeinheit
Mehrzahl	Astronomie
oder	astronomisch
Trennung	Biologie
Verhaeltniswort	brasilianisch
Stilkunde	Brasilien
brasilianisch	bretonisch
philosophisch	Deutschland
medizinisch	dichterisch
dichterisch	Druckersprache
Druckersprache	englisch
Astronomie	franzoesisch
mechanisch	Franzosen
scherzhaft	Landwirtschaft
bretonisch	Mechanik
Biologie	mechanisch
Brasilien	medizinisch
Allgemeinheit	Mehrzahl
astronomisch	mexikanisch
Mechanik	oder
Franzosen	philosophisch
englisch	Scherz
Landwirtschaft	scherzen
franzoesisch	scherzhaft
Abendland	Stilkunde
Scherz	Trennung
Deutschland	Verhaeltniswort

10.5.20 Beispiel 20: SORT als Unterprogramm (Ebene 0)

Eingabe: SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat.

Ausgabe: SAM-Datei SORT.UPRG.SORT mit festem Satzformat und über SYSOUT an der Datenstation

Satzstruktur der Eingabedatei RESTAURANT.SAM.FIX

Restaurantname	Straße	Telefon	Art d. Küche
1	21	48	56 66

Der Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“ auf Seite 389](#) dargestellt.

Das Hauptprogramm liest die Sätze aus der Datei RESTAURANT.SAM.FIX ein und gibt sie in neu aufbereiteter Form in die Datei SORT.UPRO aus. Die Sätze von SORT.UPRO sollen sortiert werden (Vollsortieren) und gleichzeitig in die Datei SORT.UPRG.SORT und an der Datenstation ausgegeben werden.

Satzstruktur der Zwischendatei SORT.UPRO

Art der Küche	Restaurantname	Straße	Telefon
1	15	38	65 72

Die Steuerinformation an SORT wird über Ebene 0 übergeben (SORT-Steueranweisungen von SYSDTA).

Quellprogramm-Datei

```

SUPROGO  START
          TITLE 'DIESES PROGRAMM RUFT SORT ALS UPRO UEBER EBENE 0 AUF'
          PRINT NOGEN
SUPROGO  AMODE ANY
SUPROGO  RMODE ANY
          GPARMOD 31
ANF      BALR 3,0
          USING *,3
          OPEN  EIN,INPUT           EROEFFNEN DER EINGABEDATEI
          OPEN  AUS,OUTPUT          EROEFFNEN DER AUSGABEDATEI
          MVI  AUSB,X'40'
          MVC  AUSB+1(71),AUSB
LIES1    EQU  *
          GET  EIN,EINB             EINLESEN EINES SATZES
          MVC  AF1,EF4              NEUAUFBEREITEN DES
          MVC  AF2,EF1              EINGABESATZES
          MVC  AF3,EF2
          MVC  AF4,EF3
          PUT  AUS,AUSB            AUSGABE EINES SATZES
          B    LIES1

*
*
          FCB  DER EINGABEDATEI
EIN      FCB  FCBTYP=SAM,          -
          LINK=EIN,                -
          RECFORM=F,                -
          RECSIZE=66,               -
          EXIT=EXITEIN
EXITEIN  EXLST EOFADDR=ENDE1      LETZTER EINGABESATZ
*
          FCB  DER AUSGABEDATEI
AUS      FCB  FCBTYP=SAM,          -
          LINK=AUS,                -
          RECFORM=F,                -
          RECSIZE=72,               -
          EXIT=EXITAUS
EXITAUS  EXLST EOFADDR=ENDE9
          DS  OF
ENDE1    EQU  *
          CLOSE ALL
          FILE SORT.UPRO,LINK=SORTIN  EINGABEDATEI (SORTIERLAUF)
          FILE SORT.UPRG.SORT,LINK=SORTOUT  AUSGABEDATEI FUER
*
          SORTIERLAUF
          LA 1,B1                   EBENE 0
          LA 13,SAVE                 SAVE-BEREICHADR NACH REG13
          L 15,=V(SORTU)             SORT-EINSPRUNGSPUNKT
          BALR 14,15

```

```

ENDE2    EQU    *                DIE AUSGABEDATEI ERHAELT DIE
        FILE  SORT.UPRG.SORT,LINK=AUS MERKMALE DER DATEI MIT DER
        OPEN  AUS,INPUT          ADRESSE AUS.
LIES2    EQU    *
        GET   AUS,AUSB
        WROUT OUT,FEHLER
        B     LIES2
FEHLER   CLOSE ALL
        TERM  MODE=ABNORMAL,UNIT=STEP
ENDE9    CLOSE ALL
        TERM
*
B1       SRT0 STXIT=NO,MSGPROT=BOTH,SDF=YES  VERSORGUNG UEBER EBENE 0
*
SAVE     DS    18F
EINB     DS    0CL72
EF1      DS    CL20
EF2      DS    CL27
EF3      DS    CL08
EF4      DS    CL11
OUT      DC    H'77'
        DC    C' '
        DC    X'01'
AUSB     DS    0CL72
AF1      DS    CL11
        DS    CL03
AF2      DS    CL20
        DS    CL03
AF3      DS    CL24
        DS    CL03
AF4      DS    CL08
        DS    0F
        END   SUPROGO

```

Erläuterung zum Quellprogramm

Der Makroaufruf SRT0 dient zur Versorgung über Ebene 0.

Das Quellprogramm wird mit ASSEMBH übersetzt. Dazu muss die Bibliothek \$.SYSLIB.SORT.080, in der die SORT-Makros stehen, als Makrobibliothek zugewiesen werden.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen SUPROG0.

Ablaufprotokoll

```

/add-file-link link-name=ein,file-name=restaurant.sam.fix
/create-file file-name=sort.upro
/add-file-link link-name=aus,file-name=sort.upro
/add-file-link link-name=b1slib00,file-name=$.syslnk.sort.080
/start-executable-program library=sort.example.lib, -
/                               element-or-symbol=suprog0, -
/                               alternate-library=*b1slib##
% SRT1001 B1 2014-10-12/13:57:52/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 B1 PLEASE ENTER SORT STATEMENTS
//sort-records
//end
% SRT1016 B1 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1030 B1 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 B1 2014-10-12/13:58:08/000000.26 SORT/MERGE COMPLETED
French      Le Gourmet          Lime Street 46      505397
German      Willi's Bar         Westland Street 113 748293
Indonesian  Java                Hope Street 51      522221
Italian     Orlando's           Thompson Street 62   220061
Japanese    Chayota's           Thurston Street 60   292742
Mexican     Palenque Mexico     Millwood Drive 2    980149
Persian     Persepolis          Salford Square 20   597004
Vegetarian  Strawberry           Sauchiehall Street 8 595521
Vietnamese  Vietnam             Thurston Street 47   522518
Yugoslavian Golden Fleece        Arran Street 44     242437

```

Kommentar zum Ablaufprotokoll

- Zuweisen der Datei RESTAURANT.SAM.FIX als Eingabedatei für das Hauptprogramm SUPROG0.
- Zuweisen der Datei SORT.UPRO als Ausgabedatei für das Hauptprogramm SUPROG0. In diese Datei gibt das Hauptprogramm die aufbereiteten Sätze aus.
- Zuweisen der Modulbibliothek \$.SYSLNK.SORT.080, in der die SORT-Module stehen.
- Aufruf des Hauptprogramms SUPROG0
- Das Hauptprogramm ruft SORT auf, und SORT fordert zur Eingabe der Anweisungen auf.
- Die SORT-RECORDS-Anweisung fordert eine Vollsartierung an.
- Die END-Anweisung schließt die Definition des Sortierlaufs ab und startet den Sortierlauf.

10.5.21 Beispiel 21: SORT als Unterprogramm (Ebene 1)

Eingabe: SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat

Ausgabe: SAM-Datei SORT.UPRG.SORT mit festem Satzformat

Satzstruktur der Eingabedatei RESTAURANT.SAM.FIX

Restaurantname	Straße	Telefon	Art d. Küche
1	21	48	56 66

Der Inhalt dieser Datei ist im [Abschnitt „Verzeichnis der Beispieldateien“ auf Seite 389](#) dargestellt.

Das Hauptprogramm liest die Sätze aus der Datei RESTAURANT.SAM.FIX ein und gibt sie in neu aufbereiteter Form in die Datei SORT.UPRO aus.

Satzstruktur der Datei SORT.UPRO

Art der Küche	Restaurantname	Straße	Telefon
1	15	38	65 72

Die Sätze von SORT.UPRO sollen nach dem Sortierfeld „Art der Küche“ sortiert werden (Vollsortieren) und in die Datei SORT.UPRG.SORT und an der Datenstation ausgegeben werden. Dabei sollen nur Sätze in die Sortierung übernommen werden, deren Sortierfeld „Art der Küche“ mit den Buchstaben 'I' oder 'J' beginnt.

Die Steuerinformation an SORT wird über Ebene 1 übergeben (SORT-Steueranweisungen im Hauptspeicher).

Quellprogramm-Datei

```

SUPROG1  START
          TITLE 'DIESES PROGRAMM RUFT SORT ALS UPRO UEBER EBENE 1 AUF'
          PRINT NOGEN
SUPROG1  AMODE ANY
SUPROG1  RMODE ANY
          GPARMOD 31
ANF      BALR 3,0
          USING *,3
          OPEN  EIN,INPUT           EROEFFNEN DER EINGABEDATEI
          OPEN  AUS,OUTPUT          EROEFFNEN DER AUSGABEDATEI
          MVI  AUSB,X'40'
          MVC  AUSB+1(71),AUSB
LIES1    EQU  *
          GET  EIN,EINB             EINLESEN EINES SATZES
          MVC  AF1,EF4              NEUAUFBEREITEN DES
          MVC  AF2,EF1              EINGABESATZES
          MVC  AF3,EF2
          MVC  AF4,EF3
          PUT  AUS,AUSB             AUSGABE EINES SATZES
          B    LIES1
*
          *                          FCB DER EINGABEDATEI
EIN      FCB  FCBTYPE=SAM,          -
          LINK=EIN,                 -
          RECFORM=F,                -
          RECSIZE=66,               -
          EXIT=EXITEIN
EXITEIN  EXLST EOFADDR=ENDE1       LETZTER EINGABESATZ
*
AUS      FCB  FCBTYPE=SAM,          -
          LINK=AUS,                 -
          RECFORM=F,                -
          RECSIZE=72,               -
          EXIT=EXITAUS
EXITAUS  EXLST EOFADDR=ENDE9
          DS  OF                     AUSRICHTEN AUF WORTGRENZE
ENDE1    EQU  *
          CLOSE ALL
          FILE SORT.UPRO,LINK=SORTIN EINGABEDATEI (SORTIERLAUF)
          FILE SORT.UPRG.SORT,LINK=SORTOUT AUSGABEDATEI FUER
*
          *                          SORTIERLAUF
          LA 1,B1                     EBENE 1
          LA 13,SAVE                   SAVE-BEREICHADR NACH REG13
          L 15,=V(SORTU)                SORT-EINSPRUNGSPUNKT
          BALR 14,15
ENDE2    EQU  *
          FILE SORT.UPRG.SORT,LINK=AUS DIE AUSGABEDATEI ERHAELT DIE

```



```

*          OPEN  AUS,INPUT          MERKMALE DER DATEI MIT DER
LIES2     EQU   *                   ADRESSE AUS.
          GET   AUS,AUSB
          WROUT OUT,FEHLER
          B     LIES2
FEHLER    CLOSE ALL
          TERM  MODE=ABNORMAL,UNIT=STEP
ENDE9     CLOSE ALL
          TERM
          PRINT GEN
B1        SRT1  (SORT-RECORDS  FIELDS=*FIELD-EXPLICIT(POSITION=1,LENGTH=1-
          1)),SDF=YES
          SRT1  (SELECT-INPUT-RECORDS  CONDITION=((1,1,CH)='I')OR((1,1,C-
          H)='J'))
          SRT1  (SET-SORT-OPTIONS  MIN-MSG-WEIGHT=*ALL)
          SRT1  (END)
SAVE      DS    18F
EINB      DS    0CL66
EF1       DS    CL20
EF2       DS    CL27
EF3       DS    CL08
EF4       DS    CL11
OUT       DC    H'77'
          DC    C' '
          DC    X'01'
AUSB      DS    0CL72
AF1       DS    CL11
          DS    CL03
AF2       DS    CL20
          DS    CL03
AF3       DS    CL24
          DS    CL03
AF4       DS    CL08
          DS    0F
          END   SUPROG1

```

Erläuterung zum Quellprogramm

Mit dem Makro SRT1 werden folgende Steueranweisungen an SORT übergeben (Versorgung über Ebene1):

- SORT-RECORDS zur Anforderung einer Vollsortierung
- SELECT-INPUT-RECORDS, um alle Eingabesätze auszuschließen, die nicht mit 'I' oder 'J' beginnen
- SET-SORT-OPTIONS zur Steuerung der Meldungsausgabe
- END-Anweisung als Abschluss der Anweisungsfolge

Das Quellprogramm wird mit ASSEMBH übersetzt. Dazu muss die Bibliothek \$.SYSLIB.SORT.080, in der die SORT-Makros stehen, als Makrobibliothek zugewiesen werden.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen SUPROG1.

Syntax des SRT1-Makros

Der SDF-Anweisungstext des SRT1-Makros muss in runde Klammern eingeschlossen sein.

Im ersten SRT1-Makro muss am Ende, außerhalb der den Makro umschließenden Klammer, der Zusatz SDF=YES enthalten sein.

Alle SDF-Anweisungen des Makros müssen konsequent bis zum Fortsetzungszeichen in Spalte 72 ohne füllende Leerzeichen (Blank) fortgeschrieben werden. Auf diese Weise können Trennungen entstehen, die nicht grammatikalischen Regeln entsprechen, wie z.B. bei der in diesem Beispiel verwendeten SELECT-INPUT-RECORDS-Anweisung.

Ablaufprotokoll

```

/add-file-link link-name=ein,file-name=restaurant.sam.fix
/create-file file-name=sort.upro
/add-file-link link-name=aus,file-name=sort.upro
/add-file-link link-name=blslib00,file-name=$.syslnk.sort.080
/start-executable-program library=sort.example.lib, -
/                               element-or-symbol=suprog1, -
/                               alternate-libraries=*blslib##
% SRT1001 B1 2014-10-12/12:56:17/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1046 B1 2014-10-12/12:56:19/000000.26 END OF PREPARATORY PHASE
% SRT1013 B1 MAIN MEMORY SORT
% SRT1016 B1 SORT/MERGE INPUT RECORDS:.....10 (FROM 01)
% SRT1024 B1 DELETED SELECT-INPUT-RECORDS RECORDS:.....7
% SRT1017 B1 RECORDS TO BE SORTED/MERGED:.....3
% SRT1030 B1 SORT/MERGE OUTPUT RECORDS:.....3
% SRT1002 B1 2014-10-12/12:56:20/000000.28 SORT/MERGE COMPLETED
Indonesian   Java                Hope Street 51           522221
Italian      Orlando's             Thompson Street 62       220061
Japanese     Chayota's             Thurston Street 60       292742

```

Kommentar zum Ablaufprotokoll

- Zuweisen der Datei RESTAURANT.SAM.FIX als Eingabedatei für das Hauptprogramm SUPROG1.
- Zuweisen der Datei SORT.UPRO als Ausgabedatei für das Hauptprogramm SUPROG1. In diese Datei werden die vom Hauptprogramm aufbereiteten Sätze ausgegeben.
- Zuweisen der Modulbibliothek \$.SYSLNK.SORT.080, in der die SORT-Module stehen.
- Aufruf des Hauptprogramms SUPROG1.
- Das Hauptprogramm ruft SORT auf und übergibt SORT die SORT-Steueranweisungen direkt.

Dateimerkmale der Eingabedatei RESTAURANT.SAM.FIX

```

/show-file-attributes file-name=restaurant.sam.fix, -
/
      information=*parameters(organization=*yes)
%0000000003:CTID:$EXAMPLE.RESTAURANT.SAM.FIX
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)       REC-SIZE   = 66
% AVAIL       = *STD
% WORK-FILE   = *NO          F-PREFORM = *K              SO-MIGR    = *ALLOWED

```

Dateimerkmale der Ausgabedatei SORT.UPRG.SORT

```

/show-file-attributes file-name=sort.uprg.sort, -
/
      information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.SORT.UPRG.SORT
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)       REC-SIZE   = 72
% AVAIL       = *STD
% WORK-FILE   = *NO          F-PREFORM = *K              SO-MIGR    = *ALLOWED

```

10.5.22 Beispiel 22: SORT-Zugriffsmethode

Eingabe: SAM-Datei RESTAURANT.SAM mit variablem Satzformat
 Ausgabe: SAM-Datei RESTAURANT.SAM.AUS mit variablem Satzformat
 Quelltext: SORTZM01

Übung zu:

- Zuweisen einer Eingabedatei und einer Ausgabedatei mit den Kettungsnamen EIN und AUS statt SORTIN und SORTOUT
- Verwendung der SORTZM-Makros

In diesem Beispiel wird das Sortieren mit der SORT-Zugriffsmethode SORTZM gezeigt.

Die Eingabesätze stehen in einer SAM-Datei RESTAURANT.SAM mit variablem Satzformat zur Verfügung, die mit dem Dateikettungsnamen EIN zugewiesen wird.

