Deutsch

Fujitsu Software

BS2000 OS DX Dienstprogramme

Benutzerhandbuch

Stand der Beschreibung:
BS2000 V21.0C

Ausgabe Juni 2026

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 bs2000.info@fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

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: 2015 erfüllt.

Copyright und Handelsmarken

Copyright © 2026 Fujitsu

Alle Rechte vorbehalten.
Liefermöglichkeiten und technische Änderungen vorbehalten.

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

Inhaltsverzeichnis

Dienstprogramme 16
1 Einleitung 17
1.1 Zielsetzung und Zielgruppen des Handbuchs 19
1.2 Konzept des Handbuchs 20
1.3 Änderungen gegenüber dem Vorgänger-Handbuch 21
1.4 Darstellungsmittel 22
2 DPAGE Auflisten und Verändern von Plattendateien 24
2.1 Unterstützung von Pubsets 25
2.2 Programmlauf starten 26
2.3 Anweisungen 27
2.3.1 BKPT - DPAGE unterbrechen 28
2.3.2 DISPLAY - Seite nach SYSOUT ausgeben 29
2.3.3 EDT - EDT aufrufen 31
2.3.4 END / HALT - DPAGE beenden 32
2.3.5 MODIFY - Inhalt einer Seite verändern 33
2.3.6 OPEN - Datei oder Datenträger öffnen 35
2.3.7 PRINT - Seite ausdrucken 37
2.3.8 READ - Seite einlesen 38
2.3.9 WRITE - Internen Arbeitsbereich zurückschreiben 39
2.4 Meldungen von DPAGE 40
3 INIT Initialisieren von (emulierten) Magnetbändern 43
3.1 Betriebsarten 45
3.1.1 Normalbetrieb 46
3.1.2 Konsolbetrieb 47
3.2 Programmlauf 48
3.2.1 Programmstart 49
3.2.2 Initialisieren eines Magnetbandes (Beispiel) 50
3.2.3 Programmende 52
3.2.4 Probleme bei der Datenträgerbearbeitung 53
3.3 Anweisungen 54
3.3.1 INIT - Kennsätze schreiben 56
3.3.2 LIST - Magnetbandkennsätze anzeigen 62
3.3.3 OPTION - Programmfunktionen ein- und ausschalten 65
3.3.4 END - Beenden des Programmlaufs 69
3.3.5 HELP - Kurzbeschreibung der INIT-Anweisungen anzeigen 70
3.4 Aufbau der Kennsätze 71
3.4.1 Datenträgerkennsatz VOL1 für Magnetbänder 72

3.4.2 Dateikennsatz HDR1 für Magnetbänder 73
3.4.3 Dateikennsatz HDR2 für Magnetbänder 75
3.4.4 Dateikennsatz HDR3 für Magnetbänder 76
4 IORM Dynamische Steuerung von I/O-Ressourcen 77
4.1 IOPT: I/O-Prioritäten-Steuerung für Tasks 82
4.1.1 IOPT-Anweisungen 84
4.1.1.1 IOPT aktivieren und deaktivieren 85
4.1.1.2 Grenzwerte für I/O-Prioritätsklassen festlegen und abfragen 86
4.1.1.3 Plattengeräte für IOPT aktivieren und deaktivieren 87
4.1.1.4 Gerätegruppen 89
4.1.1.5 Grenzwerte für I/O-Priorität LOW festlegen 91
4.1.1.6 Grenzwerte für I/O-Priorität MEDIUM festlegen 96
4.1.1.7 Grenzwerte für I/O-Prioritäten abfragen 100
4.1.1.8 Auslastung abfragen 102
4.1.1.9 Prüfbetrieb 105
4.1.2 Einsatzbeispiele 109
4.2 DPAV: Monitoring für Alias-Geräte und Begrenzung der Alias-Anzahl für FastDPAV 110
4.2.1 Anweisungen 111
4.2.1.1 DPAV_CHK_ALI 112
4.2.1.2 DPAV_BAS_ADD/DPAV_BAS_REM 115
4.3 DDAL: Optimierte Lastverteilung im Betrieb von ETERNUS CS 117
4.4 IOLVM: Begrenzung der I/O-Leistungsaufnahme einzelner virtueller Maschinen 119
5 JMP Rekonstruktion von ENTER-Kommandos aus dem Jobpool des JMS . . . 120
5.1 Ablauf von JMP 121
5.2 Anweisungen 122
5.2.1 Übersicht über die Anweisungen von JMP 123
5.2.2 Beschreibung der Anweisungen 124
5.2.2.1 CONVERT-JOBPOOL - Jobpool-Datei in neueres oder älteres Format überführen 125
5.2.2.2 CREATE-PROCEDURE-FILE - SAM-Datei mit BS2000-Prozedur erzeugen . 127
5.2.2.3 END - Eingabe von Anweisungen beenden 128
5.2.2.4 OPEN-JOBPOOL-FILE - Jobpool-Datei öffnen 129
5.2.2.5 SHOW-JOBPOOL-STATUS - Informationen über den Jobpool abrufen . . 130
5.3 Hinweise zu den rekonstruierten Attributen 138
5.4 Meldungen von JMP 147
6 JMU Einrichten und Warten der Systemdatei SJMSFILE 148
6.1 Auftragsverwaltung 149
6.2 Ablauf von JMU 151

6.3 Anweisungen 153
6.3.1 Übersicht über die Anweisungen von JMU 154
6.3.2 Beschreibung der Anweisungen 155
6.3.2.1 CREATE-PROCEDURE-FILE - SAM-Datei erzeugen, die eine BS2000-Prozedur enthält 156
6.3.2.2 DEFINE-JOB-CLASS - Auftragsklassen-Definitionen in die Datei SJMSFILE schreiben 158
6.3.2.3 DEFINE-JOB-STREAM - Streamdefinitionen in die Datei SJMSFILE schreiben 168
6.3.2.4 DELETE-JOB-CLASS - Klassendefinitionen löschen 172
6.3.2.5 DELETE-JOB-STREAM - Streamdefinitionen löschen 173
6.3.2.6 END - Eingabe von Anweisungen beenden 174
6.3.2.7 GRANT-JOB-CLASS-ACCESS - Zugriff von Benutzerkennungen zu einer Auftragsklasse regeln 175
6.3.2.8 MODIFY-JOB-CLASS - Auftragsklassen-Definitionen ändern 176
6.3.2.9 MODIFY-JOB-STREAM - Streamdefinitionen ändern 179
6.3.2.10 REMOVE-USER - Zugang zu nichtöffentlichen Auftragsklassen sperren . . . 180
6.3.2.11 SET-JOB-CLASS-DEFAULT - Standardklassen für Benutzer spezifizieren . 181
6.3.2.12 SET-MODIFICATION-MODE - Änderungsmodus einstellen 182
6.3.2.13 SET-POSIX-JOB-CLASS-DEFAULT - POSIX-Standardklassen für Benutzer spezifizieren 183
6.3.2.14 SHOW-JOB-CLASS - Inhalt von Klassendefinitionen oder Namen von Klassen auflisten 184
6.3.2.15 SHOW-JOB-STREAM - Inhalt von Streamdefinitionen oder Namen von Streams auflisten 186
7 LMSCONV Erstellen und Verwalten von Bibliotheken 188
7.1 Bibliotheken 192
7.1.1 Logischer Aufbau einer Bibliothek 193
7.1.2 Ein- und Ausgabebibliotheken 194
7.1.3 Mehrfachzugriff auf Bibliotheken 195
7.2 Elemente 196
7.2.1 Mehrfachzugriff auf Elemente 197
7.2.2 Beschreibung der Elementtypen 198
7.2.3 Konvention für Elementbezeichnungen 200
7.2.4 Elementbezeichnung in den Anweisungen 201
7.2.5 Protokollierung der Elementbezeichnung 202
7.2.6 Auswahlangabe für Elementbezeichnungen 203
7.2.7 Konstruktionsangabe für Elementbezeichnungen 204
7.2.8 Elementattribute 206
7.2.9 Typabhängigkeiten 207

7.2.10 Versionsverwaltung 208
7.2.11 Datenschutz durch Überschreiben 210
7.2.12 Beweissicherung 211
7.2.13 Extended Host Code Support (XHCS) 212
7.3 Funktionen von LMSCONV 214
7.3.1 Starten von LMSCONV 215
7.3.2 Zuweisen von Bibliotheken 217
7.3.3 Bearbeiten von Elementen 218
7.3.4 Steuern des LMSCONV-Laufs 223
7.3.5 Platten ohne PAM-Schlüssel 226
7.3.6 NK4-Platten 230
7.3.7 Auswirkungen von ACS 232
7.4 Anweisungen 233
7.4.1 Übersicht über die Anweisungen von LMSCONV 234
7.4.2 LMSCONV-Anweisungen von ADD-ELEMENT bis MODIFY-ELEMENT 237
7.4.2.1 ADD-ELEMENT - Element in Bibliothek aufnehmen 238
7.4.2.2 CLOSE-LIBRARY - Bibliothek schließen 244
7.4.2.3 COPY-ELEMENT - Element kopieren 245
7.4.2.4 COPY-LIBRARY - Bibliothek kopieren 252
7.4.2.5 DELETE-ELEMENT - Ein Element logisch löschen 254
7.4.2.6 END - LMSCONV beenden 259
7.4.2.7 EXTRACT-ELEMENT - Elemente in Datei ausgeben 261
7.4.2.8 MODIFY-DEFAULTS - Voreinstellungen verändern 268
7.4.2.9 MODIFY-ELEMENT - Element verändern 284
7.4.3 Subanweisungen von MODIFY-ELEMENT für Elemente vom Basistyp R, C und L 291
7.4.3.1 ADD-REP-RECORD - REP-Sätze in Bindemodul einfügen 292
7.4.3.2 ADD-TEXT-MODIFICATION - Textsätze eines Bindemoduls korrigieren . 293
7.4.3.3 DELETE-RECORD-TYPE - Satzarten aus Eingabeelement auschließen 295
7.4.3.4 END-MODIFY - Eingabe von Subanweisungen beenden 296
7.4.3.5 MODIFY-CSECT-ATTRIBUTES - Programmabschnittsmerkmale verändern . 297
7.4.3.6 MODIFY-MODIFICATION-DEFAULTS - Globale Voreinstellungen festlegen 299
7.4.3.7 REMOVE-MODIFICATION - Korrekturen rückgängig machen 301
7.4.3.8 RENAME-SYMBOLS - Symbole umbenennen 302
7.4.4 Subanweisungen von MODIFY-ELEMENT für Textelemente 303
7.4.4.1 ADD-RECORD - Sätze einfügen 304
7.4.4.2 END-MODIFY - Subanweisungen abschließen 305
7.4.4.3 REMOVE-RECORD - Satz oder Satzbereich in Element löschen 306
7.4.5 LMSCONV-Anweisungen von MODIFY-ELEMENT-ATTRIBUTES bis WRITE-COMMENT 307

7.4.5.1 MODIFY-ELEMENT-ATTRIBUTES - Elementattribute verändern 308
7.4.5.2 MODIFY-LOGGING-PARAMETERS - Protokolleinstellungen ändern . . . 312
7.4.5.3 OPEN-LIBRARY - Globale Bibliothek öffnen 315
7.4.5.4 SHOW-DEFAULTS - Aktuelle Belegung der Voreinstellungen ausgeben 317
7.4.5.5 SHOW-ELEMENT - Elementinhalt anzeigen 319
7.4.5.6 SHOW-ELEMENT-ATTRIBUTES - Elementeigenschaften anzeigen 332
7.4.5.7 SHOW-LIBRARY-ATTRIBUTES - Attribute einer Bibliothek anzeigen . . . 342
7.4.5.8 SHOW-LIBRARY-STATUS - Status der Bibliothek anzeigen 343
7.4.5.9 SHOW-LOGGING-PARAMETERS - Globale LMSCONV-Parameter anzeigen 344
7.4.5.10 SHOW-TYPE-ATTRIBUTES - Attribute eines Elementtyps anzeigen . . . 345
7.4.5.11 SHOW-USER-EXITS - LMSCONV-Version anzeigen 347
7.4.5.12 WRITE-COMMENT - Kommentare in Ausgabemedium schreiben 348
7.5 Beispiel: Ändern eines Bindelademoduls 349
7.6 LMSCONV im Vergleich zu LMS 350
8 MSGMAKER Bearbeiten von BS2000-Meldungsdateien 354
8.1 Ablauf von MSGMAKER 356
8.1.1 Programm starten 357
8.1.2 Definition einer programmüberwachenden Jobvariablen 358
8.1.3 Bedienungsmöglichkeiten von MSGMAKER 359
8.1.4 Besondere Zeichensätze 360
8.1.5 Meldungen von MSGMAKER 361
8.2 Menümodus 362
8.2.1 Übersicht über die Masken 363
8.2.2 Reihenfolge der Masken 364
8.2.3 Allgemeiner Maskenaufbau 366
8.2.4 Eingaben in die Maske 370
8.2.5 Beschreibung häufig auftretender Felder 371
8.2.6 Beschreibung der Masken 373
8.2.6.1 Maske MENU - Hauptmaske von MSGMAKER 374
8.2.6.2 Maske MSG-FILE-ATTRIBUTES - Eintragen und Ändern der Meldungsdatei-Attribute 380
8.2.6.3 Maske COPY - Kopieren von Meldungseinheiten 382
8.2.6.4 Maske MOVE - Kopieren und Löschen von Meldungseinheiten 387
8.2.6.5 Maske SHOW - Inhalt einer Meldungsdatei anzeigen 392
8.2.6.6 Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen 397
8.2.6.7 Maske ADD-MSG - Einfügen einer Meldungseinheit 402
8.2.6.8 Maske MODIFY-MSG - Meldungseinheit ändern 407
8.2.6.9 Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes 413

8.2.6.10 Maske MEANING/RESPONSE - Einfügen oder Ändern des Bedeutungs- und Maßnahmetextes 416
8.2.6.11 Maske INSERT-ATTRIBUTES - Einfügen oder Ändern der Insert-Attribute . 421
8.2.6.12 Maske DELETE-MSG - Meldungseinheit löschen 425
8.2.6.13 Maske ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen . . . 429
8.2.6.14 Maske MODIFY-DOCUMENTATION - Ändern, Hinzufügen und Löschen von Dokumentationszeilen 432
8.2.6.15 Maske DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen 434
8.3 Anweisungen 436
8.3.1 Übersicht über die Anweisungen 437
8.3.2 Beschreibung der Anweisungen 438
8.3.2.1 ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen 439
8.3.2.2 ADD-MSG - Einfügen einer Meldungseinheit 441
8.3.2.3 COPY - Meldungseinheit kopieren 449
8.3.2.4 DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen 455
8.3.2.5 DELETE-MSG - Meldungseinheit löschen 457
8.3.2.6 END - MSGMAKER beenden 459
8.3.2.7 MERGE-MSG-FILES - Meldungsdateien mischen 460
8.3.2.8 MODIFY-DOCUMENTATION - Verändern und Löschen von Dokumentationszeilen 462
8.3.2.9 MODIFY-MSG - Meldungseinheit verändern 467
8.3.2.10 MODIFY-OPTION - Überschreiben einer Meldungseinheit 477
8.3.2.11 MOVE - Meldungseinheit kopieren und löschen 478
8.3.2.12 OPEN-MSG-FILE - Meldungsdatei eröffnen 484
8.3.2.13 SHOW - Inhalt einer Meldungsdatei anzeigen 488
8.3.3 Besonderheiten der Anweisungen im Menümodus 491
8.3.3.1 ADD-DOCUMENTATION 493
8.3.3.2 ADD-MSG 494
8.3.3.3 COPY 496
8.3.3.4 DELETE-DOCUMENTATION 497
8.3.3.5 DELETE-MSG 498
8.3.3.6 END 499
8.3.3.7 GO-TO - Verzweigen zu einer angegebenen Maske 500
8.3.3.8 MERGE-MSG-FILES 501
8.3.3.9 MODIFY-DOCUMENTATION 502
8.3.3.10 MODIFY-MSG 503
8.3.3.11 MOVE 505
8.3.3.12 SHOW 506
8.3.4 Beispiel 507

9 PAMCONV Konvertieren von Dateiformaten 512
9.1 Programmlauf starten 515
9.2 Funktionalität von PAMCONV 516
9.3 Konvertieren von Dateiformaten 522
9.3.1 Konvertierungsarten 523
9.3.2 Anforderungen an die Systemumgebung 529
9.3.3 Angabe der Quelldateien und Zieldateien 530
9.4 Umblockung 532
9.4.1 Explizite Umblockung 533
9.4.2 Implizite Umblockung 534
9.4.3 Umblockung von PAM-DATA-Dateien ohne Änderung des Dateiformats . . . 535
9.4.4 Schwierigkeit beim Herabsetzen des Blockungsfaktors 536
9.5 Durchführung der Konvertierung und Umblockung 537
9.5.1 Besonderheiten bei der Konvertierung 538
9.5.2 Weitere Hinweise zur Konvertierung 543
9.6 Anweisungen 544
9.6.1 Übersicht über die Anweisungen von PAMCONV 545
9.6.2 Beschreibung der Anweisungen 546
9.6.2.1 CHANGE-TO-SYSTEM-MODE - Wechsel in den System-Modus 547
9.6.2.2 CLASSIFY-FILE - Dateien nach Konvertierbarkeit klassifizieren 548
9.6.2.3 CONVERT-FILE - Konvertieren von Dateien 554
9.6.2.4 END - PAMCONV beenden 562
9.6.2.5 MODIFY-CONVERT-FILE-DEFAULTS - Standardwerte für CONVERT-FILE einstellen 563
9.6.2.6 MODIFY-LOGGING-OPTIONS - Protokollierungswerte einstellen 570
9.6.2.7 SHOW-CONVERT-FILE-DEFAULTS - Standardwerte für CONVERT-FILE auflisten 571
9.6.2.8 SHOW-LOGGING-OPTIONS - Eingestellte Protokoll-Optionen auflisten . 572
9.7 PAMCONV-Programmablauf 573
9.8 Fehlerbehandlung 575
9.9 Meldungen von PAMCONV 577
10 PASSWORD Verschlüsseln von Kennwörtern 578
10.1 Arbeitsweise und Programmablauf 579
10.1.1 Kennwörter 580
10.1.2 Programmablauf 582
10.2 Anweisungen 583
10.2.1 Übersicht über die Anweisungen von PASSWORD 584
10.2.2 Beschreibung der Anweisungen 585
10.2.2.1 CONVERT - Stärkstes Kennwort einer Datei verschlüsseln 586
10.2.2.2 ENCPASS - Dateikennwort verschlüsseln und in PASSWORD-Tabelle eintragen 588

10.2.2.3 ENCRYPTD - Angegebenes Dateikennwort verschlüsseln 589
10.2.2.4 ENCRYPTJ - Angegebenes Logon-Kennwort verschlüsseln 590
10.2.2.5 END - PASSWORD beenden 591
10.2.2.6 HELP - Hilfe zu PASSWORD ausgeben lassen 592
10.2.2.7 JVCONV - Kennwort für Jobvariable verschlüsseln 593
10.2.2.8 MODE - Verschlüsselungsroutine auswählen 594
10.2.2.9 PASSWORD - Dateikennwort verschlüsseln 595
11 PVSREN Umbenennen von Pubsets und Volume-Sets 596
11.1 Voraussetzungen für PVSREN 597
11.1.1 Voraussetzungen für den Ablauf von PVSREN 598
11.1.2 Voraussetzung für eine Umbenennung 599
11.1.3 Voraussetzungen für das Rückbenennen von Spiegel-Pubsets 601
11.1.4 Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets . . 602
11.2 Ablauf von PVSREN 603
11.3 Einschränkungen und Nacharbeiten 607
11.4 Starten und Beenden von PVSREN 612
11.5 Meldungen von PVSREN 613
11.6 Anweisungen von PVSREN 614
11.6.1 Übersicht über die Anweisungen von PVSREN 615
11.6.2 Beschreibung der Anweisungen 616
11.6.2.1 CHECK-FILENAME-LENGTH - Prüfen der Länge von Datei- und Jobvariablennamen 617
11.6.2.2 CREATE-PUBSET-FROM-MIRROR - Neuen Pubset aus Spiegel-Platten eines Pubset erzeugen 619
11.6.2.3 MODIFY-JOINFILE - Ändern der Standardkatalogkennung im Benutzerkatalog 624
11.6.2.4 MODIFY-LOGGING-OPTIONS - Ändern voreingestellter Protokollierungswerte 625
11.6.2.5 RENAME-PUBSET-OR-VOLUME-SET - Konvertieren der Pubset-Notation oder Umbenennen eines SF- oder SM-Pubsets oder Volume-Sets 626
11.6.2.6 RESTART-RENAMING - Neustart der Konvertierung oder Umbenennung . 629
11.6.2.7 RESTORE-LABELS-OF-PUBSET - Pubsets aus Spiegel-Pubsets mit Doppelpunkt-Notation rückbenennen 630
11.6.2.8 SET-NAME-OF-NEW-VOLUME-SET - Volume-Set-Name beim Erzeugen eines SM-Pubsets aus Spiegel- Platten festlegen 631
11.6.2.9 SHOW-LOGGING-OPTIONS - Anzeigen der gültigen Werte für die Protokollierung 632
12 RMS - Rep-Montage-System 633
12.1 Anwendergruppen und Aktionen 636
12.2 RMS-Depot 637
12.3 Flexibler Einsatz von RMS 638

12.4 Bedienung von RMS 639
12.4.1 Starten und Beenden von RMS 640
12.4.2 Eingaben an RMS 641
12.5 Funktionsübersicht 643
12.5.1 Transfer von Korrekturmengen 644
12.5.2 Laderdienste 645
12.5.3 Korrekturverwaltung - Notizbuch 646
12.5.4 Informationsdienste 647
12.5.5 Textverarbeitung 648
12.5.6 RMS-Verwaltung 649
12.5.7 Übersicht über die Funktionen des Standardmenüs 650
12.6 Dialogorientierte Anwendung 652
12.6.1 Transfer von Korrekturmengen 653
12.6.2 Laderdienste 657
12.6.3 Korrekturverwaltung - Notizbuch 663
12.6.4 Informationsdienste 667
12.6.4.1 Korrektur - Dialoginformation 668
12.6.4.2 Korrektur - Gesamtübersicht 669
12.6.4.3 Notizbuch - Dialoginformation 670
12.6.4.4 Notizbuch - Gesamtübersicht 671
12.6.4.5 Vergleich von Rep-Mengen 672
12.6.4.6 Zeichenorientiertes Suchen 673
12.6.5 Textverarbeitung 674
12.6.6 RMS-Verwaltung 676
12.6.6.1 Installation 677
12.6.6.2 Startprozedur und Basisparameter 678
12.6.6.3 Funktionsumfang definieren 681
12.6.6.4 Beispiel einer Definitionsdatei 683
12.6.6.5 DEPOT-Pflege 684
12.7 Typische Einsatzfälle für RMS 686
12.7.1 Depot aufbauen 687
12.7.2 Neue Lieferpakete ins Depot einfahren 688
12.7.3 Optionale Reps einschalten/aktivieren 689
12.7.4 Lader bauen im Dialog 690
12.7.5 Lader bauen über Batchdatei 691
12.7.6 Übergabe einer REP-Menge an die nachfolgende Anwendergruppe 692
12.7.7 Zurückweisung einer falschen Rep-Korrektur 693
12.8 Anweisungen 694
12.8.1 Erzeugen und Bearbeiten einer Software-Konfiguration 695
12.8.1.1 CREATE-SW-CONF - Software-Konfiguration neu erstellen 696
12.8.1.2 MODIFY-SW-CONF - Software-Konfiguration ändern 698

12.8.1.3 DELETE-SW-CONF - Software-Konfiguration löschen 700
12.8.2 Optionale Rep-Auswahl 701
12.8.2.1 CREATE-OPT-PACKET - Optionales REP-Auswahlpaket erstellen 702
12.8.2.2 MODIFY-OPT-PACKET - Optionales REP-Auswahlpaket erweitern . . . 703
12.8.2.3 DELETE-OPT-PACKET - Optionales Paket löschen 704
12.8.3 Ladermodifikationen 705
12.8.3.1 CREATE-MOD-PACKET - Ladermodifikations-Element neu erstellen . . 706
12.8.3.2 MODIFY-MOD-PACKET - Ladermodifikations-Element verändern 707
12.8.3.3 DELETE-MOD-PACKET - Ladermodifikations-Element löschen 708
12.8.4 REPFILE-Definitionen 709
12.8.4.1 CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen 710
12.8.4.2 MODIFY-REPFILE-DEFINITION - Laderdefinitionen ändern 712
12.8.4.3 DELETE-REPFILE-DEFINITION - Laderdefinitions-Element löschen . . 714
12.8.5 Anlagenladerdatei erstellen 715
12.8.5.1 SELECT-REPFILE-TO-BUILD - Anlagenladerdefinitionen auswählen . . 716
12.8.5.2 BUILD-REPFILE - Anlagenlader erstellen bzw. installieren 718
12.8.6 Transport von Korrekturliefermengen 719
12.8.6.1 INPUT-DELIVERY-PACKET - Lieferpaket übernehmen 720
12.8.6.2 MODIFY-DELIVERY-PACKET - Abweichungen erfassen (Notizbuch) . . 721
12.8.6.3 CHECK-DELIVERY-PACKET - Lieferpaket prüfen 722
12.8.6.4 CREATE-DELIVERY-PACKET - Lieferpaket erzeugen 723
12.8.7 Informationsfunktionen 724
12.8.7.1 CREATE-MATRIX-LIST - MATRIX-Liste erstellen 725
12.8.7.2 COMPARE-REPFILE - Repmengen vergleichen 726
12.8.7.3 CREATE-RMS-OPTIONS - RMS-Einstellungen ändern 727
12.8.7.4 CALL-FUNCTION - Modul aufrufen 729
12.8.8 Steueranweisungen 730
12.8.8.1 END - Eingaben abschließen oder RMS im Batch-Mode beenden 731
12.9 Referenzinformation 732
12.9.1 Verwendung der Funktionstasten 733
12.9.2 Verwendung der P-Tasten 734
12.9.3 Darstellung des JD 735
12.9.4 Namenskonventionen für Eingabedateien 736
12.9.5 Ausgabedateien 737
12.9.6 Modulliste 738
12.10 Anweisungen von RMS vor V7.0 740
12.10.1 Beispiele 742
12.10.2 Beschreibung der Anweisungen 744
13 SIR Pubsets einrichten 746
13.1 Einrichten von Pubsets mit SIR 747
13.1.1 Volume-Sets 748

13.1.2 SM-Pubsets 750
13.1.3 SF-Pubsets 754
13.2 Pubset-Zustände 758
13.3 Übersicht der SIR-Funktionen und -Anweisungen 760
13.4 Installation und Betrieb von SIR 766
13.5 Übertragen von Dateien mit SIR 769
13.6 Anweisungen 770
13.6.1 Übersicht über die Anweisungen von SIR 771
13.6.2 Eingaberegeln 772
13.6.3 Anweisungs-Editor 774
13.6.4 Beschreibung der Anweisungen 775
13.6.4.1 BEGIN-VOLUME-SET-DECLARATION - Einrichten eines Volume-Set . 776
13.6.4.2 COPY - Dateien übertragen 782
13.6.4.3 CREATE-CATALOG - Dateikatalog festlegen 786
13.6.4.4 CREATE-IPL-VOLUME - Platte urladefähig machen 788
13.6.4.5 CREATE-PAGING-FILE - Paging-Datei einrichten 791
13.6.4.6 CREATE-SNAP-FILE - Snapshot-Datei einrichten 793
13.6.4.7 CREATE-VOLUME - Platten initialisieren 794
13.6.4.8 DECLARE-PUBSET - Pubset und Verarbeitungsart definieren 798
13.6.4.9 DELETE-IPL-FACILITY - Urladefähigkeit und Urladedateien einer Platte löschen 804
13.6.4.10 END - SIR beenden 805
13.6.4.11 END-VOLUME-SET-DECLARATION - Anweisungsfolge für Volume-Set beenden 806
13.6.4.12 INITIALIZE-PRIVATE-VOLUME - Privatplatten initialisieren 807
13.6.4.13 INITIALIZE-PUBLIC-VOLUME - Public-Platten initialisieren 809
13.6.4.14 MODIFY-IPL-VOLUME - Modifizieren einer Urladeplatte 812
14 SMPGEN Konvertieren von SF-Pubsets zu SM-Pubsets 815
14.1 Beispielszenario 816
14.2 Voraussetzungen für den Ablauf von SMPGEN 820
14.3 Prüffunktion (Konsistenz-Check) 822
14.3.1 Was wird geprüft? 823
14.3.2 Ausgaben 825
14.4 Erzeugen neuer SM-Pubsets 831
14.4.1 Bestimmung von Kontingenten und anderen Eigenschaften 834
14.4.2 Einschränkungen 836
14.4.3 Ausgaben 837
14.5 Erweitern eines bestehenden SM-Pubsets 838
14.5.1 Einschränkungen 841
14.5.2 Ausgaben 843
14.6 Starten und Beenden von SMPGEN 844

14.7 Anweisungen 845
14.7.1 Übersicht über die Anweisungen von SMPGEN 846
14.7.2 Beschreibung der Anweisungen 847
14.7.2.1 CREATE-SYSTEM-MANAGED-PUBSET - Konvertieren eines SM-Pubsets aus SF-Pubsets 848
14.7.2.2 MODIFY-SYSTEM-MANAGED-PUBSET - Erweitern eines bestehenden SM-Pubsets 863
14.8 Fehlerverhalten 876
14.9 Ausgaben in Bildschirmmasken 878
14.10 Meldungen von SMPGEN 892
14.11 Eigenschaften von SM-Pubsets und Volume-Sets 893
15 SPCCNTRL Prüfen und Verwalten von Platzzuweisungen auf Platten 896
15.1 Bedienung 897
15.2 Anweisungen 898
15.2.1 Übersicht über die Anweisungen von SPCCNTRL 899
15.2.2 Beschreibung der Anweisungen 900
15.2.2.1 BKPT - Das Programm SPCCNTRL unterbrechen 901
15.2.2.2 CHECK - Durchführen von Belegungsprüfungen 902
15.2.2.3 DISPLAY - Ausgeben über SYSOUT 905
15.2.2.4 END - SPCCNTRL beenden 912
15.2.2.5 HELP - Beschreibung über SPCCNTRL-Anweisungen ausgeben 913
15.2.2.6 LIST - Ausgeben über SYSLST 914
15.2.2.7 MODIFY - Standardmäßig implementierte Voreinstellungen verändern . 918
15.2.2.8 PURGE - Entfernen von totem Platz und Löschen von Katalogeinträgen . . . 919
15.2.2.9 TRACE - Identifizieren und Ausgeben von Blöcken und Einträgen im Systemkatalog und VTOC-Bereich einer Privatplatte 922
16 TPCOMP2 Vergleichen der Daten zweier Magnetbänder 926
16.1 Funktionen 927
16.2 Programmlauf starten 928
16.3 Anweisungen 929
16.3.1 Übersicht über die Anweisungen von TPCOMP2 930
16.3.2 Beschreibung der Anweisungen 931
16.3.2.1 COM - Bereiche von Eingabebänder vergleichen 932
16.3.2.2 END - TPCOMP2 beenden 933
16.3.2.3 LIM - Bandvergleich nach bestimmter Anzahl von Blöcken beenden . . . 934
16.3.2.4 POS - Magnetbänder positionieren 935
16.3.2.5 RCD - Verarbeitung von Eingabesätzen variabler Länge festlegen 936
16.3.2.6 STP - TPCOMP2-Lauf bei abweichenden Magnetbändern anhalten . . . 937
16.4 Anwendung von /ADD-FILE-LINK 938
16.5 Meldungen von TPCOMP2 940

17 VOLIN Initialisieren von Plattenspeichern 943
17.1 Ablauf von VOLIN 944
17.2 Erstellung der BS2000-Kennsätze (Kennsatzgenerierung) 945
17.3 Programmablauf 947
17.4 Bedienung 949
17.5 Anweisungen 950
17.5.1 Übersicht über die Anweisungen von VOLIN 951
17.5.2 Beschreibung der Anweisungen 952
17.5.3 Gegenüberstellung von ISP- und SDF-Anweisungen 955
18 Literatur 956

Dienstprogramme

1 Einleitung

Dieses Handbuch beschreibt folgende Dienstprogramme, die zur Steuerung und Überwachung des Betriebssystems BS2000 dienen:

Name des Dienstprogramms	Privilegierung erforderlich	beschrieben im Kapitel ...
DPAGE	für bestimmte Anweisungen	"DPAGE Auflisten und Verändern von Plattendateien"
INIT	für bestimmte Anweisungen	"INIT Initialisieren von (emulierten) Magnetbändern"
IORM	ja	"IORM Dynamische Steuerung von I/O-Ressourcen"
JMP	ja	"JMP Rekonstruktion von ENTER-Kommandos aus dem Jobpool des JMS"
JMU	ja	"JMU Einrichten und Warten der Systemdatei SJMSFILE"
LMSCONV	nein	"LMSCONV Erstellen und Verwalten von Bibliotheken"
MSGMAKER	nein	"MSGMAKER Bearbeiten von BS2000-Meldungsdateien"
PAMCONV	nein	"PAMCONV Konvertieren von Dateiformaten"
PASSWORD	nein	"PASSWORD Verschlüsseln von Kennwörtern"
PVSREN	ja	"PVSREN Umbenennen von Pubsets und Volume-Sets"
RMS	ja	"RMS - Rep-Montage-System"
SIR	für bestimmte Anweisungen	"SIR Pubsets einrichten"
SMPGEN	für bestimmte Anweisungen	"SMPGEN Konvertieren von SF-Pubsets zu SM-Pubsets"
SPCCNTRL	für bestimmte Anweisungen	"SPCCNTRL Prüfen und Verwalten von Platzzuweisungen auf Platten"
TPCOMP2	nein	"TPCOMP2 Vergleichen der Daten zweier Magnetbänder"
VOLIN	ja	"VOLIN Initialisieren von Plattenspeichern"

Drei Themenbereiche werden von den Dienstprogrammen behandelt:

- Die Bearbeitung von Datenträgern
- Die Bearbeitung von Dateien
- Das Steuern von I/O-Ressourcen (Geräte, Steuerungen, Kanäle, Pfade)

Die Dienstprogramme lassen sich diesen Themenbereichen wie folgt zuordnen:

Dienstprogramme zur Bearbeitung von Datenträgern

DPAGE	Ausgeben und Ändern von Daten in PAM-Seiten
INIT	Initialisieren von Magnetbändern
PVSREN	Umbenennen von Pubsets und Erzeugen von Pubsets aus Pubset-Spiegeln
SMPGEN	Konvertieren von SF-Pubsets zu SM-Pubsets
SIR	Pubsets einrichten
SPCCNTRL	Prüfen und Verwalten von Platzzuweisungen auf Platte
TPCOMP2	Vergleichen der Daten zweier Magnetbänder
VOLIN	Initialisieren von Plattenspeichern

Dienstprogramme zur Bearbeitung von Dateien

DPAGE	Ausgeben und Ändern von Daten in PAM-Seiten
JMP	Rekonstruktion von ENTER-Kommandos aus dem Jobpool des JMS
JMU	Einrichten und Warten der Datei für Stream- und Auftragsklassendefinition
LMSCONV	Erstellen und Verwalten von Bibliotheken
MSGMAKER	Erstellen und Bearbeiten von BS2000-Meldungsdateien
PAMCONV	Konvertieren von Dateiformaten
PASSWORD	Verschlüsseln von Kennwörtern
RMS	Verwalten, Dokumentieren, Liefern und Montieren von Rep-Mengen und Vorab-Korrekturen

Dienstprogramm zum Steuern von I/O-Resourcen

IORM	Optimierung der Auslastung der I/O-Resourcen

1.1 Zielsetzung und Zielgruppen des Handbuchs

Dieses Handbuch wendet sich sowohl an den privilegierten als auch an den nichtprivilegierten Benutzer von BS2000.

1.2 Konzept des Handbuchs

Dieses Handbuch beschreibt die Dienstprogramme in alphabetischer Reihenfolge.

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 *https://bs2manuals.ts.fujitsu.com*.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Folgende wesentliche Änderungen gegenüber der letzten Ausgabe haben sich für das Handbuch „Dienstprogramme “ ergeben:

- Beim Dienstprogramm JMP wurde die Anweisung CONVERT-JOBPOOL hinzugefügt.
- Das Dienstprogramm SIR unterstützt neben der Herstellung der Urladefähigkeit für die aktuelle BS2000 Version jetzt auch die Herstellung der Urladefähigkeit für die zwei Vorgängerversionen.
  Die Auswahl der BS2000 Version für die Herstellung der Urladefähigkeit erfolgt beim Start des SIR über den neuen Parameter TARGET-VERSION.

1.4 Darstellungsmittel

Wegen der häufigen Nennung der Bezeichnungen, werden der Einfachheit und Übersichtlichkeit halber folgende Abkürzungen gebraucht:

- **BS2000-Server** für die Server mit /390-Architektur und die Server mit x86-Architektur. Diese Server werden mit dem entsprechenden BS2000-Betriebssystem betrieben.
- **/390-Server** für die Server Unit /390 der Fujitsu Server BS2000 SE Serie
- **x86-Server** für die Server Unit x86 der Fujitsu Server BS2000 SE Serie
- **SE Server** für die Fujitsu Server BS2000 SE Serie (Server Units /390 und x86)

Die Zeichenfolgen `<date>`, `<time>` und `<version>` bezeichnen in Beispielen die aktuellen Ausgaben für Datum, Uhrzeit und Version eines Software-Produkts, wenn die Beispiele sonst Datums-, Zeit- und Versions-unabhängig sind.

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

Dieses Zeichen kennzeichnet Hinweise auf wichtige Informationen.

Dieses Zeichen kennzeichnet einen Warnhinweis, der auf die Möglichkeit des Datenverlustes oder anderer ernsthafter Schäden an Daten hinweist.

`[ ]` 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.

`Eingabe` In Anwendungsbeispielen sind Eingaben an das System und Ausgaben des Systems in `Schreibmaschinenschrift` dargestellt.

Meldungen und ihre Bedeutung

Meldungen werden in diesem Handbuch zumeist nur durch ihren siebenstelligen Meldungsschlüssel dargestellt. Der Meldungsschlüssel beginnt mit 3 Buchstaben (Meldungsklasse) gefolgt von einer vierstelligen Sedezimalzahl.

Mit dem BS2000-Kommando HELP-MSG-INFORMATION können Sie die Bedeutungs- und Maßnahmetexte zu einer Meldung im laufenden Betrieb abfragen.

Alle Meldungen finden Sie über eine HTML-Anwendung auf dem Manual-Server (URL: *http://bs2manuals.ts.fujitsu.com*) unter der aktuellen Version von BS2000 OS DX anstelle des füheren Handbuchs „Systemmeldungen“ und auf der DVD „BS2000 SoftBooks“.

Alle Meldungen können Sie auch mit dem Dienstprogramm MSGMAKER (siehe "MSGMAKER Bearbeiten von BS2000-Meldungsdateien") der entsprechenden Meldungsdatei entnehmen.

Metasyntax der Anweisungen

SDF-Format

Zur Beschreibung der SDF-Metasyntax siehe Handbuch „Kommandos“ [1].

ISP-Format

Anweisungsformat in den Dienstprogrammen DPAGE, INIT, PASSWORD, RMS, SPCCNTRL, TPCOMP2 und VOLIN.

Die ISP-Syntaxdarstellung ist an das SDF-Format angeglichen. Zusätzlich werden noch einige zusätzliche Metazeichen verwendet und Vereinbarungen getroffen, die nachfolgend kurz beschrieben sind.

Darstellung	Bedeutung
WRPASS	Großbuchstaben bezeichnen Konstanten, die in dieser Form eingegeben werden müssen.
programmname	Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe durch aktuelle Werte ersetzt werden.
{ YES / NO }	Geschweifte Klammern schließen Alternativen ein, d.h. aus den angegebenen Größen muss eine Angabe ausgewählt werden. Die Alternativen werden durch einen Schrägstrich getrennt.
[ ]	Eckige Klammern schließen Wahlangaben ein, die weggelassen werden dürfen. Die in den Klammern stehende Angabe muss als Ganzes weggelassen werden.
( )	Runde Klammern gehören zum Operanden und müssen mit eingegeben werden.
'BLANK'	bedeutet einen Zwischenraum (Leerzeichen, „blank“, SPACE).
D2,ALL	Das Komma, das Operanden trennt, muss mit eingegeben werden.
(dateiname1),... (vsn1, vsn2, ...)	3 Punkte bedeuten, dass die vor dem Komma stehende Einheit wiederholt werden kann (evtl. nur bis zu einem angegebenen Maximalwert).

Tabelle 1: Metasyntax ISP

2 DPAGE Auflisten und Verändern von Plattendateien

Versionsstand:	**DPAGE V17.0A**
Privilegierung:	**STD-PROCESSING** (für nicht-privilegierte Funktionen) **TSOS** (Öffnen eines Datenträgers)

Das Dienstprogramm DPAGE bietet dem nichtprivilegierten Anwender ebenso wie der Systembetreuung und dem Systemprogrammierer die Möglichkeit, folgende Funktionen auszuführen:

- Ausgeben von Dateien im PAM-Format nach SYSOUT (in einem Dialogprozess ist dies die Datenstation, in einem Batch-Prozess eine Datei)
- Ausgeben von Dateien im PAM-Format in die SYSLST-Datei (für Ausgaben mit großem Umfang)
- Verändern von Daten in einer PAM-Seite (2048 Byte) oder im PAM-Schlüssel (PAM-Key; 16 Byte)

Das Bearbeiten von Datenträgern ist der Systembetreuung vorbehalten.

Hinweis

DPAGE verlangt, dass jeder geöffnete Datenträger in PAM-Seiten formatiert sein muss, sonst sind die Ergebnisse nicht vorhersehbar.

Zu beachten ist, dass gemeinschaftliche Datenträger IPL-Sätze (Initial Program Load) und SVL-Sätze (Standard Volume Label) in den physikalischen Seiten 1, 2 und 3 enthalten.

> **!** **ACHTUNG!** Eine Veränderung dieser Seiten kann dazu führen, dass der Datenträger unbrauchbar wird. Ist der Datenträger „shareable“ geöffnet, werden Modifikationen der Seite 2 (SVL) beim WRITE nicht übernommen.

2.1 Unterstützung von Pubsets

In einem System mit mehreren Pubsets (Public Volume Sets) hat jeder Pubset seinen eigenen Dateikatalog TSOSCAT. Jeder dieser Kataloge ist durch die Katalogkennung (catid) eindeutig gekennzeichnet. Die Katalogkennung ist Bestandteil des Dateinamens.

Ein Dateiname hat das Format: :catid:$userid.dateiname

catid	Katalogkennung. Die Katalogkennung ist 1..4 Zeichen lang und muss in Doppelpunkte eingeschlossen sein.
userid	Benutzerkennung, unter der die Datei eingetragen ist. Der Benutzerkennung muss ein $-Zeichen vorangehen. Sie darf maximal 8 Zeichen lang sein und ist durch einen Punkt abzuschließen.
dateiname	Name der Datei, wie er im TSOSCAT eingetragen ist. Der Dateiname darf einschließlich aller Punkte (als Kennzeichen für Teilqualifikation) maximal 41 Zeichen lang sein.

Ein vollqualifizierter Dateiname, bestehend aus Katalogkennung, Benutzerkennung und Dateinamen, darf insgesamt 54 Zeichen lang sein:

bis zu 4 Zeichen	Katalogkennung
2 Zeichen	Doppelpunkte als Begrenzer vor und nach der Katalogkennung
1 Zeichen	$ als Beginn der Benutzerkennung
bis zu 8 Zeichen	Benutzerkennung
1 Zeichen	. (Punkt) als Abgrenzung zwischen Benutzerkennung und Dateinamen
bis zu 41 Zeichen	Dateiname einschließlich aller Punkte für Teilqualifizierung; max. Länge abhängig von der Länge der catid und der userid

2.2 Programmlauf starten

Das Programm wird gestartet mit: `/START-DPAGE`

START-DPAGE	Alias: **DPAGE**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Alternativ: `/START-EXECUTABLE-PROGRAM FROM-FILE=$DPAGE`

2.3 Anweisungen

Übersicht über die Anweisungen von DPAGE

Operation	Bedeutung
`BKPT / /[bs-cmd]`	DPAGE unterbrechen bzw. BS2000-Kommando geben
`DISPLAY / D`	Ausgeben eines bestimmten Teils einer oder mehrerer PAM-Seiten auf SYSOUT
`EDT / @`	Dateibearbeiter EDT als Unterprogramm aufrufen
`HALT / H / END / E`	Beenden des Programms DPAGE
`MODIFY / M`	Ändern des Inhalts einer Seite bzw. des Schlüssels in einem internen Arbeitsbereich
`OPEN / O`	Öffnen einer Datei oder eines Datenträgers
`PRINT / P`	Ausgeben eines bestimmten Teils einer oder mehrerer PAM-Seiten über SYSLST
`READ / R`	Lesen einer Seite in einen internen Arbeitsbereich
`WRITE / W`	Zurückschreiben der im internen Arbeitsbereich enthaltenen Seite in die Datei

Formate

Die folgenden Begriffe werden zur Beschreibung der Anweisungen verwendet:

seite	Maximal 10-stellige Dezimalzahl im Bereich von 1 bis 2147483647.
byte,byte1,byte2	Maximal 4-stellige Dezimalzahl im Bereich von 1 bis 2048. *Ausnahme* Nach dem Öffnen von 4K-formatierten Datenträgern eine Dezimalzahl im Bereich von 1 bis 4096.
Kn	Buchstabe K, dem eine ganze Zahl im Bereich von 1 bis 16 folgt.

Die in den Anweisungen anzugebende Seitennummer bezeichnet die Physical Half Page Number (PHP=2048 Byte).

Die Seitennummer bezeichnet jedoch in folgenden Fällen die logische 4K-Blocknummer (enthält 2PHP = 4096 Byte):

- wenn ein 4K-formatierter Datenträger geöffnet wurde
- wenn eine 4K-Datei auf einem Datenträger ohne PAM-Schlüssel (NK2, NK4) geöffnet wurde

2.3.1 BKPT - DPAGE unterbrechen

Mit der BKPT-Anweisung bzw. durch die Angabe eines Schrägstriches kann DPAGE unterbrochen werden (Breakpoint). Über `/RESUME-PROGRAM` erhält DPAGE wieder die Kontrolle.

Mit „/“ und „bs-cmd“ können beliebige BS2000-Kommandos gegeben werden.

Format

{ /[bs-cmd] / BKPT }

Operanden

bs-cmd
BS2000-Kommando

2.3.2 DISPLAY - Seite nach SYSOUT ausgeben

Die DISPLAY-Anweisung gibt Teile einer Seite oder mehrere Seiten nach SYSOUT aus.

Format

DISPLAY / D
page / page1-page2 / page-$ / * [,byte / ,byte1-byte2 / ,K]

Operanden

page
bezeichnet die Seite der Datei (oder des Datenträgers), die ausgegeben werden soll.

page1-page2
bezeichnet einen Bereich von Seiten, der ausgegeben werden soll (seite1 < seite2).

page-$
bedeutet, dass alle Seiten - beginnend bei „page“ - bis zum Ende der Datei (oder des Datenträgers) ausgegeben werden sollen.

*****
bedeutet, dass die Seite, die augenblicklich im internen Arbeitsbereich ist, ausgegeben werden soll.

byte
bezeichnet das Byte, das ausgegeben werden soll.

byte1-byte2
bezeichnet den Byte-Bereich, der ausgegeben werden soll (byte1 < byte2).
Für byte2 können maximal 2048 Byte eingesetzt werden, bei 4K-formatierten Datenträgern maximal 4096 Byte.

K
Nur der PAM-Schlüssel soll ausgegeben werden.

Beispiel

```
DISPLAY 1,1-224
*OPEN TEST
OPEN COMPLETED
*DISPLAY 1,1-224
 PAGE:0000000001            PAMKEY:  57739BDE  01000001  00000138  00010006
 001 --> (0001)  7CD7C602  02F8F5C1  D7C1D4C5  C4C9E340   @PF  90ATEST
 011 --> (0017)  000057E8  000057E8  F8F7F0F2  F2F50001      Y   Y870225
 021 --> (0033)  00000000  00000006  00000001  00000000
 031 --> (0049)  00010000  00000000  00000000  00000000
 041 --> (0065)  00D4010D  00000000  0000000C  027CD7C6    M           @PF
 051 --> (0081)  E5F9F04B  C1F0F040  40400000  00000000   V90.A00
 061 --> (0097)  00000000  00000000  D7C1D4C5  C4C9E340          TEST
=071 --> (0113)  40404040  40404040  40404040  40404040
 091 --> (0145)  40000000  00000000  00000000  00000000
 0A1 --> (0161)  00000000  00000000  40404040  40404040
 0B1 --> (0177)  40404040  F1F9F8F7  00000000  00000000       1987
 0C1 --> (0193)  00000000  00000000  00000000  E5F2F100              V21
 0D1 --> (0209)  00000000  00000000  6CD9D6D6  E3404040           %ROOT
```

> ¡ Die Ausgabe „=071“ am Anfang einer Zeile bedeutet, dass die nachfolgenden Zeilen, die nicht explizit ausgegeben werden, den gleichen Inhalt wie 071 haben und ebenfalls mit Blanks gefüllt sind.
>
> Im Beispiel ist die Zeile 71 markiert ("=71") und die Zeile 81 fehlt, d.h. die Zeile 81 ist ebenfalls mit Blanks gefüllt.

2.3.3 EDT - EDT aufrufen

Mit der EDT-Anweisung wird der Dateibearbeiter EDT aufgerufen.
Im Dialogbetrieb wird dabei der Full-Screen-Modus des EDT eingestellt. Im Batch-Betrieb werden die EDT-Anweisungen von SYSDTA gelesen.

Mit den EDT-Anweisungen @RETURN oder @HALT – im Full-Screen-Modus auch durch Drücken der Taste K1 – erhält DPAGE wieder die Kontrolle.

Die internen Daten des EDT (Arbeitsdateien, Variable) bleiben für einen eventuellen weiteren Aufruf des EDT erhalten. Sie werden erst bei der Beendigung von DPAGE freigegeben.

Format

EDT / @

2.3.4 END / HALT - DPAGE beenden

Die END- bzw. HALT-Anweisung beendet DPAGE.

Die geöffnete Datei wird geschlossen.

Format

HALT / H / END / E

> ¡ Falls in DPAGE irgendein Fehler auftrat, z.B. Syntaxfehler bei Eingabe eines Kommandos, wird DPAGE mit TERM UNIT=STEP beendet. In diesem Fall ist eine Fehlerbehandlung in Prozeduren möglich.

2.3.5 MODIFY - Inhalt einer Seite verändern

Mit der Anweisung MODIFY wird der Inhalt der Seite verändert, die augenblicklich im internen Arbeitsbereich steht (siehe READ).
Dabei wird die Original-Seite in der Datei (auf dem Datenträger) nicht berührt (siehe WRITE).

Format

MODIFY / M
[byte1 / Kn1] [,X'hex-string' / ,nnX'hex-string' / ,'character-string' / ,nn'character-string' / ,C'character-string' / ,nnC'character-string'] [,byte2 / ,Kn2]

Operanden

Erster Operand

byte1
bezeichnet die Stelle in der PAM-Seite (1..2048), an der der im zweiten Operanden angegebene Text den Originaltext ersetzt. Alle Bytes vor byte1 bleiben unverändert.

Kn1
bezeichnet die Stelle im PAM-Schlüssel (1..16), an der der im zweiten Operanden angegebene Text den Originaltext ersetzt. Alle Bytes vor Kn1 bleiben unverändert.

Standardwert ist 1: der im zweiten Operanden angegebene Text verändert den Inhalt der Seite, beginnend ab dem ersten Byte.

Zweiter Operand

X'hex-string' / nnX'hex-string' / 'character-string' / nn'character-string' / C'character-string' / nnC'character-string'
bezeichnet den Text, der den ursprünglichen Text ersetzen soll. Ab byte1 wird in der Länge des neuen Textes der alte Text ersetzt, wenn byte2 nicht angegeben wird.
nn ist eine ganze Zahl, die einen Wiederholungsfaktor für die aufgeführte Folge angibt. Bei Eingabe von C'...' wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Standardwert ist: kein Text wird verändert.

Der Text darf über die PAM-Seite oder den PAM-Schlüssel nicht hinausreichen.

Dritter Operand

byte2
bezeichnet die Stelle in der Original-PAM-Seite, an der der Text beginnt, der an den neuen Text angefügt werden soll - unabhängig davon, ob der neue Text in der Länge gleich, kürzer oder länger ist.

Kn2
bezeichnet die Stelle im Original-PAM-Schlüssel, an der der Text beginnt, der an den neuen Text angefügt werden soll.

Die Stellen, die der erste Operand und der dritte Operand bezeichnen, müssen beide entweder in der PAM-Seite oder im PAM-Schlüssel liegen.

Modifikationen in den ersten 8 Bytes des PAM-Schlüssels werden nicht in die Datei zurück übertragen. Ebenso werden bei Dateien ohne PAM-Schlüssel die ersten 12 Byte einer Seite (Control Block Field) nicht in die Datei übertragen.

Beispiel

Die Originalseite soll enthalten: `'ABCCATDOGXX...'`
und der Originalschlüssel soll wie folgt aussehen: `'1234DEFGO...'`

```
MODIFY 5, 'OW', 7
```

Verändert die Seite zu: `'ABCCOWDOGXX...'`

```
MODIFY 5, 'OW'
```

Verändert die Seite, wie die Anweisung in Beispiel a) zu: `'ABCCOWDOGXX...'`

```
MODIFY , 3X'C8C1', 10
```

Verändert die Seite zu: `'HAHAHAXX...'`

Die Seite wird am Ende mit 3 Byte sedezimalen Nullen aufgefüllt (X'000000').

```
M K1, 16x'00'
```

Setzt den Schlüssel auf sedezimal Null.
Nach der Anweisung WRITE werden nur die Byte 9..16 des Schlüssels auf Platte zurückgeschrieben. Byte 1..8 werden nicht verändert.

2.3.6 OPEN - Datei oder Datenträger öffnen

Mit der OPEN-Anweisung kann die vorher geöffnete Datei (der vorher geöffnete Datenträger) geschlossen und eine andere (ein anderer) geöffnet werden.
Das Öffnen eines Datenträgers ist nur der Systembetreuung gestattet.

Format

OPEN / O
{ filename [,INOUT / ,INPUT/ ,PHYSICAL] / 'vsn' [,[device-type] [**E**XCLUSIVE / ,**S**HAREABLE]] }

Operanden

Öffnen einer Datei

filename
Vollqualifizierter Dateiname oder Name einer Dateigenerationsgruppe. Der vom DVS gebotene Dateischutz kommt beim Öffnen der Datei zur Geltung (z.B. Kennwort, Fremdzugriff, Schreib-/Lesezugriff).

INOUT
Die Datei wird für Schreiben und Lesen geöffnet.

INPUT
Die Datei wird nur zum Lesen geöffnet. WRITE-Anweisungen führen zu einem PAM-WRITE-Fehler.

PHYSICAL
Die Datei wird nur zum Lesen geöffnet. WRITE-Anweisungen führen zu einem PAM-WRITE-Fehler. Ist die Datei verschlüsselt, wird durch nachfolgende Ausgabe-Anweisungen der Dateiinhalt im verschlüsselten Format ausgegeben.

Öffnen eines Datenträgers

'vsn'
Die maximal sechs Zeichen lange Archivnummer (vsn) eines Datenträgers (entweder gemeinschaftlich oder privat); wird in Apostrophe eingeschlossen.

device-type
Legt den Gerätetyp des Datenträgers fest.
Mögliche Angaben siehe Handbuch „Systeminstallation“ [7].

EXCLUSIVE
Der Datenträger wird exklusiv für eine Task eröffnet.

SHAREABLE
Der Datenträger wird gemeinschaftlich eröffnet.

Beispiel

```
OPEN 'ABCDEF'
```

Der Datenträger 'ABCDEF' wird exklusiv eröffnet, wenn er verfügbar ist.

2.3.7 PRINT - Seite ausdrucken

Die PRINT-Anweisung veranlasst das Ausdrucken von Teilen einer Seite oder ganzer Seiten über die Datei SYSLST.

Format

PRINT / P
page / page1-page2 / page-$ / * [,byte / ,byte1-byte2 / ,K]

Operanden

Diese Anweisung ist, abgesehen von der Ausgabe auf SYSLST, mit der Anweisung DISPLAY identisch. Zur Bedeutung der Operanden siehe "DISPLAY - Seite in SYSOUT-Datei ausgeben".

2.3.8 READ - Seite einlesen

Die READ-Anweisung liest eine Seite in den internen Arbeitsbereich ein.

Die eingelesenen Daten sind dann zweimal vorhanden, einmal auf der Original-PAM-Seite und einmal im internen Arbeitsbereich.

Mit der Anweisung MODIFY lässt sich der Inhalt dieser PAM-Seite im internen Arbeitsbereich verändern. Die Daten aus dem internen Arbeitsbereich können durch die Anweisung WRITE in den Originalbereich zurückgeschrieben werden.

Der interne Arbeitsbereich wird durch die Anweisung HALT gelöscht.

Format

READ / R
[page]

Operanden

page
bezeichnet die logische Seitennummer, die gelesen werden soll. Ist keine Seite angegeben, wird die nächste Seite gelesen.

Beispiel

```
READ  7
READ  8
READ  3
```

Diese Anweisungsfolge veranlasst, dass DPAGE die Seiten 7, 8 und 3 nacheinander liest. Zu beachten ist, dass nur ein interner Arbeitsbereich zur Verfügung steht.

2.3.9 WRITE - Internen Arbeitsbereich zurückschreiben

Die Anweisung WRITE schreibt die augenblicklich im internen Arbeitsbereich stehende Seite zurück in die Datei (oder auf den Datenträger).

Die Seite lässt sich mit der Anweisung MODIFY verändern.

Format

WRITE / W

Beispiel

```
READ 1
MODIFY 10,X'FF'
WRITE
```

Die Seite 1 wird in den Arbeitsspeicher gelesen.
Das zehnte Byte wird mit X'FF' belegt.
Die veränderte Seite 1 wird in die Datei zurückgeschrieben.

2.4 Meldungen von DPAGE

`CLOSE ERROR, ERROR-CODE: xxxx`

Bedeutung
Fehler beim Schließen einer Datei bzw. Platte. xxxx ist der DVS-Fehlercode.

Maßnahme
Die Analyse des Fehlercodes xxxx ist im Handbuch "DVS Makros" bzw. mit dem BS2000 HELP-Kommando xxxx zu finden.

`ERROR IN EDT-CALL`

Bedeutung
Programm-Fehler im EDT-Aufruf, EDT konnte nicht ordnungsgemäß gestartet werden.

`INVALID BS2000-Command`

Bedeutung
Falsches BS2000-Kommando angegeben, bzw. Fehler im CMD-Makro-Aufruf.

`INVALID COMMAND`

Bedeutung
Falsches Kommando, keine Aktion durchgeführt.

`INVALID OPERAND, COMMAND REJECTED`

Bedeutung
Falsche Angabe bei den Operanden, Kommando ignoriert.

`NO PAMKEY AVAILABLE`

Bedeutung
Die Datei bzw. Platte besitzt keinen PAM-Schlüssel.

`OPEN COMMAND MUST BE GIVEN FIRST, COMMAND REJECTED`

Bedeutung
Die Reihenfolge der Anweisungen ist falsch.

Maßnahme
OPEN zuerst eingeben.

`OPEN ERROR, ERROR-CODE: xxxx`

Bedeutung
Fehler beim Öffnen einer Datei bzw. Platte. xxxx ist der DVS-Fehlercode.

Maßnahme
Die Analyse des Fehlercodes xxxx ist im Handbuch "DVS Makros" bzw. mit dem BS2000 HELP-Kommando xxxx zu finden.

`OPEN VOLUME RESTRICTED TO SYSTEM ADMINISTRATOR`

Bedeutung
Das Öffnen einer Platte ist nur dem System-Verwalter erlaubt.

PAM-READ ERROR, ERROR-CODE: xxxx

Bedeutung
Fehler beim Lesen einer PAM-Seite. xxxx ist der DVS-Fehlercode.

Maßnahme
Die Analyse des Fehlercodes xxxx ist im Handbuch "DVS Makros" bzw. mit dem BS2000 HELP-Kommando xxxx zu finden.

PAM-WRITE ERROR, ERROR-CODE: xxxx

Bedeutung
Fehler beim Schreiben einer PAM-Seite. xxxx ist der DVS-Fehlercode.

Maßnahme
Die Analyse des Fehlercodes xxxx ist im Handbuch "DVS Makros" bzw. mit dem BS2000 HELP-Kommando xxxx zu finden.

READ COMMAND MUST BE GIVEN FIRST. PAGE NOT DISPLAYED.

Bedeutung
DISPLAY wurde angegeben, bevor eine Seite eingelesen wurde.

Maßnahme
READ-Anweisung zuerst eingeben.

READ COMMAND MUST BE GIVEN FIRST. PAGE NOT MODIFIED.

Bedeutung
MODIFY wurde angegeben, bevor eine Seite eingelesen war.

Maßnahme
READ-Anweisung zuerst eingeben.

READ COMMAND MUST BE GIVEN FIRST. PAGE NOT PRINTED.

Bedeutung
PRINT wurde angegeben, bevor eine Seite eingelesen war.

Maßnahme
READ-Anweisung eingeben oder PRINT mit Seitenangabe.

READ COMMAND MUST BE GIVEN FIRST. PAGE NOT WRITTEN

Bedeutung
WRITE wurde angegeben, bevor eine Seite eingelesen war.

Maßnahme
READ-Anweisung zuerst eingeben.

REQUEST MEMORY ERROR

Bedeutung
Fehler beim Anfordern von Arbeitsspeicher mit dem REQM-Makro.

SEARCHED STRING NOT FOUND

Bedeutung
Der gesuchte String wurde nicht gefunden

VOLUME NOT FOUND

Bedeutung
Angegebender Datenträger konnte von DPAGE nicht gefunden werden.

Maßnahme
Richtige Archiv-Nummer des Plattenträgers angeben oder Systemverwaltung anrufen.

ERROR OCCURED IN DPAGE-RUN: TERM UNIT=STEP

Bedeutung
Während des DPAGE-Laufs ist ein Fehler aufgetreten. Innerhalb einer Prozedur wird zum nächsten STEP, ENDP, LOGOFF oder zur Fehlerbehandlung (z.B. IF-BLOCK-ERROR) verzweigt.

Maßnahme
SYSOUT- und SYSLST-Protokolle prüfen und ggf. Prozedur ändern und wiederholen.

3 INIT Initialisieren von (emulierten) Magnetbändern

Versionsstand: **INIT V21.0C**

Privilegierung: **STD-PROCESSING** (für nicht-privilegierte Funktionen)
TAPE-PROCESSING (für privilegierte Funktionen)

Das Dienstprogramm INIT initialisiert Magnetbänder (Magnetbandkassetten, MBK) und emulierte Magnetbänder.

Initialisieren bedeutet, dass an den logischen Anfang des Datenträgers ein Datenträgerkennsatz (VOL1) und eventuell noch zwei Dateikennsätze (HDR1 und HDR2 einer Dummy-Datei) geschrieben werden. Der logische Anfang des Datenträgers ist die Position auf dem Datenträger, ab der BS2000 Daten ablegen kann. Dies muss nicht zwingend der physikalische Datenträgeranfang sein.
Die Kennsätze werden bei Magnetbandkassetten stets nicht-komprimiert geschrieben, auch wenn der komprimierte Schreibmodus (Datenträgertyp TAPE-C4) angegeben wurde. Der Datenträgerkennsatz (VOL1) wird stets unverschlüsselt geschrieben, auch wenn der verschlüsselte Schreibmodus angegeben wurde.

Mit einer Anweisung können bis zu 16 Datenträger bearbeitet werden.

Bei allen INIT- oder LIST-Funktionen kann ein definiertes Gerät ausgewählt werden, das den entsprechenden Datenträgertyp unterstützt.

Zum Schutz vor unbeabsichtigtem Überschreiben wird vor jedem Schreiben neuer Kennsätze zunächst versucht, eventuell vorhandene Kennsätze zu lesen. Der Inhalt vorhandener Kennsätze wird zur Kontrolle auf SYSOUT ausgegeben. Der Benutzer hat dann die Möglichkeit, das Initialisieren abzubrechen.
Bei Magnetbändern unterbleibt das Kontrolllesen nur dann, wenn der Datenträger von der Geräteverwaltung bereits als leer erkannt und der INIT-Operand NEW angegeben wurde.

Wurde beim Kontrolllesen eines Datenträgers in einem der Kennsätze VOL1, HDR1 oder HDR3 eine Zugriffsbeschränkung (Zugriffsvermerk, Freigabedatum, Lese- oder Schreibkennwort, Zugriffsart "nur lesen") erkannt, wird eine entsprechende Meldung ausgegeben.

Die Funktionen "Lesen von Kennsätzen" (Anweisung LIST) und "Schreiben neuer Kennsätze oder Abschnittsmarken" (Anweisung INIT) werden nur ausgeführt, wenn der Anwender das Privileg TAPE-ADMINISTRATION besitzt.

Das Dienstprogramm INIT bietet folgende Funktionen:

1. Für Magnetbänder:
   - Schreiben des Bandanfangskennsatzes VOL1 sowie der Dateianfangskennsätze HDR1 und HDR2 (abgeschlossen mit zwei Bandabschnittsmarken).

     Bandinhalt: VOL1-HDR1-HDR2-TM-TM
   - Schreiben nur des Bandanfangskennsatzes VOL1 (abgeschlossen mit zwei Bandabschnittsmarken).

     Bandinhalt: VOL1-TM-TM
   - Schreiben von zwei Abschnittsmarken an den logischen Anfang des Datenträgers. Es werden keine Kennsätze geschrieben.

     Bandinhalt: TM-TM
   - Ausgabe des Inhalts vorhandener VOL1, HDR1-, HDR2- und HDR3- Kennsätze auf SYSOUT oder Bedienplatz. Neue Kennsätze werden dabei nicht geschrieben.
   - Formatieren des Magnetbandes, falls der Datenträger und das Bandgerät dies zulässt.

2. Allgemein:
   - Ein- bzw. Ausschalten von Sonderfunktionen, die während des weiteren Programmlaufs bis zum Rücksetzen für alle INIT- und LIST-Anweisungen gültig sind.
   - Ausgabe einer Kurzbeschreibung der Anweisungen und Operanden.
   - Umschalten der Ein-/Ausgabe auf den Bedienplatz, nach Starten des Programms mittels ENTER-Auftrag.

   a. Schutzfunktionen:
      - Angabe einer Soll-Archivnummer. Ein Magnetband wird nur bearbeitet, wenn die Archivnummer im vorhandenen VOL1-Kennsatz mit der angegebenen Soll-Archivnummer übereinstimmt.
      - Löschen des gesamten Datenträgers vor dem Schreiben neuer Kennsätze (DSE = Data Security Erase).

        > ℹ Löschen bedeutet in diesem Zusammenhang, dass der Datenträger mit einem geräteabhängigen Löschmuster überschrieben wird. Einige neuere Geräte realisieren das Löschen möglicherweise nur auf logischer Ebene durch das Löschen von Daten-Verwaltungsinformationen auf dem Datenträger. Mit Spezialgeräten und hohem technischen Aufwand kann die Information früherer Aufzeichnungen unter Umständen trotzdem wieder lesbar gemacht werden. Vollständige Sicherheit vor unbefugtem Lesen von Restinformationen, die auf dem Datenträger verblieben sind, ist daher nur durch die physische Zerstörung des Datenträgers gewährleistet.

      - Abbrechen der Initialisierung, wenn beim Kontrolllesen eine Zugriffsbeschränkung gefunden wurde.
      - Ausschluss von Sonderzeichen, die nicht DIN 66003 entsprechen, von der Verwendung in Kennsätzen.

3.1 Betriebsarten

Die Funktionen des Dienstprogramms INIT können in zwei Betriebsarten ausgeführt werden: Normalbetrieb und Konsolbetrieb.

3.1.1 Normalbetrieb

Diese Betriebsart ist normalerweise wirksam, wenn das Programm INIT mit `/START-INIT` gestartet wird.

START-INIT	Alias: **INIT**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Um die Kompatibilität zu früheren Versionen zu gewährleisten, wird auch noch das folgende Kommando unterstützt:

```
/START-EXECUTABLE-PROGRAM FROM-FILE=$INIT
```

Nach dem Starten des Programms erfolgt die Meldungsausgabe über SYSOUT. Die Eingaben werden von SYSDTA erwartet.
Das Programm läuft ab, wie im Abschnitt „Programmlauf“ beschrieben. Die Anweisungen an das Programm INIT können hierbei:

- im Dialogbetrieb vom Terminal aus eingegeben werden
- innerhalb einer Kommandoprozedur stehen (Umweisung von SYSDTA nach SYSCMD)
- in einem ENTER-Job stehen (SYSDTA standardmäßig auf SYSCMD umgewiesen)

Beispiele

Kommandoprozedur:

```
/BEGIN-PROCEDURE
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/START-INIT
...    (INIT statements)
END
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROCEDURE
```

ENTER-Job:

```
/SET-LOGON-PARAMETERS
/START-INIT
...    (INIT statements)
END
/EXIT-JOB
```

3.1.2 Konsolbetrieb

In dieser Betriebsart erfolgt die Bedienung des Programms INIT im Dialogbetrieb über den Bedienplatz (Konsole). Dazu muss zunächst vom Bedienplatz aus ein ENTER-Auftrag gegeben werden, der das Programm INIT startet und mittels OPTION CONS die Ein/Ausgaben auf den Bedienplatz umschaltet.

```
/SET-LOGON-PARAMETERS
/START-INIT
OPTION CONS
/EXIT-JOB
```

Nach dem Starten des Auftrags werden alle Meldungen und Eingaben über den Bedienplatz mit der Angabe der Auftragsfolgenummer (TSN) abgewickelt. Das Programm läuft dann wie im Abschnitt „Programmlauf“ beschrieben ab.

Die Anweisung OPTION CONS ist nur innerhalb eines ENTER-Auftrags zulässig und wird im Dialogbetrieb abgewiesen.

Beispiel

`<tsn>.INIT TAPE-C4,VSN=MBK001`

`<tsn>.LIST TAPE-C4`

mit <tsn> = Auftragsfolgenummer

> ¡ Wenn Sie eine SOLIS2-Lieferung im aktuellen HOME-PVS installieren, so muss vor Übernahme der Lieferung das Dienstprogramm INIT an der Konsole mit <tsn>.END beendet werden, wenn es im Batch Mode geladen ist.

3.2 Programmlauf

Im Folgenden wird der Ablauf der Initialisierung eines Magnetbandes anhand von Protokollbeispielen gezeigt.

Meldungen, die mit `(OUT)` gekennzeichnet sind, werden im Normalbetrieb auf SYSOUT ausgegeben. Diejenigen, die mit `(MSG)` gekennzeichnet sind, werden stets am Bedienplatz ausgegeben.

3.2.1 Programmstart

Beispiel

```
(IN)  /start-init                                          1.
(OUT) % ...                                                2.
(OUT) %  NVI0000 PROGRAM INIT READY                        3.
(OUT) %  NVI0001 ENTER COMMAND. (END = TERMINATE INIT)     4.
```

1. Das Programm INIT wird gestartet.
2. Es werden verschiedene Lademeldungen ausgegeben
3. INIT meldet die Bereitschaft ...
4. ... und fordert Anweisungen an.

3.2.2 Initialisieren eines Magnetbandes (Beispiel)

```
(OUT) % NVI0001 ENTER COMMAND. (END = TERMINATE INIT)
(IN) INIT TAPE-C4,VSN=TAPE1,ERASE                                    1.
(MSG) %  NKVT013 MOUNT TAPE '*SCRAT' ON DEVICE 'T9';                 2.
                (USE='SPECIAL',WR='UNDEF',TYPE='TAPE-C4',
                INIT T-C4,VSN=TAPE1).
               '(ETX = YES; MN; N = NO )'?
(IN)  0ABC.T7
(OUT)  % NVI0003 LABELS ON TAPE (ISO7):                              3.
(OUT)  % NVI0004 VOL1 LABEL:
             'VOL1TAPE01                              '
               '                                       1'
(OUT)  % NVI0004 HDR1 LABEL:
                'HDR1SAMPLE.FILE   TAPE010001000100010'
                '0 92104 92104 000000BS2000              '
(OUT)  % NVI0004 HDR2 LABEL:
                'HDR2V8000102044 0                P     '
                '      .C..04                            '
(OUT)  % NVI0004 HDR3 LABEL:
                'HDR3TSOS    SAMPLE.FILE                 '
                '                .............00          '
(OUT)  %  NVI0007 OVERWRITE TAPE?                                    4.
                 REPLY (YES=YES N=NO)
(IN)   YES
(OUT)  %  NVI0208 DATA SECURITY ERASE STARTED                        5.
(OUT)  %  NVI0209 DATA SECURITY ERASE COMPLETED
(OUT)  %  NVI0210 INITIALIZATION OF TAPE 'TAPE1'                     6.
                 ON DEVICE 'T7' COMPLETED
(OUT)  %  NVI0001 ENTER COMMAND. (END = TERMINATE INIT)              7.
```

1. Soll ein Datenträger initialisiert werden, ist die Anweisung INIT mit einem eindeutigen, gültigen Datenträgertyp anzugeben (hier mit Angabe ERASE (Datensicherheitslöschen)).
2. Der Operator erhält am Bedienplatz die Aufforderung, den Datenträger zu montieren. Wenn der Datenträger montiert ist (hier auf Gerät T7), bestätigt er dies unter Angabe der Auftragsfolgenummer (TSN). Falls der Datenträger auf dem in der MOUNT-Aufforderung genannten Gerät (T9) montiert wurde, kann die Angabe des mnemotechnischen Gerätenamens (T7) entfallen.
3. Der Datenträger wird gelesen und die gelesenen Kennsätze werden ausgegeben. Sind die Kennsätze im ISO7-Code geschrieben, wird ein entsprechender Hinweis ausgegeben. Im Beispiel wurde der Inhalt der Kennsätze auf mehrere Zeilen verteilt und mit Apostrophen abgegrenzt. Im Betrieb wird der Inhalt der Kennsätze als eine einzige Zeichenkette ausgegeben, wobei Leerzeichen am Ende der Zeichenkette abgeschnitten werden.

   Enthält der Datenträger keinen oder einen fehlerhaften Datenträgerkennsatz VOL1, werden an Stelle der Kennsätze Hinweismeldungen ausgegeben:

   Leerer Datenträger:

   `(OUT) % NVI0102 NO LABELS ON TAPE. TAPE EMPTY`

   Abschnittsmarken am Beginn des Datenträgers:

   `(OUT) % NVI0103 NO LABELS ON TAPE. TAPE MARKS READ`

   Fehler beim Lesen des VOL1-Kennsatzes :

   `(OUT) % NVI0104 I/O-ERROR WHILE READING LABELS`

4. Darauf folgt die Frage, ob der Datenträger überschrieben werden soll oder nicht. Dies ist die letzte Möglichkeit, das Überschreiben des Datenträgers zu verhindern. Die Antwort YES (für Überschreiben) ist voll auszuschreiben. Abkürzungen werden aus Sicherheitsgründen nicht angenommen. Alle anderen Eingaben werden als Ablehnung (also Nicht-Überschreiben) interpretiert.
5. Wurde über den Parameter ERASE Datensicherheitslöschen gefordert, wird der Datenträger zunächst gelöscht. Der Beginn und das Ende dieses Vorgangs, der je nach Gerät und Datenträger mehrere Minuten bis einige Stunden dauern kann, werden gemeldet.
6. Danach werden die neuen Kennsätze geschrieben und das Ende der Initialisierung gemeldet.
   Bei nicht erfolgreicher Initialisierung wird zunächst eine Meldung über die Fehlerursache ausgegeben und dann der Abbruch der Initialisierung:

   ```
   (OUT) % NVI0309 TAPE WITH VSN ' ' CANNOT BE USED.
                   INITIALIZATION ABORTED
   ```
7. Die nächste Anweisung wird angefordert.

3.2.3 Programmende

```
(OUT)  %  NVI0001 ENTER STATEMENT. (END = TERMINATE INIT)
(IN)   END                                                          1.
(OUT)  %  NVI0011 PROGRAM INIT TERMINATED NORMALLY                  2.
```

1. Die Anweisung END wird eingegeben.
2. Das Programm INIT wird beendet und das normale Programmende gemeldet.

3.2.4 Probleme bei der Datenträgerbearbeitung

Beim Kontrolllesen vor dem Schreiben neuer Kennsätze sowie beim Löschen oder Schreiben neuer Kennsätze werden im Fehlerfall evtl. Meldungen der Gerätefehlerbehandlung am Bedienplatz ausgegeben, die nähere Hinweise auf die Fehlerursache geben.
Bei Meldungen, die eine Antwort erfordern, sind die möglichen Antworten im jeweiligen Meldungstext vorgegeben. Die Antwort muss unter Angabe der Auftragsfolgenummer an der Bedienungsstation erfolgen, z.B. wenn die Antwort „NO“ gegeben werden soll:

```
<tsn>.NO
```

mit <tsn> = aktuelle Auftragsfolgenummer

3.3 Anweisungen

Eingabe der Anweisungen von INIT

INIT liest Anweisungen aus der Systemdatei SYSDTA (standardmäßig vom Terminal im Dialogbetrieb) oder vom Bedienplatz (im Konsolbetrieb).