Die Sätze der Eingabedatei RESTAURANT.SAM sind wie folgt strukturiert:

SL	Restaurantname	Straße	Telefon	Art der Küche
1	5	25	52	60
				76

Die Ausgabesätze sollen von SORT in die Ausgabedatei RESTAURANT.SAM.AUS mit variablem Satzformat übergeben werden, die mit dem Dateikettungsnamen AUS zugewiesen wird. Folgende Felder werden ausgewählt:

SL	Art der Küche	Restaurantname	Straße
1	5	22	42
			68

Die Sätze von RESTAURANT.SAM.AUS sollen nach dem Sortierfeld „Art der Küche“ sortiert werden. Die übrigen Felder sind Restfelder.

Quellprogramm-Datei

```

SORTZM01 START
          PRINT NOGEN
SORTZM01 AMODE ANY
SORTZM01 RMODE ANY
          GPARMOD 31
          BALR 10,0
          USING ANFANG,10,11
ANFANG   L    11,BASADR
          B    BEGINN
BASADR   DC   A(ANFANG+4096)
*
BEGINN   EQU   *
          OPEN EIN,INPUT           EROEFFNEN EINGABEDATEI
          OPEN AUS,OUTPUT          EROEFFNEN AUSGABEDATEI
*
*          *****
          SRTOPEN SCB=B1,ERROR=FEHLER  SORTIERLAUF EROEFFNEN
          *****
*
LIES     GET   EIN,EINBER           SATZ EINLESEN
*
*          *****
          SRTPUT SCB=B1,RECORD=EINBER  SATZ AN SORTIERLAUF UEBERGEHEN
          *****
*
          B    LIES
ENDEIN   LA   4,AUSBER
*
*          *****
SCHREIB  SRTGET SCB=B1,RECORD=(4),EOS=CLOSE  SATZ UEBERNEHMEN
          *****
*
          PUT  AUS,AUSBER           SATZ AUSGEBEN
          B    SCHREIB
*
*          *****
CLOSE    SRTCLSE SCB=B1             SORTIERLAUF SCHLIESSEN
          *****
*
CLALL    CLOSE ALL                   E-/A-DATEIEN SCHLIESSEN
          TERM
*
FEHLER   CLOSE ALL                   FEHLERAUSGANG BEI
          TERM  MODE=ABNORMAL,UNIT=STEP  ABNORMALER BEENDIGUNG
*
EIN      FCB   FCBTYP=SAM,LINK=EIN,RECFORM=V,EXIT=EXITEIN

```

```
AUS      FCB      FCBTYPE=SAM, LINK=AUS, RECFORM=V, EXIT=EXITAUS
*
EXITEIN  EXLST    EOFADDR=ENDEIN, COMMON=CLALL
EXITAUS  EXLST    COMMON=CLALL
*
EINBER   DS      0CL76
SL1      DS      CL4
DATEN1   DS      CL72
*
AUSBER   DS      0CL68
SL2      DS      CL4
DATEN2   DS      CL64
*
B1       SRT1    (SORT-RECORDS FIELDS=(*FIELD-EXPLICIT(POSITION=60, LENGTH-
=17), REMAINDER-EXPLICIT(POSITION=5, LENGTH=20), *REMAINDER-
-EXPLICIT(POSITION=25, LENGTH=27)), SORT-TYPE=*COMPOUND-RE-
CORD), SDF=YES
          SRT1    (SET-RECORD-ATTRIBUTES OUTPUT=*VARIABLE)
          SRT1    (SET-SORT-OPTIONS MIN-MSG-WEIGHT=*ALL)
          SRT1    (END)
          END     SORTZM01
```

Erläuterung zum Quellprogramm

Im Quellprogramm werden folgende Aktionen durchgeführt:

- Eröffnen der Eingabedatei EIN.
- Eröffnen der Ausgabedatei AUS.
- Mit dem Makroaufruf SRTOPEN wird der SORT-Lauf eröffnet.
- Eingabesatz aus der Eingabedatei EIN lesen.
- Mit dem Makroaufruf SRTPUT wird ein Eingabesatz an den SORT-Lauf übergeben.
- Der nächste Eingabesatz wird gelesen. Die Leseschleife wird so oft durchlaufen, bis der letzte Eingabesatz gelesen ist.
- Ende der Eingabe (EOFADDR im EXLST-Makroaufruf).
- Mit dem Makroaufruf SRTGET werden die sortierten Sätze des Sortierlaufs übernommen. Sind alle Sätze übernommen, wird zum Schließen des Sortierlaufs verzweigt (Operand EOS).
- Ausgabe der Sätze des Sortierlaufs in die Ausgabedatei AUS.
- Nach der Ausgabe der Sätze wird der Sortierlauf mit dem Makroaufruf SRTCLSE geschlossen.
- Schließen der Ein- und Ausgabedatei.
- Makroaufrufe SRT1 für den Sortierlauf (Auswahl-sortieren).

Hinweis zur Syntax des SRT1-Makros

Der SDF-Anweisungstext des SRT1-Makros muss in runde Klammern eingeschlossen sein.

Im ersten SRT1-Makro muss am Ende, außerhalb der den Makro abschließenden Klammer der Zusatz SDF=YES enthalten sein.

Alle SDF-Anweisungen des Makros müssen konsequent bis zum Fortsetzungszeichen in Spalte 72 ohne füllende Leerzeichen (Blank) fortgeschrieben werden. Auf diese Weise können Trennungen entstehen, die nicht grammatikalischen Regeln entsprechen.

Das Quellprogramm wird mit ASSEMBH übersetzt. Dazu muss die Bibliothek \$.SYSLIB.SORT.080, in der die SORT-Makros stehen, als Makrobibliothek zugewiesen werden.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen SORTZM01.

Ablaufprotokoll

```

/add-file-link link-name=blslib00,file-name=$.syslnk.sort.080
/add-file-link link-name=ein,file-name=restaurant.sam
/create-file file-name=sort.upro
/add-file-link link-name=aus,file-name=sort.upro
/start-executable-program library=sort.example.lib, -
/                               element-or-symbol=sortzm01, -
/                               alternate-library=*blslib##
% SRT1001 B1 2014-10-12/12:56:17/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1046 B1 12:56:19/000000.26 END OF PREPARATORY PHASE
% SRT1013 B1 MAIN MEMORY SORT
% SRT1021 B1 SORTED/MERGED RECORDS:.....10
% SRT1030 B1 SORT/MERGE OUTPUT RECORDS:.....10
% SRT1002 B1 12:56:20/000000.28 SORT/MERGE COMPLETED

```

Erläuterung zum Ablauf:

- Zuweisen der Modulbibliothek \$.SYSLNK.SORT.080, in der die SORT-Module stehen.
- Zuweisen der Datei RESTAURANT.SAM als Eingabedatei für das Hauptprogramm SORTZM01.
- Zuweisen der Datei SORT.RESTAURANT.AUS als Ausgabedatei für den Sortierlauf.
- Aufruf des Programms SORTZM01, das SORT als Unterprogramm aufruft.
- SORT meldet die Anzahl der sortierten Sätze und die normale Beendigung des Sortierlaufs.

Inhalt der Ausgabedatei RESTAURANT.SAM.AUS

French	Le Gourmet	Lime Street 46
German	Willi's Bar	Westland Street 113
Indonesian	Java	Hope Street 51
Italian	Orlando's	Thompson Street 62
Japanese	Chayota's	Thurston Street 60
Mexican	Palenque Mexico	Millwood Drive 2
Persian	Persepolis	Salford Square 20
Vegetarian	Strawberry	Sauchiehall Street 8
Vietnamese	Vietnam	Thurston Street 47
Yugoslavian	Golden Fleece	Arran Street 44

10.5.23 Beispiel 23: SORT-Zugriffsmethode (Mehrfachsordieren)

Eingabe: SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat

Ausgabe: SAM-Dateien SORT.AUS1, SORT.AUS2 und SORT.AUS3 mit festem Satzformat

Quelltext: SORTZM02

Bibliotheken: \$.SYSLIB.SORT.080 Makrobibliothek für die Übersetzung
\$.SYSLNK.SORT.0080 als Bibliothek zum Nachladen der SORT-Module

Übung zu:

- Zuweisen einer Eingabedatei und mehreren Ausgabedateien mit den Kettungsnamen EIN und AUS1, AUS2 und AUS3 statt SORTIN und SORTOUT.
- Definition verschiedener Sortierkriterien für die drei Ausgabedateien
- Aufruf von SORT als Unterprogramm

In diesem Beispiel wird das *Mehrfachsordieren* mit der SORT-Zugriffsmethode SORTZM gezeigt.

Die Eingabesätze stehen in einer SAM-Datei RESTAURANT.SAM.FIX mit festem Satzformat zur Verfügung, die mit dem Dateikettungsnamen EIN zugewiesen wird.

Die Sätze der Eingabedatei RESTAURANT.SAM.FIX sind wie folgt strukturiert:

Restaurantname	Straße	Telefon	Art der Küche
1	21	48	56 66

Die Eingabesätze werden einzeln eingelesen und an drei parallel ablaufende SORT-Läufe übergeben.

Erster SORT-Lauf (Vollsortieren)

Ausgabedatei SORT.AUS1 mit festem Satzformat, die mit dem Dateikettungsnamen AUS1 zugewiesen wird.

Restaurantname	Straße	Telefon	Art der Küche
1	21	48	56 66

Die Sätze von SORT.AUS1 sollen nach dem Sortierfeld „Restaurantname“ sortiert werden.

Zweiter SORT-Lauf (Auswahlsortieren)

Ausgabedatei SORT.AUS2 mit festem Satzformat, die mit dem Dateikettungsnamen AUS2 zugewiesen wird. Folgende Felder werden ausgewählt:

Art der Küche	Restaurantname	Straße
1	12	32
		58

Die Sätze von SORT.AUS2 sollen nach dem Sortierfeld „Art der Küche“ sortiert werden. Die übrigen Felder sind Restfelder.

Dritter SORT-Lauf (Auswahlsortieren)

Ausgabedatei SORT.AUS3 mit festem Satzformat, die mit dem Dateikettungsnamen AUS3 zugewiesen wird. Folgende Felder werden ausgewählt:

Restaurantname	Art der Küche	Straße
1	21	32
		58

Die Sätze von SORT.AUS3 sollen nach dem Sortierfeld „Straße“ sortiert werden. Die übrigen Felder sind Restfelder.

Quellprogramm-Datei

```

SORTZM02 START
        PRINT NOGEN
SORTZM02 AMODE ANY
SORTZM02 RMODE ANY
        GPARMOD 31
        BALR 10,0
        USING ANFANG,10,11
ANFANG  L    11,BASADR
        B    BEGINN
BASADR  DC   A(ANFANG+4096)
*
BEGINN  EQU   *
        OPEN  EIN,INPUT           EROEFFNEN EINGABEDATEI
        OPEN  AUS1,OUTPUT         EROEFFNEN AUSGABEDATEI1
        OPEN  AUS2,OUTPUT         EROEFFNEN AUSGABEDATEI2
        OPEN  AUS3,OUTPUT         EROEFFNEN AUSGABEDATEI3
*
*
* *****
        SRTOPEN SCB=S01,ERROR=FEHLER SORTIERLAEUFE EROEFFNEN
        SRTOPEN S02,FEHLER
        SRTOPEN S03,FEHLER
*
* *****
*

```

```

LIES      GET   EIN,EINBER           SATZ EINLESEN
*
*
*          *****
          SRTPUT SCB=S01,RECORD=EINBER   SATZ AN
          SRTPUT S02,EINBER             SORTIERLAEUFE
          SRTPUT S03,EINBER             UEBERGEBEN
*          *****
*
          B      LIES
*
ENDEIN    LA    4,AUSBER
*
*          *****
L1        SRTGET SCB=S01,RECORD=(4),EOS=L2
*          *****
*
          PUT   AUS1,AUSBER           AUSGABE DES 1. SORTLAUFS
          B     L1
*
*          *****
L2        SRTGET S02,(4),L3
*          *****
*
          PUT   AUS2,AUSBER           AUSGABE DES 2. SORTLAUFS
          B     L2
*
*          *****
L3        SRTGET S03,,CLOSE
*          *****
*
          LR    0,1
          PUT   AUS3,(0)             AUSGABE DES 3. SORTIERLAUFS
          B     L3
*
*          *****
CLOSE     SRTCLSE SCB=S01           SORTIERLAEUFE
          SRTCLSE S02               SCHLIESSEN
          SRTCLSE S03
*          *****
*
CLALL     CLOSE ALL                 E/A-DATEIEN SCHLIESSEN
          TERM
*
FEHLER    CLOSE ALL
          TERM  MODE=ABNORMAL,UNIT=STEP
*
EIN       FCB   FCBTYP=SAM,LINK=EIN,RECFORM=F,RECSIZE=66,EXIT=EXITEIN
AUS1     FCB   FCBTYP=SAM,LINK=AUS1,RECFORM=F,RECSIZE=66,EXIT=EXITAUS
    
```

```

AUS2    FCB    FCBTYPE=SAM, LINK=AUS2, RECFORM=F, RECSIZE=58, EXIT=EXITAUS
AUS3    FCB    FCBTYPE=SAM, LINK=AUS3, RECFORM=F, RECSIZE=58, EXIT=EXITAUS
*
EXITEIN EXLST  EOFADDR=ENDEIN, COMMON=CLALL
EXITAUS EXLST  COMMON=CLALL
*
EINBER  DS    CL72
AUSBER  DS    CL72
*
S01     SRT1  (SORT-RECORDS FIELDS=*FIELD-EXPLICIT(POSITION=1, LENGTH=2-
              0)), SDF=YES
          SRT1  (SET-RECORD-ATTRIBUTES OUTPUT=*FIXED(RECORD-SIZE=66))
          SRT1  (SET-SORT-OPTIONS MIN-MSG-WEIGHT=*ALL, LINK-PREFIX-CHANGE-
              =S01)
          SRT1  (END)
*
S02     SRT1  (SORT-RECORDS FIELDS=(FIELD-EXPLICIT(POSITION=56, LENGTH=-
              11), REMAINDER-EXPLICIT(POSITION=1, LENGTH=20), REMAINDER-E-
              XPLICIT(POSITION=21, LENGTH=27)), SORT-TYPE=*COMPOUND-RECO-
              RD), SDF=YES
          SRT1  (SET-RECORD-ATTRIBUTES OUTPUT=*FIXED(RECORD-SIZE=58))
          SRT1  (SET-SORT-OPTIONS MIN-MSG-WEIGHT=*ALL, LINK-PREFIX-CHANGE-
              =S02)
          SRT1  (END)
*
S03     SRT1  (SORT-RECORDS FIELDS>(*REMAINDER-EXPLICIT(POSITION=1, LEN-
              GTH=20), *REMAINDER-EXPLICIT(POSITION=56, LENGTH=11), *FIEL-
              D-EXPLICIT(POSITION=21, LENGTH=27)), SORT-TYPE=*COMPOUND-R-
              ECORD), SDF=YES
          SRT1  (SET-RECORD-ATTRIBUTES OUTPUT=*FIXED(RECORD-SIZE=58))
          SRT1  (SET-SORT-OPTIONS MIN-MSG-WEIGHT=*ALL, LINK-PREFIX-CHANGE-
              =S03)
          SRT1  (END)
*
          END  SORTZM02

```

Erläuterung zum Quellprogramm

Für jede der drei Ausgabedateien werden mit dem Makro SRT1 eigene Sortierbedingungen festgelegt.

Hinweis zur Syntax des SRT1-Makros

Der SDF-Anweisungstext des SRT1-Makros muss in runde Klammern eingeschlossen sein.

Im ersten SRT1-Makro muss am Ende, außerhalb der den Makro abschließenden Klammer der Zusatz SDF=YES enthalten sein.

Alle SDF-Anweisungen des Makros müssen konsequent bis zum Fortsetzungszeichen in Spalte 72 ohne füllende Leerzeichen (Blank) fortgeschrieben werden. Auf diese Weise können Trennungen entstehen, die nicht grammatikalischen Regeln entsprechen.

Im Quellprogramm werden folgende Aktionen durchgeführt:

- Eröffnen der Eingabedatei EIN.
- Eröffnen der Ausgabedateien AUS1, AUS2 und AUS3.
- Mit dem Makroaufruf SRTOPEN werden alle drei SORT-Läufe eröffnet.
- Eingabesatz aus der Eingabedatei EIN lesen.
- Mit dem Makroaufruf SRTPUT wird ein Eingabesatz an alle drei SORT-Läufe übergeben.
- Der nächste Eingabesatz wird gelesen. Die Leseschleife wird so oft durchlaufen, bis der letzte Eingabesatz gelesen ist.
- Ende der Eingabe (EOFADDR im EXLST-Makroaufruf).
- Mit dem Makroaufruf SRTGET werden die sortierten Sätze des ersten Sortierlaufs übernommen. Sind alle Sätze übernommen, wird zum zweiten Sortierlauf verzweigt (EOS-Operand).
- Ausgabe der Sätze des ersten Sortierlaufs in die Ausgabedatei AUS1.
- Mit dem Makroaufruf SRTGET werden die sortierten Sätze des zweiten Sortierlaufs übernommen. Sind alle Sätze übernommen, wird zum dritten Sortierlauf verzweigt (EOS-Operand).
- Ausgabe der Sätze des zweiten Sortierlaufs in die Ausgabedatei AUS2.
- Mit dem Makroaufruf werden die sortierten Sätze des dritten Sortierlaufs übernommen.
- Ausgabe der Sätze des dritten Sortierlaufs in die Ausgabedatei AUS3.
- Nach der Ausgabe der Sätze des dritten Sortierlaufs werden alle drei Sortierläufe mit dem Makroaufruf SRTCLSE geschlossen.