Eine INIT-Anweisung besteht aus dem Operationsnamen (INIT, LIST usw.) und eventuellen Operanden mit Operandenwerten. Erfordert eine Anweisung mehrere Operanden, werden diese durch Kommata getrennt. Zwischen Operand und Operandenwert(en) darf kein Leerzeichen stehen.

Eine Eingabezeile darf aus maximal 72 Zeichen bestehen. Überzählige Zeichen werden ohne Warnung abgeschnitten. Führende Leerzeichen werden ignoriert.

Bei Bedarf kann eine INIT-Anweisung in Folgezeilen fortgesetzt werden. Dazu muss die Anweisung an beliebiger Stelle mit einem Fortsetzungszeichen (Bindestrich: -) versehen werden. INIT fordert dann zur Fortsetzung der Anweisungseingabe auf:

```
(OUT) % NVI0002 ENTER ADDITIONAL OPERANDS
```

Alle Zeichen, die einem Leerzeichen nach dem Fortsetzungszeichen folgen, werden ignoriert (Zeilenkommentar). Folgen einem Bindestrich unmittelbar andere Zeichen als Leerzeichen, wird er nicht als Fortsetzungszeichen erkannt.

Führende Leerstellen einer Fortsetzungszeile bleiben unberücksichtigt. Die in einer Fortsetzungszeile angegebenen Operanden werden an die vorherige Zeile an der Stelle des Fortsetzungszeichens angehängt.

Eine INIT-Anweisung darf, auch wenn sie sich über mehrere Fortsetzungszeilen erstreckt, aus maximal 240 Zeichen bestehen (einschließlich zulässiger Leerzeichen, ausschließlich führender Leerzeichen und Zeichen nach einem Fortsetzungszeichen).

Beispiel

```
Anweisungsteile                  Kommentar

(IN)  INIT    TAPE-C4,-          INITIALISIERE 3 MAGNETBANDKASSETTEN
(IN)  3
(IN)          VSN=(VSN01,-       VSN 1
(IN)               VSN02,-       VSN 2
                   VSN03),-      VSN 3
(IN)
              UNIT=A1
(IN)
```

Dies ist äquivalent zu folgender Eingabezeile:

```
INIT 3 TAPE-C4,VSN=(VSN01,VSN02,VSN03),UNIT=A1
```

Enthält eine Eingabezeile in der ersten Spalte einen Stern *, wird der Inhalt dieser Zeile nicht ausgewertet. Damit können Prozeduren, in denen das Dienstprogramm INIT oder ein Ablaufprotokoll aufgerufen wird, mit Kommentarzeilen versehen werden.

Übersicht über die Anweisungen von INIT

Anweisung	Bedeutung
END	Beendigung des Programms
HELP	Ausgabe einer Kurzbeschreibung der Anweisungen und Operanden
INIT	Schreiben des Datenträgerkennsatzes VOL1 sowie der Dateianfangskennsätze HDR1 und HDR2 oder Schreiben von zwei Abschnittsmarken. Optional ist das Formatieren des Datenträgers vor dem Schreiben neuer Kennsätze möglich, wenn der Datenträgertyp dies zulässt. Das Systemprivileg TAPE-ADMINISTRATION ist erforderlich.
LIST	Ausgabe des Inhalts vorhandener VOL1, HDR1- , HDR2- und HDR3-Kennsätze. Das Systemprivileg TAPE-ADMINISTRATION ist erforderlich.
OPTION	Ein- und Ausschalten von Sonderfunktionen, die während des weiteren Programmlaufs bis zum Rücksetzen für alle INIT- und LIST-Anweisungen gültig sind.

3.3.1 INIT - Kennsätze schreiben

Mit der Anweisung INIT werden Datenträgerkennsätze auf Magnetbänder oder Magnetbandkassetten geschrieben. Es wird der Datenträgertyp und entweder die zu schreibende VSN oder die Initialisierungsart „nur Bandabschnittsmarken schreiben“ festgelegt.

Wahlweise können noch festgelegt werden:

- die Anzahl der zu bearbeitenden Datenträger
- ein bestimmter Datenträger
- ein bestimmtes Gerät
- die Codierung der Kennsätze (ISO7 oder EBCDIC)
- die Behandlung des Datenträgers nach dem Initialisieren

Vor dem Schreiben neuer Kennsätze wird versucht, eventuell vorhandene Kennsätze zu lesen. Konnten Kennsätze gelesen werden, werden sie zur Kontrolle ausgegeben.
Bei eingeschalteter OPTION CONS erfolgt die Ausgabe auf Bedienplatz, andernfalls auf SYSOUT.

> ¡ Die Funktion INIT wird nur ausgeführt, wenn die Task das Systemprivileg TAPE-ADMINISTRATION besitzt.

Format

INIT
[no] voltyp { ,VSN={ vsn / (vsn1,vsn2,...vsn16) / (initval) / *} [,OWN={ name / * }][,ZERO][,ISO7] / ,WTM } [,UNIT = mn] [,NEW / CHECK = vsn] [,REW / RUN] [,ERASE] [,FORMAT]

Operanden

no
Anzahl der Datenträger, die mit dieser Anweisung bearbeitet werden sollen. Gültige Werte: 1, 2, ..., 16.

> ¡ Zwischen den Operanden no und voltyp muss mindestens ein Leerzeichen stehen.

voltyp
Legt den Datenträgertyp (volume type) fest. Dieser definiert die Art des Datenträgers und die Aufzeichnungsart.

Die in der aktuellen BS2000-Version unterstützten Datenträgertypen finden Sie im Handbuch „Systeminstallation“ [7]. INIT unterstützt darüber hinaus aus Kompatibilitätsgründen einige weitere Datenträgertypen. Die von INIT akzeptierten Datenträgertypen können Sie mit der HELP-Anweisung (siehe "HELP - Kurzbeschreibung der INIT-Anweisungen anzeigen") ermitteln.

> **i** Die Schlüsselwörter für den Datenträgertyp können abgekürzt werden, solange sie eindeutig bleiben, z.B. T-C4=TAPE-C4. Bei der Einführung neuer Datenträgertypen kann eine ursprünglich eindeutige Abkürzung möglicherweise mehrdeutig werden. Garantierte Abkürzungen gibt es nicht.

Da keine Voreinstellung wirksam wird, muss dieser Operand immer angegeben werden.

VSN
Archivnummer (Volume Sequence Number), die in den VOL1-Kennsatz einzutragen ist (keine Voreinstellung). Sollen mehrere Datenträger mit einer einzigen Anweisung initialisiert werden, müssen entsprechend viele Archivnummern oder ein Anfangswert angegeben werden.

VSN=vsn
Einzelne Archivnummer.
Es soll ein einzelner Datenträger initialisiert werden. Der Operand "no" muss daher entweder den Wert 1 haben oder er darf nicht angegeben sein.
Gültige Werte: Maximal 6 Zeichen (A..Z, 0..9, #,$,@).

Die von INIT erlaubten Sonderzeichen #,$,@, dürfen bei normgerechten Kennsätzen (DIN 66029, DIN 66003) nicht verwendet werden. Die Angabe dieser Zeichen wird bei eingeschalteter OPTION DIN abgewiesen.

VSN=(vsn1,vsn2,...,vsn16)
Mehrere Datenträger sollen mit den angegebenen Archivnummern initialisiert werden.Die Anzahl der Archivnummern muss mit der bei dem Operanden "no" gemachten Angabe übereinstimmen.
Gültige Werte:
Jede angegebene Archivnummer muss eine gültige Archivnummer vsn sein (s.o.).

VSN=(initval)
Mehrere Datenträger, beginnend mit der Archivnummer anfwert, sollen initialisiert werden. Nach jedem erfolgreichen Initialisieren wird die Archivnummer automatisch um 1 erhöht. Die Anzahl der Datenträger muss mit dem Operanden "no" angegeben werden.

Gültige Werte:
anfwert muss eine gültige Archivnummer sein und mindestens eine Dezimalziffer enthalten. Die angegebene Anfangs-Archivnummer wird von links nach rechts ausgewertet. Der Startwert der zu bildenden Archivnummer ist die erste gefundene Folge von Dezimalziffern. Enthält anfwert mehrere Ziffernfolgen, die durch andere Zeichen (z. B. Buchstaben) getrennt sind, wird nur die am weitesten links stehende Ziffernfolge als Startwert benutzt.

Die Anzahl der Ziffern des ermittelten Startwerts muss auch die höchste gebildete Archivnummer aufnehmen können, andernfalls wird die Anweisung abgewiesen.

Beispiele

```
INIT 5 TAPE-C4,VSN=(A12B12)
```

Es werden Datenträger mit den Archivnummern A12B12, A13B12, A14B12, A15B12 und A16B12 bearbeitet.

```
INIT 16 TAPE-C4,VSN=(A1B)
```

Die Anweisung wird abgewiesen, da nur einstellige Dezimalzahlen gebildet werden können.

VSN=*
Die bereits vorhandene Archivnummer soll übernommen werden.
Enthält der Datenträger keinen lesbaren VOL1-Kennsatz mit einer gültigen Archivnummer, wird eine entsprechende Warnung ausgegeben, und es werden keine neuen Kennsätze geschrieben.

OWN
Nur gültig, wenn nicht WTM angegeben ist. Name des Eigentümers, der in den VOL1-Kennsatz eingetragen werden soll.

OWN=name
Name, der in den VOL1-Kennsatz eingetragen werden soll.
Gültige Werte: Maximal 8 Zeichen (A...Z, 0...9, #, $, @, ., -)
Voreinstellung: Bei fehlender Angabe werden Leerzeichen eingetragen.

OWN=*
Ein bereits im VOL1-Kennsatz vorhandener Name des Eigentümers soll in den neuen VOL1-Kennsatz übernommen werden. Enthält der Datenträger keinen lesbaren VOL1-Kennsatz mit einem gültigen Eigentümernamen (auch nur Leerzeichen), wird eine entsprechende Warnung ausgegeben und es werden keine neuen Kennsätze geschrieben.

ZERO
Im VOL1-Kennsatz (Byte 11) wird als Zugriffskennzeichen das abdruckbare Zeichen 0 eingetragen. Wird dieser Operand nicht angegeben, wird als Zugriffskennzeichen ein Leerzeichen eingetragen.

ISO7
Die Kennsätze sollen im ISO7-Bit-Code geschrieben werden.

WTM
Schreiben von zwei Abschnittsmarken an den logischen Anfang des Datenträgers.Es werden keine Kennsätze geschrieben.

UNIT=mn
Mnemotechnischer Gerätename zur definierten Belegung eines Gerätes. Ein Datenträger soll auf dem angegebenen Gerät montiert werden.
Ungeeignete Geräte, die z.B. den geforderten Datenträgertyp nicht bearbeiten können, werden abgewiesen.

CHECK=vsn
Angabe einer Soll-Archivnummer.
Ein Datenträger wird nur bearbeitet, wenn die Archivnummer im VOL1-Kennsatz mit der angegebenen Archivnummer übereinstimmt. So soll das versehentliche Überschreiben eines falschen Datenträgers verhindert werden.
Wird der Operand CHECK angegeben und der angeforderte Datenträger ist bereits auf dem richtigen Gerät montiert, ergeht keine MOUNT-Aufforderung an den Operator und der Datenträger wird sofort bearbeitet.
Gültige Werte: Maximal 6 Zeichen (A..Z, 0..9, #,$,@).
Die von INIT erlaubten Sonderzeichen #,$,@, dürfen bei normgerechten Kennsätzen
(DIN 66029, DIN 66003) nicht verwendet werden. Die Angabe dieser Zeichen wird bei eingeschalteter OPTION DIN abgewiesen.

> CHECK darf nur angegeben werden, wenn nicht mehr als ein Datenträger mit dieser Anweisung initialisiert werden soll (Operand "no"=1 oder nicht angegeben).

NEW
Die Wirkung des Operanden hängt davon ab, ob der Datenträger bereits beschrieben ist oder von der Geräteverwaltung des Betriebssystems (Tape-Monitor) bereits beim Montieren als leer erkannt wurde.
Bei als leer erkannten Datenträgern werden sowohl das Kontrolllesen der Kennsätze als auch die OVERWRITE-Abfrage unterdrückt.
Bei bereits beschriebenen Datenträgern wird das Kontrolllesen der Kennsätze ohne Fehlerbehandlung durchgeführt.

Wird der Operand NEW nicht angegeben, versucht INIT immer, vorhandene Kennsätze zu lesen. Dies erfolgt mit voller Fehlerbehandlung, falls der Tape-Monitor von BS2000 das Band nicht bereits beim Montieren als leer erkannt hat. In einem solchen Fall wird sicherheitshalber versucht, ohne Fehlerbehandlung zu lesen.

REW
Legt fest, dass ein Magnetband bzw. eine Magnetbandkassette nach der Bearbeitung rückgespult, aber der Datenträger nicht entladen werden soll. Bei der Bearbeitung eines einzelnen Datenträgers wird REW standardmäßig angenommen.

RUN
Legt fest, dass ein Magnetband bzw. eine Magnetbandkassette nach der Bearbeitung rückgespult und der Datenträger entladen werden soll. Bei der Bearbeitung mehrerer Datenträger mit einer einzigen Anweisung (Operand "no" = 2..16) wird RUN standardmäßig angenommen.

ERASE
Der gesamte Datenträger soll vor dem Schreiben neuer Kennsätze gelöscht werden.

> ¡ Dieser Vorgang kann, je nach Gerät und Datenträger, von mehreren Minuten bis zu einigen Stunden dauern.

Wird dieser Operand nicht angegeben, werden nur die neuen Kennsätze an den Anfang des Datenträgers geschrieben und das logische Ende des Datenträgers mit einer doppelten Bandabschnittsmarke markiert. Die ursprünglichen Daten dahinter bleiben jedoch erhalten und können möglicherweise noch gelesen werden (siehe Hinweis unter „Schutzfunktionen“ in "INIT Initialisieren von (emulierten) Magnetbändern").
Konnte ein gültiger VOL1-Kennsatz gelesen werden, bleibt dieser während des Löschens erhalten, andernfalls erfolgt das Löschen vom Beginn des Datenträgers an.

FORMAT
Bewirkt eine Formatierung des Datenträgers, falls der Datenträgertyp und das Gerät dies zulassen. Anderenfalls erfolgt eine Hinweismeldung, der Operand wird ignoriert und die Initialisierung wird fortgesetzt.

> ¡ Das Formatieren eines Datenträgers kann mehrere Minuten dauern.

Beispiele für das Schreiben von Kennsätzen

1. Initialisierung eines Datenträgers (Mindestangaben):

   ```
   INIT TAPE-C4,VSN=VSN001
   ```

2. Initialisierung
   - von drei Datenträgern
   - mit einer Anweisung
   - mit explizit angegebenen Archivnummern
   - mit Sonderzeichen in den Archivnummern:

   ```
   INIT 3 TAPE-C4,VSN=(VSN#00,VSN$00,VSN@00)
   ```

3. Initialisierung
   - von 8 Datenträgern
   - mit einer Anweisung
   - mit fortlaufenden Archivnummern:

   ```
   INIT 8 TAPE-C4,VSN=(VSN001)
   ```

   Die Datenträger erhalten die Archivnummern VSN001, VSN002, ...,VSN008.

4. Initialisierung
   - eines fabrikneuen Datenträgers
   - auf einem bestimmten Gerät (mit mnemotechnischer Bezeichnung M1):

   ```
   INIT TAPE-C4,VSN=MBK01,UNIT=M1,NEW
   ```

5. Initialisierung
   - eines bestimmten Datenträgers (mit Archivnummer VSN001)
   - auf einem bestimmten Gerät (mit mnemotechnischer Bezeichnung T1)
   - im ISO7-Bit-Code
   - mit Entladen nach dem Initialisieren
   - mit Eintrag eines Eigentümerkennzeichens
   - mit Datensicherheitslöschen
   - mit Eintrag des Zeichens „0“ als Zugriffskennzeichen:

   ```
   INIT TAPE-C4,VSN=TAPE01,CHECK=VSN001,UNIT=T1,ISO7,RUN,-
   OWN=RZ#A0001,ERASE,ZERO
   ```

Beispiele für das Schreiben von zwei Abschnittsmarken

1. Schreiben von zwei Abschnittsmarken an den Anfang des Datenträgers mit Mindestangaben:

   ```
   INIT TAPE-C4,WTM
   ```

2. Schreiben von zwei Abschnittsmarken an den Anfang
   - von drei fabrikneuen Datenträgern
   - mit einer Anweisung:

   ```
   INIT 3 TAPE-C4,WTM,NEW
   ```

3. Schreiben von zwei Abschnittsmarken an den Anfang
   - eines bestimmten Datenträgers
   - auf einem bestimmten Gerät (mit mnemotechnischer Bezeichnung M1)
   - und mit Datensicherheitslöschen:

   ```
   INIT TAPE-C4,WTM,UNIT=M1,ERASE
   ```

4. Schreiben von zwei Abschnittsmarken an den Anfang
   - eines bestimmten Datenträgers (mit Archivnummer VSN001)
   - auf einem bestimmten Gerät (mit mnemotechnischer Bezeichnung M1)
   - mit Datensicherheitslöschen
   - und mit Entladen nach dem Initialisieren:

   ```
   INIT TAPE-C4,WTM,UNIT=M1,CHECK=VSN001,RUN
   ```

3.3.2 LIST - Magnetbandkennsätze anzeigen

Mit der Anweisung LIST werden Datenträgerkennsätze gelesen und angezeigt.Wahlweise kann zusätzlich festgelegt werden:

- die Anzahl der zu bearbeitenden Datenträger
- ein bestimmter Datenträger
- und ein bestimmtes Gerät

Bei Eingeschalteter OPTION CONS erfolgt die Ausgabe auf Bedienplatz, andernfalls auf SYSOUT.

> ¡ Die Funktion LIST wird nur ausgeführt, wenn die Task das Systemprivileg TAPE-ADMINISTRATION besitzt.

Format

LIST
[no] voltyp [,VSN = vsn / (vsn1,vsn2,...,vsn16) / (initval)] [,UNIT = mn] [,REW / RUN]

Operanden

no
Anzahl der Datenträger, die mit dieser Anweisung bearbeitet werden sollen. Gültige Werte: 1, 2, ..., 16.

> ¡ Zwischen den Operanden no und voltyp muss mindestens ein Leerzeichen stehen.

voltyp
Legt den Datenträgertyp (volume type) fest. Dieser definiert die Art des Datenträgers und die Aufzeichnungsart.

Die in der aktuellen BS2000-Version unterstützten Datenträgertypen finden Sie im Handbuch „Systeminstallation“ [7]. INIT unterstützt darüber hinaus aus Kompatibilitätsgründen einige weitere Datenträgertypen. Die von INIT akzeptierten Datenträgertypen können Sie mit der HELP-Anweisung (siehe "HELP - Kurzbeschreibung der INIT-Anweisungen anzeigen") ermitteln.

> ¡ Die Schlüsselwörter für den Datenträgertyp können abgekürzt werden, solange sie eindeutig bleiben, z.B. T-C4=TAPE-C4. Bei der Einführung neuer Datenträgertypen kann eine ursprünglich eindeutige Abkürzung möglicherweise mehrdeutig werden. Garantierte Abkürzungen gibt es nicht.

Da keine Voreinstellung wirksam wird, muss dieser Operand immer angegeben werden.

VSN
Die Kennsätze des Datenträgers mit der angegebenen Archivnummer sollen ausgegeben werden. Sollen mehrere Datenträger mit einer einzigen Anweisung bearbeitet werden, müssen entsprechend viele Archivnummern oder ein Anfangswert angegeben werden.

Voreinstellung: Wird der Operand VSN nicht angegeben, wird ein Datenträger vom Typ voltyp mit beliebiger Archivnummer angefordert.
Damit können die Kennsätze eines unbekannten Datenträgers gelesen werden.

VSN=vsn
Einzelne Archivnummer.
Die Kennsätze eines einzelnen Datenträgers sollen ausgegeben werden. Bei Angabe einer einzelnen Archivnummer muss der Operand "no" den Wert 1 haben oder er darf nicht angegeben sein.

Gültige Werte: Maximal 6 Zeichen (A..Z, 0..9, #,$,@).
Die von INIT erlaubten Sonderzeichen #,$ und @ dürfen bei normgerechten Kennsätzen (DIN 66029, DIN 66003) nicht verwendet werden. Die Angabe dieser Zeichen wird bei eingeschalteter OPTION DIN abgewiesen.

VSN=(vsn1,vsn2,...,vsn16)
Die Kennsätze der Datenträger mit den angegebenen Archivnummern sollen ausgegeben werden. Die entsprechenden Datenträger werden nacheinander angefordert. Die Anzahl der Archivnummern muss mit der Angabe bei dem Operanden "no" übereinstimmen. Gültige Werte siehe oben und "INIT - Kennsätze schreiben".

VSN=(initval)
Die Kennsätze mehrerer Datenträger, beginnend mit der Archivnummer anfwert, sollen ausgegeben werden. Nach jeder erfolgreichen Bearbeitung wird die Archivnummer automatisch um 1 erhöht. Die Anzahl der Datenträger muss mit dem Operanden "no" angegeben werden. Gültige Werte siehe oben und "INIT - Kennsätze schreiben".

UNIT=mn
Mnemotechnischer Gerätename zur definierten Belegung eines Gerätes. Ein Datenträger soll auf dem angegebenen Gerät montiert werden.
Ungeeignete Geräte, die z.B. den geforderten Datenträgertyp nicht bearbeiten können, werden abgewiesen.

REW
Legt fest, dass ein Magnetband bzw. eine Magnetbandkassette nach der Bearbeitung rückgespult, aber der Datenträger nicht entladen werden soll.
Bei der Bearbeitung eines einzelnen Datenträgers wird REW standardmäßig angenommen.

RUN
Legt fest, dass ein Magnetband bzw. eine Magnetbandkassette nach der Bearbeitung rückgespult und der Datenträger entladen werden soll.
Bei der Bearbeitung mehrerer Datenträger mit einer einzigen Anweisung (Operand "no" = 2..16) wird RUN standardmäßig angenommen.

Beispiele

1. Ausgabe der Kennsätze eines einzelnen Datenträgers mit beliebiger Archivnummer (Mindestangaben):

   ```
   LIST TAPE-C4
   ```

2. Ausgabe der Kennsätze von drei bestimmten Datenträgern mit einer Anweisung:

   ```
   LIST 3 TAPE-C4,VSN=(TAPE01,TAPE02,TAPE03)
   ```

3. Ausgabe der Kennsätze von 8 bestimmten Datenträgern mit einer Anweisung:

   ```
   LIST 8 TAPE-C4,VSN=(VSN001)
   ```

   Die Datenträger mit den Archivnummern VSN001, VSN002,...,VSN008 werden nacheinander angefordert.

4. Ausgabe der Kennsätze
   - eines bestimmten Datenträgers (mit Archivnummer VSN001),
   - auf einem bestimmten Gerät (mit mnemotechnischer Bezeichnung M1),
   - und mit Entladen nach dem Initialisieren:

   ```
   LIST TAPE-C4,VSN=VSN001,UNIT=M1,RUN
   ```

3.3.3 OPTION - Programmfunktionen ein- und ausschalten

Die Anweisung OPTION dient dazu, Zusatzfunktionen ein- und auszuschalten, die für alle folgenden Operationen INIT oder LIST gültig sind. Sie bleiben bis zum Programmende oder bis zum Rücksetzen in Kraft.

Format

OPTION
CONS / PROT / DIN / NOHDR / NONE

Operanden

CONS
Umschalten der Ein-/Ausgaben auf Bedienplatz, nach Starten des Programms vom Bedienplatz mittels ENTER-Auftrag.

> ℹ Die Angabe von CONS ist nur innerhalb eines ENTER-Auftrags zulässig und wird im Dialogbetrieb abgewiesen.

PROT
Abbrechen der Initialisierung eines Datenträgers, wenn beim Kontrolllesen eine Zugriffsbeschränkung gefunden wurde.

Als Zugriffsbeschränkung gilt jedes der folgenden Kriterien:

- Der Zugriffsvermerk im Datenträgerkennsatz VOL1 erlaubt Zugriffe auf den Datenträger nur durch den Eigentümer (Byte 11 im VOL1-Kennsatz enthält weder '0' noch ' ' (Blank)).
- Der Zugriffsvermerk im Dateikennsatz der ersten Datei erlaubt den Zugriff auf die Datei nur durch den Eigentümer (Byte 54 im HDR1-Kennsatz enthält weder '0' noch ' ' (Blank)).
- Das Verfallsdatum der ersten Datei ist noch nicht erreicht (Byte 48-53 im HDR1-Kennsatz).
- Die erste Datei ist mit einem Lese- und/oder einem Schreibkennwort geschützt (Byte 57-64 im HDR3-Kennsatz).
- Für die erste Datei ist nur Lesezugriff gestattet (Byte 69 im HDR3-Kennsatz enthält eine '1').

In allen genannten Fällen wird nach dem Lesen und der Ausgabe der vorhandenen Kennsätze eine entsprechende Meldung ausgegeben und erneut zur Eingabe einer Anweisung aufgefordert.
Soll der betreffende Datenträger dennoch initialisiert werden, muss der Schreibschutz durch die Anweisung OPTION NONE ausgeschaltet werden.

DIN
Ausschluss von Sonderzeichen, die nicht DIN 66003 entsprechen, von einer Verwendung in Kennsätzen. Sollen die Kennsätze dennoch Sonderzeichen enthalten (#, $, @), muss diese Zeichenauswahl-Begrenzung durch OPTION NONE wieder ausgeschaltet werden.

NOHDR
Initialisierung eines Datenträgers nur mit dem Datenträgerkennsatz VOL1. Nach Einschalten dieser Option werden alle folgenden Initialisierungen bis zum Rücksetzen ohne Schreiben der HDR-Kennsätze durchgeführt. Soll das Initialisieren wieder mit HDR1- und HDR2-Kennsatz erfolgen, muss die eingeschränkte Initialisierung durch OPTION NONE wieder ausgeschaltet werden.

NONE
Rücksetzen aller Optionen.

> Die Option CONS kann nicht zurückgesetzt werden.

Beispiele

1. OPTION PROT

```
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   OPTION PROT                                                    a)
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   INIT TAPE-C4,VSN=TEST01                                        b)
(OUT)  %  NVI0003 LABELS ON TAPE       :                              c)
(OUT)  %  NVI0004 VOL1 LABEL:
                  'VOL1DAR15K                                  '
                  '                                           1'
(OUT)  %  NVI0004 HDR1 LABEL:
                 'HDR1V11.TEST-R.TXT.TEDAR15K0001000100010'
                 '0 92104 92107 000000BS2000                  '
(OUT)  %  NVI0101 EXPIRATION DATE OF FIRST FILE NOT YET REACHED       d)
(OUT)  %  NVI0004 HDR2 LABEL:
                 'HDR2V8000102044 0                P     '
                 '      .C..04                           '
(OUT)  %  NVI0004 HDR3 LABEL:
                 'HDR3TSOS    V11.TEST-R.TXT.TESTFILE.T-C'
                 '4               ............00          '
(OUT)  %  NVI0201 TAPE PROTECTED.                                     e)
                  INITIALIZATION REQUEST REJECTED
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
```

   a. OPTION PROT wird gesetzt.
   b. Es soll ein Datenträger vom Typ TAPE-C4 initialisiert werden.
   c. Die Kennsätze des Datenträgers werden ausgegeben. Das Verfallsdatum der ersten Datei auf dem Datenträger ist noch nicht abgelaufen (Byte 48-53 des HDR1-Kennsatzes).
   d. Eine entsprechende Meldung wird ausgegeben.
   e. Es werden keine neuen Kennsätze geschrieben. Die Initialisierung wird abgebrochen.

2. OPTION DIN

```
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   OPTION DIN                                                  a)
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   LIST TAPE-C4,VSN=T#$@                                       b)
(OUT)  %  NVI0405 INVALID OPERAND VALUE ',VSN=T#'
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   INIT TAPE-C4,VSN=DAR10A,OWN=TEST#$@                         c)
(OUT)  %  NVI0405 INVALID VALUE ',OWN=TEST#'
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
```

   a. OPTION DIN wird gesetzt.

b. Es sollen die Kennsätze eines Datenträgers vom Typ TAPE-C4 ausgegeben werden, dessen Archivnummer Sonderzeichen enthält. Diese Anweisung wird zurückgewiesen.

c. Es soll ein Datenträger vom Typ TAPE-C4 initialisiert werden und ein Eigentümername in den VOL1-Kennsatz eingetragen werden, der Sonderzeichen enthält. Diese Anweisung wird zurückgewiesen.

1. OPTION NOHDR

```
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   OPTION NOHDR                                                    a)
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)  INIT TAPE-C4,VSN=TAPE01,CHECK=TAPE01                             b)
(OUT)  %  NVI0003 LABELS ON TAPE                                       c)
(OUT)  %  NVI0004 VOL1 LABEL: 'VOL1TAPE01...
(OUT)  %  NVI0004 HDR1 LABEL: 'HDR1      ...
(OUT)  %  NVI0004 HDR2 LABEL: 'HDR2U00001...
(OUT)  %  NVI0007 OVERWRITE TAPE ?                                     d)
                  REPLY (YES=YES  N=NO)
(IN)   YES
(OUT)  %  NVI0010 INITIALIZATION OF TAPE 'TAPE01'
             ON DEVICE 'TA' COMPLETED
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
(IN)   LIST TAPE-C4,VSN=TAPE01                                         e)
(OUT)  %  NVI0003 LABELS ON TAPE       :                               f)
(OUT)  %  NVI0004 VOL1 LABEL: 'VOL1TAPE01...
(OUT)  %  NVI0001 ENTER COMMAND (END = TERMINATE INIT)
```

a. OPTION NOHDR wird gesetzt.

b. Es soll ein Datenträger initialisiert werden.

c. Die auf dem Datenträger enthaltenen Kennsätze VOL1, HDR1 und HDR2 werden ausgegeben.

d. Der Datenträger wird neu initialisiert.

e. Die Kennsätze sollen ausgegeben werden.

f. Der Datenträger wurde nur mit einem VOL1-Kennsatz initialisiert.

3.3.4 END - Beenden des Programmlaufs

Die Anweisung END beendet das Programm INIT. Das INIT-Subsystem wird zum Entladen freigegeben.
In Kommandoprozeduren wird das Programm INIT auch bei fehlendem END beendet, sobald ein BS2000-Kommando gelesen wird (beginnend mit /) oder das Dateiende erreicht ist.

Format

END / E

3.3.5 HELP - Kurzbeschreibung der INIT-Anweisungen anzeigen

Die Anweisung HELP bewirkt die Ausgabe von Kurzinformationen zu allen Operationen und Operanden des Programms INIT.
Bei eingeschalteter OPTION CONS erfolgt die Ausgabe auf Bedienplatz, andernfalls auf SYSOUT.

Format

HELP / H

3.4 Aufbau der Kennsätze

Das allgemeine Format der Kennsätze ist an die DIN-Norm DIN 66029 (Magnetbänder) angepasst. Diese Kennsätze werden auch für andere Datenträger verwendet, die in BS2000 als Magnetbänder bzw. Magnetbandkassetten emuliert werden.

Magnetbandkennsätze haben eine Länge von 80 Zeichen.

Im Folgenden ist nur der Aufbau und Inhalt derjenigen Magnetband-Kennsätze beschrieben, die von INIT geschrieben oder gelesen werden.

In der Spalte 'Feldinhalt' ist der von INIT geschriebene Inhalt des betreffenden Feldes aufgeführt (Beim HDR3-Kennsatz der mögliche Inhalt).

Zeichenerklärung

x: aktueller Wert

_: Leerzeichen

Die übrigen Felder enthalten feste Voreinstellungen.

3.4.1 Datenträgerkennsatz VOL1 für Magnetbänder

Stelle	Feldname	Länge	Feldinhalt
1 - 3	Kennsatzname	3	VOL
4	Kennsatznummer	1	1
5 - 10	Archivnummer <vsn>	6	xxxxxx
11	Zugriffsvermerk <zer>	1	_ oder 0
13 - 37	- reserviert -	26	_..._
38 - 51	Eigentümerkennz. <owner>	14	_ _ _ _xxxxxxxx_ _
52 - 79	- reserviert -	28	_..._
80	Normvermerk	1	1

V	O	L	1	vsn						flg	_	_	_	_	_
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
_	_	_	_	_	_	_	_	_	owner						
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	1
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

3.4.2 Dateikennsatz HDR1 für Magnetbänder

Stelle	Feldname	Länge	Feldinhalt
1 - 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	1
5 - 21	Dateiname	17	_..._
22 - 27	Archivnummer <vsn>	6	xxxxxx
28 - 31	Dateiabschnittsnummer	4	0001
32 - 35	Dateifolgenummer	4	0001
36 - 39	Generationsnummer	4	0001
40 - 41	Versionsnummer	2	00
42 - 47	Erstellungsdatum <credat>	6	xxxxxx
48 - 53	Verfallsdatum <reldat>	6	xxxxxx
54	Zugriffsvermerk	1	_
55 - 60	Blockzähler	6	000000
61 - 73	System-Code	13	_..._
74 - 80	- reserviert -	7	_..._

Format des Erstellungs- und Verfallsdatums : cyyjjj

c Kennzeichen für Jahrhundert (century):
Leerzeichen = „20. Jahrhundert“; 0 = „21. Jahrhundert“

yy Jahresangabe (year).

jjj Julianisches Datum (Nummer des Tages im Jahr).

Beispiele

3. Juli 1998: _98185

10. Januar 2004: 004010

H	D	R	1	_	_	_	_	_	_	_	_	_	_	_	_
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
_	_	_	_	_	vsn						0	0	0	1	0
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
0	0	1	0	0	0	1	0	0	credat						
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
reldat					_	0	0	0	0	0	0	_	_	_	_
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

3.4.3 Dateikennsatz HDR2 für Magnetbänder

Stelle	Feldname	Länge	Feldinhalt
1 - 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	2
5	Satzformat	1	U
6 - 10	Blocklänge	5	00001
11 - 15	Satzlänge	5	00001
16 - 50	- reserviert -	35	_..._
51 - 52	Pufferverschiebung	2	00
52 - 80	- reserviert -	28	_..._

H	D	R	2	U	0	0	0	0	1	0	0	0	0	1	_
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
_	_	0	0	_	_	_	_	_	_	_	_	_	_	_	_
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

3.4.4 Dateikennsatz HDR3 für Magnetbänder

Der Dateikennsatz HDR3 wird beim Initialisieren in keinem Fall geschrieben. Er wird aber, falls vorhanden, beim Lesen der Kennsätze ausgegeben.

Stelle	Feldname	Länge	Feldinhalt
1 - 3	Kennsatzname	3	HDR
4	Kennsatznummer	1	3
5 - 12	Eigentümerkennung <uid>	8	xxxxxxxx
13 - 56	Dateiname <fnam>	44	xxx...xxx
57 - 60	Lesekennwort <rpass>	4	xxxx oder X' 00000000'
61 - 64	Schreibkennwort <wpass>	4	xxxx oder X' 00000000'
65 - 68	Ablaufkennwort <epass>	4	xxxx oder X' 00000000'
69	Zugriffsart <acc>	1	0 = read + write 1 = read only
70 - 80	- reserviert -	11	_..._

H	D	R	3	uid								fnam			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
fnam															
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
fnam															
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
fnam								rpass				wpass			
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
epass				acc	_	_	_	_	_	_	_	_	_	_	_
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80

4 IORM Dynamische Steuerung von I/O-Ressourcen

Versionsstand: **IORM V21.0C**

Privilegierung: **TSOS** oder **OPERATING**

Das Dienstprogramm IORM wird mit `/START-IORM` (Privileg TSOS oder OPERATING) gestartet.

Das Dienstprogramm IORM (**I/O R**essource **M**anagement) verbessert die I/O-Eigenschaften von BS2000 im Native- und im VM2000-Betrieb.

In IORM sind folgende Funktionen zur autonomen, dynamischen Steuerung der I/O-Betriebsmittel Kanal, Steuerung, Pfad und Gerät realisiert:

- IOPT: I/O-Prioritäten-Steuerung für Tasks (I/O Priority Handling for Tasks)
-
- DDAL: Optimierte Lastverteilung im Betrieb von ETERNUS CS
  (Dynamic Device Allocation)
- IOLVM (/390-Server):
  Begrenzung der I/O-Aufnahme einzelner virtueller Maschinen (I/O Limit for Virtual Machines)

In IORM gibt es noch die folgende Funktion:

- DPAV (/390-Server) Einstellungen für PAV und FastDPAV:
  Jetzt nicht mehr unterstützt zur Benutzung als Dynamische I/O-Lastverteilung für Platten (Dynamic Parallel Access Volume), sondern ausschließlich zum Monitoring der Auslastung und zur Begrenzung der FastDPAV Alias-Geräte-Anzahl für einzelne Platten

IORM sammelt im laufenden Betrieb Daten über die Auslastung der I/O-Betriebsmittel und steuert den I/O-Betrieb abhängig von vorgegebenen Grenzwerten.

Die IORM-Funktionen IOPT und IOLVM steuern Plattengeräte.
Die IORM-Funktion DDAL steuert Bandgeräte, siehe "IORM Dynamische Steuerung von I/O-Ressourcen".

Wenn im VM2000-Betrieb IORM im Monitorsystem und in den beteiligten BS2000-Gastsystemen im Einsatz ist, dann tauschen die IORM-Subsysteme über eine interne Schnittstelle I/O-Daten und Kontrollinformationen aus.

IORM kann im Native-Betrieb und unter VM2000 auf allen BS2000-Servern eingesetzt werden. IORM arbeitet VM-übergreifend, aber nicht Server-übergreifend.

Installation von IORM

IORM wird mit dem Installationsmonitor IMON installiert.

`SYSLNK.IORM.<version>`	Modulbibliothek für /390-Server
`SKMLNK.IORM.<version>`	Modulbibliothek für x86-Server
`SYSMES.IORM.<version>`	Meldungsdatei
`SYSSDF.IORM.<version>`	SDF-Syntaxdatei
`SYSSSC.IORM.<version>`	Subsystemkatalog

SYSDAT.IORM.<version>	Parameterdatei
SYSSII.IORM.<version>	Strukturinformation
SYSNRF.IORM.<version>	NOREF-Datei
SYSRMS.IORM.<version>	RMS-Liefermenge
SYSRME.IORM.<version>.D/E	Readme-Datei (deutsch/englisch)
SYSFGM.IORM.<version>.D/E	Freigabemitteilung (deutsch/englisch)

Tabelle 2: Lieferbestandteile von IORM

Parameterdatei von IORM

In der Parameterdatei $<benutzerkennung>.SYSDAT.IORM.<version> können Sie Vorgaben für IORM treffen, die beim ersten Start von IORM verarbeitet werden. Die Parameterdatei besteht aus Kommentarzeilen und Wertezeilen mit Anweisungen für IORM. Bei Auslieferung enthält die Parameterdatei Kommentarzeilen mit Musteranweisungen für IORM.

Subsystem IORM

Das Subsystem IORM muss vor dem ersten Start des Dienstprogramms IORM geladen werden mit:

```
/START-SUBSYSTEM SUBSYSTEM-NAME=IORM
```

Das Subsystem IORM wird beendet durch:

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=IORM
```

Wenn das Dienstprogramm IORM zu diesem Zeitpunkt noch benutzt wird, dann verbleibt das Subsystem solange im Zustand IN DELETE / WAIT-DISCON, bis die Benutzung endet.

Start des Dienstprogramms IORM

Das Dienstprogramm IORM wird mit /START-IORM (Privileg TSOS oder OPERATING) gestartet.

START-IORM	Alias: **IORM**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Beim ersten Start des Dienstprogramms IORM werden die für den IORM-Betrieb erforderlichen Aktionen unter der Benutzerkennung, in der /START-IORM eingegeben wurde, ausgeführt: Einrichten von Tabellen, Anschluss an das BS2000-I/O-System, Bearbeitung der Anweisungen aus der IORM-Parameterdatei. Danach beendet sich IORM. Im VM2000-Betrieb sollte IORM in allen BS2000-Gastsystemen gestartet werden.

Bei einem weiteren Start von IORM können Sie Anweisungen an IORM über SYSDTA eingeben. Diese Anweisungen steuern den Ablauf von IORM. Die Anweisungen können direkt im Dialog oder in Form einer Datei eingegeben werden.

Beispiel

```
/START-SUBSYSTEM SUBSYSTEM-NAME=IORM                                    1.
ESM0216 FUNCTION 'CREATE' STARTED FOR SUBSYSTEM 'IORM    /<version>'
/...
/start-iorm                                                             2.
BLS0523 ELEMENT 'IORM-TU', VERSION '<version>', TYPE 'L' FROM LIBRARY
        ':SBZ7:$TSOS.SYSLNK.IORM.<version>' IN PROCESS
BLS0524 LLM 'IORM-TU', VERSION '<version>' OF '<date> <time>' LOADED
/start-iorm                                                             3.
BLS0523 ELEMENT 'IORM-TU', VERSION '<version>', TYPE 'L' FROM LIBRARY
        ':SBZ7:$TSOS.SYSLNK.IORM.<version>' IN PROCESS
BLS0524 LLM 'IORM-TU', VERSION '<version>' OF '<date> <time>' LOADED
* iopt_set_on?
IOPT_SET_ON=NO                                                          4.
* ...
//END
/...
/ASSIGN-SYSDTA TO=SYSDAT.IORM.NIGHT                                    5.
/START-IORM
BLS0523 ELEMENT 'IORM-TU', VERSION '<version>', TYPE 'L' FROM LIBRARY
        ':SBZ7:$TSOS.SYSLNK.IORM.<version>' IN PROCESS
BLS0524 LLM 'IORM-TU', VERSION '<version>' OF '<date> <time>' LOADED
/...
/ASSIGN-SYSDTA TO=SYSDAT.IORM.DAY                                       6.
/START-IORM
BLS0523 ELEMENT 'IORM-TU', VERSION '<version>', TYPE 'L' FROM LIBRARY
        ':SBZ7:$TSOS.SYSLNK.IORM.<version>' IN PROCESS
BLS0524 LLM 'IORM-TU', VERSION '<version>' OF '<date> <time>' LOADED
/...
```

1. Das Subsystem IORM wird gestartet.
2. Das Dienstprogramm IORM wird zur Initialisierung erstmals gestartet.
3. Das Dienstprogramm IORM wird zur Eingabe von Anweisungen nochmals gestartet.
4. Es gelten die Vorgaben aus der Parameterdatei.
5. Am Abend erhält das bereits aktive Subsystem IORM neue Vorgaben aus der Datei `SYSDAT.IORM.NIGHT`.
6. Am nächsten Morgen erhält das noch aktive Subsystem IORM neue Vorgaben aus der Datei `SYSDAT.IORM.DAY`.

Anweisungen an IORM

IORM-Anweisungen bestehen aus Zeichen- und Ziffernfolgen ohne Leerzeichen.Kommentarzeilen beginnen mit `*`.

IORM kennt folgende allgemeine Anweisungen:

END	Beendet das Dienstprogramm IORM. Das Subsystem IORM mit seinen Funktionen bleibt aktiv.
HELP	Gibt die IORM-Anweisungen aus.
IORM_DUMP	Erstellt Diagnose-Unterlagen von IORM.

Die weiteren Anweisungen sind funktionsspezifisch und werden bei der jeweiligen Funktion beschrieben.

Hardware-Abhängigkeiten von IORM

IORM bedient folgende Hardwarekomponenten:

IORM-Funktion	Bediente Hardware
IOPT	Platten an allen BS2000-Servern
DPAV	Platten am Fibre Channel der /390-Server Am Fibre Channel der x86-Server steht DPAV nicht zur Verfügung. Hier wird für emulierte Platten die IO-Parallelität unterstützt (RSC).
DDAL	Bandgeräte an allen BS2000-Servern
IOLVM	Platten an allen BS2000-Servern

Verhalten im Fehlerfall, Erstellung von Diagnoseunterlagen

Zur Diagnose und Behebung von IORM-Problemen ist es notwendig, dass ausreichende Fehlerunterlagen zum frühestmöglichen Zeitpunkt erstellt und gesichert werden.
Bei reproduzierbaren Fehlern sollte genau beschrieben werden, wie der Fehler erzeugt werden kann.

Zusätzlich zur CONSLOG-Datei und den IORM-Anweisungen sollte stets eine Datei mit dem IORM-Dump mit folgender Anweisungsfolge erstellt werden:

```
/ASSIGN-SYSLST TO=<file>
/START-IORM
* IORM_DUMP
* END
/ASSIGN-SYSLST TO=*PRIMARY
```

Bei Problemen mit DPAV oder DDAL im VM2000-Betrieb werden diese Unterlagen auch aus dem Monitorsystem und den beteiligten Gastsystemen benötigt.

4.1 IOPT: I/O-Prioritäten-Steuerung für Tasks

Eine I/O-intensive Anwendung mit niedriger Priorität kann eine andere, höher-priore Anwendung behindern, wenn diese Anwendungen I/Os auf das gleiche (logische) Gerät ausführen. Behinderungen können auch entstehen, wenn die I/Os auf verschiedene (logische) Geräte ausgeführt werden, die jedoch auf demselben physikalischen Gerät liegen oder über dieselben Pfade angeschlossen, über dieselben Ports erreichbar oder an denselben Kanälen angeschlossen sind.

Verbesserte Prioritätensteuerung für Tasks mit IOPT

IORM kann in der Funktion IOPT (**I/O P**riority Handling for **T**asks) solche Konfliktsituationen erkennen und steuernd in den I/O-Betrieb eingreifen.

Dazu betrachtet IOPT sowohl den Auslastungsgrad der I/O-Einheiten (Geräte, Pfade, Ports und Kanäle) als auch die I/O-Prioritäten der nutzenden Tasks.

I/O-Prioritäten für Tasks

Es gibt drei I/O-Prioritäten für Tasks:

- HIGH (hohe I/O-Priorität)
- MEDIUM (mittlere I/O-Priorität)
- LOW (niedrige I/O-Priorität)

Die I/O-Priorität für Tasks kann auf zwei Arten festgelegt werden:

- Mit `/MODIFY-TASK-CATEGORIES`, Operand `IO-PRIORITY=*HIGH/*MEDIUM/*LOW`. Dies ist die empfohlene Vorgehensweise.

  Das Kommando legt die I/O-Priorität für die Task-Kategorien fest.
  Neben den vier Standardkategorien SYS, TP, DIALOG und BATCH kann es weitere Kategorien geben, deren Namen bei der Jobklassen-Definition vereinbart werden.
- Mit den IORM-Anweisungen IOPT_PRI_HIGH und IOPT_PRI_MED. Diese Einstellungen wirken aber nur, falls `IO-PRIORITY=*NONE` bei `/MODIFY-TASK-CATEGORIES` wirksam ist (Standardwert).

  Die IORM-Anweisungen legen die I/O-Priorität aufgrund der Task-Prioritäten fest. Die Task-Prioritäten werden bei der Jobklassen-Definition und benutzerspezifisch im Benutzerkatalog festgelegt.

Vorgaben für die I/O-Einheiten

Mit der IORM-Anweisung IOPT_DEV_ADD können Plattengeräte festgelegt werden, auf die die IORM-Prioritätensteuerung Anwendung finden soll.

Mit den IORM-Anweisungen IOPT_LOW_xxx bzw. IOPT_MED_xxx können Grenzwerte bzw. I/O-Anteile für Tasks mit der Priorität LOW und MEDIUM in Bezug auf die I/O-Einheiten Gerät (xxx=DEV), Pfad (xxx=PTH), Port (xxx=POR) und Kanal (xxx=CHN) festgelegt werden.

Die in IORM dafür eingestellten Standardwerte entsprechen den Angaben in der ausgelieferten Parameterdatei `SYSDAT.IORM.<version>`.

> **i** Das deutlich schnellere Ansteigen der I/O-Auslastung an einem Gerät im Vergleich zu einem Pfad, Port oder Kanal, insbesondere bei mehrpfadigem Geräte-Anschluss, sollte bei den Vorgaben für die I/O-Einheiten berücksichtigt werden.
> Es kann vorkommen, dass IORM eine Konfliktsituation auf einer physikalischen Platte nicht schnell genug erkennt, weil die Auslastung für die Pfade, Ports, Kanäle und logischen Geräte noch unterhalb der vorgegebenen Grenzwerte liegt. In diesem Fall kann es sinnvoll sein, alle logischen Geräte auf der physikalischen Platte zu einer Gerätegruppe in IORM zusammenzufassen (IOPT_GRP_ADD).

Ablauf von IOPT

IOPT ist zunächst deaktiviert. Nach seiner Aktivierung (IOPT_SET_ON=YES) sammelt IOPT für alle bekannten I/O-Einheiten laufend Auslastungswerte getrennt nach den I/O-Prioritäten HIGH / MEDIUM / LOW.

IOPT prüft periodisch, ob Tasks mit der I/O-Priorität LOW oder MEDIUM höher-priore Tasks (I/O-Priorität MEDIUM oder HIGH) in einer I/O-Einheit behindern. Wenn dies der Fall ist, dann wird für Tasks der betroffenen I/O-Priorität nur noch ein begrenzter Auslastungsgrad auf der I/O-Einheit zugelassen. Es werden damit in Abhängigkeit von der Auslastung der I/O-Einheit Tasks mit geringerer I/O-Priorität bei der Ausführung von I/Os auf Geräte, die zuvor mit IOPT_DEV_ADD für IOPT aktiviert wurden, „gebremst“.

IOPT wirkt nur lokal, sowohl im Native-Betrieb, als auch im BS2000-Gastsystem. Es werden nur Plattengeräte berücksichtigt.
Tasks für FDDRL, ARCHIVE, VOLIN und PAGING werden nicht gebremst.

4.1.1 IOPT-Anweisungen

Standardwerte in den Syntaxkästen sind unterstrichen dargestellt.

4.1.1.1 IOPT aktivieren und deaktivieren

IOPT_SET_ON aktiviert oder deaktiviert IOPT. Außerdem kann die aktuelle Einstellung abgefragt werden.

IOPT_SET_ON
IOPT_SET_ON=YES IOPT_SET_ON=NO IOPT_SET_ON=CHK IOPT_SET_ON?

YES IOPT wird im System aktiviert. Im VM-Betrieb ist diese Einstellung in allen Systemen erforderlich, in denen IOPT aktiv sein soll.

NO IOPT wird im System deaktiviert. Im VM-Betrieb ist diese Einstellung in allen Systemen erforderlich, in denen IOPT deaktiviert werden soll.

CHK IOPT läuft im Probebetrieb, d.h. IOPT greift nicht in die I/O-Steuerung ein, aber es kann ermittelt werden, welche Eingriffe IOPT bei einer Aktivierung vornehmen würde. Damit kann überprüft werden, ob ein Einsatz von IOPT sinnvoll ist, siehe Abschnitt „Prüfbetrieb“.

? Die aktuelle Einstellung wird abgefragt.

4.1.1.2 Grenzwerte für I/O-Prioritätsklassen festlegen und abfragen

IOPT_PRI_HIGH und IOPT_PRI_MED definieren die Grenzwerte für die Tasks mit hoher, mittlerer und niedriger I/O-Priorität. Die I/O-Priorität eines Tasks wird dabei aus seiner Task-Priorität abgeleitet. Die aktuell gültigen Grenzwerte können abgefragt werden.

IOPT_PRI
IOPT_PRI_HIGH=x IOPT_PRI_MED=y IOPT_PRI_HIGH? IOPT_PRI_MED?

x Grenzwert für die I/O-Priorität HIGH. Alle Tasks mit einer Task-Priorität, die kleiner oder gleich x ist, gehören zur Klasse HIGH (Prio <= x).
Wertebereich: 0 <= x <= 255
Standardwert: 155

y Grenzwert für die I/O-Priorität MEDIUM und LOW. Alle Tasks mit einer I/O-Priorität, die größer x und kleiner oder gleich y ist, gehören zur Klasse MEDIUM (x < Prio <= y). Alle Tasks mit einer Task-Priorität, die größer y ist, gehören zur Klasse LOW (y < Prio).
Wertebereich: x < y <= 255
Standardwert: 205

? Die Grenzwerte für die Klassen HIGH und MEDIUM werden abgefragt.

Die Grenzwerte sind nur dann wirksam, wenn bei `/MODIFY-TASK-CATEGORIES` der Operand `IO-PRIORITY` den Wert `*NONE` hat. Allerdings ist eine Festlegung der I/O-Prioritäten über `/MODIFY-TASK-CATEGORIES` empfehlenwert, weil dann die I/O-Priorität unabhängig von der Task-Priorität vergeben werden kann. So kann es wünschenswert sein, dass zwei Tasks mit gleicher Task-Priorität unterschiedliche I/O-Priorität haben, weil sie unterschiedlich I/O-intensiv sind.

4.1.1.3 Plattengeräte für IOPT aktivieren und deaktivieren

Plattengeräte für IOPT aktivieren

IOPT_DEV_ADD aktiviert logische Geräte für IOPT. Die Geräte werden dann durch IOPT überwacht.

IOPT_DEV_ADD
IOPT_DEV_ADD=ALL IOPT_DEV_ADD=D-R(mn1,mn2)

ALL Alle Geräte werden für IOPT aktiviert.

D-R(mn1,mn2)

Alle Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden aktiviert. Wenn mn1 und mn2 gleich sind, wird nur ein Gerät aktiviert.

Plattengeräte für IOPT deaktivieren

IOPT_DEV_REM deaktiviert logische Geräte für IOPT. Die Geräte werden durch IOPT nicht (mehr) überwacht

IOPT_DEV_REM
IOPT_DEV_REM=ALL IOPT_DEV_REM=D-R(mn1,mn2)

ALL Alle Geräte werden für IOPT deaktiviert.

D-R(mn1,mn2)

Alle Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden deaktiviert. Wenn mn1 und mn2 gleich sind, wird nur ein Gerät deaktiviert.

Aktivierte Geräte abfragen

IOPT_DEV_ADD? gibt eine Liste aller Geräte aus, die mit IOPT_DEV_ADD für IOPT aktiviert wurden.

IOPT_DEV_ADD?
IOPT_DEV_ADD?mask

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens.
Folgende Angaben sind möglich:

* Alle aktivierten Geräte werden aufgelistet.

n* Alle aktivierten Geräte, deren Namen mit n beginnen, werden aufgelistet.

nm* Alle aktivierten Geräte, deren Namen mit nm beginnen, werden aufgelistet.

nml* Alle aktivierten Geräte, deren Namen mit nml beginnen, werden aufgelistet.

nmlo Das Gerät mit dem Namen nmlo wird aufgelistet.

4.1.1.4 Gerätegruppen

Geräte zu einer Gerätegruppe hinzufügen

IOPT_GRP_ADD fügt logische Geräte zu einer Gerätegruppe hinzu. Wenn eine Gruppe mit dem angegebenen Namen nicht existiert, wird sie neu eingerichtet. Nur Geräte, die (zuvor oder danach) mit IOPT_DEV_ADD für IOPT aktiviert werden, werden auch von IOPT überwacht. Die Festlegung einer Gerätegruppe allein reicht für eine Aktivierung nicht aus.

IOPT überwacht eine Gerätegruppe wie ein Gerät, betrachtet also die Auslastung der gesamten Gruppe. Beispielsweise können alle logischen Geräte, die einer physikalischen Platte zugeordnet sind, eine Gerätegruppe bilden. IOPT erkennt auf diese Weise I/O-Engpässe auf der Platte auch dann, wenn bei den einzelnen logischen Geräten die Auslastungsgrenzwerte noch nicht überschritten sind.

IOPT_GRP_ADD
IOPT_GRP_xxx_ADD=D-R(mn1.mn2)

xxx Dreistelliger numerischer Name der Gerätegruppe.
Wertebereich: 000 < x <= 255

D-R(mn1,mn2)

Alle Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden zu der Gruppe hinzugefügt. Wenn mn1 und mn2 gleich sind, wird nur ein Gerät zur Gruppe hinzugefügt.

Geräte aus einer Gerätegruppe entfernen

IOPT_GRP_REM entfernt logische Geräte aus einer Gerätegruppe. Wenn das letzte Gerät aus einer Gruppe entfernt wird, dann wird auch die Gruppe gelöscht.

IOPT_GRP_REM
IOPT_GRP_xxx_REM=ALL IOPT_GRP_xxx_REM=D-R(mn1,mn2)

xxx Dreistelliger numerischer Name der Gerätegruppe.
Wertebereich: 000 < x <= 255

ALL Alle Geräte der Gruppe werden entfernt. Die Gruppe wird gelöscht.

D-R(mn1,mn2)

Alle Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden aus der Gruppe entfernt. Wenn mn1 und mn2 gleich sind, wird nur ein Gerät aus der Gruppe entfernt.

Geräte einer Gerätegruppe abfragen

IOPT_GRP_ADD? gibt eine Liste aller Geräte aus, die zu einer Gerätegruppe gehören.

IOPT_GRP_ADD?
IOPT_GRP_xxx_ADD?mask

xxx Dreistelliger numerischer Name der Gerätegruppe.
Wertebereich: 000 < x <= 255

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens.
Folgende Angaben sind möglich:

- `*` Alle Geräte der Gruppe werden aufgelistet.
- `n*` Alle Geräte der Gruppe, deren Namen mit n beginnen, werden aufgelistet.
- `nm*` Alle Geräte der Gruppe, deren Namen mit nm beginnen, werden aufgelistet.
- `nml*` Alle Geräte der Gruppe, deren Namen mit nml beginnen, werden aufgelistet.
- `nmlo` Das Gerät der Gruppe mit dem Namen nmlo wird aufgelistet.

4.1.1.5 Grenzwerte für I/O-Priorität LOW festlegen

Grenzwerte für I/O-Priorität LOW und Kanäle festlegen

IOPT_LOW_CHN legt Grenzwerte für die Kanalauslastung fest. Sobald die Kanalauslastung mit Tasks der Priorität HIGH und MEDIUM die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität LOW beschränkt.

IOPT_LOW_CHN
IOPT_LOW_CHN_1HM=20 / <integer 0...100>
IOPT_LOW_CHN_2HM=30 / <integer 0...100>
IOPT_LOW_CHN_3HM=40 / <integer 0...100>
IOPT_LOW_CHN_1QH=40 / <integer 0...100>
IOPT_LOW_CHN_2QH=30 / <integer 0...100>
IOPT_LOW_CHN_3QH=20 / <integer 0...100>
IOPT_LOW_CHN_1QM=55 / <integer 0...100>
IOPT_LOW_CHN_2QM=40 / <integer 0...100>
IOPT_LOW_CHN_3QM=30 / <integer 0...100>

IOPT_LOW_CHN_1HM, IOPT_LOW_CHN_2HM, IOPT_LOW_CHN_3HM

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH und MEDIUM.

IOPT_LOW_CHN_1QH, IOPT_LOW_CHN_2QH, IOPT_LOW_CHN_3QH

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei hohem Anteil an Tasks mit Priorität HIGH.

IOPT_LOW_CHN_1QM, IOPT_LOW_CHN_2QM, IOPT_LOW_CHN_3QM

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei niedrigem Anteil an Tasks mit Priorität HIGH.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM kleiner IOPT_LOW_CHN_1HM ist, so werden Tasks der Priorität LOW nicht beschränkt
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_CHN_1HM und IOPT_LOW_CHN_2HM ist, so begrenzen IOPT_LOW_CHN_1QH oder IOPT_LOW_CHN_1QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_CHN_1QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_CHN_1QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_CHN_2HM und IOPT_LOW_CHN_3HM ist, so begrenzen IOPT_LOW_CHN_2QH oder IOPT_LOW_CHN_2QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_CHN_2QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_CHN_2QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM größer IOPT_LOW_CHN_3HM ist, so begrenzen IOPT_LOW_CHN_3QH oder IOPT_LOW_CHN_3QM den Anteil an Tasks der Priorität LOW:

- IOPT_LOW_CHN_3QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
- IOPT_LOW_CHN_3QM, falls der Anteil an Tasks mit Priorität HIGH klein ist

Grenzwerte für I/O-Priorität LOW und Ports festlegen

IOPT_LOW_POR legt Grenzwerte für die Port-Auslastung fest. Sobald die Port-Auslastung mit Tasks der Priorität HIGH und MEDIUM die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität LOW beschränkt.

IOPT_LOW_POR
IOPT_LOW_POR_1HM=20 / <integer 0...100>
IOPT_LOW_POR_2HM=30 / <integer 0...100>
IOPT_LOW_POR_3HM=40 / <integer 0...100>
IOPT_LOW_POR_1QH=40 / <integer 0...100>
IOPT_LOW_POR_2QH=30 / <integer 0...100>
IOPT_LOW_POR_3QH=20 / <integer 0...100>
IOPT_LOW_POR_1QM=55 / <integer 0...100>
IOPT_LOW_POR_2QM=40 / <integer 0...100>
IOPT_LOW_POR_3QM=30 / <integer 0...100>

IOPT_LOW_POR_1HM, IOPT_LOW_POR_2HM, IOPT_LOW_POR_3HM

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH und MEDIUM.

IOPT_LOW_POR_1QH, IOPT_LOW_POR_2QH, IOPT_LOW_POR_3QH

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei hohem Anteil an Tasks mit Priorität HIGH.

IOPT_LOW_POR_1QM, IOPT_LOW_POR_2QM, IOPT_LOW_POR_3QM

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei niedrigem Anteil an Tasks mit Priorität HIGH.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM kleiner IOPT_LOW_POR_1HM ist, so werden Tasks der Priorität LOW nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_POR_1HM und IOPT_LOW_POR_2HM ist, so begrenzen IOPT_LOW_POR_1QH oder IOPT_LOW_POR_1QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_POR_1QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_POR_1QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_POR_2HM und IOPT_LOW_POR_3HM ist, so begrenzen IOPT_LOW_POR_2QH oder IOPT_LOW_POR_2QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_POR_2QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_POR_2QM, falls der Anteil an Tasks mit Priorität HIGH klein ist

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM größer IOPT_LOW_POR_3HM ist, so begrenzen IOPT_LOW_POR_3QH oder IOPT_LOW_POR_3QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_POR_3QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_POR_3QM, falls der Anteil an Tasks mit Priorität HIGH klein ist

Grenzwerte für I/O-Priorität LOW und Pfade festlegen

IOPT_LOW_PTH legt Grenzwerte für die Pfadauslastung fest. Sobald die Pfadauslastung mit Tasks der Priorität HIGH und MEDIUM die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität LOW beschränkt.

IOPT_LOW_PTH
IOPT_LOW_PTH_1HM=20 / <integer 0...100>
IOPT_LOW_PTH_2HM=30 / <integer 0...100>
IOPT_LOW_PTH_3HM=40 / <integer 0...100>
IOPT_LOW_PTH_1QH=40 / <integer 0...100>
IOPT_LOW_PTH_2QH=30 / <integer 0...100>
IOPT_LOW_PTH_3QH=20 / <integer 0...100>
IOPT_LOW_PTH_1QM=55 / <integer 0...100>
IOPT_LOW_PTH_2QM=40 / <integer 0...100>
IOPT_LOW_PTH_3QM=30 / <integer 0...100>

IOPT_LOW_PTH_1HM, IOPT_LOW_PTH_2HM, IOPT_LOW_PTH_3HM

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH und MEDIUM.

IOPT_LOW_PTH_1QH, IOPT_LOW_PTH_2QH, IOPT_LOW_PTH_3QH

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei hohem Anteil an Tasks mit Priorität HIGH.

IOPT_LOW_PTH_1QM, IOPT_LOW_PTH_2QM, IOPT_LOW_PTH_3QM

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei niedrigem Anteil an Tasks mit Priorität HIGH.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM kleiner IOPT_LOW_PTH_1HM ist, so werden Tasks der Priorität LOW nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_PTH_1HM und IOPT_LOW_PTH_2HM ist, so begrenzen IOPT_LOW_PTH_1QH oder IOPT_LOW_PTH_1QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_PTH_1QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_PTH_1QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
  - Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_PTH_2HM und IOPT_LOW_PTH_3HM ist, so begrenzen IOPT_LOW_PTH_2QH oder IOPT_LOW_PTH_2QM den Anteil an Tasks der Priorität LOW:
    - IOPT_LOW_PTH_2QH, falls der Anteil an Tasks mit Priorität HIGH groß ist

        - IOPT_LOW_PTH_2QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
    - Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM größer IOPT_LOW_PTH_3HM ist, so begrenzen IOPT_LOW_PTH_3QH oder IOPT_LOW_PTH_3QM den Anteil an Tasks der Priorität LOW:
        - IOPT_LOW_PTH_3QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
        - IOPT_LOW_PTH_3QM, falls der Anteil an Tasks mit Priorität HIGH klein ist

Grenzwerte für I/O-Priorität LOW und Geräte festlegen

IOPT_LOW_DEV legt Grenzwerte für die Geräteauslastung fest. Sobald die Geräteauslastung mit Tasks der Priorität HIGH und MEDIUM die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität LOW beschränkt.

IOPT_LOW_DEV
IOPT_LOW_DEV_1HM=15 / <integer 0...100>
IOPT_LOW_DEV_2HM=22 / <integer 0...100>
IOPT_LOW_DEV_3HM=30 / <integer 0...100>
IOPT_LOW_DEV_1QH=35 / <integer 0...100>
IOPT_LOW_DEV_2QH=25 / <integer 0...100>
IOPT_LOW_DEV_3QH=10 / <integer 0...100>
IOPT_LOW_DEV_1QM=50 / <integer 0...100>
IOPT_LOW_DEV_2QM=40 / <integer 0...100>
IOPT_LOW_DEV_3QM=30 / <integer 0...100>
IOPT_LOW_DEV_0SU=70 / <integer 0...100>
IOPT_LOW_DEV_0HM=1 / <integer 0...100>
IOPT_LOW_DEV_0RQ=25 / <integer 0...100>

IOPT_LOW_DEV_1HM, IOPT_LOW_DEV_2HM, IOPT_LOW_DEV_3HM

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH und MEDIUM.

IOPT_LOW_DEV_1QH, IOPT_LOW_DEV_2QH, IOPT_LOW_DEV_3QH

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei hohem Anteil an Tasks mit Priorität HIGH.

IOPT_LOW_DEV_1QM, IOPT_LOW_DEV_2QM, IOPT_LOW_DEV_3QM

Grenzwerte 1, 2 und 3 für Tasks der Priorität LOW bei niedrigem Anteil an Tasks mit Priorität HIGH.

IOPT_LOW_DEV_0SU, IOPT_LOW_DEV_0HM, IOPT_LOW_DEV_0RQ

Grenzwerte für eine Beschränkung der Tasks der Priorität LOW bei einer hohen Gesamtauslastung des Geräts bei gleichzeitig relativ niedriger Auslastung durch Tasks der Priorität HIGH und MEDIUM (d.h. IOPT_LOW_DEV_1HM wird nicht erreicht).

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM kleiner IOPT_LOW_DEV_1HM ist, so werden Tasks der Priorität LOW nicht beschränkt.

- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_DEV_1HM und IOPT_LOW_DEV_2HM ist, so begrenzen IOPT_LOW_DEV_1QH oder IOPT_LOW_DEV_1QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_DEV_1QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_DEV_1QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM zwischen IOPT_LOW_DEV_2HM und IOPT_LOW_DEV_3HM ist, so begrenzen IOPT_LOW_DEV_2QH oder IOPT_LOW_DEV_2QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_DEV_2QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_DEV_2QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Wenn die Auslastung mit Tasks der Priorität HIGH und MEDIUM größer IOPT_LOW_DEV_3HM ist, so begrenzen IOPT_LOW_DEV_3QH oder IOPT_LOW_DEV_3QM den Anteil an Tasks der Priorität LOW:
  - IOPT_LOW_DEV_3QH, falls der Anteil an Tasks mit Priorität HIGH groß ist
  - IOPT_LOW_DEV_3QM, falls der Anteil an Tasks mit Priorität HIGH klein ist
- Die Auslastung durch Tasks der Priorität LOW wird durch IOPT_LOW_DEV_1QH bzw. IOPT_LOW_DEV_1QM auch dann beschränkt, wenn folgende Bedingungen erfüllt sind:
  - die Gesamtauslastung eines Gerätes überschreitet den Wert IOPT_LOW_DEV_0SU
  - die Summe der Auslastungen durch Tasks der Priorität HIGH und MEDIUM überschreitet den Wert IOPT_LOW_DEV_0HM
  - die zehnfache durchschnittliche Anzahl von I/O-Aufträgen für das Gerät liegt über dem Wert IOPT_LOW_DEV_0RQ

  Wenn die Auslastung durch Tasks der Priorität HIGH die Werte IOPT_LOW_DEV_2HM oder IOPT_LOW_DEV_3HM überschreitet, so gelten die für diese Grenzwerte definierten Beschränkungen. Wenn nach Einleitung einer Beschränkung trotz deutlich geringerer Gesamtauslastung die Auslastung durch Tasks der Priorität HIGH und MEDIUM nicht wesentlich zunimmt, wird die Beschränkung wieder aufgehoben.

4.1.1.6 Grenzwerte für I/O-Priorität MEDIUM festlegen

Grenzwerte für I/O-Priorität MEDIUM und Kanäle festlegen

IOPT_MED_CHN legt Grenzwerte für die Kanalauslastung fest. Sobald die Kanalauslastung mit Tasks der Priorität HIGH die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität MEDIUM beschränkt.

IOPT_MED_CHN
IOPT_MED_CHN_1H=20 / <integer 0...100>
IOPT_MED_CHN_2H=30 / <integer 0...100>
IOPT_MED_CHN_3H=40 / <integer 0...100>
IOPT_MED_CHN_1Q=50 / <integer 0...100>
IOPT_MED_CHN_2Q=40 / <integer 0...100>
IOPT_MED_CHN_3Q=30 / <integer 0...100>

IOPT_MED_CHN_1H, IOPT_MED_CHN_2H, IOPT_MED_CHN_3H

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH.

IOPT_MED_CHN_1Q, IOPT_MED_CHN_2Q, IOPT_MED_CHN_3Q

Grenzwerte 1, 2 und 3 für Tasks der Priorität MEDIUM.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH kleiner IOPT_MED_CHN_1H ist, so werden Tasks der Priorität MEDIUM nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_CHN_1H und IOPT_MED_CHN_2H ist, so begrenzt IOPT_MED_CHN_1Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_CHN_2H und IOPT_MED_CHN_3H ist, so begrenzt IOPT_MED_CHN_2Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH größer IOPT_MED_CHN_3H ist, so begrenzt IOPT_MED_CHN_3Q den Anteil an Tasks der Priorität MEDIUM.

Grenzwerte für I/O-Priorität MEDIUM und Ports festlegen

IOPT_MED_POR legt Grenzwerte für die Port-Auslastung fest. Sobald die Port-Auslastung mit Tasks der Priorität HIGH die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität MEDIUM durch Grenzen beschränkt.

IOPT_MED_POR
IOPT_MED_POR_1H=20 / <integer 0...100>
IOPT_MED_POR_2H=30 / <integer 0...100>
IOPT_MED_POR_3H=40 / <integer 0...100>
IOPT_MED_POR_1Q=50 / <integer 0...100>
IOPT_MED_POR_2Q=40 / <integer 0...100>

IOPT_MED_POR_3Q=30 / <integer 0...100>

IOPT_MED_POR_1H, IOPT_MED_POR_2H, IOPT_MED_POR_3H

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH.

IOPT_MED_POR_1Q, IOPT_MED_POR_2Q, IOPT_MED_POR_3Q

Grenzwerte 1, 2 und 3 für Tasks der Priorität MEDIUM.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH kleiner IOPT_MED_POR_1H ist, so werden Tasks der Priorität MEDIUM nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_POR_1H und IOPT_MED_POR_2H ist, so begrenzt IOPT_MED_POR_1Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_POR_2H und IOPT_MED_POR_3H ist, so begrenzt IOPT_MED_POR_2Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH größer IOPT_MED_POR_3H ist, so begrenzt IOPT_MED_POR_3Q den Anteil an Tasks der Priorität MEDIUM.

Grenzwerte für I/O-Priorität MEDIUM und Pfade festlegen

IOPT_MED_PTH legt Grenzwerte für die Pfadauslastung fest. Sobald die Pfadauslastung mit Tasks der Priorität HIGH die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität MEDIUM beschränkt.

IOPT_MED_PTH
IOPT_MED_PTH_1H=20 / <integer 0...100>
IOPT_MED_PTH_2H=30 / <integer 0...100>
IOPT_MED_PTH_3H=40 / <integer 0...100>
IOPT_MED_PTH_1Q=50 / <integer 0...100>
IOPT_MED_PTH_2Q=40 / <integer 0...100>
IOPT_MED_PTH_3Q=30 / <integer 0...100>

IOPT_MED_PTH_1H, IOPT_MED_PTH_2H, IOPT_MED_PTH_3H

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH.

IOPT_MED_PTH_1Q, IOPT_MED_PTH_2Q, IOPT_MED_PTH_3Q

Grenzwerte 1, 2 und 3 für Tasks der Priorität MEDIUM.

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH kleiner IOPT_MED_PTH_1H ist, so werden Tasks der Priorität MEDIUM nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_PTH_1H und IOPT_MED_PTH_2H ist, so begrenzt IOPT_MED_PTH_1Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_PTH_2H und IOPT_MED_PTH_3H ist, so begrenzt IOPT_MED_PTH_2Q den Anteil an Tasks der Priorität MEDIUM.

- Wenn die Auslastung mit Tasks der Priorität HIGH größer IOPT_MED_PTH_3H ist, so begrenzt IOPT_MED_PTH_3Q den Anteil an Tasks der Priorität MEDIUM.

Grenzwerte für I/O-Priorität MEDIUM und Geräte festlegen

IOPT_MED_DEV legt Grenzwerte für die Geräteauslastung fest. Sobald die Geräteauslastung mit Tasks der Priorität HIGH die Grenzwerte erreicht, wird der Anteil der Tasks mit Priorität MEDIUM beschränkt.

IOPT_MED_DEV
IOPT_MED_DEV_1H=15 / <integer 0...100>
IOPT_MED_DEV_2H=22 / <integer 0...100>
IOPT_MED_DEV_3H=30 / <integer 0...100>
IOPT_MED_DEV_1Q=50 / <integer 0...100>
IOPT_MED_DEV_2Q=35 / <integer 0...100>
IOPT_MED_DEV_3Q=25 / <integer 0...100>
IOPT_MED_DEV_0S=70 / <integer 0...100>
IOPT_MED_DEV_0H=1 / <integer 0...100>
IOPT_MED_DEV_0R=25 / <integer 0...100>

IOPT_MED_DEV_1H, IOPT_MED_DEV_2H, IOPT_MED_DEV_3H

Grenzwerte 1, 2 und 3 für die Auslastung mit Tasks der Priorität HIGH.

IOPT_MED_DEV_1Q, IOPT_MED_DEV_2Q, IOPT_MED_DEV_3Q

Grenzwerte 1, 2 und 3 für Tasks der Priorität MEDIUM.

IOPT_MED_DEV_0S, IOPT_MED_DEV_0H, IOPT_MED_DEV_0R

Grenzwerte für eine Beschränkung der Tasks der Priorität MEDIUM bei einer hohen Gesamtauslastung des Geräts bei gleichzeitig relativ niedriger Auslastung durch Tasks der Priorität HIGH (d.h. IOPT_MED_DEV_1H wird nicht erreicht).

Die Grenzwerte wirken folgendermaßen:

- Wenn die Auslastung mit Tasks der Priorität HIGH kleiner IOPT_MED_DEV_1H ist, so werden Tasks der Priorität MEDIUM nicht beschränkt.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_DEV_1H und IOPT_MED_DEV_2H ist, so begrenzt IOPT_MED_DEV_1Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH zwischen IOPT_MED_DEV_2H und IOPT_MED_DEV_3H ist, so begrenzt IOPT_MED_DEV_2Q den Anteil an Tasks der Priorität MEDIUM.
- Wenn die Auslastung mit Tasks der Priorität HIGH größer IOPT_MED_DEV_3H ist, so begrenzt IOPT_MED_DEV_3Q den Anteil an Tasks der Priorität MEDIUM.
- Die Auslastung durch Tasks der Priorität MEDIUM wird durch IOPT_MED_DEV_1Q auch dann beschränkt, wenn folgende Bedingungen erfüllt sind:
  - die Gesamtauslastung eines Gerätes überschreitet den Wert IOPT_MED_DEV_0S

- die Summe der Auslastungen durch Tasks der Priorität HIGH und MEDIUM überschreitet den Wert IOPT_MED_DEV_0H
- die zehnfache durchschnittliche Anzahl von I/O-Aufträgen für das Gerät liegt über dem Wert IOPT_MED_DEV_0R

Wenn die Auslastung durch Tasks der Priorität HIGH die Werte IOPT_MED_DEV_2H oder IOPT_MED_DEV_3H überschreitet, so gelten die für diese Grenzwerte definierten Beschränkungen.
Wenn nach Einleitung einer Beschränkung trotz deutlich geringerer Gesamtauslastung die Auslastung durch Tasks der Priorität HIGH nicht wesentlich zunimmt, wird die Beschränkung wieder aufgehoben.

4.1.1.7 Grenzwerte für I/O-Prioritäten abfragen

Grenzwerte für die I/O-Priorität LOW abfragen

IOPT_LOW? liefert die aktuell gültigen Grenzwerte.

IOPT_LOW?
IOPT_LOW? IOPT_LOW_xxx_iyy?

ohne Zusatz

Alle Grenzwerte und Grenzen für die Auslastung der Kanäle, Ports, Pfade und Geräte werden ausgegeben.

xxx Auswahl, ob die Grenzwerte und Grenzen für Kanäle, Ports, Pfade oder Geräte ausgegeben werden. Folgende Angaben sind möglich:

CHN	Kanäle
POR	Ports
PTH	Pfade
DEV	Geräte

iyy Auswahl, ob die ersten, zweiten oder dritten Grenzwerte oder Grenzen ausgegeben werden. Folgende Angaben sind möglich:

i 1, 2 oder 3

yy HM (Grenzwert für die Tasks mit I/O-Priorität HIGH und MEDIUM)

QH (Grenze für die Tasks der I/O-Priorität LOW bei hohem Anteil an Tasks mit Priorität HIGH)

QM (Grenze für die Tasks der I/O-Priorität LOW bei niedrigem Anteil an Tasks mit Priorität HIGH)

Grenzwerte für die I/O-Priorität MEDIUM abfragen

IOPT_MED? liefert die aktuell gültigen Grenzwerte.

IOPT_MED?
IOPT_MED? IOPT_MED_xxx_iy?

ohne Zusatz

Alle Grenzwerte und Grenzen für die Auslastung der Kanäle, Ports, Pfade und Geräte werden ausgegeben.

xxx Auswahl, ob die Grenzwerte und Grenzen für Kanäle, Ports, Pfade oder Geräte ausgegeben werden. Folgende Angaben sind möglich:

CHN	Kanäle
POR	Ports
PTH	Pfade
DEV	Geräte

iy Auswahl, ob die ersten, zweiten oder dritten Grenzwerte oder Grenzen ausgegeben werden. Folgende Angaben sind möglich:

i 1, 2 oder 3

y H (Grenzwert für die Tasks mit I/O-Priorität HIGH und MEDIUM)

Q (Grenzwert für die Tasks mit I/O-Priorität HIGH)

4.1.1.8 Auslastung abfragen

Mit den Anweisungen IOPT_INF_... werden prozentuale Auslastungen von Geräten, Kanälen, Ports und Pfaden durch Tasks der I/O-Prioritäten HIGH, MEDIUM und LOW abgefragt. Außerdem werden die expliziten und ggf. impliziten Sperrfaktoren für Tasks der I/O-Priorität MEDIUM und LOW ausgegeben.

Der Sperrfaktor ist das Verhältnis von Dauer der Sperre durch IORM zu I/O-Dauer.

Zu einer expliziten Sperre für Tasks der I/O-Priorität MEDIUM oder LOW für einen Kanal (bzw. einen Port, einen Pfad, ein Gerät) kommt es dann, wenn der erste Grenzwert für MEDIUM oder LOW für die Kanalauslastung (bzw. die Portauslastung, die Pfadauslastung, die Geräteauslastung) überschritten wird.

Zu einer impliziten Sperre für Tasks der I/O-Priorität MEDIUM oder LOW für ein Gerät kommt es dann, wenn ein Kanal, ein Port oder ein Pfad, an dem das Gerät angeschlossen ist, explizit gesperrt ist.

Geräte

IOPT_INF_DEV?
IOPT_INF_DEV?mask

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens. Folgende Angaben sind möglich:

- `*` Die Auslastung aller Geräte wird aufgelistet.
- `n*` Die Auslastung aller Geräte, deren Namen mit n beginnen, wird aufgelistet.
- `nm*` Die Auslastung aller Geräte, deren Namen mit nm beginnen, wird aufgelistet.
- `nml*` Die Auslastung aller Geräte, deren Namen mit nml beginnen, wird aufgelistet.
- `nmlo` Die Auslastung des Geräts mit dem Namen nmlo wird aufgelistet.

Kanäle

IOPT_INF_CHN?
IOPT_INF_CHN?mask

mask Angabe eines Kanals. Folgende Angaben sind möglich:

- `*` Die Auslastung aller Kanäle wird aufgelistet.
- `nm` Die Auslastung des Kanals nm wird aufgelistet.

Pfade

IOPT_INF_PTH?
IOPT_INF_PTH?mask

mask Angabe eines Pfads. Folgende Angaben sind möglich:

* Die Auslastung aller Pfade wird aufgelistet.

nmlo/pq Die Auslastung des Pfads vom Kanal pq zur Steuerung nmlo wird aufgelistet.

Ports

IOPT_INF_POR?
IOPT_INF_POR?mask

mask Angabe eines Ports. Folgende Angaben sind möglich:

* Die Auslastung aller Ports wird aufgelistet.

portname Die Auslastung des Ports mit dem angegebenen Namen wird aufgelistet.

Als Portname kann der WWPN (World Wide Port Name - 16 Zeichen) des Steuerungsports angegeben werden.

Beispiel

Für die Geräte 3801 und 3803 an der Steuerung 3800 und am Kanal 50 sind I/Os aktiv. Der FC-Port an der Steuerung besitzt die WWPN 5006048448586C01.

Abfrage der Auslastung am Kanal 50

```
IOPT_INF_CHN?50
```

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+---------------
50                |  24  19  13 |    0      0.01 |    0       0
```

Abfrage für Pfad 3800/50

```
IOPT_INF_PTH?3800/50
```

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+--------------
3800/50           |  24  19  13 |    0      0.01 |    0       0
```

Abfrage für Port 5006048448586C01

```
IOPT_INF_POR?5006048448586C01
```

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+--------------
5006048448586C01  |  24  19  13 |    0      0.01 |    0       0
```

Abfrage für Geräte 3801 und 3803:

```
IOPT_INF_DEV?380*
```

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+--------------
 3800             |  0   0   0  |    0       0   |    0       0
 3801             |  70  0   11 |    0      5.81 |    0       0
 3802             |  0   0   0  |    0       0   |    0       0
 3803             |  0   60  29 |    0      1.62 |    0       0
```

4.1.1.9 Prüfbetrieb

Bevor IOPT mit IOPT_SET_ON=YES aktiviert wird, kann mit IOPT-Mitteln geprüft werden, ob ein IOPT-Einsatz sinnvoll ist.

Der Prüfbetrieb setzt voraus:

- IOPT wird mit der Option CHK aktiviert (IOPT_SET_ON=CHK, siehe "IOPT aktivieren und deaktivieren").
- I/O-Prioritätsklassen für Tasks sind festgelegt (`/MODIFY-TASK-CATEGORIES` oder IOPT-Anweisung IOPT_PRI (siehe "Grenzwerte für I/O-Prioritätsklassen festlegen und abfragen")).
- Die Grenzwerte für die Auslastung durch Tasks der verschiedenen Prioritätsklassen sind festgelegt (siehe IOPT-Anweisungen IOPT_LOW_... und IOPT_MED_...).

Die Anweisung IOPT_CHK_RESET legt den Zeitpunkt für den Beginn der Prüfung fest. Danach (beispielsweise nach einer Stunde oder nach einem Tag) kann mit Anweisungen IOPT_CHK_...? abgefragt werden, wie oft IOPT niederpriore Task gesperrt hätte.

Der berücksichtigte Prüfzeitraum beginnt stets mit der Anweisung IOPT_CHK_RESET und endet mit dem Zeitpunkt der Abfrage.

Prüfung starten

IOPT_CHK_RESET
IOPT_CHK_RESET

Abfrage der Sperren für Kanäle

IOPT_CHK_CHN?
IOPT_CHK_CHN?mask

mask Angabe eines Kanals. Folgende Angaben sind möglich:

* Die Sperren aller Kanäle werden aufgelistet.

nm Die Sperren des Kanals nm werden aufgelistet.

Abfrage der Sperren für Pfade

IOPT_CHK_PTH?
IOPT_CHK_PTH?mask

mask Angabe eines Pfads. Folgende Angaben sind möglich:

* Die Sperren aller Pfade werden aufgelistet.

nmlo/pq Die Sperre des Pfads vom Kanal pq zur Steuerung nmlo wird aufgelistet.

Abfrage der Sperren für Ports

IOPT_CHK_POR?
IOPT_CHK_POR?mask

mask Angabe eines Ports. Folgende Angaben sind möglich:

- * Die Sperren aller Ports werden aufgelistet.
- portname Die Sperren des Ports mit dem angegebenen Namen wird aufgelistet.

Abfrage der Sperren für Geräte

IOPT_CHK_DEV?
IOPT_CHK_DEV?mask

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens. Folgende Angaben sind möglich:

- * Die Sperren aller Geräte werden aufgelistet.
- n* Die Sperren aller Geräte, deren Namen mit n beginnen, werden aufgelistet.
- nm* Die Sperren aller Geräte, deren Namen mit nm beginnen, werden aufgelistet.
- nml* Die Sperren aller Geräte, deren Namen mit nml beginnen, werden aufgelistet
- nmlo Die Sperre des Geräts mit dem Namen nmlo wird aufgelistet.

Beispiel

Für die Geräte 3801 und 3803 an der Steuerung 3800 und am Kanal 50 sind I/Os aktiv. Der FC-Port an der Steuerung besitzt die WWPN 5006048448586C01.

Abfrage für Kanäle

```
IOPT_CHK_CHN?*
```

Ausgabe:

```
    UNIT NAME    | FREQUENCY %
-----------------+-------------
50               |      73
```

Für Kanal 50 waren zu 73 % die Sperrbedingungen erfüllt.

Zusätzliche quantitative Analyse für Kanal 50:

`IOPT_INF_CHN?50`

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+---------------
50                |  22  22  44 |    0       0   |    0       0
```

Abfrage für Pfade

`IOPT_CHK_PTH?*`

Ausgabe:

```
    UNIT NAME    | FREQUENCY %
-----------------+-------------
3800/50          |      73
```

Für Pfad 3800/50 (Steuerung/Kanal) waren zu 73 % die Sperrbedingungen erfüllt.

Zusätzliche quantitative Analyse für Pfad 3800/50:

`IOPT_INF_PTH?3800/50`

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+--------------
3800/50           |  22  22  44 |    0       0   |    0       0
```

Abfrage für Ports

`IOPT_CHK_POR?*`

Ausgabe:

```
    UNIT NAME    | FREQUENCY %
-----------------+-------------
5006048448586C01 |      73
```

Für Port 5006048448586C01 waren zu 73 % die Sperrbedingungen erfüllt.

Zusätzliche quantitative Analyse für Port 5006048448586C01:

`IOPT_INF_POR?5006048448586C01`

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+---------------
5006048448586C01  |  22  22  44 |    0       0   |    0       0
```

Abfrage für Geräte

`IOPT_CHK_DEV?*`

Ausgabe:

```
    UNIT NAME    | FREQUENCY %
-----------------+-------------
3801             |      73
3803             |      73
```

Für die Geräte 3801 und 3803 waren die Sperrbedingungen zu 73 % erfüllt.

Zusätzliche quantitative Analyse für die Geräte 3801 und 3803:

`IOPT_INF_DEV?380*`

Ausgabe:

```
    UNIT NAME     | HIG MED LOW | EXPLICIT DELAY | IMPLICIT DELAY
                  |  %   %   %  |   MED     LOW  |   MED     LOW
------------------+-------------+----------------+---------------
 3800             |  0   0   0  |    0       0   |    0       0
 3801             |  48  0   48 |    0       0   |    0       0
 3802             |  0   0   0  |    0       0   |    0       0
 3803             |  0   48  48 |    0       0   |    0       0
```

Bei dieser Auslastung ist es sinnvoll, die I/Os für die Geräte 3801 und 3803 durch IOPT steuern zu lassen.

4.1.2 Einsatzbeispiele

Folgende Beispiele zeigen die IOPT-Anweisungen für zwei verschiedene Szenarien.

Beispiel 1

IOPT überwacht alle Geräte. Für die I/O-Prioritätsklassen gelten die Einstellungen von `/MODIFY-TASK-CATEGORIES` oder die Standardwerte der IOPT-Anweisung IOPT_PRI (siehe "Grenzwerte für I/O-Prioritätsklassen festlegen und abfragen"):

```
/START-IORM
IOPT_SET_ON=YES
IOPT_DEV_ADD=ALL
END
```

Die IOPT-Anweisungen können dabei im Dialog eingegeben oder in der Datei `SYSDAT.IORM.nnn` hinterlegt werden.

Beispiel 2

Die logischen Geräte 8800, 8801, 8810, 8820, 8821 und 8822 liegen auf einem physikalischen Gerät. Es sind Anwendungen mit IO-PRIO=HIGH aktiv auf 8800 und 8801, sowie Anwendungen mit IO-PRIO=LOW auf 8810, 8820, 8821 und 8822. Die Anwendungen mit IO-PRIO=LOW stören die Anwendungen mit IO-PRIO=HIGH. An den logischen Geräten erkennt IORM keinen Konflikt, da an den Ports, Pfaden und Kanälen die Auslastung noch unterhalb der vorgegebenen Grenzwerte liegt. Daher wird eine Gerätegruppe 001 definiert.

Für die I/O-Prioritätsklassen gelten die Einstellungen von `/MODIFY-TASK-CATEGORIES` oder die Standardwerte der IOPT-Anweisung IOPT_PRI (siehe "Grenzwerte für I/O-Prioritätsklassen festlegen und abfragen"):

```
/START-IORM
IOPT_GRP_001_ADD=D-R(8800,8801)
IOPT_GRP_001_ADD=D-R(8810,8810)
IOPT_GRP_001_ADD=D-R(8820,8822)
END
```

Die IOPT-Anweisungen können dabei im Dialog eingegeben oder in der Datei `SYSDAT.IORM.nnn` hinterlegt werden.

4.2 DPAV: Monitoring für Alias-Geräte und Begrenzung der Alias-Anzahl für FastDPAV

Alternativ zum einfachen Plattenzugriff (Standard) gibt es einen parallelen Plattenzugriff an allen BS2000-Servern über Parallel Access Volumes („PAV“). PAV gibt es historisch bedingt in den beiden (veralteten) statischen Verfahren 'Standard-PAV' und 'Extended-PAV' (XPAV) und in dem dynamischen Verfahren 'Fast Dynamic Parallel Access Volume' (FastDPAV), siehe Handbuch „Systembetreuung“ [5]. Mit PAV lassen sich die Antwortzeiten bei stark ausgelasteten Platten reduzieren.

Statisches PAV verlangt eine vorausschauende Planung auf die zukünftige Geräteauslastung, d.h. den stark ausgelasteten Platten muss im Voraus die richtige Anzahl von Alias-Geräten zugewiesen werden.

Das wird sehr viel einfacher bei Verwendung von FastDPAV, weil für eine Menge von Platten ein Pool von FastDPAV Alias-Geräten ohne feste Zuordnung zu einer Platte generiert wird. Erst beim Start der I/O wird durch das BS2000-I/O-System (BS2IOS) entschieden, welches Gerät zugewiesen wird. Geplant werden muss noch die Anzahl der Aliasgeräte in dem Pool.

Abfrage der I/O-Rate

Mit der Anweisung DPAV_CHK_ALI kann für Alias-Geräte sowohl des statischen PAV als auch des FastDPAV die I/O-Rate abgefragt werden. Aus dem Ergebnis lässt sich bei Verwendung von statischem PAV erschließen, für welche Basis-Geräte mehr oder weniger Alias-Geräte sinnvoll wären. Bei Verwendung von FastDPAV lässt sich erschließen, ob der Alias-Pool optimal dimensioniert ist.

Begrenzung unerwünschter I/O-Last bei FastDPAV

Für FastDPAV bietet DPAV mit den Anweisungen DPAV_BAS_ADD und DPAV_BAS_REM noch eine weitere Option, die Lastverteilung zu optimieren. Nicht alle Platten und ggfs. Gastsysteme (bei Verwendung von VM2000) sind (gleich) wichtig. Unerwünschte I/O-Last kann dadurch begrenzt werden, dass für einzelne Platten die erlaubte Alias-Anzahl (ggfs. bis auf 0) reduziert wird oder einzelnen Gastsystemen keine oder nur wenige Alias-Geräte zugewiesen werden.

4.2.1 Anweisungen

- DPAV_CHK_ALI
- DPAV_BAS_ADD/DPAV_BAS_REM

4.2.1.1 DPAV_CHK_ALI

Abfrage vorhandener Alias-Geräte und ihrer I/O-Aktivität

DPAV_CHK_ALI? listet vorhandene Alias-Geräte auf und gibt deren I/O-Aktivität (durchschnittliche I/Os pro Sekunde) aus. Die I/O-Aktivität bezieht sich nur auf das eigene System. Im VM-Modus könnten zusätzlich I/Os an anderen Gastsystemen aktiv sein.

DPAV_CHK_ALI?
DPAV_CHK_ALI?mask

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens. Folgende Angaben sind möglich:

n*	Alle Alias-Geräte werden aufgelistet.
n*	Alle Alias-Geräte, deren Namen mit n beginnen, werden aufgelistet.
nm*	Alle Alias-Geräte, deren Namen mit nm beginnen, werden aufgelistet.
nml*	Alle Alias-Geräte, deren Namen mit nml beginnen, werden aufgelistet.
nmlo	Das Alias-Gerät mit dem Namen nmlo wird aufgelistet.

Beispiel

```
UNIT NAME  | FREQUENCY %
-----------+------------
3803       |     100
```

Abfrage vorhandener Alias-Geräte und ihrer I/O-Aktivität

Prüfung, ob an Steuerung 3800 Alias-Geräte vorhanden sind:

```
DPAV_CHK_ALI?38*
```

Ausgabe:

```
UNIT NAME  | I/O PER SEC
-----------+------------
380A       |        0
380B       |        0
380C       |        0
380D       |        0
380E       |        0
380F       |        0
3804       |        0
3805       |        0
3806       |        0
3807       |     2569
3808       |        0
3809       |        0
```

Am Alias-Gerät 3807 sind 2569 I/Os pro Sekunde aktiv, die restlichen Alias-Geräte zeigen keine I/O-Aktivität am eigenen System.

```
UNIT NAME  | FREQUENCY %
-----------+------------
           |
```

4.2.1.2 DPAV_BAS_ADD/DPAV_BAS_REM

Mit den folgenden Anweisungen kann für FastDPAV die Anzahl der Aliasgeräte begrenzt werden. Im VM2000-Betrieb müssen die Anweisungen jeweils im lokalen BS2000-System eingegeben werden.

Per Default werden alle FastDPAV-Basis-Geräte für die Nutzung von FastDPAV-Alias-Geräten berücksichtigt. Mit der Anweisung DPAV_BAS_REM können FastDPAV-Basis-Geräte von der Nutzung ausgeschlossen werden. Mit DPAV_BAS_ADD kann die Nutzung (wieder) erlaubt werden. Dabei kann auch die maximale Alias-Anzahl pro Platte modifiziert werden. Der Defaultwert entspricht dem Maximalwert von 7.

Basis-Geräte für FastDPAV aktivieren

DPAV_BAS_ADD aktiviert Basis-Geräte für FastDPAV.

DPAV_BAS_ADD
DPAV_BAS_ADD=ALL[,number] DPAV_BAS_ADD=D-R(mn1,mn2)[,number]

ALL Alle Basis-Geräte werden für FastDPAV aktiviert.

D-R(mn1,mn2)

Alle Basis-Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden aktiviert. Wenn mn1 und mn2 gleich sind, wird nur ein Basis-Gerät aktiviert.

number

Maximale Anzahl Alias-Geräte für ein angegebenes Basis-Gerät. Standardwert: 7.

Aktivierte Basis-Geräte abfragen

DPAV_BAS_ADD? gibt eine Liste aller Basis-Geräte aus, die für FastDPAV aktiviert sind.

DPAV_BAS_ADD?
DPAV_BAS_ADD?mask

mask Vollständige oder maskierte Angabe eines mnemotechnischen Gerätenamens.
Folgende Angaben sind möglich:

* Alle aktivierten Basis-Geräte werden aufgelistet.

n* Alle aktivierten Basis-Geräte, deren Namen mit n beginnen, werden aufgelistet.

nm* Alle aktivierten Basis-Geräte, deren Namen mit nm beginnen, werden aufgelistet.

nml* Alle aktivierten Basis-Geräte, deren Namen mit nml beginnen, werden aufgelistet.

nmlo Das Basis-Gerät mit dem Namen nmlo wird aufgelistet.

Basis-Geräte für FastDPAV deaktivieren

DPAV_BAS_REM deaktiviert Basis-Geräte für FastDPAV.

DPAV_BAS_REM
DPAV_BAS_REM=ALL DPAV_BAS_REM=D-R(mn1,mn2)

ALL Alle Basis-Geräte werden für FastDPAV deaktiviert.

D-R(mn1,mn2)

Alle Basis-Geräte mit mnemotechnischen Namen von mn1 bis mn2 werden deaktiviert. Wenn mn1 und mn2 gleich sind, wird nur ein Basis-Gerät deaktiviert.

4.3 DDAL: Optimierte Lastverteilung im Betrieb von ETERNUS CS

BS2000 kennt zwei Verfahren zur Auswahl von Bandgeräten:

- Auswahl eines geeigneten Bandgeräts in der Reihenfolge der Generierung. Wenn gleichzeitig mehrere Bänder in Benutzung sind, kann es dabei zu einer ungünstigen Lastverteilung kommen. Innerhalb eines ETERNUS CS können z.B. mehrere Geräte an einem ICP (Integrated Channel Processor) in Benutzung sein, während gleichzeitig an anderen ICPs keine Geräte aktiv sind.
- Optimierte Geräteauswahl für ETERNUS CS .
  Die Geräteverwaltung zählt für jeden ICP die Anzahl der aktiven Geräte und berücksichtigt diesen Zähler bei der Geräteauswahl.
  Die optimierte Geräteauswahl wird eingestellt mit

  `/MODIFY-MOUNT-PARAMETER NEXT-TAPE-MOUNT=*BY-CONTROLLER.`

In beiden Fällen kennt die Geräteverwaltung jedoch nur die Gerätebelegungen innerhalb eines Native-Systems oder innerhalb eines BS2000-Gastsystems unter VM2000.

Optimierte Geräteauswahl mit DDAL

Im Native-Betrieb realisiert die BS2000-Geräteverwaltung die optimierte Geräteauswahl für ETERNUS CS unabhängig von DDAL.

Im VM2000-Betrieb auf /390-Servern realisiert die Funktion DDAL (**D**ynamic **D**evice **A**llocation) die optimierte Geräteauswahl für ETERNUS CS über **alle** BS2000-Gastsysteme eines Servers hinweg.

IORM muss dazu im Monitorsystem und in allen BS2000-Gastsystemen gestartet und die Funktion DDAL mit DDAL_SET_ON=YES aktiviert sein.

Durch eine interne Kommunikation kennt IORM die Belegung der ICPs durch alle BS2000-Gastsysteme des Servers. Bei einer Gerätebelegung stellt IORM der lokalen Geräteverwaltung die globalen Belegungszähler zur Verfügung.

DDAL aktivieren und deaktivieren

DDAL_SET_ON aktiviert oder deaktiviert DDAL. Außerdem kann die aktuelle Einstellung abgefragt werden.

DDAL_SET_ON
DDAL_SET_ON=YES DDAL_SET_ON=NO DDAL_SET_ON?

YES DDAL wird aktiviert. Im VM-Betrieb ist diese Einstellung in allen Gastsystemen erforderlich, in denen DDAL aktiv sein soll.

NO DDAL wird im System deaktiviert. Im VM2000-Betrieb ist diese Einstellung in allen Gastsystemen erforderlich, in denen DDAL nicht aktiv sein soll.

? Die aktuelle Einstellung wird abgefragt.

4.4 IOLVM: Begrenzung der I/O-Leistungsaufnahme einzelner virtueller Maschinen

Im VM2000-Betrieb können weniger wichtige, jedoch I/O-intensive Gastsysteme andere, wichtigere Gastsysteme im I/O-Betrieb behindern. Behinderungen können entstehen, wenn die I/O-intensiven Gastsysteme I/Os auf das gleiche (logische) Gerät ausführen. Behinderungen entstehen auch, wenn I/Os auf verschiedene (logische) Geräte erfolgen, die jedoch auf demselben physikalischen Gerät liegen oder über dieselben Pfade angeschlossen oder über dieselben Ports erreichbar oder an denselben Kanälen angeschlossen sind.

IOLVM (**I**/**O** **L**imit for **V**irtual **M**achines) kann solche Konfliktsituationen erkennen und bremst gezielt den Ein-/Ausgabebetrieb des eigenen Gastsystems, wenn gemeinsam benutzte IO-Ressourcen (Kanal, Port, Pfad, Platte) das Gastsystem-spezifische IO-Limit überschreiten. Das IO-Limit wird als Prozentwert des durchschnittlichen Ein/Ausgabedurchsatzes der gemeinsam benutzten IO-Ressource angegeben.

In IOLVM werden nur Plattengeräte berücksichtigt.

Das I/O-Limit für IOLVM wird als maximale I/O-Leistungsaufnahme der VM im Operanden MAX-IO-UTILIZATION in den VM2000-Kommandos `/CREATE-VM` bzw. `/MODIFY-VM-ATTRIBUTES` und `/CREATE-VM-DEFINITION` bzw. `/MODIFY-VM-DEFINITION` festgelegt.

In den VM2000-Informationskommandos `/SHOW-VM-ATTRIBUTES`, `/SHOW-VM-RESOURCES` und `/SHOW-VM-DEFINITION` wird in der Spalte `MAX-IO` angezeigt, welcher Wert für die maximale I/O-Leistungsaufnahme der VM eingestellt ist.

5 JMP Rekonstruktion von ENTER-Kommandos aus dem Jobpool des JMS

Versionsstand: JMP V21.0C

Das Programm JMP (Jobpool Management Program) rekonstruiert aus den Informationen, die das JMS über akzeptierte Batch-Aufträge im Jobpool (Systemdatei SYSTEM.JOBPOOL) hinterlegt hat, ENTER-JOB-Kommandos. Die Kommandos werden von JMP in eine Datei geschrieben. Von dort können sie entnommen, nach Bedarf modifiziert und dem System wieder übergeben werden. Die Informationen des Jobpools können editiert und ausgedruckt werden. Gegenüber bisherigen Warmstartfunktionen bietet dies der Systemverwaltung zusätzliche Möglichkeiten, mit denen die Verarbeitung von Batch-Aufträgen (z.B. nach einem Versionswechsel) wieder aufgenommen werden kann.

Zu beachten ist, dass die rekonstruierten Kommandos ENTER-JOB in der Regel nicht ohne Prüfung und Modifikation durch den Auftraggeber oder die Systemverwaltung erneut gestartet werden können. Es ist nicht möglich, alle ursprünglich vom Auftraggeber eines Batch-Auftrages beabsichtigten Auftragsattribute zu rekonstruieren. Daher ist der Abschnitt „Hinweise zu den rekonstruierten Attributen“ sorgfältig zu lesen.

Das Programm kann im Batch- und Dialogmodus ablaufen. Es ist unter der privilegierten Benutzerkennung TSOS einzusetzen.

5.1 Ablauf von JMP

JMP öffnet die Jobpool-Datei (Organisationsform PAM) und richtet eine SAM-Datei (im Folgenden Prozedurdatei genannt) für die zu erzeugenden ENTER-Kommandos ein.
Den Dateinamen der Jobpool-Datei kann JMP entweder über einen Standard-Linknamen (Linkname SJOBPOOL) ermitteln oder der Dateiname wird in der JMP-Anweisung `//OPEN-JOBPOOL-FILE` explizit angegeben. Bei Übergabe des Namens der Jobpool-Datei anhand eines Linknamens ist vor dem Aufruf von JMP `/ADD-FILE-LINK LINK-NAME=SJOBPOOL, FILE-NAME=<dateiname>` aufzurufen.

Den Namen der zu erzeugenden Prozedurdatei ermittelt JMP analog entweder über den Standard-Linknamen PJOBPOOL oder er ist mittels JMP-Anweisung `//CREATE-PROCEDURE-FILE` explizit anzugeben. Die Prozedurdatei wird entweder neu erzeugt, oder eine bereits vorhandene wird überschrieben. Das Programm wird gestartet mit `/START-JMP.`

START-JMP	Alias: **JMP**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

JMP wird durch Steueranweisungen gesteuert, die von SYSDTA gelesen werden.

Hinweise zur Jobpool-Datei

Die während eines Systemlaufs aktuelle Jobpool-Datei namens SYSTEM.JOBPOOL befindet sich auf dem Home-Pubset. Sie ist während des Systemlaufs von der Jobpool-Task (Systemtask, TSN JOBP) eröffnet und daher vom Programm JMP nicht zugreifbar.

Wird der Home-Pubset (nach einem Shutdown) als Datenpubset importiert, ist die Datei nicht mehr gesperrt. Sie ist jedoch noch mit einem READ-Passwort geschützt und muss daher zunächst mit `/COPY-FILE ...,IGNORE-PROTECTION=*SOURCE` in eine ungeschützte Datei umkopiert werden, die dann mit JMP bearbeitet werden kann.

Der Jobpool enthält Informationen über alle gerade in Bearbeitung befindlichen und die noch zu startenden Jobs. Dies sind gerade laufende oder unterbrochene und alle noch nicht gestarteten Batch-Aufträge (bei Repeat- bzw. Kalenderjobs die noch zu startenden Ausprägungen). Unmittelbar vor einem Shutdown sollte die Jobpool-Datei daher nur Jobs mit einem Startzeitpunkt enthalten, der nach dem geplanten Session-Neubeginn liegt. Bei Neubeginn eines Systemlaufs (STARTUP) wird die Jobpool-Datei neu angelegt und unter Rückgriff auf die Informationen des Jobpools des vorhergehenden Systemlaufs neu aufgebaut. Alle im neuen Systemlauf nicht mehr zu berücksichtigenden Batch-Aufträge werden entfernt.

5.2 Anweisungen

Eine Anweisung kann sich über mehrere Zeilen erstrecken. Als Fortsetzungszeichen dient der Bindestrich. Er kündigt eine Folgezeile an. Ihm dürfen nur Leerzeichen bis zum Zeilenende folgen.

5.2.1 Übersicht über die Anweisungen von JMP

Anweisung	Bedeutung
CONVERT-JOBPOOL	Jobpool-Datei in neueres oder älteres Format überführen
CREATE-PROCEDURE-FILE	SAM-Datei erzeugen, die eine BS2000-Prozedur mit ENTER-Kommandos enthält
OPEN-JOBPOOL-FILE	Jobpool-Datei eröffnen
SHOW-JOBPOOL-STATUS	Informationen über den Inhalt einer Jobpool-Datei ausgeben
END	Programm JMP beenden

5.2.2 Beschreibung der Anweisungen

- CONVERT-JOBPOOL - Jobpool-Datei in neueres oder älteres Format überführen
- CREATE-PROCEDURE-FILE - SAM-Datei mit BS2000-Prozedur erzeugen
- END - Eingabe von Anweisungen beenden
- OPEN-JOBPOOL-FILE - Jobpool-Datei öffnen
- SHOW-JOBPOOL-STATUS - Informationen über den Jobpool abrufen

5.2.2.1 CONVERT-JOBPOOL - Jobpool-Datei in neueres oder älteres Format überführen

Mit dieser Anweisung kann aus der aktuellen, zuvor mit //OPEN-JOBPOOL-FILE geöffneten, Jobpool-Datei eine neue Jobpool-Datei für eine ältere oder neuere Betriebssystemversion erzeugt werden.

Die erzeugte Jobpool-Datei kann für einen Warmstart verwendet werden, sofern folgende Voraussetzungen der Systemumgebung erfüllt sind:
- die Benutzerkennungen und Jobklassen, welche die Jobs im Jobpool verwenden sowie
- die Pubsets, auf denen sich die Kommandodateien, Kalenderdateien und MONJVs befinden, müssen vorhanden sein.

Beim Wechsel von BS2000 kleiner gleich V21.0B nach BS2000 V21.0C hat sich die Struktur der jobbeschreibenden Daten (JTBP), die im JOBPOOL gespeichert sind, geändert.

Format

CONVERT-JOBPOOL
TO-FILE = *STD-FILE-LINK / <filename 1..54 without-gen-vers> ,OVERWRITE = *NO / *YES ,VERSION = *V3.0 / *V4.0

Operanden

TO-FILE = *STD-FILE-LINK / <filename 1..54 without-gen-vers>
Dateiname des neuen Jobpools.

TO-FILE = *STD-FILE-LINK
Der Dateiname soll aus der Task File Table (TFT) gelesen werden. Der vom Benutzer nicht veränderbare Linkname ist CJOBPOOL. Der Dateiname kann daher vom Benutzer vor dem Aufruf von JMP vereinbart werden durch:

```
/ADD-FILE-LINK LINK-NAME=CJOBPOOL,FILE-NAME=<dateiname>
```

Fehlt die Vereinbarung durch LINK-NAME, wird die Anweisung mit der Meldung `JMP0015` abgewiesen.

TO-FILE = <filename 1..54 without-gen-vers>
Vollqualifizierter Dateiname des neuen Jobpools.

OVERWRITE = *NO / *YES
Veranlasst oder verhindert das Überschreiben einer bereits vorhandenen Datei mit dem bei TO-FILE angegebenen Namen.

OVERWRITE = *NO
Das Überschreiben einer bereits vorhandenen Datei wird verhindert. Ist die Datei bereits vorhanden, wird die Anweisung mit der Meldung `JMP0012` abgewiesen.

OVERWRITE = *YES
Überschreibt eine bereits existierende Datei mit dem bei TO-FILE angegebenen Namen.

VERSION = *V3.0 / *V4.0
Gibt die Versionsnummer von BS2000 an, in welche die Jobpool-Datei konvertiert werden soll. Bei der Konvertierung des Jobpools wird die Struktur der jobbeschreibenden Daten modifiziert.

VERSION = *V3.0
Es wird eine Jobpool-Datei für Betriebssystemversionen größer gleich BS2000/OSD-BC V3.0 bis einschließlich BS2000 V21.0B erstellt.

VERSION = *V4.0
Es wird eine Jobpool-Datei für die Betriebssystemversion BS2000 V21.0C erstellt.

5.2.2.2 CREATE-PROCEDURE-FILE - SAM-Datei mit BS2000-Prozedur erzeugen

Mit dieser Anweisung kann eine SAM-Datei erzeugt werden, die eine BS2000-Prozedur mit den rekonstruierten ENTER-JOB-Kommandos enthält. Inhalt der Datei und Format der rekonstruierten Kommandos sind unten anhand eines Beispiels beschrieben.

Diese Anweisung ist erst zulässig, wenn vorher mindestens einmal `//OPEN-JOBPOOL-FILE` ausgeführt wurde.

Format

CREATE-PROCEDURE-FILE
FILE-NAME = *STD-FILE-LINK / <filename 1..54 without-gen-vers> ,OVERWRITE = *NO / *YES

Operanden

FILE-NAME = *STD-FILE-LINK / <filename 1..54 without-gen-vers>
Name der zu schreibenden Prozedurdatei. Für die Datei muss Schreibzugriff erlaubt sein.

FILE-NAME = *STD-FILE-LINK
Der Dateiname soll aus der Task File Table (TFT) gelesen werden. Der vom Benutzer nicht veränderbare Linkname ist PJOBPOOL.
Der Dateiname kann somit vom Benutzer vor Aufruf des JMP durch `/ADD-FILE-LINK LINK-NAME=PJOBPOOL, FILE-NAME=<dateiname>` vereinbart werden.

FILE-NAME = <filename 1..54 without-gen-vers>
Vollqualifizierter Dateiname. Die Angabe einer Dateigeneration oder Dateigenerationsgruppe und die Angabe des Dateinamens in der Form „datei(nr)“ (nr = Versionsnummer) ist nicht erlaubt.

OVERWRITE =
Veranlasst oder verhindert das Überschreiben einer bereits existierenden Datei mit dem bei FILE-NAME angegebenen Namen.

OVERWRITE = *NO
Das Überschreiben einer bereits existierenden Datei wird verhindert. Die ursprüngliche Datei bleibt unverändert. Die Prozedurdatei wird nicht erzeugt. Der Benutzer bekommt die Meldung `JMP0012`.

OVERWRITE = *YES
Besteht bereits eine Datei gleichen Namens, soll sie überschrieben und eine Prozedurdatei in jedem Fall erzeugt werden.

5.2.2.3 END - Eingabe von Anweisungen beenden

Diese Anweisung beendet die Eingabe von Anweisungen des Programms JMP.

Format

END

5.2.2.4 OPEN-JOBPOOL-FILE - Jobpool-Datei öffnen

Mit dieser Anweisung wird der zu rekonstruierende Jobpool angegeben.

Format

OPEN-JOBPOOL-FILE
FILE-NAME = *STD-FILE-LINK / <filename 1..54 without-gen-vers>

Operanden

FILE-NAME = *STD-FILE-LINK / <filename 1..54 without-gen-vers>
Dateiname des zu rekonstruierenden Jobpools. Für die Datei muss Lesezugriff erlaubt sein.

FILE-NAME = *STD-FILE-LINK
Der Dateiname soll aus der Task File Table (TFT) gelesen werden.
Der vom Benutzer nicht veränderbare Linkname ist SJOBPOOL.
Der Dateiname kann somit vom Benutzer vor Aufruf des JMP durch `/ADD-FILE-LINK LINK-NAME=SJOBPOOL, FILE-NAME=<dateiname>` vereinbart werden.

FILE-NAME = <filename 1..54 without-gen-vers>
Vollqualifizierter Dateiname. Die Angabe einer Dateigeneration oder Dateigenerationsgruppe und die Angabe des Dateinamens in der Form „datei(nr)“ (nr = Versionsnummer) ist nicht erlaubt.

5.2.2.5 SHOW-JOBPOOL-STATUS - Informationen über den Jobpool abrufen

Mit dieser Anweisung können Informationen über eine Teilmenge der Jobbeschreibungen in der aktuell geöffneten Jobpool-Datei ausgegeben werden.

Die Jobs werden mit dem Operanden JOB-IDENTIFICATION anhand bestimmter Jobattribute ausgewählt. Die Menge der ausgegebenen Attribute lässt sich mit dem Operanden INFORMATION steuern. Das Ausgabeziel wird mit dem Operanden OUTPUT bestimmt.

Format

```
SHOW-JOBPOOL-STATUS

 JOB-IDENTIFICATION = *ALL / *NONE / *TSN(...) / *JOB-STATE (...) / *USER-IDENTIFICATION(...) /
                      *JOB-CLASS(...) / *JOB-NAME(...) / *PUBSET-OF-SYSCMD-FILE(...)
     *TSN(...)
         |  TSN = <alphanum-name 1..4>
     *JOB-STATE(...)
         |  JOB-STATE = *EXECUTING / *WAITING / *DORMANT / *REPEAT / *CALENDAR
     *USER-IDENTIFICATION(...)
         |  USER-IDENTIFICATION = <alphanum-name 1..8>
     *JOB-CLASS(...)
         |  JOB-CLASS = <alphanum-name 1..8>
     *JOB-NAME(...)
         |  JOB-NAME = <alphanum-name 1..8>
     *PUBSET-OF-SYSCMD-FILE(...)
         |  PUBSET-OF-SYSCMD-FILE = <alphanum-name 1..4>
 ,INFORMATION = *SUMMARY / *JOB-LIST / *FULL
 ,OUTPUT = list-poss(2): *SYSOUT / *SYSLST
```

Operanden

JOB-IDENTIFICATION = *ALL / *NONE / *TSN(...) / *JOB-STATE (...) / *USER-IDENTIFICATION(...) / *JOB-CLASS(...) / *JOB-NAME(...) / *PUBSET-OF-SYSCMD-FILE(...)
Anhand der angegebenen Parameter werden die Jobs ausgewählt, über die Informationen ausgegeben werden.

JOB-IDENTIFICATION = *ALL
Alle Aufträge im Jobpool sind ausgewählt.

JOB-IDENTIFICATION = *NONE
Falls INFORMATION=*SUMMARY gewählt ist, werden Informationen über die aktuelle Jobpool-Datei ausgegeben. Ansonsten erfolgt keine Ausgabe.

JOB-IDENTIFICATION = *TSN(...)
Der Auftrag wird über die TSN ausgewählt.

TSN = <alphanum-name 1..4>
TSN des Auftrags der ausgewählt werden soll.

JOB-IDENTIFICATION = *JOB-STATE(...)
Die Jobs werden anhand des Zustands bzw. Typs der Aufträge aus JMS-Sicht ausgewählt.

JOB-STATE = *EXECUTING
Alle aktiven Aufträge, zu denen bereits eine Task im System war, werden ausgewählt.

JOB-STATE = *WAITING
Alle Aufträge, die unter der Kontrolle eines Jobstreams auf ihren Systemstart warteten, werden ausgewählt.

¡ Wartende Repeatjob-Wiederholungen mit Wiederholungszähler > 0 werden nicht ausgewählt.

JOB-STATE = *DORMANT
Alle Aufträge, deren Jobstream inaktiv war, sowie nicht aktive Repeatjob-Wiederholungen werden ausgewählt.

JOB-STATE = *REPEAT
Alle Repeatjobs werden ausgewählt.

JOB-STATE = *CALENDAR
Alle Kalenderjobs werden ausgewählt.

USER-IDENTIFICATION = *USER-IDENTIFICATION(...)
Alle Aufträge, die unter der angegebenen Kennung laufen sollen, werden ausgewählt.

USER-IDENTIFICATION = <alphanum-name 1..8>
Benutzerkennung, unter der die ausgewählten Aufträge laufen sollen.

JOB-IDENTIFICATION = *JOB-CLASS(...)
Alle Aufträge, die zu der angegebenen Jobklasse gehören, werden ausgewählt.

JOB-CLASS = <alphanum-name 1..8>
Jobklasse der Aufträge, die ausgewählt werden sollen.

JOB-IDENTIFICATION = *JOB-NAME(...)
Alle Aufträge mit dem angegebenen Jobnamen sind ausgewählt.

JOB-NAME = *NONE / <alphanum-name 1..8>
Name der Jobs, die ausgewählt werden sollen. Mit dem Wert *NONE werden Jobs ohne Namen ausgewählt.

JOB-IDENTIFICATION = *PUBSET-OF-SYSCMD-FILE(...)
Alle Aufträge, deren Kommandodatei auf dem angegebenen Pubset liegt, werden ausgewählt.

PUBSET-OF-SYSCMD-FILE = <alphanum-name 1..4>
Kennung des Pubsets, auf dem die Kommandodatei der augewählten Jobs liegt.

INFORMATION = *SUMMARY / *JOB-LIST / *FULL
Es wird spezifiziert, wieviel Informationen über die ausgewählten Aufträge ausgegeben wird.

INFORMATION = *SUMMARY
Die ausgegebenen Informationen werden auf Ausgabe des Namens, Version der aktuellen Jobpool-Datei und die Anzahl der ausgewählten Jobs beschränkt.
Wenn JOB-INFORMATION=*NONE gewählt ist, entfällt die Angabe über Anzahl der Jobs (siehe Beispiel 1).

INFORMATION = *JOB-LIST
Es wird eine Überschriftzeile, eine Zeile mit 9 Attributen für jeden ausgewählten Job und eine Zeile, welche die Anzahl dieser Jobs zusammenfasst, ausgegeben (siehe Beispiel 2).

INFORMATION = *FULL
Für jeden ausgewählten Job werden in bis zu 12 Ausgabezeilen die wichtigsten Jobattribute ausgegeben.
Die Ausgabe wird durch eine Zeile mit der Anzahl der Jobs abgeschlossen (siehe Beispiel 3).

OUTPUT = list-poss(2): *SYSOUT / *SYSLST
Bestimmt, wohin die Ausgabe erfolgen soll.

OUTPUT = *SYSOUT
Die Ausgabe soll nach SYSOUT erfolgen.

OUTPUT = *SYSLST
Die Ausgabe soll nach SYSLST erfolgen.

Beispiel 1

```
%//show-jobpool-status *none,*summary
JOBPOOL :  :2OS6:$YLA.SYSTEM.JOBPOOL.V160
VERSION :  OSD V3.0 OR HIGHER
```

Die betrachtete JOBPOOL-Datei kann von OSD-Versionen >= V3.0 verwendet werden.

Beispiel 2

```
%//show-jobpool-status *job-class(jcjs2xsb),*job-list
TSN  JNAME    USER     JCLASS   CAT   TYPE  START        REPEAT LTSN
0AVX          TSOS     JCJS2XSB G     2     SOON
0AVZ          TSOS     JCJS2XSB G     1 WT  A040809.2300
0AV0          TSOS     JCJS2XSB G     1 WT  E041010.2222
0AV2          TSOS     JCJS2XSB G     2     W0345
0AV3          TSOS     JCJS2XSB G     1 WT  A000809.1200 CALNDR
0AV4          TSOS     JCJS2XSB G     1 DO  STUP         STUP
0AV6 R1       TSOS     JCJS2XSB G     1 DO  A040810.1400 DAILY  0AV5
0AV5 R1       TSOS     JCJS2XSB G     1 WT  A040809.1400 DAILY
0AV8          TSOS     JCJS2XSB G     1 DO  SOON         0234   0AV7
0AV7          TSOS     JCJS2XSB G     2     SOON         0234
0AWB          TSOS     JCJS2XSB G     1 WT  A000809.1600 CALNDR
            11 JOB(S) DISPLAYED
```

Erläuterungen

Die betrachtete JOBPOOL-Datei enthält 11 Jobbeschreibungen in der Jobklasse JCJS2XSB. Die Jobattribute werden nach einer Überschriftszeile ausgegeben:

TSN: Die Jobs werden in der Reihenfolge ausgegeben, in der sie in der Jobpool-Datei stehen. Die TSNs sind daher i.A. nicht sortiert.

JNAME: Job-Name

USER: Benutzerkennung

JCLASS: Jobklasse

CAT: Katalogkennung des Pubsets, auf dem die Kommandodatei steht.

TYPE:

1 WT:	Der Job wartet im Jobstream auf Anstart, entspricht dem Auswahlkriterium JOB-STATE=*INACTIVE.
1 DO:	Der Jobstream des Jobs ist nicht aktiv oder der Job ist ein wartender Repeatjob-Nachfolger (*DORMANT).
1 HO:	Der Job wurde mit HOLD-JOB angehalten.
1 HOC:	Der Job ist im Ressourcen-HOLD, weil bei der Terminierung auf seine Kalenderdatei nicht zugegriffen werden konnte.
2 :	Der Job ist aktiv (*ACTIVE).

START: Das START-Attribut in folgenden Formaten:

SOON	
IMME	IMMEDIATELY
STUP	AT-STREAM-STARTUP
Whhmm	WITHIN(HOURS=hh,MINUTES=mm)
Ayymmdd.hhmm	AT(DATE=yyyy-mm-dd,TIME=hh:mm)
Eyymmdd.hhmm	EARLIEST(.. see AT ..)
Lyymmdd.hhmm	LATEST(.. see AT ..)

REPEAT: Das REPEAT-Attribut in folgenden Formaten:

DAILY, WEEKLY	
STUP	AT-STREAM-STARTUP
hhmm	PERIOD(HOURS=hh,MINUTES=mm)
CALENDR	Der Job ist ein Kalenderjob.

LTSN: Nur für Repeatjob-Wiederholungen, deren Vorgänger ebenfalls im Jobpool war: TSN des Vorgängers („Last“ TSN).

- Falls die betrachtete Jobpool-Datei mit MOVE-JOBS erstellt wurde, wurde bei Repeatjobs von den maximal zwei Ausprägungen immer nur die mit dem höchsten Zähler übernommen (das NEXT-Image im Typ 1 DO). Sie enthält die gesamte Repeat-Information. In diesem Fall fehlen im Allgemeinen die Jobs mit der TSN <LTSN>.

Bei IMPORT oder Warmstart wird das NEXT-Image in seinen Vorgänger zurückverwandelt, der dann wieder die TSN <LTSN> bekommt (sofern sie noch frei ist).

- Informationen über Repeatjobs, die von `/SHOW-JOB-STATUS` zur Verfügung gestellt werden, sind nicht abrufbar. Insbesondere ist die Verknüpfung vom Vorgänger- zum Nachfolger-Repeatjob (Ausgabeparameter START und NTSN) in der Jobpool-Datei nicht abrufbar. Die Verknüpfung kann nur vom Nachfolger-Repeatjob zum Vorgänger-Repeatjob über dessen TSN (Ausgabeparameter LTSN) erfolgen.

Beispiel 3

```
%//show-jobpool-status *t(0avx),*full
 TSN:     0AVX
 JOBNAME:            TYPE:    2     USER-ENTER                              1.
 USERID:  TSOS        ACCNB:   ADMINSTR   JCLASS:  JCJS2XSB
 PRI:     8 230       CPU-MAX: 200        START:   SOON
 RERUN:   NO          FLUSH:   NO         SPOOLIN: <date>.<time>
 CREATOR: TSOS        PROTECT: *NONE      LOGON:   <date>.<time>              2.
 ORIGFILE::G:$TSOS.OS232.E
 CMD-FILE:*SAME
          1 JOB(S) DISPLAYED
//show-jobpool-status *t(0awb),*full
 TSN:     0AWB
 JOBNAME:            TYPE:    1 WT  USER-ENTER
 USERID:  TSOS        ACCNB:   ADMINSTR   JCLASS:  JCJS2XSB
 PRI:     8 230       CPU-MAX: 200        START:    A<date>.<time>
 RERUN:   NO          FLUSH:   NO         SPOOLIN:  <date>.<time>
 CREATOR: TSOS        PROTECT: *NONE      LOGON:
 REPEAT:  CALENDAR    COUNT:   0          LIMIT:   *STD                       3.
 SYMDAT:  SYM.16.00
 CAL-NAME::G:$TSOS.ULTIMATIVER.CALENDAR
 ORIGFILE::G:$TSOS.S.199.0AV7.E
 CMD-FILE::G:$TSOS.S.E.0AV7.<date>.<time>
 MONJV:   :G:$TSOS.JV
          1 JOB(S) DISPLAYED
//show-jobpool-status *t(0av6),*full
 TSN:     0AV6
 JOBNAME:            TYPE:    1 DO  USER-ENTER
 USERID:  TSOS        ACCNB:   ADMINSTR   JCLASS:  JCJS2XSB
 PRI:     8 230       CPU-MAX: 200        START:   A<date>.<time>
 RERUN:   NO          FLUSH:   NO         SPOOLIN:  <date>.<time>
 CREATOR: TSOS        PROTECT: *NONE      LOGON:
 REPEAT:  DAILY       COUNT:   1          LTSN:    0AV5                       4.
 RTIME:   <date>.<time>
 ORIGFILE::G:$TSOS.OS232.E
 CMD-FILE::G:$TSOS.S.E.0AV2.<date>.<time>
```

`1 JOB(S) DISPLAYED`

Erläuterungen

Die meisten der ausgegebenen Attribute orientieren sich an der Ausgabe von `/SHOW-JOB-STATUS`, unterscheiden sich aber in ihrer Anordnung. Wesentliche Unterschiede sind:

1.	Auf die Ausgabe des TYPE wie in Beispiel 2 folgt eine Angabe zur Herkunft des Jobs:	
	`USER-ENTER:`	Wurde mit ENTER-JOB oder ENTER-PROCEDURE von einer Benutzertask erzeugt. Den Fall ENTER-PROCEDURE erkennt man am Namen der CMD-FILE: S.E.<tsn>. <yyyy-mm-dd>.<hh.mm.ss>.
	`PRIVILEGED-ENTER:`	Wurde vom System mit privilegiertem ENTER erzeugt, beispielsweise die Jobscheduler-Tasks.

	CONSOLE-ENTER:	Wurde von einer Benutzertask mit OPERATING-Privileg unter Verwendung des Operanden DEFAULT-FROM-FILE = *YES oder von einer Konsole aus erzeugt.

2.	CREATOR:	Die Kennung, unter der die erzeugende Task lief.Der Job ist auch von dieser Kennung aus administrierbar.
	PROTECT:	*CANCEL : Der Job wurde mit dem ENTER-Operanden PROTECTION=*CANCEL gestartet. *NONE : Der Job wurde mit dem ENTER-Operanden PROTECTION=*NONE gestartet.
3.	REPEAT:	Zusätzlich zu den ursprünglichen Repeat-Attributen werden hier auch Kalenderjobs gezeigt.
	LIMIT:	Wird nur bei Kalenderjobs ausgegeben, Bedeutung wie bei SHOW-JOB-STATUS.
4.	LTSN:	Verweist bei Repeatjobs auf den Vorgänger, vgl. Beispiel 2

5.3 Hinweise zu den rekonstruierten Attributen

Am folgenden Beispiel eines rekonstruierten Jobpools werden Hinweise zur Rekonstruktion einzelner Jobattribute und zu Problemfällen gegeben. In die Prozedurdatei schreibt JMP die folgenden Kommandos:

```
/"REJECTED:    TSN:  0EY4  TYPE:   1 DO       REASON: NEXT RPT. IMAGE "   1.
/SET-JOB-STEP " ------------------------------------------- "              2.
/"IMPORTED:    TSN:  0EY3  TYPE:   1 DO       ORIGIN: USER-ENTER      "   3.
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 0   START: <date>.<time>"
/ENTER-JOB                                                             -  4.
/  FROM-FILE = :2BV:$TSOS.OS27.E                                       -  5.
/ ,PROC-ADMI = *PAR(                                                   -  6.
/    USER-ID = TSOS                                                    -
/   ,ACCOUNT = ADMINSTR                                                -
/   ,PASS    = '????????')                                             -  7.
/ ,FILE-PASS = *NONE                                                   -  7.
/ ,DELETE    = *NO                                                     -
/ ,JOB-CLASS = JCBTSOS                                                 -
/ ,JOB-NAME  = OS27KF                                                  -
/ ,MONJV     = *NONE                                                   -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 5                                                       -
/ ,RERUN-AFTER-CRASH    = *NO                                          -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                          -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *AT       (DATE = *TODAY    , TIME = 11:00)             -  8.
/   ,REP-JOB = *DAILY )                                                -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  210                                                 -
/   ,CPU-LIMIT  =  20000                                               -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,PROTECTION= *NONE
/"REJECTED:    TSN:  0EYR  TYPE:   2          REASON: PRIVILEGED ENTER"
/"REJECTED:    TSN:  0EYS  TYPE:   2          REASON: PRIVILEGED ENTER"
/"REJECTED:    TSN:  0EYT  TYPE:   2          REASON: PRIVILEGED ENTER"
/"REJECTED:    TSN:  0EYU  TYPE:   2          REASON: PRIVILEGED ENTER"
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EYZ  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EYX  USERID: TSOS       HOST:                   "
/ENTER-JOB                                                              -
/  FROM-FILE = :2BV:$TSOS.SYSENT.TCP-IP-AP.031.FTPD                     -
/ ,PROC-ADMI = *PAR(                                                    -
/    USER-ID = TSOS                                                     -
/   ,ACCOUNT = ADMINSTR                                                 -
/   ,PASS    = '????????')                                              -
/ ,FILE-PASS = *NONE                                                    -
/ ,DELETE    = *NO                                                      -
/ ,JOB-CLASS = JCBTSOS                                                  -
/ ,JOB-NAME  = FTPSR                                                    -
/ ,MONJV     = *NONE                                                    -
/ ,JV-PASS   = *NONE                                                    -
/ ,JOB-PRIO  = 5                                                        -
/ ,RERUN-AFTER-CRASH    = *NO                                           -
```

```
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                          -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *IMMEDIATELY                                            -  9.
/   ,REP-JOB = *NO )                                                   -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  120                                                 -
/   ,CPU-LIMIT  =  *NO                                                 -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EYW  TYPE:   1 DO       ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0AAU  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 51  START: 0000-00-00.0000"
/ENTER-JOB                                                             -
/  FROM-FILE = :2BV:$SYSPRIV.SYSENT.HOLD-SAT                           -
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = SYSPRIV                                                 -
/   ,ACCOUNT = SYSACC                                                  -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *NO                                                     -
/ ,JOB-CLASS = JCBSTD                                                  -
/ ,JOB-NAME  = HOLDSAT                                                 -
/ ,MONJV     = *NONE                                                   -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 9                                                       -
/ ,RERUN-AFTER-CRASH    = *NO                                          -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                          -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *AT-STREAM-STARTUP                                      -
/   ,REP-JOB = *AT-STREAM-STARTUP )                                    -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  220                                                 -
/   ,CPU-LIMIT  =  32000                                               -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EYY  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EYX  USERID: TSOS       HOST:                   "
/ENTER-JOB                                                             -
/  FROM-FILE = :2BV:$TSOS.SYSENT.TCP-IP-AP.031.TELNETD                 -
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = TSOS                                                    -
/   ,ACCOUNT = ADMINSTR                                                -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *NO                                                     -
/ ,JOB-CLASS = JCBTSOS                                                 -
/ ,JOB-NAME  = TELSR                                                   -
/ ,MONJV     = *NONE                                                   -
/ ,JV-PASS   = *NONE                                                   -
```

```
/ ,JOB-PRIO  = 5                                                    -
/ ,RERUN-AFTER-CRASH    = *NO                                       -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                       -
/ ,SCHEDULING-TIME = *PAR(                                          -
/    START   = *IMMEDIATELY                                         -
/   ,REP-JOB = *NO )                                                -
/ ,LIMIT =     *STD                                                 -
/ ,RESOURCES = *PAR(                                                -
/    RUN-PRIO   =  120                                              -
/   ,CPU-LIMIT  =  *NO                                              -
/   ,SYSLST-LIM =  *NO                                              -
/   ,SYSOPT-LIM =  *NO )                                            -
/ ,LOGGING = *PAR(                                                  -
/    LISTING = *NO )                                                -
/ ,PROTECTION= *NONE
/"REJECTED:    TSN:  0EY6  TYPE:   1 DO       REASON: NEXT RPT. IMAGE "
/SET-JOB-STEP " ------------------------------------------ "
/"IMPORTED:    TSN:  0EY5  TYPE:   1 DO       ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 0   START: <date>.<time>"
/ENTER-JOB                                                          -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                    -
/ ,PROC-ADMI = *PAR(                                                -
/    USER-ID = TSOS                                                 -
/   ,ACCOUNT = ADMINSTR                                             -
/   ,PASS    = '????????')                                          -
/ ,FILE-PASS = *NONE                                                -
/ ,DELETE    = *NO                                                  -
/ ,JOB-CLASS = JCBTSOS                                              -
/ ,JOB-NAME  = OS27KF                                               -
/ ,MONJV     = *NONE                                                -
/ ,JV-PASS   = *NONE                                                -
/ ,JOB-PRIO  = 5                                                    -
/ ,RERUN-AFTER-CRASH    = *NO                                       -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                       -
/ ,SCHEDULING-TIME = *PAR(                                          -
/    START   = *AT       (DATE = *TODAY    , TIME = 11:00)          -
/   ,REP-JOB = *WEEKLY )                                            -
/ ,LIMIT =     *STD                                                 -
/ ,RESOURCES = *PAR(                                                -
/    RUN-PRIO   =  210                                              -
/   ,CPU-LIMIT  =  20000                                            -
/   ,SYSLST-LIM =  *NO                                              -
/   ,SYSOPT-LIM =  *NO )                                            -
/ ,LOGGING = *PAR(                                                  -
/    LISTING = *NO )                                                -
/ ,PROTECTION= *NONE
/"REJECTED:    TSN:  0EY8  TYPE:   1 DO       REASON: NEXT RPT. IMAGE "
/SET-JOB-STEP " ------------------------------------------ "
/"IMPORTED:    TSN:  0EY7  TYPE:   1 DO       ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 0   START: <date>.<time>"
/ENTER-JOB                                                          -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                    -
/ ,PROC-ADMI = *PAR(                                                -
/    USER-ID = TSOS                                                 -
/   ,ACCOUNT = ADMINSTR                                             -
/   ,PASS    = '????????')                                          -
/ ,FILE-PASS = *NONE                                                -
```

```
/ ,DELETE    = *NO                                                    -
/ ,JOB-CLASS = JCBTSOS                                                -
/ ,JOB-NAME  = OS27KF                                                 -
/ ,MONJV     = *NONE                                                  -
/ ,JV-PASS   = *NONE                                                  -
/ ,JOB-PRIO  = 5                                                      -
/ ,RERUN-AFTER-CRASH    = *NO                                         -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                         -
/ ,SCHEDULING-TIME = *PAR(                                            -
/    START   = *SOON                                                  -
/   ,REP-JOB = *PERIOD (HOURS = 01, MINUTES = 02))                    -
/ ,LIMIT =     *STD                                                   -
/ ,RESOURCES = *PAR(                                                  -
/    RUN-PRIO   =  210                                                -
/   ,CPU-LIMIT  =  20000                                              -
/   ,SYSLST-LIM =  *NO                                                -
/   ,SYSOPT-LIM =  *NO )                                              -
/ ,LOGGING = *PAR(                                                    -
/    LISTING = *NO )                                                  -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------ "
/"IMPORTED:    TSN:  0EY9  TYPE:   1 WT       ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/ENTER-JOB                                                             -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                       -
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = TSOS                                                    -
/   ,ACCOUNT = ADMINSTR                                                -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *NO                                                     -
/ ,JOB-CLASS = JCBTSOS                                                 -
/ ,JOB-NAME  = OS27KF                                                  -
/ ,MONJV     = *NONE                                                   -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 5                                                       -
/ ,RERUN-AFTER-CRASH    = *NO                                          -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                          -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *EARLIEST (DATE = <date>, TIME = <time>)                -
/   ,REP-JOB = *NO )                                                   -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  210                                                 -
/   ,CPU-LIMIT  =  20000                                               -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------ "
/"IMPORTED:    TSN:  0EZA  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 0   START: <date>.<time>"
/ENTER-JOB                                                             -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                       -
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = TSOS                                                    -
/   ,ACCOUNT = ADMINSTR                                                -
```

```
/   ,PASS    = '????????')                                                -
/ ,FILE-PASS = *NONE                                                      -
/ ,DELETE    = *NO                                                        -
/ ,JOB-CLASS = JCBTSOS                                                    -
/ ,JOB-NAME  = OS27KF                                                     -
/ ,MONJV     = *NONE                                                      -
/ ,JV-PASS   = *NONE                                                      -
/ ,JOB-PRIO  = 5                                                          -
/ ,RERUN-AFTER-CRASH    = *NO                                             -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                             -
/ ,SCHEDULING-TIME = *BY-CALENDAR(                                        -
/    CALENDAR-NAME =      :2BV:$TSOS.OS27.CALENDAR                        -
/   ,SYMBOLIC-DATE =      HEMUL                )                          -
/ ,LIMIT =     *STD                                                       -
/ ,RESOURCES = *PAR(                                                      -
/    RUN-PRIO   =  210                                                    -
/   ,CPU-LIMIT  =  20000                                                  -
/   ,SYSLST-LIM =  *NO                                                    -
/   ,SYSOPT-LIM =  *NO )                                                  -
/ ,LOGGING = *PAR(                                                        -
/    LISTING = *NO )                                                      -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EZB  TYPE:   1 WT       ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" REPETITION: REPCNT: 0   START: <date>.<time>"
/ENTER-JOB                                                                -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                          -
/ ,PROC-ADMI = *PAR(                                                      -
/    USER-ID = TSOS                                                       -
/   ,ACCOUNT = ADMINSTR                                                   -
/   ,PASS    = '????????')                                                -
/ ,FILE-PASS = *NONE                                                      -
/ ,DELETE    = *NO                                                        -
/ ,JOB-CLASS = JCBTSOS                                                    -
/ ,JOB-NAME  = OS27KF                                                     -
/ ,MONJV     = *NONE                                                      -
/ ,JV-PASS   = *NONE                                                      -
/ ,JOB-PRIO  = 5                                                          -
/ ,RERUN-AFTER-CRASH    = *NO                                             -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                             -
/ ,SCHEDULING-TIME = *BY-CALENDAR(                                        -
/    CALENDAR-NAME =      :2BV:$TSOS.OS27.CALENDAR                        -
/   ,SYMBOLIC-DATE =      ELCH                 )                          -
/ ,LIMIT =     *STD                                                       -
/ ,RESOURCES = *PAR(                                                      -
/    RUN-PRIO   =  210                                                    -
/   ,CPU-LIMIT  =  20000                                                  -
/   ,SYSLST-LIM =  *NO                                                    -
/   ,SYSOPT-LIM =  *NO )                                                  -
/ ,LOGGING = *PAR(                                                        -
/    LISTING = *NO )                                                      -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EZC  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/ENTER-JOB                                                                -
/  FROM-FILE = :2BV:$TSOS.OS27.E                                          -
/ ,PROC-ADMI = *PAR(                                                      -
```

```
/    USER-ID = TSOS                                                    -
/   ,ACCOUNT = ADMINSTR                                                -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *NO                                                     -
/ ,JOB-CLASS = JCB02000                                                -
/ ,JOB-NAME  = OS27KF                                                  -
/ ,MONJV     = :2BV:$TSOS.OS27.JV.1                                    -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 9                                                       -
/ ,RERUN-AFTER-CRASH    = *YES                                         -
/ ,FLUSH-AFTER-SHUTDOWN = *YES                                         -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *SOON                                                   -
/   ,REP-JOB = *NO )                                                   -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  210                                                 -
/   ,CPU-LIMIT  =  2000                                                -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,JOB-PAR =                                                           -
/C'WRTLPRMFT'                                                          -
/ ,PROTECTION= *NONE
/SET-JOB-STEP " ------------------------------------------- "
/"IMPORTED:    TSN:  0EZD  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/ENTER-JOB                                                              -
/  FROM-FILE = :2BV:$HEMUL.S.IN.0EZD.040323.0913.C                      -  10.
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = HEMUL                                                   -
/   ,ACCOUNT = HEMUL                                                   -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *YES                                                    -
/ ,JOB-CLASS = JCB02000                                                -
/ ,JOB-NAME  = HOPPLA                                                  -
/ ,MONJV     = :2BV:$TSOS.OS27.JV.2                                    -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 9                                                       -
/ ,RERUN-AFTER-CRASH    = *YES                                         -
/ ,FLUSH-AFTER-SHUTDOWN = *YES                                         -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *SOON                                                   -
/   ,REP-JOB = *NO )                                                   -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  230                                                 -
/   ,CPU-LIMIT  =  2000                                                -
/   ,SYSLST-LIM =  *NO                                                 -
/   ,SYSOPT-LIM =  *NO )                                               -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *NO )                                                   -
/ ,JOB-PAR =                                                           -
/C'OCHGOTTCHEN'                                                        -
/ ,PROTECTION= *CANCEL
/SET-JOB-STEP " ------------------------------------------- "
```

```
/"IMPORTED:    TSN:  0EZE  TYPE:   2          ORIGIN: USER-ENTER      "
/" CALLER:     TSN:  0EY2  USERID: TSOS       HOST:                   "
/" WARNING:    COPY OF SYSCMD-FILE NOT POSSIBLE; COPY-RC: 0501"          11.
/ENTER-JOB                                                             -
/  FROM-FILE = :2BV:$HEMUL.S.IN.0EZE.040323.0929                       -
/ ,PROC-ADMI = *PAR(                                                   -
/    USER-ID = HEMUL                                                   -
/   ,ACCOUNT = HEMUL                                                   -
/   ,PASS    = '????????')                                             -
/ ,FILE-PASS = *NONE                                                   -
/ ,DELETE    = *YES                                                    -
/ ,JOB-CLASS = JCB00050                                                -
/ ,JOB-NAME  = OS27KF                                                  -
/ ,MONJV     = *NONE                                                   -
/ ,JV-PASS   = *NONE                                                   -
/ ,JOB-PRIO  = 9                                                       -
/ ,RERUN-AFTER-CRASH    = *YES                                         -
/ ,FLUSH-AFTER-SHUTDOWN = *NO                                          -
/ ,SCHEDULING-TIME = *PAR(                                             -
/    START   = *EARLIEST (DATE = *TODAY    , TIME = 08:00)             -
/   ,REP-JOB = *NO )                                                   -
/ ,LIMIT =     *STD                                                    -
/ ,RESOURCES = *PAR(                                                   -
/    RUN-PRIO   =  233                                                 -
/   ,CPU-LIMIT  =  100                                                 -
/   ,SYSLST-LIM =  2222                                                -
/   ,SYSOPT-LIM =  3333     )                                          -
/ ,LOGGING = *PAR(                                                     -
/    LISTING = *YES)                                                   -
/ ,PROTECTION= *NONE
```

Erläuterungen

1. Nicht alle Jobs, die der Jobpool enthält, werden rekonstruiert. Im vorliegenden Beispiel existiert zu einem Repeatjob bereits das „Next Image“. Würde dieser Job zu einem ENTER-Kommando rekonstruiert und gestartet, käme es zu einer Verdopplung der Ausprägungen des Repeatjobs. Der Hinweis auf solche nicht rekonstruierten Jobs erfolgt in Form eines Kommandokommentars mit Informationen zu diesem Job und einer Auflistung der Ursachen wie z.B.:
   - Next Repeat Image
   - Privilegierter ENTER
2. Beginn eines rekonstruierten Batch-Auftrages.

3. Hinweise zu dem rekonstruierten Batch-Auftrag und Attribute, die nicht in Form von Operanden des ENTER-Kommandos angegeben werden können. Die Informationen sind in der Regel analog zu denen der STATUS-Kommandos aufgebaut (siehe Handbuch „Kommandos“ [1]).

   Im vorliegenden Beispiel:

   - IMPORTED: Hinweis, dass der Job, der mit der TSN `0EY3` akzeptiert war, rekonstruiert wurde.
   - CALLER: Hinweise auf den Initiator des Batch-Auftrages (TSN, Benutzerkennung, Host-Name).
   - REPETITION: Nur bei Repeatjobs und Kalenderjobs, Wiederholungszähler und designierter Startzeitpunkt.

   > ¡ Termin-/Repeatjobs werden ungeachtet des Starttermins rekonstruiert. Liegt z.B. der Starttermin vor dem Rekonstruktionslauf und wird das rekonstruierte ENTER-Kommando ungeändert aktiviert, so gilt das Kommando als inkorrekt und wird bei der erneuten Akzeptierung zurückgewiesen.

4. Die Jobs werden grundsätzlich in Form des SDF-Kommandos `/ENTER-JOB` rekonstruiert (ungeachtet dessen, ob der Job mit dem ISP-Kommando `/ENTER` oder mit `/ENTER-PROCEDURE` gestartet wurde).
5. Die Kommandodatei muss nicht die vom Initiator des ENTER-Kommandos angegebene Originaldatei sein. Siehe hierzu auch unter (10).
6. Die mit dem Operanden PROCESSING-ADMISSION angegebenen Attribute für die Zugangskontrolle (Benutzerkennung, Abrechnungsnummer, Passwort) sind nur obligatorisch, wenn die Auftraggeber- und Ablaufbenutzerkennung verschieden sind. Bei Gleichheit der Benutzerkennungen ist die Angabe optional. Im rekonstruierten Kommando erscheint dieser Operand immer, unabhängig davon, ob er spezifiziert wurde oder nicht.

   Zu beachten ist, dass ein rekonstruiertes ENTER-JOB-Kommando nach Möglichkeit von derselben Benutzerkennung wieder abgegeben wird wie das ursprüngliche. Nur dann gelten kennungsspezifische Defaultierungen weiterhin. Ferner kann dieser Auftrag - wenn er auf einer anderen Kennung zum Ablauf gebracht werden soll - noch von derselben wie der ursprünglichen Auftraggeberkennung administriert werden (Kommandos `/CANCEL-JOB`, `/CHANGE-TASK-PRIORITY`, `/SHOW-JOB-STATUS`).
7. Weder das LOGON-Passwort noch eventuell angegebene Passwörter zum Schutz von ENTER-Datei bzw. MONJV werden in das rekonstruierte ENTER-Kommando übernommen. Dies ist aus technischen Gründen nicht möglich, weil in der Regel Passwörter in den Datenstrukturen des JMS nicht abgelegt sind. Bei Neustart des ENTER-Kommandos sind die Passwörter in unverschlüsselter Form einzusetzen.
8. Als Startzeitpunkte werden die Originalwerte (z.B. auch *TODAY) ohne Prüfung eingesetzt. Dies ist zu beachten, wenn im Original feste Termine oder Attribute wie *WITHIN, *LATEST angegeben wurden. Die rekonstruierten Werte können in Bezug auf den beabsichtigten Zeitpunkt des Wiederablaufs fehlerhaft sein.

   Bei Wiederholungsläufen von Repeatjobs ist das Startattribut gegenüber der ursprünglichen Angabe des Anwenders u.U. verändert (siehe Beschreibung von `/ENTER-JOB` und `/ENTER-PROCEDURE` im Handbuch „Kommandos“ [1]). In diesem Fall wird versucht, aus dem Repeat-Attribut und dem Startzeitpunkt der nächsten Wiederholung ein sinnvolles Startattribut abzuleiten (das Startattribut bestimmt nur den Start des ersten Laufes des Repeatjobs).
9. Wie oben (8) beschrieben, werden Startzeitpunkte von JMP nicht aktualisiert.

10. S.IN-/S.E-Dateien: Das JMS legt Hilfsdateien mit diesem Präfix unter folgenden Umständen an (siehe Beschreibung von `/ENTER-JOB` und `/ENTER-PROCEDURE` im Handbuch „Kommandos“ [1]):

    Bei Aufruf von `/ENTER-PROCEDURE` wird eine S.E-Datei angelegt, bei Aufruf von `/ENTER-JOB` wird die Originalkommandodatei in eine S.IN-Zwischendatei kopiert, wenn angenommen werden muss, dass die Originaldatei zum Zeitpunkt des Ablaufs des Batch-Auftrags nicht zugreifbar ist (z.B. wenn als Originalkommandodatei eine Temporärdatei angegeben wurde). Diese Zwischendateien werden mit einem Zufallspasswort geschützt. Daher kann bei erneutem Aufruf (des rekonstruierten ENTER-Kommandos) auf die ursprünglich vom JMS angelegte Zwischendatei nicht zugegriffen werden. JMP kopiert die Zwischendatei daher in eine neue Zwischendatei ohne Passwort mit dem Namen `<name ursprüngliche s.in-/s.e-Datei>.C` .
11. Ist das Erstellen einer Kopie der S.IN- bzw. S.E-Datei nicht möglich (vgl. Punkt 10), so wird eine Kommentarzeile mit dem DMS-Returncode (COPY) ausgegeben.

5.4 Meldungen von JMP

Die Meldungen des Dienstprogramms JMP haben die Meldungsklasse JMP.

Siehe auch den Abschnitt „Meldungen und ihre Bedeutung“ (Darstellungsmittel).

6 JMU Einrichten und Warten der Systemdatei SJMSFILE

Versionsstand: **JMU V21.0C**

Privilegierung: **TAPE-PROCESSING** (für nicht-privilegierte Funktionen)
TSOS (Änderungen im laufenden Betrieb und für $TSOS.SJMSFILE)

Mit dem Programm JMU (Job Management Utility) kann die Systemdatei SJMSFILE eingerichtet und verwaltet werden. Die SJMSFILE enthält die Stream- und Auftragsklassendefinitionen, die in ihr in einem internen Tabellenformat gespeichert sind. Das Programm kann im Batch- oder Dialogmodus ablaufen.

Bei der Systemeinleitung (BS2000-Startup) wird die Datei SJMSFILE gelesen und die Auftragsklassen- und Streamdefinitionen ins System übernommen.

Darüber hinaus können mit JMU bestimmte JMS-Daten im laufenden System geändert werden:

- Sie können Änderungen der Zugriffsrechte sofort wirksam werden lassen.
- Sie können neuen Benutzern geeignete Jobklassen zuordnen.
- Sie können Jobklassen und Jobstreams modifizieren, löschen und neu einführen.

6.1 Auftragsverwaltung

Die Auftragsverwaltung hat die Aufgabe, Aufträge bis zum Auftragsstart zu steuern.
Das Job Scheduling System mit Auftragsklassen ist ein Mittel, das ermöglicht, für das Data Center eine administrative Strategie zur Klassifizierung von Benutzern und der Systemlast vorzunehmen.

Aufträge, die in bestimmten Eigenschaften übereinstimmen, werden gleichen Auftragsklassen zugeordnet. Das gilt sowohl für Aufträge im Batch- als auch im Dialogbetrieb. Die Eigenschaften legt die Systemverwaltung bei der Definition der Klassen fest und bestimmt, welche Benutzerkennungen von der jeweiligen Klasse bedient werden sollen.
Es können auch Standardklassen definiert werden, die für Benutzer bestimmt sind, die keine Klasse explizit angegeben haben.
Durch Setzen eines Limits pro Klasse und Festlegen einer Klassenpriorität erzielt das Data Center eine verbesserte Zugangskontrolle zum System und kann zu jeder Tageszeit eine optimale Mischung von Aufträgen, z.B. Kurzläufer und Langläufer, vornehmen.

Mit Auftragsklassen ist eine Einteilung z.B. nach CPU-Zeitbedarf möglich, um Kurzläufer gegenüber Langläufern zu bevorzugen. Ebenso kann die Systemverwaltung unter den Benutzern Privilegien verteilen, wie z.B. Termin- oder Wiederholungsaufträge starten zu dürfen.

Für die Auftragsverwaltung steht das Dienstprogramm zum Einrichten und Warten der Datei für Stream- und Auftragsklassendefinition (JMU) zur Verfügung. Beschreibung der Jobstreams siehe Handbuch „Systembetreuung“ [5].