- Schließen der Ein-/ und Ausgabedateien.
- S01: Makroaufrufe SRT1 für den ersten Sortierlauf (Vollsortieren).
- S02: Makroaufrufe SRT1 für den zweiten Sortierlauf (Auswahlsortieren).
- S03: Makroaufrufe SRT1 für den dritten Sortierlauf (Auswahlsortieren).

Das Quellprogramm wird mit ASSEMBH übersetzt. Dazu muss die Bibliothek \$.SYSLIB.SORT.080, in der die SORT-Makros stehen, als Makrobibliothek zugewiesen werden.

Das übersetzte Programm wird als LLM in der Bibliothek SORT.EXAMPLE.LIB abgelegt und erhält den Namen SORTZM02.

Ablaufprotokoll

```

/add-file-link link-name=b1slib00,file-name=$.syslnk.sort.080
/add-file-link link-name=ein,file-name=restaurant.sam.fix
/create-file file-name=sort.aus1
/add-file-link link-name=aus1,file-name=sort.aus1
/create-file file-name=sort.aus2
/add-file-link link-name=aus2,file-name=sort.aus2
/create-file file-name=sort.aus3
/add-file-link link-name=aus3,file-name=sort.aus3
/start-executable-program library=sort.example.lib, -
/                               element-or-symbol=sortzm02, -
/                               alternate-library=*b1slib##
% SRT1001 S01 2014-10-12/15:39:23/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1046 S01 2014-10-12/15:39:24/000000.25 END OF PREPARATORY PHASE
% SRT1001 S02 2014-10-12/15:39:24/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1046 S02 2014-10-12/15:39:24/000000.05 END OF PREPARATORY PHASE
% SRT1001 S03 2014-10-12/15:39:24/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1046 S03 2014-10-12/15:39:24/000000.05 END OF PREPARATORY PHASE
% SRT1013 S01 MAIN MEMORY SORT
% SRT1013 S02 MAIN MEMORY SORT
% SRT1013 S03 MAIN MEMORY SORT

```

```
% SRT1021 S01 SORTED/MERGED RECORDS:.....10
% SRT1002 S01 2014-10-12/15:39:24/000000.38 SORT/MERGE COMPLETED
% SRT1021 S02 SORTED/MERGED RECORDS:.....10
% SRT1002 S02 2014-10-12/15:39:24/000000.13 SORT/MERGE COMPLETED
% SRT1021 S03 SORTED/MERGED RECORDS:.....10
% SRT1002 S03 2014-10-12/15:39:24/000000.08 SORT/MERGE COMPLETED
```

Kommentar zum Ablaufprotokoll

Bei diesem Beispiel müssen vor dem Start des Programms 5 Dateien zugewiesen werden:

- \$.SYSLNK.SORT.080 als Modulbibliothek, in der die SORT-Module stehen.
- RESTAURANT.SAM.FIX als Eingabedatei für das Hauptprogramm SORTZM02.
- SORT.AUS1 als Ausgabedatei für den ersten Sortierlauf.
- SORT.AUS2 als Ausgabedatei für den zweiten Sortierlauf.
- SORT.AUS3 als Ausgabedatei für den dritten Sortierlauf.
- Das Hauptprogramm SORTZM02 wird gestartet
- SORT wird über die Zugriffsmethode SORTZM aufgerufen und gibt für jeden Sortierlauf eine Startmeldung aus
- Nach Beendigung jedes Sortierlaufs meldet SORT die Anzahl der sortierten Sätze und die normale Beendigung des Sortierlaufs.

Dateimerkmale der Eingabedatei RESTAURANT.SAM.FIX

```

/show-file-attributes file-name=restaurant.sam.fix, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.RESTAURANT.SAM.FIX
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)      REC-SIZE   = 66
% AVAIL       = *STD
% WORK-FILE   = *NO        F-PREFORM  = *K              SO-MIGR   = *ALLOWED

```

Dateimerkmale der Ausgabedateien SORT.AUS1, SORT.AUS2 und SORT.AUS3

```

/show-file-attributes file-name=sort.aus*, -
/
information=*parameters(organization=*yes)
%0000000003 :CTID:$EXAMPLE.SORT.AUS1
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)      REC-SIZE   = 66
% AVAIL       = *STD
% WORK-FILE   = *NO        F-PREFORM  = *K              SO-MIGR   = *ALLOWED
%0000000003 :CTID:$EXAMPLE.SORT.AUS2
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)      REC-SIZE   = 58% AVAIL      = *STD
% AVAIL       = *STD
% WORK-FILE   = *NO        F-PREFORM  = *K              SO-MIGR   = *ALLOWED
%0000000003 :CTID:$EXAMPLE.SORT.AUS3
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN    = STD(1)          BLK-CONTR = PAMKEY
% IO(USAGE)   = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
% REC-FORM    = (F,N)      REC-SIZE   = 58
% AVAIL       = *STD
% WORK-FILE   = *NO        F-PREFORM  = *K              SO-MIGR   = *ALLOWED

```


Inhalt der Ausgabedatei SORT.AUS1

Chayota's	Thurston Street 60	292742	Japanese
Golden Fleece	Arran Street 44	242437	Yugoslavian
Java	Hope Street 51	522221	Indonesian
Le Gourmet	Lime Street 46	505397	French
Orlando's	Thompson Street 62	220061	Italian
Palenque Mexico	Millwood Drive 2	980149	Mexican
Persepolis	Salford Square 20	597004	Persian
Strawberry	Sauchiehall Street 8	595521	Vegetarian
Vietnam	Thurston Street 47	522518	Vietnamese
Willi's Bar	Westland Street 113	748293	German

Inhalt der Ausgabedatei SORT.AUS2

French	Le Gourmet	Lime Street 46
German	Willi's Bar	Westland Street 113
Indonesian	Java	Hope Street 51
Italian	Orlando's	Thompson Street 62
Japanese	Chayota's	Thurston Street 60
Mexican	Palenque Mexico	Millwood Drive 2
Persian	Persepolis	Salford Square 20
Vegetarian	Strawberry	Sauchiehall Street 8
Vietnamese	Vietnam	Thurston Street 47
Yugoslavian	Golden Fleece	Arran Street 44

Inhalt der Ausgabedatei SORT.AUS3

Golden Fleece	Yugoslavian	Arran Street 44
Java	Indonesian	Hope Street 51
Le Gourmet	French	Lime Street 46
Palenque Mexico	Mexican	Millwood Drive 2
Persepolis	Persian	Salford Square 20
Strawberry	Vegetarian	Sauchiehall Street 8
Orlando's	Italian	Thompson Street 62
Vietnam	Vietnamese	Thurston Street 47
Chayota's	Japanese	Thurston Street 60
Willi's Bar	German	Westland Street 113

10.5.24 Beispiel 24: Vollsordieren nach Daten im Unicode

Eingabe: SAM-Datei NAME mit Daten im Unicode

Ausgabe: SAM-Datei NAME.SORT mit den sortierten Sätzen

Übung zu:

- Sortieren nach Unicode (UTF-16)

Satzstruktur der Eingabedatei NAME

Name	Vorname
Müller (004D 00FC 006C 006C 0065 0072)	Paul (0050 0061 0075 006C)
Ahberg (0041 0068 0062 0065 0072 0067)	Tom (0054 006F 006D)
Werner (0057 0065 0072 006E 0065 0072)	Tom (0054 006F 006D)
Mueller (004D 0075 0065 006C 006C 0065 0072)	Peter (0050 0065 0074 0065 0072)
Ähberg (00C5 0068 0062 0065 0072 0067)	Mike (004D 0069 006B 0065)
Werner (0057 0065 0072 006E 0065 0072)	Tim (0054 0069 006D)
Werners (0057 0065 0072 006E 0065 0072 0073)	Max (004D 0061 0078)
werner (0077 0065 0072 006E 0065 0072)	Moritz (004D 006F 0072 0069 0074 007A)

5

45

84

Dateimerkmale der Datei Name

```

/show-file-attributes file-name=NAME, -
/
      information=*parameters(organization=*yes)

%0000000003 :CTID:$EXAMPLE.NAME
% ----- ORGANIZATION -----
% FILE-STRUC = SAM          BUF-LEN   = STD(1)      BLK-CONTR = PAMKEY
% IO(USAGE)  = READ-WRITE  IO(PERF)  = STD        DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)       REC-SIZE   = 0
% AVAIL      = *STD
% WORK-FILE  = *NO         F-PREFORM  = *K          SO-MIGR    = *ALLOWED

/start-sort
% SRT1001 2014-12-07/12:03:49/000000.00 SORT/MERGE STARTED,
  VERSION 08.0A00/BS2000V18.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
//assign-files input-files=NAME,output-file=NAME.SORT
//sort-records fields>(*field-explicit(position=5,length=40, -
//                      format=*unicode-character),-
//                      *field-explicit(position=45,length=40,-
//                      format=*unicode-character))
//end
% SRT1016 SORT/MERGE INPUT RECORDS:.....8 (FROM 01)
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....8
% SRT1002 2014-12-07/12:10:40/000000.14 SORT/MERGE COMPLETED

```

Inhalt der Ausgabedatei NAME.SORT (sortiert)

Zeile	Name	Vorname
1	Ahberg (0041 0068 0062 0065 0072 0067)	Tom (0054 006F 006D)
2	Åhberg (00C5 0068 0062 0065 0072 0067)	Mike (004D 0069 006B 0065)
3	Mueller (004D 0075 0065 006C 006C 0065 0072)	Peter (0050 0065 0074 0065 0072)
4	Müller (004D 00FC 006C 006C 0065 0072)	Paul (0050 0061 0075 006C)
5	werner (0077 0065 0072 006E 0065 0072)	Moritz (004D 006F 0072 0069 0074 007A)
6	Werner (0057 0065 0072 006E 0065 0072)	Tim (0054 0069 006D)
7	Werner (0057 0065 0072 006E 0065 0072)	Tom (0054 006F 006D)
8	Werners (0057 0065 0072 006E 0065 0072 0073)	Max (004D 0061 0078)

5

45

84

Erläuterung zur Ausgabedatei

- Zeile 1-2 Auf Ebene1 sind Ahberg und Åhberg gleich.
Die Sortierreihenfolge wird auf Ebene 2 festgelegt: Å > A
- Zeile 3-4 Bei Müller wird das ü nicht wie ue behandelt, sondern wie ein u mit diakritischem Zeichen.
Die Sortierreihenfolge wird auf Ebene 1 festgelegt: ü > ue
- Zeile 5-6 Auf Ebene1 ist das klein geschriebene werner identisch mit dem groß geschriebenen Werner.
Die Sortierreihenfolge wird auf Ebene 3 festgelegt: W > w
- Zeile 6-7 Bei Werner ist im ersten Sortierfeld kein Unterschied.
Die Sortierreihenfolge wird durch das nächste Sortierfeld festgelegt: Tom > Tim
- Zeile 7-8 Die Sortierreihenfolge wird auf Ebene 1 festgelegt:
Werners > Werner

10.6 Verzeichnis der Beispieldateien

10.6.1 Vorbemerkung

In den Beispielen werden verschiedene Dateien zur Demonstration verwendet. Der Inhalt der unsortierten Dateien ist dann gleich, wenn der erste Teil des Dateinamens gleich ist. Die Dateieigenschaften ändern sich jedoch bei verschiedenen Beispielen, da schließlich die Zugriffsmethode die Position der Daten bestimmt. Die Art der Datei ist deshalb im Dateinamen enthalten.

So wird die Datei mit dem ersten Namensteil RESTAURANT in folgenden Ausprägungen verwendet:

Eigenschaften	Dateiname
SAM-Datei mit fester Satzlänge	RESTAURANT.SAM.FIX
SAM-Datei mit variabler Satzlänge	RESTAURANT.SAM
ISAM-Datei mit variabler Satzlänge	RESTAURANT.ISAM.

Variable Dateien weisen also im Dateinamen nur den Zugriffstyp auf, Dateien mit fester Satzlänge wurden mit dem Zusatz FIX versehen.

Die Ausgabedateien sind mit dem Zusatz SORT entsprechend gekennzeichnet. RESTAURANT.ISAM.SORT wäre die sortierte Datei zu RESTAURANT.ISAM.

Es folgt deshalb ein Verzeichnis der unsortierten Ausgangsdateien. Am Ende der Anzeige des Inhalts findet sich eine Zahlenreihe, die es erleichtern soll, die einzelnen Positionen zu bestimmen.

Es bleibt zu beachten, dass bei SAM-Dateien mit variabler Satzlänge zu der Buchstabenposition das vier Byte lange Satzlängenfeld hinzuaddiert werden muss. Bei ISAM-Dateien gilt dies entsprechend für den ISAM-Schlüssel.

10.6.2 Inhalt der Datei RESTAURANT

Orlando's	Thompson Street 62	220061	Italian
Java	Hope Street 51	522221	Indonesian
Golden Fleece	Arran Street 44	242437	Yugoslavian
Le Gourmet	Lime Street 46	505397	French
Palenque Mexico	Millwood Drive 2	980149	Mexican
Strawberry	Sauchiehall Street 8	595521	Vegetarian
Persepolis	Salford Square 20	597004	Persian
Vietnam	Thurston Street 47	522518	Vietnamese
Chayota's	Thurston Street 60	292742	Japanese
Willi's Bar	Westland Street 113	748293	German
1	21	48	56 SAM-fix 66
5	25	52	60 SAM-variabel 70
13	33	60	68 ISAM-variabel 78

10.6.3 Inhalt der Datei LITERATURE

Pasternak	Boris	Doctor Zhivago	Novel
Capote	Truman	In Cold Blood	Novel
Boyle	Jimmy	A Sense of Freedom	Autobiography
Arden	John	Sergeant Musgrave's Dance	Theatre
Milligan	Spike	Puckoon	Novel
Dahl	Roald	Kiss Kiss	Short Stories
Shakespeare	William	Romeo and Juliet	Theatre
Fielding	Henry	Tom Jones	Novel
Jonson	Ben	Volpone	Theatre
Dumas	Alexandre	The Three Musketeers	Novel
Troyat	Henri	Pushkin	Biography
Shaw	Bernard	Pygmalion	Theatre
Sharpe	Tom	Riotous Assembly	Novel
Thomas	Dylan	Fern Hill	Poem
Gogol	Nikolai	Dead Souls	Novel
1	15	26	59 SAM-fix 71
5	19	30	63 SAM-variabel 75
13	27	38	71 ISAM-variabel 83

10.6.4 Inhalt der Datei MUSEUM

Ad Lib	Bryant Street 21	331452	Dixieland, trad. jazz	
British China Museum	Majestic Square 2	221315	China, porcelain	
British Opera House	Prince Albert Square 1	222591	Opera, classical concerts	
British Theatre	Majestic Square 2	2185413	Classical theatre	
Folk Company	St. Vincent Street 15	241977	Traditional folk theatre	
Gallery Dubois	Sanders Street 59	333510	Contemporary artists	
Gallery Dumont	Milton Drive 29	298841	Latim American artists	
Gallery Franke	Milton Drive 22	226420	Beckmann, Dix collection	
Gallery Schubert	Prince Albert Square 6	475859	Modern art	
Gallery Thomas	Milton Drive 60	295517	Contemporary sculpture	
Kensington Club	Milton Drive 47	221859	Various art exhibitions	
King's Theatre	Gallery Street 4	221152	Experimental theatre	
Modern Theatre	High Street 3	3445154	Fringe theatre	
Municipal Museum	St James's Square 1	2332254	History of the city	
New Art Gallery	Prince Albert Square 7	5591307	Impressionists	
New City Museum	Prince Albert Square 7	5591307	Beuys, Picasso	
Scottish Art House	Prince Albert Square 9	224407	Scottish artists	
Subway	Soho Lane 23	399482	Modern jazz	
Susi's Hideaway	Pratt Street 10	2342660	Nightclub, cabaret	
Symposium	George Street 47	363546	Poetry readings	
Theatre Royal	Marshall Square 11	225754	Repertory theatre	
Vic's	Islington Road 15	223266	Stand-up comics	
Wings	Green Street 3	795088	Pop music, disco	
1	22	45	53 SAM-fix	77
5	26	49	57 SAM-variabel	81
13	34	57	65 ISAM-variabel	89

10.6.5 Inhalt der Datei CULTURE.1

Aquitaine	Acacia Avenue 39	284028	French
August Gardens	Newton Street 16	2604106	Argentinian
Bosna	Freeling Street 11	64115447	Yugoslavian
Bouillabaisse	Falcon Street 10	297909	French
Datscha	King Street 3	341218	Russian
Don Quixote	Billington Street 6	342318	Spanish
Frank's Grill	Forest Road 14	281235	Yugoslavian
Mandarin	Lime Street 21	226888	Chinese
Mifune	Iffman Street 138	987572	Japanese
Nitaya	Thompson Street 19	197772	Thai
Saint George's	Upman Street 67	363666	English
Siracusa	Polson Street 33	770613	Italian
Slavonia	Allcock Street 16	564906	Yugoslavian
Spiros	Upman Street 65	366883	Greek
Zung-Hua	Bond Street 33	555320	Chinese
1	22	44	54 SAM-fix 65
5	26	48	58 SAM-variabel 69
13	34	56	66 ISAM-variabel 77

10.6.6 Inhalt der Datei CULTURE.2

Asado Steak	Taylor Drive 1	294577	Argentinian
Auberge	Richmond Drive 15	347577	French
Aubergine	Penman Street 19	674829	Argentinian
Baltic Grillhouse	Dartford Road 33	554401	Yugoslavian
Bologna	Leslie Street 23	393939	Italian
Dreamhouse	Robertson Street 5	268408	Argentinian
Lung Fung	Zeppelin Street 10	593741	Chinese
Makarska Grill	Stone Street 2	369766	Yugoslavian
Molina's	Pitman Square 33	765263	Spanish
Sultana	Fulton Street 28	332871	Indian
Tai Tung	Acacia Avenue 77	281104	Chinese
Tivoli	Wilberforce Drive 52	221274	Italian
Venezia	Landers Street 84	847414	Italian
Veracruz	Landers Street 207	5702520	Mexican
1	22	44	54 SAM-fix 65
5	26	48	58 SAM-variabel 69
13	34	56	66 ISAM-variabel 77

10.6.7 Inhalt der Datei CULTURE.3

Alcazar's	Doubleway Drive 39	8111590	Argentinian
Buenos Aires	Zoo Road 22	779646	Argentinian
Canton	Theresa Street 49	522185	Chinese
China House	May Street 20	531620	Chinese
El Cid	Belgrade Street 45	3003268	Spanish
Golden Bough	Oswald Street 44	242437	Hungarian
La Barca	Blytheswood Square 1	77613	Italian
Opatija	Robertson Street 2	268353	Yugoslavian
Scorpio	Leslie Street 35	399897	Greek
Tivoli	Wilberforce Drive 52	221274	Italian
Torino	Gardener Lane 8	469571	Italian
Watermill	Leslie Street 33	348000	Swiss
Why Not?	Wimbledon Drive 11	399936	French
Zigzag	Taylor Drive 72	226750	Argentinian
1	22	44	54 SAM-fix 65
5	26	48	58 SAM-variabel 69
13	34	56	66 ISAM-variabel 77

11 Meldungen des Sortier-/Mischprogramms

Die Meldungen des Sortier-/Mischprogramms werden immer über SYSOUT ausgegeben. Zusätzlich können so genannte garantierte Meldungen in S-Variablen ausgegeben werden.