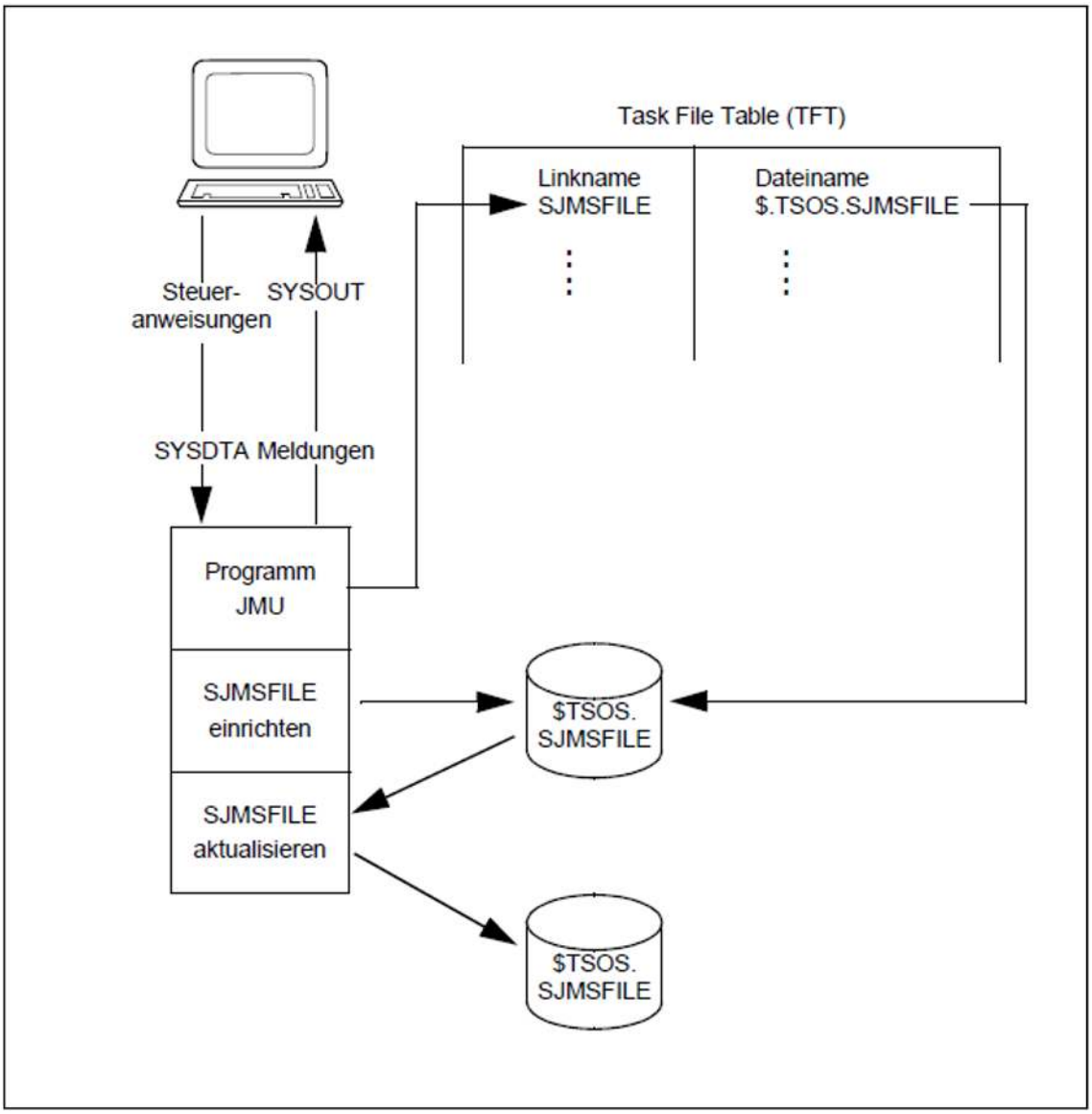


Bild 1: Einrichten und Aktualisieren der Datei SJMSFILE

6.2 Ablauf von JMU

JMU richtet die ISAM-Datei SJMSFILE ein. Der Dateiname wird aus der Task File Table (Linkname SJMSFILE) ermittelt. Er kann mit folgendem Kommando vereinbart werden:

`/ADD-FILE-LINK LINK-NAME=SJMSFILE,FILE-NAME=<filename>`

Die Datei muss vor Ausführung von `/ADD-FILE-LINK` existieren. Eine bereits vorhandene SJMSFILE aktualisiert JMU, eine leere Datei erstellt JMU neu.

`/ADD-FILE-LINK` ist nicht zwingend. Wenn der Linkname SJMSFILE nicht vergeben wurde, bearbeitet JMU wie bisher die Datei mit dem Dateinamen SJMSFILE und vergibt für diese selbst den Linknamen SJMSFILE.

Der Benutzer darf den Linknamen SJMSFILE nicht für eine Datei vergeben, die nicht von JMU bearbeitet werden soll.

Das Programm wird gestartet mit `/START-JMU`.

START-JMU	Alias: **JMU**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

JMU wird durch Steueranweisungen gesteuert, die von SYSDTA gelesen werden.

JMU verwendet die Dialogschnittstelle SDF (siehe Handbücher „Kommandos“ [1] und „Dialogschnittstelle SDF“ [20]). Durch syntaktische Fehler wird im Dialogbetrieb ein Korrekturdialog mit dem Benutzer angestoßen. Im Prozedur- und Batch-Betrieb ist ein Korrekturdialog nicht möglich, SDF erlaubt allerdings das Übergehen der Anweisung nicht. Im Fall einer fehlerhaften Anweisung werden alle folgenden Anweisungen bis zu einer STEP- oder der END-Anweisung übersprungen, erst die darauf folgende Anweisung (bzw. das darauf folgende Kommando) wird bearbeitet. Der Benutzer wird durch die Ausgabe der Meldung `CMD0230` über das Überspringen informiert.

Die Datei SJMSFILE wird im gleichen Systemlauf aktualisiert, in dem die Steueranweisungen gegeben wurden. Die Aktualisierung der Datei wirkt sich aber erst im nächsten Systemlauf aus.
Die Änderungen sollten in einer Kopie der SJMSFILE vorgenommen werden, und erst, wenn man ganz sicher ist, dass die Aktualisierung korrekt ist, soll die aktualisierte Datei im Systemlauf verwendet werden

Es empfiehlt sich, eine Kopie der Datei oder eine Prozedur zur Wiederherstellung des SJMSFILE zu behalten.

> ¡ Es besteht die Gefahr, dass die Datei SJMSFILE von der Auftragsverwaltung späterer Betriebssystemversionen als die, die für die Einrichtung der Datei verwendet wurde, nicht verarbeitet werden kann. Es ist daher zu empfehlen, eine Kopie der Steueranweisungen von JMU zum Wiederherstellen der Datei in einer neuen Betriebssystemversion aufzubewahren.

Verwenden von Linknamen durch JMU

JMU wertet die Linknamen SJMSFILE und SJMUPROC aus. Wenn der Linkname SJMSFILE in der Task File Table definiert ist, wird die zugehörige Datei von JMU verarbeitet, ansonsten wird SJMSFILE als Dateiname verwendet.

Bei der Bearbeitung der Anweisung CREATE-PROCEDURE-FILE wird der Linkname SJMUPROC ausgewertet und gegebenenfalls einer Datei zugewiesen. Die Verwendung dieses Linknamens sollte also auf die bei der Anweisung CREATE-PROCEDURE-FILE beschrieben Fälle beschränkt werden .

Kompatibilität

Bei der Bearbeitung einer SJMSFILE mit einer kleineren JMU-Version als die Version, die bei der Erstellung verwendet wurde, kann es zu Inkonsistenzen kommen.

6.3 Anweisungen

Eine Anweisung kann sich über mehrere Zeilen erstrecken. Als Fortsetzungszeichen dient der Bindestrich. Er kündigt eine Folgezeile an. Ihm dürfen nur Leerzeichen bis zum Zeilenende folgen.

6.3.1 Übersicht über die Anweisungen von JMU

Anweisung	Bedeutung
CREATE-PROCEDURE-FILE	SAM-Datei erzeugen, die eine BS2000-Prozedur enthält.
DEFINE-JOB-CLASS	Neue Auftragsklassendefinition schreiben.
DEFINE-JOB-STREAM	Neue Streamdefinition in die SJMSFILE schreiben.
DELETE-JOB-CLASS	Existierende Auftragsklassendefinition löschen.
DELETE-JOB-STREAM	Existierende Streamdefinition löschen.
END	Programm beenden.
GRANT-JOB-CLASS-ACCESS	Zugriff zu einer Auftragsklasse für einen oder mehrere Anwender zulassen oder sperren.
MODIFY-JOB-CLASS	Existierende Auftragsklassendefinition ändern.
MODIFY-JOB-STREAM	Existierende Streamdefinition ändern.
REMOVE-USER	Zugang zu nicht-öffentlichen Auftragsklassen sperren.
SET-JOB-CLASS-DEFAULT	Standardklassen für Benutzer spezifizieren.
SET-MODIFICATION-MODE	Bearbeitungsmodus ändern.
SET-POSIX-JOB-CLASS-DEFAULT	POSIX-Standardklassen für Benutzer spezifizieren.
SHOW-JOB-CLASS	Inhalt von Auftragsklassendefinitionen anfordern.
SHOW-JOB-STREAM	Inhalt von Streamdefinitionen anfordern.

6.3.2 Beschreibung der Anweisungen

- CREATE-PROCEDURE-FILE - SAM-Datei erzeugen, die eine BS2000-Prozedur enthält
- DEFINE-JOB-CLASS - Auftragsklassen-Definitionen in die Datei SJMSFILE schreiben
- DEFINE-JOB-STREAM - Streamdefinitionen in die Datei SJMSFILE schreiben
- DELETE-JOB-CLASS - Klassendefinitionen löschen
- DELETE-JOB-STREAM - Streamdefinitionen löschen
- END - Eingabe von Anweisungen beenden
- GRANT-JOB-CLASS-ACCESS - Zugriff von Benutzerkennungen zu einer Auftragsklasse regeln
- MODIFY-JOB-CLASS - Auftragsklassen-Definitionen ändern
- MODIFY-JOB-STREAM - Streamdefinitionen ändern
- REMOVE-USER - Zugang zu nichtöffentlichen Auftragsklassen sperren
- SET-JOB-CLASS-DEFAULT - Standardklassen für Benutzer spezifizieren
- SET-MODIFICATION-MODE - Änderungsmodus einstellen
- SET-POSIX-JOB-CLASS-DEFAULT - POSIX-Standardklassen für Benutzer spezifizieren
- SHOW-JOB-CLASS - Inhalt von Klassendefinitionen oder Namen von Klassen auflisten
- SHOW-JOB-STREAM - Inhalt von Streamdefinitionen oder Namen von Streams auflisten

6.3.2.1 CREATE-PROCEDURE-FILE - SAM-Datei erzeugen, die eine BS2000-Prozedur enthält

Mit dieser Anweisung kann eine SAM-Datei erzeugt werden, die eine BS2000-Prozedur enthält. Die Prozedur enthält ein START-Kommando für das Programm JMU.
Wird JMU beim Ablauf der Prozedur aufgerufen, wird eine neue Systemdatei SJMSFILE geschrieben, die die SJMSFILE ablöst, die zum Zeitpunkt vor dem Start der Prozedur in Bearbeitung war. Die neue SJMSFILE entspricht der, die zum Zeitpunkt der Ausführung der CREATE-PROCEDURE-FILE-Anweisung in Bearbeitung war.

Mit dieser Anweisung kann ein bestimmter Stand einer offenen SJMSFILE während der Bearbeitung in Form der erzeugenden BS2000-Prozedur festgehalten werden.

Mit der Anweisung CREATE-PROCEDURE-FILE kann das Format einer SJMSFILE aktualisiert werden. Bei der Bearbeitung von SJMSFILEs durch verschiedene JMU-Versionen wird das Format bestehender SJMSFILEs in keinem Fall verändert. Nur wenn eine SJMSFILE neu erstellt wird, dann bekommt sie das der verwendeten JMU-Version entsprechende Format.

Allerdings wird wegen der Funktionserweiterung des JMU eine Umwandlung des SJMSFILE-Formats nur unterstützt, wenn zur Ausführung der Anweisung CREATE-PROCEDURE-FILE und zur Abarbeitung der BS2000-Prozedur dieselbe JMU-Version verwendet wird.

Aus diesem Grund wird die JMU-Version, mit der die Prozedur erzeugt wurde, zu Dokumentationszwecken innerhalb eines REMARK-Kommandos festgehalten. Soll die Umwandlung in anderer Art erfolgen, müssen gegebenenfalls die JMU-Anweisungen in der erzeugten Prozedur mit der für die aufzurufende JMU-Version gültigen
Bedienungsanweisung abgeglichen werden.

Format

CREATE-PROCEDURE-FILE
FILE-NAME = *STD-FILE-LINK / <filename 1..54 without-gen-vers> ,OVERWRITE = *NO / *YES

Operanden

FILE-NAME =
Name der zu schreibenden Prozedurdatei. Für die Datei muss Schreibzugriff erlaubt sein.

FILE-NAME = *STD-FILE-LINK
Der Dateiname soll aus der Task File Table (TFT) gelesen werden. Der vom Benutzer nicht veränderbare Linkname ist SJMUPROC. Der Dateiname kann somit vom Benutzer vor Aufruf des JMU vereinbart werden durch:

```
/ADD-FILE-LINK LINK-NAME=SJMUPROC,FILE-NAME=<filename>
```

Ist SJMUPROC nicht als Linkname vergeben, verwendet JMU den Namen SJMUPROC als Dateinamen.

FILE-NAME = <filename 1..54 without-gen-vers>
Ein vollqualifizierter Dateiname. Die Angabe einer Dateigeneration oder Dateigenerationsgruppe und die Angabe des Dateinamens in der Form „datei(nr)“ (nr = Versionsnummer) ist nicht erlaubt.

OVERWRITE =
Das Überschreiben einer bereits existierenden Datei mit dem im Operanden FILE-NAME angegebenen Namen kann veranlasst oder verhindert werden.

OVERWRITE = *NO
Das Überschreiben einer bereits existierenden Datei wird verhindert. Die ursprüngliche Datei bleibt unverändert. Der Benutzer bekommt die Meldung `JMU0114`.

Die Prozedurdatei wird nicht erzeugt.

OVERWRITE = *YES
Besteht bereits eine Datei gleichen Namens, soll sie überschrieben und eine Prozedurdatei in jedem Fall erzeugt werden.

Aufbau der erzeugten BS2000-Prozedur

Als Prozedurparameter können für die Prozedur der Dateiname für die zu erzeugende SJMSFILE und die aufzurufende JMU-Phase angegeben werden.

Bedeutung der Parameter:

&SJMSFILE	bezeichnet den Dateinamen der zu erzeugenden SJMSFILE. Innerhalb der Prozedur wird die Datei mit dem Dateinamen &SJMSFILE gelöscht, um eine Neuerstellung durch JMU zu ermöglichen. Standardwert ist SJMSFILE.
&JMU	bezeichnet die innerhalb der Prozedur aufgerufene JMU-Phase. Standardwert ist $.JMU

BS2000-Kommandos innerhalb der Prozedur

Direkt anschließend an den Prozedurkopf stehen eine Reihe von REMARK-Kommandos, die einige Informationen über die zu erzeugende SJMSFILE enthalten: Der Dateiname der SJMSFILE, die durch die CREATE-PROCEDURE-FILE-Anweisung festgehalten wurde, Datum und Uhrzeit der Ausführung dieser Anweisung, die verwendete JMU-Version und einige Eigenschaften der SJMSFILE.

Es folgen die zur Ausführung der Prozedur notwendigen Kommandos und JMU-Anweisungen.

JMU-Anweisungen innerhalb der Prozedur

Für das Erzeugen der SJMSFILE werden die JMU-Anweisungen DEFINE-JOB-STREAM, DEFINE-JOB-CLASS, GRANT-JOB-CLASS-ACCESS, SET-JOB-CLASS-DEFAULT, SET-POSIX-JOB-CLASS-DEFAULT und END benutzt.

Die Reihenfolge der Anweisungen innerhalb der Prozedur ist eindeutig festgelegt: Zuerst kommen alle Streamdefinitionen in alphabetischer Reihenfolge. Dazu ist je eine DEFINE-JOB-STREAM-Anweisung erforderlich. Anschließend werden alle Auftrags-Klassen, wiederum in alphabetischer Reihenfolge, definiert. Dafür sind neben der DEFINE-JOB-CLASS-Anweisung auch noch gegebenenfalls die Anweisungen GRANT-JOB-CLASS-ACCESS, SET-JOB-CLASS-DEFAULT und SET-POSIX-JOB-CLASS-DEFAULT zur Festlegung der Zugriffsrechte der Benutzer auf die Klassen erforderlich. Die END-Anweisung beendet den JMU-Lauf.

6.3.2.2 DEFINE-JOB-CLASS - Auftragsklassen-Definitionen in die Datei SJMSFILE schreiben

Mit dieser Anweisung wird eine neue Auftragsklassen-Definition in die SJMSFILE bzw. JMS-Datenbasis geschrieben und deren Eigenschaften festgelegt.

Format

```
DEFINE-JOB-CLASS

 NAME = <name 1..8>

 ,STREAM = *DEFAULT-STREAM / <name 1..8>

 ,CLASS-LIMIT = <integer 0..4095>

 ,CLASS-WEIGHT = <integer 1..9>

 ,CLASS-OPTIMUM = 0 / <integer 0..4095>

 ,JOB-PRIORITY = *NO / *PARAMETERS(...)

     *PARAMETERS(...)

         |    DEFAULT = <integer 1..9>

         |    ,MAXIMUM = *NO / <integer 1..9>

 ,JOB-TYPE = *BATCH / *DIALOG

 ,TP-ALLOWED = *NO / *YES(...)

     *YES(...)

         |    CATEGORY = TP / <name 1..7>

 ,DIALOG-ALLOWED = *NO / *YES(...)

     *YES(...)

         |    CATEGORY = DIALOG / <name 1..7>

 ,BATCH-ALLOWED = *NO / *YES(...)

     *YES(...)

         |    CATEGORY = BATCH / <name 1..7>

 ,START-ATTRIBUTE = *BATCH / *DIALOG / *TP

 ,RUN-PRIORITY = *PARAMETERS(...)

     *PARAMETERS(...)

         |    DEFAULT = <integer 30..255>

         |    ,MAXIMUM = *NO / <integer 30..255>
```

```
,NO-CPU-LIMIT = *NO / *YES
,CPU-LIMIT = *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *NO-LIMIT / <integer 1..32767>
        |   ,MAXIMUM = *NO / <integer 1..32767>
,SYSLST-LIMIT = *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *NO-LIMIT / <integer 0..999999>
        |   ,MAXIMUM = *NO / *NO-LIMIT / <integer 0..999999>
,START = *NO / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *SOON / *WITHIN(...)
        |       *WITHIN(...)
        |           |   HOURS = 0 / <integer 0..23>
        |           |   ,MINUTES = 00 / <integer 0..59>
        |   ,ALLOWED = list-poss(7): *AT-STREAM-STARTUP / *AT / *EARLIEST / *SOON /
        |                   *LATEST / *WITHIN / *IMMEDIATELY
,REPEAT-JOB = *NO / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *NO / *AT-STREAM-STARTUP / *WEEKLY / *DAILY / *PERIOD(...)
        |       *PERIOD(...)
        |           |   HOURS = 0 / <integer 0..23>
        |           |   ,MINUTES = 00 / <integer 0..59>
        |   ,ALLOWED = list-poss(5): *NO / *AT-STREAM-STARTUP / *DAILY / *WEEKLY /
        |                   *PERIOD
,JOB-PARAMETER = *NO / <c-string 0..127>
```

Operanden

NAME = <name 1..8>
Ist der Name der neuen Auftragsklassen-Definition, die in die SJMSFILE geschrieben werden soll. Er darf aus 1 bis 8 alphanumerischen Zeichen bestehen. Das erste Zeichen muss ein Buchstabe A bis Z oder das Zeichen @ oder # sein.

> ¡ Die Anweisung wird abgewiesen, wenn bereits eine Auftragsklasse mit dem gleichen Namen existiert.

STREAM = *DEFAULT-STREAM
Standardstream, der mit der Anweisung DEFINE-JOB-STREAM festgelegte wurde.

> ¡ Der in der SJMSFILE vereinbarte Standardstream muss nicht mit dem des Systems identisch sein.

STREAM = <name 1..8>
Name des Streams, unter dem der Jobscheduler läuft, dem die Auftragsklasse zugeordnet wird. Der Name darf nicht $SYSJS sein. Der Name des Streams muss bereits vorher schon mit der DEFINE-JOB-STREAM-Anweisung definiert worden sein.

CLASS-LIMIT = <integer 0..4095>
Maximale Anzahl von Aufträgen, die in der Klasse gestartet werden sollen.

> ¡ Es empfiehlt sich, den Wert 0 nur anzugeben, wenn man den Start von Aufträgen nach dem Systemstart verhindern will. Andernfalls könnte der Jobscheduler behindert werden. Der angegebene Wert ist eine absolute Grenze, die aber von Expressaufträgen überschritten werden kann.

CLASS-WEIGHT = <integer 1..9>
Bestimmt die Startdringlichkeit der Klasse relativ zu anderen Klassen, deren Aufträge darauf warten gestartet zu werden.
1 ist die niedrigste, 9 die höchste Dringlichkeit.

CLASS-OPTIMUM =
Anzahl von Aufträgen, die idealerweise in der Auftragsklasse laufen sollten, um im System ein bestimmtes Verhältnis unter den Aufträgen zu erreichen.
CLASS-OPTIMUM hat Einfluss auf die Reihenfolge, in der der Klassenscheduler Auftragsklassen auswählt, um die Aufträge daraus zu starten.

CLASS-OPTIMUM = 0 / <integer 0..4095>
Anzahl der Aufträge. Es kann der Wert 0 bis zum Wert, der bei dem Operanden CLASS-LIMIT definiert ist, angegeben werden: 0 <= CLASS-OPTIMUM <= CLASS-LIMIT <= 4095.

JOB-PRIORITY =
Gibt die Priorität des Jobscheduling für Batch-Aufträge an, die die Wichtigkeit des Auftrags relativ zu anderen Aufträgen der Klasse bestimmt.

JOB-PRIORITY = *NO
Ist nur eine formale Angabe ohne Bedeutung, die aber gemacht werden muss, wenn beim Operanden JOB-TYPE=DIALOG angegeben wurde.

JOB-PRIORITY = *PARAMETERS(...)

DEFAULT = <integer 1..9>
Für die Auftragsklasse voreingestellte Priorität.
1 ist die höchste, 9 die niedrigste Priorität.

Hat der Benutzer für seinen Auftrag keine Priorität angegeben, kommt DEFAULT=<integer...> zur Anwendung. Hat der Benutzer eine Priorität nicht höher als MAXIMUM=<integer...> angegeben, so gilt die vom Benutzer bei /ENTER-JOB angegebene.
Es darf für die DEFINE-JOB-CLASS-Anweisung bei DEFAULT keine höhere Priorität als bei MAXIMUM angegeben werden, sonst wird die Anweisung mit Syntaxfehler abgewiesen.

MAXIMUM = NO / <integer 1..9>
Maximal zugelassene Priorität der Auftragsklasse.
Bei MAXIMUM=NO bekommt der Auftrag die Priorität, die mit DEFAULT angegeben ist, ganz gleich, welche Priorität der Benutzer bei /ENTER-JOB angegeben hat.

JOB-TYPE =
Bezeichnet die Art der Auftragsklasse.

JOB-TYPE = *BATCH
Gibt an, dass die Auftragsklasse eine Batch-Auftragsklasse sein soll. Das bedeutet, dass ein Auftrag, der zu dieser Klasse gehört, nicht durch /SET-LOGON-PARAMETERS im Dialog eingegeben werden darf.

JOB-TYPE = *DIALOG
Gibt an, dass die Auftragsklasse eine Dialog-Auftragsklasse sein soll. Ein zu dieser Auftragsklasse gehörender Auftrag darf nicht durch /ENTER-JOB eingegeben werden.

> Bei einer POSIX-Standardklasse (siehe die Anweisung SET-POSIX-JOB-CLASS-DEFAULT) wird der JOB-TYPE nur dann ausgewertet, wenn der Benutzer versucht, in dieser Klasse normale Batch- oder Dialogaufträge zu starten. Die Angabe wirkt sich auf POSIX-FORK-Tasks nicht aus.

TP-ALLOWED = *NO / *YES(...)
Gibt an, ob das Taskattribut TP in der Auftragsklasse erlaubt ist.

TP-ALLOWED = *NO
Bedeutet, dass das Taskattribut TP in der Auftragsklasse nicht erlaubt ist. Aufträge dieser Klasse dürfen nicht unter diesem Taskattribut gestartet werden bzw. der Wechsel zu diesem Taskattribut durch den Makro TINF ist verboten, es sei denn, er ist laut Eintrag im Benutzerkatalog zugelassen.

TP-ALLOWED = *YES(...)

CATEGORY = TP / <name 1..7>
Für das Taskattribut TP kann ein Kategoriename vergeben werden.
Es kann der Standardkategoriename (TP) sein oder ein vom Benutzer frei definierter Name. Der Standardkategoriename SYS ist verboten.
Zusätzlich zu den vier vordefinierten dürfen bis zu zwölf weitere Kategorienamen definiert werden.
Bei Änderungen in der Datenbasis wird der Standardkategoriename TP verwendet, falls die angegebene Kategorie im System nicht bekannt ist.
(Siehe „Hinweise zu allen 3 Taskattributen“.)

DIALOG-ALLOWED = *NO / *YES(...)
Gibt an, ob das Taskattribut DIALOG in der Auftragsklasse erlaubt ist.

DIALOG-ALLOWED = *NO
Bedeutet, dass das Taskattribut DIALOG in der Auftragsklasse nicht erlaubt ist. Aufträge dieser Klasse dürfen nicht unter diesem Taskattribut gestartet werden bzw. der Wechsel zu diesem Taskattribut durch den Makro TINF ist verboten, es sei denn, er ist laut Eintrag im Benutzerkatalog zugelassen.

DIALOG-ALLOWED = *YES(...)

CATEGORY = DIALOG / <name 1..7>
Für das Taskattribut DIALOG kann ein Kategoriename vergeben werden.
Es kann der Standardkategoriename (DIALOG) sein oder ein vom Benutzer frei definierter Name. Der Standardkategoriename SYS ist verboten.
Zusätzlich zu den vier vordefinierten dürfen bis zu zwölf weitere Kategorienamen definiert werden.
Bei Änderungen in der Datenbasis wird der Standardkategoriename DIALOG verwendet, falls die angegebene Kategorie im System nicht bekannt ist.
(Siehe „Hinweise zu allen 3 Taskattributen“.)

BATCH-ALLOWED = *NO / *YES(...)
Gibt an, ob das Taskattribut BATCH in der Auftragsklasse erlaubt ist.

BATCH-ALLOWED = *NO
Bedeutet, dass das Taskattribut BATCH in der Auftragsklasse nicht erlaubt ist. Aufträge dieser Klasse dürfen nicht unter diesem Taskattribut gestartet werden bzw. der Wechsel zu diesem Taskattribut durch den Makro TINF ist verboten, es sei denn, er ist laut Eintrag im Benutzerkatalog zugelassen.

BATCH-ALLOWED = *YES(...)

CATEGORY = BATCH / <name 1..7>
Für das Taskattribut BATCH kann ein Kategoriename vergeben werden.
Es kann der Standardkategoriename (BATCH) sein oder ein vom Benutzer frei definierter Name. Der Standardkategoriename SYS ist verboten.
Zusätzlich zu den vier vordefinierten dürfen bis zu zwölf weitere Kategorienamen definiert werden.
Bei Änderungen in der Datenbasis wird der Standardkategoriename BATCH verwendet, falls die angegebene Kategorie im System nicht bekannt ist.

Hinweise zu allen 3 Taskattributen:

Die Abhängigkeiten zum Wert des Operanden START-ATTR sind zu beachten.

Ein Kategoriename darf nicht zwei verschiedenen Taskattributen zugeordnet werden, z.B. ist folgende Angabe zweideutig und daher nicht erlaubt: BATCH-ALLOWED=*YES(CATEGORY=HUGO) und DIALOG-ALLOWED=*YES(CATEGORY=HUGO)

START-ATTRIBUTE = *BATCH / *DIALOG / *TP
Legt das Taskattribut für den Auftrag fest. Gleichzeitig muss das entsprechende Taskattribut im Operanden TP-, DIALOG- oder BATCH-ALLOWED angegeben werden, z.B. START-ATTRIBUTE=*TP und TP-ALLOWED=*YES (...).

RUN-PRIORITY =
Gibt die Taskpriorität an, mit der ein Auftrag gestartet wird.

RUN-PRIORITY = *PARAMETERS(...)

DEFAULT = <integer 30..255>
Für die Auftragsklasse voreingestellter Wert.
Bei DEFAULT darf keine höhere Priorität als bei MAXIMUM angegeben werden.

MAXIMUM = *NO / <integer 30..255>
Gibt die maximale zugelassene Priorität für die Auftragsklasse an.
Bedeutet, dass ein Auftrag die Priorität bekommt, die der Benutzer angegeben hat, wenn sie nicht die maximal zugelassene Priorität überschreitet. Ist allerdings im Benutzerkatalog für den Benutzer eine höhere Priorität zugelassen als im Operanden MAXIMUM angegeben, dann darf er den Wert von MAXIMUM überschreiten.

MAXIMUM=*NO bedeutet, dass keine maximale Taskpriorität definiert ist. Ein Auftrag bekommt die Priorität, die der Benutzer angegeben hat, falls diese nicht höher ist als die für ihn laut Eintrag im Benutzerkatalog geltende.

NO-CPU-LIMIT = *NO / *YES
Gibt an, ob Aufträge in dieser Klasse ohne Zeitbeschränkung laufen dürfen (NTL).
Die Angabe NO bedeutet, dass Aufträge in dieser Klasse nicht ohne Zeitbeschränkung laufen dürfen (siehe /SET-LOGON-PARAMETERS). Ist für einen Benutzer im Benutzerkatalog der Eintrag NO-CPU-LIMIT=*YES angegeben, so kann er Aufträge ohne Zeitbeschränkung auch dann laufen lassen, wenn in der Auftragsklasse NO-CPU-LIMIT=*NO gilt.

CPU-LIMIT =
CPU-Zeit, die ein Auftrag dieser Klasse verbrauchen darf.

CPU-LIMIT = *PARAMETERS(...)

DEFAULT = *NO-LIMIT
In dieser Auftragsklasse ist voreingestellt, dass die Aufträge ohne Zeitbeschränkung laufen. Der Wert ist nur erlaubt, wenn NO-CPU-LIMIT=*YES und MAXIMUM=32767 ist.

DEFAULT = <integer 1..32767>
Für die Auftragsklasse voreingestellter Wert.

MAXIMUM = <integer 1..32767>
Ist die maximal explizit anforderbare CPU-Zeit für einen Auftrag der Auftragklasse.Dieser Wert darf nicht überschritten werden, wenn beim Erzeugen eines Auftrags durch ENTER-JOB, ENTER-PROCEDURE oder SET-LOGON-PARAMETERS für CPU-LIMIT ein Zahlenwert spezifiziert wird. Die maximale CPU-Zeit, die insgesamt für eine Abrechnungsnummer verbraucht werden darf, hängt vom CPU-Eintrag im Benutzerkatalog ab.
Der durch DEFAULT angegebene Wert darf den bei MAXIMUM angegebenen Wert nicht überschreiten. Für DEFAULT=*NO-LIMIT muss für MAXIMUM der
höchstmögliche Wert 32767 vergeben und NO-CPU-LIMIT=*YES angegeben sein. Ist eine dieser Bedingungen verletzt, so wird die Anweisung mit einem Syntaxfehler abgewiesen.

MAXIMUM = *NO
Der Auftrag darf die CPU-Zeit verbrauchen, die mit DEFAULT angegeben ist, unabhängig davon, was der Benutzer an CPU-Zeit gefordert hat.

SYSLST-LIMIT =
Bestimmt die Anzahl der Zeilen für einen Auftrag bei einer Ausgabe über SYSLST.

SYSLST-LIMIT = *PARAMETERS(...)

DEFAULT = *NO-LIMIT / <integer 0..999999>
Für die Auftragsklasse voreingestellte Anzahl Zeilen. *NO-LIMIT bedeutet, die Anzahl ist unbeschränkt. Der bei DEFAULT angegebene Wert darf den bei MAXIMUM angegebenen Wert nicht überschreiten, sonst wird die Anweisung DEFINE-JOB-CLASS mit Syntaxfehler abgewiesen.

MAXIMUM =
Maximal zugelassene Zeilenzahl für die Auftragsklasse.

MAXIMUM = *NO
Der Auftrag erhält die Anzahl der zugelassenen Zeilen, die mit DEFAULT angegeben ist, ganz gleich, welche Anzahl der Benutzer angefordert hat.

MAXIMUM = *NO-LIMIT
Die zugelassene Anzahl Zeilen für einen Auftrag dieser Klasse ist unbeschränkt. Die vom Benutzer angegebene Anzahl Zeilen ist in jedem Fall gültig.

MAXIMUM = <integer 0..999999>
Bedeutet, dass die Anzahl Zeilen für den Auftrag gültig ist, die der Benutzer angegeben hat, wenn der von ihm angegebene Wert nicht den Wert von MAXIMUM überschreitet.

START =
Ordnet den Startanforderungen für Aufträge entsprechende Startmöglichkeiten zu.

START = *NO
Ist nur eine formale Angabe ohne Bedeutung, die aber bei gleichzeitiger Angabe des Operanden JOB-TYPE=DIALOG erforderlich ist.

START = *PARAMETERS(...)

DEFAULT =
Ist Standardwert, der angenommen wird, falls ein Benutzer bei /ENTER-JOB keine Anforderung für eine bestimmte Startart gemacht hat. Der unter DEF definierte Wert muss nicht unter ALLOWED (siehe unten) aufgelistet werden.

DEFAULT = *SOON
Der Auftrag soll so schnell wie möglich gestartet werden. Wenn mehrere Aufträge SOON angefordert haben, entscheidet die Auftragspriorität.

DEFAULT = *WITHIN(...)
Der Auftrag soll innerhalb der in Stunden und Minuten angegebenen Zeit gestartet werden.

HOURS = 0 / <integer 0..23>
Ist ein Wert von 0 bis 23 Uhr.

MINUTES = 00 / <integer 0..59>
Ist ein Wert von 0 bis 59 Minuten.

ALLOWED =
Gibt die Werte an, die der Benutzer im Operanden START= bei /ENTER-JOB für die entsprechende Auftragsklasse angeben darf.

ALLOWED = *IMMEDIATELY
Ein Auftrag in dieser Auftragsklasse darf sofort gestartet werden, auch wenn er andere Aufträge mit höherer Priorität, die zu dieser Zeit gestartet werden sollten, zurückdrängt.

ALLOWED = *SOON
Hat die gleiche Bedeutung wie DEFAULT=*SOON, siehe oben.

ALLOWED = *AT
Der Auftrag darf am angegebenen Datum und genau zur angegebenen Zeit (Stunde, Minuten) gestartet werden, wenn es möglich ist.

ALLOWED = *LATEST
Ein Auftrag darf spätestens zum angegebenen Zeitpunkt (Datum, Uhrzeit) gestartet werden.

ALLOWED = *EARLIEST
Ein Auftrag darf frühestens zum angegebenen Zeitpunkt (Datum, Uhrzeit) gestartet werden.

ALLOWED = *WITHIN
Ein Auftrag darf innerhalb der angegebenen Zeit gestartet werden.

ALLOWED = *AT-STREAM-STARTUP
Ein Auftrag darf zum Zeitpunkt gestartet werden, wenn der Jobscheduler gestartet wird.

> ¡ Fehlt eine Angabe unter ALLOWED, ist sie verboten, es sei denn, es handelt sich um die Angabe *SOON oder *WITHIN, die unter DEFAULT als Standardwert definiert ist.
> Eine Ausnahme bildet auch ALLOWED=*IMMEDIATELY: Hat der Benutzer bei `/SET-LOGON-PARAMETERS`, `/ENTER-JOB` oder `/MODIFY-JOB` den Wert START=*IMMEDIATELY angegeben und in seinem Eintrag im Benutzerkatalog ist START-IMMEDIATE=*YES festgelegt, dann wird sein Auftrag auch dann sofort gestartet, wenn in der Auftragsklasse ALLOWED =*IMMEDIATELY nicht angegeben ist.

REPEAT-JOB =
Regelt die Häufigkeit von Auftragswiederholungen nach bestimmten Zeitabschnitten. Wird bei JOB-TYPE=*DIALOG ignoriert.

REPEAT-JOB = *NO
Ist nur eine formale Angabe ohne Bedeutung, die aber bei gleichzeitiger Angabe des Operanden JOB-TYPE=DIALOG erforderlich ist.

REPEAT-JOB = *PARAMETERS(...)

DEFAULT =
Ist der Standardwert, der angenommen wird, falls der Benutzer bei `/ENTER-JOB` bzw. `/SET-LOGON-PARAMETERS` keine Angaben für die Häufigkeit von Auftragswiederholungen gemacht hat, d.h. den Operanden REPEAT-JOB bzw. REPEAT weggelassen oder REPEAT-JOB=STD bzw. REPEAT=STD angegeben hat.

DEFAULT = *NO
Bedeutet, dass der Auftrag nicht wiederholt wird.

DEFAULT = *AT-STREAM-STARTUP
Aufträge, die zu dieser Klasse gehören, werden nach jedem Start des Jobschedulers wiederholt zum Laufen gebracht, wenn der Benutzer es bei `/ENTER-JOB` bzw. `/SET-LOGON-PARAMETERS` gefordert hat.

DEFAULT = *WEEKLY
Aufträge dieser Klasse werden wöchentlich gestartet.
Die genaue Startzeit hängt von den Angaben im Operanden START bei `/ENTER-JOB` ab.

DEFAULT = *DAILY
Aufträge dieser Klasse werden täglich gestartet.
Die genaue Startzeit hängt von den Angaben im Operanden START bei /ENTER-JOB ab.

DEFAULT = *PERIOD(...)
Aufträge werden jedes Mal nach dem angegebenen Zeitintervall wiederholt.

HOURS = 0 / <integer 0..23>
Als Zeitintervall kann ein Wert von 0 bis 23 Stunden angegeben werden.

MINUTES = 00 / <integer 0..59>
Es können 0 bis 59 Minuten angegeben werden.
Insgesamt muss das Zeitintervall größer als 0 sein. Die genaue Startzeit hängt von den Angaben im Operanden START bei /ENTER-JOB ab.

ALLOWED =
Gibt die Werte an, die der Benutzer im Operanden REPEAT-JOB bzw. REPEAT bei /ENTER-JOB bzw. /SET-LOGON-PARAMETERS angeben darf.

ALLOWED = *NO
Die Wiederholung von Aufträgen in dieser Klasse ist nicht möglich, es sei denn bei DEFAULT ist eine andere Angabe als NO gemacht worden.

ALLOWED =*AT-STREAM-STARTUP
Aufträge können, wenn gewünscht, nach jedem Start des Jobschedulers wiederholt zum Laufen gebracht werden.

ALLOWED = *DAILY
Aufträge können täglich wiederholt werden. Die genaue Startzeit hängt von der Angabe im START-Operanden bei /ENTER-JOB ab.

ALLOWED = *WEEKLY
Aufträge können wöchentlich wiederholt werden. Die genaue Startzeit hängt von der Angabe im START-Operanden bei /ENTER-JOB ab.

ALLOWED = *PERIOD
Aufträge können jedes Mal nach dem angegebenen Zeitintervall wiederholt werden.

JOB-PARAMETER =
Zusätzliche Auftragsklassenattribute, die vom System Exit 32 ausgewertet werden.

JOB-PARAMETER = *NO
Es werden keine zusätzlichen Klassenattribute definiert.

JOB-PARAMETER = <c-string 1..127>
Dieser Operanden definiert zusätzliche Klassenattribute in freier Syntax. Damit können system-spezifische Informationen in jeder Auftragsklassendefinition gespeichert werden.

Der Operand wird weder vom System noch vom vordefinierten Jobscheduler ausgewertet. Soll er ausgewertet werden, muss die Systemverwaltung eine der beiden folgenden Aktionen ausführen:

1. Erstellen einer Exit-Routine, die die Angaben des Benutzers im JOB-PARAMETER bei /SET-LOGON-PARAMETERS, /ENTER-JOB oder /MODIFY-JOB mit den Angaben im JOB-PARAMETER vergleicht und deren Gültigkeit bestätigt.

2. Den Scheduling-Algorithmus des für die Jobklasse zuständigen Schedulers entsprechend festlegen.

Die Exit-Routine wird bei der Bearbeitung der oben genannten Kommandos des Benutzers aufgerufen.

Hinweise

- Die Operanden CLASS-LIMIT und CLASS-WEIGHT werden vom Klassenscheduler des Betriebssystems, der unabhängig von den Jobschedulern ist, ausgewertet, um die Systemlast an Aufträgen zu steuern (siehe Handbuch „Systembetreuung“ [5]).
- Die Bedeutung der Operanden JOB-PRIORITY, START und JOB-PARAMETER hängt vom Jobscheduling-Algorithmus ab, der vom Stream benutzt wird, der im Operanden STREAM definiert ist.
- Die Anzahl der Auftragsklassen ist unbeschränkt zugelassen.

6.3.2.3 DEFINE-JOB-STREAM - Streamdefinitionen in die Datei SJMSFILE schreiben

Mit dieser Anweisung wird eine neue Streamdefinition in die Datei SJMSFILE bzw. JMS-Datenbasis geschrieben.

Format

```
DEFINE-JOB-STREAM

 NAME = <name 1..8>

 ,FILE = <filename 1..54> / *LIBRARY-ELEMENT(...)

       *LIBRARY-ELEMENT(...)

            |      LIBRARY = <filename 1..41>

            |      ,ELEMENT = <name 1..8>

 ,RUN-PRIORITY = 65 / <integer 30..255>

 ,DEFAULT = *NO / *YES

 ,START = *AT-LOAD / *BY-OPERATOR / *AT(...) / *EARLIEST(...)

       *AT(...)

            |      TIME = <time 1..8>

       *EARLIEST(...)

            |      TIME = <time 1..8>

 ,STOP = *AT-SHUTDOWN / *BY-OPERATOR / *AT(...) / *AFTER(...)

       *AT(...)

            |      TIME = 00:00 / <time 1..8>

       *AFTER(...)

            |      HOURS = 00 / <integer 0..23>

            |      ,MINUTES = 00 / <integer 0..59>

 ,STREAM-PARAMETER = *NO / <c-string 1..127>
```

Operanden

NAME = <name 1..8>
Ist der Name der Streamdefinition, die in die Datei SJMSFILE geschrieben werden soll. Es kann eine Folge von 1 bis 8 alphanumerischen Zeichen angegeben werden, die mit den Zeichen A-Z, @ oder # beginnen muss.

FILE = <filename 1..54>
Ist der Name der ENTER-Datei, die den Auftrag enthält, der während des Streamstarts eingeleitet wird und den Jobscheduler aktiviert.
Der Jobscheduler kann nur Batch-Aufträge bearbeiten.

FILE = *LIBRARY-ELEMENT(...)

LIBRARY = <filename 1..41>
Dateiname der Bibliothek

ELEMENT = <name 1..8>
Das Bibliothekselement, das die oben genannte ENTER-Datei enthält.

RUN-PRIORITY = 65 / <integer 30..255>
Gibt die Anfangspriorität an, die der Streamtask zugeordnet werden soll, unter der der Jobscheduler läuft.

DEFAULT = *NO / *YES
Gibt an, ob der betreffende Stream der Standardstream für das System sein soll. YES bedeutet, der Stream soll der Standardstream für Auftragsklassen sein, die angegeben haben, dass sie den Standardstream benutzen wollen.

START =
Gibt an, wann der Stream gestartet werden soll.

START = *AT-LOAD
Der Stream soll automatisch beim Laden des Systems gestartet werden.

START = *BY-OPERATOR
Der Stream muss von der Systembetreuung mit `/START-JOB-STREAM` gestartet werden.

START = *AT(...)
Bei jedem Systemlauf soll der Stream automatisch zu der angegebenen Zeit gestartet werden. Wenn ein Systemlauf nach der angegebenen Zeit gestartet wird, kann der Stream nur innerhalb der nächsten 30 Minuten nach der angegebenen Zeit gestartet werden. Ein Start zu einem späteren Zeitpunkt ist im aktuellen Systemlauf nicht möglich.

> ¡ Falls der Jobstream in die JMS-Datenbasis eingebracht wird, gilt: Wenn der START-Zeitpunkt um nicht mehr als 5 Minuten überschritten und der STOP-Zeitpunkt noch nicht erreicht ist, so wird der Stream gestartet.

START = *EARLIEST(...)
Bei jedem Systemlauf soll der Stream frühestens automatisch zu der angegebenen Zeit gestartet werden. Ist das System zum angegebenen Zeitpunkt nicht aktiv, wird der Start bis zu einem evtl. angegebenen STOP-Zeitpunkt oder bis 24:00 Uhr nachgeholt.

> ¡ Falls der Jobstream in die JMS-Datenbasis eingebracht wird, gilt: Wenn der START-Zeitpunkt überschritten und der STOP-Zeitpunkt noch nicht erreicht ist, wird der Stream gestartet.

TIME = 00:00 / <time 1..8>
Ist die Uhrzeit, Angabe nur in hh:mm, Stunden und Minuten, Sekunden werden ignoriert.

STOP =
Gibt an, wann der Stream gestoppt werden soll.

STOP = *AT-SHUTDOWN
Der Stream soll beim Herunterfahren des Systems gestoppt werden.

STOP = *BY-OPERATOR
Der Stream muss von der Systembetreuung mit `/STOP-JOB-STREAM` gestoppt werden.

STOP = *AT(...)
Der Stream soll automatisch zu der angegebenen Zeit gestoppt werden (hh:mm siehe bei START=*AT...).

TIME = 00:00 / <time 1..8>
Ist die Uhrzeit, Angabe nur in hh:mm, Stunden und Minuten, Sekunden werden ignoriert.

STOP = *AFTER(...)
Der Stream soll nach seinem Start nach Ablauf der angegebenen Zeit gestoppt werden.

HOURS = 00 / <integer 0..23>
Für HOURS= können 0 bis 23 Stunden angegeben werden.

MINUTES = 00 / <integer 0..59>
Es können 0 bis 59 Minuten angegeben werden.

> Falls das System aus irgendeinem Grund in der Zeit zwischen dem Systemstart und der Zeit, die für das Stoppen des Streams angegeben war, heruntergefahren wird, so wird, falls das System noch vor der Stoppzeit des Streams erneut hochgefahren wird, auch der Stream automatisch neu gestartet.

STREAM-PARAMETER =
Dieser Operanden definiert für den Jobscheduler spezielle Scheduling-Parameter in freier Syntax. Der Inhalt dieses Operanden wird nicht vom System ausgewertet.
Der betroffene Jobscheduler muss aber sowohl die Syntax als auch die Bedeutung von STREAM-PARAMETER kennen, um auf die dort definierten Scheduling-Parameter einzugehen.
Diese Informationen besorgt sich der Jobscheduler über die Jobscheduler-Schnittstelle. Die Jobscheduler-Schnittstelle stellt über eine TU-Schnittstelle Funktionen zur Verfügung, die der Jobscheduler zur Durchführung seiner Aufgaben benötigt (näheres siehe Handbuch „Systembetreuung“ [5]).

STREAM-PARAMETER = <c-string 1..127>
Ist die Folge von zusätzlichen Parametern, die vom Jobscheduler ausgewertet werden. Beim Hochfahren des Systems werden die in den Parametern enthaltenen Informationen in interne Tabellen übernommen, aus denen der Jobscheduler die Informationen abfragt.

Der Standard-Jobscheduler kennt folgende Parameter:

```
S-PAR =  'JOB-PRIORITY = YES / NO
         ,CPU-TIME = NO / YES
         ,WAIT-TIME = NO / YES
         ,JOB-QUOTA = 1 / <integer 1..255>
         ,LOGGING = YES / NO
         ,CATID-LIST = (catid1,...)
         ,CAT-TIME = min
         ,MSG-OUTPUT = YES / STD'
```

STREAM-PARAMETER = *NO
Bedeutet, es werden keine speziellen Parameter für den Jobscheduler definiert.

Hinweise

- Die Anweisung wird zurückgewiesen, wenn ein Stream mit dem angegebenen Namen in der SJMSFILE bereits enthalten ist.
- Es darf nur ein Standardstream im System existieren. Der Versuch, mehrere Standardstreams zu definieren, wird abgewiesen.
- Es dürfen maximal 16 Streams definiert werden.

6.3.2.4 DELETE-JOB-CLASS - Klassendefinitionen löschen

Eine bestehende Klassendefinition kann mit dieser Anweisung aus der SJMSFILE oder in der JMS-Datenbasis gelöscht werden.

Wird die Definition in der Datei gelöscht, gehen die Aufträge dieser Klasse, die beim Start des nächsten Systemlaufs im Auftragspool sind, dabei verloren. Soll die Definition im System gelöscht werden, erhält sie den Zustand 'IN-DELETE'. Erst wenn dieser Klasse kein Auftrag mehr zugeordnet ist, wird ihre Definition aus der Datenbasis entfernt.
Für Jobklassen im Zustand 'IN-DELETE' werden keine neuen Aufträge mehr akzeptiert. Es ist nicht erlaubt, eine Standardklasse zu löschen. Wenn eine Standardklasse gelöscht werden soll, müssen zuerst alle Zugriffsrechte auf sie zurückgesetzt werden.

Format

DELETE-JOB-CLASS
NAME = <name 1..8>

Operanden

NAME = <name 1..8>
Ist der Name bestehend aus 1 bis 8 Zeichen einer bestehenden Klassendefinition, die gelöscht werden soll.

6.3.2.5 DELETE-JOB-STREAM - Streamdefinitionen löschen

Mit dieser Anweisung kann eine Streamdefinition aus der SJMSFILE und der JMS-Datenbasis gelöscht werden.

Format

DELETE-JOB-STREAM
NAME = <name 1..8>

Operanden

NAME = <name 1..8>
Name des Jobstreams, der gelöscht werden soll.

> Wenn dem Stream Auftragsklassen zugeordnet sind, wird die Löschanforderung abgewiesen.
> Wenn der Stream aktiv ist und aus der JMS-Datenbasis gelöscht werden soll, wird die Löschanforderung abgewiesen.

6.3.2.6 END - Eingabe von Anweisungen beenden

Diese Anweisung beendet die Eingabe von Anweisungen des Programms JMU.

Format

END

Diese Anweisung besitzt keine Operanden.

6.3.2.7 GRANT-JOB-CLASS-ACCESS - Zugriff von Benutzerkennungen zu einer Auftragsklasse regeln

Mit dieser Anweisung wird der Zugriff von Benutzerkennungen zu einer Auftragsklasse geregelt.

Wenn eine Auftragsklasse mit der Anweisung SET-JOB-CLASS-DEFAULT als Standardklasse für einen Benutzer definiert wird, dann ist es nicht nötig, zusätzlich mit GRANT-JOB-CLASS-ACCESS den Zugriff zu regeln.

Wenn während des Systemlaufs eine neue Benutzerkennung in den Benutzerkatalog eingetragen wird, bekommt sie sofort Zugriff zu allen gemeinschaftlichen Klassen.

Format

GRANT-JOB-CLASS-ACCESS
NAME = <name 1..8> ,ACTION = *ADD / *REMOVE ,USER = *ALL / list-poss(255): <name 1..8>

Operanden

NAME = <name 1..8>
Name einer Auftragsklasse, zu der eine Benutzerkennung Zugriff erhalten oder zu der der Zugriff gesperrt werden soll.

ACTION =
Zugriff auf eine Auftragsklasse zulassen oder sperren.

ACTION = *ADD
Die unter USER angegebene Benutzerkennung darf auf die Auftragsklasse zugreifen.

ACTION = *REMOVE
Der unter USER angegebenen Benutzerkennung wird der Zugriff auf die bei NAME angegebene Auftragsklasse gesperrt.

USER =
Zugriff auf eine Auftragsklasse regeln.

USER = *ALL
Der Zugriff auf bei NAME angegebene Auftragsklasse soll für alle Benutzerkennungen geregelt werden.

USER = list-poss(255)
Der Zugriff wird für die in dieser Liste aufgeführten Benutzerkennungen geregelt.

USER = <name 1..8>
Der Zugriff wird für diese eine Benutzerkennung, deren Name aus 1 bis 8 Zeichen besteht, geregelt.
Eine Überprüfung, ob die angegebene Benutzerkennung auch in den Benutzerkatalog eingetragen ist, erfolgt nur bei Änderungen im laufenden System. In die SJMSFILE werden alle Kennungen eingetragen.

6.3.2.8 MODIFY-JOB-CLASS - Auftragsklassen-Definitionen ändern

Mit dieser Anweisung kann eine bestehende Auftragsklassen-Definition in der SJMSFILE oder in der JMS-Datenbasis geändert werden.

Format

```
MODIFY-JOB-CLASS

 NAME = <name 1..8>

 ,STREAM = *UNCHANGED / *DEFAULT-STREAM / <name 1..8>

 ,CLASS-LIMIT = *UNCHANGED / <integer 0..4095>

 ,CLASS-WEIGHT = *UNCHANGED / <integer 1..9>

 ,CLASS-OPTIMUM = *UNCHANGED / <integer 0..4095>

 ,JOB-PRIORITY = *UNCHANGED / *PARAMETERS(...)

     *PARAMETERS(...)

         |    DEFAULT = <integer 1..9>

         |    ,MAXIMUM = *UNCHANGED / *NO / <integer 1..9>

 ,JOB-TYPE = *UNCHANGED / *BATCH / *DIALOG

 ,TP-ALLOWED = *UNCHANGED / *NO / *YES(...)

     *YES(...)

         |    CATEGORY = TP / <name 1..7>

 ,DIALOG-ALLOWED = *UNCHANGED / *NO / *YES(...)

     *YES(...)

         |    CATEGORY = DIALOG / <name 1..7>

 ,BATCH-ALLOWED = *UNCHANGED / *NO / *YES(...)

     *YES(...)

         |    CATEGORY = BATCH / <name 1..7>

 ,START-ATTRIBUTE = *UNCHANGED / *BATCH / *DIALOG / *TP

 ,RUN-PRIORITY = *UNCHANGED / *PARAMETERS(...)

     *PARAMETERS(...)

         |    DEFAULT = <integer 30..255>

         |    ,MAXIMUM = *UNCHANGED / *NO / <integer 30..255>

 ,NO-CPU-LIMIT = *UNCHANGED / *NO / *YES
```

```
,CPU-LIMIT = *UNCHANGED / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = <integer 1..32767>
        |   ,MAXIMUM = *UNCHANGED / *NO / <integer 1..32767>
,SYSLST-LIMIT = *UNCHANGED / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *NO-LIMIT / <integer 0..999999>
        |   ,MAXIMUM = *UNCHANGED / *NO / *NO-LIMIT / <integer 0..999999>
,START = *UNCHANGED / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *SOON / *WITHIN(...)
        |       *WITHIN(...)
        |           |   HOURS = 0 / <integer 0..23>
        |           |   ,MINUTES = 00 / <integer 0..59>
        |   ,ALLOWED = list-poss(7): *IMMEDIATELY / *AT / *EARLIEST / *LATEST /
        |                   *AT-STREAM-STARTUP / *WITHIN / *SOON
,REPEAT-JOB = *UNCHANGED / *PARAMETERS(...)
    *PARAMETERS(...)
        |   DEFAULT = *NO / *AT-STREAM-STARTUP / *WEEKLY / *DAILY / *PERIOD(...)
        |       *PERIOD(...)
        |           |   HOURS = 0 / <integer 0..23>
        |           |   ,MINUTES = 00 / <integer 0..59>
        |   ,ALLOWED = list-poss(5): *NO / *AT-STREAM-STARTUP / *DAILY / *WEEKLY /
        |                   *PERIOD
,JOB-PARAMETER = *UNCHANGED / *NO / <c-string 0..127>
```

Beschreibung der Operanden siehe bei der Anweisung DEFINE-JOB-CLASS.

> **i** Änderungen in der Datenbasis werden für die Operanden STREAM, JOB-TYPE und START-ATTRIBUTE ignoriert.
> Änderung des Operanden JOB-TYPE ist für Standardjobklassen nicht zulässig.

6.3.2.9 MODIFY-JOB-STREAM - Streamdefinitionen ändern

Mit dieser Anweisung kann eine existierende Streamdefinition geändert werden.

Format

MODIFY-JOB-STREAM

```
NAME = <name 1..8>

,FILE = *UNCHANGED / <filename 1..54> / *LIBRARY-ELEMENT(...)

        *LIBRARY-ELEMENT(...)

                |       LIBRARY = <filename 1..41>

                |       ,ELEMENT = <name 1..8>

,RUN-PRIORITY = *UNCHANGED / <integer 30..255>

,DEFAULT = *UNCHANGED / *NO / *YES

,START = *UNCHANGED / *AT-LOAD / *BY-OPERATOR / *AT(...) / *EARLIEST(...)

        *AT(...)

                |       TIME = 00:00 / <time 1..8>

        *EARLIEST(...)

                |       TIME = 00:00 / <time 1..8>

,STOP = *UNCHANGED / *AT-SHUTDOWN / *BY-OPERATOR / *AT(...) / *AFTER(...)

        *AT(...)

                |       TIME = 00:00 / <time 1..8>

        *AFTER(...)

                |       HOURS = 00 / <integer 0..23>

                |       ,MINUTES = 00 / <integer 0..59>

,STREAM-PARAMETER = *UNCHANGED / *NO / <c-string 1..127>
```

Die Bedeutung der Operanden ist der Anweisung DEFINE-JOB-STREAM zu entnehmen.

Bei der Änderung des Operanden DEFAULT ist zu berücksichtigen, dass es immer nur einen Standardstream im System geben darf. Der Versuch, mehrere zu definieren, wird abgewiesen.

Wird der Operand START oder STOP direkt in der JMS-Datenbasis geändert, so wird der Jobstream entsprechend den neuen Operandenwerten ggf. gestartet oder gestoppt.

6.3.2.10 REMOVE-USER - Zugang zu nichtöffentlichen Auftragsklassen sperren

Mit dieser Anweisung kann den angegebenen Benutzerkennungen der Zugriff zu allen nichtöffentlichen Auftragsklassen gesperrt werden. Diese Anweisung ist eine Ergänzung des gleichnamigen Systemkommandos, das die Benutzereinträge im Benutzerkatalog löscht. Die Anweisung kann aber nicht dazu verwendet werden, um speziellen Benutzern den Zugang zu öffentlichen Auftragsklassen oder Systemstandard-Klassen zu sperren.

Format

REMOVE-USER
USER-IDENTIFICATION = *ALL / list-poss(8): <name 1..8>

Operanden

USER-IDENTIFICATION =
Gibt die zu löschenden Benutzerkennungen an.

USER-IDENTIFICATION = *ALL
Veranlasst das Löschen aller Zugriffslisten. Alle nichtöffentlichen Auftragsklassen sind damit gesperrt. Eventuelle kennungsspezifische Standard-Auftragsklassen sind auf die Systemvoreinstellung zurückgesetzt.

USER-IDENTIFICATION = list-poss(8): <name 1..8>
Ist (sind) die Benutzerkennung(en), der (denen) der Zugriff zu allen nichtöffentlichen Auftragsklassen gesperrt werden soll. Eventuelle Standard-Auftragsklassen dieser Kennung(en) werden zurückgesetzt. Es können bis zu 8 Kennungen angegeben werden.

6.3.2.11 SET-JOB-CLASS-DEFAULT - Standardklassen für Benutzer spezifizieren

Mit dieser Anweisung werden Standardklassen für Benutzer spezifiziert. Gleichzeitig wird den Benutzern auch der Zugriff eingeräumt.

Wenn keine gemeinschaftliche Standardklasse für einen Auftragstyp spezifiziert wurde, ist $SYSJC die Systemstandardklasse. Da nur der privilegierte Benutzer Zugriff auf
die $SYSJC hat, gibt JMU eine Warnmeldung aus.

Wenn ein neuer Benutzer in den Benutzerkatalog während des Systemlaufs eingetragen wird, bekommt er sofort Zugriff zu allen Systemstandardklassen.

Format

SET-JOB-CLASS-DEFAULT
NAME = <name 1..8> ,ACTION = *ADD / *REMOVE ,USER = *ALL / list-poss(255): <name 1..8>

Die Bedeutung der Operanden ist die gleiche wie bei der Anweisung GRANT-JOB-CLASS-ACCESS.

> ¡ Ein Ändern der Systemstandardklasse beinhaltet stets die beiden Anweisungen:
>
> - SET-JOB-CLASS-DEFAULT standardklasse.alt,*REMOVE,*ALL
> - SET-JOB-CLASS-DEFAULT standardklasse.neu,*ADD,*ALL.
>
> Im laufenden System ist diese Änderung nicht zu empfehlen, da zwischen den beiden Anweisungen $SYSJC als Systemstandardklasse eingetragen wird.

6.3.2.12 SET-MODIFICATION-MODE - Änderungsmodus einstellen

Mit dieser Anweisung kann im Dienstprogramm JMU ein Änderungsmodus eingestellt werden, der es erlaubt, folgende Änderungen von JMS-Daten bereits im laufenden Betrieb vorzunehmen:

- Änderungen der Zugriffsrechte auf Jobklassen
- Zuordnung von Standard-Jobklassen an neue Benutzer
- Einrichten, Löschen und Modifizieren von Jobklassen-Definitionen.
- Einrichten, Löschen und Modifizieren von Jobstream-Definitionen.

Die Änderungen können, falls gewünscht, sofort wirksam werden.

> ¡ Die Anweisung SET-MODIFICATION-MODE ist nur erlaubt, wenn der Aufrufer das Privileg TSOS besitzt.

Format

SET-MODIFICATION-MODE
SCOPE = *FILE / *SYSTEM / *ALL

Operanden

SCOPE =
Legt fest, ob die nachfolgenden Änderungen in der Datei, in der Datenbasis oder in beiden erfolgen soll.

SCOPE = *FILE
Die Änderungen sollen nur in der Datei erfolgen, d.h. sie werden erst beim nächsten Startup wirksam.

SCOPE = *SYSTEM
Die Änderungen sollen nur in der Datenbasis erfolgen, d.h sie gelten nur für den aktuellen Systemlauf (für Testzwecke).

SCOPE = *ALL
Die Änderungen sollen sowohl in der Datei als auch in der Datenbasis erfolgen.

Hinweise

- Bei SCOPE=*SYSTEM / *ALL werden die Operanden STREAM, JOB-TYPE und START-ATTRIBUTE der Anweisung MODIFY-JOB-CLASS ignoriert.
- Bei der Vergabe von Kategorien können nur bereits definierte Kategorienamen verwendet werden.
- Beim Löschen von Jobklassen im aktuellen System bekommt die Klasse, falls noch Jobs in ihr aktiv sind, den Vermerk JOB-CLASS-IN-DELETE. Die bereits akzeptierten Jobs dürfen noch beendet werden, neue werden jedoch nicht mehr akzeptiert.
- SHOW-JOB-CLASS gibt als JMU-Anweisung nur den Inhalt der SJMSFILE aus. Bei SCOPE=*SYSTEM wird die Anweisung SHOW-JOB-CLASS nicht ausgeführt.
- Bei SCOPE=*ALL wird eine Änderung nur ausgeführt, wenn sie sowohl in der SJMSFILE als auch in der Datenbasis möglich ist.

6.3.2.13 SET-POSIX-JOB-CLASS-DEFAULT - POSIX-Standardklassen für Benutzer spezifizieren

Mit dieser Anweisung werden systemweite und/oder benutzerspezifische POSIX-Standardklassen für POSIX-FORK-Tasks spezifiziert.

Wenn weder eine systemweite noch eine für den betreffenden Benutzer geltende POSIX-Standardklasse definiert ist, dann werden POSIX-FORK-Tasks des Benutzers in der für ihn geltenden Standardklasse für Batch- oder Dialog-Aufträge gestartet.

Format

SET-POSIX-JOB-CLASS-DEFAULT
NAME = <name 1..8> ,ACTION = *ADD / *REMOVE ,USER = *ALL / list-poss(255): <name 1..8>

Die Bedeutung der Operanden ist die gleiche wie bei der Anweisung GRANT-JOB-CLASS-ACCESS.

Ein Ändern der POSIX-Systemstandardklasse beinhaltet die beiden Anweisungen:

- SET-POSIX-JOB-CLASS-DEFAULT standardklasse.alt,*REMOVE,*ALL
- SET-POSIX-JOB-CLASS-DEFAULT standardklasse.neu,*ADD,*ALL.

Wenn noch keine POSIX-Standardklasse zugewiesen war, dann entfällt die erste Anweisung.

Mit der Zuweisung einer Jobklasse als POSIX-Standardklasse wird, anders als bei SET-JOB-CLASS-DEFAULT, den ausgewählten Benutzern nicht automatisch der Zugriff auf diese Jobklasse gewährt. Dies muss, wenn erforderlich, vom Administrator explizit mit GRANT-JOB-CLASS-ACCESS vereinbart werden. Damit wird das Starten normaler Dialog- und Batch-Jobs in dieser Jobklasse nicht automatisch erlaubt. POSIX-FORK-Tasks können die Jobklasse jedoch nutzen.

Eine mit USER=*ALL vereinbarte POSIX-Standardklasse gilt zunächst systemweit für alle Benutzerkennungen. Sollen für bestimmte Benutzerkennungen andere POSIX-Standardklassen gelten, kann dies mit weiteren Anweisungen SET-POSIX-JOB-CLASS-DEFAULT vereinbart werden, z.B. mit den Operanden NAME=andere_jobklasse,ACTION=*ADD,USER=liste_der_benutzer.

Es können auch bereits mit SET-JOB-CLASS-DEFAULT definierte Standardklassen als POSIX-Standardklassen definiert werden.

Eine POSIX-Standardklasse kann zur Kategorie Dialog oder Batch gehören. Relevant ist sie für alle POSIX-FORK-Tasks vom Auftragstyp Dialog oder Batch.

6.3.2.14 SHOW-JOB-CLASS - Inhalt von Klassendefinitionen oder Namen von Klassen auflisten

Die Anweisung veranlasst die Auflistung des Inhalts von Klassendefinitionen oder von Klassennamen aus der SJMSFILE. In der Auflistung einer Klassendefinition erscheinen die Namen aller Benutzer, die Zugriff zu dieser Klasse haben.

Format

SHOW-JOB-CLASS
NAME = *ALL / *ALL-NAMES / list-poss(255): <name 1..8> ,OUTPUT = *SYSOUT / *SYSLST

Operanden

NAME =
Name der auszugebenden Klasse.

NAME = *ALL
Es sollen alle Klassendefinitionen aufgelistet werden.

NAME = *ALL-NAMES
Es sollen alle Namen der Klassen aufgelistet werden (ohne Inhalt der Klassendefinitionen).

NAME = <list-poss(255): <name 1..8>
Die Definitionen der Klassen, deren Namen angegeben sind, sollen aufgelistet werden. Maximal dürfen 255 Namen angegeben werden.

OUTPUT =
Bestimmt, wohin die Ausgabe erfolgen soll.

OUTPUT = *SYSOUT
Die Ausgabe soll nach SYSOUT erfolgen.

OUTPUT = *SYSLST
Die Ausgabe soll nach SYSLST erfolgen.

Beispiel

```
REQUESTED DETAILS OF JOB CLASS: JCBATCHF
NAME.........:JCBATCHF
STREAM........:JSSTD
CLASS LIMIT...:255
CLASS OPTIMUM.:0
WEIGHT........:3
JOB PRIORITY..:DEFAULT=3            MAXIMUM= 1
JOB ATTRIBUTES:JOBTYPE=BATCH        ST-ATTR= BATCH
TP ALLOWED....:TP
DIALOG ALLOWED:NO
BATCH ALLOWED.:BATCH
RUN PRIORITY..:DEFAULT=180          MAXIMUM= 30
NO CPU LIMIT..:YES
CPU LIMIT.....:DEFAULT=32767        MAXIMUM= 32767
SYSLST LIMIT..:DEFAULT=NO-LIMIT     MAXIMUM= NO-LIMIT
SYSOPT LIMIT..:DEFAULT=NO-LIMIT     MAXIMUM= NO-LIMIT
```

```
START.........:DEFAULT=SOON         ALLOWED= SOON EARLY AT LATE IN IMM STUP
REPEAT JOB....:DEFAULT=NO           ALLOWED= NO STUP DAILY WEEKLY PERIOD
JOB PARAMETER.:UNDEFINED
JCBATCHF IS AVAILABLE TO:
ALL USERS
JCBATCHF IS A DEFAULT FOR:
NO USERS
REQUESTED DETAILS OF JOB CLASS: JCBSTD
NAME..........:JCBSTD
STREAM........:JSSTD1
CLASS LIMIT...:50
CLASS OPTIMUM.:0
WEIGHT........:8
JOB PRIORITY..:DEFAULT=9            MAXIMUM= 1
JOB ATTRIBUTES:JOBTYPE=BATCH        ST-ATTR= BATCH
TP ALLOWED....:TP
DIALOG ALLOWED:NO
BATCH ALLOWED.:BATCH
RUN PRIORITY..:DEFAULT=220          MAXIMUM= 180
NO CPU LIMIT..:YES
CPU LIMIT.....:DEFAULT=32000        MAXIMUM= 32767
SYSLST LIMIT..:DEFAULT=NO-LIMIT     MAXIMUM= NO-LIMIT
SYSOPT LIMIT..:DEFAULT=NO-LIMIT     MAXIMUM= NO-LIMIT
START.........:DEFAULT=SOON         ALLOWED= SOON EARLY AT LATE IN IMM
REPEAT JOB....:DEFAULT=NO           ALLOWED= NO STUP DAILY WEEKLY PERIOD
JOB PARAMETER.:UNDEFINED
JCBSTD   IS AVAILABLE TO:
ALL USERS
JCBSTD   IS A SYSTEM DEFAULT
```

6.3.2.15 SHOW-JOB-STREAM - Inhalt von Streamdefinitionen oder Namen von Streams auflisten

Mit dieser Anweisung wird das Auflisten des Inhalts von Streamdefinitionen oder der Streamnamen aus der SJMSFILE veranlasst.

Format

SHOW-JOB-STREAM
NAME = *ALL / *ALL-NAMES / list-poss(255): <name 1..8> ,OUTPUT = *SYSOUT / *SYSLST

Operanden

NAME =
Name des auszugebenden Jobstreams.

NAME = *ALL
Es sollen alle Streamdefinitionen aufgelistet werden.

NAME = *ALL-NAMES
Es sollen alle Namen der Streams aufgelistet werden.

NAME = <list-poss(255): <name 1..8>
Es sollen alle Streamdefinitionen aufgelistet werden, deren Namen in dieser Liste angegeben sind. Maximal dürfen 255 Namen angegeben werden.

OUTPUT =
Bestimmt, wohin die Ausgabe erfolgen soll.

OUTPUT = *SYSOUT
Die Ausgabe der Streamdefinition(en) oder der Streamnamen soll nach SYSOUT erfolgen.

OUTPUT = *SYSLST
Die Ausgabe soll nach SYSLST erfolgen.

Beispiel

```
REQUESTED DETAILS OF JOB STREAM: JSSTD
NAME.........:JSSTD
FILE.........:SYSENT.JOBSCHED.150
RUN PRIORITY..:125
DEFAULT.......:NO
START.........:AT-LOAD
STOP.........:AT-SHUTDOWN
STREAMPARAM...:JOB-PRIORITY=Y,CPU-TIME=Y,WAIT-TIME=Y,JOB-QUOTA=50,LOGGING=NO
REQUESTED DETAILS OF JOB STREAM: JSSTD1
NAME.........:JSSTD1
FILE.........:SYSENT.JOBSCHED.150
RUN PRIORITY..:130
DEFAULT.......:YES
START.........:AT-LOAD
STOP.........:AT-SHUTDOWN
STREAMPARAM...:JOB-PRIORITY=Y,CPU-TIME=Y,WAIT-TIME=Y,JOB-QUOTA=30,LOGGING=NO
REQUESTED DETAILS OF JOB STREAM: JSSTD2
```

```
NAME.........:JSSTD2
FILE.........:SYSENT.JOBSCHED.150
RUN PRIORITY..:150
DEFAULT.......:NO
START.........:AT-LOAD
STOP..........:AT-SHUTDOWN
STREAMPARAM...:JOB-PRIO=Y,CPU-TIME=Y,WAIT-TIME=N,JOB-QUOTA=20,LOGGING=NO
REQUESTED DETAILS OF JOB STREAM: JSTSOS
NAME.........:JSTSOS
FILE.........:SYSENT.JOBSCHED.150
RUN PRIORITY..:120
DEFAULT.......:NO
START.........:AT-LOAD
STOP..........:AT-SHUTDOWN
STREAMPARAM...:JOB-PRIORITY=Y,CPU-TIME=Y,WAIT-TIME=Y,JOB-QUOTA=50,LOGGING=NO
```

7 LMSCONV Erstellen und Verwalten von Bibliotheken

Versionsstand: LMSCONV (SDF-Format) V3.6A

LMSCONV (Library Maintenance System-Converter) ist ein Programm zum Erstellen und Verwalten von Bibliotheken und den darin enthaltenen Elementen.

LMSCONV führt folgende Funktionen aus:

- Bibliotheken einrichten
- Dateien als Elemente in eine Bibliothek aufnehmen
- Elemente in Dateien ausgeben
- Elemente in eine andere Bibliothek kopieren
- Elemente auflisten
- Elemente löschen
- Elemente korrigieren
- Inhaltsverzeichnis einer Bibliothek ausgeben

LMSCONV unterstützt die Verarbeitung von Dateien > 32 GB (LARGE OBJECTS).

LMSCONV, als funktionell eingeschränkte Variante des Softwareprodukts LMS, wird sowohl mit ISP- als auch mit SDF-Oberfläche bereitgestellt. LMSCONV (ISP-Format) ist funktionell unverändert seit BS2000/OSD-BC V2.0. Das vorliegende Kapitel beschreibt die Funktionalität von LMSCONV (SDF-Format).

LMSCONV (SDF-Format) ist aus dem Software-Produkt LMS (SDF-Format) abgeleitet. Der Funktionsumfang ist eine Teilmenge von LMS (SDF-Format). Eine Auflistung der Einschränkungen im Vergleich zu LMS finden Sie im Abschnitt „LMSCONV im Vergleich zu LMS“.

Hinweise

- LMSCONV zeigt alle Attribute für Bibliotheken, Typen und Elemente an, d.h. auch solche, die nur mit LMS geändert werden können.
- Für die Verarbeitung der Bibliotheken unter LMSCONV sind alle, also auch die nicht direkt über LMSCONV beeinflussbaren Attribute maßgeblich. Somit ist eine Bibliotheksverarbeitung u.U. nur eingeschränkt möglich (z. B. kein Leserecht für Element).
- Elemente eines definierten Benutzertyps sind mit LMSCONV bearbeitbar.
- Bibliothekslisten und Typumsteuerung können benutzt werden.
- Bei der Installation von LMSCONV kann festgelegt werden, welchen Namen die einzelnen Produktdateien erhalten und in welcher Kennung sie abgelegt sind. Die vollständigen Pfadnamen aller Produktdateien von LMSCONV können über /SHOW-INSTALLATION-PATH ermittelt werden.

Überblick über die LMSCONV-Anweisungen

Anweisung	Bemerkungen
ADD-ELEMENT	Aufnehmen von Dateien als Elemente
CLOSE-LIBRARY	Schließen von Bibliotheken
COPY-ELEMENT	Kopieren von Elementen
COPY-LIBRARY	Kopieren einer Bibliothek
DELETE-ELEMENT	Löschen von Elementen
END	Beenden des LMSCONV-Laufes
EXTRACT-ELEMENT	Elemente in Dateien ausgeben
MODIFY-DEFAULTS	Ändern von globalen Default-Werten
MODIFY-ELEMENT	Ändern von Elementen
MODIFY-ELEMENT-ATTRIBUTES	Ändern von Elementattributen
MODIFY-LOGGING-PARAMETERS	Ändern der Protokollmenge
OPEN-LIBRARY	Eröffnen einer globalen Bibliothek
SHOW-DEFAULTS	Anzeigen der globalen Default-Werte
SHOW-ELEMENT	Anzeigen des Elementinhaltes
SHOW-ELEMENT-ATTRIBUTES	Anzeigen der Elementattribute
SHOW-LIBRARY-ATTRIBUTES	Anzeigen der Bibliotheksattribute
SHOW-LIBRARY-STATUS	Anzeigen der Bibliothekszustände
SHOW-LOGGING-PARAMETERS	Anzeigen der Werte für die Protokollmenge
SHOW-TYPE-ATTRIBUTES	Anzeigen der Typattribute
SHOW-USER-EXITS	Anzeigen der LMSCONV-Version
WRITE-COMMENT	Kommentar ins Protokoll schreiben

LMSCONV Eingabe- und Ausgabestrom

Das folgende Bild stellt die Ein- und Ausgabemöglichkeiten von LMSCONV dar:

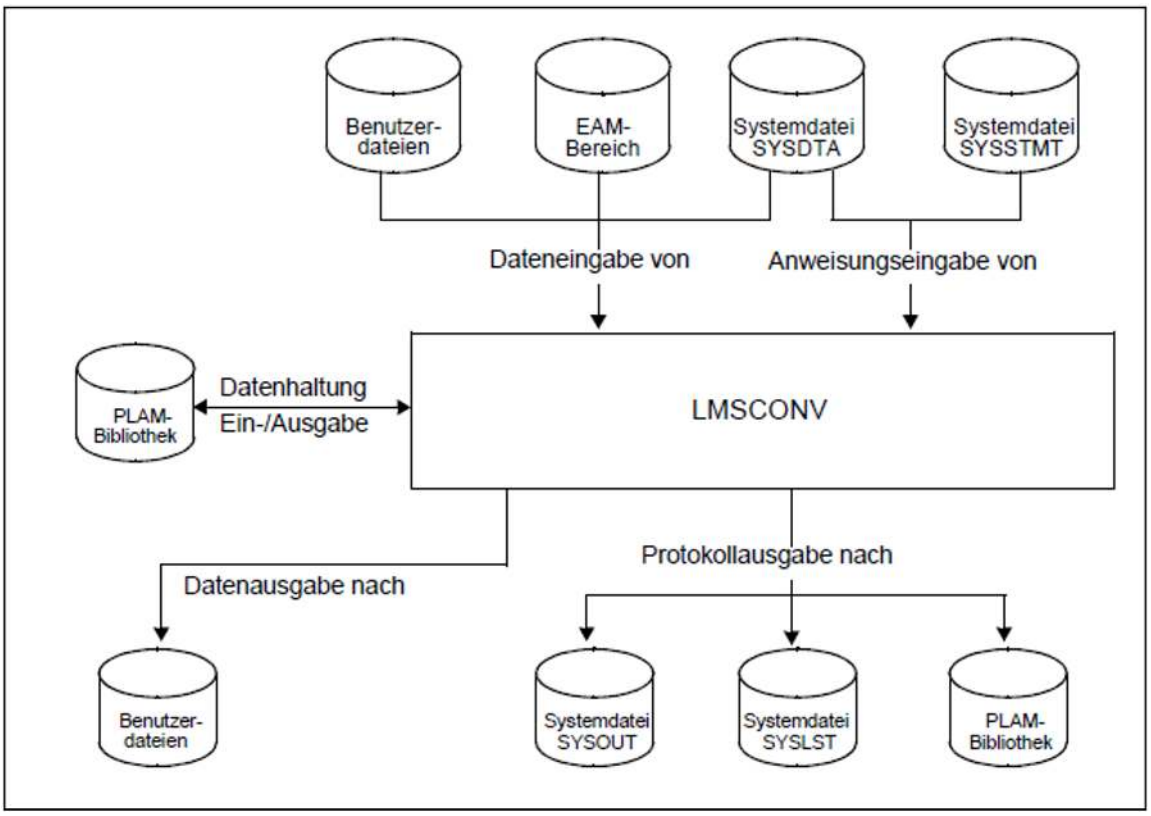


Bild 2: Ein-/Ausgabestrom von LMSCONV

LMSCONV liest alle Anweisungseingaben über die Dialogschnittstelle SDF. Als Unterlage siehe dazu das Handbuch „Dialogschnittstelle SDF“ [20].

Beispiel für einen LMSCONV-Lauf

```
(IN)     start-lmsconv                                                  1.
(OUT)    %  LMC0310 LMSCONV VERSION '<version>' STARTED
(IN)     modify-logging-parameters logging=*maximum                     2.
(IN)     open-library library=ueb.bib,mode=update                       3.
(OUT)    LIBRARY IS CLEARED AND PREPARED                                4.
(IN)     add-element from-file=test, to-elem=(elem=test.ueb,type=s)     5.
(OUT)    INPUT  FILE
(NL)     OUTPUT LIBRARY= :1OSN:$USER.UEB.BIB
(NL)              ADD :1OSN:$USER.TEST AS (S)TEST.UEB/@(0001)/<date>
(IN)     show-element-attributes                                        6.
(OUT)    INPUT  LIBRARY= :1OSN:$USER.UEB.BIB
(NL)     TYP NAME     VER (VAR#) DATE
(NL)     (S) TEST.UEB @   (0001) <date>
(NL)            1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS             7.
(IN)     show-element (element=test.ueb, type=s)                        8.
(OUT)    INPUT  LIBRARY= :1OSN:$USER.UEB.BIB
(NL)     INPUT  ELEMENT= (S)TEST.UEB/@(0001)/<date>
(NL)     TESTFILE, CAN BE DELETED AT ANY TIME !!                        9.
(NL)     NUMBER OF PROCESSED RECORDS IS       1
```

```
(IN)     end                                                          10.
(OUT)    %  LMC0311 LMSCONV VERSION '<version>' TERMINATED NORMALLY
```

1. LMSCONV wird aufgerufen.
2. Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
3. Die Bibliothek UEB.BIB wird als globale Bibliothek (d.h. durch die Anweisung OPEN-LIBRARY zugewiesen) neu eingerichtet und als Ein- und Ausgabebibliothek zugewiesen.
4. Die Bibliothek UEB.BIB ist eingerichtet.
5. Die Datei TEST wird als Element TEST.UEB vom Typ S in die Bibliothek aufgenommen.
6. Das Inhaltsverzeichnis der Bibliothek UEB.BIB soll aufgelistet werden.
7. Inhaltsverzeichniseintrag der Bibliothek UEB.BIB.
8. Das Element TEST.UEB soll aufgelistet werden.
9. Inhalt des Elementes TEST.UEB.
10. LMSCONV wird beendet.

LMSCONV im Dialog- und Batch-Betrieb

LMSCONV läuft im Dialog- und Batch-Betrieb.

Dialogbetrieb

Nachdem Elemente auch durch „wildcard“-Angaben ausgewählt werden können, ist zunächst nicht erkennbar, welches Element gerade bearbeitet wird. Daher ist bei Anweisungen, die Elementdaten löschen oder überschreiben, ein „schrittweises“ Vorgehen möglich.

Für jedes Element wird der Anwender gefragt, ob er das Element bearbeiten, überspringen oder die Anweisung abbrechen will.

Der Fragemechanismus wird durch den Operanden DIALOG-CONTROL=*YES eingeleitet.

Mit der Anweisung MODIFY-DEFAULTS kann der Defaultwert für DIALOG-CONTROL geändert werden.

Mit DIALOG-CONTROL = *ERROR kann der Fragemechanismus auf Fehlerfälle beschränkt werden. Bei folgenden nicht behebbaren Fehlern wird im Dialog auch bei DIALOG-CONTROL = *NO ein Fragemechanismus eingeleitet:

- wenn ein Element nicht zugreifbar ist, z.B. weil es durch einen anderen Benutzer gesperrt ist (vorübergehende exklusive Nutzung eines Elementes z.B. durch Modifizieren)
- wenn eine Bibliothek nicht zugreifbar ist, z.B. weil die aktuellen Zugriffsrechte keinen Zugriff erlauben (vorübergehende exklusive Nutzung einer Bibliothek).

Mit der Anweisung MODIFY-DEFAULTS, Operand RUN-MODE=*BATCH, kann LMSCONV so eingestellt werden, dass es wie im Batchbetrieb abläuft, was u.a. den Fragemechanismus vollständig deaktiviert.

Batch-Betrieb

Ist eine Bibliothek, ein Element oder ein Typ gesperrt, kann der Benutzer im Batch-Betrieb die Anzahl der Neuversuche und das Zeitintervall zwischen zwei Versuchen über die Anweisung MODIFY-DEFAULTS ..., NEXT-ATTEMPT einstellen. Standardmäßig wird kein Neuversuch unternommen.

7.1 Bibliotheken

LMSCONV bearbeitet PLAM-Bibliotheken. PLAM-Bibliotheken sind PAM-Dateien von BS2000, die mit der Bibliotheks-Zugriffsmethode PLAM (Program Library Access Method) bearbeitet werden.

Eine PLAM-Bibliothek enthält Elemente und ein Inhaltsverzeichnis der gespeicherten Elemente. PLAM-Bibliotheken dienen zum Speichern von Quellprogrammen, Makros, Bindemodulen, Phasen, Listen, Prozeduren, Text usw.

PLAM-Bibliotheken werden auch kurz als „Bibliotheken“ bezeichnet.

7.1.1 Logischer Aufbau einer Bibliothek

Eine Bibliothek ist eine Datei mit Unterstruktur. Sie enthält Elemente und ein Inhaltsverzeichnis.

Ein Element ist eine logisch zusammengehörige Datenmenge wie z.B. eine Prozedur, ein Bindemodul oder ein Quellprogramm. Jedes Element ist in der Bibliothek einzeln ansprechbar.

Das Speichern von Dateien als Elemente in einer Bibliothek entlastet den Dateinamenskatalog, da jede Bibliothek nur einen Katalogeintrag hat. Speicherplatz wird gespart, da die Elemente in der Bibliothek in komprimierter Form gespeichert werden. Weiterhin können Elemente auch als Delta-Elemente gespeichert werden. Dies ist aber nur mit LMS möglich..

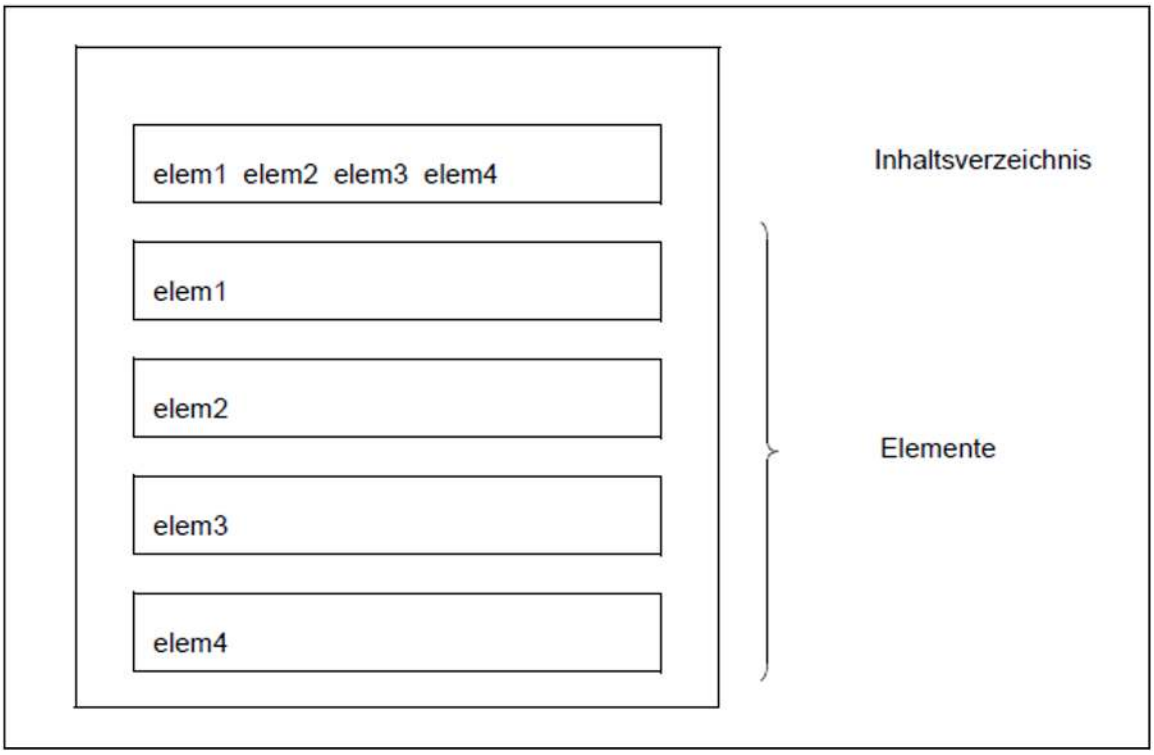


Bild 3: Logischer Aufbau einer Bibliothek

Jede Bibliothek hat einen Eintrag im Systemkatalog. Der Benutzer kann den Namen und andere Dateimerkmale, wie z.B. die Schutzfrist oder die Mehrbenutzbarkeit, festlegen. Der Eintrag im Katalog bzw. Änderungen darin werden vom Benutzer über Systemkommandos vorgenommen.

7.1.2 Ein- und Ausgabebibliotheken

LMSCONV bearbeitet eine Bibliothek als Ein- und/oder Ausgabebibliothek:

Eine Eingabebibliothek wird global über die LMSCONV-Anweisung OPEN-LIBRARY oder lokal über den Operanden ELEMENT=*LIBRARY-ELEMENT(LIBRARY=...) zugewiesen.

Eine Ausgabebibliothek wird global über die LMSCONV-Anweisung OPEN-LIBRARY oder lokal über den Operanden TO-ELEMENT=*LIBRARY-ELEMENT(LIBRARY=...) zugewiesen.

7.1.3 Mehrfachzugriff auf Bibliotheken

Unter Mehrfachzugriff auf Bibliotheken versteht man den gleichzeitigen mehrfachen lesenden oder schreibenden Zugriff auf eine Bibliothek. Die Zugriffe können aus verschiedenen Tasks erfolgen. Die Tasks können unter verschiedenen Benutzerkennungen ablaufen.

Mehrfachzugriff auf Bibliotheken ist unter folgenden Bedingungen uneingeschränkt möglich:

- Die für den Zugriff erforderlichen Zugriffsrechte sind gegeben
- Es erfolgt kein Zugriff über Remote File Access (RFA)
- Für Bibliotheken auf Shared Pubsets:
  Die zugreifenden Tasks laufen auf verschiedenen Systemen ab, die einen XCS-Verbund bilden.

Der Mehrfachzugriff kann durch den Anwender mit folgenden Kommandos eingeschränkt werden:

`/ADD-FILE-LINK ...,SUPPORT=*DISK(SHARED-UPDATE=*NO)`

> Bibliotheken werden standardmäßig mit `SHARED-UPDATE=*YES` eröffnet.
> `/ADD-FILE-LINK ...,SUPPORT=*DISK(SHARED-UPDATE=*NO)` und ein Eröffnen dieser Bibliothek mit dem angegebenen Linknamen sperrt weitere Update-Zugriffe aus.

`/SECURE-RESOURCE-ALLOCATION FILE=`

> Die Bibliothek wird exklusiv dieser Task reserviert.

`/MODIFY-FILE-ATTRIBUTES ...,PROTECTION=`

> Nur noch Benutzer, die das erforderliche Zugriffsrecht besitzen, können auf die Bibliothek zugreifen.

Beschreibung dieser Kommandos, siehe Handbuch „Kommandos“ [1].

7.2 Elemente

Ein Element ist eine logisch zusammengehörige Datenmenge wie z.B. eine Datei, eine Prozedur, ein Bindemodul oder ein Quellprogramm. Jedes Element ist in der Bibliothek über seine Elementbezeichnung einzeln ansprechbar.

Die Elementbezeichnung identifiziert ein Element und besteht aus den drei Teilen Name, Version und Typ.

Name: Der Bestandteil Name beschreibt den logischen Inhalt des Elements.

Version: Der Bestandteil Version beschreibt den aktuellen Entwicklungsstand des Elementes.

Typ: Der Bestandteil Typ dient der Klassifizierung der Elemente.

Bibliotheken können alle von LMSCONV unterstützten Elementtypen enthalten.

7.2.1 Mehrfachzugriff auf Elemente

Ein Element kann gleichzeitig von mehreren Benutzern gelesen, jedoch nur von einem geschrieben werden. Während das Element zum Schreiben eröffnet ist, kann kein anderer Zugriff - auch kein lesender - auf dieses Element erfolgen, jedoch auf andere Elemente der Bibliothek.

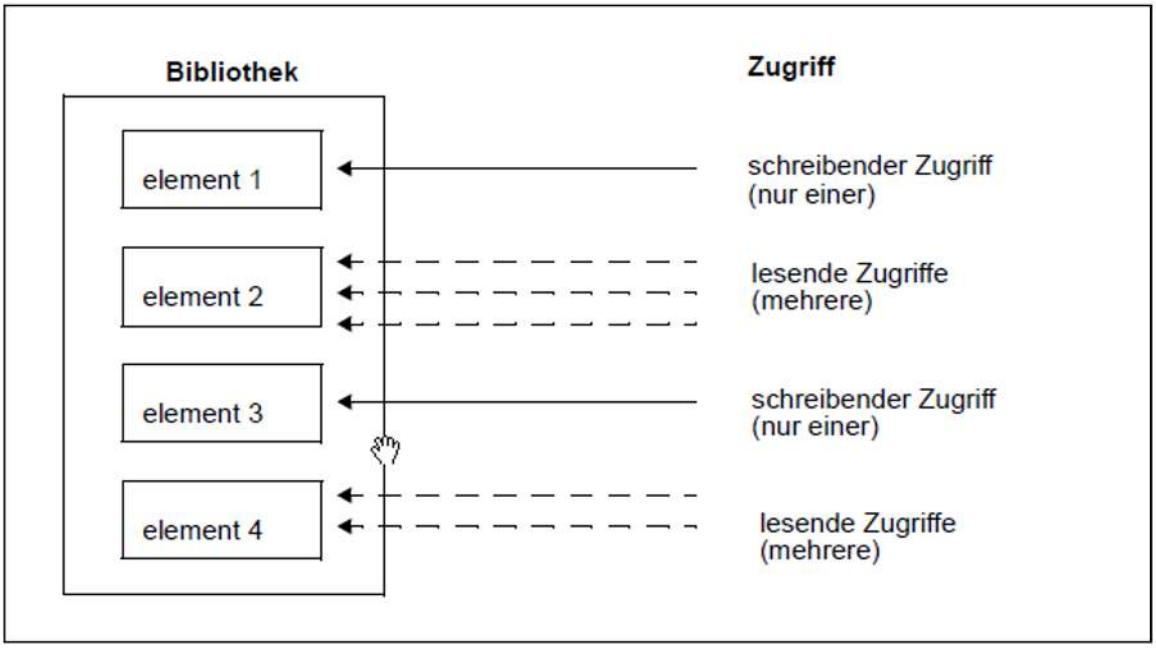


Bild 4: Mehrfachzugriff auf Elemente

Durch den Mehrfachzugriff auf eine Bibliothek ist es möglich, dass ein Element beim Auflisten des Inhaltsverzeichnisses noch vorhanden ist, beim anschließenden Zugriff jedoch nicht mehr existiert. Ein anderer Benutzer hat es in der Zwischenzeit gelöscht.Das Auflisten des Inhaltsverzeichnisses (Anweisung SHOW-ELEMENT-ATTRIBUTES) zeigt also nur den momentanen Zustand der Eingabebibliothek.

7.2.2 Beschreibung der Elementtypen

Der Elementtyp gibt an, wie der Inhalt der Elemente zu interpretieren ist.

Standardtypen

Standard- oder vordefinierte Typen sind 1 Zeichen lang bzw. beginnen mit $ oder SYS

C	Phasen
D	Textdaten
F	IFG-Formatmasken
H	Compiler-Ergebnisinformationen
J	Prozeduren
L	Bindelademodule (LLM)
M	Makros
P	druckaufbereitete Daten
R	Bindemodule
S	Quellprogramme
U	IFG-Benutzer-Profile
X	Daten beliebigen Formates
SYSJ	kompilierte Prozedur
$...	Reserviert
SYS...	Reserviert

Zur Archivierung können Elemente mit einer Satzlänge von maximal 32 KByte (inkl. Satzkopf) aufgenommen werden.

Elementtyp C - Phasen

Eine vom Binder TSOSLNK erzeugte Phase wird standardmäßig in einer Datei abgelegt. Diese Datei kann mit LMSCONV in einer Bibliothek als Element vom Typ C abgelegt werden. Wahlweise können die vom Binder erzeugten Phasen auch direkt in einer Bibliothek abgelegt werden.

Elementtyp D - Textdaten

In Elemente vom Typ D kann beliebiger Text geschrieben werden. Es sind die gleichen Funktionen wie bei Elementtyp S möglich.

Elementtyp F - IFG-Formatmasken

Elemente dieses Typs werden von IFG erzeugt und in Bibliotheken abgelegt.

Elementtyp H - Compiler-Ergebnisinformationen

Elemente dieses Typs werden von den Compilern und vom Assembler erzeugt und in Bibliotheken abgelegt.

Elementtyp J - Prozeduren

In diesem Elementtyp werden BS2000-Prozeduren abgelegt.

Elementtyp L - Bindelademodule (LLM)

In Elemente vom Typ L legt der Binder BINDER (siehe Handbuch „BINDER“ [12]) als auch die COMPILER, die erzeugten Bindelademodule (LLMs) ab.

Elementtyp M - Makros

Der Assembler entnimmt der zugewiesenen Bibliothek die im Programm angesprochenen Makroelemente.

Elementtyp P - Listenelemente

Als Listenelemente werden druckaufbereitete Daten bezeichnet. Das erste Zeichen des Satzes muss ein gültiges Vorschubsteuerzeichen sein; dies wird bei der Ausgabe in die Systemdatei SYSLST geprüft.

Elementtyp R - Bindemodule

Von Compilern und dem Assembler erzeugte Bindemodule im temporären EAM-Bereich können mit LMSCONV in der Bibliothek als Elemente vom Typ R abgelegt werden. Wahlweise können die von Compilern und dem Assembler erzeugten Bindemodule auch direkt in einer Bibliothek abgelegt werden.
Den Bindern und dem dynamischen Bindelader dienen diese Elemente als Eingabe.

Elementtyp S - Quellprogramme

Quellprogramme in Bibliotheken dienen den Compilern und dem Assembler bei Übersetzungsläufen als Eingabe.

Elementtyp U - IFG-Benutzer-Profile

Elemente dieses Typs werden von IFG erzeugt und in Bibliotheken abgelegt.

Elementtyp X - Daten beliebigen Formates

Der Elementtyp X kann beliebige Daten aufnehmen.

textartige Elementtypen - Textelemente

Als textartige Elementtypen werden die Typen S, M, J, P, D, X bezeichnet. Textelemente sind Elemente dieser Typen, soweit sie keine blockorientierten Sätze enthalten.

PAM-Elemente

Elemente mit blockorientierten Sätzen werden im Folgenden auch als PAM-Elemente bezeichnet, da sie i.d.R. durch Aufnehmen einer Datei mit FILE-STRUCTURE=PAM als Element entstanden sind.

7.2.3 Konvention für Elementbezeichnungen

Elemente werden in Bibliotheken über eine Elementbezeichnung identifiziert. Diese wird im Inhaltsverzeichnis der Bibliothek hinterlegt und kann mit der LMSCONV-Anweisung SHOW-ELEMENT-ATTRIBUTES ausgegeben werden.

Die Elementbezeichnung besteht aus den drei folgenden Teilen:

- Elementname für den logischen Inhalt der Elemente,
- Elementversion für den aktuellen Stand der Elemente,
- Elementtyp für die Klassifizierung der Elemente.

7.2.4 Elementbezeichnung in den Anweisungen

Die Elementbezeichnung, d.h. Elementname, -version und -typ, entspricht in den LMSCONV-Anweisungen den Operanden ELEMENT, VERSION und TYPE in der Datenstruktur *LIBRARY-ELEMENT (siehe "Konstruktionsangabe für Elementbezeichnungen").

Die Angabe der Version ist wahlfrei. Wird in einer Anweisung kein Wert für Version eingegeben, wird standardmäßig das Element mit der höchsten Version ausgewählt. Soll eine andere Version ausgewählt werden, so ist zu beachten, dass die Versionsangabe eine Unterstruktur des Elementnamens ist.

```
*LIBRARY-ELEMENT(...)
    |
    |  ELEMENT = <composed-name ... >(...)
    |     <composed-name ...>(...)
    |        |  VERSION = ...
    |  ,TYPE = ...
 ...
```

Syntax der Elementbezeichnung

ELEMENT	composed-name 1..64 with-under with-wild(132)(...) Der Elementname darf mit einer, maximal vierstelligen, Catid beginnen.
VERSION	composed-name 1..24 with-under with-wild(52)(...) Weitere Informationen zu den möglichen Versionsangaben, vor allem über die Verwendung und Bedeutung von Schlüsselwörtern, siehe Abschnitt „Versionsverwaltung“. Die Version '@' ist nicht erlaubt, da sie als Darstellung für die Versionsangabe '*UPPER-LIMIT' dient.
TYPE	alphanum-name 1..8 with-wild(20)

7.2.5 Protokollierung der Elementbezeichnung

Bei jeder Ausgabe von LMSCONV werden die Elementbezeichnungen wie folgt protokolliert:

```
(Typ)Elementname/Version[(Variantennummer)]/Datum
```

Die Variantennummer ist standardmäßig auf (0001) gesetzt und wird durch jeden schreibenden Zugriff um 1 erhöht.

7.2.6 Auswahlangabe für Elementbezeichnungen

Sollen in den LMSCONV-Anweisungen bestimmte Elemente zur Bearbeitung ausgewählt werden, so ist dies auf zwei Arten möglich:

- durch die Verwendung von „wildcard“-Angaben in den Operanden ELEMENT, VERSION und TYPE.

  Die Wildcardsyntax ist im Benutzerhandbuch „Kommandos“ [1] beschrieben.

  Bei den Operanden ELEMENT, VERSION und TYPE kann für den Einzelstern (*) auch das Schlüsselwort *ALL verwendet werden.

  Eine Negativauswahl, d.h. Ausschließen von Elementen, kann durch ein Minuszeichen vorgenommen werden.
- durch die Qualifikation von Attributen, z.B. Datum und Uhrzeit.

Beispiele für die Auswahlangabe

- Auswahlangabe

  ELEMENT=AB/C*

  Alle Elemente, deren Name mit AB beginnt, an der 3. Stelle ein beliebiges Zeichen, an der 4. Stelle ein C besitzt, werden ausgewählt. Ab der 5. Stelle kann der Elementname beliebigen Inhalt haben.

  ELEMENT=<:999>(VERSION=B*)

  Alle Elemente mit einer Namenslänge von maximal 3 Zeichen, die an der ersten Stelle der Version ein B besitzen, werden ausgewählt.

  ELEMENT=*(VERSION=*), CREATION-DATE=*INTERVAL(FROM=2016-01-01)

  Alle Elemente, die seit dem 1.1.2016 eingetragen wurden, werden ausgewählt.

  ELEMENT=AB*(VERSION=*HIGHEST-EXISTING)

  Von allen Elementen, deren Name mit AB beginnt, wird jeweils die höchste Version ausgewählt.
- Auswahlangaben mit Grenzwerten

  ELEMENT=A*(VERSION=*HIGHEST-EXISTING), USER-DATE=*INTERVAL(TO=2016-12-31)

  Alle Elemente der höchsten Version, deren Name mit A beginnt, und die älteren Datums als 1.1.2017 sind, werden ausgewählt.

  ELEMENT=AB<:9>(V=107)

  Alle Elemente, deren Name mit AB beginnt, maximal 3 Zeichen lang ist und die Version 107 haben, werden ausgewählt.
- Auswahlangaben mit auszuschließenden Elementen

  ELEMENT=-ABC

  Alle Elemente, außer dem Element ABC, werden ausgewählt.

  TYPE=-S

  Alle Elemente, außer Elemente vom TYPE=S, werden ausgewählt.

7.2.7 Konstruktionsangabe für Elementbezeichnungen

Bei LMSCONV-Anweisungen, die neben der Auswahlangabe auch eine zweite Elementbezeichnung zulassen, kann die Bezeichnung des zweiten Elementes aus der Bezeichnung der Auswahlangabe konstruiert werden.

Die Konstruktionsangabe beschränkt sich auf die Elementbezeichnung, d.h. auf Elementname, -version und -typ. Dies entspricht in den Anweisungen den Operanden ELEMENT, VERSION und TYPE in der Datenstruktur *LIBRARY-ELEMENT. Dabei werden jeweils gleichnamige Operanden, die durch bestimmte Platzhalter gekennzeichnet sind, aufeinander abgebildet.

Die Konstruktionssyntax ist im Benutzerhandbuch „Dialogschnittstelle SDF“ [20] beschrieben.

Beispiele für Konstruktionsangaben

Im Folgenden wird dargestellt, wie LMSCONV Elementnamen konstruiert. Der Elementtyp S sei mit der Anweisung MODIFY-DEFAULTS voreingestellt. Die einzelnen Beispiele sind unabhängig voneinander zu sehen, d.h. das Ergebnis wird nicht weiter verwendet.

Es seien Elemente wie folgt gegeben:

```
TYP  NAME    VER (VAR#) DATE          NAME    VER (VAR#) DATE
(S)  ABC     001 (0001) <date>        ABCD    001 (0001) <date>
(S)  ABCDE   001 (0001) <date>        X.1     001 (0001) <date>
```

Anweisung	Wirkung
`//COPY-ELEMENT (,ELEM= ABC),` `  TO-ELEM=(,ELEM= ABX)`	`ABC` wird kopiert nach `ABX` `ABCD` wird nicht kopiert `ABCDE` wird nicht kopiert

`//COPY-ELEMENT (,ELEM= AB*),`
`  TO-ELEM=(,ELEM= XY*(A02))`
	`ABC` wird kopiert nach `XYC(A02)`
	`ABCD` wird kopiert nach `XYCD(A02)`
	`ABCDE` wird kopiert nach `XYCDE(A02)`

`//COPY-ELEMENT (,EL= ABC<,D>),`
`  TO-ELEM=(,ELEM= AXC<1>)`
	`ABC` wird kopiert nach `AXC`
	`ABCD` wird kopiert nach `AXCD`
	`ABCDE` wird nicht kopiert

`//COPY-ELEMENT (,ELEM= AB*),`
`  TO-ELEM=(,ELEM= S.AB*)`
	`ABC` wird kopiert nach `S.ABC`
	`ABCD` wird kopiert nach `S.ABCD`
	`ABCDE` wird kopiert nach `S.ABCDE`

`//COPY-ELEMENT (,ELEM= *B*),`
`  TO-ELEM=(,ELEM= *.*)`
	`ABC`
	`ABCD`
	`ABCDE`
		wird kopiert nach `A.C`
		wird kopiert nach `A.CD`
		wird kopiert nach `A.CDE`

`//COPY-ELEMENT (,ELEM= /B/),`
`TO-ELEM=(,ELEM= //)`
	`ABC`
	`ABCD`
	`ABCDE`
		wird kopiert nach `AC`
		wird nicht kopiert
		wird nicht kopiert

`//COPY-ELEMENT (,ELEM= *CD*),`
`  TO-ELEM=(,ELEM= <1>.<1>)`
	`ABC`
	`ABCD`
	`ABCDE`
		wird nicht kopiert
		wird kopiert nach `AB.AB`
		wird kopiert nach `AB.AB`

`//COPY-ELEMENT (,ELEM= /B/),`
`  TO-ELEM=(,ELEM= <2> <1>)`
	`ABC`
	`ABCD`
	`ABCDE`
		wird kopiert nach `CA`
		wird nicht kopiert
		wird nicht kopiert

`//COPY-ELEMENT (,ELEM= /B/),`
`  TO-ELEM=(,ELEM= XYZ<2>)`
	`ABC`
	`ABCD`
	`ABCDE`
		wird kopiert nach `XYZC`
		wird nicht kopiert
		wird nicht kopiert

`//COPY-ELEMENT (,ELEM= X.),` `  TO-ELEM=(,ELEM= Y.)`	`X.1`	wird kopiert nach `Y.1`

Tabelle 3: Wirkung der COPY-Anweisung

Bei Verwendung von Auswahl- und Konstruktionsangabe ist zu beachten:

- Mindestens ein Platzhalter des Quellelementes muss in der Konstruktionsangabe vorkommen.
- Möglicherweise werden verschiedene Eingabebezeichnungen auf dieselbe Ausgabebezeichnung abgebildet. Je nachdem, wie der Verarbeitungsoperand WRITE-MODE gesetzt ist, werden die verschiedenen Daten überschrieben. Beispiel: /A/ -> BA/. Es wird sowohl das Element XA1 als auch das Element XA2 auf BAX abgebildet.
- *ALL aus der Auswahlangabe kann wie der Einzelstern (*) aus der Auswahlangabe mit einem * in der Konstruktionsangabe referenziert werden. *ALL in der Konstruktionsangabe ist nicht möglich. Beispiel: *ALL -> *B* ist gleichbedeutend mit * -> *B*

7.2.8 Elementattribute

Alle Elemente besitzen unabhängig von ihrem Typ gewisse Attribute:

- Erstellungsdatum und -zeit (CREATION-DATE und CREATION-TIME)
- Änderungsdatum und -zeit (MODIFICATION-DATE und MODIFICATION-TIME)
- Benutzerdatum und -zeit (USER-DATE und USER-TIME)
- Sekundärnamen und -attribut (SECONDARY-NAME und SECONDARY-ATTRIBUTE) (Referenzeinträge) (*1)
- Name eines Zeichensatzes (CODED-CHARACTER-SET)

Die Werte dieser Attribute werden bei der Ausgabe des Inhaltsverzeichnisses einer Bibliothek angezeigt.

(*1) Referenzeinträge sind Einträge im Sekundär_inhalts_verzeichnis der Bibliothek. Sie entstehen, wenn der Anwender beim Schreiben eines Elementes Referenzsätze (Satzart 163) der Form <Sekundärname> <Sekundärattribut> erzeugt.

Durch die Referenzeinträge wird die Beziehung „Ein bestimmter <sekundärname> und <sekundärattribut> kommt im Element vor“ dokumentiert. Die typweise Sortierung der Referenzeinträge erlaubt die Abfrage: „In welchem Element des Typs TYPE kommt ein bestimmter Referenzeintrag vor?“.

Dies wird durch die folgende LMSCONV-Anweisung realisiert:

```
//SHOW-ELEMENT-ATTR (*STD,*,TYPE= ...,SECONDARY-NAME=.., -
      SECONDARY-ATTRIBUTE=...,)
```

Diese Beziehung wird bei Modulen (Typ R oder L) als Basis der Autolink-Funktion (siehe Handbuch „Binder “ [12]) verwendet;
Referenzeinträge sind z.B. <name> <CSECT> bzw. <name> <ENTRY>.

Eingabeformat des Datums

Das Eingabeformat eines Datums lautet:

```
[JJ]JJ-MM-TT      [JJ]JJ  :  Jahr; wahlweise 2- oder 4-stellig

                  MM      : Monat

                  TT      : Tag
```

Bei 2-stelliger Eingabe der Jahresangabe ergänzt LMSCONV die Jahrhundertangabe durch die Verwendung eines Referenzjahres.

```
JJ   <    60      -->      20JJ

JJ   >=   60      -->      19JJ
```

7.2.9 Typabhängigkeiten

Die folgende Tabelle zeigt, welche Elementtypen bei den einzelnen LMSCONV-Anweisungen möglich sind und welche Typprüfungen LMSCONV vornimmt. Nur wenn die in der Spalte Typprüfung aufgeführten Bedingungen zutreffen, wird die Anweisung ausgeführt.

Anweisung	Elementtyp		
	Quelle	Ziel	Prüfung
SHOW-ELEMENT SHOW-ELEM-ATTR DELETE-ELEMENT	alphanum 1...8 alphanum 1...8 alphanum 1...8		
COPY-ELEMENT	alphanum 1...8	alphanum 1...8	Quelle = Ziel
MODIFY-ELEMENT	R,C,L S,M,P,D,J,X	R,C,L S,M,P,D,J,X	Quelle = Ziel Quelle = Ziel
MOD-LOGGING-PAR		P	
ADD-ELEMENT	"text "(I)SAM-file "blocks " PAM-file "module " file,*OMF "phase " PAM-file	S,M,P,D,J,X X R C	
EXTRACT-ELEMENT	S,M,P,D,J,X X R C	(I)SAM-file PAM-File (I)SAM-file PAM-File	

Tabelle 4: Mögliche Elementtypen bei LMSCONV-Anweisungen

7.2.10 Versionsverwaltung

Die Version eines Elementes ist in der Elementbezeichnung festgelegt und kennzeichnet den aktuellen Stand des Elementes.

Das nachfolgende Kapitel beschreibt die möglichen Versionsbezeichnungen und gibt Hinweise zur Versionshaltung und -speicherung.

Versionshaltung und -speicherung

In Bibliotheken wird ein Element durch seinen Typ, seinen Namen und eine Versionsbezeichnung eindeutig bestimmt. Weiter ist es möglich, zu einem Elementtyp und -namen mehrere Versionen zu speichern.

Falls der Benutzer bei der Bearbeitung keine Angaben zur Version macht, führt LMSCONV standardmäßig folgende Aktionen aus:

- Beim Lesen wird das Element gesucht, das zu dem angegebenen Namen die höchste Versionsbezeichnung besitzt. Das Datum wird dabei nicht berücksichtigt.
- Beim Schreiben hängt die Versionsbehandlung von der Anweisung ab:
  - `//ADD-ELEMENT` und `MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=`

    Das Element wird mit der höchsten Version X'FF' erzeugt bzw. überschrieben. LMSCONV stellt diese Version mit @ dar.
  - sonstige Anweisungen
    Das Ausgabeelement erhält die Versionsbezeichnung des Eingabeelementes.

    Wird dabei ein gleichnamiges Element überschrieben, wird die interne Variantennummer um 1 erhöht. Sie dient als Schreibzugriffszähler.

Verschiedene Stände eines Entwicklungsobjektes sind in verschiedenen Elementen abgelegt. Die Beziehungen zwischen einzelnen Elementen kennt zunächst nur der Anwender; sie sind nicht in der Bibliothek verankert. Jedes Element ist eine eigenständige Einheit im Kontext einer Bibliothek.

Versionsbezeichnungen

Die durch LMSCONV zu bearbeitenden Elementversionen werden in den LMSCONV-Anweisungen durch Angabe der Operanden VERSION bestimmt. Es wird unterschieden zwischen Bestimmung der Quellversion und der Zielversion.

- **Quellversion**
  Wird ein Element als Eingabe einer Funktion verwendet, z.B. wenn es kopiert oder verändert werden soll, dann wird die Elementversion Quellversion genannt.

  Die Quellversion kann wie folgt bestimmt werden:
  - composed-name
    Die durch composed-name bestimmte Version wird ausgewählt.
  - *UPPER-LIMIT
    Es wird die höchste mögliche Version (intern durch X'FF' dargestellt) ausgewählt.
  - *HIGHEST-EXISTING
    Die höchste existierende Version des angegebenen Elementnamens wird ausgewählt.

- **Zielversion**
  Wird ein Element als Ergebnis einer Funktion verwendet, z.B. wenn es zurückgeschrieben wird, dann wird die Elementversion Zielversion genannt.

  Die Zielversion kann wie folgt bestimmt werden:

  - *BY-SOURCE
    Die Quellversion ist auch Zielversion. Ist die Quelle kein Bibliothekselement, ist *UPPER-LIMIT Zielversion.
  - *UPPER-LIMIT
    Die absolut höchste Version, intern dargestellt durch X'FF', ist Zielversion.
  - composed-name
    Die durch composed-name bestimmte Version ist Zielversion. Die Angabe von '@' wird zurückgewiesen.

7.2.11 Datenschutz durch Überschreiben

Datenschutz durch Überschreiben bedeutet, dass der Anwender gezielt nicht mehr benötigte Dateien durch Überschreiben löscht. Die Daten werden dadurch physikalisch gelöscht, d.h. mit X'00' überschrieben. Das Überschreiben der Daten wird lokal gesteuert durch die LMSCONV-Anweisung DELETE-ELEMENT bzw. global gesteuert in MODIFY-DEFAULTS, jeweils durch den Operanden DESTROY-DATA.

Der Operand DESTROY-DATA ist einerseits ein Elementattribut, d.h. das Überschreiben wirkt automatisch auf dieses Element, und andererseits ein Verarbeitungsparameter der LMSCONV-Anweisung DELETE-ELEMENT. Als Verarbeitungsparameter bewirkt DESTROY-DATA, dass alle Elemente, die durch die Anweisung erfasst sind, beim Löschen überschrieben werden.

Die Daten werden mit X'00' überschrieben, wenn eine der folgenden Angaben das Überschreiben verlangt:

- Systemparameter DESTLEV
- Angabe beim Element: Wert von DESTROY-DATA beim letzten Anlegen bzw. beim letzten schreibenden Zugriff auf das Element
- Angabe über den Operanden DESTROY-DATA der Anweisung MODIFY-DEFAULTS

7.2.12 Beweissicherung

Die von LMSCONV verwendete Zugriffsmethode PLAM besitzt einen Anschluss an das Subsystem SAT (security audit trail) im Sicherheitspaket SECOS (siehe Handbuch „SECOS“ [21]). Bei aktivem SAT können vom Sicherheitsbeauftragten folgende Ereignisse zur Protokollierung ausgewählt werden:

- CREATE ELEMENT
- MODIFY ELEMENT
- READ ELEMENT
- EXECUTE ELEMENT
- CLOSE ELEMENT
- DELETE ELEMENT
- RENAME ELEMENT
- CREATE SECURITY ATTRIBUTES
- MODIFY SECURITY ATTRIBUTES
- DELETE SECURITY ATTRIBUTES

7.2.13 Extended Host Code Support (XHCS)

LMSCONV unterstützt die Verwendung spezieller (nationaler) Zeichensätze. Dieser, dem Element zugeordneter Coded-Character-Set-Name (CCSN) wird - soweit möglich - an Schnittstellen weitergegeben und bei Ausgaben berücksichtigt.

Wird XHCS an der entsprechenden Schnittstelle nicht angeboten, so wird stets der Default „kein Code“ verwendet

LMSCONV selbst setzt keinen bestimmten Zeichensatz voraus und wertet auch nicht die Default-Einstellung der Userid aus; interne Sortiervorgänge, z.B. der Elementbezeichnung, erfolgen unabhängig vom eingestellten CCS.

1. Implizites Setzen des CCSN für ein Element
   Jedem Element einer Bibliothek kann ein Zeichensatz zugeordnet werden. Dabei überträgt LMSCONV immer den CCSN der Quelle auf das Zielelement.
   Das kann beim Aufnehmen einer Datei mittels der ADD-ELEMENT-Anweisung durch das Übernehmen des Katalogattributes CCS geschehen. Der CCSN wird jedoch nicht zusätzlich im Attributsatz (Satzart 164) gespeichert, um Inkonsistenzen zu vermeiden.

   Werden die Daten aus der logischen Systemdatei SYSDTA aufgenommen, so wird jeweils der eingestellte Zeichensatz bestimmt, und der Name wird dem Element als Attribut zugewiesen.

   Bei Aufnahme von Moduln aus der EAM-Datei erhalten diese das Attribut „kein Code“.

   Beim Kopieren von Elementen wird immer der CCSN des Quellelementes dem Zielelement zugewiesen.
2. Protokollierung des CCSN
   Mit der Anweisung SHOW-ELEMENT-ATTRIBUTES und dem Operanden INFORMATION=*MAXIMUM besteht die Möglichkeit, sich über die Zuordnung von Coded-Character-Sets zu Elementen zu informieren. Bei Elementen, die den CCSN „kein Code“ haben, wird dieses Attribut nicht im Inhaltsverzeichnis angezeigt.
3. Auswertung und Weitergabe des CCSN eines Elementes
   Bei der Ausgabe von Elementen in Dateien wird der entsprechende CCSN des Elementes der Datei zugewiesen.

   Für die von LMSCONV erzeugten Ausgabeinformationen gilt:
   Bei der Ausgabe von Elementsätzen nach SYSOUT (auch in aufbereiteter Form) mit der Anweisung SHOW-ELEMENT wird das CCS des entsprechenden Elementes benutzt. Ist SYSOUT einer Datei zugewiesen, so muss der Anwender dieser Datei explizit mit /MODIFY-FILE-ATTRIBUTES den gewünschten Zeichensatz zuweisen.

   Bei der Ausgabe der Elementsätze nach SYSLST wird kein CCSN ausgewertet. Ist SYSLST einer Datei zugeordnet, so kann der Anwender dieser Datei explizit mit /MODIFY-FILE-ATTRIBUTES den gewünschten Zeichensatz zuweisen.

   Das Extrahieren eines Elements in ein SAM-Node-File auf Net-Storage kann daran scheitern, dass der CCSN des Elements nicht in den NETCCSN des Node-Files konvertiert werden kann. Dies kann mehrere Ursachen haben:

   a. Dem Element ist ein unpassender CCSN zugewiesen. In diesem Fall kann mit //MODIFY-ELEMENT-ATTRIBUTES ein passender CCSN zugewiesen werden.
   b. Für das Element ist in den Dateiattributen ein unpassender NETCCSN gespeichert (zu erkennen in der Ausgabe von //SHOW-ELEMENT TEXT-INFORMATION=*FILE-ATTRIBUTES). In diesem Fall muss dem Node-File, in das das Element extrahiert werden soll, mit /MODIFY-FILE-ATTRIBUTES der passende NETCCSN zugewiesen werden. Anschließend kann das Element unter Angabe von FILE-ATTRIBUTES=*BY-CATALOG extrahiert werden.
   c. Für das Element ist kein NETCCSN gespeichert (z.B. weil es ursprünglich kein Node-File war oder weil beim Aufnehmen des Node-Files in die Bibliothek der Operand SOURCE-ATTRIBUTES=*KEEP nicht

angegeben wurde), und der voreingestellte NETCCSN des Benutzers passt nicht zum CCSN. Allgemein sollte in diesem Fall das Benutzerattribut NET-CODED-CHARACTER-SET passend geändert werden. Im Einzelfall kann aber auch genauso vorgegangen werden wie in Fall b.

Ist der Ausgabestrom durch //MODIFY-LOGGING-PARAMETERS in ein Bibliothekselement umgewiesen worden, so erhält dieses Element den CCSN „kein Code“.

Bei der Ausgabe von Inhaltsverzeichnissen oder sonstiger von LMSCONV selbst erzeugter Elementinformation wird immer der CCSN „kein Code“ angenommen.

4. Erweiterung von Elementen und Dateien mit WRITE-MODE=*EXTEND
   Wird mit WRITE-MODE=*EXTEND gearbeitet, so prüft LMSCONV die CCS-Namen von Quelle und Ziel. Stimmen diese nicht überein, wird die Bearbeitung mit einer Fehlermeldung abgewiesen.

7.3 Funktionen von LMSCONV

Dieses Kapitel gibt einen Überblick über die Funktionen von LMSCONV.

7.3.1 Starten von LMSCONV

LMSCONV wird durch folgendes Kommando aufgerufen:

```
/START-LMSCONV
```

START-LMSCONV	Alias: **LMSCONV**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Voreinstellungen nach dem Starten von LMSCONV

Nach dem Aufrufen von LMSCONV sind Default-Werte gültig. Diese gelten für einige Operanden der Anweisungen, in denen das Schlüsselwort *DEFAULT angegeben werden kann. Die Default-Werte können durch die Anweisung MODIFY-DEFAULTS geändert werden.

Die folgende Tabelle stellt dar, auf welche Anweisung die Default-Werte wirken und welche Default-Werte das Verhalten von LMSCONV beeinflussen.

LMSCONV-Default	**Default-Wert**	**Wirkung auf**	**Bedeutung der Defaults**
ELEMENT-ATTRIBUTES TYPE ELEMENT-VERSION TO-ELEM-VERSION SOURCE-ATTRIBUTES	*UNDEFINED *ALL *BY-SOURCE *IGNORE	Anweisungen mit Typangabe SHOW-ELEMENT-ATTR Zielversion ADD-ELEMENT	Festlegung des Elementtyps Festlegung der Quellversion Festlegung der Zielversion Behandlung der Dateiattribute
FILE-ATTRIBUTES	*ISAM	EXTRACT-ELEMENT	Attribute für die Zieldatei
DESTROY-DATA	*NO	DELETE-ELEMENT[1)]	Überschreiben von Daten
WRITE-MODE	*CREATE	ADD-ELEMENT COPY-ELEMENT EXTRACT-ELEMENT MODIFY-ELEMENT	Überschreiben der Elemente

LMSCONV-Default	Default-Wert	Wirkung auf	Bedeutung der Defaults
DIALOG-CONTROL	*NO	ADD-ELEMENT COPY-ELEMENT DELETE-ELEMENT EXTRACT-ELEMENT MODIFY-ELEMENT MODIFY-ELEMENT-ATTRIBUTES	dialoggeführter Fragemechanismus, der jedes Element einzeln anbietet.
INFORMATION LAYOUT SORT	*MEDIUM *VARIABLE *BY-NAME	SHOW-ELEM-ATTR " "	Informationsmenge Darstellung Sortierreihenfolge
OUTPUT-FORM	*STD	SHOW-ELEMENT	Ausgabeform
DELETE-SOURCE	*NO	ADD-ELEMENT	Quelldatei löschen
MAX-ERROR-WEIGHT	*SERIOUS	alle Anweisungen	Spin-off-Steuerung
NEXT-ATTEMPT	*NO	Anweisungen mit *LIB-ELEM-Angabe	Steuert die Öffnungsversuche bei Datei/Typ /Elementsperre
TEXT-INFORMATION MODULE-INFORMATION PHASE-INFORMATION LLM-INFORMATION	*ALL *ALL *ALL *ALL	SHOW-ELEMENT " " "	Info für textart. Elemente Modulinformation (TypR) Phaseninformation (Typ C) LLM-Information (Typ L)

Tabelle 5: Wirkung der Default-Werte

[1)] Wenn der Operand DESTROY-DATA mit MODIFY-DEFAULTS verändert wird, wirkt er indirekt auch auf die Anweisungen ADD-, COPY- und EXTRACT-ELEMENT.

7.3.2 Zuweisen von Bibliotheken

In LMSCONV-Anweisungen werden Bibliotheken durch den Operanden LIBRARY angegeben. Es gibt mehrere Möglichkeiten, die letztlich direkt oder indirekt auf Bibliotheken führen:

1. Global vereinbarte Bibliothek (*STD)
2. Direkter Name einer Bibliothek
3. Indirekt über Linkname

Die Möglichkeiten 2 und 3 können zur Vereinbarung der globalen Bibliothek genutzt werden.

Wenn in einer LMSCONV-Anweisung Bibliotheken verwendet werden, dann öffnet LMSCONV alle Bibliotheken, die in der Anweisung verwendet werden. LMSCONV schließt alle anderen offenen Bibliotheken unabhängig davon, wieviele Bibliotheken bisher im LMSCONV-Programmlauf verwendet worden waren. Anweisungen, die keine Bibliotheken verwenden (z.B. SHOW-DEFAULTS), ändern die Zustände von Bibliotheken nicht.

Die LMSCONV-Anweisung SHOW-LIBRARY-STATUS gibt Auskunft über den Zustand der bearbeiteten Bibliotheken.

Global vereinbarte Bibliothek (*STD)

Die global vereinbarte Bibliothek wird durch OPEN-LIBRARY zugewiesen und durch CLOSE-LIBRARY auf undefiniert zurückgesetzt.

Direkter Name einer Bibliothek

Der angegebene Dateiname bezeichnet eine Bibliothek. Diese Angabe wird beim Arbeiten mit LMSCONV häufig benutzt.

Indirekt über Linkname

Der unter *LINK angegebene Name <link> bezeichnet einen Linknamen. Im Allgemeinen ist der Linkname ein mit /ADD-FILE-LINK vereinbarter Dateikettungsname, dem über den Operanden FILE-NAME eine Bibliothek zugeordnet ist.

7.3.3 Bearbeiten von Elementen

Das folgende Kapitel gibt einen Überblick über die Möglichkeiten, mit LMSCONV Elemente zu bearbeiten. LMSCONV kann Elemente

- in Bibliotheken aufnehmen,
- in Dateien ausgeben,
- in andere Bibliotheken ausgeben (kopieren),
- auflisten,
- löschen,
- korrigieren,
- und das Inhaltsverzeichnis der Bibliothek ausgeben.

Aufnehmen von Elementen in eine Bibliothek

Folgende Anweisungen nehmen Elemente in die zugewiesene Bibliothek auf: ADD-ELEMENT, COPY-ELEMENT und MODIFY-LOGGING-PARAMETERS.

Der Operand WRITE-MODE bestimmt dabei, ob ein gleichbezeichnetes Element in der Ausgabebibliothek überschrieben wird oder nicht.

ADD-ELEMENT

Die Anweisung ADD-ELEMENT (siehe "ADD-ELEMENT - Element in Bibliothek aufnehmen") nimmt Dateien, Module aus dem EAM-Bereich und Datensätze aus dem LMSCONV-Anweisungsstrom als Elemente in die zugewiesene Bibliothek auf. Wird keine Bibliothek angegeben, wird die durch Anweisung OPEN-LIBRARY eröffnete Bibliothek genommen.

Die Satzformate FIXED und UNDEFINED werden in das Satzformat VARIABLE umgewandelt; d.h. mit 4-Byte-Satzkopf versehen. Die Satzlänge inklusive Satzkopf darf 32KByte nicht überschreiten.

Wird eine ISAM-Datei aufgenommen, bestimmt der Operand SOURCE-ATTRIBUTES, ob die Dateiattribute, der ISAM-Schlüssel und Informationen über ISAM-Sekundärschlüssel mit aufgenommen werden.

Es können dann ISAM-Schlüssel bis zu einer Länge von 255 Byte abgelegt werden. Elemente mit ISAM-Schlüssel eignen sich nur für die Archivierung (siehe Hinweise unten).

Ist der Operand SOURCE-ATTRIBUTES=*KEEP gesetzt, ist es auch möglich, Dateien mit RECORD-FORMAT=FIXED aufzunehmen, ansonsten ist nur RECORD-FORMAT = VARIABLE zulässig.

Der Last Byte Pointer einer PAM-Datei bleibt stets erhalten, wenn sie als Element vom Typ X in eine Bibliothek aufgenommen wird. Wird sie dagegen als Element vom Typ C aufgenommen, bleibt ihr LBP nur bei Angabe von SOURCE-ATTRIBUTES=
*KEEP(KEEP-TYPES=*ALL) erhalten.

Der Zeichensatz von SAM-Node-Files auf Net-Storage (NETCCSN) bleibt nur erhalten, wenn sie mit ADD-ELEMENT SOURCE-ATTRIBUTES=*KEEP in die Bibliothek aufgenommen werden.

> ¡ Die ISAM-Schlüssel einer Quellprogrammdatei sollten nicht mit ins Element aufgenommen werden. Der Compiler kann das Quellprogramm, wenn ISAM-Schlüssel enthalten sind, aus diesem Element nicht fehlerfrei übersetzen.

Wird die Systemdatei SYSDTA einem Element zugewiesen, das die ISAM-Schlüssel gespeichert hat, werden die ISAM-Schlüssel mitgelesen. Die ISAM-Schlüssel müssen dann vom verarbeitenden Programm selbst entfernt werden.

Dateien können unter folgenden Elementtypen abgelegt werden:

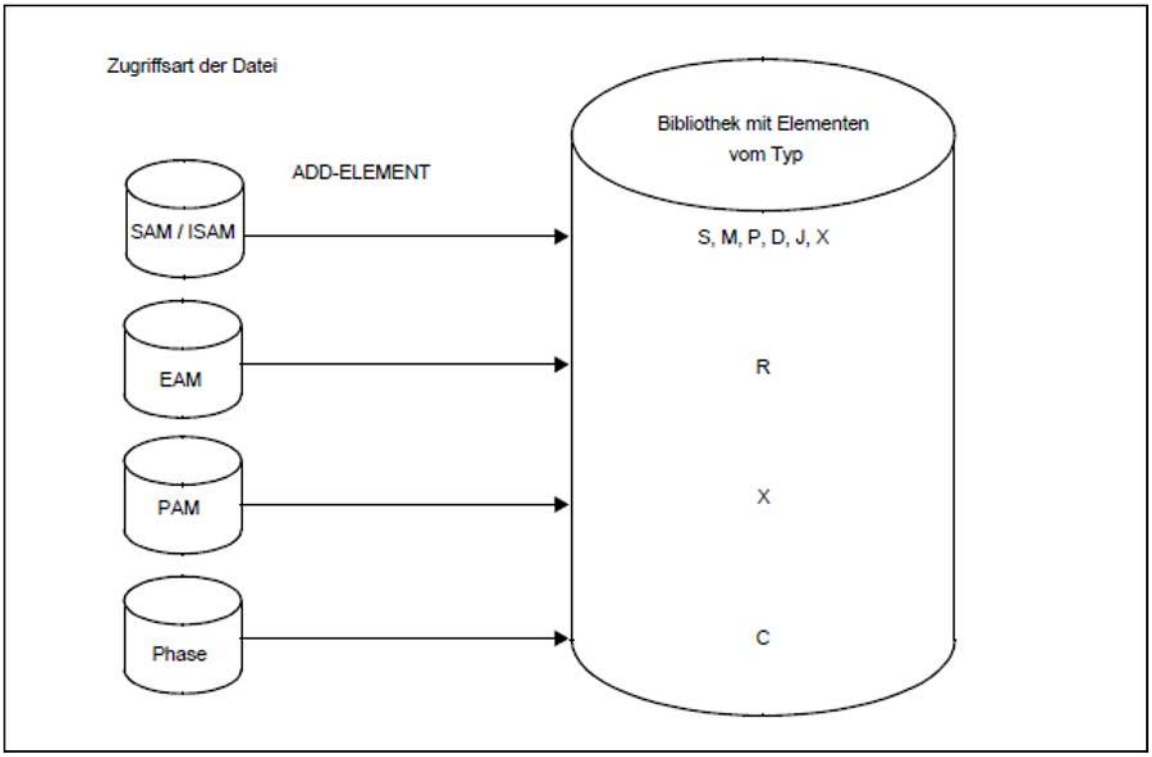


Bild 5: Aufnehmen von Elementen mit ADD-ELEMENT

COPY-ELEMENT

Die Anweisung COPY-ELEMENT (siehe "COPY-ELEMENT - Element kopieren") kopiert Elemente von der Eingabebibliothek in die Ausgabebibliothek und legt sie dort, falls gewünscht, mit einer anderen Elementbezeichnung ab:

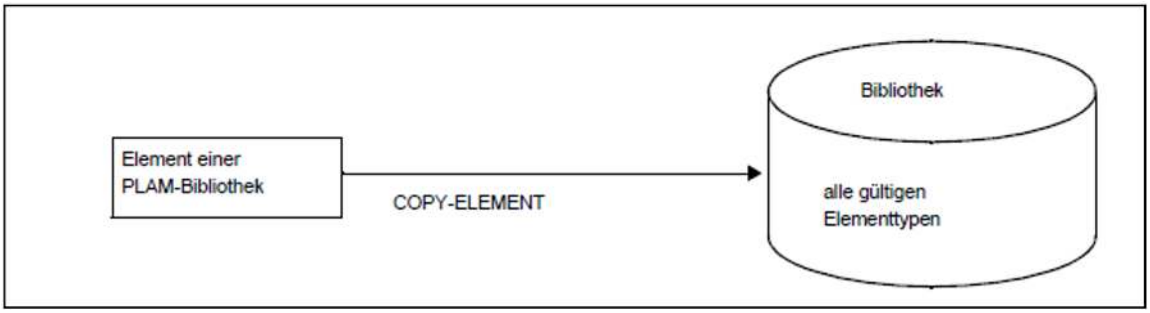


Bild 6: Aufnehmen von Elementen mit COPY-ELEMENT

MODIFY-LOGGING-PARAMETERS

Die Anweisung MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=*LIBRARY-ELEMENT (siehe "MODIFY-LOGGING-PARAMETERS - Protokolleinstellungen ändern") schreibt das LMSCONV-Protokoll in das durch *LIBRARY-ELEMENT spezifizierte Element:

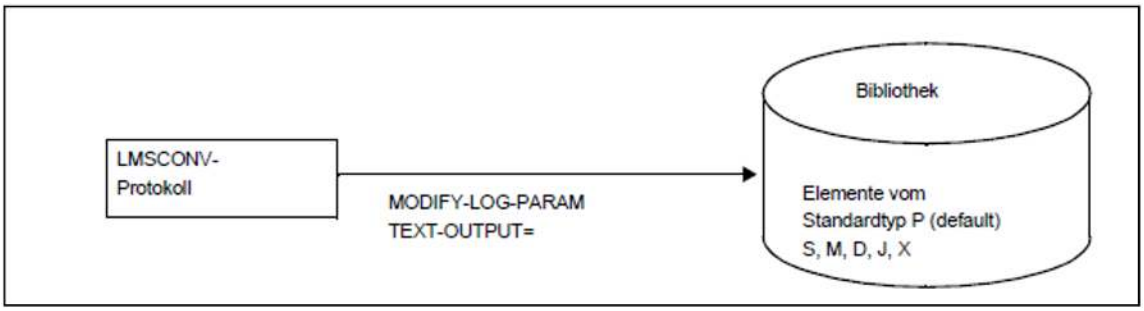


Bild 7: Schreiben des LMSCONV-Protokolls in ein Element

Ausgeben von Elementen in eine Datei

Die Elemente einer Bibliothek werden mit der Anweisung EXTRACT-ELEMENT (siehe "EXTRACT-ELEMENT-Elemente in Datei ausgeben") in eine Datei ausgegeben:

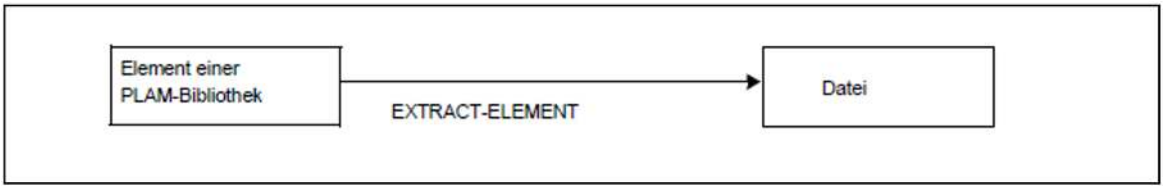


Bild 8: Ausgeben von Elementen

Dateiattribute, die beim Aufnehmen eines Elements gespeichert worden waren, werden beim Ausgeben des Elements wieder hergestellt. Dies gilt auch für den Last Byte Pointer (LBP) bei PAM-Dateien und den CCSN von SAM-Node-Files im Net-Storage (NETCCSN).

> Wenn beim Aufnehmen eines Elements der LBP gespeichert worden war, dann kann das Element nicht auf Band oder Privatplatte ausgegeben werden.
> LMS mit einer Version kleiner als V3.4C kann Elemente vom Typ C mit gespeicherten Dateiattributen nicht ausgeben und liefert eine leere PAM-Datei und/oder eine Fehlermeldung.

Auflisten von Elementen

Die Anweisung SHOW-ELEMENT (siehe "SHOW-ELEMENT - Elementinhalt anzeigen") zeigt den Elementinhalt an. Dabei kann bestimmt werden, in welchem Format und welche Informationsmenge ausgegeben werden soll.

Löschen von Elementen

Die Anweisung DELETE-ELEMENT (siehe "DELETE-ELEMENT - Ein Element logisch löschen") löscht Elemente in der zugewiesenen Bibliothek. Es wird zwischen logischem und physikalischem Löschen unterschieden:

- Logisches Löschen
  Die Einträge im Inhaltsverzeichnis werden gelöscht und der Speicherplatz des entsprechenden Elementes wird freigegeben (Freigabe erfolgt nur innerhalb der Bibliothek, d.h. die Bibliothek wird dadurch als Ganzes nicht kleiner).
- Physikalisches Löschen
  Zusätzlich zum logischen Löschen wird der Speicherplatz des entsprechenden Elementes mit binären Nullen überschrieben.

Ein Element einer Bibliothek wird physikalisch gelöscht, wenn der Operand DESTROY-DATA=*YES gesetzt ist oder im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder der Systemparameter DESTLEV es verlangt.

Korrigieren von Elementen

LMSCONV besitzt folgende Korrekturanweisung:

MODIFY-ELEMENT (siehe "MODIFY-ELEMENT - Element verändern") korrigiert Binde-, Bindelademodule, Phasen und Textelemente über Subanweisungen (Elementtyp R, L, C und Elementtyp S, M, J, P, D, X).

Das Korrigieren dieser Elemente wird durch verschiedene Subanweisungen gesteuert. Diese werden unmittelbar hinter MODIFY-ELEMENT bis zur Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

Das korrigierte Element wird dann in die zugewiesene Bibliothek zurückgeschrieben. Es kann dabei eine neue Elementbezeichnung erhalten.

Für Typ R, L und C sind folgende Funktionen möglich:

- Textsätze korrigieren
- Korrekturen rückgängig machen
- Satzarten aus dem Eingabeelement löschen

Zusätzlich sind nur für Typ R noch folgende Funktionen möglich:

- REP-Sätze erzeugen
- Merkmale der Programmabschnitte ändern
- Symbole umbenennen

Für Textelemente sind folgende Funktionen möglich:

- Einfügen von Sätzen
- Löschen von Sätzen

Inhaltsverzeichnis einer Bibliothek ausgeben

Die Anweisung SHOW-ELEMENT-ATTRIBUTES (siehe "SHOW-ELEMENT-ATTRIBUTES - Elementeigenschaften anzeigen") protokolliert die Inhaltsverzeichniseinträge der angegebenen Elemente oder der gesamten Bibliothek.

Das Inhaltsverzeichnis wird immer nach dem Elementtyp sortiert ausgegeben. Die weitere Sortierreihenfolge bestimmt der Operand SORT. Die Elementbezeichnungen werden standardmäßig nach Typ, Namen und Version sortiert ausgegeben.

Um das vollständige Inhaltsverzeichnis einer Bibliothek zu erhalten, genügt die Angabe der Anweisung SHOW-ELEMENT-ATTRIBUTES ohne weitere Operanden, sofern über MODIFY-DEFAULTS kein bestimmter Elementtyp festgelegt wurde.

Speichern von Prozeduren

LMSCONV ermöglicht dem Benutzer Prozeduren und ENTER-Jobs als Elemente in Bibliotheken abzulegen (Elementtyp J).

Bestehende Prozedurdateien können mit ADD-ELEMENT als Elemente in Bibliotheken aufgenommen werden.

Durch das Speichern von Prozeduren kann, insbesondere bei kleinen Kommandodateien, Speicherplatz eingespart werden. Die Zahl der Katalogeinträge wird verringert.

Es ist jedoch darauf zu achten, dass aus den Elementen, die evtl. vorhandene ISAM-Schlüssel mitgespeichert haben (siehe "Bearbeiten von Elementen"), diese Schlüssel vor dem Aufruf der Prozedur entfernt werden.

Ein Bibliothekselement kann auch als Systemeingabedatei (SYSDTA) mit /ASSIGN-SYSDTA zugewiesen werden (siehe Handbuch „Kommandos“ [1]).

7.3.4 Steuern des LMSCONV-Laufs

Der nachfolgende Abschnitt beschreibt die Möglichkeiten, die LMSCONV bietet, um den gesamten LMSCONV-Lauf zu steuern.

Protokollausgabe steuern

Das LMSCONV-Protokoll enthält alles, was LMSCONV ausgibt, wie z.B. das Ergebnis der Anweisungen, deren Ausführung oder Abbruch, die zugewiesenen Ein- und Ausgabebibliotheken und Listen, die beim Auflisten von Elementen erzeugt wurden.

Das Protokoll wird in die Systemdatei SYSOUT, SYSLST oder in ein Bibliothekselement geschrieben. Wohin LMSCONV ausgibt, legt die Anweisung MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=... fest.

Wenn das Protokoll in ein Element geschrieben wird, dann erzeugt LMSCONV standardmäßig ein Element mit dem Typ P.

Fehlermeldungen werden immer ausgegeben.

Die folgende Tabelle zeigt, welche Operanden in welchen Anweisungen die Ausgabe des Protokolls steuern:

Anweisung	Operand	Funktion
MODIFY-LOGGING-PARAMETERS	LOGGING	Legt fest, ob Erfolgsmeldungen protokolliert werden oder nicht
	TEXT-OUTPUT	Legt das Ausgabemedium für das Protokoll fest
	OUTPUT-LAYOUT	Legt das Ausgabeformat des Protokolls fest
	LINES-PER-PAGE	Legt die Anzahl der Zeilen auf einer Protokollseite fest
	LINE-SIZE	Legt die Länge einer Zeile fest.
	EXTRA-FORM-FEED	Steuert den Seitenvorschub des Protokolls bei Elementwechsel
	HEADER-LINES	Legt fest, ob Überschriften ausgegeben werden
SHOW-ELEMENT	OUTPUT-FORM	Legt die Satzdarstellung beim Auflisten eines Elementes fest.
	TEXT-/ MODULE-/ PHASE-/ LLM-INFORMATION	Legt den Ausgabeumfang beim Auflisten eines Elementes fest.
SHOW-ELEMENT-ATTRIBUTES	SORT	Legt die Sortierung des Inhaltsverzeichnisses fest.
	LAYOUT	Legt das Protokollformat des Inhaltsverzeichnisses fest.
	INFORMATION	Legt den Protokollumfang des Inhaltsverzeichnisses fest.

	TEXT-OUTPUT	Steuert die Protokollausgabe

Tabelle 6: Steuerung der Protokollausgabe

Erfolgs- und Misserfolgsmeldungen

Ist in der LMSCONV-Anweisung MODIFY-LOGGING-PARAMETERS der Operand LOGGING=*MAXIMUM gesetzt, wird die Ausführung jeder LMSCONV-Anweisung, die ein Element betrifft, protokolliert. Wird die Anweisung fehlerfrei ausgeführt, dann gibt LMSCONV eine Erfolgsmeldung aus.

Kann die Anweisung nicht ausgeführt werden, protokolliert LMSCONV dies mit einer Misserfolgsmeldung und evtl. der entsprechenden LMSCONV-Fehlermeldung.

Alle Erfolgs- und Misserfolgsmeldungen haben folgendes Format:

```
[NO] anweisung elem[wort elem][ursache]
```

Bedeutung

NO	Die Anweisung wird nicht ausgeführt
anweisung	Anweisungsname
elem	Elementbezeichnung oder Dateiname (bei ADD-ELEMENT und EXTRACT-ELEMENT)
wort	Schlüsselwort: AS, INTO, WITH
ursache	Ergebnis: EXISTING, REPLACED, u.Ä.

Bildschirmwechsel steuern

LMSCONV führt keine eigene Bildschirmsteuerung durch. Die Steuerung übernimmt BS2000. LMSCONV-Ausgaben können daher nur abgebrochen werden, wenn die Programmunterbrechungstaste (K2) gedrückt wird und anschließend /INFORM-PROGRAM eingegeben und abgeschickt wird, siehe "MODIFY-DEFAULTS - Voreinstellungen verändern".

Fehlerbehandlung im Dialog- und Prozedurbetrieb

LMSCONV unterscheidet zwischen den beiden Ablaufarten Dialog- und Prozedurbetrieb.

- Dialogbetrieb

  Im Dialog erfolgt nach der Ausgabe der Fehlermeldung die Aufforderung zur Eingabe der nächsten Anweisung mit //.

- Prozedurbetrieb

  Der Anwender kann selbst bestimmen, in welchen Fehlerfällen LMSCONV den Spin-off-Mechanismus auslösen soll. Dies steuert er über den Operanden MAX-ERROR-WEIGHT (siehe Anweisung MODIFY-DEFAULTS). Abhängig von diesem Operanden verzweigt LMSCONV nach Ausgabe der Fehlermeldung zur nächsten STEP- oder END-Anweisung. Tritt während der Verarbeitung von Subanweisungen ein schwer wiegender Fehler auf, wird die zugehörige Hauptanweisung abgebrochen und der Spin-off-Mechanismus ausgelöst. D.h. hinter STEP wird immer eine Hauptanweisung erwartet. Wurde in LMSCONV der Spin-off-Mechanismus ausgelöst, ohne dass eine STEP-Anweisung gelesen wurde, so beendet sich LMSCONV mit TERM UNIT=STEP, MODE = ABNORMAL.

Unterbrechen des LMSCONV-Laufs

Der Benutzer kann den LMSCONV-Lauf durch Betätigen einer Programmunterbrechungstaste (z.B. K2) unterbrechen.

Die Fortsetzung des LMSCONV-Laufs kann durch /INFORM-PROGRAM gesteuert werden, das wahlweise mit einem Eingabetext versehen werden kann. Dieser Eingabetext wird dann in der Unterbrechungsbehandlung von LMSCONV interpretiert. Die gerade laufende Funktion wird von der Art des Abbruchs unterrichtet und beendet sich in der gewünschten Form. Die möglichen Eingaben sind beim Operanden DIALOG-CONTROL der Anweisung MODIFY-DEFAULTS (siehe "MODIFY-DEFAULTS - Voreinstellungen verändern") beschrieben.

Abbruch des LMSCONV-Laufs durch Fehler

Die Fehlerbehandlung wird ebenfalls über die STXIT-Routine gesteuert.

Bei Programmbeendigung, Angabe von `/START-EXECUTABLE-PROGRAM`, `/LOAD-EXECUTABLE-PROGRAM`, `/CANCEL-JOB`, `/LOGOFF`, `/CANCEL-PROGRAM`, `/ABEND`, `/EXIT-JOB` oder bei Leitungsverlust wird sichergestellt, dass die Bibliotheken konsistent bleiben.

Für alle Fälle der Programmbeendigung gilt:

- Alle STXIT-Routinen in LMSCONV werden ausgeschaltet, um ein irreguläres Weiterarbeiten durch /INFORM-PROGRAM zu unterbinden.
- LMSCONV simuliert eine END-Anweisung. Dadurch werden alle offenen Bibliotheken geschlossen.

7.3.5 Platten ohne PAM-Schlüssel

Neue Plattenformate, insbesondere das Non-Key-Format (NK-Format), wurden eingeführt um die Speicherkapazität und die Transferraten der Plattenspeicher zu erhöhen. Eine NK-Platte wird ohne PAM-Schlüssel formatiert. Für die Nutzung von NK-Platten müssen Dateien mit PAM-Schlüssel in Dateien ohne PAM-Schlüssel konvertiert werden (PAM-key-Eliminierung).

In BS2000 werden nur noch Platten mit fester Blockgröße (2 KByte, 4 KByte...) bedient. Diese festen Blockgrößen verhindern die einfache Unterbringung der PAM-Schlüssel (PAM key). Aus diesem Grund ist der PAM-Schlüssel entfallen.

Für SAM-, ISAM- und UPAM-Dateien gibt es zwei unterschiedliche Dateiformate auf Platte: das Format mit PAM-Schlüssel (kurz K) und das Format ohne PAM-Schlüssel (kurz NK2 und NK4).

Das Dateiformat wird durch den BLKCTRL-Wert festgelegt. BLKCTRL kann den Wert PAMKEY, DATA, DATA2K, DATA4K oder NO annehmen. Einzelheiten zu den Dateiformaten siehe Handbuch „Einführung in das DVS“ [4].

Bibliotheksdateien

Die Unterscheidung zwischen K und NK ist zunächst eine DVS bedingte Unterscheidung. Sie überträgt sich wie folgt auf die interne Dateiorganisation der Bibliothek:

Der PAM-Schlüssel ist nicht notwendig. Dateiseitig ergibt sich dennoch eine Unterscheidung, die durch das Dateiattribut BLKCTRL repräsentiert wird.

Bibliotheken brauchen bei der Migration zwischen K-Welt und NK-Welt nicht mit PAMCONV umgesetzt zu werden.

Elementverarbeitung

Das nachfolgende Bild zeigt einen Überblick über die möglichen Situationen beim Transfer von Daten zwischen Datei und Bibliothekselementen. Für die Elemente sind die logischen Informationseinheiten aufgeführt; für die Dateien ist der BLKCTRL-Wert angegeben. Die Pfeile beschreiben die Transfer-Richtung.

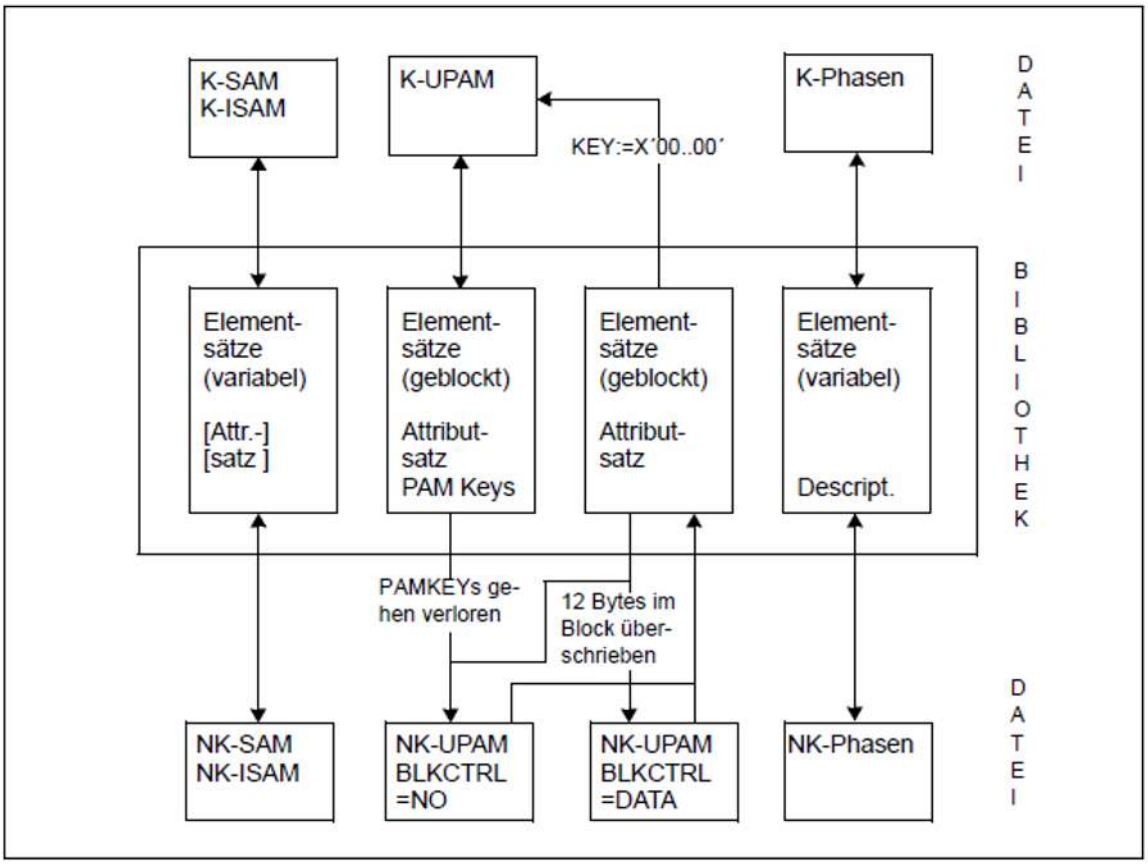


Bild 9: Transfer von Daten zwischen Datei und Bibliothekselementen

Verhalten bei der ADD-ELEMENT-Anweisung

Über die ADD-ELEMENT-Anweisung werden Datei-Inhalte in Elemente abgelegt. Im Einzelnen gilt für:

- SAM/ISAM-Dateien

  Bei Aufnahme von SAM- und ISAM-Dateien wird der BLKCTRL-Wert mit abgespeichert, wenn SOURCE-ATTRIBUTES=*KEEP gesetzt ist, d.h. die ursprüngliche, vom BLKCTRL-Wert geprägte Blockstruktur der Datei wird im Attributsatz dokumentiert.