11.1 Ausgabe von Meldungen auf SYSOUT

Im Normalfall erscheinen die Meldungen des Sortier-/Mischprogramms im Meldungsformat mit den Kennzeichen SRT1001 bis SRT13xx. Wird SORT als Unterprogramm oder über die Schnittstelle SORTZM aufgerufen und über die Makroaufrufe SRT0 oder SRT1 versorgt, erscheint hinter jeder Meldungsnummer das beim ersten SORT-Makroaufruf angegebene Präfix. Die Meldungen haben dann das Format „SRT1xxx präfix“.

Die Meldungen enthalten Hinweise über die Maßnahmen und den Zustand des Sortier-/Mischprogramms nach Ausgabe der Meldungen. Jeder Meldung ist eine Priorität zugeordnet. Sie ist im Bedeutungstext der Meldung angegeben, z.B. 2 für die Meldung SRT1161. Die Ausgabe der Meldungen durch SORT kann über diese Prioritäten mit dem MIN-MSG-WEIGHT-Operanden der SET-SORT-OPTIONS-Anweisung gesteuert werden. Die Meldungen werden nur ausgegeben, wenn die über MIN-MSG-WEIGHT angegebene Priorität kleiner oder gleich der angegebenen ist. Als Prioritätsangabe sind die entsprechenden Schlüsselwerte erlaubt. Dabei ist *ALL die niedrigste und *NONE die höchste Priorität. Eine Zuordnung von Schlüsselwort zu Meldungspriorität kann vom Anwender vorgenommen werden. Die Prioritätsangabe *NONE bewirkt, dass alle Meldungen außer den Meldungen über interne Fehler unterdrückt werden. Für MIN-MSG-WEIGHT=*NORMAL gilt folgende Voreinstellung:

- bei autonomen Sortier-/Mischläufen werden Meldungen ab Priorität 2 bzw. der kennungs- oder anlagenspezifisch eingestellten Priorität ausgegeben.
- beim Aufruf von SORT als Unterprogramm werden Meldungen ab Priorität 3 ausgegeben.

Sonderfälle gibt es während der Auswertung der SORT-Anweisungen:

- ist SYSDTA keine Datenstation, werden alle Eingaben, z.B. die Anweisungen an SORT, unverändert auf SYSOUT aufgelistet.
- Im Dialogbetrieb muss, abhängig von der eingestellten GUIDANCE bei fehlerhafter Eingabe einer Anweisung (Kommando MODIFY-SDF-OPTIONS, siehe Handbuch „Kommandos“ [1]) die Anweisung nach Ausgabe der SORT-Meldung noch einmal richtig eingegeben werden bzw. der von SDF angebotene Korrekturdialog durchgeführt werden. Findet das Sortier-/Mischprogramm den Fehler jedoch erst während der Gegenprüfung oder in der Planungsphase, so müssen alle Anweisungen noch einmal eingegeben werden.

An bestimmten Stellen während der Verarbeitung, z.B. am Anfang oder Ende eines Sortier-/Mischlaufs, beim Aktivieren eines Benutzerausgangs oder bei einer fehlerhaften Beendigung des Sortier-/Mischlaufs, erscheinen Meldungen, die die Situation beschreiben. Im Dialogbetrieb ist bei einigen Meldungen eine Antwort nur möglich, wenn beim Benutzerausgang die Option DIALOG verwendet wurde. Diese Aktionsantworten können abgekürzt werden. Die zulässigen Antworten werden in der anfordernden Meldung aufgelistet, wobei Zeichen der Antwort, die weggelassen werden dürfen, in eckigen Klammern angegeben sind (z.B. C[ONTINUE]).

Hinweise

- Meldungen des Sortier-/Mischprogramms, die den Ablaufzustand anzeigen, beginnen in der Regel mit der Angabe des Datums, der Uhrzeit und der aufgelaufenen CPU-Zeit. Dabei wird das Datum in der Form yyyy-mm-dd, die Uhrzeit in der Form hh:mm:ss und die CPU-Zeit in der Form ssssss.ss dargestellt. Beim Datum bedeuten yyyy = Jahr, mm = Monat und dd = Tag. Bei der Uhrzeit bedeuten hh = Stunden, mm = Minuten und ss = Sekunden.
Die CPU-Zeit bezieht sich jeweils auf den Start von SORT/MERGE (SRT1001). Teilablaufzeiten sind deshalb durch Differenzbildung zu ermitteln, ebenso die Laufzeit über die Uhrzeiten-Differenz.
- Die nachfolgenden Angaben über die Fortsetzung bzw. den Abbruch des Sortier-/Mischlaufs beziehen sich auf den Dialogbetrieb. Im Stapelbetrieb wird bei einer fehlerhaften Anweisung die nächste Anweisung eingelesen. In der Regel führt das zum fehlerhaften Abbruch des Sortier-/Mischprogramms.
- Die Meldungen SRT1016 bis SRT1026 informieren über SORT-Leistungen. Die Meldungen werden bei Bedarf und bei Beendigung von SORT ausgegeben. Ist der Zähler der Leistung gleich null oder seit der letzten Ausgabe unverändert, so erfolgt die Ausgabe der Meldung nur bei Angabe von MIN-MSG-WEIGHT=*ALL in der SET-SORT-OPTIONS-Anweisung.

- Die Dateizustände nach einer normalen oder fehlerhaften Beendigung des Sortier-/Mischlaufs sind im [Abschnitt „Endebehandlung der SORT-Dateien“ auf Seite 117](#) beschrieben.
- Beim Multitasksortieren gibt es von jeder beliebigen Task (Hauptsortier- und Nebentaskläufe) ein eigenes Ablaufprotokoll.
Die Ablaufprotokolle der Nebentaskläufe werden jedoch nur erstellt, wenn in der SET-SORT-OPTIONS-Anweisung der Operand MIN-MSG-WEIGHT=*ALL angegeben wurde oder ein Fehler in der Sortiernebantask auftrat.

11.2 Ausgabe von Meldungen in S-Variablen

Für eine Reihe von Meldungen werden Meldungsschlüssel und Inserts (Nummer und Semantik) als unveränderliche Bestandteile für zukünftige SORT- und BS2000/OSD-BC-Versionen garantiert. Solche Meldungen werden als garantierte Meldungen bezeichnet.

Garantierte Meldungen des Produkts SORT

Garantierte Meldungen von SORT sind Meldungen, die Auskunft geben über

- Start oder Beendigung des SORT
- Anzahl der sortierten Sätze
- Anzahl der gelöschten, eingefügten oder summierten Sätze

Die Nummern dieser Meldungen und ihre Bedeutung sind folgender Tabelle zu entnehmen:

Meldungsnummer	Bedeutung
SRT1001	SORT startet
SRT1002	SORT endet normal
SRT1003	Unterbrechung durch Anwender
SRT1016	Anzahl der Eingabesätze
SRT1017	Anzahl der zu sortierenden Sätze
SRT1018	Anzahl der eingefügten Eingabesätze
SRT1019	Anzahl der gelöschten Eingabesätze
SRT1020	Anzahl der gelöschten SUM-Sätze
SRT1021	Anzahl der sortierten Sätze
SRT1022	Anzahl der eingefügten Ausgabesätze
SRT1023	Anzahl der gelöschten Ausgabesätze
SRT1024	Anzahl der gelöschten Sätze auf Grund von SELECT-INPUT-RECORDS oder INCLUDE/OMIT
SRT1030	Anzahl der Ausgabesätze
SRT1034	keine Sätze zu sortieren
SRT1038	SORT-Abbruch
SRT1059	SORT fehlerhaft beendet

Bei Einsatz des kostenpflichtigen Produkts SDF-P besteht die Möglichkeit, diese garantierten Meldungen in strukturierte S-Variablen auszugeben. Über S-Variablen können Sie direkt auf bestimmte Meldungsdaten zugreifen, ohne dass Sie das Ausgabelayout der Meldungen kennen müssen. Damit ist es möglich, in SDF-P-Prozeduren abhängig vom Inhalt dieser Variablen die Weiterverarbeitung zu steuern.

Bei garantierten Meldungen wird das Meldungsattribut „Garantie“ mit der Zeile „◆ Warranty: Y“ nach dem Meldungstext dokumentiert.

Vorgehensweise

Eine S-Variable, die Meldungen aufnehmen kann, muss als Liste von Strukturen aufgebaut sein. Ihr Name ist frei wählbar und wird im Folgenden durch `varname` dargestellt. Sie können die Variable mit folgendem Kommando vereinbaren:

```
/DECLARE-VARIABLE NAME=varname(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC)), -
/                               MULTIPLE-ELEMENTS=*LIST,SCOPE=*TASK
```

Hinweis

Die Angabe `SCOPE=*TASK` ist nur erforderlich, wenn die Variable auch in Prozeduren gültig sein soll, die nach der Vereinbarung der Variablen aufgerufen werden. Dies ist insbesondere dann der Fall, wenn `SORT` über das Kommando `SORT-FILE` gestartet wird, da dieses Kommando als Prozedur implementiert ist.

Für jede auszugebende Meldung ist ein Listenelement der Variablen vorgesehen, welches wiederum eine Struktur ist. Diese Struktur besteht aus folgenden Elementen:

Name des Strukturelements	Inhalt
<code>varname(*LIST).MSG-TEXT</code>	vollständiger Meldungstext
<code>varname(*LIST).MSG-ID</code>	Meldungsschlüssel
<code>varname(*LIST).IO</code>	Insert 0
:	
<code>varname(*LIST).In</code>	Insert <i>n</i> (<i>n</i> ist die Anzahl der Inserts, die von der Meldung abhängt)

Nach der Vereinbarung der S-Variablen weisen Sie ihr den Meldungsstrom zu:

```
/ASSIGN-STREAM STREAM-NAME=SYSMSG,T0=*VARIABLE(varname)
```

Dann starten Sie SORT oder ein Programm, das SORT als Unterprogramm aufruft. Nach der Beendigung von SORT oder des Programms, weisen Sie den Meldungsstrom wieder dem Standardausgabemedium für Meldungen zu.

```
/ASSIGN-STREAM STREAM-NAME=SYSMSG,TO=*STD
```

Die Variable `varname` enthält alle **garantierten** Meldungen, die zwischen den beiden ASSIGN-STREAM-Kommandos ausgegeben werden. Das bedeutet:

- Wenn Sie SORT als autonomes Programm mit START-SORT oder SORT-FILE starten, enthält die Variable alle garantierten Meldungen, die von SORT ausgegeben wurden, sowie die während des SORT-Laufs ausgegebenen garantierten Meldungen des Binder-Lader-Systems (BLS).
- Wenn Sie SORT als Unterprogramm aufrufen, enthält die Variable zusätzlich alle garantierten Meldungen, die vom Hauptprogramm ausgegeben wurden. Um die Ausgabe der zusätzlichen Meldungen zu unterdrücken, können Sie die beiden ASSIGN-STREAM-Kommandos auch per CMD-Makro unmittelbar vor und nach dem SORT-Aufruf im Hauptprogramm ausführen.

Hinweis

Der Operand MIN-MSG-WEIGHT in der SET-SORT-OPTIONS-Anweisung hat keinen Einfluss auf die Meldungsausgabe in S-Variablen.

Eine detaillierte Beschreibung der S-Variablen finden Sie im Handbuch „[SDF-P \(BS2000\)](#)“ [\[13\]](#)

Auswertung von S-Variablen

Der Inhalt einer S-Variablen kann mit dem Kommando SHOW-VARIABLE ausgegeben werden. Nach einem SORT-Lauf könnte eine S-Variable folgendermaßen belegt sein:

```
/show-variable variable-name=sortmip
.
.
.
SORTMIP(*LIST).MSG-ID = SRT1001
SORTMIP(*LIST).IO =
SORTMIP(*LIST).I1 = 2014-10-12/18:11:11
SORTMIP(*LIST).I2 = 000000.00
SORTMIP(*LIST).I3 = 08.0A00/BS2000V18.0
SORTMIP(*LIST).MSG-TEXT = % SRT1016 SORT/MERGE INPUT RECORDS:
.....10 (FROM 01)
SORTMIP(*LIST).MSG-ID = SRT1016
SORTMIP(*LIST).IO =
SORTMIP(*LIST).I1 = .....10
SORTMIP(*LIST).I2 = 01
SORTMIP(*LIST).MSG-TEXT = % SRT1030 SORT/MERGE OUTPUT RECORDS:
.....10
SORTMIP(*LIST).MSG-ID = SRT1030
SORTMIP(*LIST).IO =
SORTMIP(*LIST).I1 = .....10
SORTMIP(*LIST).MSG-TEXT = % SRT1002 2014-10-12/18:11:11/000000.08 SORT/
MERGE COMPLETED
SORTMIP(*LIST).MSG-ID = SRT1002
SORTMIP(*LIST).IO =
SORTMIP(*LIST).I1 = 2014-10-12/18:11:11
SORTMIP(*LIST).I2 = 000000.08
```

Die einzelnen Elemente der Variablen können auch gezielt angesprochen werden. So erhält man die Struktur der zweiten Meldung mit folgendem Kommando:

```
/show-variable variable-name=sortmip#2
```

Den Meldungsschlüssel der dritten Meldung ermittelt man entsprechend:

```
/show-variable variable-name=sortmip#3.msg-id
```

Im Allgemeinen lässt sich die Position einer gesuchten Meldung innerhalb der Variablen nicht vorhersagen. Daher empfiehlt es sich, zur Abfrage einer bestimmten Meldung eine SDF-P-Prozedur mit einer FOR-Schleife zu verwenden, wie sie im folgenden Beispiel vorgestellt wird.

Beispiel für eine SDF-P-Prozedur zur Auswertung von S-Variablen

Nach einem SORT-Lauf soll geprüft werden, ob die Meldung SRT1034 (keine Sätze sortiert) ausgegeben wurde. Das Ergebnis wird auf dem Bildschirm angezeigt.

```

/DECLARE-VARIABLE NAME=SORTMIP(TYPE=STRUCTURE),           - (1)
/
      MULTIPLE-ELEMENTS=LIST,SCOPE=TASK
/DECLARE-VARIABLE NAME=CURRENT(TYPE=STRUCTURE)           (2)
/DECLARE-VARIABLE NAME=FOUND,TYPE=BOOLEAN,INITIAL-VALUE=FALSE (3)
/FREE-VARIABLE NAME=SORTMIP                             (4)
/ASSIGN-STREAM STREAM-NAME=SYSMSG,TO=*VARIABLE(SORTMIP) (5)
/SORT-FILE INPUT-FILES=EINGABE,OUTPUT-FILE=AUSGABE      (6)
/ASSIGN-STREAM STREAM-NAME=SYSMSG,TO=*STD              (7)
/FOR CURRENT=*LIST(SORTMIP)                            (8)
/ IF (CURRENT.MSG-ID = 'SRT1034')
/   WRITE-TEXT '-----'
/   WRITE-TEXT '---  keine Saetze sortiert  ---'
/   WRITE-TEXT '-----'
/   FOUND = TRUE
/ END-IF
/END-FOR
/IF (FOUND = FALSE)                                     (9)
/ WRITE-TEXT '-----'
/ WRITE-TEXT '--- mindestens ein Satz sortiert ---'
/ WRITE-TEXT '-----'
/END-IF

```

- (1) Die S-Variable SORTMIP wird als Liste von Strukturen vereinbart.
- (2) Die Strukturvariable CURRENT wird vereinbart. Sie soll als Laufvariable der FOR-Schleife verwendet werden.
- (3) Der Bitschalter FOUND wird vereinbart, um das Ergebnis der Suche nach der Meldung SRT1034 aufzunehmen. Er wird mit dem Wert FALSE initialisiert.
- (4) Für den Fall, dass die Variable SORTMIP bereits vor der Vereinbarung in dieser Prozedur existierte, wird ihr Inhalt gelöscht.
- (5) Ab jetzt werden garantierte Meldungen in der Variablen SORTMIP abgelegt.
- (6) SORT wird gestartet.
- (7) Das Ablegen von Meldungen in der Variablen wird gestoppt.
- (8) In einer FOR-Schleife wird die gesamte Strukturen-Liste nach einem Eintrag mit MSG-ID SRT1034 durchsucht. Falls solch ein Eintrag gefunden wird, dann wird der Text „keine Sätze sortiert“ ausgegeben und der Schalter FOUND auf TRUE gesetzt.
- (9) Nach Abschluss der FOR-Schleife wird geprüft, ob der Schalter FOUND noch auf FALSE steht. Wenn ja, dann wird der Text „mindestens ein Satz sortiert“ ausgegeben.