  Die Daten werden über die logische Zugriffsmethode SAM/ISAM gelesen (Einzelsätze) und unverändert ins Element geschrieben (als Sätze variablen Formates).

  Die erzeugte Elementstruktur ist unabhängig von der ursprünglichen BLKCTRL-Eigenschaft.

- PAM-Dateien

  Bei Aufnahme von PAM-Dateien wird der BLKCTRL-Wert generell mit abgespeichert. Die Blöcke der Datei werden über die Zugriffsmethode UPAM gelesen und unverändert als Block im Element gespeichert. Sind PAM-Schlüssel vorhanden, d.h. BLKCTRL=PAMKEY, so werden diese PAM-Schlüssel beim Element abgelegt.

  Das erzeugte Element behält somit die vom BLKCTRL-Wert geprägte Blockstruktur bei.

- Phasen

  Bei Aufnahme von Phasen wird der BLKCTRL-Wert nicht abgespeichert. Die entsprechende Formatangabe ist in der Phaseninformation auf Datei hinterlegt. In der Bibliothek haben K-Phasen und NK-Phasen gleiches Format. Die PAM-Schlüssel-Informationen sind in Deskriptoren abgelegt.

ADD datei > element	Datei-Typ	BLKCTRL-Eintrag im Attributsatz	PAMKEY-Speicherung
Datei liegt auf NK-Platte	SAM/ISAM	--[1)]	
	SAM/ISAM	aus dem Katalog	nein
	UPAM	aus dem Katalog	nein
Datei liegt auf K-Platte	SAM/ISAM	--[1)]	
	SAM/ISAM	aus dem Katalog	nein
	UPAM	aus dem Katalog	für BLKCTRL=PAMKEY

Tabelle 7: BLKCTRL und PAMKEY bei ADD-ELEMENT

[1)] die Ablage kann über den Operanden SOURCE-ATTRIBUTES gesteuert werden

Verhalten bei der EXTRACT-ELEMENT-Anweisung

Über die EXTRACT-ELEMENT-Anweisung werden Elementinhalte in Dateien ausgegeben. Der BLKCTRL-Wert wird über folgende Hierarchie bestimmt:

1. Angabe im Katalogeintrag oder mit /ADD-FILE-LINK.
2. gespeicherter BLKCTRL-Wert beim Element. Er ist nur bei ursprünglichen PAM-Dateien relevant.
3. Einstellung des Systemparameters BLKCTRL=PAMKEY / NONKEY. Er kann mit /SHOW-SYSTEM-PARAMETERS angezeigt werden.
4. Platteneigenschaft PAMKEY oder NONKEY.

Ist kein Katalogeintrag vorhanden und der BLKCTRL-Wert nicht gespeichert, so entscheidet der Systemparameter und die Platteneigenschaft über den BLKCTRL-Wert:

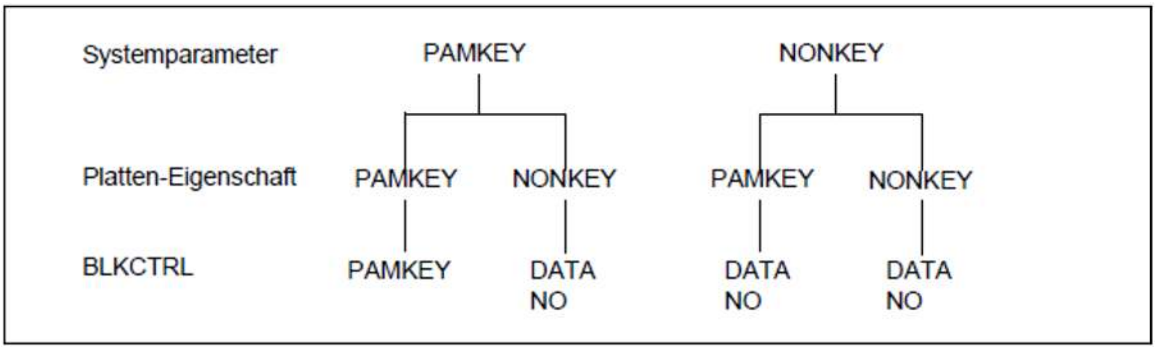


Bild 10: Zusammenhang zwischen Systemparameter und BLKCTRL-Wert

Ist der Systemparameter auf PAMKEY gesetzt, so lässt LMSCONV das System den BLKCTRL-Wert bestimmen, d. h. BLKCTRL ist nicht spezifiziert.

Ist der Systemparameter auf NONKEY gesetzt, so setzt LMSCONV für SAM- und ISAM-Dateien BLKCTRL=DATA und für PAM-Dateien BLKCTRL=NO.

Im Einzelnen gilt für:

- ISAM-Dateien

  Die variabel langen Elementsätze werden mit der logischen Zugriffsmethode ISAM geschrieben. Der BLKCTRL-Wert der Datei wird nach dem oben beschriebenen Algorithmus bestimmt, wobei allerdings Punkt 2 der Hierarchie entfällt, da der beim Element gespeicherte BLKCTRL-Wert nur dokumentarischen Charakter hat; er wird ignoriert.

- SAM-Dateien

  Bei BLKCTRL=DATA tritt ein DVS-Fehler auf, wenn im Element Sätze länger als 32Kbyte - 16Byte sind. In der K-Welt dürfen diese Sätze bis zu 32Kbyte - 4Byte lang sein. LMSCONV reicht beim Selektieren zu lange Sätze ungeprüft an DVS weiter. Der BLKCTRL-Wert wird wie bei ISAM ermittelt.

- PAM-Dateien

  In der NK-Welt gehen die PAM-Schlüssel verloren. Zusätzlich werden bei BLKCTRL=DATA die ersten 12 Bytes eines jeden logischen Blockes durch das System überschrieben. In beiden Fällen gibt LMSCONV eine Warnung aus.

- Phasen (Elemente vom Typ C)

  Phasen werden gesondert behandelt. Neben dem alten Phasenformat (K-Phase) gibt es ein neues Phasenformat ohne PAM-Schlüssel (NK-Phase) auf Dateiseite.

Zusammenfassung

- SAM/ISAM-Dateien

  Das Aufnehmen der Dateien geht immer; ebenso der Selektiervorgang.
  Ein evtl. gespeicherter BLKCTRL-Wert hat nur dokumentarischen Charakter.

  Die innere Dateiform wird stets durch die Zugriffsmethode SAM/ISAM bestimmt. Diese übernimmt auch die Konversion der Datensätze in das „innere“ Blockformat der Datei.

- UPAM-Dateien

  Eine evtl. notwendige Konversion der Daten kann nicht automatisch erfolgen (weder durch die Zugriffsmethode UPAM noch durch LMSCONV), da in diesen Fällen ein Datenverlust eintritt.

  Die Steuerung liegt letztendlich beim Benutzer.

Datei-Typ UPAM	Erzeugter / abgespeicherter BLKCTRL-Eintrag im Attributsatz			
	PAMKEY	DATA	NO	--
Datei liegt auf NK-Platte	[1] [2]	ADD EXTRACT	ADD EXTRACT	--- EXTRACT
Datei liegt auf K-Platte	ADD EXTRACT	ADD EXTRACT	ADD EXTRACT	--- EXTRACT

Tabelle 8: BLCTRL-Eintrag in Abhängigkeit des Plattenformates bei UPAM-Dateien

[1] der Wert BLKCTRL=PAMKEY ist nicht möglich

[2] der Selektiervorgang muss vom Benutzer gesteuert werden, z.B. über Angabe eines Dateikettungsnamen in der Anweisung.

7.3.6 NK4-Platten

In BS2000 gibt es zwei Formate von PLAM-Bibliotheken. Ein 2K-orientiertes Format (NK2-PLAM-Datei) und ein 4K-orientiertes Format (NK4-PLAM-Datei). Die Konvertierung zwischen den verschiedenen Formaten wird mit der LMSCONV-Anweisung COPY-LIBRARY vorgenommen. Der Anwender legt das jeweilige Format mit `/ADD-FILE-LINK ...,BUFFER-LENGTH=*STD(1` oder 2) fest. LMSCONV unterstützt beide Bibliotheksformate. Des weiteren unterstützt LMSCONV NK4-Platten mit den Anweisungen ADD-ELEMENT und EXTRACT-ELEMENT.

Aufnehmen von Dateien mit ADD-ELEMENT

Mit der Anweisung ADD-ELEMENT können Dateien beliebiger BUFFER-LENGTH in eine Bibliothek aufgenommen werden.

Ausgeben von Dateien mit EXTRACT-ELEMENT

Für die EXTRACT-ELEMENT-Anweisung sind folgende Fälle zu unterscheiden:

1. Das Element enthält einen Attributsatz mit der originalen BUFFER-LENGTH-Angabe (z.B. nach ADD-ELEMENT-Anweisung mit SOURCE-ATTR=*KEEP bzw. für ursprüngliche UPAM-Dateien, z.B. auch Bibliotheken).
   a. Für die Zieldatei ist ein BUFFER-LENGTH-Wert explizit vorgegeben, sei es in der TASK-FILE-TABLE (TFT) über /ADD-FILE-LINK oder direkt im Katalog. In diesem Fall wird stets diese Vorgabe herangezogen. Folgende Probleme können auftreten:
      - SAM/ISAM-Datei

        Die Elementsätze sind zu lang für die vorgegebene BUFFER-LENGTH. Dies führt zu einem DMS-Fehler.
      - UPAM-Datei

        Beim Erzeugen von UPAM-Dateien füllt LMSCONV einen logischen Block (bis auf den Letzten) dicht mit 2K-Einheiten auf und gibt ihn erst dann mit UPAM aus.

        Bei BLKCTRL=DATA fängt jeder logische Block (BUFFER-LENGTH) mit einem 12 Bytes langem Kontrollfeld (CF) an. Entspricht die angegebene BUFFER-LENGTH nicht der gespeicherten, so können Daten von DMS mit dem CF überschrieben werden. Die Datei ist damit unbrauchbar.

        Bei BLKCTRL=NO können aber auch unbrauchbare Dateien erzeugt werden, wenn die BUFFER-LENGTH verändert wird. (z.B. PLAM-Dateien).

        Deshalb wird generell bei unterschiedlichen BUFFER-LENGTH-Angaben (Benutzervorgabe versus gespeicherter Wert) von LMSCONV eine Warnung ausgegeben. Es wird jedoch immer versucht, die Datei zu erzeugen.

   b. Für die Zieldatei ist **kein** BUFFER-LENGTH-Wert explizit vorgegeben oder bekannt. In diesem Fall wird die Angabe aus dem Attributsatz benutzt.

      Ist n in STD(n) ungerade, erhöht LMSCONV auf n+1.
2. Das Element enthält **keinen** Attributsatz, z.B. für Phasen-Elemente.
   a. Für die Zieldatei ist ein BUFFER-LENGTH-Wert explizit vorgegeben. Vorgehensweise wie unter 1.a.

      Bei der Erzeugung von Phasen führen BUFFER-LENGTH-Angaben ungleich STD(1) oder STD(2) auf Fehler.
   b. Für die Zieldatei ist **kein** BUFFER-LENGTH-Wert explizit vorgegeben oder bekannt.
      - Für Phasen ergibt sich die BUFFER-LENGTH aus der aktuellen Umgebung, d.h. auf NK2-Platten BUFFER-LENGTH=STD(1) und auf NK4-Platten BUFFER-LENGTH=STD(2). Inhaltlich unterscheiden sich die Phasen nicht.

- Ansonsten wird die BUFFER-LENGTH anhand der maximalen Satzlänge berechnet.

Zusammenfassend wird folgende Vorgehensweise empfohlen, wenn Dateien über eine Bibliothek auf eine NK4-Platte gebracht werden sollen:

1. Aktionen auf der NK2-Platte:
   Alle „kritischen“ Elemente der Bibliothek als Datei extrahieren. Das sind „PAM“-Elemente unter Typ X, die als Datei
   - BUFFER-LENGTH=STD(n) und n ungerade haben oder
   - Pamkey-behaftete Phasen oder
   - 2K-orientierte PLAM-Dateien sind.
2. Alle Dateien mit ungerader BUFFER-LENGTH (außer PLAM-Dateien) mit PAMCONV in NK4-Dateien konvertieren.
3. Alle Pamkey-behafteten Phasen mit PAMCONV in NK-Phasen konvertieren.
4. NK2-PLAM-Dateien mit der LMSCONV-Anweisung COPY-LIBRARY in NK4-PLAM-Dateien konvertieren.
5. Danach die NK4-Dateien mit der Anweisung ADD-ELEMENT in eine NK4-PLAM-Datei aufnehmen und diese auf die NK4-Platte transferieren.

7.3.7 Auswirkungen von ACS

Mit dem Subsystem ACS (Alias Catalog System) können Aliasnamen für Dateien verwaltet werden. Im Folgenden wird beschrieben, wie LMSCONV die ACS-Aliasnamen behandelt.

Konstruktion von Elementnamen

Für die Konstruktion von Elementnamen wird immer der bereits von ACS umgesetzte Dateiname herangezogen.

Beispiel

Alias X wird zu Dateiname FILE.X

LMSCONV-Anweisung `ADD-ELEMENT  X,(,*BY-SOURCE(001),S)` erzeugt das Element S/FILE.X/001

Konstruktion von Dateinamen

Für die Konstruktion von Dateinamen wird immer der aktuelle Elementname herangezogen, der dann ggf. von ACS umgesetzt wird.

Beispiel

Alias X wird zu Dateiname FILE.X

LMSCONV-Anweisung `EXTRACT-ELEMENT  (,X,S),*BY-SOURCE` erzeugt die Datei FILE.X

Protokollierung von Dateinamen

LMSCONV protokolliert immer die vollständigen umgesetzten Dateinamen.

7.4 Anweisungen

- Übersicht über die Anweisungen von LMSCONV
- LMSCONV-Anweisungen von ADD-ELEMENT bis MODIFY-ELEMENT
  - ADD-ELEMENT - Element in Bibliothek aufnehmen
  - CLOSE-LIBRARY - Bibliothek schließen
  - COPY-ELEMENT - Element kopieren
  - COPY-LIBRARY - Bibliothek kopieren
  - DELETE-ELEMENT - Ein Element logisch löschen
  - END - LMSCONV beenden
  - EXTRACT-ELEMENT - Elemente in Datei ausgeben
  - MODIFY-DEFAULTS - Voreinstellungen verändern
  - MODIFY-ELEMENT - Element verändern
- Subanweisungen von MODIFY-ELEMENT für Elemente vom Basistyp R, C und L
  - ADD-REP-RECORD - REP-Sätze in Bindemodul einfügen
  - ADD-TEXT-MODIFICATION - Textsätze eines Bindemoduls korrigieren
  - DELETE-RECORD-TYPE - Satzarten aus Eingabeelement auschließen
  - END-MODIFY - Eingabe von Subanweisungen beenden
  - MODIFY-CSECT-ATTRIBUTES - Programmabschnittsmerkmale verändern
  - MODIFY-MODIFICATION-DEFAULTS - Globale Voreinstellungen festlegen
  - REMOVE-MODIFICATION - Korrekturen rückgängig machen
  - RENAME-SYMBOLS - Symbole umbenennen
- Subanweisungen von MODIFY-ELEMENT für Textelemente
  - ADD-RECORD - Sätze einfügen
  - END-MODIFY - Subanweisungen abschließen
  - REMOVE-RECORD - Satz oder Satzbereich in Element löschen
- LMSCONV-Anweisungen von MODIFY-ELEMENT-ATTRIBUTES bis WRITE-COMMENT
  - MODIFY-ELEMENT-ATTRIBUTES - Elementattribute verändern
  - MODIFY-LOGGING-PARAMETERS - Protokolleinstellungen ändern
  - OPEN-LIBRARY - Globale Bibliothek öffnen
  - SHOW-DEFAULTS - Aktuelle Belegung der Voreinstellungen ausgeben
  - SHOW-ELEMENT - Elementinhalt anzeigen
  - SHOW-ELEMENT-ATTRIBUTES - Elementeigenschaften anzeigen
  - SHOW-LIBRARY-ATTRIBUTES - Attribute einer Bibliothek anzeigen
  - SHOW-LIBRARY-STATUS - Status der Bibliothek anzeigen
  - SHOW-LOGGING-PARAMETERS - Globale LMSCONV-Parameter anzeigen
  - SHOW-TYPE-ATTRIBUTES - Attribute eines Elementtyps anzeigen
  - SHOW-USER-EXITS - LMSCONV-Version anzeigen
  - WRITE-COMMENT - Kommentare in Ausgabemedium schreiben

7.4.1 Übersicht über die Anweisungen von LMSCONV

Die Syntax der SDF-Anweisungssprache ist im Handbuch „Kommandos“ [1] beschrieben. Folgende abkürzende Schreibweisen werden verwendet:

cat-id	cat
completion	compl
generation	gen
lower-case	low
underscore	under
userid	user
version	vers
wildcards	wild

Das Schlüsselwort *DEFAULT wird bei den einzelnen Anweisungen nicht mehr beschrieben. Es bedeutet immer den mit der Anweisung MODIFY-DEFAULTS eingestellten Wert.

Für den Elementtyp gilt zusätzlich:

Der Elementtyp ist mit *UNDEFINED voreingestellt. Deshalb muss zuerst mit der Anweisung MODIFY-DEFAULTS ein Elementtyp vereinbart werden, da sonst die Angabe *DEFAULT zu einem Fehler führt.

Eingaberegeln

Die LMSCONV-Anweisungen werden über die SDF-Benutzeroberfläche gelesen und vom Kommandoprozessor SDF (System Dialog Facility) verarbeitet. Damit gibt es verschiedene Formen des geführten oder ungeführten Dialogs mit der Möglichkeit, Hilfemenüs zu den Anweisungen anzufordern. Siehe dazu das Handbuch „Dialogschnittstelle SDF“ [20].

Folgezeilen

> Anweisungen können sich über mehrere Sätze erstrecken. Für die Trennung gelten die Konventionen der BS2000-Kommandosprache. Als Trennzeichen wird ein Bindestrich (-) verwendet. Anweisungszeilen können maximal 16364 Zeichen lang sein.

Abkürzungsmöglichkeiten

> Für die Eingabe der LMSCONV-Anweisungen gilt, dass Anweisungsnamen, Operandennamen und Schlüsselwörter abgekürzt werden dürfen.
>
> Es gilt folgende Regel:
> Es kann jeweils von rechts nach links bis zur Eindeutigkeit abgekürzt werden. Dies betrifft sowohl den ganzen Namen als auch Teilnamen (beginnend mit einem Bindestrich) und schließt ein, dass evtl. der Teilname vollständig weggelassen werden kann.
>
> Die garantierten Abkürzungsmöglichkeiten aller Anweisungen, Operanden und Operandenwerte sind in den Syntaxbeschreibungen der Anweisungen durch halbfette Schreibweise gekennzeichnet. Es können aber auch kürzere Angaben (Abkürzungen bis zur Eindeutigkeit innerhalb einer Struktur) gemacht werden.

In Prozeduren sollten keine oder garantierte Abkürzungen verwendet werden.

Stellungsoperanden

SDF erlaubt die wahlweise Angabe von Operanden als Schlüsselwort- oder als Stellungsoperanden. Es kann jedoch nicht völlig ausgeschlossen werden, dass sich bei einem Versionswechsel eine Operandenposition ändert. Es wird daher empfohlen, in Prozeduren Stellungsoperanden zu vermeiden.

7.4.2 LMSCONV-Anweisungen von ADD-ELEMENT bis MODIFY-ELEMENT

- ADD-ELEMENT - Element in Bibliothek aufnehmen
- CLOSE-LIBRARY - Bibliothek schließen
- COPY-ELEMENT - Element kopieren
- COPY-LIBRARY - Bibliothek kopieren
- DELETE-ELEMENT - Ein Element logisch löschen
- END - LMSCONV beenden
- EXTRACT-ELEMENT - Elemente in Datei ausgeben
- MODIFY-DEFAULTS - Voreinstellungen verändern
- MODIFY-ELEMENT - Element verändern

7.4.2.1 ADD-ELEMENT - Element in Bibliothek aufnehmen

ADD-ELEMENT nimmt Dateien als Elemente in eine Bibliothek auf. Die Elementdaten werden standardmäßig von SYSDTA gelesen. Sie können aber auch von einer explizit angegebenen Datei oder *OMF gelesen werden. Die Dateien werden immer ohne Präfix, d.h. ohne Catid und Userid, als Element in eine Bibliothek aufgenommen, es sei denn, der Benutzer hat in der Konstruktionsangabe explizit einen Präfix angegeben.

Dateien, die mit RECORD-FORMAT=*UNDEFINED katalogisiert sind, können ebenfalls in Bibliotheken aufgenommen werden. Dateien, die RECORD-FORMAT=*FIXED haben, können nur über SOURCE-ATTRIBUTES=*KEEP abgespeichert werden. Die Satzformate *FIXED und *UNDEFINED werden in das Satzformat VARIABLE umgewandelt; d.h. mit einem 4-Byte-Satzkopf versehen. Die Satzlänge inklusive Satzkopf darf 32 kByte nicht überschreiten.

Dateigenerationsgruppen können nur über Linknamen und eine LMSCONV-gültige Elementbezeichnung aufgenommen werden.

LMSCONV übernimmt bei der ADD-ELEMENT-Anweisung das Katalogattribut CCS der Datei als Elementattribut. Werden die Daten von SYSDTA gelesen, so erhält das erzeugte Element als Attribut den für SYSDTA eingestellten CCS-Namen. Beim Lesen von *OMF wird den Elementen „kein Code“ zugeordnet.

Format

ADD-ELEMENT

FROM-FILE = **<u>*STD</u>** / ***SYSDTA**(...) / ***ALL** / <filename 1..80 without-vers with-wild> / ***LINK**(...) / ***OMF**

***SYSDTA**(...)

| **END** = '***END**' / <c-string 1..8>

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,**TO-E**LEMENT = **<u>*LIB</u>** <u>RARY</u> **<u>-ELEM</u>** <u>ENT</u> (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = **<u>*STD</u>** / ***LINK**(...) / <filename 1..54 without-vers>

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = **<u>*BY-SOUR</u>** <u>CE</u> (...) / <composed-name 1..132 with-under with-wildcard-constr>(...)

| **<u>*BY-SOURCE</u>** (...)

| | **VERSION** = **<u>*DEFAULT</u>** / ***UP**PER**-LIM**IT / <composed-name 1..24 with-under>

| <composed-name 1..132 with-under with-wildcard-constr>(...)

| | **VERSION** = **<u>*DEFAULT</u>** / ***UP**PER**-LIM**IT / <composed-name 1..24 with-under>

| ,**TYPE** = ***DEF**AULT / <alphanum-name 1..8>

| ,**USER-DATE** = **<u>*TODAY</u>** / ***BY_SOURCE** / <date 8..10 with-compl>

,ELEMENT**-ATTR**IBUTES = **<u>*DEFAULT</u>** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **SOUR**CE**-ATTR**IBUTES = **<u>*DEFAULT</u>** / ***STD** / ***IGNORE** / ***KEEP**(...)

| ***KEEP**(...)

| | **KEEP-TYPES** = **<u>*DEFAULT</u>** / ***STD** / ***ALL**

,DELETE**-SOUR**CE = **<u>*DEFAULT</u>** / ***NO** / ***Y**ES

,**WR**ITE**-MODE** = **<u>*DEFAULT</u>** / ***CREATE** / ***REPLACE** / ***EXTEND** / ***ANY**

,**DIALOG-CONTR**OL = **<u>*DEFAULT</u>** / ***NO** / ***Y**ES / ***ERROR**

Operanden

FROM-FILE = <u>*STD</u> / * SYSDTA(...) / *ALL / <filename 1..80 without-vers with-wild> / *LINK (...) / *OMF
Angabe derjenigen Datei, die als Element in die Bibliothek aufgenommen werden soll.

FROM-FILE = <u>*STD</u>
Standardmäßig werden die Datensätze von der Systemdatei SYSDTA gelesen.Zulässige Elementtypen: S, M, P, J, D, X

FROM-FILE = *SYSDTA(...)
Die Datensätze werden mit RDATA von der Systemdatei SYSDTA gelesen. Die Datensätze müssen direkt auf die ADD-ELEMENT-Anweisung folgen.
Zulässige Elementtypen: S, M, P, J, D, X, R

END = '*END' / <c-string 1..8>
Endekriterium für die Eingabe. Die Folge der Datensätze muss mit '*END' oder einem selbst definierten Endekriterium abgeschlossen werden. Fehlt das Endekriterium in den Eingabedaten, wird bis EOF gelesen.

> Werden Datensätze von der Systemdatei SYSDTA=(SYSCMD) gelesen, so dürfen sie nicht mit dem Zeichen „/“ beginnen. Der Grund dafür ist, dass der RDATA-Makro solche Sätze als Kommandos interpretiert und daher den Return-Code für EOF übergibt. Es ist damit nicht möglich, System-Kommandos als Datensätze zu übergeben.

FROM-FILE = <filename 1..80 without-vers with-wild >
Die Daten werden aus der angegebenen Datei gelesen.
Zulässige Elementtypen: S, M, P, J, D, X, R, C
Dateien vom Filetyp PAM können nur als Elemente unter dem Elementtyp X abgelegt werden.

FROM-FILE = *LINK(...)
Die Daten werden aus der über den Dateikettungsnamen angegebenen Datei gelesen.

LINK-NAME = <structured-name 1..8>
Dateikettungsname, der auf die Datei verweist.

FROM-FILE = *OMF
Gilt nur für Typ R Elemente.
Die Daten werden aus der OMF-Datei gelesen. Es werden alle Module aus der OMF-Datei übernommen. Sind im

EAM-Bereich mehrere gleichnamige Module enthalten, übernimmt LMSCONV den zuletzt übersetzten Modul in die Bibliothek.

TO-ELEMENT = *LIBRARY-ELEMENT(...)
Angabe, wohin und unter welchem Namen das Element aufgenommen werden soll.

LIBRARY = *STD / *LINK(...) / <filename 1..54 without-vers>
Angabe der Bibliothek, in die das Element aufgenommen werden soll.

LIBRARY =*STD
Die durch OPEN-LIBRARY global eröffnete Bibliothek.

LIBRARY = *LINK(...)
Die über einen Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in die die Datei als Element aufgenommen werden soll.

ELEMENT = *BY-SOURCE(...) / <composed-name 1..132 with-under with-wildcard-constr>(...)
Name, den das neu aufzunehmende Element erhalten soll. Eine Konstruktionsangabe bezieht sich auf den Dateinamen.

ELEMENT = *BY-SOURCE(...)
Der Elementname entspricht dem Dateinamen bzw. dem Modulnamen bei *OMF.

VERSION = *DEFAULT / *UPPER-LIMIT / <composed-name 1..24 with-under>
Version, die das neu aufzunehmende Element erhalten soll.

VERSION = *DEFAULT
Der Standardwert ist *BY-SOURCE oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

VERSION = *UPPER-LIMIT
Die höchstmögliche Version X'FF' wird erzeugt.

VERSION = <composed-name 1..24 with-under>
Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

ELEMENT = <composed-name 1..132 with-under with-wildcard-constr>(...)
Das Element wird mit dem hier angegebenen Namen abgelegt.

VERSION = *DEFAULT / *UPPER-LIMIT / <composed-name 1..24 with-under>
Version, die das neu aufzunehmende Element erhalten soll.
Beschreibung der Operanden siehe oben.

TYPE = *DEFAULT / <alphanum-name 1..8>
Typ, den das neu aufzunehmende Element erhalten soll. Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *TODAY / *BY-SOURCE / <date 8..10 with-compl>
Datum, das vom Benutzer vergeben wird.

USER-DATE = *TODAY
Das aktuelle Tagesdatum wird vergeben.

USER-DATE = *BY-SOURCE
Das Änderungsdatum der Datei (CHANGE-DATE) wird übernommen.

USER-DATE = <date 8..10 with-compl>
Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

ELEMENT-ATTRIBUTES = *DEFAULT / *PARAMETERS(...)
Bestimmt, ob die Dateieigenschaften und auch der ISAM-Schlüssel in das Ausgabeelement mit übernommen werden.

ELEMENT-ATTRIBUTES = *DEFAULT
Es werden die Elementattribute TYPE=*UNDEFINED, VERSION=*BY-SOURCE und SOURCE-ATTRIBUTES=*IGNORE bzw. die mit MODIFY-DEFAULTS aktuell eingestellten gesetzt.

ELEMENT-ATTRIBUTES = *PARAMETERS(...)

SOURCE-ATTRIBUTES = *DEFAULT / *STD / *IGNORE / *KEEP(...)
Speichern von Dateiattributen. Dieser Operand ist wirkungslos, wenn die Daten von SYSDTA oder *OMF gelesen werden. Es werden keine Ursprungsattribute gespeichert. Werden die Daten aus einer Datei vom Filetyp UPAM gelesen, so ist die Angabe wirkungslos; es wird immer wie im Fall *KEEP verfahren.

SOURCE-ATTRIBUTES = *DEFAULT
Der Standardwert ist *STD (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

SOURCE-ATTRIBUTES = *STD
Es werden keine Dateiattribute und auch kein ISAM-Schlüssel gespeichert. In dem Fall ist es nur möglich, ISAM-Dateien mit KEY-POSITION=5, KEY-LENGTH <= 16 und RECORD-FORMAT=*VARIABLE in das Element aufzunehmen.
Für ISAM-Dateien wird eine Warnung ausgegeben, dass die ISAM-Schlüssel nicht aufgenommen wurden.

SOURCE-ATTRIBUTES = *IGNORE
Wie SOURCE-ATTRIBUTES=*STD, es wird jedoch keine Warnung ausgegeben.

SOURCE-ATTRIBUTES = *KEEP(...)
Folgende Dateiattribute werden unverändert im neu aufzunehmenden Element gespeichert: ACCESS-METHOD, RECORD-FORMAT, RECORD-SIZE, BUFFER-LENGTH, PERFORMANCE, USAGE, ACCESS und USER-ACCESS. Ist ACCESS-METHOD=ISAM, so werden PADDING-FACTOR, LOGICAL-FLAG-LENGTH, VALUE-FLAG-LENGTH, PROPAGATE-VALUE-FLAG und auch die ISAM-Schlüssel sowie Informationen über ISAM-Sekundärschlüssel zusätzlich gespeichert. Für SAM-Node-Files wird ferner der Zeichensatz auf Net-Storage (NETCCSN) gespeichert.

KEEP-TYPES = *DEFAULT / *STD / *ALL
Bestimmt die Elementtypen, für die Dateiattribute gespeichert werden sollen.

KEEP-TYPES = *DEFAULT
Der Standardwert ist *STD (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

KEEP-TYPES = *STD
Attribute werden für Elemente der Basistypen S, M, P, D, J, X gespeichert.

KEEP-TYPES = *ALL
Attribute werden für Elemente aller Basistypen gespeichert, die für ADD-ELEMENT zulässig sind.

> ¡ LMSCONV speichert den Last Byte Pointer (LBP) zusatzlich zu den bisher schon gespeicherten Dateiattributen

- für PAM-Elemente vom Typ X
- für Elemente vom Typ C, wenn KEEP-TYPES = *ALL angegeben wurde

DELETE-SOURCE = *DEFAULT / *NO / *YES
Hier kann der Benutzer bestimmen, ob die ursprüngliche Datei erhalten oder gelöscht werden soll. Dieser Operand ist wirkungslos, wenn die Daten von SYSDTA oder *OMF gelesen werden.

DELETE-SOURCE = *DEFAULT
Der Standardwert ist *NO (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DELETE-SOURCE = *NO
Die ursprüngliche Datei wird nicht gelöscht.

DELETE-SOURCE = *YES
Die ursprüngliche Datei wird gelöscht.

WRITE-MODE = *DEFAULT / *CREATE / *REPALCE / *EXTEND / *ANY
Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

WRITE-MODE = *DEFAULT
Der Standardwert ist *CREATE (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

WRITE-MODE = *CREATE
Das Zielelement darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Ein Element wird nur dann überschrieben, wenn schon ein Element gleichen Namens vorhanden ist. Sonsten wird ADD-ELEMENT mit einer Fehlermeldung abgewiesen.

WRITE-MODE = *EXTEND
Ein Element wird jedoch nur dann erweitert, wenn im Element keine ISAM-Schlüssel gespeichert sind und die im Element gespeicherten Dateimerkmale ACCESS-METHOD, RECORD-FORMAT und RECORD-SIZE mit den Merkmalen der Datei übereinstimmen. Sonst wird ADD-ELEMENT mit einer Fehlermeldung zurückgewiesen. *EXTEND ist nicht bei der Eingabe von SYSDTA erlaubt.

WRITE-MODE = *ANY
Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung siehe die MODIFY-DEFAULTS Anweisung.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist *NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES / *ERROR
Siehe die Beschreibung bei der Anweisung MODIFY-DEFAULTS.

Hinweise

- Wird SOURCE-ATTRIBUTES=*KEEP angegeben, ist zu beachten: Durch eventuell vorhandene ISAM-Schlüssel kann die nachfolgende Verarbeitung wie Übersetzen und die Ausführung von /CALL-PROCEDURE behindert werden. Dieser Parameterwert eignet sich vor allem zur Archivierung.
- Bei Aufnahme von temporären Dateien mit Wildcards ist keine Konstruktionsangabe des Zielelementnamens erlaubt, d.h. ELEM darf nur *BY-SOURCE sein.
- Bei Aufnahme von temporären Dateien mit ELEM=*BY-SOURCE erhält das Element den internen Dateinamen. Dieses Element kann unter einer anderen TASK ohne explizite Angabe eines Dateinamens nicht wieder in eine Datei ausgegeben werden.

Beispiel

Aufnehmen eines Elementes

Die Datei testelem wird in die Bibliothek LIB1 unter dem gleichen Namen als Element aufgenommen. Die Typangabe muss hier explizit in der ADD-ELEMENT-Anweisung angegeben werden, da standardmäßig der TYP mit *UNDEFINED voreingestellt ist.

```
/start-lmsconv
//open-library lib1,*update
//add-element from-file=testelem,to-elem=(type=d)
. . .
```

7.4.2.2 CLOSE-LIBRARY - Bibliothek schließen

Diese Anweisung schließt die angegebene(n) Bibliothek(en). Wenn diese Anweisung für eine globale Bibliothek eingegeben wird, dann wird diese Bibliothek geschlossen und der Name der Bibliothek wird zurückgesetzt.

Liegt die LMSCONV-Ausgabe auf einem Bibliothekselement, so wird die zugehörige Bibliothek nicht geschlossen und eine Fehlermeldung wird ausgegeben.

Wird diese Anweisung ohne Parameter angegeben, werden alle offenen Bibliotheken geschlossen und der Name der globalen Bibliothek wird zurückgesetzt.

Format

CLOSE-LIBRARY

LIBRARY = **<u>*ALL</u>** / ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

Operanden

LIBRARY = <u>*ALL</u> / *STD / <filename 1..54 without-vers> / *LINK (...)
Angabe derjenigen Bibliothek bzw. Bibliotheken, die geschlossen werden sollen.

LIBRARY = <u>*ALL</u>
Alle offenen Bibliotheken werden geschlossen.

LIBRARY = *STD
Die durch OPEN-LIBRARY geöffnete Bibliothek wird geschlossen.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, die geschlossen werden soll.

LIBRARY = *LINK(...)
Die über einen Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der vor dem LMSCONV-Lauf mit /ADD-FILE-LINK vereinbart wurde.

Beispiele

Alle offenen Bibliotheken werden geschlossen und der Name der globalen Bibliothek wird zurückgesetzt.

```
//close-library
```

Die Bibliothek lib1 wird geschlossen.

```
//close-library library=lib1
```

Die Bibliothek, die über den Dateikettungsnamen lib2 zugewiesen wurde, wird geschlossen.

```
//close-library library=*link(link-name=lib2)
```

7.4.2.3 COPY-ELEMENT - Element kopieren

COPY-ELEMENT kopiert Elemente und Bibliotheken im Verhältnis eins zu eins. Die kopierten Elemente können neue Elementbezeichnungen erhalten.

Es gibt folgende Kopiermöglichkeiten:

- Kopieren von einem oder mehreren Elementen in die gleiche Bibliothek
- Kopieren von einem oder mehreren Elementen in eine andere Bibliothek
- Kopieren einer kompletten Bibliothek

Kopieren mit WRITE-MODE=*SUBSTITUTE

Mit WRITE-MODE=*SUBSTITUTE kann erreicht werden, dass das kopierte Element in der Zielbibliothek das einzige Element mit gleichem Typ und Namen ist. LMSCONV löscht vor dem Kopieren in der Zielbibliothek alle Elemente mit dem Typ und dem Namen des Zielelementes.

Einschränkungen

- Eingabe- und Ausgabebibliothek müssen unterschiedlich sein.
- Tritt während des Löschens ein Fehler auf, so wird die Anweisung abgebrochen.

Format

COPY-ELEMENT

ELEMENT = ***LIB**RARY**-ELEM**ENT (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEFAULT** / ***ALL** / <alphanum-name 1..8 with-wild(20)>

| ,**USER-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**CRE**ATION**-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**MOD**IFICATION**-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

```
,TO-ELEMENT = *LIBRARY-ELEMENT (...)

    *LIBRARY-ELEMENT(...)

        |   LIBRARY = *STD / *BY-SOURCE / <filename 1..54 without-vers> / *LINK(...)

        |       *LINK(...)

        |           |   LINK-NAME = <structured-name 1..8>

        |   ,ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wildcard-constr>(...)

        |       *BY-SOURCE(...)

        |           |   VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /

        |           |       <composed-name 1..52 with-under with-wildcard-constr>

        |       <composed-name 1..132 with-under with-wildcard-constr>(...)

        |           |   VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /

        |           |       <composed-name 1..52 with-under with-wildcard-constr>

        |   ,TYPE = *BY-SOURCE / *DEFAULT / <alphanum-name 1..20 with-wildcard-constr>

        |   ,USER-DATE = *BY-SOURCE / *TODAY / <date 8..10 with-compl>

,WRITE-MODE = *DEFAULT / *CREATE / *REPLACE / *EXTEND / *SUBSTITUTE / *ANY

,DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
```

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich die zu kopierenden Elemente befinden.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, aus der die Elemente kopiert werden sollen.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL (...) / <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elementes, welches kopiert werden soll.

VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT /<composed-name 1..24 with-under with-wild(52)>
Version, die das zu kopierende Element besitzt.

VERSION = *HIGHEST-EXISTING
Das Element mit der höchsten bestehenden Version wird kopiert.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird kopiert.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Typ, den das zu kopierende Element besitzt. Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das zu kopierende Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird kopiert.

USER-DATE = <date 8..10 with-compl>
Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird kopiert.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden kopiert.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Erzeugungsdatum des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

TO-ELEMENT = *LIBRARY-ELEMENT(...)
Angabe, wohin und unter welchem Namen das Element kopiert werden soll.

LIBRARY = *STD / *BY-SOURCE / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in die das Element kopiert werden soll.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = *BY-SOURCE
Das Element wird in die Bibliothek kopiert, in der sich auch das zu kopierende Element befindet.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in die die Datei als Element kopiert werden soll. Existiert die Bibliothek noch nicht, wird sie neu eingerichtet.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *BY-SOURCE(...) /<composed-name 1..132 with-under with-wildcard-constr>(...)
Name, den das Zielelement erhalten soll.

ELEMENT = *BY-SOURCE(...)
Das Zielelement erhält den Namen des Quellelementes.

VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /<composed-name 1..52 with-under with-wildcard-constr>
Version, die das Zielelement erhalten soll.

VERSION = *DEFAULT
Der Standardwert ist *BY-SOURCE oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

VERSION = *BY-SOURCE
Das Zielelement erhält die Version des Quellelementes.

VERSION = *UPPER-LIMIT
Die höchstmögliche Version X'FF' wird erzeugt.

VERSION = <composed-name 1..52 with-under with-wildcard-constr>
Das Zielelement erhält die hier angegebene Version.

ELEMENT = <composed-name 1..132 with-under with-wildcard-constr>(...)
Name des übertragenen Elementes. Er kann auch mit Wildcards eingegeben werden.

VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /<composed-name 1..52 with-under with-wildcard-constr>
Version, die das Zielelement erhalten soll.
Beschreibung der Operanden siehe oben.

TYPE = *BY-SOURCE / *DEFAULT / <alphanum-name 1..20 with-wildcard-constr>
Typ, den das neu aufzunehmende Element erhalten soll.

TYPE = *BY-SOURCE
Das Zielelement erhält die gleiche Typbezeichnung wie das Quellelement.

TYPE = *DEFAULT
Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *BY-SOURCE / *TODAY / <date 8..10 with-compl>
Datum, das vom Benutzer vergeben wird.

USER-DATE = *BY-SOURCE
Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

USER-DATE = *TODAY
Das aktuelle Tagesdatum wird vergeben.

USER-DATE = <date 8..10 with-compl>
Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

WRITE-MODE = *DEFAULT / *CREATE / *REPLACE / *EXTEND / *SUBSTITUTE / *ANY
Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

WRITE-MODE = *DEFAULT
Der Standardwert ist *CREATE (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

WRITE-MODE = *CREATE
Das Zielelement darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Das Zielelement muss bereits existieren und wird dann ersetzt.

WRITE-MODE = *EXTEND
Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt.

WRITE-MODE = *SUBSTITUTE
Alle Elemente mit Typ und Namen des Zielelementes werden in der Zielbibliothek gelöscht. Danach wird das Quellelement kopiert.

WRITE-MODE = *ANY
Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-DEFAULTS.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist *NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES / *ERROR
Siehe die Beschreibung bei der Anweisung MODIFY-DEFAULTS.

Beispiele

In der Eingabebibliothek X ist unter Typ S das Element A/1 enthalten. In der Ausgabebibliothek Y ist unter Typ S das Element A/2 enthalten. Dann ist nach

```
//COPY-ELEM  ELEMENT= *LIB(LIB=X,ELEM=A,TYPE=S),-
             TO-ELEMENT= *LIB(LIB=Y), WRITE-MODE=*SUBSTITUTE
```

in der Ausgabebibliothek unter Typ S und Namen A nur das Element A/1 vorhanden. Das Element A/2 wurde gelöscht.

In der Eingabebibliothek X befinden sich alle Elemente einer Produktversion. Diese Elemente sollen in eine existierende Ausgabebibliothek Y so kopiert werden, dass Y nach dem Kopieren nur die kopierte Produktversion enthält und sonst keine andere Version. Das erreicht man mit

```
//COPY-ELEM  ELEMENT= *LIB(LIB=X,ELEM=*,TYPE=*),-
             TO-ELEMENT= *LIB(LIB=Y), WRITE-MODE=*SUBSTITUTE
```

Kopieren einer kompletten Bibliothek

Die Bibliothek lib1 wird komplett kopiert und erhält den Namen lib2. Durch die Angabe von '*' bei Element, Version und Typ ist keine Kenntnis über die enthaltenen Elemente erforderlich, d.h. alle Elemente werden im Verhältnis eins zu eins in die Bibliothek lib2 kopiert.

```
//COPY-ELEM  ELEMENT= *LIB(LIB=lib1,ELEM=*,VERSION=*,TYPE=*),-
             TO-ELEMENT= *LIB(LIB=lib2)
```

7.4.2.4 COPY-LIBRARY - Bibliothek kopieren

Die COPY-LIBRARY-Anweisung kopiert eine Bibliothek komplett mit allen Bibliotheks-, Typ- und Elementattributen. Die Zielbibliothek darf nicht existieren oder muss FILE-STRUCTURE=NONE haben. Die Zielbibliothek erhält das Bibliotheksformat entsprechend deren Wert für BUFFER-LENGTH. Die Anweisung ist damit zum Konvertieren des Bibliotheksformats geeignet.

Die Dateischutzattribute der Quellbibliothek können auf die Zielbibliothek übernommen werden. Die Anweisung ist damit zum Reorganisieren von Bibliotheken geeignet. Die Zielbibliothek ist logisch mit dem Original identisch und belegt nur noch den minimal notwendigen Plattenplatz.

Tritt während der COPY-LIBRARY-Verarbeitung ein Fehler auf (z.B. unzureichender Plattenplatz), ist die Zielbibliothek nicht vollständig.

Format

COPY-LIBRARY

LIBRARY = <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,TO-LIBRARY = <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,FILE-**ATTR**IBUTES = **<u>*STD</u>** / ***BY-SOUR**CE

Operanden

LIBRARY = <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, die kopiert werden soll.

LIBRARY = <filename 1..54 without-vers>
Die Bibliothek mit dem hier angegebenen Namen wird kopiert.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek wird kopiert.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der mit /ADD-FILE-LINK vereinbart wurde.

TO-LIBRARY = <filename 1..54 without-vers> / *LINK(...)
Angabe der Ziel-Bibliothek.

TO-LIBRARY = <filename 1..54 without-vers>
Die Bibliothek mit dem hier angegebenen Namen wird erstellt.

TO-LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek wird erstellt.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der mit /ADD-FILE-LINK vereinbart wurde.

FILE-ATTRIBUTES = *STD / *BY-SOURCE
Attribute der Ziel-Bibliotheksdatei.

FILE-ATTRIBUTES = *STD
Die Dateiattribute der Zielbibliothek werden nicht verändert. Neue Dateien werden mit den vom Dateiverwaltungssystem festgelegten Default-Werten erzeugt.

FILE-ATTRIBUTES = *BY-SOURCE
Die Dateischutzattribute der Quellbibliothek werden auf die Zielbibliothek übernommen (analog /COPY-FILE .., PROTECTION=*SAME).

Beispiele

Kopieren einer Bibliothek auf einen NK4-Pubset

```
/start-lmsconv
//copy-library library=lib,to-library=:nk4:lib
//end
```

Vorab erstellen des NK4-Bibliothekformates

```
/add-file-link file-name=nk4lib,link-name=nk4,buffer-length=*std(2)
/start-lmsconv
//copy-library library=nk2lib,to-library=*link(nk4)
//end
```

Reorganisation einer Bibliothek mit Zwischenspeicher

```
/delete-file file-name=tolib
/start-lmsconv
//copy-library library=lib,to-library=tolib,file-attributes=*by-source
//end
/copy-file from-file=tolib,to-file=lib
/delete-file file-name=tolib
```

7.4.2.5 DELETE-ELEMENT - Ein Element logisch löschen

Die Anweisung DELETE-ELEMENT löscht die angegebenen Elemente in der zugewiesenen Bibliothek (logisches Löschen). Dabei werden die
Inhaltsverzeichniseinträge gelöscht und der Speicherplatz freigegeben (Freigabe erfolgt nur innerhalb der Bibliothek, d.h. die Bibliothek wird dadurch als Ganzes nicht kleiner).

Ein Element einer Bibliothek wird physikalisch gelöscht, wenn der Operand DESTROY-DATA=*YES gesetzt ist oder im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder der Systemparameter DESTLEV es verlangt.

Die Anweisung wird nur ausgeführt, wenn in der Anweisung explizit eine Bibliothek angegeben wurde oder die bei OPEN-LIBRARY angegebene Bibliothek mit MODE=*UPDATE eröffnet wurde.

Die DELETE-ELEMENT-Anweisung ist für alle Elementtypen erlaubt.

Format

DELETE-**ELEM**ENT

ELEMENT = ***LIB**RARY-**ELEM**ENT (...)

***LIB**RARY-**ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER-**LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER-**LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEFAULT** / ***ALL** / <alphanum-name 1..8 with-wild(20)>

| ,**USER-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**CRE**ATION-**DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**MOD**IFICATION-**DATE** =***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

,**DESTROY-DATA** = ***DEFAULT** / ***NO** / ***Y**ES

,**DIALOG-CONTR**OL = ***DEFAULT** / ***NO** / ***Y**ES

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angabe, welches Element gelöscht werden soll. Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, aus der das Element gelöscht werden soll.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, aus der das Element gelöscht werden soll.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)
Element, das gelöscht werden soll

ELEMENT = *ALL(...)
Alle Elemente werden gelöscht.

ELEMENT = <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elementes, welches gelöscht werden soll.

VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>
Version, die das zu löschende Element besitzt.

VERSION = *HIGHEST-EXISTING
Das Element mit der höchsten bestehenden Version wird gelöscht.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird gelöscht.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Explizite Angabe der Version des Elementes, das gelöscht werden soll.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Typ, den das zu löschende Element besitzt.

TYPE = *DEFAULT
Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

TYPE = *ALL
Alle Typen werden gelöscht

TYPE = <alphanum-name 1..8 with-wild(20)>
Nur der angegebene Typ wird gelöscht.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das zu löschende Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird gelöscht.

USER-DATE = <date 8..10 with-compl>
Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird gelöscht.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden gelöscht.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Erzeugungsdatum des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

DESTROY-DATA = *DEFAULT / *NO / *YES
Löschen der Daten für alle Elemente, die durch *LIBRARY-ELEMENT bestimmt wurden.

DESTROY-DATA = *DEFAULT
Der Standardwert ist *NO (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DESTROY-DATA = *NO
Ein Element einer Bibliothek wird nur physikalisch gelöscht, wenn im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder der Systemparameter DESTLEV es verlangt.

DESTROY-DATA = *YES
Nach dem logischen Löschen werden die Daten, falls vorhanden, physikalisch gelöscht, d.h. mit X'00' überschrieben.

DIALOG-CONTROL = *DEFAULT / *NO / *YES
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist *NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES
LMSCONV fragt bei jedem Element nach, wie es sich verhalten soll, z.B. ob das Element bearbeitet, übersprungen oder die Anweisung abgebrochen werden soll.

Nähere Erläuterungen zur Dialogführung unter MODIFY-DEFAULTS, wobei der dort ggf. eingestellte Wert *ERROR wie *NO wirkt. Ebenso wirkt der bei /INFORM-PROGRAM ggf. angegebene Wert für DIALOG-CONTROL=*ERROR bei DELETE-ELEMENT wie *NO.

Beispiel

Aus der Bibliothek LIB1 wird das Element TEST3 gelöscht.

```
/start-lmsconv
//open-library lib1
//show-element-attributes
INPUT  LIBRARY= :N:$USER.LIB1
TYP  NAME    VER (VAR#) DATE         NAME    VER (VAR#) DATE
(S)  TEST1   @   (0001) <date>       TEST2   @   (0001) <date>
(S)  TEST3   @   (0001) <date>       TEST4   @   (0001) <date>
//delete-element (elem=test3,type=s)
//show-element-attributes
INPUT  LIBRARY= :N:$USER.LIB1
TYP  NAME    VER (VAR#) DATE         NAME    VER (VAR#) DATE
(S)  TEST1   @   (0001) <date>       TEST2   @   (0001) <date>
(S)  TEST4   @   (0001) <date>
. . .
```

7.4.2.6 END - LMSCONV beenden

Mit END wird das Programm LMSCONV beendet. Alle noch geöffneten Bibliotheken werden geschlossen.

Format

END

Wurde beim Aufruf von LMSCONV mittels `/START-LMSCONV MONJV=<name>` eine Monitor-Jobvariable angegeben, so wird diese bei Beendigung von LMSCONV gefüllt. Der MONJV-Wert gliedert sich in eine 3 Byte lange Zustandsanzeige und in eine 4 Byte lange Rückkehrcode-Anzeige. Die Rückkehrcode-Anzeige gliedert sich in einen 1 Byte langen Beendigungscode (BC) und eine 3 Byte lange Programminformation (PI).

LMSCONV setzt die Zustandsanzeige und den Beendigungscode wie folgt:

Zustandsanzeige

Zustandsanzeige (Byte 1 - 3)	Beendigungscode BC (Byte 4)	Programminformation PI (Byte 5 - 7)	Bemerkung
$T_	0 oder 1	siehe unten	Normale Beendigung
$A_	2 oder 3	siehe unten	Abnormale Beendigung

Beendigungscode

BC	Erläuterung
0	Normale Beendigung LMSCONV lief fehlerfrei.
1	Normale Beendigung Es traten Warnungen oder Fehler auf, die schwächer waren, als der mit MAX-ERROR-WEIGHT (siehe MODIFY-DEFAULTS) eingestellte Wert.
2	Abnormale Beendigung Es wurde ein durch MAX-ERROR-WEIGHT gesetztes Abbruchkriterium erreicht (siehe MODIFY-DEFAULTS).
3	Abnormale Beendigung Der aufgetretene Fehler ist so schwer wiegend, dass eine Fortsetzung des LMSCONV-Laufes nicht möglich oder nicht mehr sinnvoll war. Der LMSCONV-Lauf wurde intern beendet.

Programminformation

PI	Erläuterung
000	BC: 0 LMSCONV lief fehlerfrei.
001	BC: 1 Es traten höchstens Warnungen auf.
002	BC: 1 oder 2 Es traten höchstens Fehler der Klasse RECOVERABLE auf, d.h. ein Element konnte nicht gefunden oder überschrieben werden.
003	BC: 1 oder 2 Es traten höchstens Fehler der Klasse SIGNIFICANT auf, d.h. nicht schwer wiegende Fehler.
004	BC: 2 Es traten höchstens Fehler der Klasse SERIOUS auf, d.h. schwer wiegende Fehler.
005	BC: 3 Der aufgetretene Fehler ist so schwer wiegend, dass eine Fortsetzung des LMSCONV-Laufes nicht möglich oder nicht mehr sinnvoll war. Der LMSCONV-Lauf wurde intern beendet.

7.4.2.7 EXTRACT-ELEMENT - Elemente in Datei ausgeben

Die Anweisung EXTRACT-ELEMENT gibt Elemente in Dateien aus. LMSCONV erzeugt die Dateien entsprechend

- dem Eintrag in der TASK-FILE-TABLE (TFT), wenn die Datei mit dem Dateikettungsnamen zugeordnet ist.
- den gespeicherten Dateimerkmalen und dem Operanden FILE-ATTRIBUTES.
- dem Katalogeintrag.

Die Dateien können RECORD-FORMAT=*UNDEFINED und beliebige Angaben zu BUFFER-LENGTH und RECORD-SIZE haben. Die maximale Satzlänge von 32 KByte (inkl. Satzlänge) darf allerdings nicht überschritten werden.

Wurden die ISAM-Schlüssel einer ISAM-Datei mit in das Element aufgenommen, werden die ISAM-Schlüssel mit EXTRACT-ELEMENT auch wieder mit ausgegeben.

Wenn ein Textelement in eine existierende PAM-Datei ausgegeben wird, dann bestimmt die (möglicherweise implizit getroffene) Einstellung des Operanden ACCESS-METHOD die neue Zugriffsmethode (SAM oder ISAM) der Ausgabedatei.

Wurden bei der Aufnahme der Datei Informationen über ISAM-Sekundärschlüssel gespeichert, werden die Sekundärschlüssel wieder eingerichtet. Ist das Einrichten einzelner oder aller Sekundärschlüssel nicht möglich, wird die Datei ohne oder nicht mit allen Sekundärschlüsseln erzeugt.

Die Anweisung EXTRACT-ELEMENT ist für die Elementtypen S, M, R, J, P, D, X, C erlaubt. Elemente vom Typ C und PAM-Elemente unter dem Typ X werden als PAM-Dateien erzeugt.

Die erzeugte Datei erhält als Katalogattribut CCS den CCS-Namen des Quellelementes.

> ¡ Zulässige Elementnamen sind nicht in jedem Fall als Dateinamen zulässig.

Erzeugen von ISAM-Dateien

Beim Ausgeben von Elementen in ISAM-Dateien erzeugt LMSCONV die ISAM-Schlüssel wie folgt:

- Wurden bei der Aufnahme einer ISAM-Datei als Bibliothekselement die ISAM-Schlüssel mit aufgenommen, erzeugt LMSCONV die ISAM-Datei mit diesen gespeicherten ISAM-Schlüsseln.
- Sind im Eingabeelement keine ISAM-Schlüssel gespeichert, wird eine ISAM-Datei mit KEY-POSITION=5 und KEY-LENGTH=8 erstellt. Standardmäßig erzeugt LMSCONV dann ISAM-Schlüssel mit einem Anfangswert von 1000 und einer Schrittweite von 1000. Ist das Element zu groß für diese Schrittweite (mehr als 100000 Sätze), so wird die Schrittweite aus der Anzahl der Sätze berechnet.

> ¡
> - Elemente vom Typ R werden bis zum END-Satz ausgegeben. Eventuell nachfolgende Sätze werden ignoriert.
> - Korrekturjournalsätze (TXTP) werden bei Elementen vom Typ C nicht mit ausgegeben.
> - Die RECORD-SIZE wird nur bei RECORD-FORMAT=*FIXED versorgt; sie hat bei RECORD-FORMAT=*VARIABLE den Wert 0.

Format

EXTRACT-ELEMENT

ELEMENT = ***LIB**RARY**-ELEM**ENT (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEF**AULT / ***ALL** / <alphanum-name 1..8 with-wild(20)>

| ,**USER-DATE** =***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**CRE**ATION**-DATE** =***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**MOD**IFICATION**-DATE** =***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

,**TO-F**ILE = ***STD** / ***BY-SOUR**CE / <filename 1..54 without-gen-vers with-wild-constr> /

***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,**F**ILE**-ATTR**IBUTES = ***BY-ELEM**ENT / ***BY-CAT**ALOG / ***DEF**AULT / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **ACCESS-METHOD** = ***DEFAULT** / ***ISAM** / ***SAM**

,**WR**ITE**-MODE** = ***DEFAULT** / ***CREATE** / ***REPLACE** / ***EXTEND** / ***ANY**

,**DIALOG-CONTR**OL = ***DEFAULT** / ***NO** / ***Y**ES / ***ERROR**

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich das Element befindet.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in der sich das Element befindet.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)
Angabe der Elemente, die aus der Bibliothek übernommen werden sollen.

ELEMENT = *ALL(...)
Alle Elemente werden aus der Bibliothek übernomen.

ELEMENT = <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elementes, welches aus der Bibliothek in eine Datei übernommen werden soll.

VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT /
<composed-name 1..24 with-under with-wild(52)>
Version, die das auszugebende Element besitzt.

VERSION = *HIGHEST-EXISTING
Das Element mit der höchsten bestehenden Version wird verwendet.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird ausgegeben.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Explizite Angabe der Version des Elementes, das ausgegeben werden soll.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Typ, den das zu übernehmende Element besitzt.

TYPE = *DEFAULT
Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

TYPE = *ALL
Alle Typen werden übernommen

TYPE = <alphanum-name 1..8 with-wild(20)>
Nur der angegebene Typ wird übernommen.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das auszugebende Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird ausgewählt.

USER-DATE = <date 8..10 with-compl>
Das Element wird ausgewählt, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden ausgewählt.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Erzeugungsdatum des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

TO-FILE = *STD / *BY-SOURCE / <filename 1..54 without-gen-vers with-wild-constr> / *LINK(...)
Name der Zieldatei.

TO-FILE = *STD
Standardmäßig werden die Elementdaten in Dateien ausgegeben, wobei die Datei den gleichen Namen wie das Element erhält.

TO-FILE = *BY-SOURCE
Der Dateiname ist der gleiche Name wie der Elementname.

TO-FILE = <filename 1..54 without-gen-vers with-wild-constr>
Name der Zieldatei. Eine Konstruktionsangabe bezieht sich auf den Elementnamen.

TO-FILE = *LINK(...)
Das Element wird in die Datei ausgegeben, die über den Dateikettungsnamen zugewiesen wurde.

LINK-NAME = <structured-name 1..8>
Dateikettungsname.

FILE-ATTRIBUTES = *BY-ELEMENT / *BY-CATALOG / *DEFAULT / *PARAMETERS(...)
Dateiattribute, die bei der Erzeugung der Datei festgelegt werden. LMSCONV bestimmt die Dateiattribute nach folgender Hierarchie:

1. LINK-Eintrag
2. im Element gespeicherte Dateiattribute
3. Katalogeintrag
4. Default-Werte von LMSCONV

Die nachfolgenden Angaben wirken nur, wenn nicht TO-FILE=*LINK angegeben wurde.

FILE-ATTRIBUTES = *BY-ELEMENT
Die im Element gespeicherten Dateiattribute haben Vorrang.

FILE-ATTRIBUTES = *BY-CATALOG
Die im Katalogeintrag gespeicherten Attribute haben Vorrang. Fehlt der Katalogeintrag, wirkt die Angabe *BY-CATALOG wie die Angabe *BY-ELEMENT.

> Die Datei kann auch als SAM-Node-File auf Net-Storage mit dem als Dateiattribut gespeicherten Zeichensatz (NETCCSN) erstellt werden.

FILE-ATTRIBUTES = *PARAMETERS(...)

ACCESS-METHOD = *DEFAULT / *ISAM / *SAM
Legt die Zugriffsmethode ISAM oder SAM für die Zieldatei fest. Der Standardwert ist *ISAM oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert

WRITE-MODE = *DEFAULT / *CREATE / *REPLACE / *EXTEND / *ANY
Überschreiben einer Datei gleichen Namens. Existiert die Datei nicht unter diesem Namen, wird sie neu angelegt.

WRITE-MODE = *DEFAULT
Der Standardwert ist *CREATE (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

WRITE-MODE = *CREATE
Die neue Datei darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Die neue Datei muss bereits existieren und wird dann ersetzt.

WRITE-MODE = *EXTEND
Die Datei wird erweitert, wenn sie bereits existiert. Ansonsten wird sie neu angelegt.

WRITE-MODE = *ANY
Die Datei wird ersetzt, wenn sie bereits existiert. Ansonsten wird sie neu angelegt.

DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-DEFAULTS.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist *NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES / *ERROR
Siehe die Beschreibung bei der Anweisung MODIFY-DEFAULTS.

Hinweis

Das Extrahieren eines Elements in ein SAM-Node-File auf Net-Storage kann daran scheitern, dass der CCSN des Elements nicht in den NETCCSN des Node-Files konvertiert werden kann. Dies kann mehrere Ursachen haben:

1. Dem Element ist ein unpassender CCSN zugewiesen. In diesem Fall kann mit //MODIFY-ELEMENT-ATTRIBUTES ein passender CCSN zugewiesen werden.
2. Für das Element ist in den Dateiattributen ein unpassender NETCCSN gespeichert (zu erkennen in der Ausgabe von //SHOW-ELEMENT TEXT-INFORMATION=*FILE-ATTRIBUTES). In diesem Fall muss dem Node-File, in das das Element extrahiert werden soll, mit /MODIFY-FILE-ATTRIBUTES der passende NETCCSN zugewiesen werden. Anschließend kann das Element unter Angabe von FILE-ATTRIBUTES=*BY-CATALOG extrahiert werden.
3. Für das Element ist kein NETCCSN gespeichert (z.B. weil es ursprünglich kein Node-File war oder weil beim Aufnehmen des Node-Files in die Bibliothek der Operand SOURCE-ATTRIBUTES=*KEEP nicht angegeben wurde), und der voreingestellte NETCCSN des Benutzers passt nicht zum CCSN. Allgemein sollte in diesem Fall das Benutzerattribut NET-CODED-CHARACTER-SET passend geändert werden. Im Einzelfall kann aber auch genauso vorgegangen werden wie in Fall 2.

Beispiele

Das Element ELEM1 wird mit EXTRACT-ELEMENT in die Datei TEST mit den spezifizierten Dateieigenschaften ausgegeben.

```
/add-file-link file-name=test,link-name=out,access-method=*sam, -
/              record-format=*variable
/start-lmsconv
//open-library library=libin
//extract-element (,elem1,s),*link(link-name=out)
//end
```

Sollen alle Elemente einer Bibliothek unter ihrem Namen ausgegeben werden, ist folgende Anweisung anzugeben:

```
//extract-element (elem=*all,type=*all)
```

7.4.2.8 MODIFY-DEFAULTS - Voreinstellungen verändern

Mit der Anweisung MODIFY-DEFAULTS können die Default-Werte geändert werden. Wird in einer LMSCONV-Anweisung lokal ein expliziter Wert benutzt, hat dieser Vorrang vor dem Default-Wert.

Der Bezug in den LMSCONV-Anweisungen zu den hier eingestellten Werten ist die Angabe von *DEFAULT.

Zu Beginn des LMSCONV-Laufs gelten die unmittelbar auf *UNCHANGED folgenden Werte. Wird einer dieser Werte durch die MODIFY-DEFAULTS-Anweisung verändert, wird diese neue Einstellung zur aktuellen Einstellung. Diese bleibt solange für den LMSCONV-Lauf gültig (*UNCHANGED), bis eine neue Anweisung MODIFY-DEFAULTS für diesen Wert gegeben wird.

Format

MODIFY-**DEFAULT**S

ELEMENT-**ATTR**IBUTES = ***UNCHANGED** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **TYPE** = ***UNCHANGED** / ***NONE** / <alphanum-name 1..8>

| ,**ELEM**ENT-**VERSION** = ***UNCHANGED** / ***ALL** / ***HIGH**EST-**EXIST**ING

| ,**TO-ELEM**ENT-**VERSION** = ***UNCHANGED** / ***BY-SOUR**CE

| ,**SOURC**E-**ATTR**IBUTES = ***UNCHANGED** / ***STD** / ***IGNORE** / ***KEEP**(...)

| ***KEEP**(...)

| | **KEEP-TYPES** = ***UNCHAN**GED / ***STD** / ***ALL**

,**F**ILE-**ATTR**IBUTES = ***UNCHANGED** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **ACCESS-METHOD** = ***UNCHANGED** / ***ISAM** / ***SAM**

,**DESTROY-DATA** = ***UNCHANGED** / ***NO** / ***Y**ES / ***BY-SOUR**CE

,**WR**ITE-**MODE** = ***UNCHANGED** / ***CREATE** / ***REPLACE** / ***EXTEND** / ***ANY**

,**DIALOG-CONTR**OL = ***UNCHANGED** / ***NO** / ***Y**ES / ***ERROR**

,**INF**ORMATION = ***UNCHANGED** / ***MED**IUM / ***MIN**IMUM / ***MAX**IMUM / ***SUMM**ARY

,**LAYOUT** = ***UNCHANGED** / ***VAR**IABLE / ***FIXED**

,**SORT** = ***UNCHANGED** / ***BY-NAME** / ***BY-VERS**ION / ***BY-USER-DATE** / ***BY-SEC**ONDARY-**NAME**

,**OUT**PUT**-FORM** = ***UNCHANGED** / ***STD** / ***CHAR**ACTER / ***HEX**ADECIMAL / ***DUMP**

,**DEL**ETE**-SOUR**CE = ***UNCHANGED** / ***NO** / ***Y**ES

,**MAX-ERROR-WEIGHT** = ***UNCHANGED** / ***SERIOUS** / ***SIGNIFICANT** / ***RECOV**ERABLE

,**RUN-MODE** = ***UNCHANGED** / ***STD** / ***BATCH**

,**NEXT-ATTEM**PT = ***UNCHANGED** / ***NO** / ***Y**ES(...)

***Y**ES(...)

| **NUM**BER**-OF-ATTEM**PTS = ***UNCHANGED** / <integer 1..2147483647>

| ,**PER**IOD = ***UNCHANGED** / <integer 1..21599>

,**TEXT-INF**ORMATION = ***UNCHANGED** / ***ALL** / ***STATIS**TICS / ***F**ILE**-ATTR**IBUTES / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **INF**ORMATION = ***UNCHANGED** / ***ALL** / list-poss(2): ***TEXT** / ***COMMENT**

| ,**REC**ORD**-RANGE** = ***UNCHANGED** / ***ALL** / ***RANGE**(...)

| ***RANGE**(...)

| | **FROM** = ***UNCHANGED** / <integer 1..2147483647>

| | ,**TO** = ***UNCHANGED** / ***LAST** / <integer 1..2147483647>

| ,**REC**ORD**-PART** = ***UNCHANGED** / ***ALL** / ***PART**(...)

| ***PART**(...)

| | **START** = ***UNCHANGED** / <integer 1..32764>

| | ,**LEN**GTH = ***UNCHANGED** / ***REST** / <integer 1..32764>

| ,**REC**ORD**-NUM**BER = ***UNCHANGED** / ***BY-OUT**PUT / ***Y**ES / ***NO**

,**MOD**ULE**-INF**ORMATION = ***UNCHANGED** / ***ALL** / ***STATIS**TICS / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **INF**ORMATION = ***UNCHANGED** / ***ALL** / ***TXT**(...) / ***TXTP**(...) / list-poss(9): ***ESD** / ***ISD** /

| ***LSD** / ***RLD** / ***REP** / ***INCL**UDE / ***DSDD** / ***REF** / ***END**

| ***TXT**(...)

| | **CSECT**-NAME = ***UNCHANGED** / ***ALL** / <c-string 1..32 with-low> / <text 1..32>

| | ,**ADDR**ESS = ***UNCHANGED** (...) / <x-string 1..8>(...)

| | ***UNCHANGED**(...)

| | | **BASE-ADDR**ESS = ***UNCHANGED** / <x-string 1..8>

| | <x-string>(...)

| | | **BASE-ADD**RESS = **<u>*UNCHANGED</u>** / <x-string 1..8>

| | ,**LEN**GTH = **<u>*UNCHANGED</u>** / ***REST** / <integer 1..2147483647> / <x-string 1..8>

| ***TXTP**(...)

| | **MOD**IFICATION**-ID** = ***UNCHANGED** / ***ALL** / <c-string 1..8 with-low> / ***RANGE**(...)

| | ***RANGE**(...)

| | | **FROM** = ***UNCHANGED** / ***LOW**EST / <c-string 1..8 with-low>

| | | ,**TO** = ***UNCHANGED** / ***HIGHEST** / <c-string 1..8 with-low>

,**PHASE-INF**ORMATION = ***UNCHANGED** / ***ALL** / ***STATIS**TICS / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **SEGMENT** = ***UNCHANGED** / ***ALL** / ***ROOT** / <name 1..8>

| ,**INF**ORMATION = ***UNCHANGED** / ***ALL** / ***TXT**(...) / ***TXTP**(...) / list-poss(4): ***ESD** / ***ISD** /

| ***LSD** / ***RLD**

| ***TXT**(...)

| | **ADDR**ESS = ***UNCHANGED** (...) / <x-string 1..8>(...)

| | ***UNCHANGED**(...)

| | | **BASE-ADD**RESS = ***UNCHANGED** / <x-string 1..8>

| | <x-string>(...)

| | | **BASE-ADD**RESS = ***UNCHANGED** / <x-string 1..8>

| | ,**LEN**GTH = ***UNCHANGED** / ***REST** / <integer 1..2147483647> / <x-string 1..8>

| ***TXTP**(...)

| | **MOD**IFICATION**-ID** = ***UNCHANGED** / ***ALL** / <c-string 1..8 with-low> / ***RANGE**(...)

| | ***RANGE**(...)

| | | **FROM** = ***UNCHANGED** / ***LOWE**ST <c-string 1..8 with-low>

| | | ,**TO** = ***UNCHANGED** / ***HIGHEST** / <c-string 1..8 with-low>

,**LLM-INF**ORMATION = ***UNCHANGED** / ***ALL** / ***STATIS**TICS / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **LLM-PART** = ***UNCHANGED** / ***ALL** / ***SLICE**(...) / ***SUB-LLM**(...)

| ***SLICE**(...)

| | **NAME** = ***UNCHANGED** / <structured-name 1..32>

| ***SUB-LLM**(...)

| | **PATH-NAME** = ***UNCHANGED** / <c-string 1..255 with-low> / <text 1..255>

| ,**INF**ORMATION = ***UNCHANGED** / ***ALL** / ***TXT**(...) / ***TXTP**(...) / ***LOGICAL**(...) / ***PHYS**ICAL / ***REF** /

| list-poss(4): ***RELOCATION / *ESVD** / ***ESVR** / ***LRLD**

| ***TXT**(...)

| | **CSECT**-NAME = **<u>*UNCHANGED</u>** / ***ALL** / <c-string 1..32 with-low> / <text 1..32>

| | ,**ADDR**ESS = **<u>*UNCHANGED</u>** (...) / <x-string 1..8>(...)

| | ***UNCHANGED**(...)

| | | **BASE-ADDR**ESS = **<u>*UNCHANGED</u>** / <x-string 1..8>

| | <x-string>(...)

| | | **BASE-ADDR**ESS = **<u>*UNCHANGED</u>** / <x-string 1..8>

| | ,**LEN**GTH = **<u>*UNCHANGED</u>** / ***REST** / <integer 1..2147483647> / <x-string 1..8>

| ***TXTP**(...)

| | **CSECT**-NAME = **<u>*UNCHANGED</u>** / ***ALL** / <c-string 1..32 with-low> / <text 1..32>

| | ,**MOD**IFICATION**-ID** = **<u>*UNCHANGED</u>** / ***ALL** / <c-string 1..12 with-low> / ***RANGE**(...)

| | ***RANGE**(...)

| | | **FROM** = **<u>*UNCHANGED</u>** / ***LOW**EST / <c-string 1..12 with-low>

| | | ,**TO** = **<u>*UNCHANGED</u>** / ***HIGHEST** / <c-string 1..12 with-low>

| ***LOGICAL**(...)

| | **LEVEL** = **<u>*UNCHANGED</u>** / ***ALL** / ***NEXT**

Operanden

ELEMENT-ATTRIBUTES = <u>*UNCHANGED</u> / *PARAMETERS(...)
Bestimmt den Elementtyp, die Elementversion und die Speicherungsform sowie die Dateieigenschaften.

TYPE = <u>*UNCHANGED</u> / *NONE / <alphanum-name 1..8>
Angabe des Elementtyps.

TYPE = *NONE
Es wird kein globaler Elementtyp festgelegt, d.h. die Typangaben müssen anweisungslokal erfolgen.

TYPE = <alphanum-name 1..8>
Der hier angegebene Name wird in den Anweisungen als Typ verwendet.

ELEMENT-VERSION = <u>*UNCHANGED</u> / *ALL / *HIGHEST-EXISTING
Festlegung der Elementversion für SHOW-ELEMENT-ATTRIBUTES.

ELEMENT-VERSION = *ALL
Es werden alle Versionen zu einem Element ausgegeben.

ELEMENT-VERSION = *HIGHEST-EXISTING
Es wird nur die höchste Version eines Elementes ausgegeben.

TO-ELEMENT-VERSION = <u>*UNCHANGED</u> / *BY-SOURCE
Festlegung der Version des Zielelementes.

TO-ELEMENT-VERSION = *BY-SOURCE
Das Zielelement erhält die gleiche Version wie das Quellelement.

SOURCE-ATTRIBUTES = *UNCHANGED / *STD / *IGNORE / *KEEP(...)
Wird nur bei der Anweisung ADD-ELEMENT ausgewertet.

SOURCE-ATTRIBUTES = *STD
Es werden keine Dateiattribute und auch kein ISAM-Schlüssel gespeichert. In diesem Fall ist es nur möglich, ISAM-Dateien mit KEY-POSITION=5, KEY-LENGTH <= 16 und RECORD-FORMAT=*VARIABLE in das Element aufzunehmen.
Für ISAM-Dateien wird eine Warnung ausgegeben, dass die ISAM-Schlüssel nicht aufgenommen wurden.

SOURCE-ATTRIBUTES = *IGNORE
Wie SOURCE-ATTRIBUTES=*STD, es wird jedoch keine Warnung ausgegeben.

SOURCE-ATTRIBUTES = *KEEP(...)
Folgende Dateiattribute werden unverändert im neu aufzunehmenden Element gespeichert: ACCESS-METHOD, RECORD-FORMAT, RECORD-SIZE, BUFFER-LENGTH, PERFORMANCE, USAGE, ACCESS und USER-ACCESS. Ist ACCESS-METHOD=ISAM, so werden PADDING-FACTOR, LOGICAL-FLAG-LENGTH, VALUE-FLAG-LENGTH, PROPAGATE-VALUE-FLAG und auch die ISAM-Schlüssel sowie Informationen über ISAM-Sekundärschlüssel zusätzlich gespeichert. Für SAM-Node-Files wird ferner der Zeichensatz auf Net-Storage (NETCCSN) gespeichert.