11.3 SORT-/MERGE-Meldungen

Den Meldungstext und optional auch Bedeutungs- und Maßnahmetext zu einem Meldungsschlüssel können Sie mit dem Kommando HELP-MSG-INFORMATION abfragen.

Die Meldungen von SORT/MERGE finden Sie auch über eine HTML-Anwendung auf dem Manual-Server (URL: <http://manuals.ts.fujitsu.com>) und auf der DVD „BS2000 SoftBooks“.

12 Anhang

12.1 SORT-Fehlerbehandlung

12.1.1 Behandlung von SORT-internen Fehlern

Interne Fehler des SORT können erkannt werden über

- Kommandoreturncodes mit Subcode1=32
- die SORT-Meldungen „SORT-INTERNER FEHLER“ oder
- ein sonstiges Fehlerverhalten des Sortier-/Mischprogramms.

Treten solche Fehler auf, sollten die folgenden Fehlerunterlagen gesammelt werden und dem Systemkundendienst zur Verfügung gestellt werden.

Ablaufprotokolle

Die Ablaufprotokolle sollen eine Auflistung der Kommando- und SORT/MERGE-Anweisungsfolgen enthalten. Gegebenenfalls sollte auch das Bedienungsplatzprotokoll des Operateurs mitgeliefert werden.

Speicherabzüge

Der fehlerhafte Zustand wird auf Speicherabzügen festgehalten.

Meldet SORT einen internen Fehler über SRT1039, druckt SORT einen speziellen SORT-DUMP aus. Außerdem wird noch ein allgemeiner DUMP in eine Datei ausgegeben, der über das Dienstprogramm DAMP auszudrucken ist. Die Ausgabe eines Dump kann durch den Operanden DUMP der SET-SORT-OPTIONS-Anweisung gesteuert werden.

Verhalten bei Datenfehlern

Ist das angegebene Format eines Sortier-, Summen- oder Vergleichsfeldes nicht mit dem Inhalt des Eingabesatzes verträglich, wird die Meldung

```
SRT1047 *** DATENFEHLER X'ff' BEI BEFEHLSZAEHLER X'zzzzzzzzz' ***
```

ausgegeben. In Abhängigkeit von der Option DUMP werden in diesem Fall ein Speicherabzug und die beteiligten Sätze nach SYSLST und im Dialog zusätzlich maximal 8 Zeilen pro Satz nach SYSOUT ausgegeben.

Bei sonstigen Fehlern sollte der Fehler mit DUMP festgehalten werden.

Dateien

Die Dateimerkmale aller am Sortier-/Mischlauf beteiligten Dateien sollten mit dem Kommando

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=dateiname,INFORMATION=*ALL-ATTRIBUTES
```

im Protokoll aufgelistet werden. Daneben sollten alle diese Dateien in dem Zustand aufgehoben werden, der der Fehlersituation entspricht. Dies gilt insbesondere bei Fehlern im Zusammenhang mit einem Fixpunkt oder einem RESTART-PROGRAM.

12.1.2 Fehlerinformation beim Aufruf von SORT als autonomes Programm

Die Information, ob SORT als autonomes Programm erfolgreich oder mit Fehler beendet wird, ist durch die Beendigungsart von SORT festgelegt. SORT kann wie folgt beenden:

- Normale (fehlerfreie) SORT-Beendigung mit Makroaufruf TERM MODE=NORMAL
- Fehlerhafte SORT-Beendigung bzw. vorzeitiger Abbruch mit Makroaufruf TERM UNIT=STEP,MODE=ABNORMAL

Wird SORT in einer Prozedur aufgerufen, kann der Benutzer bei fehlerhafter SORT-Beendigung (TERM UNIT=STEP,MODE=ABNORMAL) Maßnahmen treffen, indem er über den SPIN-OFF-Mechanismus mit entsprechenden Kommandos (z.B. SET-JOB-STEP-Kommando) in der Prozedur verzweigt (siehe auch Handbuch „Kommandos“ [1], SET-JOB-STEP-Kommando).

Zustand des SORT-Laufs in einer Jobvariablen

Für den Ablauf von SORT kann der Benutzer eine programmüberwachende Jobvariable (JV) verwenden (siehe Handbuch „Jobvariablen“ [8]). Voraussetzung ist, dass die Funktion JV installiert ist. Informationen über den SORT-Lauf werden in dieser JV von SORT hinterlegt und können daraus vom Benutzer entnommen werden. Die Jobvariable wird vom Benutzer beim Aufruf von SORT zugeordnet mit dem Kommando