KEEP-TYPES = *UNCHANGED / *STD / *ALL
Bestimmt die Elementtypen, für die Dateiattribute gespeichert werden sollen.

KEEP-TYPES = *STD
Attribute werden für Elemente der Basistypen S, M, P, D, J, X gespeichert.

KEEP-TYPES = *ALL
Attribute werden für Elemente aller Basistypen gespeichert, die für ADD-ELEMENT zulässig sind.

ℹ LMSCONV speichert den Zeichensatz von SAM-Node-Files auf Net-Storage (NETCCSN) als Elementattribut. LMSCONV speichert den Last Byte Pointer (LBP) zusatzlich zu den bisher schon gespeicherten Dateiattributen

- für PAM-Elemente vom Typ X
- für Elemente vom Typ C, wenn KEEP-TYPES = *ALL angegeben wurde

FILE-ATTRIBUTES = *UNCHANGED / *PARAMETERS(...)
Dateiattribute, die bei der Erzeugung der Datei festgelegt werden.

ACCESS-METHOD = *UNCHANGED / *ISAM / *SAM
Festlegung der Dateizugriffsmethode.

ACCESS-METHOD = *ISAM
ISAM-Datei erzeugen.

ACCESS-METHOD = *SAM
SAM-Datei erzeugen.

DESTROY-DATA = *UNCHANGED / *NO / *YES / *BY-SOURCE
Bestimmt, ob die Daten physikalisch gelöscht, d.h. mit X'00' überschrieben werden.

DESTROY-DATA = *NO
Ein Element einer Bibliothek wird nur physikalisch gelöscht, wenn im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder der Systemparameter DESTLEV es verlangt.

DESTROY-DATA = *YES
Nach dem logischen Löschen werden die Daten, falls vorhanden, physikalisch gelöscht.

DESTROY-DATA = *BY-SOURCE
Das Kennzeichen zum Überschreiben der Daten wird dem Quellelement bzw. der Quelldatei entnommen und dem Zielelement bzw. der Zieldatei zugeordnet. Fehlt die Quelle, dann wirkt *BY-SOURCE wie DESTROY-DATA=*NO.

WRITE-MODE = *UNCHANGED / *CREATE / *REPLACE / *EXTEND / *ANY
Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

Ist der ausgewählte Wert für WRITE-MODE anweisungslokal nicht möglich, gilt die Einstellung WRITE-MODE= *CREATE.

WRITE-MODE = *CREATE
Das Zielelement darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Das Zielelement muss bereits existieren und wird dann ersetzt.

WRITE-MODE = *EXTEND
Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt.

WRITE-MODE = *ANY
Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

DIALOG-CONTROL = *UNCHANGED / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht. (Dieser Operand wirkt nicht im Prozedur- oder Batchbetrieb.)

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage bearbeitet.

DIALOG-CONTROL = *YES
LMSCONV fragt bei jedem Element nach, wie es sich verhalten soll, z.B. ob das Element bearbeitet, übersprungen oder die Anweisung abgebrochen werden soll.

DIALOG-CONTROL = *ERROR
Tritt während der Elementbearbeitung ein behebbarer Fehler auf, z.B. Überschreiben eines Elementes, wird der Anwender gefragt, wie LMSCONV sich verhalten soll.

Der Benutzer hat ggf. folgende Eingriffsmöglichkeiten:

YES	Das Element soll bearbeitet werden.
NO	Das Element soll nicht bearbeitet werden.
ALL	Die Anweisung soll ohne Dialog beendet werden.
TERMINATE	Die Anweisung soll abgebrochen werden.

Nach Drücken der Taste K2 und Eingabe von „/INFORM-PROGRAM“ hat der Anwender die Möglichkeit, den Wert des Operanden DIALOG-CONTROL zu verändern. Befindet sich LMSCONV mitten in der Elementverarbeitung, so kann der Anwender die Weiterverarbeitung beeinflussen mit:

`/INFORM-PROGRAM '[N-I / N-E / C][,DIALOG-CONTROL=*NO / *YES / *ERROR]'`

`/INFORM-PROGRAM 'NEXT-INPUT (N-I)'`: die aktuelle Anweisung wird abgebrochen; LMSCONV liest eine weitere Anweisung ein, sobald es wieder aktiv ist. Ein unbekannter oder fehlender Text wirkt wie NEXT-INPUT.

`/INFORM-PROGRAM 'NEXT-ELEMENT (N-E)'`: die Bearbeitung des aktuellen Elementes in der laufenden Anweisung wird abgebrochen; LMSCONV setzt die Verarbeitung mit dem nächsten Element (sofern vorhanden) fort. Ist kein nächstes Element vorhanden, wirkt NEXT-ELEMENT wie NEXT-INPUT.

`/INFORM-PROGRAM 'CONTINUE (C)'`: die LMSCONV-Verarbeitung wird regulär fortgesetzt.

C,NO	Das Element soll bearbeitet, aber der Dialog nicht eingeschaltet werden.
N-E/N-I,NO	Die Elementverarbeitung bzw. die Anweisung soll abgebrochen, aber der Dialog nicht eingeschaltet werden.
C,YES/ERROR	Das Element soll bearbeitet und danach der Dialog eingeschaltet werden.
N-E/N-I,YES/ERROR	Die Elementverarbeitung bzw. die Anweisung soll abgebrochen und danach der Dialog eingeschaltet werden.

> ¡ Der mit `/INFORM-PROGRAM` eingestellte Wert für `DIALOG-CONTROL` gilt nur für die aktuelle Anweisung.

INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM / *SUMMARY
Dieser Parameter bestimmt den Umfang des auszugebenden Inhaltsverzeichnisses.

INFORMATION = *MEDIUM
Es wird Typ, Name, Version, Variantennummer und das Benutzerdatum ausgegeben.

INFORMATION = *MINIMUM
Es wird nur Typ, Name und Version ausgegeben.

INFORMATION = *MAXIMUM
Zusätzlich zu Typ, Name, Version, Variantennummer und Benutzerdatum wird für alle Elemente der bestehende Elementschutz, die Speicherungsform, der zugeordnete Zeichensatz, wie auch Benutzer-, Erzeugungs- und Änderungsdatum und -uhrzeit mit protokolliert.

INFORMATION = *SUMMARY
Es wird nur die Anzahl der Elemente pro Typ ausgegeben.

LAYOUT = *UNCHANGED / *VARIABLE / *FIXED
Dieser Parameter bestimmt das Format des auszugebenden Inhaltsverzeichnisses.

LAYOUT = *VARIABLE
Die Anzahl der Druckspalten ist abhängig von der längsten Elementbezeichnung innerhalb eines Elementtypes.

LAYOUT = *FIXED
Das Inhaltsverzeichnis wird einspaltig im festen Format gedruckt. Eine Spalte bedeutet die untereinander stehenden Einträge im Inhaltsverzeichnis.

SORT = *UNCHANGED / *BY-NAME / *BY-VERSION / *BY-USER-DATE / *BY-SECONDARY-NAME
Sortierkriterium für die Inhaltsverzeichniseinträge der ausgewählten Elemente. Als erstes Sortierkriterium wird immer der Typ verwendet.

SORT = *BY-NAME
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Name und Version.

SORT = *BY-VERSION
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Version und Name.

SORT = *BY-USER-DATE
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Benutzerdatum, Name und Version.

SORT = *BY-SECONDARY-NAME
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Sekundärname, Sekundärattribut, Name und Version.

Weitere Hinweise siehe Anweisung SHOW-ELEMENT-ATTRIBUTES.

OUTPUT-FORM = *UNCHANGED / *STD / *CHARACTER / *HEXADEZIMAL / *DUMP
Festlegung des Darstellungsformates für die Ausgabe.

OUTPUT-FORM = *STD
Das Darstellungsformat wird abhängig vom Typ des Elementes gewählt.

OUTPUT-FORM = *CHARACTER
Die Ausgabe wird alphanumerisch dargestellt.

OUTPUT-FORM = *HEXADECIMAL
Die Ausgabe wird alphanumerisch und hexadezimal übereinander dargestellt.

OUTPUT-FORM = *DUMP
Die Ausgabe wird alphanumerisch und hexadezimal nebeneinander dargestellt. Für die Elementtypen S, P, D, J und M wirkt dieser Operand wie *HEXADECIMAL.

DELETE-SOURCE = *UNCHANGED / *NO / *YES
Hier kann der Benutzer bestimmen, ob die ursprüngliche Datei erhalten werden soll (Default-Wert *NO) oder gelöscht werden soll (Parameter *YES). Dieser Operand ist wirkungslos, wenn von *OMF gelesen wird.

MAX-ERROR-WEIGHT = *UNCHANGED / *SERIOUS / *SIGNIFICANT / *RECOVERABLE
Dieser Operand bestimmt, in welchen Fehlerfällen LMSCONV den Spin-off Mechanismus auslösen soll.

MAX-ERROR-WEIGHT = *SERIOUS
Der Spin-off-Mechanismus wird bei schwer wiegenden Fehlern, d.h. Fehler bei denen eine Fortsetzung der Anweisung nicht sinnvoll ist, ausgelöst.

MAX-ERROR-WEIGHT = *SIGNIFICANT
Der Spin-off-Mechanismus wird wie bei *SERIOUS und zusätzlich bei sonstigen Fehlern (außer Element konnte nicht gefunden oder überschrieben werden) ausgelöst.

MAX-ERROR-WEIGHT = *RECOVERABLE
Der Spin-off-Mechanismus wird bei allen Fehlern ausgelöst.

RUN-MODE = *UNCHANGED / *STD / *BATCH
Wenn LMSCONV im Dialog abläuft, dann legt dieser Operand fest, ob LMSCONV normal abläuft oder sich so verhält, als würde es im Batchbetrieb ablaufen.
Dieser Operand hat in Prozeduren oder im Batchbetrieb keine Wirkung.

RUN-MODE = *STD
LMSCONV soll normal ablaufen.

RUN-MODE = *BATCH
LMSCONV soll sich so verhalten, als würde es im Batchbetrieb ablaufen.

NEXT-ATTEMPT = *UNCHANGED / *NO / *YES(...)
Steuert die weiteren Öffnungsversuche bei Datei-, Typ- oder Elementsperre im Prozedur- oder Batchbetrieb.

NEXT-ATTEMPT = *NO
Es werden keine weiteren Öffnungsversuche vorgenommen.

NEXT-ATTEMPT = *YES(...)
Es werden weitere Öffnungsversuche vorgenommen.

NUMBER-OF-ATTEMPTS = *UNCHANGED / <integer 1..2147483647>
Anzahl der weiteren Öffnungsversuche (Standardmäßig 9 Versuche).

PERIOD = *UNCHANGED / <integer 1..21599>
Wartezeit in Sekunden zwischen zwei Versuchen (Standardmäßig 6 Sekunden).

TEXT-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *FILE-ATTRIBUTES / *PARAMATERS(...)
Legt die Informationsmenge für alle Elemente außer den Elementtypen R, C und L fest. Für PAM-Elemente wirken alle Angaben außer *FILE-ATTRIBUTES wie *ALL.

TEXT-INFORMATION = *ALL
Es wird alles ausgegeben.

TEXT-INFORMATION = *STATISTICS
Es wird die Anzahl der Sätze pro Satzart und die Summe der Sätze ausgegeben.

TEXT-INFORMATION = *FILE-ATTRIBUTES
Es werden nur die gespeicherten Dateiattribute ausgegeben.

TEXT-INFORMATION = *PARAMETERS(...)
Es wird ein Elementausschnitt festgelegt, der ausgegeben werden soll.

INFORMATION = *UNCHANGED / *ALL / list-poss(2): *TEXT / *COMMENT
Der anzuzeigende Teilbereich des Elementes.

INFORMATION = *ALL
Es werden alle Anwendersatzarten angezeigt.

INFORMATION = *TEXT
Es wird der eigentliche Text d.h. Satzart 1 angezeigt.

INFORMATION = *COMMENT
Es wird der separat gespeicherte Kommentar d.h. Satzart 2 angezeigt.

RECORD-RANGE = *UNCHANGED / *ALL / *RANGE(...)
Der zu bearbeitende Teilbereich des Elementes.

RECORD-RANGE = *ALL
Es werden alle Anwendersatzarten bearbeitet.

RECORD-RANGE = *RANGE(...)
Der zu bearbeitende Satznummernbereich wird bestimmt. Die Satznummern beziehen sich nicht auf eine Satzart, sondern auf den durch INFORMATION= festgelegten Teilbereich des Elementes. Innerhalb dieses Teilbereichs werden die Sätze von 1 bis n durchnummeriert.

FROM = *UNCHANGED / <integer 1..2147483647>
Beginn des Bereichs durch Angabe der ersten Satznummer. Standardmäßig wird die Satznummer 1 genommen.

TO = *UNCHANGED / *LAST / <integer 1..2147483647>
Ende des Bereichs durch Angabe der letzten Satznummer. Standardmäßig wird die letzte Satznummer genommen.

RECORD-PART = *UNCHANGED / *ALL / *PART(...)
Der zu bearbeitende Satzbereich wird bestimmt.

RECORD-PART = *ALL
Es wird der ganze Satz bearbeitet.

RECORD-PART = *PART(...)
Der zu bearbeitende Satzbereich wird bestimmt. Werden die Default-Werte nicht verändert, wird der ganze Satz bearbeitet.

START = *UNCHANGED / <integer 1..32764>
Beginn des Bereichs durch Angabe des ersten Zeichens im Satz. Standardmäßig wird das erste Zeichen genommen.

LENGTH = *UNCHANGED / *REST / <integer 1..32764>
Länge des Bereichs. Standardmäßig wird der Rest des Satzes genommen.

RECORD-NUMBER = *UNCHANGED / *BY-OUTPUT / *YES / *NO
Bestimmt die Ausgabe der Satznummern.

RECORD-NUMBER = *BY-OUTPUT
Nur wenn die Ausgabe nach SYSOUT erfolgt, werden keine Satznummern ausgegeben. Bei jedem anderen Ausgabemedium werden sie mit ausgegeben.

RECORD-NUMBER = *YES
Die Satznummern werden auch nach SYSOUT ausgegeben.

RECORD-NUMBER = *NO
Es werden keine Satznummern mit ausgegeben.

MODULE-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *PARAMETERS(...)
Legt die Informationsmenge für Bindemodule (Elemente vom Typ R) fest.

MODULE-INFORMATION = *ALL
Es wird alles ausgegeben.

MODULE-INFORMATION = *STATISTICS
Es werden Name, Länge und Adresse der CSECTS sowie die Gesamtlänge des Moduls ausgegeben.

MODULE-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) / list-poss(9): *ESD / *ISD / *LSD / *RLD / *REP / *INCLUDE / *DSDD / *REF / *END
Die hier aufgezählten Satzarten können ausgewählt werden.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

CSECT-NAME = *UNCHANGED / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die Textsätze können auf eine CSECT eingeschränkt werden.

ADDRESS = *UNCHANGED(...) / <x-string 1..8>(...)
Anfangsadresse des Textes.

BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

LENGTH = *UNCHANGED / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

MODIFICATION-ID = *UNCHANGED / *ALL / <c-string 1..8 with-low> /*RANGE(...)
Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *UNCHANGED / *LOWEST / <c-string 1..8 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *UNCHANGED / *HIGHEST / <c-string 1..8 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

PHASE-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *PARAMETERS(...)
Legt die Informationsmenge für Phasen (Elemente vom Typ C) fest.

PHASE-INFORMATION = *ALL
Es wird alles ausgegeben.

PHASE-INFORMATION = *STATISTICS
Es werden Name, Länge und Adresse des Segmentes sowie die Gesamtlänge des Segmentes ausgegeben.

PHASE-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

SEGMENT = *UNCHANGED / *ALL / *ROOT / <name 1..8>
Phasensegment, das ausgewählt wird.

INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) / list-poss(4): *ESD / *ISD / *LSD / *RLD
Die hier aufgezählten Satzarten können ausgewählt werden.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

ADDRESS = *UNCHANGED (...) / <x-string 1..8>(...)
Anfangsadresse des Textes.

BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

LENGTH = *UNCHANGED / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

MODIFICATION-ID = *UNCHANGED / *ALL / <c-string 1..8 with-low> /*RANGE(...)
Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *UNCHANGED / *LOWEST / <c-string 1..8 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *UNCHANGED / *HIGHEST / <c-string 1..8 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

LLM-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / PARAMETERS(...)
Legt die Informationsmenge für Bindelademodule (Elemente vom Typ L) fest.

LLM-INFORMATION = *ALL
Es wird alles ausgegeben.

LLM-INFORMATION = *STATISTICS
Allgemeine Informationen über das Bindelademodule (Name, Copyright,..) werden ausgegeben.

LLM-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

LLM-PART = *UNCHANGED / *ALL / *SLICE(...) / *SUB-LLM(...)
Angabe des LLM-Teils, das ausgewählt werden soll. Standardmäßig wird der gesamte LLM ausgewählt.

LLM-PART = *SLICE(...)
Angabe der SLICE, die ausgegeben werden soll.

NAME = *UNCHANGED / <structured-name 1..32>
Name der SLICE, die ausgegeben werden soll.

LLM-PART = *SUB-LLM(...)
Angabe des SUB-LLM, der ausgegeben werden soll.

PATH-NAME = *UNCHANGED / <c-string 1..255 with-low> / <text 1..255>
Der auszugebende SUB-LLM wird durch seinen Pfadnamen bestimmt.

INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) / *LOGICAL(...) / *PHYSICAL / *REF / list-poss(3): *ESVD / *ESVR / *LRLD / *RELOCATION
Die hier aufgezählten Satzarten können ausgewählt werden.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

CSECT-NAME = *UNCHANGED / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die Textsätze können auf eine CSECT eingeschränkt werden.

ADDRESS = *UNCHANGED(...) / <x-string 1..8>(...)
Anfangsadresse des Textes.

BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

LENGTH = *UNCHANGED / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

CSECT-NAME = *UNCHANGED / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die TXTP-Sätze können auf eine CSECT eingeschränkt werden.

MODIFICATION-ID = *UNCHANGED / *ALL / <c-string 1..12 with-low> / *RANGE(...)
Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *UNCHANGED / *LOWEST / <c-string 1..12 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *UNCHANGED / *HIGHEST / <c-string 1..12 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

INFORMATION = *LOGICAL(...)
Es wird die logische Struktur des LLM ausgegeben.

LEVEL= *UNCHANGED / *ALL / *NEXT
Standardmäßig werden alle oder sonst nur die nächste Unterstruktur ausgegeben.

INFORMATION = *PHYSICAL
Es wird die physikalische Struktur des LLM ausgegeben.

Beispiel

Die Datei TEST1 soll als Element vom Typ D in die Bibliothek LIB3 aufgenommen werden.

Die standardmäßige Einstellung für den Elementtyp wird mit der Anweisung MODIFY-DEFAULTS auf den gewünschten Wert geändert. Dadurch ist es nicht mehr nötig, bei der nachfolgenden ADD-ELEMENT-Anweisung jeweils den Elementtyp anzugeben. Damit LMSCONV die erfolgreiche Aufnahme der Dateien meldet, wird durch die Anweisung MODIFY-LOGGING-PARAMETERS der Umfang der Protokollmenge auf das komplette LMSCONV-Protokoll gesetzt.

```
/start-lmsconv
//open-library lib3,*update
//modify-defaults type=d
//modify-log-param logging=*maximum
//add-element test1
INPUT  FILE
OUTPUT LIBRARY= :N:$USER.LIB3
    ADD TEST1   AS (D)TEST1/ (0001)/<date>
. . .
```

7.4.2.9 MODIFY-ELEMENT - Element verändern

Die Anweisung MODIFY-ELEMENT leitet die Modifikation von Elementen ein. Die Änderungen selber werden über Subanweisungen gesteuert.

Durch MODIFY-ELEMENT werden die Elemente ausgewählt, die geändert werden sollen.

Wurde die MODIFY-ELEMENT-Anweisung abgeschickt, erwartet LMSCONV als nächste Anweisung eine Subanweisung. Wird statt einer Subanweisung eine andere Anweisung eingegeben, wird eine Fehlermeldung ausgegeben.

Format

MODIFY**-ELEM**ENT

ELEMENT = ***LIB**RARY**-ELEM**ENT (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEF**AULT / <alphanum-name 1..8>

| ,**USER-DATE** =***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| **CRE**ATION-**DATE** =**_ANY_** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **_1900-01-01_** / <date 8..10 with-compl>

| | ,**TO** = **_*TODAY_** / <date 8..10 with-compl>

| ,**MOD**IFICATION-**DATE** =**_ANY_** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **_1900-01-01_** / <date 8..10 with-compl>

| | ,**TO** = **_*TODAY_** / <date 8..10 with-compl>

,**TO-E**LEMENT = **_*LIB_**RARY-**_ELEM_**ENT (...)

***LIB**RARY-**ELEM**ENT(...)

| **LIB**RARY = **_*STD_** / ***BY-SOUR**CE / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = **_*BY-SOUR_**CE (...) / <composed-name 1..132 with-under with-wildcard-constr>(...)

| ***BY-SOUR**CE(...)

| | **VERSION** = **_*DEFAULT_** / ***BY-SOUR**CE / ***UP**PER-**LIM**IT /

| | <composed-name 1..52 with-under with-wildcard-constr>

| <composed-name>(...)

| | **VERSION** = **_*DEFAULT_** / ***BY-SOUR**CE / ***UP**PER-**LIM**IT /

| | <composed-name 1..52 with-under with-wildcard-constr>

| ,**TYPE** = **_*BY-SOUR_**CE / ***DEF**AULT / <alphanum-name 1..8>

| ,**USER-DATE** = **_*TODAY_** / ***BY-SOUR**CE / <date 8..10 with-compl>

,**TEXT-PARAM**ETERS = **_*NONE_** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **INPUT-REC**ORD-**ID** = **_*NONE_** / ***REC**ORD-**PART**(...)

| ***REC**ORD-**PART**(...)

| | **START** = <integer 1..251>

| | ,**LEN**GTH = <integer 1..16>

,**WR**ITE-**MODE** = **_*DEFAULT_** / ***CREATE** / ***REPLACE** / ***ANY**

,**DIALOG-CONTR**OL = **_*DEFAULT_** / ***NO** / ***Y**ES / ***ERROR**

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich das zu ändernde Element befindet.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in der sich das zu ändernde Element befindet.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)

ELEMENT = *ALL(...)
Alle Elemente werden geändert.

ELEMENT = <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elements, welches geändert werden soll.

VERSION = *HIGHEST-EXISTING / *ALL/ *UPPER-LIMIT /
<composed-name 1..24 with-under with-wild(52)>
Version, die das zu ändernde Element besitzt.

VERSION = *HIGHEST-EXISTING
Das Element mit der höchsten bestehenden Version wird geändert.

VERSION = *ALL
Alle Versionen des Elements werden geändert.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird geändert.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Explizite Angabe der Version des Elements, das geändert werden soll.

TYPE = *DEFAULT / <alphanum-name 1..8>
Typ, den das zu ändernde Element besitzt. Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das zu ändernde Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird geändert.

USER-DATE = <date 8..10 with-compl>
Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird geändert.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden geändert.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der Erzeugung des Elements. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

TO-ELEMENT = *LIBRARY-ELEMENT(...)
Angabe, wohin und unter welchem Namen das korrigierte Element zurückgeschrieben werden soll.

LIBRARY = *STD / *BY-SOURCE / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in die das korrigierte Element zurückgeschrieben werden soll.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = *BY-SOURCE
Das korrigierte Element wird in die ursprüngliche Bibliothek zurückgeschrieben.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in die das korrigierte Element aufgenommen werden soll.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *BY-SOURCE(...) / <composed-name 1..132 with-under with-wildcard-constr>(...)
Name, den das korrigierte Element erhalten soll.

ELEMENT = *BY-SOURCE(...)
Das Zielelement erhält den Namen des Quellelementes.

VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /
<composed-name 1..52 with-under with-wildcard-constr>
Version, die das korrigierte Element erhalten soll.

VERSION = *DEFAULT
Der Standardwert ist *BY-SOURCE (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

VERSION = *BY-SOURCE
Das korrigierte Element erhält die gleiche Version wie das ursprüngliche Element. Hat das ursprüngliche Element keine Versionsangabe, erhält das korrigierte Element X'FF' als Versionsangabe.

VERSION = *UPPER-LIMIT
Die höchstmögliche Version X'FF' wird erzeugt.

VERSION = <composed-name 1..52 with-under with-wildcard-constr>
Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

ELEMENT = <composed-name 1..132 with-under with-wildcard-constr>(...)
Name des korrigierten Elements. Er kann auch mit Wildcards eingegeben werden.

VERSION = *DEFAULT / *BY-SOURCE / *UPPER-LIMIT /
<composed-name 1..52 with-under with-wildcard-constr>
Version, die das korrigierte Element erhalten soll.
Beschreibung der Operanden siehe oben.

TYPE = *BY-SOURCE / *DEFAULT / <alphanum-name 1..8>
Typ, den das korrigierte Element erhalten soll.

TYPE = *BY-SOURCE
Das korrigierte Element erhält die gleiche Typbezeichnung wie das ursprüngliche Element.

TYPE = *DEFAULT
Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *TODAY / *BY-SOURCE / <date 8..10 with-compl>
Datum, das vom Benutzer vergeben wird.

USER-DATE = *TODAY
Das aktuelle Tagesdatum wird vergeben.

USER-DATE = *BY-SOURCE
Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

USER-DATE = <date 8..10 with-compl>
Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

TEXT-PARAMETERS = *NONE / *PARAMETERS(...)
Legt Parameter für Textelemente fest.

TEXT-PARAMETERS = *NONE
Es werden keine Parameter für Textelemente festgelegt.

TEXT-PARAMETERS = *PARAMETERS(...)
Legt Parameter für Textelemente fest.

INPUT-RECORD-ID = *NONE / *RECORD-PART(...)
Legt den Bereich der Satzkennung (siehe Abschnitt „Subanweisungen von MODIFY-ELEMENT für Textelemente“) im Eingabesatz fest.

INPUT-RECORD-ID = *NONE
Es wird keine Satzkennung im Eingabesatz festgelegt.

INPUT-RECORD-ID = *RECORD-PART(...)
Beginn und Länge des Satzkennungsbereiches werden festgelegt.
Es muss gelten: Beginn + Länge <= 252.

START = <integer 1..251>
Beginn des Satzkennungsbereiches durch Angabe des ersten Zeichens im Satz.

LENGTH = <integer 1..16>
Länge der Satzkennung.

WRITE-MODE = *DEFAULT / *CREATE / *REPLACE / *ANY
Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt. Ist Quellelement = Zielelement so wird der Operand WRITE-MODE ignoriert.

WRITE-MODE = *DEFAULT
Der Standardwert ist *CREATE (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

WRITE-MODE = *CREATE
Der Name des korrigierten Elements darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Das korrigierte Element muss bereits existieren und wird dann ersetzt.

WRITE-MODE = *ANY
Das korrigierte Element wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-DEFAULTS.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES / *ERROR
Siehe die Beschreibung bei der Anweisung MODIFY-DEFAULTS.

7.4.3 Subanweisungen von MODIFY-ELEMENT für Elemente vom Basistyp R, C und L

Sie werden bis zur Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

Die Subanweisungen führen Änderungen in Bindemodulen, Phasen und Bindelademodulen durch.

LMSCONV sammelt zunächst die Subanweisungen und führt sie erst nach Eingabe der Subanweisung END-MODIFY aus.

Tabellarische Übersicht der LMSCONV-Subanweisungen

Die Subanweisungen sind abhängig vom gewählten Element-Typ und nur für Elemente der Typen R, C oder L erlaubt.

Anweisung	Elementtyp	Funktion
ADD-REP-RECORD	R	Erzeugen von REP-Sätzen
ADD-TEXT-MODIFICATION	R,C,L	Ändern von Textsätzen
DELETE-RECORD-TYPE	R,C,L	Löschen von Satzarten
END-MODIFY	R,C,L	Beenden der Änderungen
MODIFY-CSECT-ATTRIBUTES	R	Ändern der CSECT-Attribute
MODIFY-MODIFICATION-DEFAULTS	R,C,L	Festlegen der globalen Parameter in der MODIFY-ELEMENT-Anweisung
REMOVE-MODIFICATION	R,C,L	Zurücknehmen von Korrekturen
RENAME-SYMBOLS	R	Umbenennen von CSECT-, ENTRY-, EXTERN- und COMMON-Namen

Tabelle 9: MODIFY-ELEMENT-Subanweisungen

ℹ Als Subanweisungen sind auch die SDF-Standardanweisungen zulässig.

7.4.3.1 ADD-REP-RECORD - REP-Sätze in Bindemodul einfügen

Die Subanweisung ADD-REP-RECORD fügt REP-Sätze in den Bindemodul ein. Diese REP-Sätze wertet der Dynamische Bindelader (DBL) aus.

Die Subanweisung ADD-REP-RECORD ist nur für Bindemodule (Elemente des Types R) zulässig.

Format

ADD-REP-RECORD

ADDRESS = <x-string 1..8>(...)

<x-string 1..8>(...)

| **BASE-ADDR**ESS = ***MOD**IFICATION**-DEF**AULT / <x-string 1..8>

,**NEW-CONTEN**TS = <x-string 1..100> / <c-string 1..50 with-low>

Operanden

ADDRESS = <x-string 1..8>(...)
Angabe der Adresse, an der das durch MODIFY-ELEMENT gewählte Element geändert werden soll.

BASE-ADDRESS = *MODIFICATION-DEFAULT / <x-string 1..8>
Basisadresse.
BASE-ADDRESS wird auf ADDRESS addiert. Die so entstehende Korrekturadresse muss bei Großmoduln großmodul-relativ (nicht CSECT-relativ) sein. Die Voreinstellung von BASE-ADDRESS ist 0.

NEW-CONTENTS = <x-string 1..100> / <c-string 1..50 with-low>
Ersetzungstext, der zeichenweise oder sedezimal angegeben wird.
Wird der Text zeichenweise angegeben, darf er höchstens 50 Zeichen lang sein. Ein Hochkomma im Text muss doppelt angegeben werden.
Wird der Text sedezimal angegeben, darf er höchstens 100 Zeichen lang sein.

7.4.3.2 ADD-TEXT-MODIFICATION - Textsätze eines Bindemoduls korrigieren

Die Subanweisung ADD-TEXT-MODIFICATION korrigiert Textsätze eines Bindemoduls, eines Bindelademoduls und Phasen. Die Subanweisung erzeugt einen Korrekturjournalsatz (TXTP-Satz), der den ursprünglichen Inhalt des Textbereiches enthält.

Mit der Anweisung MODIFY-MODIFICATION-DEFAULTS kann eingestellt werden, dass kein Korrekturjournalsatz erzeugt wird. Korrekturen ohne Korrekturjournalsatz sind über die Subanweisung REMOVE-MODIFICATION nicht rückgängig zu machen.

Sie ist für Elemente der Typen R, C und L zulässig.

Format

ADD-TEXT-MODIFICATION

ADDRESS = <x-string 1..8>(...)

<x-string 1..8>(...)

| **BASE-ADD**RESS = ***MOD**IFICATION**-DEF**AULT / <x-string 1..8>

,**NEW-CONTEN**TS = <x-string 1..100>(...) / <c-string 1..50 with-low>(...)

<x-string 1..100>(...)

| **OLD-CONTEN**TS = ***ANY** / <x-string 1..100> / <c-string 1..50 with-low>

<c-string 1..50 with-low>(...)

| **OLD-CONTEN**TS = ***ANY** / <x-string 1..100> / <c-string 1..50 with-low>

,**MOD**IFICATION**-ID** = ***MOD**IFICATION**-DEF**AULT / ***SPACES** / <c-string 1..12 with-low>

Operanden

ADDRESS = <x-string 1..8>(...)
Angabe der Adresse, an der das durch MODIFY-ELEMENT gewählte Element geändert werden soll.

BASE-ADDRESS = *MODIFICATION-DEFAULT / <x-string 1..8>
Basisadresse. Die Voreinstellung von BASE-ADDRESS ist 0.
Die Basisadresse wird auf ADDRESS addiert. Die so entstehende Korrekturadresse ist:

bei ...	relativ zu:
Moduln	CSECT-Anfang (das gewünschte CSECT wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben.)
Phasen	Phasen-Anfang
LLMs	CSECT-Anfang, wenn ein CSECT angegeben wurde
	SUB-LLM-Anfang, wenn ein Sub-LLM angegeben wurde (der gewünschte Sub-LLM wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben)
	SLICE-Anfang, wenn eine SLICE angegeben wurde (die gewünschte SLICE wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben)
	LLM-Anfang, wenn nichts angegeben wurde und das LLM nur aus einem SLICE besteht. Besteht das LLM aus mehreren SLICES, ist die Angabe einer CSECT, eines Sub-LLM oder eines SLICES nötig.

Tabelle 10: Ermittlung der Basisadresse

NEW-CONTENTS = <x-string 1..100>(...) / <c-string 1..50 with-low>(...)
Ersetzungstext, der zeichenweise oder sedezimal angegeben wird.

OLD-CONTENTS = *ANY / <x-string 1..100> / <c-string 1..50 with-low>
Originaltext des Elementes. Der Originaltext muss immer in der Länge des Ersetzungstextes angegeben werden.

OLD-CONTENTS = *ANY
Es wird ein beliebiger Originaltext ersetzt.

MODIFICATION-ID = *MODIFICATION-DEFAULT / *SPACES / <c-string 1..12 with-low>
Identifikation, die im Änderungsjournalsatz (TXTP-Satz) gehalten wird. Wird SPACES angegeben, so werden als Identifikation Blanks verwendet. Die Voreinstellung sind Blanks oder der mit MODIFY-MODIFICATION-DEFAULTS eingestellte Wert.

Für Elementtyp R und C sind nur 8 Zeichen erlaubt.
Die Identifikation sollte fuer Typ R und C die Form 'Annnnnnn' und für Typ L die Form 'Annnnnnn-jjj' haben, wobei Annnnnnn eine Problemmeldungsnummer und jjj ein julianisches Datum ist.

7.4.3.3 DELETE-RECORD-TYPE - Satzarten aus Eingabeelement ausschließen

Die Subanweisung DELETE-RECORD-TYPE schließt aus dem Eingabeelement folgende Satzarten aus:

- ISD-Sätze (gilt nur für Typ R Elemente)
- LSD-Sätze (gilt nur für Typ R Elemente)
- REP-Sätze (gilt nur für Typ R Elemente)
- INCLUDE-Sätze (gilt nur für Typ R Elemente)
- TXTP-Sätze (gilt für Typ R, C und L Elemente)
- DSDD-Sätze (gilt nur für Typ R Elemente)

Sie ist für Elemente der Typen R, C und L zulässig.

Format

DELETE-**REC**ORD- **TYPE**
TYPE = ***TXTP**(...) / list-poss(5): ***ISD** / ***LSD** / ***REP** / ***DSDD** / ***INCL**UDE ***TXTP**(...) | **MOD**IFICATION-**ID** = ***ALL** / ***SPACES** / <c-string 1..12 with-low>

Operanden

TYPE = *TXTP(...) / list-poss(5): *ISD / *LSD / *REP / *DSDD / *INCLUDE
Legt die Satzart fest, die nicht vom Eingabeelement ins Ausgabeelement übernommen werden soll.

MODIFICATION-ID = *ALL / *SPACES / <c-string 1..12 with-low>
Es werden nur die TXTP-Sätze mit der angegebenen Identifizierung gelöscht. Für Elementtyp R und C sind nur 8 Zeichen erlaubt.
Diese Identifizierung gilt nur für dieses DELETE-RECORD-TYPE.

ℹ Gelöschte Satzarten sind unwiderruflich verloren.

7.4.3.4 END-MODIFY - Eingabe von Subanweisungen beenden

END-MODIFY schließt die Folge der Subanweisungen ab. Anschließend prüft LMSCONV alle Anweisungen auf Durchführbarkeit und führt die Anweisungsfolge aus.

Format

END-MODIFY

Diese Anweisung besitzt keine Operanden.

7.4.3.5 MODIFY-CSECT-ATTRIBUTES - Programmabschnittsmerkmale verändern

Die Subanweisung MODIFY-CSECT-ATTRIBUTES verändert Programmabschnittsmerkmale.

Sie ist nur für Bindemodule (Elemente des Types R) zulässig.

Zu Beginn der Anweisung MODIFY-ELEMENT sind die Operanden mit dem unmittelbar auf *UNCHANGED folgenden Wert voreingestellt.

Format

MODIFY-**CSECT-ATTR**IBUTES
NAME = ***ALL** / <c-string 1..8 with-low> / <text 1..8> ,**VISIBLE** = ***UNCHANGED** / ***Y**ES / ***NO** ,**READ-ONLY** = ***UNCHANGED** / ***Y**ES / ***NO** ,**PAGE-ALIGNMENT** = ***UNCHANGED** / ***Y**ES / ***NO** ,**RESIDENCY-MODE** = ***UNCHANGED** / 24 / ***ANY** ,**ADDRESSING-MODE** = ***UNCHANGED** / 24 / 31 / ***ANY**

Operanden

NAME = *ALL / <c-string 1..8 with-low> / <text 1..8>
Name der CSECT, deren Attribute geändert werden sollen. Es können alle CSECTs oder eine spezielle CSECT angegeben werden.

VISIBLE = *UNCHANGED / *YES / *NO
Maskierung (Sichtbarkeit) der Programmabschnitte. Voreingestellt ist der Wert *YES.

VISIBLE = *YES
Die angegebenen Programmabschnitte werden nicht maskiert (siehe Handbuch „Bindelader-Starter“ [13]). Für diese Abschnitte wird ein Sekundärnamenssatz angelegt und die Namen werden im Sekundärnamensverzeichnis eingetragen.

VISIBLE = *NO
Die angegebenen Programmabschnitte werden maskiert. Für sie wird weder ein Sekundärnamenssatz angelegt noch werden die Namen im Sekundärnamensverzeichnis eingetragen. Ein eventuell vorhandener Sekundärnamenssatz wird gelöscht.

Werden alle Programmabschnitte eines Bindemoduls maskiert, wird ein Bibliothekselement ohne Sekundärnamenseintrag erzeugt. Dieser Bindemodul ist nur über Primärnamen auffindbar.

Der Modulname kann jedoch aus dem ersten Programmabschnittsnamen mithilfe aller ESD-Sätze abgeleitet werden, da dazu auch maskierte Programmabschnitte verwendet werden.

> ¡ Bindemodule, die nur maskierte Programmabschnitte besitzen, kann der Binder nicht bearbeiten, z.B. Ausschließen eines Bindemoduls bei der Autolink-Funktion. Der Operand VISIBLE ist auch auf ENTRYs anwendbar.

READ-ONLY = *UNCHANGED / *YES / *NO
Legt den Schreibschutz fest. Voreingestellt ist der Wert *YES.

READ-ONLY = *YES
Legt fest, dass die angegebenen Programmabschnitte zur Ablaufzeit des Programms nur gelesen werden dürfen.

READ-ONLY = *NO
Erlaubt, dass während des Programmlaufs auch in die angegebenen Programmabschnitte geschrieben werden darf.

PAGE-ALIGNMENT = *UNCHANGED / *YES / *NO
Legt die Seitenausrichtung fest. Voreingestellt ist der Wert *YES.

PAGE-ALIGNMENT = *YES
Legt fest, dass die angegebenen Programmabschnitte auf Seitengrenze ausgerichtet werden sollen, d.h. die Ladeadresse soll ein Vielfaches von dezimal 4096 bzw. sedezimal 1000 sein.

PAGE-ALIGNMENT = *NO
Lässt Seitengrenzen unberücksichtigt. Die Programmabschnitte beginnen jeweils bei der nächsten Doppelwortadresse, die sich beim Binden ergibt.

RESIDENCY-MODE = *UNCHANGED / 24 / *ANY
Legt den Lademodus fest. Voreingestellt ist der Wert 24.

RESIDENCY-MODE = 24
Legt fest, dass die angegebenen Programmabschnitte in den Adressraum unterhalb der 16 MB Grenze zu laden sind.

RESIDENCY-MODE = *ANY
Es existiert keine Einschränkung.

ADDRESSING-MODE = *UNCHANGED / 24 / 31 / *ANY
Legt den Ablaufmodus fest. Voreingestellt ist der Wert 24.

ADDRESSING-MODE = 24
Legt fest, dass die angegebenen Programmabschnitte im 24-Bit-Modus ablauffähig sein sollen.

ADDRESSING-MODE = 31
Legt fest, dass die angegebenen Programmabschnitte im 31-Bit-Modus ablauffähig sein sollen.

ADDRESSING-MODE = *ANY
Der Ablaufmodus ist beliebig.

7.4.3.6 MODIFY-MODIFICATION-DEFAULTS - Globale Voreinstellungen festlegen

Die Subanweisung MODIFY-MODIFICATION-DEFAULTS legt innerhalb der MODIFY-ELEMENT-Anweisung die globalen Voreinstellungen fest.

Sie ist für Elemente der Typen R, C und L zulässig.

Zu Beginn der Anweisung MODIFY-ELEMENT sind die Operanden mit dem unmittelbar auf *UNCHANGED folgenden Wert voreingestellt.

Format

MODIFY**-MOD**IFICATION**-DEF**AULTS

CSECT-NAME = ***UNCHANGED** / ***NONE** / <c-string 1..32 with-low> / <text 1..32>

,**PHASE-SEG**MENT = ***UNCHANGED** / ***ROOT** / <name 1..8>

,**LLM-PART** = ***UNCHANGED** / ***NONE** / ***SLICE**(...) / ***SUB-LLM**(...)

***SLICE**(...)

| **NAME** = <structured-name 1..32>

***SUB-LLM**(...)

| **PATH-NAME** = <c-string 1..255 with-low> / <text 1..255>

,**MOD**IFICATION**-LOG**GING = ***UNCHANGED** / ***Y**ES(...) / ***NO**

***Y**ES(...)

| **MOD**IFICATION**-ID** = ***UNCHANGED** / ***SPACES** / <c-string 1..12 with-low>

,**BASE-ADD**RESS = ***UNCHANGED** / <x-string 1..8>

Operanden

CSECT-NAME = *UNCHANGED / *NONE / <c-string 1..32 with-low> / <text 1..32>
Name der zu korrigierenden CSECT. (Nur relevant für Typ R und L). Voreingestellt ist der Wert *NONE.

CSECT-NAME = *NONE
Wird kein CSECT-Name angegeben, wird bei R-Moduln der erste CSECT-Name verwendet.

PHASE-SEGMENT = *UNCHANGED / *ROOT / <name 1..8>
Angabe des Phasen-Segmentes, das korrigiert werden soll. Wird kein Segment angegeben, wird das erste Segment (*ROOT) verwendet. Voreingestellt ist der Wert *ROOT.

LLM-PART = *UNCHANGED / *NONE / *SLICE(...) / *SUB-LLM(...)
Wird kein LLM-Teil angegeben, wird der gesamte LLM genommen. Voreingestellt ist der Wert *NONE.

LLM-PART = *SLICE(...)
Angabe der SLICE, die korrigiert werden soll.

NAME = <structured-name 1..32>
Name der SLICE, die korrigiert werden soll.

LLM-PART = *SUB-LLM(...)
Angabe des SUB-LLM, der korrigiert werden soll.

PATH-NAME = <c-string 1..255 with-low> / <text 1..255>
Der zu korrigierende SUB-LLM wird durch seinen Pfadnamen bestimmt.

MODIFICATION-LOGGING = *UNCHANGED / *YES(...) / *NO
Bestimmt die Erzeugung der TXTP-Sätze. Voreingestellt ist der Wert *YES.

MODIFICATION-LOGGING = *YES(...)
Es sollen TXTP-Sätze erzeugt werden.

MODIFICATION-ID = *UNCHANGED / *SPACES / <c-string 1..12 with-low>
Identifikation, die im Änderungsjournalsatz (TXTP-Satz) gehalten wird. Wird SPACES angegeben, so werden als Identifikation Blanks verwendet. Voreingestellt ist der Wert *SPACES.

Für Elementtyp R und C sind nur 8 Zeichen erlaubt.
Die Identifikation sollte fuer Typ R und C die Form 'Annnnnnn' und für Typ L die Form 'Annnnnnn-jjj' haben, wobei Annnnnnn eine Problemmeldungsnummer und jjj ein julianisches Datum ist.

MODIFICATION-LOGGING = *NO
Es sollen keine TXTP-Sätze erzeugt werden.

BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
Hexadezimale Angabe der Basisadresse. Zu Beginn der Anweisung MODIFY-ELEMENT ist die Basisadresse 0 eingestellt.

7.4.3.7 REMOVE-MODIFICATION - Korrekturen rückgängig machen

Die Subanweisung REMOVE-MODIFICATION macht Korrekturen aus einem früheren Korrekturlauf unter den folgenden Voraussetzungen rückgängig:

Es wurde ein Korrekturjournalsatz mit der Subanweisung ADD-TEXT-MODIFICATION erstellt, d.h. der Operand MODIFICATION-LOGGING =*YES war gesetzt.

Sie ist für Elemente der Typen R, C und L zulässig.

Format

REMOVE**-MOD**IFICATION
MODIFICATION-ID = **<u>*ALL</u>** / ***SPACES** / <c-string 1..12 with-low>

Operanden

MODIFICATION-ID = <u>*ALL</u> / *SPACES / <c-string 1..12 with-low>
Für Elementtyp R und C sind nur 8 Zeichen erlaubt.
Es werden nur die Korrekturen mit der angegebenen Identifikation rückgängig gemacht. Wird eine Identifikation angegeben, so müssen dafür Korrekturjournalsätze existieren.

MODIFICATION-ID = <u>*ALL</u>
Wird keine Identifikation angegeben, werden alle Korrekturen, für die ein Korrekturjournalsatz vorliegt, rückgängig gemacht.

7.4.3.8 RENAME-SYMBOLS - Symbole umbenennen

Die Subanweisung RENAME-SYMBOLS ändert den Namen von einer CSECT, einem ENTRY, EXTRN oder COMMON. Jede Umbenennung verursacht eine Änderung von ESD-Sätzen. LMSCONV prüft die Eindeutigkeit der Namen innerhalb aller ESD-Sätze und weist eine Umbenennung zurück, wenn der neue Name bereits existiert.

Die Subanweisung RENAME-SYMBOLS ist nur für Bindemodule (Elemente des Types R) zulässig.

Format

RENAME**-SYMB**OLS
SYMBOL-NAME = <text 1..8> **,SYMBOL-TYPE** = ***CSECT** / ***ENTRY** / ***EXTRN** / ***COMMON** **,NEW-NAME** = <text 1..8>

Operanden

SYMBOL-NAME = <text 1..8>
Legt den Symbolnamen fest, der umbenannt werden soll.

SYMBOL-TYPE = *CSECT / *ENTRY / *EXTERN / *COMMON
Legt den Typ des Symbols fest, dessen Name geändert werden soll.

NEW-NAME = <text 1..8>
Neuer Symbolname.
Der Name sollte den Binder-Lader-Starter-Konventionen für den Sonderdatentyp <symbol> genügen (siehe Handbuch „Binder“ [13]). Allerdings prüft LMSCONV nicht auf diese Konvention.

> ¡ Auch maskierte (invisible) CSECT/ENTRY-Namen können umbenannt werden.

7.4.4 Subanweisungen von MODIFY-ELEMENT für Textelemente

Die Subanweisungen führen Änderungen in Textelementen durch. Sie werden bis zur Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

Tabellarische Übersicht der LMSCONV-Subanweisungen

Anweisung	Funktion
ADD-RECORD	Einfügen von Sätzen
END-MODIFY	Beenden der Änderungen
REMOVE-RECORD	Ausfügen von Sätzen

Tabelle 11: MODIFY-ELEMENT-Subanweisungen für Textelemente

> Als Subanweisungen sind auch die SDF-Standardanweisungen zulässig.
> Es werden nur Elementsätze der Länge <= 251 bearbeitet. Längere Sätze werden abgeschnitten. In diesem Fall gibt LMSCONV eine Warnung aus.

Definition der Satzidentifikation für Textelemente

Die Satzidentifikation kann eine Satznummer oder eine Satzkennung sein.

Satznummer: Die Satznummer ist die relative Position des Elementsatzes bezogen auf den Elementanfang. Ist die angegebene Satznummer größer als die höchste Satznummer des Elementes, werden die Änderungen nach dem letzten Elementsatz fortgesetzt, d.h. Sätze an das Element angefügt.

Satzkennung: Lage und Länge der Satzkennung werden mit dem Operanden INPUT-RECORD-ID (siehe Anweisung MODIFY-ELEMENT) festgelegt.
Deshalb darf die Angabe einer Satzkennung in den Subanweisungen nur bei INPUT-RECORD-ID ungleich *NONE verwendet werden. Die Satzkennung muss in der Länge, die bei INPUT-RECORD-ID vereinbart wurde, angegeben werden. Nur führende Nullen dürfen entfallen. Wenn die Satzkennung im Eingabeelement nicht vorkommt, wird vor dem ersten Satz mit größerer Satzkennung modifiziert.

Satznummer und Satzkennung können in den Subanweisungen gemischt angegeben werden. Sie müssen in den Subanweisungen und Datensätzen immer aufsteigend angegeben werden.

Wurde im Dialog ein Fehler erkannt, muss die Korrektur mit END-MODIFY beendet und die Korrektur neu gestartet werden. Nach einer ADD-RECORD-Anweisung muss zusätzlich ein *END eingegeben werden.

7.4.4.1 ADD-RECORD - Sätze einfügen

Die Subanweisung ADD-RECORD fügt die der Anweisung folgenden Sätze an gewünschter Position ein. Die einzufügenden Sätze müssen mit einem *END Satz abgeschlossen werden.

Format

ADD-RECORD
RECORD**-ID** = **<u>*NONE</u>** / <integer 0..99999999> / <c-string 1..16 with-low>

Operanden

RECORD-ID = <u>*NONE</u> / <integer 0..99999999> / <c-string 1..16 with-low>
Legt die Record-ID fest, hinter der Datensätze eingefügt werden sollen.

RECORD-ID = <u>*NONE</u>
Ist der Operand INPUT-RECORD-ID der MODIFY-ELEMENT-Anweisung ungleich *NONE, werden die der ADD-RECORD-Anweisung folgenden Datensätze gemäß ihrer Satzkennung in das zu modifizierende Element aufgenommen.
Gibt es einen Satz mit der angegebenen Satzkennung, wird er durch den Datensatz ersetzt. Gibt es keinen Satz mit der angegebenen Satzkennung, wird der Datensatz vor dem ersten Satz mit höherer Satzkennung eingefügt. Datensätze mit leerer Satzkennung, oder wenn INPUT-RECORD-ID=*NONE gesetzt ist, werden an aktueller Position eingeordnet.

RECORD-ID = <integer 0..99999999> / <c-string 1..16 with-low>
Angabe der Elementposition, hinter der die der Anweisung folgenden Datensätze eingefügt werden sollen. Ist die angegebene Satznummer oder Satzkennung nicht vorhanden, werden die Datensätze vor dem ersten Satz eingefügt, dessen Satznummer/Satzkennung größer ist als die angegebene Satznummer/Satzkennung.

7.4.4.2 END-MODIFY - Subanweisungen abschließen

END-MODIFY schließt die Folge der Subanweisungen ab.

Format

END-MODIFY

Diese Anweisung besitzt keine Operanden.

7.4.4.3 REMOVE-RECORD - Satz oder Satzbereich in Element löschen

Die Subanweisung REMOVE-RECORD löscht den angegebenen Satz oder Satzbereich im Element.

Format

REMOVE**-REC**ORD

RECORD**-ID** = <integer 0..99999999> / <c-string 1..16 with-low> / ***RANGE**(...)

***RANGE**(...)

| **FROM** = <integer 0..99999999> / <c-string 1..16 with-low>

| ,**TO** = <integer 0..99999999> / <c-string 1..16 with-low>

Operanden

RECORD-ID = <integer 0..99999999> / <c-string 1..16 with-low> / *RANGE(...)
Satznummer oder Satzkennung des zu löschenden Satzes.

RECORD-ID = <integer 0..99999999> / <c-string 1..16 with-low>
Satznummer oder Satzkennung des zu löschenden Satzes.

RECORD-ID = *RANGE(...)
Der zu löschende Satzbereich wird bestimmt.

FROM = <integer 0..99999999> / <c-string 1..16 with-low>
Beginn des Bereiches durch Angabe der ersten Satznummer oder Satzkennung.

TO = <integer 0..99999999> / <c-string 1..16 with-low>
Ende des Bereiches durch Angabe der letzten Satznummer oder Satzkennung.

7.4.5 LMSCONV-Anweisungen von MODIFY-ELEMENT-ATTRIBUTES bis WRITE-COMMENT

- MODIFY-ELEMENT-ATTRIBUTES - Elementattribute verändern
- MODIFY-LOGGING-PARAMETERS - Protokolleinstellungen ändern
- OPEN-LIBRARY - Globale Bibliothek öffnen
- SHOW-DEFAULTS - Aktuelle Belegung der Voreinstellungen ausgeben
- SHOW-ELEMENT - Elementinhalt anzeigen
- SHOW-ELEMENT-ATTRIBUTES - Elementeigenschaften anzeigen
- SHOW-LIBRARY-ATTRIBUTES - Attribute einer Bibliothek anzeigen
- SHOW-LIBRARY-STATUS - Status der Bibliothek anzeigen
- SHOW-LOGGING-PARAMETERS - Globale LMSCONV-Parameter anzeigen
- SHOW-TYPE-ATTRIBUTES - Attribute eines Elementtyps anzeigen
- SHOW-USER-EXITS - LMSCONV-Version anzeigen
- WRITE-COMMENT - Kommentare in Ausgabemedium schreiben

7.4.5.1 MODIFY-ELEMENT-ATTRIBUTES - Elementattribute verändern

Die Anweisung MODIFY-ELEMENT-ATTRIBUTES verändert den Coded-Character-Set-Namen (CCSN) eines Elements.

Format

MODIFY-**ELEM**ENT-**ATTR**IBUTES

ELEMENT = ***LIBRARY-ELEM**ENT(...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEF**AULT / ***ALL** / <alphanum-name 1..8 with-wild(20)>

| ,**USER-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**CRE**ATION**-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**MOD**IFICATION**-DATE** = ***ANY** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **1900-01-01** / <date 8..10 with-compl>

| | ,**TO** = ***TODAY** / <date 8..10 with-compl>

| ,**CODED-CHAR**ACTER-**SET** = ***ANY** / ***NONE** / <name 1..8 with-wild(20)>

,**NEW-ATTR**IBUTES = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **CODED-CHAR**ACTER-**SET** = ***BY-SOUR**CE / ***LIB**RARY-**DEF**AULT / ***NONE** / <name 1..8>

,**DIALOG-CONTR**OL = ***DEFAULT** / ***NO** / ***Y**ES / ***ERROR**

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich das Element befindet.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in der sich das Element befindet.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek; dieser Name wurde mit /ADD-FILE-LINK vor dem Aufruf von LMSCONV vereinbart.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)
Angabe der Elemente, deren Attribute verändert werden sollen.

VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT /
<composed-name 1..24 with-under with-wild(52)>
Version des Elements.

VERSION = *HIGHEST-EXISTING
Die Eigenschaften des Element mit der höchsten bestehenden Version werden verändert.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird verwendet.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Explizite Angabe der Version des Elementes.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Typ des Elements.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das auszugebende Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird ausgewählt.

USER-DATE = <date 8..10 with-compl>
Das Element wird ausgewählt, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden verwendet.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Erzeugungsdatum des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

CODED-CHARACTER-SET = *ANY / *NONE / <name 1..8 with-wild(20)>
Zeichensatz, der dem Element zugeordnet ist.

CODED-CHARACTER-SET = *ANY
Es werden die Elemente unabhängig von dem zugeordneten Zeichensatz ausgewählt.

CODED-CHARACTER-SET = *NONE
Es werden die Elemente, denen kein Zeichensatz zugeordnet ist, ausgewählt.

CODED-CHARACTER-SET = <name 1..8 with-wild(20)>
Die Elemente, denen der angegebene Zeichensatz zugeordnet ist, werden ausgewählt.

NEW-ATTRIBUTES = *PARAMETERS(...)
Bestimmt die Attribute, die die ausgewählten Elemente erhalten sollen.

CODED-CHARACTER-SET = *BY-SOURCE / *LIBRARY-DEFAULT / *NONE / <name 1..8>
Zeichensatz der den Elementen zugewiesen wird.

CODED-CHARACTER-SET = *BY-SOURCE
Den Elementen wird der Zeichensatz der Quell-Elemente zugewiesen.

CODED-CHARACTER-SET = *LIBRARY-DEFAULT
Den Elementen wird der Zeichensatz ihrer Bibliothek zugewiesen.

CODED-CHARACTER-SET = *NONE
Den Elementen wird kein Zeichensatz zugewiesen.

CODED-CHARACTER-SET = <name 1..8>
Explizite Angabe des Zeichensatzes, der den Elementen zugewiesen wird.

DIALOG-CONTROL = *DEFAULT / *NO / *YES / *ERROR
Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-DEFAULTS.

DIALOG-CONTROL = *DEFAULT
Der Standardwert ist *NO oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

DIALOG-CONTROL = *NO
Alle Elemente werden ohne Nachfrage, d.h. ohne Eingreifmöglichkeit des Benutzers, bearbeitet.
Ausnahme: Ist ein Element oder eine Bibliothek gesperrt, fragt LMSCONV nach, ob der Zugriffsversuch wiederholt werden soll.

DIALOG-CONTROL = *YES / *ERROR
Siehe die Beschreibung bei der Anweisung MODIFY-DEFAULTS.

Hinweis

Im Gegensatz zum CCSN einer Datei wird der NETCCSN, also der Name des Zeichensatzes von SAM-Node-Files auf Net-Storage, als Dateiattribut gespeichert. Er kann daher mit MODIFY-ELEMENT-ATTRIBUTES nicht geändert werden.

7.4.5.2 MODIFY-LOGGING-PARAMETERS - Protokolleinstellungen ändern

Die Anweisung MODIFY-LOGGING-PARAMETERS ändert die Einstellung für die Protokollmenge, das Ausgabemedium und das Protokollformat.

Wird einer dieser Werte durch die MODIFY-LOGGING-PARAMETERS-Anweisung verändert, wird diese neue Einstellung zur aktuellen Einstellung. Diese bleibt solange für den LMSCONV-Lauf gültig (*UNCHANGED), bis eine neue Anweisung MODIFY-LOGGING-PARAMETERS für diesen Wert gegeben wird. Zu Beginn des LMSCONV-Laufs gelten die unmittelbar auf *UNCHANGED folgenden Werte.

Format

MODIFY**-LOG**GING**-PAR**AMETERS

LOGGING = ***UNCHANGED** / ***MIN**IMUM / ***MAX**IMUM

,**TEXT-OUT**PUT = ***UNCHANGED** / ***SYSOUT** / ***SYSLST**(...) / ***NONE** / ***LIB**RARY**-ELEM**ENT(...)

***SYSLST**(...)

| **SYSLST-NUM**BER = ***STD** / <integer 1..99>

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = <composed-name 1..64 with-under>(...)

| <composed-name 1..64 with-under>(...)

| | **VERSION** = ***UPPER-LIMIT** / <composed-name 1..24 with-under>

| ,**TYPE** = **P** / <alphanum-name 1..8>

| ,**WR**ITE**-MODE** = ***UNCHANGED** / ***CREATE** / ***REPLACE** / ***EXTEND** / ***ANY**

,**OUT**PUT**-LAYOUT** = ***UNCHANGED** / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **LINE**S**-P**ER**-PAGE** = ***UNCHANGED** / <integer 1..9999>

| ,**LINE-SIZE** = ***UNCHANGED** / 132 / 80

| ,**EXTRA-F**ORM-**F**EED = ***UNCHANGED** / ***NO** / ***Y**ES

| ,**HEAD**ER**-LINE**S = ***UNCHANGED** / ***Y**ES / ***NO**

Operanden

LOGGING = *UNCHANGED / *MINIMUM / *MAXIMUM
Festlegung der LMSCONV-Protokollmenge.

LOGGING = *MINIMUM
Es werden nur Fehlermeldungen und Misserfolgsmeldungen ausgegeben.

LOGGING = *MAXIMUM
Es wird ein komplettes LMSCONV-Protokoll ausgegeben.

TEXT-OUTPUT = *UNCHANGED / *SYSOUT / *SYSLST(...) / *NONE / *LIBARARY-ELEMENT(...)
Dieser Parameter bestimmt das Ausgabemedium. Bei Medienwechsel und bei WRITE-MODE=*EXTEND beginnt die Seitennummerierung immer mit 1. Voreingestellt ist *SYSOUT.

TEXT-OUTPUT = *SYSOUT
Die Ausgabe wird nach SYSOUT geschrieben.

TEXT-OUTPUT = *SYSLST(...)
Die Ausgabe wird nach SYSLST geschrieben.

SYSLST-NUMBER = *STD / <integer 1..99>
Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

SYSLST-NUMBER = *STD
Es gilt die Systemdatei SYSLST.

SYSLST-NUMBER = <integer 1..99>
Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

TEXT-OUTPUT = *NONE
Die Ausgabe wird bis auf Fehlermeldungen unterdrückt.

TEXT-OUTPUT = *LIBRARY-ELEMENT(...)
Die Ausgabe wird in ein Bibliothekselement gespeichert.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in die die Ausgabe gespeichert werden soll. Es wird entweder standardmäßig die global durch OPEN-LIBRARY eingestellte Bibliothek oder die explizit angegebene oder die durch den Dateikettungsnamen zugewiesene Bibliothek verwendet.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = <composed-name 1..64 with-under>(...)
Angabe des Elementes, in das die Ausgabe gespeichert werden soll.

VERSION = *UPPER-LIMIT / <composed-name 1..24 with-under>
Angabe der Version, die das Element erhalten soll.

VERSION = *UPPER-LIMIT
Die höchstmögliche Version X'FF' wird erzeugt.

VERSION = <composed-name 1..24 with-under>
Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

TYPE = P / <alphanum-name 1..8>
Angabe des Elementtypes.
Standardmäßig erhält das Element, in das die Ausgabe gespeichert wird, den Typ P für druckaufbereitete Dateien.

WRITE-MODE = *UNCHANGED / *CREATE / *REPLACE / *EXTEND / *ANY
Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt. Voreingestellt ist *CREATE.

WRITE-MODE = *CREATE
Das Zielelement darf noch nicht existieren und wird neu erzeugt.

WRITE-MODE = *REPLACE
Das Zielelement muss bereits existieren und wird dann ersetzt.

WRITE-MODE = *EXTEND
Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt.

WRITE-MODE = *ANY
Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

OUTPUT-LAYOUT = *UNCHANGED / *PARAMETERS(...)
Dieser Parameter bestimmt das LMSCONV-Protokollformat.

LINES-PER-PAGE = *UNCHANGED / <integer 1..9999>
Dieser Parameter legt die Seitenlänge fest.
Default-Wert: 64 Zeilen

LINE-SIZE = *UNCHANGED / 132 / 80
Dieser Parameter legt die Zeilenlänge fest.Voreingestellt ist 132.

LINE-SIZE = 132
Die Zeile soll 132 Zeichen lang sein.

LINE-SIZE = 80
Die Zeile soll 80 Zeichen lang sein.

EXTRA-FORM-FEED = *UNCHANGED / *NO / *YES
Dieser Parameter steuert einen Extra-Seitenvorschub. Voreingestellt ist *NO.

EXTRA-FORM-FEED = *NO
Es gibt nur dann einen Seitenvorschub, wenn die Seite voll ist.

EXTRA-FORM-FEED = *YES
Es gibt einen Seitenvorschub, wenn entweder die Seite voll ist oder wenn ein Anweisungswechsel oder Elementwechsel stattfindet.

HEADER-LINES = *UNCHANGED / *YES / *NO
Dieser Parameter steuert die Ausgabe von Überschriften.Voreingestellt ist *YES.

HEADER-LINES = *YES
Es werden Überschriften mit der Bibliotheks- und Elementbezeichnung ausgegeben.

HEADER-LINES = *NO
Es werden keine Überschriften ausgegeben.

7.4.5.3 OPEN-LIBRARY - Globale Bibliothek öffnen

Mit OPEN-LIBRARY wird eine globale Bibliothek festgelegt und eröffnet. Diese wird in den anderen Anweisungen mit LIBRARY =*STD angesprochen.

Werden in einer Anweisung zwei Bibliotheken benötigt, so muss die 2. Bibliothek bei der Anweisung explizit oder über Dateikettungsnamen angegeben werden.

Globale Bibliotheken bleiben solange zugewiesen, bis sie explizit durch die CLOSE-LIBRARY-Anweisung geschlossen werden oder eine neue OPEN-LIBRARY-Anweisung gegeben wird.

Globale Bibliotheken werden standardmäßig nur zum Lesen eröffnet. Sollen sie zum Lesen und Schreiben eröffnet werden, muss der Operand MODE=*UPDATE gesetzt werden.Wird eine Bibliothek neu eingerichtet, muss sie mit MODE=*UPDATE erzeugt werden.

Format

OPEN-LIBRARY

LIBRARY = <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,**MODE** = ***READ** / ***UPD**ATE(...)

***UPD**ATE(...)

| **STATE** = ***ANY** / ***OLD** / ***NEW**

Operanden

LIBRARY = <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, die als globale Bibliothek eingerichtet und eröffnet werden soll.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, die als globale Einstellung eingerichtet und eröffnet werden soll.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek wird als globale Bibliothek eingerichtet und eröffnet.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der vor dem Aufruf von LMSCONV mit /ADD-FILE-LINK vereinbart wurde.

MODE = *READ / *UPDATE(...)
Öffnungsmodus der Bibliothek.

MODE = *READ
Die Bibliothek wird nur zum Lesen eröffnet. Sie muss bereits existieren.

MODE = *UPDATE(...)
Die Bibliothek wird zum Lesen und Schreiben eröffnet.

STATE = *ANY / *OLD / *NEW
Status der zu eröffnenden Bibliothek.

STATE = *ANY
Die Bibliothek kann existieren. Existiert sie nicht, so wird sie neu angelegt.

STATE = *OLD
Die Bibliothek muss existieren.

STATE = *NEW
Die Bibliothek darf nicht existieren. Sie wird neu angelegt.

Beispiele

Zuweisen einer existierenden Bibliothek LIB1 als globale Bibliothek:

```
//open-library library=lib1
```

Zuweisen einer existierenden Bibliothek als globale Bibliothek über den Dateikettungsnamen:

```
/add-file-link link-name=glob-lib,file-name=lib1
//start-lmsconv
. . .
//open-library library=*link(link-name=glob-lib)
```

Neuanlegen einer Bibliothek mit Zuweisung als globale Bibliothek:

```
//open-library library=lib1,mode=*update
```

7.4.5.4 SHOW-DEFAULTS - Aktuelle Belegung der Voreinstellungen ausgeben

Diese Anweisung gibt die aktuelle Belegung der LMSCONV-Default-Werte aus. Sie können mit der Anweisung MODIFY-DEFAULTS geändert werden.

Format

SHOW-DEFAULTS
DEFAULTS = **<u>*STD</u>** / ***ALL** / list-poss(2000): ***ELEMENT-ATTR**IBUTES / ***F**ILE**-ATTR**IBUTES / ***DESTROY-DATA** / ***WR**ITE**-MODE** / ***DIALOG-CONTR**OL / ***INF**ORMATION / ***LAYOUT** / ***SORT** / ***OUT**PUT**-FORM** / ***DEL**ETE**-SOUR**CE / ***MAX-ERROR-WEIGHT** / ***RUN-MODE** / ***NEXT-ATTEM**PT / ***TEXT-INF**ORMATION / ***MOD**ULE**-INF**ORMATION / ***PHASE-INF**ORMATION / ***LLM-INF**ORMATION

Operanden

DEFAULTS = <u>*STD</u>
Es werden die Default-Werte Element-Attributes, File-Attributes, Destroy-Data, Write-Mode, Dialog-Control, Information, Layout, Sort, Output-Form, Delete-Source, Max-Error-Weight, Next-Attempt und Protection mit ihrer aktuellen Belegung ausgegeben.

DEFAULTS = *ALL
Es werden alle Default-Werte mit ihrer aktuellen Belegung ausgegeben.

DEFAULTS = *ELEMENT-ATTRIBUTES
Es werden die aktuelle Belegung für Elementtyp, Quell- und Zielversion, Speicherungsform und die Dateiattribute ausgegeben.

DEFAULTS = *FILE-ATTRIBUTES
Es wird die aktuelle Belegung der Dateizugriffsmethode ausgegeben.

DEFAULTS = *DESTROY-DATA
Es wird ausgegeben, ob die Daten überschrieben werden oder nicht.

DEFAULTS = *WRITE-MODE
Es wird die aktuelle Belegung des Schreibmodus ausgegeben.

DEFAULTS = *DIALOG-CONTROL
Es wird die aktuelle Belegung der Dialogführung ausgegeben.

DEFAULTS = *INFORMATION
Es wird die aktuelle Belegung für den Umfang des auszugebenden Inhaltsverzeichnisses angezeigt.

DEFAULTS = *LAYOUT
Es wird die aktuelle Belegung für das Format des auszugebenden Inhaltsverzeichnisses angezeigt.

DEFAULTS = *SORT
Es wird die aktuelle Belegung für das Sortierkriterium des auszugebenden Inhaltsverzeichnisses angezeigt.

DEFAULTS = *OUTPUT-FORM
Es wird die aktuelle Belegung für das Darstellungsformat der Ausgabe angezeigt.

DEFAULTS = *DELETE-SOURCE
Es wird ausgegeben, ob die ursprüngliche Datei erhalten oder gelöscht werden soll. Es wird die aktuelle Belegung der Schutzübernahme ausgegeben.

DEFAULTS = *MAX-ERROR-WEIGHT
Es wird die aktuelle Belegung der Spin-off Steuerung ausgegeben.

DEFAULTS = *RUN-MODE
Es wird die aktuelle Belegung für den Ablaufmodus ausgegeben, in dem LMSCONV aufgerufen werden soll.

DEFAULTS = *NEXT-ATTEMPT
Es wird die aktuelle Belegung der Steuerung der Öffnungsversuche ausgegeben.

DEFAULTS = *TEXT-INFORMATION
Es wird die aktuelle Belegung der Informationsmenge für textartige Elemente ausgegeben.

DEFAULTS = *MODULE-INFORMATION
Es wird die aktuelle Belegung der Informationsmenge für Bindemodule ausgegeben.

DEFAULTS = *PHASE-INFORMATION
Es wird die aktuelle Belegung der Informationsmenge für Phasen ausgegeben.

DEFAULTS = *LLM-INFORMATION
Es wird die aktuelle Belegung der Informationsmenge für Bindelademodule ausgegeben.

Beispiel

```
//show-defaults   *std
ELEMENT-ATTRIBUTES
     TYPE              = *NONE
     ELEMENT-VERSION   = *ALL
     TO-ELEM-VERSION   = *BY-SOURCE
     STORAGE-FORM      = *STD
     SOURCE-ATTRIBUTES = *STD
FILE-ATTRIBUTES
     ACCESS-METHOD     = *ISAM
DESTROY-DATA           = *NO
WRITE-MODE             = *CREATE
DIALOG-CONTROL         = *NO
INFORMATION            = *MEDIUM
LAYOUT                 = *VARIABLE
SORT                   = *BY-NAME
OUTPUT-FORM            = *STD
DELETE-SOURCE          = *NO
PROTECTION             = *STD
MAX-ERROR-WEIGHT       = *SERIOUS
RUN-MODE               = *STD
NEXT-ATTEMPT           = *NO
```

Es werden alle LMSCONV-Default-Werte ausgegeben. Diese Einstellungen sind direkt nach Start von LMSCONV gültig.

7.4.5.5 SHOW-ELEMENT - Elementinhalt anzeigen

SHOW-ELEMENT zeigt den Elementinhalt abhängig vom Typ des angegebenen Elementes. Es können die Inhalte von textartigen Elementen, Moduln, Phasen und Bindelademoduln ausgegeben werden. Das Darstellungsformat der Ausgabe wird über den Operanden OUTPUT-FORM gesteuert. Die Bedeutung der Attribute bei Moduln und Bindelademoduln ist dem Handbuch „BINDER“ [12] zu entnehmen.

Die Anweisung ist für alle Elementtypen zulässig. Benutzertypen werden dabei entsprechend ihrem Basistyp behandelt. Ist der Basistyp LMSCONV nicht bekannt, so wirken nur die Operanden TEXT-INFORMATION und OUTPUT-FORM. Für textartige Elemente kann die Informationsmenge eingegrenzt werden.

Format

SHOW-ELEMENT

ELEMENT = ***LIB**RARY**-ELEM**ENT (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***HIGHEST-EXIST**ING / ***ALL** / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEF**AULT / ***ALL** / <alphanum-name 1..8 with-wild(20)>

| ,**USER-DATE** =**<u>*ANY</u>** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **<u>1900-01-01</u>** / <date 8..10 with-compl>

| | ,**TO** = **<u>*TODAY</u>** / <date 8..10 with-compl>

| ,**CRE**ATION**-DATE** =**<u>*ANY</u>** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **<u>1900-01-01</u>** / <date 8..10 with-compl>

| | ,**TO** = **<u>*TODAY</u>** / <date 8..10 with-compl>

| ,**MOD**IFICATION**-DATE** =**<u>*ANY</u>** / ***TODAY** / <date 8..10 with-compl> / ***INTERVAL**(...)

| ***INTERVAL**(...)

| | **FROM** = **<u>1900-01-01</u>** / <date 8..10 with-compl>

| | ,**TO** = **<u>*TODAY</u>** / <date 8..10 with-compl>

,TEXT-INFORMATION = **<u>*DEFAULT</u>** / ***ALL** / ***STATIS**TICS / ***F**ILE**-ATTR**IBUTES / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **INF**ORMATION = **<u>*DEFAULT</u>** / ***ALL** / list-poss(2): ***TEXT** / ***COMMENT**

| ,**REC**ORD**-RANGE** = **<u>*DEFAULT</u>** / ***ALL** / ***RANGE**(...)

| ***RANGE**(...)

| | **FROM** = **<u>*DEFAULT</u>** / <integer 1..2147483647>

| | ,**TO** = **<u>*DEFAULT</u>** / ***LAST** / <integer 1..2147483647>

| ,**REC**ORD**-PART** = **<u>*DEFAULT</u>** / ***ALL** / ***PART**(...)

| ***PART**(...)

| | **START** = **<u>*DEFAULT</u>** / <integer 1..32764>

| | ,**LEN**GTH = **<u>*DEFAULT</u>** / ***REST** / <integer 1..32764>

| ,**REC**ORD**-NUM**BER = **<u>*DEFAULT</u>** / ***BY-OUT**PUT / ***Y**ES / ***NO**

,**MOD**ULE**-INF**ORMATION = **<u>*DEFAULT</u>** / ***ALL** / ***STATIS**TICS / ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **INF**ORMATION = **<u>*DEFAULT</u>** / ***ALL** / ***TXT**(...) / ***TXTP**(...) / list-poss(9): ***ESD** / ***ISD** /

| ***LSD** / ***RLD** / ***REP** / ***INCL**UDE / ***DSDD** / ***REF** / ***END**

| ***TXT**(...)

| | **CSECT**-NAME = **<u>*DEFAULT</u>** / ***ALL** / <c-string 1..32 with-low> / <text 1..32>

```
        |           |   ,ADDRESS = *DEFAULT (...) / <x-string 1..8>(...)
        |           |       *DEFAULT(...)
        |           |           |   BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |           |       <x-string 1..8>(...)
        |           |           |   BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |           |   ,LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
        |       *TXTP(...)
        |           |   MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)
        |           |       *RANGE(...)
        |           |           |   FROM = *DEFAULT / *LOWEST / <c-string 1..8 with-low>
        |           |           |   ,TO = *DEFAULT / *HIGHEST / <c-string 1..8 with-low>
,PHASE-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
    *PARAMETERS(...)
        |   SEGMENT = *DEFAULT / *ALL / *ROOT / <name 1..8>
        |   ,INFORMATION = *DEFAULT / *ALL / *TXT(...) / *TXTP(...) / list-poss(4): *ESD / *ISD /
        |                  *LSD / *RLD
        |       *TXT(...)
        |           |   ADDRESS = *DEFAULT (...) / <x-string 1..8>(...)
        |           |       *DEFAULT(...)
        |           |           |   BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |           |       <x-string 1..8>(...)
        |           |           |   BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |           |   ,LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
        |       *TXTP(...)
        |           |   MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)
        |           |       *RANGE(...)
        |           |           |   FROM = *DEFAULT / *LOWEST / <c-string 1..8 with-low>
        |           |           |   ,TO = *DEFAULT / *HIGHEST / <c-string 1..8 with-low>
```

```
,LLM-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
    *PARAMETERS(...)
        |  LLM-PART = *DEFAULT / *ALL / *SLICE(...) / *SUB-LLM(...)
        |      *SLICE(...)
        |          |  NAME = *DEFAULT / <structured-name 1..32>
        |      *SUB-LLM(...)
        |          |  PATH-NAME = *DEFAULT / <c-string 1..255 with-low> / <text 1..255>
        |  ,INFORMATION = *DEFAULT / *ALL / *TXT(...) / *TXTP(...) / *LOGICAL(...) / *PHYSICAL / *REF /
        |                          list-poss(4): *RELOCATION / *ESVD / *ESVR / *LRLD
        |      *TXT(...)
        |          |  CSECT-NAME = *DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
        |          |  ,ADDRESS = *DEFAULT (...) / <x-string 1..8>(...)
        |          |      *DEFAULT(...)
        |          |          |  BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |          |      <x-string 1..8>(...)
        |          |          |  BASE-ADDRESS = *DEFAULT / <x-string 1..8>
        |          |  ,LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
        |      *TXTP(...)
        |          |  CSECT-NAME = *DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
        |          |  ,MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..12 with-low> / *RANGE(...)
        |          |      *RANGE(...)
        |          |          |  FROM = *DEFAULT / *LOWEST / <c-string 1..12 with-low>
        |          |          |  ,TO = *DEFAULT / *HIGHEST / <c-string 1..12 with-low>
        |      *LOGICAL(...)
        |          |  LEVEL = *DEFAULT / *ALL / *NEXT
,OUTPUT-FORM = *DEFAULT / *STD / *CHARACTER / *HEXADECIMAL / *DUMP
```

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich das Element befindet.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, in der sich das Element befindet.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)

ELEMENT = *ALL(...)
Es werden Informationen zu allen Elementen ausgegeben.