```
/START-SORT MONJV=jvname
```

Die programmüberwachende Jobvariable enthält im Byte 16 das Zeichen „P“ als Kennzeichen einer programmüberwachenden Jobvariablen. Die Bytes 0 - 6 haben nachfolgenden Aufbau und Inhalt. Alle übrigen Bytes sind reserviert.

Byte 0	1	2	3	4	5	6
Zustandsanzeige			Rückkehrcode-Anzeige			
			Be- endigungs- code	Programminformation		

Die Zustandsanzeige gibt den aktuellen Zustand des SORT-Laufs wieder und kann folgende Werte annehmen:

Wert	Bedeutung	Anlass der Wertzuweisung
\$R_	SORT läuft	SORT-Lauf gestartet
\$T_	SORT-Lauf normal beendet	SORT-Lauf wurde erfolgreich abgeschlossen (TERM MODE=NORMAL).
\$A_	SORT-Lauf fehlerhaft beendet	SORT-Lauf wurde durch einen Programmfehler oder durch einen definierten Fehlerausgang vorzeitig beendet (TERM UNIT=STEP,MODE=ABNORMAL).

Die folgenden 4 Byte (Byte 3 - 6) enthalten die Rückkehrcode-Anzeige. Die Rückkehrcode-Anzeige besteht aus dem Beendigungscode (Byte 3) und der Programminformation (Byte 4 - 6).

Der Beendigungscode kann folgende Werte annehmen:

Wert	Bedeutung
C'0'	Der SORT-Lauf wurde normal beendet. Keine Fehler.
C'1'	Während des SORT-Laufs wurde eine oder mehrere Warnungsmeldungen ausgegeben. Das Ergebnis sollte überprüft werden.
C'2'	Der SORT-Lauf wurde fehlerhaft beendet. Die Ergebnisse sind fehlerhaft oder unvollständig.
C'3'	Der SORT-Lauf wurde wegen schwerwiegender Fehler beendet oder vorzeitig abgebrochen. Ergebnisse sind nicht verwertbar oder nicht vorhanden.

Die Programminformation enthält die letzten 3 Dezimalziffern (im EBCDIC) des Meldungsschlüssels der Fehler- bzw. Warnungsmeldung, die den Fehler beschreibt.

Beispiel

Wird der SORT-Lauf mit der Fehlermeldung SRT1206 vorzeitig abgebrochen, hat die Rückkehrcode-Anzeige in Byte 3-6 den Inhalt C'3206'.

Hinweis

Zum Auswerten der Jobvariablen sollte sich der Benutzer ihren Inhalt mit dem Kommando /SHOW-JV ausgeben lassen (siehe Handbuch „Jobvariablen“ [8]).

12.1.3 Fehlerinformation beim Aufruf von SORT als Unterprogramm

Fehlerinformation in Register 15

Wird SORT als Unterprogramm aufgerufen, enthält Register 15 im niedrigstwertigen Byte ein Rücksprungkennzeichen, das bei Beendigung des Sortier-/Mischlaufs übergeben wird.

X'00' der Sortier-/Mischlauf endete normal.

X'FF' der Sortier-/Mischlauf endete fehlerhaft. Die Fehler werden über eine Meldung angezeigt.

Im Fehlerfall enthalten die beiden höchstwertigen Byte zusätzlich die letzten 4 Stellen des SORT-Meldungsschlüssels. Der SORT-Meldungsschlüssel wird dabei als gepackte Dezimalzahl ohne Vorzeichenzone abgespeichert.

Beispiel

Wird der SORT-Lauf mit der Fehlermeldung SRT1035 fehlerhaft beendet, hat Register 15 den Inhalt X'103500FF'.

Fehlerinformation im RCF-Bereich

Beim Aufruf von SORT als Unterprogramm mit Versorgung durch die Makroaufrufe SRT0 oder SRT1 kann ein RCF-Bereich im Hauptprogramm angegeben werden (Operand RCF im Makroaufruf). In diesem Bereich übergibt SORT beim Beenden des SORT-Laufs Rückkehrinformation und legt die ausgegebenen SORT-Meldungen ab. Auf die Adresse des RCF-Bereich (A-Adresse oder S-Adresse) verweist der Übergabebereich.

12.1.4 Aufbau des RCF-Bereiches

Byte 0	1	2	3	4	
RC				Ablage für SORT-Meldungen			

Die einzelnen Felder des RCF-Bereichs haben folgende Bedeutung:

RC

Die RC-Information hat den gleichen Aufbau wie die Rückkehrcode-Anzeige der programmüberwachenden Jobvariablen (siehe Seite 407).

- Byte 0: Beendigungscode
 Der Beendigungscode kann folgende Werte annehmen:
- C'0' Der SORT-Lauf wurde normal beendet. Keine Fehler.
 - C'1' Während des SORT-Laufs wurden eine oder mehrere Warnungsmeldungen ausgegeben. Das Ergebnis sollte überprüft werden.
 - C'2' Der SORT-Lauf wurde fehlerhaft beendet. Die Ergebnisse sind fehlerhaft oder unvollständig.
 - C'3' Der SORT-Lauf wurde wegen schwerwiegender Fehler beendet oder vorzeitig abgebrochen. Ergebnisse sind nicht verwertbar oder nicht vorhanden.

Byte 1 - 3: Programmierinformation

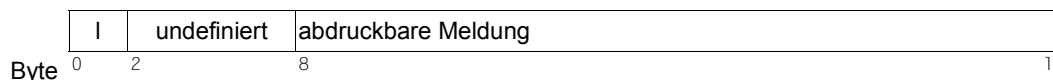
Sie enthält die letzten 3 Dezimalziffern (im EBCDIC) des Meldungsschlüssels der Fehler- bzw. Warnungsmeldung, die den Fehler beschreibt.

Beispiel

Wird der SORT-Lauf mit der Fehlermeldung SRT1045 vorzeitig abgebrochen, hat die RC-Information den Inhalt C'3045'.

Ablage für SORT-Meldungen

Die Meldungen werden aufeinander folgend bis zum Ende des RCF-Bereiches abgelegt (SYSLST-Format). Das Format einer Meldung ist:



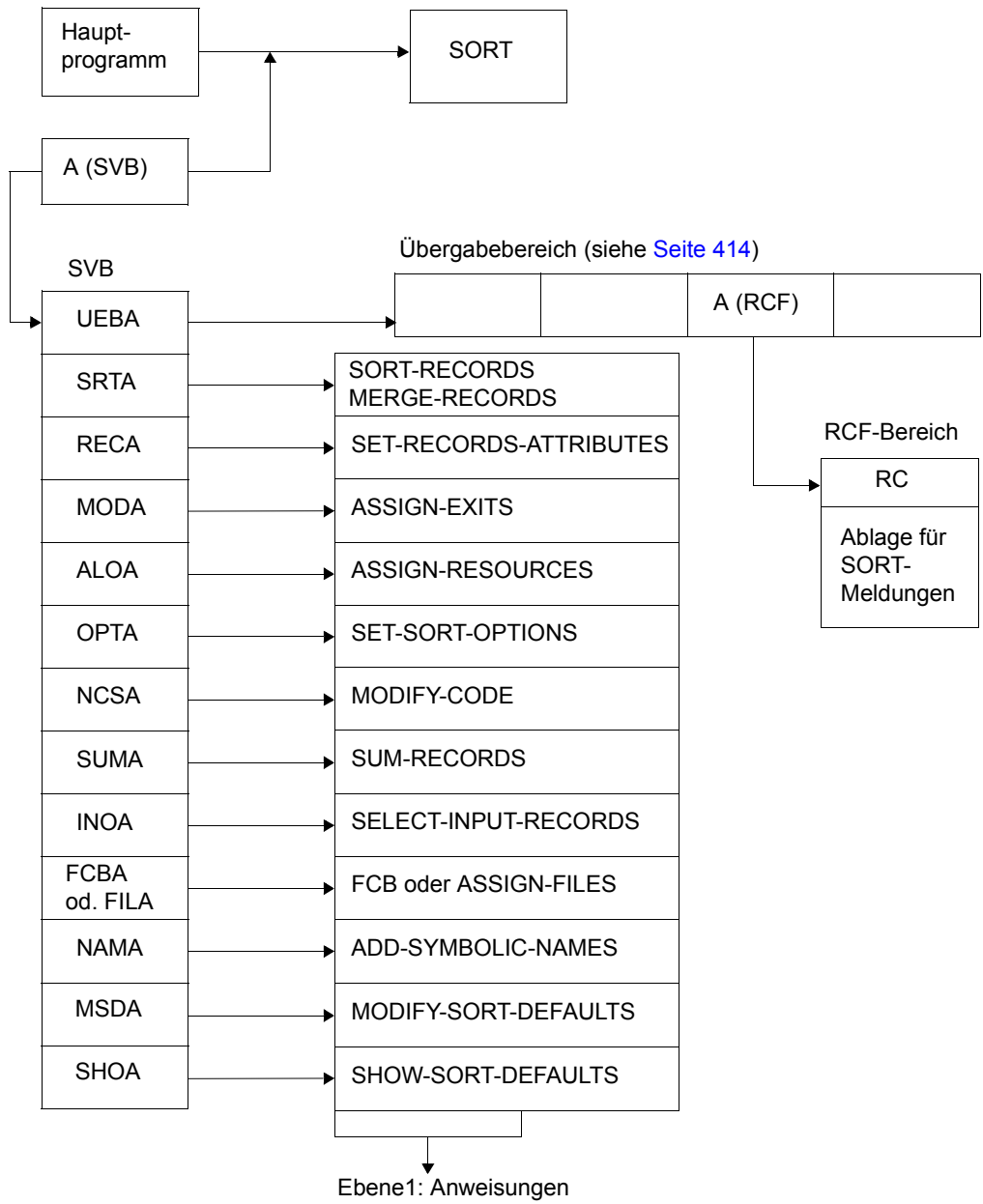
Dabei bezeichnet l die Gesamtlänge einer Meldung.

Reicht der RCF-Bereich nicht aus um alle Meldungen auszugeben, wird die letzte Meldung abgeschnitten. Das Längenfeld wird nicht korrigiert.

12.2 Aufbau der SORT-Versorgungstabellen

12.2.1 Tabellenübersicht

Die Tabellen und Bereiche, die intern für eine Versorgung von SORT als Unterprogramm angelegt werden, haben folgenden Aufbau:



12.2.2 Versorgungsblock SVB

Der Versorgungsblock SVB, auf den Register 1 verweist, besteht aus einer Adressliste variabler Länge mit fester Adresslistenzuordnung. Die Adressliste muss auf Wortgrenze ausgerichtet sein. Das erste Wort enthält eine Adresse, die auf den Übergabebereich verweist. Die restlichen Wortadressen verweisen auf die einzelnen Anweisungen (Ebene 1), wobei die Reihenfolge zu beachten ist. Nicht vorhandene Anweisungen sind durch eine Nulladresse zu kennzeichnen. Nulladressen am Ende des SVB können weggelassen werden. Das höchstwertige Bit der letzten Verweisadresse ist dann auf 1 zu setzen (COBOL-Konvention).

Aufbau des Versorgungsblocks SVB

- Byte 0 - 3: UEBA
Das 1. Wort des Versorgungsblocks SVB enthält die Adresse des Übergabebereichs.
- Byte 4 - 7: SRTA
Adresse der SORT-RECORDS/MERGE-RECORDS-Anweisung
- Byte 8 - 11: RECA
Adresse der SET-RECORD-ATTRIBUTES-Anweisung
- Byte 12 - 15: MODA
Adresse der ASSIGN-EXITS-Anweisung
- Byte 16 - 19: ALOA
Adresse der ASSIGN-RESOURCES-Anweisung
- Byte 20 - 23: OPTA
Adresse der SET-SORT-OPTIONS-Anweisung
- Byte 24 - 27: NCSA
Adresse der MODIFY-CODE-Anweisung
- Byte 28 - 31: SUMA
Adresse der SUM-RECORDS-Anweisung
- Byte 32 - 35: INOA
Adresse der SELECT-INPUT-RECORDS-Anweisung

Byte 36 - 39: Alternativ eine der folgenden Angaben:

- FCBA
Adresse der FCB-Verweistabelle.
Diese Adresse ist nur bei Ebene 2 möglich. Sie entfällt bei PARMOD=31.
oder
- FILA
Adresse der ASSIGN-FILES-Anweisung

Byte 40 - 43: NAMA

Adresse der ADD-SYMBOLIC-NAMES-Anweisung

Byte 44 - 47 MSDA

Adresse der MODIFY-SORT-DEFAULTS-Anweisung

Byte 48 - 51 SHOA

Adresse der SHOW-SORT-DEFAULTS-Anweisung

Hinweise

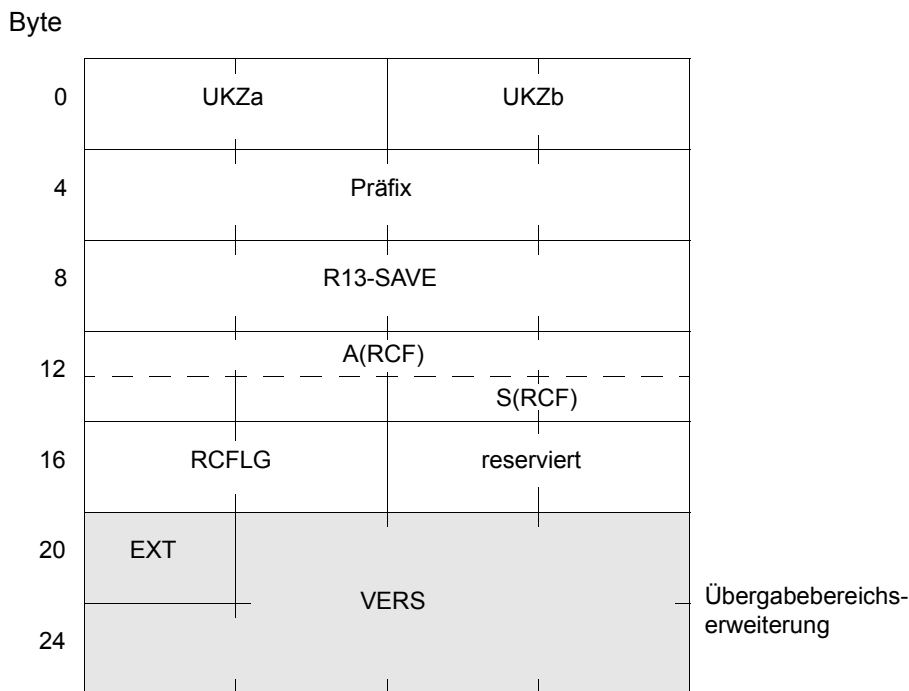
- Bei Versorgung über Ebene 0 besteht der SVB nur aus der Adresse des Übergabebereichs (UEBA). Bei Ebene 1 muss der SVB mindestens aus den Verweisadressen des Übergabebereichs (UEBA) und der SORT-RECORDS/MERGE-RECORDS-Anweisung bestehen. Diese beiden Bereiche müssen in jedem SVB angegeben werden.
- Die Anweisungen bei Ebene 1 sind Sätze variabler Länge, denen ein 4 Byte großes Satzlängenfeld vorangestellt ist. Im ersten Halbwort des Satzlängenfeldes steht die Satzlänge.
- Den Versorgungsblock SVB und den Übergabebereich kann der Benutzer mit den Makroaufrufen SRT0 und SRT1 erstellen.

12.2.3 Übergabebereich

Der Übergabebereich, auf den das erste Wort des SVB verweist, hat eine Länge von maximal 7 Worten und enthält folgende Informationen:

- Das Übergabekennzeichen.
Es legt u.a. fest, welche Übergabeebene verwendet wird (Ebene 0 oder 1).
- Ein 4 Zeichen langes Präfix, das den SORT-Lauf kennzeichnet.
- Den Sicherstellungsbereich für Register 13 bei Verwendung der SORT-Zugriffsmethode SORTZM.
- Adresse und Länge des RCF-Bereichs.
Im RCF-Bereich übergibt SORT das Rückkehrkennzeichen und SORT-Meldungen über den Ablauf (bis zur angegebenen Länge).
- Übergabebereichserweiterung:
Versionsnummer von SORT

Aufbau des Übergabebereichs



Die einzelnen Felder des Übergabebereichs haben folgende Bedeutung:

- Byte 0 - 1: UKZa
Diese Bytes enthalten die Informationen zum Operanden MULTI in SRT0 und SRT1
- X'...0' MULTI = STD
 - X'...1' MULTI = OPT
 - X'...2' MULTI = NOIMON
- Byte 2 - 3: UKZb
Diese Bytes enthalten das Übergabekennzeichen, das SORT mitteilt, über welche Ebene und in welcher Form die SORT-Anweisungen übergeben werden. Folgende Werte können in UKZ angegeben werden (angegeben sind jeweils die Wertigkeiten):
- X'...0' Die SORT-Anweisungen werden über SYSDTA übergeben (Ebene 0).
 - X'...4 Die SORT-Anweisungen werden im Hauptspeicher des rufenden Programms übergeben (Ebene 1). Register 1 verweist auf den SVB.
 - X'...8' Die SORT-Anweisungen werden als Anweisungstabellen im Hauptspeicher des rufenden Programms übergeben (Ebene 2). Das Register 1 verweist auf den SVB.
 - X'..0.' STXIT-Eingänge werden über alle Ebenen von SORT benutzt.
 - X'..2.' STXIT-Eingänge werden über alle Ebenen von SORT nicht benutzt. Es entfallen die speziellen DUMPs in Fehlerfällen und die Möglichkeit, mit dem Kommando SEND-MSG zuzugreifen.
 - X'..4.' Übergabebereichserweiterung vorhanden.
 - X'..8.' Die Anweisungen werden im SDF-Format erwartet.
 - X'.0.' Es wird kein RCF-Bereich eingerichtet. Kein Präfix. Sicherstellungsbereich für Register 13.
 - X'.2.' Adresse des RCF-Bereichs ist im A-Adressenformat angegeben.
 - X'.3.' Adresse des RCF-Bereichs ist im S-Adressenformat angegeben.
 - X'.4.' Im R13-SAVE wird von den Makros der SORTZM das Register 13 gesichert (falls RDONLY=NO).
 - X'.8.' Präfix vorhanden.
 - X'0...' SORT-Meldungen werden auf SYSOUT ausgegeben.
 - X'1...' SORT-Meldungen werden auf SYSLST ausgegeben.
 - X'2...' SORT-Meldungen werden nach SYSOUT und SYSLST ausgegeben.

X'3...' SORT-Meldungen werden nicht nach SYSOUT und/oder SYSLST ausgegeben, sondern nur in den RCF-Bereich.

Hinweis

Die Wertigkeiten dürfen über ODER-Verknüpfung sinnvoll kombiniert werden.

Beispiel

X'2284' bedeutet:
 Übergabe über Ebene 1. Register 1 verweist auf SVB.
 Anweisungen werden im SDF-Format erwartet.
 Adresse des RCF-Bereichs im A-Adressenformat.
 Ausgabe der Meldungen nach SYSOUT und SYSLST.

- Byte 4 - 7: Präfix
 Enthält ein 4 Zeichen langes Präfix, das den SORT-Lauf kennzeichnet.
- Byte 8 - 12: R13-SAVE
 Sicherstellungsbereich für Register 13 bei Verwendung der SORT-Zugriffsmethode SORTZM (falls RDONLY=NO).
- Byte 12 - 15: A(RCF)
 Adresse des RCF-Bereichs im A-Adressenformat.
- Byte 14 - 15: S(RCF)
 Adresse des RCF-Bereichs im S-Adressenformat.
- Byte 16 - 17: RCFLG
 Länge des RCF-Bereichs.
- Byte 18 - 19: reserviert
- Byte 20: EXT
 Übergabebereichserweiterung
 X'8.' SORT-Version erwartet
- Byte 21- 27: VERS
 SORT-Version in der Form nn.nann

12.2.4 SORT-Anweisungstabellen

Aus Kompatibilitätsgründen werden Anweisungstabellen (in ISP über Ebene 2) weiterhin unterstützt.

12.3 Sortiertabelle UTF-16

Die folgende Tabelle zeigt alle zur Zeit unterstützten Codes des Unicode UTF-16. Sie sind in der Reihenfolge aufgelistet, in der sie von SORT geordnet werden.

Die Spalten haben folgende Bedeutung:

hex		hexadezimale Verschlüsselung
Zeichen		druckbare Form des Zeichens oder seine Bedeutung
var	x	variables Sortierelement (Collation-Element)
	(x)	Zeichen zu dem ein variables Sortierelement (Collation-Element) gehört
multi	x	das Zeichen besteht aus mehreren Grundzeichen (keinen diakritischen Zeichen)
Ebene 1	=	das Zeichen hat auf Ebene 1 den gleichen Wert wie das Zeichen in der vorhergehenden Zeile
	>	das Zeichen hat auf Ebene 1 einen höheren Wert als alle vorhergehenden Zeichen
Ebene 2	=	das Zeichen hat auf Ebene 2 den gleichen Wert wie das Zeichen in der vorhergehenden Zeile
	>	das Zeichen hat auf Ebene 2 einen höheren Wert wie das Zeichen in der vorhergehenden Zeile
Ebene 3	=	das Zeichen hat auf Ebene 3 den gleichen Wert wie das Zeichen in der vorhergehenden Zeile
	>	das Zeichen hat auf Ebene 3 einen höheren Wert wie das Zeichen in der vorhergehenden Zeile

hex	Zeichen	var	multi	Ebene		
				1	2	3
0301	´					
0300	`			=	>	
0306	˘			=	>	
0302	^			=	>	
030C	˘			=	>	
030A	°			=	>	
0308	¨			=	>	
030B	˝			=	>	
0303	˜ COMBINING TILDE			=	>	
0307	·			=	>	
0327	¸			=	>	
0328	˙			=	>	
0304	ˉ			=	>	
0311	ˆ			=	>	
031B	’			=	>	
0323	ˆ			=	>	
0009	HORIZONTAL TABULATION*	X		>		
000A	LINE FEED*	X		>		
000B	VERTICAL TABULATION*	X		>		
000C	FORM FEED*	X		>		
000D	CARRIAGE RETURN*	X		>		
0085	NEXT LINE*	X		>		
0020	SPACE*	X		>		
00A0	NO-BREAK SPACE*	X		=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0060	`	x		>		
0384	'	x		>		
00B4	´	x		=	=	=
005E	^	x		>		
00AF	—	x		>		
02D8	˘	x		>		
02D9	˙	x		>		
00A8	¨	x		>		
0385	˝	x		=	>	
02DD	˚	x		>		
00B8	˘	x		>		
02DB	˙	x		>		
005F	—	x		>		
00AD	-	x		>		
002D	-	x		>		
2015	—	x		>		
002C	,	x		>		
003B	;	x		>		
003A	:	x		>		
0021	!	x		>		
00A1	ı	x		>		
003F	?	x		>		
00BF	¿	x		>		
002E	.	x		>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00B7	·	x		>		
0027	'	x		>		
2018	‘	x		>		
2019	’	x		>		
0022	"	x		>		
00AB	«	x		>		
00BB	»	x		>		
0028	(x		>		
0029)	x		>		
005B	[x		>		
005D]	x		>		
007B	{	x		>		
007D	}	x		>		
00A7	§	x		>		
00B6	¶	x		>		
00A9	©	x		>		
00AE	®	x		>		
0040	@	x		>		
002A	*	x		>		
002F	/	x		>		
2044	/	x		>		
005C	\	x		>		
0026	&	x		>		
0023	#	x		>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0025	%	x		>		
02C7	˘	x		>		
00B0	°	x		>		
002B	+	x		>		
00B1	±	x		>		
00F7	÷	x		>		
00D7	×	x		>		
003C	<	x		>		
003D	=	x		>		
003E	>	x		>		
00AC	¬	x		>		
007C		x		>		
00A6	¡	x		>		
007E	~	x		>		
2264	≤	x		>		
00A4	¤			>		
00A2	¢			>		
0024	\$			>		
00A3	£			>		
00A5	¥			>		
20AC	€			>		
20AF	ƒ	x		>		
0030	0			>		
0031	1			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00B9	¹			=	=	>
00BD	½	(x)		=	=	>
00BC	¼	(x)		>		
0032	2			>		
00B2	²			=	=	>
0033	3			>		
00B3	³			=	=	>
00BE	¾	(x)		=	=	>
0034	4			>		
0035	5			>		
0036	6			>		
0037	7			>		
0038	8			>		
0039	9			>		
0061	a			>		
0041	A			=	=	>
00AA	ª			=	=	>
00E1	á			=	>	
00C1	Á			=	=	>
00E0	à			=	>	
00C0	À			=	=	>
0103	ă			=	>	
0102	Ă			=	=	>
00E2	â			=	>	

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00C2	Â			=	=	>
1EAB	ã			=	>	
1EAA	Ã			=	=	>
01CE	ă			=	>	
01CD	Ă			=	=	>
00E5	å			=	>	
00C5	Å			=	=	>
00E4	ä			=	>	
00C4	Ä			=	=	>
00E3	ã			=	>	
00C3	Ã			=	=	>
0105	ą			=	>	
0104	Ą			=	=	>
0101	ā			=	>	
0100	Ā			=	=	>
1EA1	ạ			=	>	
1EA0	Ạ			=	=	>
00E6	æ			>		
00C6	Æ			=	=	>
0062	b			>		
0042	B			=	=	>
0063	c			>		
0043	C			=	=	>
0107	ć			=	>	

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0106	Ć			=	=	>
0109	ĉ			=	>	
0108	Ĉ			=	=	>
010D	č			=	>	
010C	Č			=	=	>
010B	ċ			=	>	
010A	Ā			=	=	>
00E7	ç			=	>	
00C7	Ç			=	=	>
0064	d			>		
0044	D			=	=	>
010F	ď			=	>	
010E	Ď			=	=	>
0111	đ			>		
0110	Đ			=	=	>
00F0	đ			>		
00D0	Đ			=	=	>
0065	e			>		
0045	E			=	=	>
00E9	é			=	>	
00C9	É			=	=	>
00E8	è			=	>	
00C8	È			=	=	>
0114	Ě			=	>	

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00EA	ê			=	>	
00CA	Ê			=	=	>
1EC5	ẽ			=	>	
1EC4	Ě			=	=	>
011B	ě			=	>	
011A	Ě			=	=	>
00EB	ë			=	>	
00CB	Ë			=	=	>
1EBD	ẽ			=	>	
1EBC	Ě			=	=	>
0117	è			=	>	
0116	Ê			=	=	>
0119	ę			=	>	
0118	Ę			=	=	>
0113	ē			=	>	
0112	Ě			=	=	>
0066	f			>		
0046	F			=	=	>
0067	g			>		
0047	G			=	=	>
01F5	ǵ			=	>	
01F4	Ǵ			=	=	>
011F	ǧ			=	>	
011E	Ǵ			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
011D	ĝ			=	>	
011C	Ĝ			=	=	>
01E7	ǧ			=	>	
01E6	Ǧ			=	=	>
0121	ḡ			=	>	
0120	Ḡ			=	=	>
0123	ḡ			=	>	
0122	Ḡ			=	=	>
1E21	ḡ			=	>	
1E20	Ḡ			=	=	>
0068	h			>		
0048	H			=	=	>
0125	ĥ			=	>	
0124	Ĥ			=	=	>
1E25	ḥ			=	>	
1E24	Ḥ			=	=	>
0127	ħ			>		
0126	Ĥ			=	=	>
0069	i			>		
0049	I			=	=	>
00ED	í			=	>	
00CD	Í			=	=	>
00EC	ì			=	>	
00CC	Ì			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
012D	ÿ			=	>	
012C	ÿ̇			=	=	>
00EE	î			=	>	
00CE	î̇			=	=	>
01D0	ÿ̈			=	>	
01CF	ÿ̈̇			=	=	>
00EF	ï			=	>	
00CF	ï̇			=	=	>
0129	ĩ			=	>	
0128	ĩ̇			=	=	>
0130	ı̇			=	>	
012F	ı̈			=	>	
012E	ı̈̇			=	=	>
012B	ı̇			=	>	
012A	ı̇̇			=	=	>
1ECB	ı̈̈			=	>	
1ECA	ı̈̈̇			=	=	>
0131	ı̈̈̈			>		
006A	j			>		
004A	J			=	=	>
0135	ĵ			=	>	
0134	Ĵ			=	=	>
006B	k			>		
004B	K			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
1E31	í			=	>	
1E30	Í			=	=	>
0137	ķ			=	>	
0136	Ķ			=	=	>
006C	l			>		
004C	L			=	=	>
013A	í			=	>	
0139	Í			=	=	>
013E	ř			=	>	
013D	Ř			=	=	>
013C	ĵ			=	>	
013B	Ĵ			=	=	>
0142	ł			>		
0141	Ł			=	=	>
006D	m			>		
004D	M			=	=	>
006E	n			>		
004E	N			=	=	>
0144	ń			=	>	
0143	Ń			=	=	>
0148	ň			=	>	
0147	Ň			=	=	>
00F1	ň			=	>	
00D1	Ň			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
1E45	ñ			=	>	
1E44	Ñ			=	=	>
0146	ŋ			=	>	
0145	Ŋ			=	=	>
2116	№ Zeichen entspricht No (hex: 004E 0366)		x	=	>	
0272	'			>		
014B	ŋ			>		
014A	Ɔ			=	=	>
006F	o			>		
0366	COMBINING LATIN SMALL LETTER O*			=	=	>
004F	Ō			=	=	>
00BA	º			=	=	>
00F3	ó			=	>	
00D3	Ó			=	=	>
00F2	ò			=	>	
00D2	Ò			=	=	>
014F	ř			=	>	
014E	Ř			=	=	>
00F4	ô			=	>	
00D4	Ô			=	=	>
1ED7	ř			=	>	
1ED6	Ř			=	=	>
01D2	ř			=	>	

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
01D1	Ö			=	=	>
00F6	ö			=	>	
00D6	Ö			=	=	>
0151	ö			=	>	
0150	Ö			=	=	>
00F5	ö			=	>	
00D5	Ö			=	=	>
014D	ö			=	>	
014C	Ö			=	=	>
01A1	ø			=	>	
01A0	Œ			=	=	>
1ECD	ø			=	>	
1ECC	Œ			=	=	>
0153	œ Zeichen entspricht oe (hex: 006F 0065)		x	=	>	
0152	Œ Zeichen entspricht OE (hex: 004F 0045)		x	=	=	>
00F8	ø			>		
00D8	Ø			=	=	>
0070	p			>		
0050	P			=	=	>
0071	q			>		
0051	Q			=	=	>
0138	κ			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0072	r			>		
0052	R			=	=	>
0155	ř			=	>	
0154	Ř			=	=	>
0159	ṛ̌			=	>	
0158	Ṛ̌			=	=	>
0157	ŗ			=	>	
0156	Ŕ			=	=	>
0213	ř̂			=	>	
0212	Ř̂			=	=	>
0073	s			>		
0053	S			=	=	>
015B	ś			=	>	
015A	Ś			=	=	>
015D	ŝ			=	>	
015C	Ŝ			=	=	>
0161	š			=	>	
0160	Š			=	=	>
1E61	ṣ́			=	>	
1E60	Ṣ́			=	=	>
015F	ș			=	>	
015E	Ș			=	=	>
1E63	ș̣			=	>	
1E62	Ș̣			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00DF	ß Zeichen entspricht ss (hex: 0073 0073)		x	=	>	
0074	t			>		
0054	T			=	=	>
0165	ť			=	>	
0164	Ť			=	=	>
0163	ṭ			=	>	
0162	Ṭ			=	=	>
0167	ṫ			>		
0166	Ṧ			=	=	>
0075	u			>		
0055	U			=	=	>
00FA	ú			=	>	
00DA	Ú			=	=	>
00F9	ù			=	>	
00D9	Ù			=	=	>
016D	ů			=	>	
016C	Ů			=	=	>
00FB	û			=	>	
00DB	Û			=	=	>
01D4	ů			=	>	
01D3	Ů			=	=	>
016F	ù			=	>	
016E	Û			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
00FC	ü			=	>	
00DC	Ü			=	=	>
0171	ú			=	>	
0170	Ů			=	=	>
0169	ũ			=	>	
0168	Ů			=	=	>
0173	ұ			=	>	
0172	Ҫ			=	=	>
016B	ū			=	>	
016A	Ū			=	=	>
01B0	Ƶ			=	>	
01AF	ƶ			=	=	>
1EE5	ұ			=	>	
1EE4	Ҫ			=	=	>
0076	v			>		
0056	V			=	=	>
0077	w			>		
0057	W			=	=	>
0175	ŵ			=	>	
0174	Ŵ			=	=	>
1E85	Ẅ			=	>	
1E84	Ẃ			=	=	>
0078	x			>		
0058	X			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0079	y			>		
0059	Ÿ			=	=	>
00FD	ý			=	>	
00DD	Ỳ			=	=	>
1EF3	ỳ			=	>	
1EF2	Ỳ			=	=	>
0177	ỳ			=	>	
0176	Ỳ			=	=	>
00FF	ÿ			=	>	
0178	Ỳ			=	=	>
1EF9	ÿ			=	>	
1EF8	Ỳ			=	=	>
1E8F	ỳ			=	>	
1E8E	Ỳ			=	=	>
007A	z			>		
005A	Z			=	=	>
017A	ž			=	>	
0179	Ž			=	=	>
1E91	ž			=	>	
1E90	Ž			=	=	>
017E	ž			=	>	
017D	Ž			=	=	>
017C	ž			=	>	
017B	Ž			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
1E93	Ʒ			=	>	
1E92	Ʒ̇			=	=	>
00FE	Ɓ			>		
00DE	Ɓ̇			=	=	>
03B1	α			>		
0391	Α			=	=	>
03AC	ά			=	>	
0386	Α̇			=	=	>
03B2	β			>		
0392	Β			=	=	>
03B3	γ			>		
0393	Γ			=	=	>
03B4	δ			>		
0394	Δ			=	=	>
03B5	ε			>		
0395	Ε			=	=	>
03AD	έ			=	>	
0388	Έ			=	=	>
03B6	ζ			>		
0396	Ζ			=	=	>
03B7	η			>		
0397	Η			=	=	>
03AE	ή			=	>	
0389	Ή			=	=	>

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
03B8	θ			>		
0398	Θ			=	=	>
039B	∧			>		
037A	.			=	>	
0399	ı			=	>	
03AF	í			=	>	
038A	ı			=	=	>
03CA	ï			=	>	
03AA	ï			=	=	>
0390	ï			=	>	
03BA	κ			>		
039A	Κ			=	=	>
03BB	λ			>		
039B	Λ			=	=	>
03BC	μ			>		
00B5	μ (MICRO SIGN)			=	=	>
039C	Μ			=	=	>
03BD	ν			>		
039D	Ν			=	=	>
03BE	ξ			>		
039E	Ξ			=	=	>
03BF	ο			>		
039F	Ο			=	=	>
03CC	ό			=	>	

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
038C	Ό			=	=	>
03C0	π			>		
03A0	Π			=	=	>
03C1	ρ			>		
03A1	Ρ			=	=	>
03C3	σ			>		
03A3	Σ			=	=	>
03C2	ς			=	=	>
03C4	τ			>		
03A4	Τ			=	=	>
03C5	υ			>		
03A5	Υ			=	=	>
03CD	ύ			=	>	
038E	Ϛ			=	=	>
03CB	Ϛ			=	>	
03AB	Ϛ			=	=	>
03B0	Ϛ			=	>	
03C6	φ			>		
03A6	Φ			=	=	>
03C7	χ			>		
03A7	Χ			=	=	>
03C8	ψ			>		
03A8	Ψ			=	=	>
03C9	ω			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
03A9	Ω			=	=	>
03CE	ώ			=	>	
038F	Ω			=	=	>
0430	а			>		
0410	А			=	=	>
0431	б			>		
0411	Б			=	=	>
0432	в			>		
0412	В			=	=	>
0433	г			>		
0413	Г			=	=	>
0434	д			>		
0414	Д			=	=	>
0452	ђ			>		
0402	Ђ			=	=	>
0453	ѓ			>		
0403	Ѓ			=	=	>
0435	е			>		
0415	Е			=	=	>
0451	ë			=	>	
0401	Ë			=	=	>
0454	ё			>		
0404	Ё			=	=	>
0436	ж			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0416	Ж			=	=	>
0437	з			>		
0417	З			=	=	>
0455	s			>		
0405	S			=	=	>
0438	и			>		
0418	И			=	=	>
0456	і			>		
0406	I			=	=	>
0457	ї			>		
0407	Ї			=	=	>
0439	й			>		
0419	Й			=	=	>
0458	j			>		
0408	J			=	=	>
043A	к			>		
041A	К			=	=	>
043B	л			>		
041B	Л			=	=	>
0459	лъ			>		
0409	Љ			=	=	>
043C	м			>		
041C	М			=	=	>
043D	н			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
041D	Н			=	=	>
045A	њ			>		
040A	Њ			=	=	>
043E	о			>		
041E	О			=	=	>
043F	п			>		
041F	П			=	=	>
0440	р			>		
0420	Р			=	=	>
0441	с			>		
0421	С			=	=	>
0442	т			>		
0422	Т			=	=	>
045B	ћ			>		
040B	Ћ			=	=	>
045C	ќ			>		
040C	Ќ			=	=	>
0443	у			>		
0423	У			=	=	>
045E	ў			>		
040E	Ў			=	=	>
0444	ф			>		
0424	Ф			=	=	>
0445	х			>		

* Zeichen nicht darstellbar

hex	Zeichen	var	multi	Ebene		
				1	2	3
0425	X			=	=	>
0446	ц			>		
0426	Ц			=	=	>
0447	ч			>		
0427	Ч			=	=	>
045F	џ			>		
040F	Џ			=	=	>
0448	ш			>		
0428	Ш			=	=	>
0449	щ			>		
0429	Щ			=	=	>
044A	ъ			>		
042A	Ъ			=	=	>
044B	ы			>		
042B	Ы			=	=	>
044C	ь			>		
042C	Ь			=	=	>
044D	э			>		
042D	Э			=	=	>
044E	ю			>		
042E	Ю			=	=	>
044F	я			>		
042F	Я			=	=	>

* Zeichen nicht darstellbar

Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie in auch gedruckter Form bestellen.

- [1] **BS2000/OSD-BC
Kommandos**
Benutzerhandbuch
- [2] **BS2000/OSD-BC
Einführung in das DVS**
Benutzerhandbuch
- [3] **BS2000/OSD-BC
DVS-Makros**
Benutzerhandbuch
- [4] **Assemblerbefehle (BS2000)**
Sprachbeschreibung
- [5] **ASSEMBH (BS2000)**
Beschreibung
- [6] **BS2000/OSD-BC
Makroaufrufe an den Ablaufteil**
Benutzerhandbuch
- [7] **SDF (BS2000)**
Dialogschnittstelle SDF
Benutzerhandbuch
- [8] **JV (BS2000)**
Jobvariablen
Benutzerhandbuch
- [9] **BS2000/OSD-BC
Einführung in die Systembetreuung**
Benutzerhandbuch

- [10] **XHCS** (BS2000)
8-bit-Code- und Unicode-Unterstützung im BS2000
Benutzerhandbuch
- [11] **IMON** (BS2000)
Installationsmonitor
Benutzerhandbuch
- [12] **SORT** (BS2000)
Taschenbuch
- [13] **SDF-P** (BS2000)
Programmieren in der Kommandosprache
Benutzerhandbuch
- [14] **POSIX** (BS2000)
Grundlagen für Anwender und Systemverwalter
Benutzerhandbuch
- [15] **POSIX** (BS2000)
Kommandos
Benutzerhandbuch
- [16] **Unicode im BS2000**
Übersichtshandbuch

Stichwörter

0, Ebene

SORT-Aufruf über 127, 213, 214, 216, 220, 363, 365

SORT-Aufruf über, Beispiel 363

1, Ebene

SORT-Aufruf über 127, 128, 213, 215, 219, 220, 367, 412, 416

SORT-Aufruf über, Beispiel 367

24-Bit-Adressierung 80

Benutzerausgang, Hinweis 354

31-Bit-Adressierung 80

Benutzerausgang, Hinweis 354

A

Achtung

Auswahlsortieren von Eingabesätzen mit variabler Länge 27

doppelter ISAM-Schlüssel = Standard 104

ISAM-Dateien nur aufsteigend sortieren 314

ISAM-Schlüssel sortierter Datei verändert 315

REMAINDER verlangt SORT-TYPE=* COMPOUND-RECORD 54

Satzlängengebiet als Sortierfeld 41

Überlauf bei SUM-RECORDS verhindern 331

ACS 121

ADD-FILE-LINK 21, 77, 92, 93, 95, 97, 98, 100, 101, 103, 105, 271, 275

PREMOUNT-LIST=0 98

ADD-SYMBOLIC-NAMES 37, 69, 150, 159, 160, 184, 187, 188, 194, 202, 413

Kurzbeschreibung 129

Operandenbeschreibung 130

Adressfeld 31

Adressierung

24-Bit 80

31-Bit 80

Adresslisten sortieren

Beispiel 343

Beschreibung 24

immer Sätze fester Länge 29

Adresslistenverweise vergl. TAG-HEADER 31

Adressverweissätze 30, 34

Alias Catalog Service 121

AMODE=24, Assembleroption 80

Ändern, Datei, Merkmale, Standardvorgaben 335, 340

Ändern, Format der Ausgabe durch Aufbereitungsmasken 25

Ändern, Satzformat 76 bei Benutzerausgängen 78

bei definierten Benutzerausgängen 77 mittels SET-RECORD- ATTRIBUTES-Anweisung 77

Ändern, Satzlänge 76 bei definierten Benutzerausgängen 77 mittels SET-RECORD- ATTRIBUTES-Anweisung 77

Ändern, Sortierreihenfolge FORMAT=*EBCDIC-DIN, Beispiel 322 FORMAT=*MODIFY-CODE, Beispiel 324

Ändern, Voreinstellungen 154

Angaben Version 196, 204

Anweisungen

Kurzübersicht 129

Arbeitsdateien, Eigenschaften 107

ASCII

erweiterter in EBCDIC, Umsetztabelle 47
PHYSICAL-TRANSLATE 355

ASCII-EBCDIC, Umsetztabelle 46

Assembler

ED-Aufbereitungsmaske 27, 60
Konventionen für Makroaufrufe 223
Option AMODE=24 80
Option PARMOD=24 80
Option RMODE=24 80
Schnittstelle für Adressfeld 34
vergl. Makros, Makroaufrufe 293

ASSIGN-EXITS 20, 44, 45, 76, 238, 242, 243,
248, 262, 263, 267, 271, 361, 412

Ausnahme 239

Beispiel 351

Kurzbeschreibung 129

Operandenbeschreibung 134

OUTPUT 351

PHYSICAL-TRANSLATE 355

ASSIGN-FILES 20, 23, 92, 95, 96, 107, 116, 299,
302, 321, 338, 351, 413

Beispiel 294

Kurzbeschreibung 129

Operandenbeschreibung 141

ASSIGN-RESOURCES 107, 112, 155, 272, 273,
275, 279, 412

CORE-Operand 272

Kurzbeschreibung 129

Operandenbeschreibung 144

Aufbereitungsmasken 27, 39, 60, 69, 79, 133,
185, 186, 187, 188, 193, 194

maximale Länge 133

Name festlegen 133

nicht bei Vollsordieren 25

nicht zugelassen mit Bit-Angabe 183

Regeln 62

Steuerzeichen 62

Überschneidungen 61

Aufruf von SORT als Unterprogramm 210

über Ebene 0 214

über Ebene 1 215

Ausgabe-Format durch Aufbereitungsmasken
ändern 25

Ausgabedateien, Eigenschaften 101

Ausgänge, vergl. Benutzerausgänge 256, 258

Ausschlie 340

Austauschbarkeit 291

Auswahlkriterien, vergl. Sortierkriterien 18

Auswahlsortieren 27, 294, 295

Beispiel 335, 338, 340

Definition 24

Standard feste Satzlänge 335, 340

variable Datei, Beispiel 335, 340

B

Bandbehandlung, Multifile/Multivolume-Set 98

Bandhilfsdateien 112

Bedingung für Auswahl

SELECT-INPUT-RECORDS, Beispiel 331

Beendigung des Sortier-/Mischlaufs 117
fehlerhafte 118

Beispiel

Bedingung festlegen mit SELECT-INPUT-
RECORDS 331

Dateiinhalte 389

DIN vergl. EBCDIC, Unterschiede 322

EDF03DRV 327

FORMAT=*MODIFY-CODE 324

ISAM-Datei sortieren 314

Makro SRTCLSE 373

Makro SRTGET 373

Makro SRTOPEN 373

Makro SRTPUT 373

Mischen sortierter Dateien 345

SELECT-INPUT-RECORDS 331

Sortierfeld vergl. neuer ISAM-Schlüssel 314

SORTZM 372

SRT0-Makro 363

SRT1-Makro 367

Summenbildung 331

symbolische Feldbezeichnung 320

TRANSLATE-CHARACTER 327

Vollsordieren mehrerer Dateien 317

Vollsordieren, Eingabedatei =

Ausgabedatei 320

Beispieldateien, Inhalt 389

benutzen, SORT 293

- Benutzerausgang
 INPUT 347
 OUTPUT 351
 PHYSICAL-TRANSLATE, Beispiel 355
 VIRTUAL-TRANSLATE, Beispiel 359
- Benutzerausgänge
 Ändern des Satzformats bei Ausgängen 78
 Hinweis zu 24/31-Bit-Adressierung 354
 Kurzbeschreibung 237
 Registerkonventionen 240
- Berechnen
 Arbeitsdateien 108
 CORE-Wert 271
- Bibliotheken 91, 116, 134, 141, 143, 240, 347
- Bild
 Aufruf von SORT als Unterprogramm über Ebene 0 214
 Aufruf von SORT als Unterprogramm über Ebene 1 215
 Dateien des SORT 92
 Erstellen von Dateien mit Adressverweissätzen 30
 Mischen von Dateien 35
 Multitasksortieren 277
 Satzbearbeitung und Satzänderung im SORT 74
 Schnittstellen der Zugriffsmethode SORTZM 225
 Sortieren und mischen mit SORT 17
 Sortierfunktionen des Sortier-/Mischprogramms SORT 24
 Verarbeitung einer Datei mit Adressverweissätzen 34
 von SORT bearbeitete Dateien 92
- Bindemodulbibliothek 91, 94, 116
- C**
- CCS
 coded character set 81
 EDF03DRV 327
- CCSN, coded character set name 81
- CHECKPOINT-Operand
 MERGE-RECORDS-Anweisung 269
 SORT-RECORDS-Anweisung 269
- Code 81
 coded character set (CCS) 81
 coded character set name (CCSN) 81
 Codetabellen für Benutzerroutinen 261
 codierter Zeichensatz 38, 81
 COMPOUND-RECORD, Beispiel 335, 340
 CORE-Operand, ASSIGN-RESOURCES-Anweisung 272
- D**
- Dateibearbeitung durch EDT, ISAM-Schlüssel Probleme mit 315
- Dateien
 des SORT 92
 Eingabedatei = Ausgabedatei, Beispiel 320
 Endung 389
 größer 32 GByte 122
 Merkmale ändern 335, 340
 mischen, Beispiel 345
 SORT zuordnen 94
 variable Satzlänge - Berechnung der Position eines Datenbytes 309
 verändern durch Auswahl bei der Auswahl 76
 vollsortieren, mehrere, Beispiel 317
 zuweisen vergl. ADD-FILE-LINK 21
- Dateiinhalte
 Beispiel 389
- Dateikettungsname 92
 Ausgabedatei 101
 Eingabedateien 95
 SORTIN 311, 372, 377
 SORTOUT 294, 311, 372, 377
 SORTWK 107
 SORTWKEX 111
 SORTWKx 107
 SORTWKxx 110
- Dateimerkmale
 bei mehreren Eingabedateien 96
 der Ausgabedatei 101
- Definitionsfolge bei Sortier-/Mischläufen 21
- DIALOG-Angabe, Benutzerausgänge 239
- DIN vergl. EBCDIC, Unterschiede, Beispiel 322
- DOMINO-Phase 19

DUMMY-Dateien 91
DVS-Ausgänge 256, 258

E

EBCDIC

PHYSICAL-TRANSLATE 355
EBCDIC vergl. DIN, Unterschiede, Beispiel 322
EBCDIC-DIN-Format 44
EBCDIC-INTERNATIONAL-Format 44
EBCDIC-ISO-EBCDIC-Format 45
Ebene 0 127, 213, 214, 216, 220, 363, 365
Beispiel, SORT-Aufruf über 363
Ebene 1 127, 128, 213, 215, 219, 220, 367, 412, 416
Beispiel, SORT-Aufruf über 367

EDF03DRV

Beispiel 327
CCS 327

Einführung in SORT 293

Eingabe 335

Eingabe- und Vorsortierungsphase 19

Eingabedateien

für Mischläufe 99
für Sortierläufe 95

Einrichten

von Arbeitsdateien 107
von Hilfsdateien 110

END, Kurzbeschreibung 129

Endebehandlung der SORT-Dateien

Arbeitsdateien 109
Fixpunktdateien 115
Hilfsdateien 113
Übersicht 117

Endmisch- und Ausgabephase 19

Endung, Dateien 389

Eröffnung

Eingabedateien 97
Mischeingabedateien 100

Ersatzzeichen in Masken 60

erweitern, Summenfeld

Überlauf 331

erweiterter Zeichensatz 81

EXLST-Benutzerausgang 256, 258

EXLST-FOR-INPUT 237

EXLST-FOR-OUTPUT 237

extended host code support (XHCS) 81

EXTENDED-CHARACTER-Format 48, 327

EXTERNAL-COMPARE 238, 263

F

Fehlerbehandlung vergl. STXIT 216

fehlerhafte Beendigung des Sortier-/
Mischlaufs 118

fehlerhafte SORT-Beendigung 406

Fehlerinformation 406, 408

Feldbezeichnung, symbolische, Beispiel 320

FINISH-INPUT-Angabe, Benutzerausgänge 239

Fixpunkt 269

Fixpunktdatei 114

Endebehandlung 115

Fixpunktverarbeitung 269

Format

ändern der Ausgabe durch
Aufbereitungsmasken 25

bei Sortierfeldern 38

EXTENDED-CHARACTER 48, 327

MODIFY-CODE 45

TRANSLATE-CHARACTER 48, 327

UNICODE-CHARACTER 53

zugelassen für Vergleichsfelder 159

FORMAT=*MODIFY-CODE, Beispiel 324

Füllzeichen 60

G

Gerätebelegung, ADD-FILE-LINK 98

Größe von Arbeitsdateien 108

H

Haupttask 278

hexadezimal, Zeichenpaare, MODIFY-CODE,
Beispiel 324

Hilfsdateien 110

Hinweis, Syntax SDF-Anweisung SRT1-
Makro 370

I

IGNORE-LENGTH-FIELD 120, 174, 203

ILSORT-Einsprungstelle 210

- IMON 287
 Inhalt, Beispieldateien 389
 INPUT 237
 bei 24-Bit-Adressierung 246
 bei 31-Bit-Adressierung 243
 Installation 287
 INT 238, 267
 Interne Fehler des SORT 405
 Interne Mischphase 19
 ISAM-Ausgabedateien 103
 ISAM-Datei sortieren, Beispiel 314
 ISAM-Schlüssel 31, 103
 doppelter 104, 319
 Probleme mit, Dateibearbeitung durch
 EDT 315
 standard 104
 ISP-Anweisungen 129
- J**
- Jobvariable 407
- K**
- KEEP-INPUT-TAPE 118
 Koexistenz von Produktversionen 289
 Kommando
 SEND-MESSAGE 267
 Kommando-Returncode 206
 Konstantenfelder 37, 56
 Kriterien, vergl. Sortierkriterien 17
- L**
- LOCK-FILE-LINK 98
- M**
- Maincode 206
 Makro-Beispiel
 SRT1 367
 SRTCLSE 373
 SRTGET 373
 SRTOPEM 373
 SRTPUT 373
 Makroaufrufe
 für SORT 216
 für SORTZM 224
- Maskenfelder 37, 60
 erlaubte Formate 61
 Warnung bei Überschneidung 61
 Mehrfachsortieren mit SORTZM 226
 Beispiel 377, 386
 Meldungen 395
 Meldungsnummer 395
 Meldungspriorität 395
 MERGE-RECORDS 20, 36, 38, 79, 127, 132,
 269, 317, 345, 412, 413
 beste Verwendung 317
 Kurzbeschreibung 129
 mit symbolischen Namen 130
 Operandenbeschreibung 147
 MERGE, Mischen 35
 MERGEExx 92
 MIN-MSG-WEIGHT, SET-OPTIONS-
 Anweisung 395
 Mischen 24, 35
 Beispiel 345
 Definition 17
 mögliche Reihenfolge 18
 virtuelles 273
 Mischlauf
 Befehlsfolge 21
 Eingabedateien 99
 MODIFY-CODE 45, 153, 260, 304, 412
 Beispiel 324
 Format 45, 153
 Kurzbeschreibung 129
 Operandenbeschreibung 153
 MODIFY-SORT-DEFAULTS
 Kurzbeschreibung 129
 Operandenbeschreibung 154
 Modul, Aufruf aus Bibliothek 354
 Modulbibliothek 347
 Multifile/Multivolume-Set 98
 Multitasksortieren 277
- N**
- Nebentask 279
 normale Beendigung des Sortier-/Mischlaufs 117
 normale SORT-Beendigung 406

O

offenes System 119
Operand REMAINDER-EXPLICIT
 Restfelder 374
Optimierung von Sortierläufen 271
OPTIMIZATION-Operand, SET-SORT-OPTIONS-
 Anweisung 285
OUTPUT 237
 bei 24-Bit-Adressierung 250
 bei 31-Bit-Adressierung 248
 Benutzerausgang, Beispiel 351

P

PAM-Dateien, Arbeitsdateien 107
PAM-Schlüssel 32
PARAMETER-MODE 354
 Adressierung, Hinweis 354
 Beispiel 351
PARMOD=24, Assembleroption 80
PHYSICAL-TRANSLATE 238, 260
 ASCII 355
 Beispiel 355
 EBCDIC 355
 Format 45
PLANNING 237, 242, 271
Planungsphase 19
Plattenhilfsdateien 112
POSIX 119
 Ausgabedatei 106
 Eingabedatei 99
 Eingabedateien 96
POSIX-Datei 119
POSIX-Dateisystem 119
Prioritätsklassen 272
Produktversionen, Koexistenz 289
Programm erneut starten, RESTART-
 PROGRAM 270
Programminformation 408

R

RCF-Bereich 217, 220, 409
Readme-Datei 13
RECA, SET-RECORD-ATTRIBUTES 412

RECORDS-PER-CYCLE-Operand, MERGE-
 RECORDS-Anweisung 269
Registerkonventionen, Benutzerausgänge 240
Reihenfolge
 Ändern, FORMAT=EBCDIC-DIN,
 Beispiel 322
REMAINDER-EXPLICIT 54, 57, 72, 186, 187,
 336, 341, 374, 380
REMAINDER-SYMBOLIC, vergl. REMAINDER-
 EXPLICIT 54
RESTART-PROGRAM 270
Restfelder 37, 54
 Operand REMAINDER-EXPLICIT 54, 57,
 72, 186, 187, 336, 341, 374, 380
RMODE=24, Assembleroption 80
Rückkehrcode-Anzeige 408

S

Satzauswahl bei dem Einlesen 76
Sätze, Summieren 79
Satzformatänderung 76
Satzlänge, Berechnung variabler Sätze 301
Satzlänge, variable
 Sortierfeld bei 40
Satzlängenänderung 76
Satzlängenfeld
 als Sortierfeld 41, 120
 Berücksichtigung bei Sortieranweisung 120,
 159, 174, 203, 309
Schließen
 der Eingabedateien 98
 der Mischeingabedateien 100
SDF-Anweisungen 129
SELECT-INPUT-RECORDS 66, 76, 130, 132,
 133, 223, 334, 412
 Beispiel 331
 Kurzbeschreibung 129
 Operandenbeschreibung 158
 SRT1-Makro 369, 370
SEND-MESSAGE-Kommando 267
SET-RECORD-ATTRIBUTES 27, 40, 54, 73, 76,
 77, 101, 102, 148, 181, 201, 226, 380
 162
 Beispiel, Auswahlsortieren 335, 340

- Kurzbeschreibung 129
- RECA 412
- SET-SORT-OPTIONS 92, 93, 94, 97, 98, 117, 118, 154, 248, 251, 285, 395, 405, 412
 - Kurzbeschreibung 129
 - Operandenbeschreibung 169
 - SRT1-Makro 369, 380
- SET-TASKLIB 134
- SHOW-JV-Kommando 408
- SHOW-SORT-DEFAULTS
 - Kurzbeschreibung 129
 - Operandenbeschreibung 178
- SORT
 - als Unterprogramm 210
 - Aufruf über Ebene 0 127, 213, 214, 216, 220, 363, 365
 - Aufruf über Ebene 1 127, 128, 213, 215, 219, 220, 367, 412, 416
 - benutzen 293
 - Einführung 293
 - in XS-Umgebung 80
 - Kurzübersicht der Anweisungen 129
 - Makroaufrufe 216
 - starten 195
 - starten mit START-SORT 196
 - TU-Variante 195
 - Übergabe der Steuerinformation an SORT 213
 - und ACS (Hinweise) 121
- SORT-Anweisungstabellen 416
- SORT-Aufruf
 - als Unterprogramm 210
 - über Ebene 0, Beispiel 363
 - über Ebene 1, Beispiel 367
- SORT-FILE, Operandenbeschreibung 198
- SORT-RECORDS 20, 25, 27, 29, 34, 38, 54, 56, 76, 78, 79, 107, 127
 - CHECKPOINT 269
 - Kurzbeschreibung 129
 - Operandenbeschreibung 179
 - SRT1-Makro 224
 - symbolische Namen 132
- SORT-TYPE=*COMPOUND-RECORD, Beispiel 335, 340
- SORT-TYPE=TAG-TRAILER 343
- SORT, Zugriffsmethode SORTZM
 - Beispiel 377, 386
 - Makrobeschreibung 224
- SORTCKPT 22, 92, 143
- Sortierarten 24
- Sortierbegriff 38
- Sortieren
 - Adresslistensortieren 24
 - Auswahlsortieren 24
 - Definition der Kriterien 17
 - Haupttask 278
 - Mischen 24
 - Nebentask 279
 - Vollsortieren 24
 - Zyklensortieren 274
- Sortierfelder 37, 38
 - bei variabler Satzlänge 40
 - Satzlängenfeld 41, 120
 - Überschneidung 42
 - Umcodieren 43
 - umcodieren mit Benutzerausgang PHYSICAL-TRANSLATE 260
 - Umcodieren mit Benutzerausgang VIRTUAL-TRANSLATE 261
- Sortierkriterien 18, 24, 35, 99, 147, 293, 295, 296, 317, 321, 334, 377
 - für MERGE-RECORDS 345
- Sortierlauf
 - Befehlsfolge 21
 - Eingabedateien 95
 - Optimierung 271
- Sortierreihenfolge
 - Ändern, FORMAT=EBCDIC-DIN, Beispiel 322
 - Ändern, MODIFY-CODE, Beispiel 324
- SORTIN 92, 311, 372, 377
- SORTINxx 92
- SORTLIB 294
- SORTMODS 22, 91, 92, 94, 116, 143, 240
- SORTOUT 92, 294, 311, 372, 377
- SORTU-Einsprungstelle 210
- SORTWK 92, 107
- SORTWKEX 111

- SORTWKx 92, 107
 - SORTWKxx 92, 110
 - SORTZM
 - Beispiel 372, 377, 386
 - Makrobeschreibung 224
 - Mehrfachsortieren 226
 - Zugriffsmethode 372
 - Speicherplatz für Arbeitsdateien zuweisen 108
 - SRT0
 - Beispiel 363
 - Makro-Aufruf 216
 - vergl. auch Ebene 0 216
 - SRT1
 - Beispiel 367
 - Makro-Aufruf 216
 - Syntax 370
 - vergl. auch Ebene 1 216
 - SRTCLSE
 - Beispiel 373
 - Makro-Aufruf 224, 226, 227, 233
 - SRTGET
 - Beispiel 373
 - Makro-Aufruf 224, 226, 229, 231
 - SRTOPEN
 - Beispiel 373
 - Makro-Aufruf 224, 226, 227
 - SRTPUT
 - Beispiel 373
 - Makro-Aufruf 224, 226, 229
 - Standardwerte, ändern 154
 - START-SORT, Operandenbeschreibung 196
 - starten
 - Programm erneut, RESTART-PROGRAM 270
 - SORT-FILE 198
 - START-SORT 196
 - Steuerfelder 37
 - Steuerinformationen übergeben an SORT 213
 - STXIT 216, 219, 220, 267, 415, 416
 - Subcode1 (SC1) 206
 - Subcode2 (SC2) 206
 - Subsystem ACS 121
 - SUM-RECORDS 58, 73, 78, 79, 130, 132, 133, 171, 248, 284, 412
 - Beispiel 331
 - Kurzbeschreibung 129
 - Operandenbeschreibung 191
 - Summenbildung, Beispiel 331
 - Summenfeld 37, 58
 - erweitern vergl. Überlauf 331
 - Summierung von Sätzen 79
 - symbolische Feldbezeichnung, Beispiel 320
- ## T
- Tabelle
 - Code für Benutzerrountinen 261
 - DVS-Augänge, EXLST-Makro 256, 258
 - Endebehandlung der SORT-Dateien 117, 118
 - erweiterter ASCII in EBCDIC 47
 - Kurzübersicht der SORT-Anweisungen 129
 - PHYSICAL-TRANSLATE - Umsetzung 260
 - TRANSLATE-CHARACTER 51
 - Umsetzen ASCII-EBCDIC 46
 - Versorgung SORT als Unterprogramm 411
 - Zugelassene Formate für Vergleichsfelder 159
 - Tabellenübersicht 410
 - TAG-COMPOUND 31
 - TAG-HEADER 31
 - TAG-TRAILER 31
 - TERMINATE-ABNORMAL-Angabe, Benutzerausgänge 241
 - TRANSLATE-CHARACTER
 - Beispiel 327
 - Format 48, 327
 - Umsetzungstabellen 51
- ## U
- Übergabe der Steuerinformation an SORT 213
 - Übergabebereich 414
 - Übergabekennzeichen 414
 - Übergabesteuerfeld 216
 - Überlauf, Summenfeld erweitern 331
 - Überschneidung von Sortierfeldern 42

- Umcodieren von Sortierfeldern [43](#)
 - Benutzerausgang PHYSICAL-TRANSLATE [260](#)
 - Benutzerausgang VIRTUAL-TRANSLATE [261](#)
- Umsetztabelle
 - EBCDIC in erweiterter ASCII-Code [46](#)
 - erweiterter ASCII-Code in EBCDIC [47](#)
 - TRANSLATE-CHARACTER [51](#)
- Unicode [86](#)
- Unicode Default Collation Table [53](#)
- UNLOCK-FILE-LINK [98](#)
- V**
 - variable Satzlänge, Datenbyte-Position [309](#)
 - Vergleichsfelder [37, 66](#)
 - Vergleichskonstanten [37, 67](#)
 - Versions-Angabe [196, 204](#)
 - Versionsauswahl, Prioritäten [290](#)
 - Versorgungsblock SVB [215, 216, 412](#)
 - Verzeichnis, Beispieldateien [389](#)
 - VIRTUAL-TRANSLATE [238](#)
 - Benutzerausgang [359](#)
 - Format [44](#)
 - Umcodieren von Sortierfeldern [261](#)
 - Virtuelles Mischen [273](#)
 - Vollsortieren [24, 25](#)
 - Eingabedatei = Ausgabedatei, Beispiel [320](#)
 - mehrere Dateien, Beispiel [317](#)
 - Vorbereitungsphase [19](#)
- W**
 - Wiedergewinnungsadresse,
 - Adresslistensortieren [31](#)
 - WORK-FILE- OVERFLOW [237, 254](#)
- X**
 - XHCS, extended host code support [81](#)
 - XPG4 [119](#)
 - XS-Umgebung [80](#)
- Z**
 - Zeichen, hexadezimal, MODIFY-CODE,
 - Beispiel [324](#)
 - Zeichensatz
 - codierter [38, 81](#)
 - erweiterter [81](#)
 - Zugang zu POSIX [119](#)
 - Zugriffsmethode SORTZM
 - Beispiel [372](#)
 - Makrobeschreibung [224](#)
 - Zuordnung der SORT-Dateien [94](#)
 - Zustand des SORT-Laufs [407](#)
 - zuweisen
 - CORE [271](#)
 - Dateien vergl. ADD-FILE-LINK [21](#)
 - Speicherplatz für Arbeitsdateien [108](#)
 - Zyklensortieren [274](#)