ELEMENT = <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elementes, welches angezeigt werden soll.

VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT
<composed-name 1..24 with-under with-wild(52)>
Version, die das auszugebende Element besitzt.

VERSION = *HIGHEST-EXISTING
Das Element mit der höchsten bestehenden Version wird ausgegeben.

VERSION = *UPPER-LIMIT
Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird angezeigt.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Explizite Angabe der Version des Elementes, das angezeigt werden soll.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Typ, den das auszugebende Element besitzt. Ist der Wert *DEFAULT und der mit MODIFY-DEFAULTS aktuell eingestellte Wert gleich *NONE, so verlangt LMSCONV die Angabe eines Typs.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVALL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das auszugebende Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Das Element mit dem aktuellen Tagesdatum wird angezeigt.

USER-DATE = <date 8..10 with-compl>
Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird angezeigt.

USER-DATE = *INTERVAL(...)
Alle Elemente, die in dem angegebenen Zeitraum liegen, werden angezeigt.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der Erzeugung des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

TEXT-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *FILE-ATTRIBUTES / *PARAMETER(...)
Legt die Informationsmenge für alle Elemente außer den Elementtypen R, C und L fest. Der Standardwert ist *ALL (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

TEXT-INFORMATION = *ALL
Es wird alles ausgegeben.

TEXT-INFORMATION = *STATISTICS
Es wird die Anzahl der Sätze pro Satzart und die Summe der Sätze ausgegeben. Pro Satzart wird die Summe der Satzlängen (ohne Satzlängenfelder) ausgeben und deren Summe über alle Satzarten.

TEXT-INFORMATION = *FILE-ATTRIBUTES
Es werden nur die gespeicherten Dateiattribute ausgegeben.

> Für PAM-Elemente (mit Ausnahme der Elemente vom Typ L) enthält die Ausgabe den Last Byte Pointer (LBP), wenn er als Dateiattribut gespeichert worden war. Für SAM-Node-Files enthält die Ausgabe den Zeichensatz auf Net-Storage (NETCCSN) wenn er als Dateiattribut gespeichert worden war.

TEXT-INFORMATION = *PARAMETERS(...)
Es wird ein Elementausschnitt festgelegt, der ausgegeben werden soll.

INFORMATION = *DEFAULT / *ALL / list-poss(2): *TEXT / *COMMENT
Der anzuzeigende Teilbereich des Elementes. Der Standardwert ist *ALL (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *ALL
Es werden alle Anwendersatzarten angezeigt.

INFORMATION = *TEXT
Es wird der eigentliche Text d.h. Satzart 1 angezeigt.

INFORMATION = *COMMENT
Es wird der separat gespeicherte Kommentar d.h. Satzart 2 angezeigt.

RECORD-RANGE = *DEFAULT / *ALL / *RANGE(...)
Der zu bearbeitende Teilbereich des Elementes. Der Standardwert ist *ALL (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

RECORD-RANGE = *ALL
Es werden alle Anwendersatzarten bearbeitet.

RECORD-RANGE = *RANGE(...)
Der zu bearbeitende Satznummernbereich wird bestimmt. Die Satznummern beziehen sich nicht auf eine Satzart, sondern auf den durch INFORMATION= festgelegten Teilbereich des Elementes. Innerhalb dieses Teilbereichs werden die Sätze von 1 bis n durchnummeriert.

FROM = *DEFAULT / <integer 1..2147483647>
Beginn des Bereichs durch Angabe der ersten Satznummer. Standardmäßig wird die Satznummer 1 genommen.

TO = *DEFAULT / *LAST / <integer 1..2147483647>
Ende des Bereichs durch Angabe der letzten Satznummer. Standardmäßig wird die letzte Satznummer genommen.

RECORD-PART = *DEFAULT / *ALL / *PART(...)
Der zu bearbeitende Satzbereich wird bestimmt.

RECORD-PART = *ALL
Es wird der ganze Satz bearbeitet.

RECORD-PART = *PART(...)
Der zu bearbeitende Satzbereich wird bestimmt. Werden die Default-Werte nicht verändert, wird der ganze Satz bearbeitet.

START = *DEFAULT / <integer 1..32764>
Beginn des Bereichs durch Angabe des ersten Zeichens im Satz. Standardmäßig wird das erste Zeichen genommen.

LENGTH = *DEFAULT / *REST / <integer 1..32764>
Länge des Bereichs. Standardmäßig wird der Rest des Satzes genommen.

RECORD-NUMBER = *DEFAULT / *BY-OUTPUT / *YES / *NO
Bestimmt die Ausgabe der Satznummern. Der Standardwert ist *BY-OUTPUT (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

RECORD-NUMBER = *BY-OUTPUT
Nur wenn die Ausgabe nach SYSOUT erfolgt, werden keine Satznummern ausgegeben. Bei jedem anderen Ausgabemedium werden sie mit ausgegeben.

RECORD-NUMBER = *YES
Die Satznummern werden auch nach SYSOUT ausgegeben.

RECORD-NUMBER = *NO
Es werden keine Satznummern mit ausgegeben.

MODULE-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
Legt die Informationsmenge für Bindemodule (Elemente vom Typ R) fest. Der Standardwert ist *ALL (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

MODULE-INFORMATION = *ALL
Es wird alles ausgegeben.

MODULE-INFORMATION = *STATISTICS
Es werden Name, Länge und Adresse der CSECTS sowie die Gesamtlänge des Moduls ausgegeben.

MODULE-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

INFORMATION = *DEFAULT / *ALL / *TXT(...) / *TXTP(...) / list-poss(9): *ESD / *ISD / *LSD / *RLD / *REP / *INCLUDE / *DSDD / *REF / *END
Die hier aufgezählten Satzarten können ausgewählt werden. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

CSECT-NAME = *DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die Textsätze können auf eine CSECT eingeschränkt werden. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

ADDRESS = *DEFAULT / <x-string 1..8>(...)
Anfangsadresse des Textes. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

BASE-ADDRESS = *DEFAULT / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell
eingestellte Wert.

LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes. Der Standardwert ist *REST oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *DEFAULT / *LOWEST / <c-string 1..8 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *DEFAULT / *HIGHEST / <c-string 1..8 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

PHASE-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *PARAMETER(...)
Legt die Informationsmenge für Phasen (Elemente vom Typ C) fest. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

PHASE-INFORMATION = *ALL
Es wird alles ausgegeben.

PHASE-INFORMATION = *STATISTICS
Es werden Name, Länge und Adresse des Segmentes sowie die Gesamtlänge des Segmentes ausgegeben.

PHASE-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

SEGMENT = *DEFAULT / *ALL / *ROOT / <name 1..8>
Phasensegment, das ausgewählt wird. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *DEFAULT / *ALL / *TXT(...) / *TXTP(...) / list-poss(4): *ESD / *ISD / *LSD / *RLD
Die hier aufgezählten Satzarten können ausgewählt werden. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

ADDRESS = *DEFAULT / <x-string 1..8>(...)
Anfangsadresse des Textes. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

BASE-ADDRESS = *DEFAULT / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell
eingestellte Wert.

LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes. Der Standardwert ist *REST oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)
Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *DEFAULT / *LOWEST / <c-string 1..8 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *DEFAULT / *HIGHEST / <c-string 1..8 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

LLM-INFORMATION = *DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
Legt die Informationsmenge für Bindelademodule (Elemente vom Typ L) fest. Der Standardwert ist *ALL (s.u.) oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

LLM-INFORMATION = *ALL
Es wird alles ausgegeben.

LLM-INFORMATION = *STATISTICS
Allgemeine Informationen über das Bindelademodule (Name, Copyright,..) werden ausgegeben.

LLM-INFORMATION = *PARAMETERS(...)
Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

LLM-PART = *DEFAULT / *ALL / *SLICE(...) / *SUB-LLM(...)
Angabe des LLM-Teils, das ausgewählt werden soll. Standardmäßig wird der gesamte LLM ausgewählt.

LLM-PART = *SLICE(...)
Angabe der SLICE, die ausgegeben werden soll.

NAME = <structured-name 1..32>
Name der SLICE, die ausgegeben werden soll.

LLM-PART = *SUB-LLM(...)
Angabe des SUB-LLM, der ausgegeben werden soll.

PATH-NAME = *DEFAULT / <c-string 1..255 with-low> / <text 1..255>
Der auszugebende SUB-LLM wird durch seinen Pfadnamen bestimmt. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *DEFAULT / *ALL / *TXT(...) / *TXTP(...) / *LOGICAL(...) / *PHYSICAL / *REF / list-poss (4): *ESVD / *ESVR / *LRLD / *RELOCATION
Die hier aufgezählten Satzarten können ausgewählt werden. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXT(...)
Es werden Textsätze ausgewählt.

CSECT-NAME = *DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die Textsätze können auf eine CSECT eingeschränkt werden.

ADDRESS = *DEFAULT / <x-string 1..8>(...)
Anfangsadresse des Textes. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

BASE-ADDRESS = *DEFAULT / <x-string 1..8>
Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert. Der Standardwert ist X'00000000' oder der mit MODIFY-DEFAULTS aktuell
eingestellte Wert.

LENGTH = *DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
Länge des Textes. Der Standardwert ist *REST oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *TXTP(...)
TXTP-Sätze werden ausgegeben.

CSECT-NAME = *DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
Die TXTP-Sätze können auf eine CSECT eingeschränkt werden. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

MODIFICATION-ID = *DEFAULT / *ALL / <c-string 1..12 with-low> / *RANGE(...)
Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

MODIFICATION-ID = *RANGE(...)
Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

FROM = *DEFAULT / *LOWEST / <c-string 1..12 with-low>
Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

TO = *DEFAULT / *HIGHEST / <c-string 1..12 with-low>
Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

INFORMATION = *LOGICAL(...)
Es wird die logische Struktur des LLM ausgegeben.

LEVEL = *DEFAULT / *ALL / *NEXT
Standardmäßig werden alle oder sonst nur die nächste Unterstruktur ausgegeben.

INFORMATION = *PHYSICAL
Es wird die physikalische Struktur des LLM ausgegeben.

OUTPUT-FORM = *DEFAULT / *STD / *CHARACTER / *HEXADECIMAL / *DUMP
Festlegung der Darstellungsformates für die Ausgabe. Der Standardwert ist *STD oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

OUTPUT-FORM = *STD
Das Darstellungsformat wird abhängig vom Typ des Elementes gewählt.

OUTPUT-FORM = *CHARACTER
Die Ausgabe wird alphanumerisch dargestellt.

OUTPUT-FORM = *HEXADECIMAL
Die Ausgabe wird alphanumerisch und hexadezimal übereinander dargestellt.

OUTPUT-FORM = *DUMP
Die Ausgabe wird alphanumerisch und hexadezimal nebeneinander dargestellt. Für die Elementtypen S, P, D, J und M wirkt dieser Operand wie *HEXADECIMAL.

Beispiele

Das Element LETTER.A, das den Text 'Dear ...' enthält, soll aufgelistet werden.

```
//show-element (element=letter.a,type=d)
INPUT  LIBRARY= :N:$USER.TEST.LIB
INPUT  ELEMENT= (D)LETTER.A/ (0001)/<date>
Dear ...
...
...
Yours sincerely, ...
NUMBER OF PROCESSED RECORDS IS    123
```

Die gespeicherten Dateiattribute eines Elements mit LBP sollen ausgegeben werden.

```
//show-element (element=net.lbp.2dd,type=x),text-information=file-attributes
INPUT  LIBRARY= :SQGB:$TSOS.BIB.ALI
INPUT  ELEMENT= (X)PTF04/§(0001)/<date>
ORIGINAL FILE ATTRIBUTES :
FILENAME= :CK33:$TSOS.PTF04
FILSIZE = 0000009   FCBTYPE = PAM       2ND ALLO= 00006     LBP     = 13312
SHARE   = NO        ACCESS  = WRITE
BLKCTRL = DATA      BLKSIZE = 014336    RECFORM = U         RECSIZE = 00000
PERFORM = STANDARD  USAGE   = NOT-SPEC
```

Die gespeicherten Dateiattribute eines Elements mit NETCCSN sollen ausgegeben werden.

```
//show-element (element=stf03,type=x),text-information=file-attributes
INPUT  LIBRARY= :SQGB:$TSOS.BIB.ALI
INPUT  ELEMENT= (X)STF03/@(0001)/<date>
ORIGINAL FILE ATTRIBUTES :
FILENAME= :CK33:$DMS01.STF03
```

```
FILSIZE = 0000032   FCBTYPE = SAM       2ND ALLO= 00016
SHARE   = NO        ACCESS  = WRITE
BLKCTRL = DATA      BLKSIZE = 032768    RECFORM = V         RECSIZE = 00000
PERFORM = STANDARD  USAGE   = NOT-SPEC  NETCCSN = ISO88591
```

7.4.5.6 SHOW-ELEMENT-ATTRIBUTES - Elementeigenschaften anzeigen

SHOW-ELEMENT-ATTRIBUTES gibt die Inhaltsverzeichniseinträge der angegebenen Elemente oder der gesamten Bibliothek aus. Die Ausgabe erfolgt auf das durch MODIFY-LOGGING-PARAMETERS eingestellte Ausgabemedium.

Das Inhaltsverzeichnis wird immer nach dem Typ sortiert ausgegeben. Die weitere Sortierreihenfolge bestimmt der Operand SORT. Standardmäßig gilt die Sortierreihenfolge Typ, Name und Version.

Durch die Operanden INFORMATION und LAYOUT wird der Umfang und das Format der Ausgabe des Inhaltsverzeichnisses festgelegt. Standardmäßig wird Typ, Name, Version, Variantennummer und Datum ausgegeben.

Mithilfe der Operanden SECONDARY-NAME und -ATTRIBUTE kann das Inhaltsverzeichnis auf die Elemente beschränkt werden, die einen bestimmten Referenzeintrag enthalten.

> ¡ Um den gesamten Inhalt einer Bibliothek zu erhalten (alle Elemente mit allen Versionen), genügt die Angabe von SHOW-ELEMENT-ATTRIBUTES ohne Angabe von Operanden, sofern über MODIFY-DEFAULTS kein bestimmter Elementtyp bzw. keine bestimmte Elementversion festgelegt wurde.

Format

SHOW-ELEMENT**-ATTR**IBUTES

ELEMENT = ***LIB**RARY**-ELEM**ENT (...)

***LIB**RARY**-ELEM**ENT(...)

| **LIB**RARY = ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

| ***LINK**(...)

| | **LINK**-NAME = <structured-name 1..8>

| ,**ELEM**ENT = ***ALL** (...) / <composed-name 1..64 with-under with-wild(132)>(...)

| ***ALL**(...)

| | **VERSION** = ***DEFAULT** / ***ALL** / ***HIGH**EST**-EXIST**ING / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| <composed-name 1..64 with-under with-wild(132)>(...)

| | **VERSION** = ***DEFAULT** / ***ALL** / ***HIGH**EST**-EXIST**ING / ***UP**PER**-LIM**IT /

| | <composed-name 1..24 with-under with-wild(52)>

| ,**TYPE** = ***DEF**AULT / ***ALL** / <alphanum-name 1..8 with-wild(20)>

```
        |   ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 1900-01-01 / <date 8..10 with-compl>
        |           |   ,TO = *TODAY / <date 8..10 with-compl>
        |   ,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 1900-01-01 / <date 8..10 with-compl>
        |           |   ,TO = *TODAY / <date 8..10 with-compl>
        |   ,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 1900-01-01 / <date 8..10 with-compl>
        |           |   ,TO = *TODAY / <date 8..10 with-compl>
        |   ,USER-TIME = *ANY / <time 1..8> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 00:00:00 / <time 1..8>
        |           |   ,TO = 23:59:59 / <time 1..8>
        |   ,CREATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 00:00:00 / <time 1..8>
        |           |   ,TO = 23:59:59 / <time 1..8>
        |   ,MODIFICATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
        |       *INTERVAL(...)
        |           |   FROM = 00:00:00 / <time 1..8>
        |           |   ,TO = 23:59:59 / <time 1..8>
        |   ,CODED-CHARACTER-SET = *ANY / *NONE / <name 1..8> / <composed-name 1..8 with-wild(20)>
        |   ,SECONDARY-NAME = *ANY / <alphanum-name 1..32 with-wild(68)>
        |   ,SECONDARY-ATTRIBUTE = *ANY / *CSECT / *ENTRY
,INFORMATION = *DEFAULT / *MEDIUM / *MINIMUM / *MAXIMUM / *SUMMARY
,LAYOUT = *DEFAULT / *VARIABLE / *FIXED
```

,**SORT** = **<u>*DEFAULT</u>** / ***BY-NAME** / ***BY-VERS**ION / ***BY-USER-DATE** / ***BY-SEC**ONDARY**-NAME**

Operanden

ELEMENT = *LIBRARY-ELEMENT(...)
Angaben für die gewünschte Elementbezeichnung.

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, deren Inhaltsverzeichnis ausgegeben wird.

LIBRARY = *STD
Die durch OPEN-LIBRARY eröffnete Bibliothek.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, deren Inhaltsverzeichnis ausgegeben wird.

LIBRARY = *LINK(...)
Die über den Dateikettungsnamen zugewiesene Bibliothek.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek.

ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)
Name des Elementes, dessen Bibliothekseintrag ausgegeben wird.
Wird der Default-Wert '*ALL' angegeben, werden die Bibliothekseinträge aller Elemente passend zur Version und zum Typ ausgegeben.

VERSION = *DEFAULT / *ALL / *HIGHEST-EXISTING / *UPPER-LIMIT /<composed-name 1..24 with-under with-wild(52)>
Version, die das Element besitzt. Der Standardwert ist *ALL oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

VERSION = *ALL
Es werden die Bibliothekseinträge aller oben ausgewählten Elemente ausgegeben, unabhängig von ihrer Version.

VERSION = *HIGHEST-EXISTING
Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der höchsten bestehenden Version ausgegeben.

VERSION = *UPPER-LIMIT
Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der Version X'FF' ausgegeben.

VERSION = <composed-name 1..24 with-under with-wild(52)>
Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der hier angegebenen Version ausgegeben.

TYPE = *DEFAULT / <alphanum-name 1..8 with-wild(20)>
Typ, den das Element besitzt.
Ist der LMSCONV-Standardwert für TYPE *NONE, so wirkt *DEFAULT wie *ALL.

USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum, das vom Benutzer vergeben wurde.

USER-DATE = *ANY
Das Element besitzt ein beliebiges Datum.

USER-DATE = *TODAY
Die Bibliothekseinträge aller Elemente mit dem aktuellen Tagesdatum werden ausgegeben.

USER-DATE = <date 8..10 with-compl>
Es werden die Bibliothekseinträge aller Elemente mit dem hier angegebenen Datum in der Form [JJ]JJ-MM-TT ausgegeben.

USER-DATE = *INTERVAL(...)
Es werden die Bibliothekseinträge aller Elemente, die in dem angegebenen Zeitraum liegen, ausgegeben.

FROM = 1900-01-01 / <date 8..10 with-compl>
Beginn des Zeitraums.

TO = *TODAY / <date 8..10 with-compl>
Ende des Zeitraums.

CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Erzeugungsdatum des Elementes. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
Datum der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-DATE dieser Anweisung.

USER-TIME = *ANY / <time 1..8> / *INTERVAL(...)
Uhrzeit, die vom Benutzer vergeben wurde.

USER-TIME = *ANY
Es werden die Bibliothekseinträge aller Elemente unabhängig von der Uhrzeit ausgegeben.

USER-TIME = <time 1..8>
Es werden die Bibliothekseinträge aller Elemente mit der hier angegebenen Uhrzeit in der Form HH:MM:SS ausgegeben.

USER-TIME = *INTERVAL(...)
Es werden die Bibliothekseinträge aller Elemente, die in dem angegebenen Zeitraum liegen, ausgegeben.

FROM = 00:00:00 / <time 1..8>
Beginn des Zeitraums.

TO = 23:59:59 / <time 1..8>
Ende des Zeitraums.

CREATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
Uhrzeit der Erzeugung des Elementes. Beschreibung der Operandenwerte siehe Operand USER-TIME dieser Anweisung.

MODIFICATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)

Uhrzeit der letzten Änderung für das Element. Beschreibung der Operandenwerte siehe Operand USER-TIME dieser Anweisung.

CODED-CHARACTER-SET = *ANY / *NONE / <name 1..8> /
<composed-name 1..8 with-wild(20)>
Zeichensatz, der dem Element zugeordnet ist.

CODED-CHARACTER-SET = *ANY
Es werden die Bibliothekseinträge aller Elemente unabhängig von dem zugeordneten Zeichensatz ausgewählt.

CODED-CHARACTER-SET = *NONE
Es werden die Bibliothekseinträge aller Elemente, denen kein Zeichensatz zugeordnet ist, ausgewählt.

CODED-CHARACTER-SET = <name 1..8> / <composed-name 1..8 with-wild(20)>
Es werden die Bibliothekseinträge aller Elemente, denen der angegebene Zeichensatz zugeordnet ist, ausgewählt.

SECONDARY-NAME = *ANY / <alphanum-name1...32 with-wild(68) >
Sekundärname.
Wird hier eine Angabe ungleich *ANY gemacht, erfolgt die Auswahl über das Sekundärinhaltsverzeichnis der Bibliothek. Bei der Angabe von Wildcards werden nur die ersten 32 Zeichen der Sekundärnamen zur Bestimmung der Auswahl herangezogen.

SECONDARY-ATTRIBUTE = *ANY / *CSECT / *ENTRY
Sekundärattribut.
Wird hier eine Angabe ungleich *ANY gemacht, erfolgt die Auswahl über das Sekundärinhaltsverzeichnis der Bibliothek.

INFORMATION = *DEFAULT / *MEDIUM / *MINIMUM / *MAXIMUM / *SUMMARY
Dieser Parameter bestimmt den Umfang des auszugebenden Inhaltsverzeichnisses. Er bestimmt auch den Umfang der strukturierten Ausgabe (siehe „Parameterabhängigkeiten“). Der Standardwert ist *MEDIUM oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

INFORMATION = *MEDIUM
Es wird Typ, Name, Version, Variantennummer und das Benutzerdatum ausgegeben.

INFORMATION = *MINIMUM
Es wird nur Typ, Name und Version ausgegeben.

INFORMATION = *MAXIMUM
Es wird die gesamte Information ausgegeben.

INFORMATION = *SUMMARY
Es wird nur die Anzahl der Elemente pro Typ ausgegeben.

LAYOUT = *DEFAULT / *VARIABLE / *FIXED
Dieser Parameter bestimmt das Format des auszugebenden Inhaltsverzeichnisses. Der Standardwert ist *VARIABLE oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

LAYOUT = *VARIABLE
Die Anzahl der Druckspalten ist abhängig von der längsten Elementbezeichnung innerhalb eines Elementtypes.

LAYOUT = *FIXED
Das Inhaltsverzeichnis wird einspaltig im festen Format gedruckt. Eine Spalte bedeutet die untereinander stehenden Einträge im Inhaltsverzeichnis.

SORT = *DEFAULT / *BY-NAME / *BY-VERSION / *BY-USER-DATE / *BY-SECONDARY-NAME
Sortierkriterium für die Inhaltsverzeichniseinträge der ausgewählten Elemente. Als erstes Sortierkriterium wird immer der Typ verwendet. Der Standardwert ist *BY-NAME oder der mit MODIFY-DEFAULTS aktuell eingestellte Wert.

SORT = *BY-NAME
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Name und Version.

SORT = *BY-VERSION
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Version und Name.

SORT = *BY-USER-DATE
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Benutzerdatum, Name und Version.

SORT = *BY-SECONDARY-NAME
Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Sekundärname, Sekundärattribut, Name und Version.

> ¡ Wird der Operand SECONDARY-NAME bzw. -ATTRIBUTE angegeben und ist dieser Operand ungleich *ANY, wird zusätzlich im Inhaltsverzeichnis eine Überschrift ausgegeben, die über den Sekundärnamen und das Sekundärattribut informiert. Die Sekundärnamen werden aber nicht in der maximalen Länge angezeigt.

Parameterabhängigkeiten

Zwischen den Operanden INFORMATION, SORT und LAYOUT gibt es folgende Abhängigkeiten:

- Das Benutzerdatum beeinflusst nur dann die Sortierung, wenn es auch im Inhaltsverzeichnis ausgegeben werden soll. Dazu darf nicht der Operand INFORMATION =*MINIMUM eingestellt sein.
- Der Operand LAYOUT wirkt nur, wenn INFORMATION =*MEDIUM/*MINIMUM angegeben ist. Bei allen anderen Einstellungen wird das Inhaltsverhaltsverzeichnis immer im festen Format ausgegeben.
- Wird INFORMATION =*MAXIMUM angegeben, ist die Information pro Element zu lang für eine Zeile. Sie wird dann in einem Format unabhängig von der Layout-Steuerung ausgegeben.
- Die Ausgabe des Inhaltsverzeichnisses wird beschleunigt, wenn INFORMATION=*MINIMUM gesetzt wird und die Auswahl sich auf die Operanden ELEMENT, VERSION und TYPE beschränkt. Alle anderen Operanden sollten mit dem Wert *ANY eingestellt sein.

- Die Operanden INFORMATION (außer INFORMATION=*SUMMARY) und SORT beeinflussen die Sortierreihenfolge des Inhaltsverzeichnisses.

  Die folgende Tabelle zeigt diese Abhängigkeiten:

SORT	INFORMATION	
	*MINIMUM	*MEDIUM / *MAXIMUM
*BY-NAME	1. Typ 2. Name 3. Version	1. Typ 2. Name 3. Version
*BY-VERSION	1. Typ 2. Version 3. Name	1. Typ 2. Name 3. Version
*BY-USER-DATE	1. Typ 2. Name 3. Version	1. Typ 2. Benutzerdatum 3. Name 4. Version
*BY-SECONDARY-NAME mit Referenz[1)]	1. Typ 2. Sekundärname 3. Sekundärattribut 4. Name 5. Version	1. Typ 2. Sekundärname 3. Sekundärattribut 4. Name 5. Version
*BY-SECONDARY-NAME ohne Referenz[2)]	1. Typ 2. Name 3. Version	1. Typ 2. Name 3. Version

Tabelle 12: Sortierung und Information bei SHOW-ELEMENT-ATTRIBUTES

[1)] mit Referenz bedeutet, dass bei der Elementauswahl entweder der Sekundärname oder das Sekundärattribut ungleich dem Default-Wert *ANY angegeben wurde, d.h.:

SECONDARY-NAME = <alphanum-name...> und/oder SECONDARY-ATTRIBUTE = *CSECT oder *ENTRY

[2)] ohne Referenz bedeutet, dass bei der Elementauswahl sowohl kein Sekundärname als auch kein Sekundärattribut angegeben wurde

Beispiel

Ausgeben des Inhaltsverzeichnisses der Bibliothek USER.LIB. Die Bibliothek enthält genau ein Element, welches mit all seinen Attributen angezeigt wird.

```
//SHOW-ELEMENT-ATTRIBUTES -
(LIBRARY=USER.BSPLIB,ELEMENT=*(VERSION=*)),INFORMATION=*MAXIMUM
INPUT LIBRARY= :N:$USER.USER.BSPLIB
TYPE = D
NAME = TEST
VERSION = @ VARIANT = 0001
--------------------------------GENERAL-----------------------------------
ELEM-SIZE = 12
STORAGEW = *FULL
STATE = *IN-HOLD HOLDER = MUBF
--------------------------------HISTORY-----------------------------------
USER-DATE = <date> CRE-DATE = <date> MOD-DATE = <date>
USER-TIME = <time> CRE-TIME = <time> MOD-TIME = <time>
ACC-DATE = <date>
ACC-TIME = <time>
--------------------------------SECURITY----------------------------------
READ-PASS = *NONE READ-USER = *OWNER *GROUP -
WR-PASS = *NONE WR-PASS = *OWNER - -
```

7.4.5.7 SHOW-LIBRARY-ATTRIBUTES - Attribute einer Bibliothek anzeigen

Diese Anweisung gibt alle für die Bibliothek eingestellten Attribute aus. Dies sind u.a.:

- Bibliotheksgröße in 2K-Einheiten
- Anzahl der freien 2K-Einheiten (können durch Umkopieren beseitigt werden)
- Bibliotheksformat (NK2/NK4)
- UPAM-geschützt (Y/N)

Format

SHOW-LIBRARY**-ATTR**IBUTES

LIBRARY = **<u>*STD</u>** / <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

Operanden

LIBRARY = <u>*STD</u> / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, deren Attribute ausgegeben werden sollen.

LIBRARY = <u>*STD</u>
Es wird die globale, durch OPEN-LIBRARY eröffnete Bibliothek angezeigt.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, deren Zustand angezeigt werden soll.

LIBRARY = *LINK(...)
Der Zustand der über einen Dateikettungsnamen zugewiesenen Bibliothek wird angezeigt.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der vor dem Aufruf von LMSCONV mit /ADD-FILE-LINK vereinbart wurde.

Beispiel

```
//show-library-attributes
INPUT  LIBRARY= :X:$USER.LIB
INIT-ELEM-P= *NONE
ADMINISTRAT= *NONE
FILE-SIZE  = 291       FREE-SIZE  = 62       FORMAT = NK2   UPAM-PROT = N
ACCESS-DATE= *NONE     WR-CONTROL = *NONE    STORAGE= *NONE
```

7.4.5.8 SHOW-LIBRARY-STATUS - Status der Bibliothek anzeigen

Diese Anweisung zeigt den Zustand der benutzten Bibliotheken an. LMSCONV gibt nach der Ausführung der Anweisung folgende Informationen aus:

- Name der Bibliothek(en)
- Zustand der Bibliothek(en) (geöffnet oder geschlossen)
- evtl. zugeordneter Dateikettungsname

Format

SHOW-LIBRARY**-STA**TUS

LIBRARY = **<u>*ALL</u>** / ***STD** / <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

Operanden

LIBRARY = <u>*ALL</u> / *STD / <filename 1..54 without-vers> / *LINK(...)
Angabe der benutzten Bibliotheken, deren Zustand ausgegeben werden soll.

LIBRARY = <u>*ALL</u>
Es werden alle benutzten Bibliotheken angezeigt.

LIBRARY = *STD
Es wird die globale, durch OPEN-LIBRARY eröffnete Bibliothek angezeigt.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, deren Zustand angezeigt werden soll.

LIBRARY = *LINK(...)
Der Zustand der über einen Dateikettungsnamen zugewiesenen Bibliothek wird angezeigt.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der vor dem Aufruf von LMSCONV mit /ADD-FILE-LINK vereinbart wurde.

Beispiel

Während des LMSCONV-Laufs wurde mit fünf verschiedenen Bibliotheken gearbeitet. Eine Bibliothek wurde über den Dateikettungsnamen LIB1 angesprochen:

```
//show-library-status
STATUS FILENAME                                          MODE    LINK
OPEN   :N:$USER.LMSPL.LIB                                UPDATE
CLOSED :N:$USER.MODUL.LIB                                        LIB1
CLOSED :N:$USER.MACRO.LIB
CLOSED :N:$USER.QUELL.LIB
CLOSED :N:$USER.TEST.LIB
```

7.4.5.9 SHOW-LOGGING-PARAMETERS - Globale LMSCONV-Parameter anzeigen

Diese Anweisung gibt die globalen LMSCONV-Einstellungen aus, die momentan gültig sind. Die Einstellungen werden mit der Anweisung MODIFY-LOGGING-PARAMETERS geändert.

Wird diese Anweisung ohne Operanden angegeben, werden alle Voreinstellungen ausgegeben (siehe Beispiel).

Format

SHOW-LOGGING-PARAMETERS
PARAMETERS = **<u>*ALL</u>** / list-poss(2000): ***LOG**GING / ***OUTPUT** / ***TEXT-OUT**PUT / ***OUT**PUT**-LAYOUT**

Operanden

PARAMETERS = <u>*ALL</u> / *LOGGING / *OUTPUT / *TEXT-OUTPUT / *OUTPUT-LAYOUT
Ausgabe der aktuellen Voreinstellung

PARAMETERS = <u>*ALL</u>
Es werden alle Voreinstellungen mit ihrer aktuellen Belegung ausgegeben.

PARAMETERS = *LOGGING / *OUTPUT / *TEXT-OUTPUT
Es wird das voreingestellte Ausgabemedium ausgegeben.

PARAMETERS = *OUTPUT-LAYOUT
Es werden die voreingestellten Parameter für das LMSCONV-Protokollformat ausgegeben.

Beispiel

```
//show-logging-parameters
LOGGING                = *MINIMUM
TEXT-OUTPUT            = *SYSOUT
OUTPUT-LAYOUT
  LINES-PER-PAGE       = 60
  LINE-SIZE            = 132
  EXTRA-FORM-FEED      = *NO
  HEADER-LINES         = *YES
```

Es werden alle globalen LMSCONV-Einstellungen ausgegeben. Diese Einstellungen sind direkt nach Start von LMSCONV gültig.

7.4.5.10 SHOW-TYPE-ATTRIBUTES - Attribute eines Elementtyps anzeigen

Diese Anweisung gibt alle für einen Elementtyp eingestellten Attribute aus.

Format

SHOW-TYPE-ATTRIBUTES

LIBRARY = **<u>*STD</u>** / <filename 1..54 without-vers> / ***LINK**(...)

***LINK**(...)

| **LINK**-NAME = <structured-name 1..8>

,**TYPE** = ***DEF**AULT / ***ALL** / <alphanum-name 1..8 with-wild(20)>

Operanden

LIBRARY = <u>*STD</u> / <filename 1..54 without-vers> / *LINK(...)
Angabe der Bibliothek, in der sich der Elementtyp befindet.

LIBRARY = <u>*STD</u>
Es wird die global eröffnete Bibliothek verwendet.

LIBRARY = <filename 1..54 without-vers>
Name der Bibliothek, die verwendet werden soll.

LIBRARY = *LINK(...)
Die Typattribute der über einen Dateikettungsnamen zugewiesenen Bibliothek werden angezeigt.

LINK-NAME = <structured-name 1..8>
Dateikettungsname der Bibliothek, der vor dem Aufruf von LMSCONV mit /ADD-FILE-LINK vereinbart wurde.

TYPE = *DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
Elementtyp, dessen Attribute ausgegeben werden sollen. Ist der LMSCONV-Default *NONE, so wirkt *DEFAULT wie *ALL.

> ¡ Für TYPE=* werden die Attribute für alle explizit vereinbarten Typen und für Typen, unter denen Elemente existieren, ausgeben.

Beispiel

Es werden die Typattribute für den Elementtyp S ausgegeben. Die verwendete Bibliothek ist die global eingestellte Biblothek.

```
//show-type-attributes type=s
INPUT  LIBRARY= :X:$USER.TESTLIB
TYPE       = S
SUPER-TYPE = *NONE
BASE-TYPE  = S
CONVENTION = *NONE
INIT-ELEM-P= *NONE
```

```
ADMINISTRAT= *NONE
STORAGE    = *NONE        WR-CONTROL = *NONE
```

7.4.5.11 SHOW-USER-EXITS - LMSCONV-Version anzeigen

Diese Anweisung zeigt die LMSCONV-Version an.

Format

SHOW-USER-EXITS

Operanden

Diese Anweisung besitzt keine Operanden.

7.4.5.12 WRITE-COMMENT - Kommentare in Ausgabemedium schreiben

Mit dieser Anweisung werden Kommentare in das durch die Anweisung MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=... festgelegte Ausgabemedium geschrieben, im Gegensatz zur SDF-Standardanweisung WRITE-TEXT, die immer nach SYSOUT oder SYSLST ausgibt.

Format

WRITE- **COMMENT**
COMMENT = ''BLANK'' / <c-string 1..1024 with-low>

Operanden

COMMENT = ''BLANK''** / <c-string 1..1024 with-low>**
Kommentartext. Wird nichts angegeben, wird als Default-Wert ein Leerzeichen ''BLANK'' genommen, d.h. eine Leerzeile erzeugt.

7.5 Beispiel: Ändern eines Bindelademoduls

```
/start-lmsconv                                                          1.
%  LMC0310 LMSCONV VERSION '<version>' STARTED
//modify-logging-parameters logging=*maximum                            2.
//open-library library=tst.bib,mode=*update                             3.
//modify-element (element=tst.ele,type=l)                               4.
//modify-modification-defaults csect=test                               5.
//add-text-modification address=x'e0',
       new-contents='ZZ'(old-contents='aa')                             6.
//end-modify                                                            7.
INPUT  LIBRARY= :2OSX:$USER.TST.BIB
OUTPUT LIBRARY= :2OSX:$USER.TST.BIB
INPUT  ELEMENT= (L)TST.ELE/031(0001)/<date>
OUTPUT ELEMENT= (L)TST.ELE/031(0002)/<date>
TEXT-ADR:            000000E0
TEXT BEFORE CHANGE:  a a
                     8181
TEXT AFTER  CHANGE:  Z Z
                     E9E9
         CORRECT (L)TST.ELE/031(0001)/<date> AS
                 (L)TST.ELE/031(0002)/<date> , OUTPUT REPLACED
//END                                                                   8.
```

1. LMSCONV wird aufgerufen.
2. Es wird das komplette LMSCONV-Protokoll ausgegeben.
3. Die Bibliothek TST.BIB wird zugewiesen. Sie muss zum Lesen und Schreiben geöffnet werden, andernfalls ist keine Änderung des Bindelademoduls möglich.
4. Der Bindelademodul TST.ELE soll geändert werden.
5. Diese Subanweisung legt die zu korrigierende CSECT fest.
6. Diese Subanweisung ersetzt an der Adresse E0 der CSECT TEST den Originaltext 'aa' durch den Ersetzungstext 'ZZ'.
7. Die Eingabe der Subanweisungen wird beendet. Es erfolgt die Ausführung der Subanweisungen.
8. LMSCONV wird beendet.

7.6 LMSCONV im Vergleich zu LMS

Globale Einschränkungen von LMSCONV zu LMS:

- keine Erzeugung deltagespeicherter Elemente
- keine strukturierte Ausgabe, d.h. keine Ausgabe in S-Variable
- keine Änderung von Bibliotheks- und Typattributen
- keine Unterstützung einer Startdatei

Folgende LMS-Anweisungen werden nicht unterstützt:

```
ACTIVATE-USER-EXIT
BEGIN-MAKE
CALL-EDT
COMPARE-ELEMENT
DEACTIVATE-USER-EXIT
EDIT-ELEMENT
EXECUTE-SYSTEM-COMMAND
FIND-ELEMENT
MODIFY-ELEMENT-PROTECTION
MODIFY-LIBRARY-ATTRIBUTES
MODIFY-TYPE-ATTRIBUTES
PROVIDE-ELEMENT
REORGANIZE-LIBRARY
RESET-Anweisungen
RETURN-ELEMENT
SHOW-STATISTICS
```

Die folgende Tabelle zeigt die Syntax-Einschränkungen im Vergleich zu LMS

Anweisung	Einschränkung
ADD-ELEMENT	kein *HIGH,*INCR,BASE,STORAGE-FORM bei TO-ELEMENT kein PROTECTION
CLOSE-LIBRARY	./.
COPY-ELEMENT	keine Auswahl über BASE bei ELEMENT kein EXCEPT kein *HIGH,*INCR,BASE,STORAGE-FORM bei TO-ELEMENT kein PROTECTION
COPY-LIBRARY	./.
DELETE-ELEMENT	keine Auswahl über BASE bei ELEMENT kein EXCEPT
END	./.
EXTRACT-ELEMENT	keine Auswahl über BASE bei ELEMENT kein EXCEPT kein INFORMATION kein PROTECTION

MODIFY-DEFAULTS	kein STORAGE-FORM bei ELEMENT-ATTRIBUTES kein DELTA-STRUCTURE bei INFORMATION kein PROTECTION kein COMPARE-PARAMETERS

Anweisung	Einschränkung
MODIFY-ELEMENT	keine Auswahl über BASE bei ELEMENT kein EXCEPT kein *HIGH,*INCR,BASE bei TO-ELEMENT
MODIFY-ELEMENT-ATTRIBUTES	keine Auswahl über BASE bei ELEMENT kein EXCEPT keine Auswahl über STATE kein WRITE-MODE für NEW-ATTRIBUTES gibt es nur CODED-CHARACTER-SET
MODIFY-LOGGING-PARAMETERS	kein EDT(...) bei TEXT-OUTPUT kein BASE bei ELEMENT
OPEN-LIBRARY	./.
SHOW-DEFAULTS	kein PROTECTION kein COMPARE-PARAMETERS
SHOW-ELEMENT	keine Auswahl über BASE bei ELEMENT kein EXCEPT
SHOW-ELEMENT-ATTRIBUTES	keine Auswahl über BASE bei ELEMENT kein EXCEPT keine Auswahl über ACCESS-DATE/TIME keine Auswahl über STATE keine Auswahl über STORAGE-FORM keine Auswahl über ELEMENT-SIZE keine Auswahl über PROTECTION keine Auswahl über DELTA-STRUCTURE bei INFORMATION kein *BY-CREATION-DATE, *BY-MODIFICATION-DATE, *BY-ACCESS-DATE, *BY-ELEMENT-SIZE bei SORT kein TEXT-OUTPUT kein STRUCTURE-OUTPUT
SHOW-LIBRARY-ATTRIBUTES	kein TEXT-OUTPUT kein STRUCTURE-OUTPUT
SHOW-LIBRARY-STATUS	./.
SHOW-LOGGING-PARAMETERS	./.

Anweisung	Einschränkung
SHOW-TYPE-ATTRIBUTES	kein TEXT-OUTPUT kein STRUCTURE-OUTPUT
SHOW-USER-EXITS	./.
WRITE-COMMENT	./.

Umstellung von LMSCONV auf LMS

Alle LMSCONV-Anweisungen und Operanden sind in LMS gültig, sofern die vollständigen Anweisungsnamen oder deren garantierten Abkürzungen verwendet werden. Somit ist ein Umsteigen von LMSCONV auf LMS für den Benutzer unproblematisch.

8 MSGMAKER Bearbeiten von BS2000-Meldungsdateien

Versionsstand: MSGMAKER V1.2B

MSGMAKER bietet folgende Funktionen:

- Einrichten einer Meldungsdatei
- Aufbau einer Meldungsdatei aus Meldungseinheiten
    - Eintragen neuer Meldungseinheiten (ADD-MSG-Funktion)
    - Ändern bestehender Meldungseinheiten (MODIFY-MSG-Funktion)
    - Löschen bestehender Meldungseinheiten (DELETE-MSG-Funktion)
    - Anzeigen des Inhalts einer Meldungsdatei (SHOW-Funktion)
    - Kopieren verschiedener Bestandteile einer Meldungsdatei (Meldungseinheiten, Dokumentationszeilen, Komponentenidentifikation) von einer Meldungsdatei in eine andere oder innerhalb derselben Datei. Der Sendebereich der Meldungsdatei bleibt entweder erhalten (COPY-Funktion) oder wird gelöscht (MOVE-Funktion)
    - Dokumentieren von Meldungen eines bezeichneten Meldungsintervalls(ADD-DOCUMENTATION-Funktion)
    - Ändern bestehender Dokumentationszeilen (MODIFY-DOCUMENTATION-Funktion)
    - Löschen bestehender Dokumentationszeilen (DELETE-DOCUMENTATION-Funktion)
    - Meldungs-, Bedeutungs- und Maßnahmetext können in maximal acht Sprachen definiert werden
    - Variable Textteile (Inserts) können im Meldungstext definiert werden. Jedes Insert erhält eine Nummer und einen Namen; ferner kann ein Default-Text zugewiesen werden
- Mischen der Inhalte mehrerer Meldungsdateien in eine Meldungsdatei

> **i** Hinweise zum „alten“ Format der Meldungsdatei (MSGEDIT/MSGLIB) wurden aus dieser MSGMAKER-Beschreibung entfernt. Das „alte“ Format kann aber aus Kompatibilitätsgründen noch verwendet werden. Die Programme MSGEDIT und MSGLIB sind vom Vertrieb gestrichen.

Die Steuerung von MSGMAKER erfolgt

- mit Anweisungen (Batch-Aufträge und Prozeduren)
- im maskengeführten Dialog (Menümodus)
  Es besteht die Möglichkeit, in den Kommandobereich der Maske ebenfalls Anweisungen einzugeben (mit Ausnahme der Anweisung //OPEN-MSG-FILE).

Folgende Übersicht zeigt die Funktionen von MSGMAKER und die entsprechenden Anweisungen und Bildschirmmasken, die die Funktion realisieren.

Funktion	Maske	Anweisung
Ändern bestehender Meldungseinheiten	MODIFY-MSG	//MODIFY-MSG
Ändern bestehender Dokumentationszeilen	MODIFY-DOCUMENTATION	//MODIFY-DOCUMENTATION
Anzeigen des Inhalts einer Meldungsdatei	SHOW	//SHOW
Einfügen oder Ändern von Bedeutungstext	MEANING /RESPONSE	Operand MEANING der Anweisungen //ADD-MSG und //MODIFY-MSG
Einfügen oder Ändern von Maßnahmetext	MEANING /RESPONSE	Operand RESPONSE der Anweisungen //ADD-MSG und //MODIFY-MSG
Einfügen oder Ändern des Meldungstextes	MSG-TEXT	Operand MSG-TEXT der Anweisungen //ADD-MSG und //MODIFY-MSG
Einrichten einer Meldungsdatei	MSG-FILE-ATTRIBUTES	//OPEN-MSG-FILE
Eintragen neuer Meldungseinheiten	ADD-MSG	//ADD-MSG
Eintragen von Dokumentationszeilen	ADD-DOCUMENTATION	//ADD-DOCUMENTATION
Inserts definieren	INSERT-ATTRIBUTES	Operand INSERT-ATTRIBUTES der Anweisungen //ADD-MSG und //MODIFY-MSG
Kopieren verschiedener Bestandteile einer Meldungsdatei	COPY	//COPY
Kopieren und Löschen verschiedener Bestandteile einer Meldungsdatei	MOVE	//MOVE
Löschen bestehender Meldungseinheiten	DELETE-MSG	//DELETE-MSG
Löschen von Dokumentationszeilen	DELETE-DOCUMENTATION	//DELETE-DOCUMENTATION
Mischen von Meldungsdateien	- - -	//MERGE-MSG-FILES

8.1 Ablauf von MSGMAKER

Das Dienstprogramm MSGMAKER wird in den Sprachen Englisch und Deutsch ausgeliefert und mit dem Installationsmonitor IMON auf einer beliebigen Benutzerkennung installiert.

Die Masken und Meldungen von MSGMAKER werden in der Sprache ausgegeben, in welcher die Systemkomponente MIP gerade arbeitet.

Mit dem Kommando `/MODIFY-MSG-ATTRIBUTES TASK-LANGUAGE=<lang>` können die Sprachen für die Ausgabe festgelegt werden: Englisch (TASK-LANGUAGE = E) oder Deutsch (TASK-LANGUAGE = D).

Dieses Kommando kann nicht in den Kommandobereich der Masken sondern nur auf Kommandoebene (/MOD...) eingegeben werden. Wurde vor Programmstart keine Tasksprache vereinbart, übernimmt MSGMAKER den vom System voreingestellten Wert.

8.1.1 Programm starten

Das Programm MSGMAKER wird gestartet mit `/START-MSGMAKER`.

START-MSGMAKER	Alias: **MSGMAKER**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

MSGMAKER kann auch so aufgerufen werden:

```
/START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(
                          LIBRARY= $<userid>.SYSLNK.MSGMAKER.<version>,
                            ELEMENT=MSGMAKER)
```

`<version>` ist die aktuelle Version von MSGMAKER, z.B. `012`.
Bei Programmaufruf stehen dem Anwender ferner die Operanden `CPU-LIMIT`, `TEST-OPTIONS`, `MONJV`, `RESIDENT-PAGES` und `VIRTUAL-PAGES` des Kommandos `/START-EXECUTABLE-PROGRAM` zur Verfügung, um z. B. den Programmlauf überwachen zu können. Zur Beschreibung dieser Operanden, siehe Kommando `/START-EXECUTABLE-PROGRAM` im Handbuch „Kommandos“ [1].

Das Programm läuft unter jeder Benutzerkennung.

MSGMAKER hat keine Unterprogrammschnittstelle, d.h. es kann nicht von einem anderen Programm aus aufgerufen werden.

8.1.2 Definition einer programmüberwachenden Jobvariablen

Eine programmüberwachende Jobvariable (MONJV) kann beim Starten des Programms MSGMAKER definiert werden.

Während des Programmlaufs setzt das Betriebssystem den Zustandanzeiger (ersten drei Bytes der Jobvariable) auf den Wert

- $R: Das Programm läuft.

Nach Beendigung des Programms kann der Zustandsanzeiger folgende Werte annehmen:

- $T: Das Programm wurde erfolgreich beendet.
- $A: Das Programm wurde vorzeitig durch einen Programmfehler oder durch einen definierten Fehlerausgang beendet.

Bei Programmende wird dem vierten Byte der Jobvariablen einer der folgenden Werte zugewiesen:

- 0: Das Programm lief ohne Fehler und Warnungen.
- 1: Das Programm lief ohne Fehler, jedoch mit Warnungen.
- 3: Das Programm lief mit Fehler.

Die Bytes fünf bis sieben der Jobvariablen werden von MSGMAKER nicht belegt.

Nähere Informationen zu überwachenden Jobvariablen siehe Handbuch „JV“ [16].

8.1.3 Bedienungsmöglichkeiten von MSGMAKER

Das Programm MSGMAKER ist in zwei Modi zu bedienen:

1. Dialogmodus, im folgenden Menümodus genannt (siehe "Menümodus").
   Die Bedienung erfolgt maskengesteuert oder mit Anweisungen, die in den Kommandobereich der Masken eingegeben werden.
2. Batch- und Prozedurmodus (siehe "Anweisungen"). Hierzu zählen der Batch-Betrieb, der Batch-Prozedur-Betrieb und der Dialog-Prozedur-Betrieb.
   Die Bedienung erfolgt über Anweisungen.

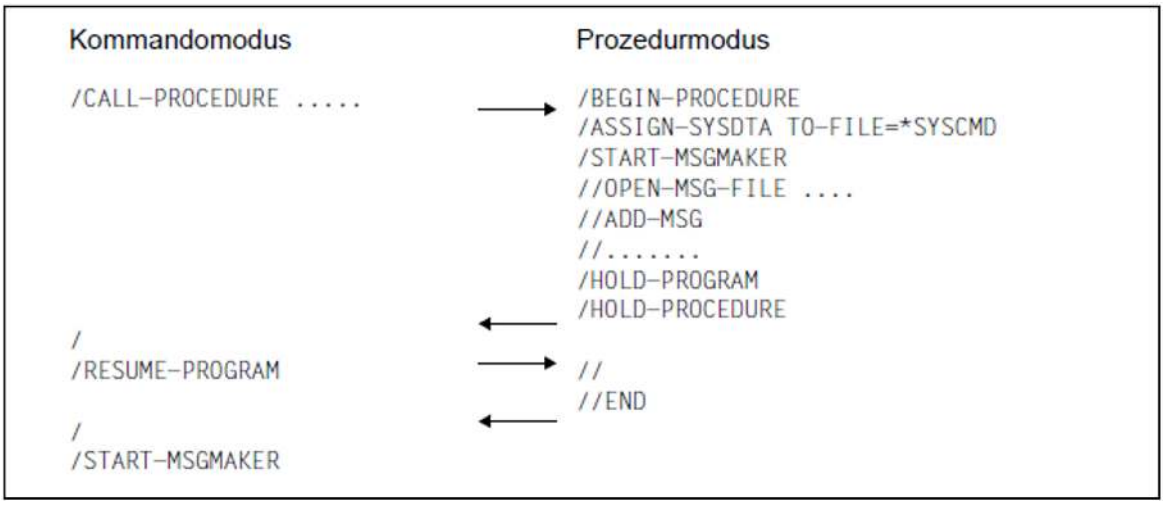
Der Bedienungsmodus von MSGMAKER wird beim Programmstart anhand der aktuellen Aufrufumgebung bestimmt. Während des Programmlaufs können Sie den Bedienungsmodus nicht verändern. Sie müssen das Kommando /START-MSGMAKER erneut eingeben, um das Programm in dem anderen Modus bedienen zu können. Das folgende Beispiel verdeutlicht dies.

Beispiel

Es liegt folgende Situation vor:
MSGMAKER wird in einer Prozedur aufgerufen. Die MSGMAKER-Anweisungen stehen ebenfalls in dieser Prozedur. MSGMAKER wird also im Prozedurmodus gestartet. Durch /HOLD-PROGRAM bzw. /HOLD-PROCEDURE wird das Programm bzw. die Prozedur unterbrochen. Sie befinden sich nun auf Kommandoebene.

Kehren Sie mit /RESUME-PROGRAM zum Programm zurück, wartet MSGMAKER mit // auf die Eingabe einer Anweisung. Sie befinden Sie sich wieder im geführten Dialog. Auch wenn der Prozedurmodus aktuell nicht mehr aktiv ist, so ändert das nichts daran, dass der ursprüngliche Programmstart in einer Prozedur erfolgt ist.

In diesem Fall ist es am Besten, wenn Sie mit K2 abbrechen oder mit der Anweisung END das Programm MSGMAKER beenden. Durch einen erneuten Programmstart, diesmal auf Kommandoebene, können Sie MSGMAKER im Menümodus bedienen.



8.1.4 Besondere Zeichensätze

Im Menümodus können Meldungen mit besonderen Zeichen aus erweiterten Zeichensätzen (z.B. 8-Bit-Zeichensätze) erstellt werden. Hierzu muss die entsprechende Bibliothek für die Bildschirmdarstellung zugewiesen werden, bevor MSGMAKER gestartet wird.

`/ADD-FILE-LINK LINK-NAME=MAPLIB,FILE-NAME=SYSFHS.MSGMAKER.<version>.<code>`

`<version>` beinhaltet die aktuelle Version von MSGMAKER, `<code>` den Namen des gewünschten Zeichensatzes.

Das Terminal muss ebenfalls im 8-Bit-Modus arbeiten. Mit dem Kommando `/MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=8-BIT-DEFAULT` wird die 8-Bit-Codetabelle des Benutzereintrags aktiviert.

8.1.5 Meldungen von MSGMAKER

Die Meldungen des Dienstprogramms MSGMAKER haben die Meldungsklasse MSM.

Siehe auch den Abschnitt „Meldungen und ihre Bedeutung“ (Darstellungsmittel).

8.2 Menümodus

Im Menümodus bietet MSGMAKER für jede Funktion ein oder zwei Masken an, in welchen der Anwender

- durch Eingabe von Buchstaben, Ziffern oder durch Markierung mit „X“ eine Funktion auswählen und durch Drücken der Taste DUE starten kann
- durch Drücken der Funktionstasten F2, F3, K1, K2 und K3 Bedienfunktionen auslösen kann
- durch Eingabe von Anweisungen in den Kommandobereich, die festgelegte Reihenfolge und Bearbeitung der Masken verlassen und jede beliebige Funktion aufrufen kann. Die gezielte Ansteuerung der einzelnen Masken ermöglicht einen schnelleren Durchlauf durch die Masken und verkürzt somit die Bearbeitung einer Meldungsdatei (siehe auch "GO-TO - Verzweigen zu einer angegebenen Maske")
- mit Hilfe von SDF jede Anweisung aufrufen kann. SDF befindet sich im EXPERT-Modus; durch Eingabe eines Fragezeichens bietet SDF alle verfügbaren Anweisungen von MSGMAKER an.

8.2.1 Übersicht über die Masken

Funktion	Kurzbeschreibung
ADD-DOCUMENTATION	Dokumentation (Verantwortliche, Kommentar) zu Meldungen eines bestimmten Intervalls eintragen
ADD-MSG	Meldungseinheiten einfügen
COPY	Meldungseinheiten und Dokumentationszeilen kopieren
DELETE-DOCUMENTATION	Dokumentation (Verantwortliche, Kommentar) zu Meldungen eines bestimmten Intervalls löschen
DELETE-MSG	Meldungseinheiten löschen
INSERT-ATTRIBUTES	Insert-Attribute einfügen, ändern oder löschen
MEANING/RESPONSE	Bedeutungs- und Maßnahmetext einfügen, ändern oder löschen
MODIFY-DOCUMENTATION	Dokumentation (Verantwortliche, Kommentar) zu Meldungen eines bestimmten Intervalls ändern
MODIFY-MSG	Meldungseinheiten ändern
MOVE	Meldungseinheiten und Dokumentationszeilen kopieren und löschen
MSG-FILE-ATTRIBUTES	Attribute der Meldungsdatei eintragen oder ändern
MSG-TEXT	Meldungstext (und Inserts) einfügen, ändern oder löschen
SHOW	Inhalt einer Meldungsdatei (Meldungseinheit und Dokumentationszeilen) anzeigen

8.2.2 Reihenfolge der Masken

In Bild 11 ist die Reihenfolge der Masken dargestellt, in welcher sie nach Eingabe von bestimmten Operationsnummern und Funktionstasten erscheinen. Die Reihenfolge der Masken ist hierarchisch gegliedert:

- die Hauptmaske MENU erscheint nach dem Aufruf von MSGMAKER
- die Masken MSG-FILE-ATTRIBUTES, ADD-MSG, MODIFY-MSG, DELETE-MSG, COPY, MOVE, SHOW, ADD-DOCUMENTATION, MODIFY-DOCUMENTATION und DELETE-DOCUMENTATION werden von der Hauptmaske MENU aus aufgerufen.
  In der Maske SHOW werden die anzuzeigenden Meldungseinheiten ausgewählt; für die Ausgabe wird eine weitere Maske mit dem Namen SHOW-OUTPUT aufgerufen. Analog hierzu wird in der Maske DELETE-MSG die zu löschende Meldungseinheit ausgewählt und in einer zweiten, gleichnamigen Maske angezeigt, bevor die DELETE-Funktion gestartet wird.
- die Masken MSG-TEXT, MEANING-RESPONSE und INSERT-ATTRIBUTES werden von den Masken ADD-MSG und MODIFY-MSG aus automatisch angeboten

Die Ziffern in der Übersicht bedeuten die in der Maske MENU ausgewählten Funktionen.

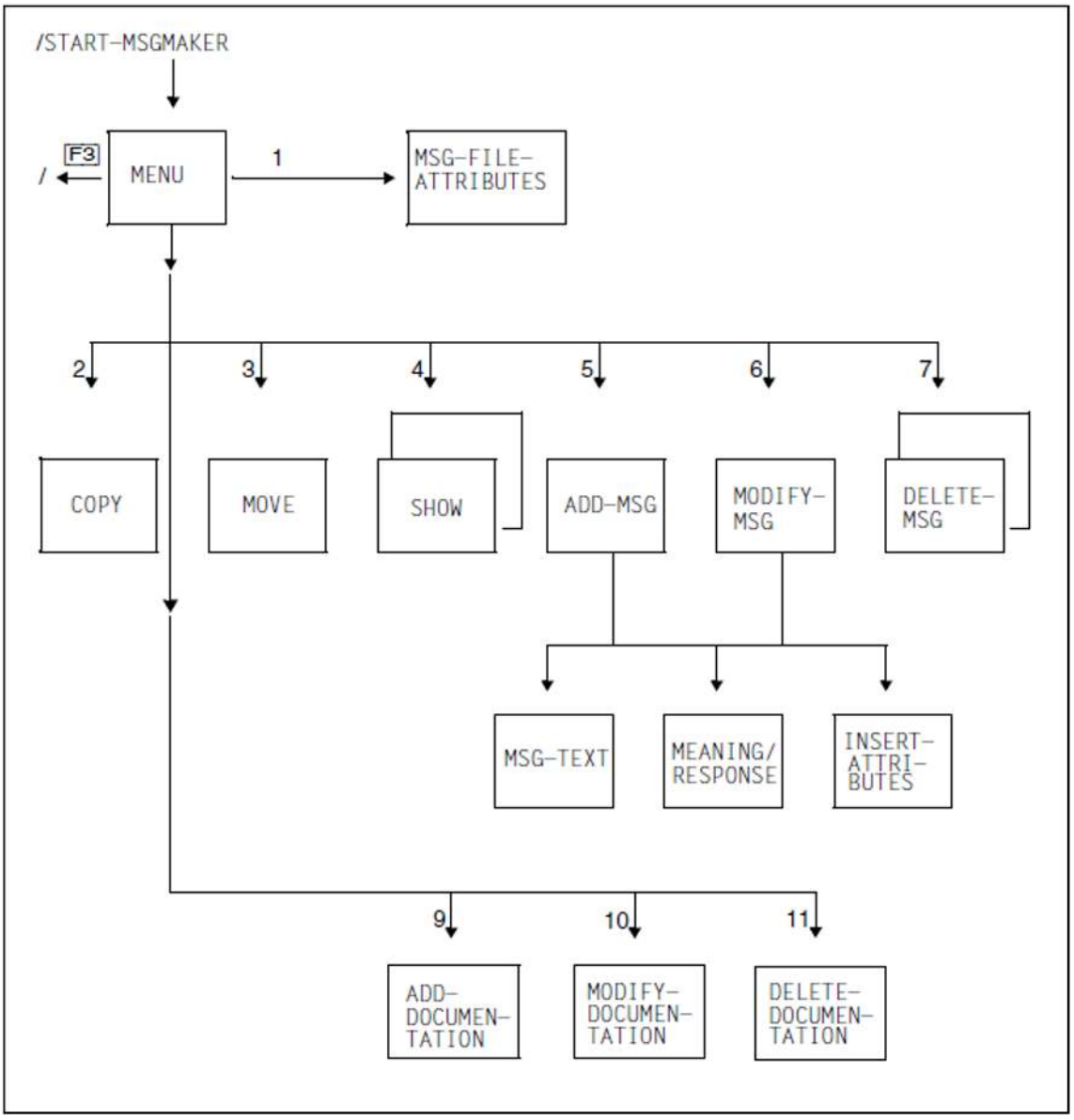
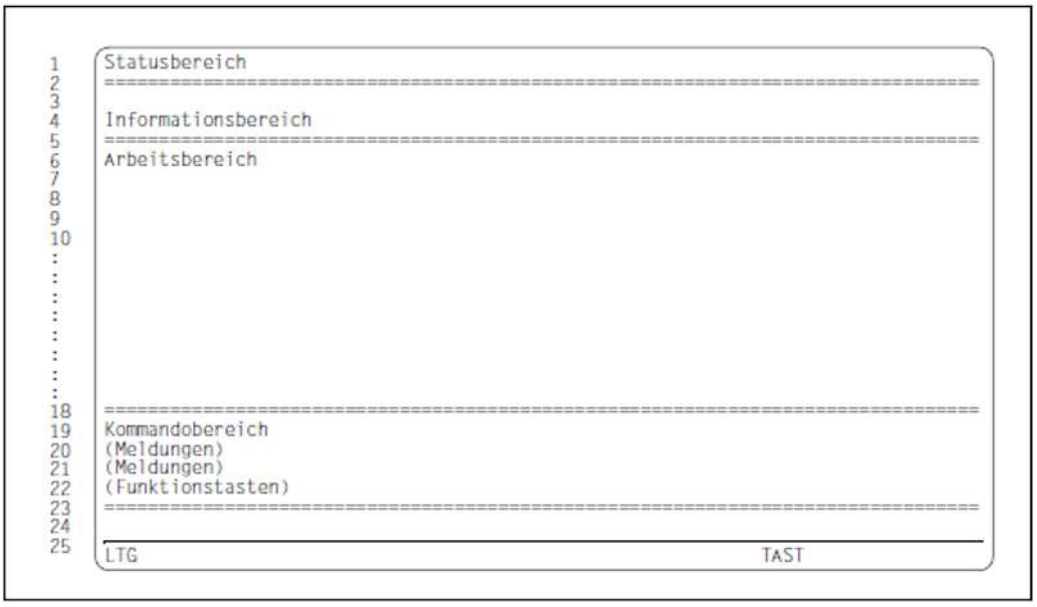


Bild 11: Maskenübersicht von MSGMAKER

8.2.3 Allgemeiner Maskenaufbau



Statusbereich

Der Statusbereich enthält den Maskentitel, der die Funktion der Maske beschreibt. Die zu dieser Maske korrespondierende Anweisung trägt ebenfalls diesen Namen.

Ausnahmen:
Anweisungen mit den Namen MENU und MSG-FILE-ATTRIBUTES existieren nicht. Die Funktionen beider Masken sind in der Anweisung //OPEN-MSG-FILE zusammengefaßt.

Informationsbereich

Der Informationsbereich gibt, wenn vorhanden, Auskunft über die aktuelle Meldungsdatei oder über die gerade bearbeitete Meldungseinheit (Meldungsschlüssel, Sprache, ...).

Arbeitsbereich

Der Arbeitsbereich ist der eigentlichen Aktionsbereich des Anwenders. In ihm können Funktionen ausgewählt oder Werte eingegeben werden.

Kommandobereich / Ausgabebereich der Meldungen

Im dreizeilige Kommandobereich besteht die Möglichkeit eine Anweisung oder die Zeichen „+“ oder „-“ einzugeben. Ferner wird in der letzten Zeile der Maske die Belegung der Funktionstasten angezeigt.

Die zweite und dritte Zeile des Kommandobereiches sind zugleich der Ausgabebereich für Meldungen des Programms oder des Systems.

Weiter (+ -)

- **+** Zu den Masken MSG-TEXT, INSERT-ATTRIBUTES und SHOW-OUTPUT kann jeweils eine zweite Maske mit „+“ aufgerufen werden, um alle restlichen definierten Meldungstexte oder Inserts sichtbar zu machen.
- **-** Mit „-“ wird zur vorausgegangenen Maske zurückgeblättert.

Kommando ==>

Es können Anweisungen eingegeben werden, wie sie in den Abschnitten „Anweisungen“ und „Besonderheiten der Anweisungen im Menümodus“ beschrieben sind.

> Eine in den Kommandobereich eingetragene Anweisung wird stets vor der angezeigten Masken-Funktion ausgeführt. Ausnahme: Maske MENU.

Wird eine vollständige Anweisung eingegeben, wird die Funktion im Hintergrund sofort ausgeführt.

Bei einer unvollständigen Anweisung, d.h. ohne oder nur mit einem Teil der Operanden, wird die der Anweisung entsprechende Maske angezeigt. Operanden, denen bereits im Anweisungsaufruf Werte zugewiesen wurden, werden in die Felder der Maske übernommen. Noch fehlende Operandenwerte können in der Maske ergänzt werden. Durch Drücken der Taste DUE wird die Anweisung ausgeführt und zur aufrufenden Maske zurückgekehrt. Der Kommandobereich der Maske ist nun leer, die weiteren Maskenbereiche befinden sich in dem Zustand, der vor Eingabe der Anweisung vorlag.

Wird auf eine Anweisung die Prompt-Funktion angewendet, wird die Anweisung nicht sofort ausgeführt, sondern die der Anweisung entsprechende Maske angezeigt. Es besteht die Möglichkeit, fehlende Operandenwerte zu ergänzen. Durch DUE wird die Ausführung der Anweisung veranlaßt. Näheres hierzu im Abschnitt „Funktionstasten“.

Mit der Anweisung //GO-TO kann gezielt eine beliebige Maske aufgerufen werden (siehe "GO-TO - Verzweigen zu einer angegebenen Maske").

Eine verkettete Eingabe von „+“ / „-“ und Anweisungen ist nicht möglich.

Im Kommandobereich erscheint nur „Kommando ==>“, wenn der Aufruf einer ergänzenden Maske nicht möglich ist.

Wird in der Kommandozeile ein Fragezeichen „?“ eingegeben und anschließend die Taste DUE gedrückt, erscheint ein Auswahlmenü mit MSGMAKER-Anweisungen und SDF-Standardanweisungen. Durch Eingabe der Nummer, die vor der Anweisung steht, in das Eingabefeld NEXT, erhält man den zur Anweisung gehörenden Operandenfragebogen. Durch Eingabe von *CANCEL, *EXIT oder *EXIT-ALL wird zur zuletzt aufgerufenenen Maskenfunktion zurückgekehrt.

Funktionstasten

Die Funktionstasten F2, F3, K1, K2 und K3 ermöglichen eine einfache und schnelle Bedienung.

F2 = Prompt

Liefert ausführliche Informationen zu einer Anweisung, die in den Kommandobereich eingetragen wurde. Durch Drücken der Taste F2 im Anschluss an eine Eingabe im Kommandobereich wird die Prompt-Funktion ausgelöst. Wird auf eine Anweisung die Prompt-Funktion angewendet, wird die der Anweisung entsprechende Maske aufgerufen und die vorher angegebenen Operanden in die Maske übertragen. Bei vollständiger Operandenangabe wird durch Drücken der Taste DUE die Ausführung der Anweisung veranlaßt und in die aufrufende Maske zurückgekehrt. War die Operandenangabe unvollständig, verzweigt MSGMAKER nach Drücken von DUE zu den Masken, die für die Eingabe noch fehlender Operanden nötig sind. MSGMAKER kehrt dann zur aufrufenden Maske zurück.

Die Anweisung wird in der Maske nicht mehr angezeigt; alle weiteren Maskenbereiche befinden sich in dem Zustand, der vor Aufruf der Prompt-Funktion vorlag.

F3 = Exit

Durch Drücken der Taste F3 wird eine gerade bearbeitete Funktion verlassen. Die Funktion wird **nicht** ausgeführt, die Eingaben gehen verloren.

Bedeutung von F3

ADD-MSG MODIFY-MSG DELETE-MSG COPY MOVE SHOW MSG-TEXT INSERT-ATTRIBUTES	F3 = Verlassen der Maske
MENU	F3 = MSGMAKER beenden

Die Taste F3 wird in folgenden Masken **nicht** angeboten:

- MEANING/RESPONSE
- MSG-FILE-ATTRIBUTES
- ADD-DOCUMENTATION
- MODIFY-DOCUMENTATION
- DELETE-DOCUMENTATION

Im SDF-geführten Modus bewirkt F3 die sofortige Ausführung einer Anweisung.

K1 = Abbrechen/Überspringen

bewirkt die Rückkehr zur vorausgegangenen Maske. Alle Eingaben in die Maske, welche zwischen dem letzten DUE und dem Drücken der Taste K1 erfolgten, gehen verloren. (K1 = Abbrechen)
Die Taste K1 = Überspringen bewirkt bei der Bearbeitung mehrerer Meldungseinheiten (in den Masken ADD-/MODIFY-/DELETE-MSG) das Verlassen der gerade in Bearbeitung stehenden Meldungseinheit und führt zur nächsten Meldungseinheit. Analog zu K1 = Abbrechen gehen die Eingaben, die zwischen dem letzten DUE und dem Drücken der Taste K1 erfolgten, verloren.
Ist die verlassene Meldungseinheit die letzte innerhalb eines bearbeiteten Meldungssatzes, wird die Funktion (ADD-/ MODIFY-/DELETE-MSG) beendet.

K2 = Unterbrechen

unterbricht das Programm MSGMAKER und wechselt in den BS2000 Kommandomodus. Mit dem Kommando /RESUME-PROGRAM wird MSGMAKER fortgesetzt.

K3 = Refresh

speichert den Maskeninhalt, der durch letztmaliges Drücken der Taste DUE festgehalten wurde. Alle Eingaben in die Maske, die zwischen DUE und K3 erfolgten, gehen verloren.

> **i** Durch Eingabe eines Fragezeichens in die Kommandozeile einer Maske, wechselt MSGMAKER in den geführten Dialog. Während des geführten Dialogs gilt die für SDF eingestellte Funktionstastenbelegung.

Ausgabebereich der Meldungen

In diesem Bereich erscheinen Fehlermeldungen, Warnungen und Informationen, welche die letzte oder die beiden letzten Zeilen des Kommandobereichs überschreiben können. In diesem Fall geht das Ende einer längeren Anweisung verloren.

Die Meldungen von MSGMAKER beginnen mit der Meldungsklasse MSM. Wird eine Warnung ausgegeben, beginnt der Meldungstext mit „WARNING“; im Falle einer Fehlermeldung steht der Cursor auf dem ersten fehlerhaften Feld der Maske.

8.2.4 Eingaben in die Maske

Jedes Eingabefeld ist durch einen Namen, links neben dem Feld, gekennzeichnet. Der Name von Ein- und Ausgabefeldern wird in diesem Handbuch-Abschnitt in **fetter Schrift** dargestellt.

Folgende Eingaben in die Felder sind möglich:

Eingabe von	Beispiel	Eingabe in das Feld
Text	Maske: MENU	**Meldungsdatei**: testdatei
Buchstaben	Maske: MENU	**Eroeffnungsmodus**: U
Zahlen	Maske: MENU	**Operationsauswahl**: 5
Zeichen „X“	Maske: MOVE	**Informationen**: X Meldungen

Die Eingaben müssen weder links- noch rechtsbündig in die Felder eingetragen werden, freie Stellen sollten jedoch mit Leerzeichen oder NIL-Zeichen aufgefüllt werden.

Symbolische Namen <name> oder konkrete Angaben rechts neben einem Eingabefeld informieren über die möglichen Eingaben in das Feld.

Beispiel

(Maske COPY)

Feld: **Erster Schluessel**: ..... (<Schluessel> / <Teil-Schluessel>* / *=alle)

Mit den Zeichen „X“ können in einer Liste ein oder mehrere Elemente markiert und ausgewählt werden.

Durch Überschreiben der markierten Felder mit dem Leerzeichen oder NIL-Zeichen kann eine getroffene Auswahl wieder rückgängig gemacht werden.

Beispiel

(Maske COPY)

Feld: **Informationen**: X Meldungen

X Dokumentation

Komponente-Id

8.2.5 Beschreibung häufig auftretender Felder

Erster Schluessel und **Letzter Schluessel**

Erst nach Angabe des Meldungsschlüssels kann eine Meldungseinheit bearbeitet werden. Die Auswahl einer oder mehrerer Meldungseinheiten kann auf unterschiedliche Arten erzielt werden.

Explizite Angabe des Meldungsschlüssels

In das Feld **Erster Schluessel** wird ein kompletter, siebenstelliger Meldungsschlüssel, z.B. AAA0001 eingetragen. Wird das Feld **Letzter Schluessel** ebenfalls mit einem vollständigen Meldungsschlüssel besetzt, z.B. AAA0005, ist ein Intervall mit der unteren Grenze AAA0001 und der oberen Begrenzung AAA0005 definiert worden.
Die vierstellige Meldungsnummer kann sich aus Ziffern und Buchstaben zusammensetzen. Es ist zu beachten, dass Buchstaben in der Reihenfolge vor Ziffern kommen, z.B. CCC01AA, CCC01AB, ..., CCC01AZ, CCC01A0, ..., CCC01A9.

Implizite Angabe des Meldungsschlüssels

In das Feld mit dem Führungstext **Erster Schluessel** wird der Meldungsschlüssel eingetragen, der mit einem unveränderlichen Teil beginnt und von den Platzhaltern * oder # abgeschlossen wird. Das Zeichen * bedeutet, dass Buchstaben und Ziffern den Meldungsschlüssel beenden können, das Zeichen # (mögliche Angabe bei der Funktion ADD-MSG und MODIFY-MSG) steht ausschließlich für Ziffern.
Die Eingabe eines vollständigen Meldungsschlüssels in das Feld **Letzter Schluessel** gibt die obere Grenze eines Meldungsintervalls an. Jede teilqualifizierte Angabe eines Meldungsschlüssels in diesem Feld wird ignoriert.

Beispiele

CCC*	steht für	CCCAAAA,	CCCAAAB,	...,	CCCAAAZ
		CCCAAA0,	CCCAAA1,	...,	CCCAAA9
		CCCAABA,	CCCAABB,	...,	CCCAABZ
		...			CCC9999
CC*	steht für	CCA...,	CCB...,	...,	CCZ...
*	steht für	alle Meldungseinheiten werden ausgewählt			
CCC01A#	steht für	CCC01A0,	CCC01A1,	...,	CCC01A9
CCC#	steht für	CCC0000,	CCC0002,	...,	CCC0009
		CCC0010,	CCC0011,	...,	CCC0019
		CCC9990,	CCC9991,	...,	CCC9999
CC# oder C# oder # nicht möglich					

Spezielle Schlüsselwörter

(same)

Das Wort (same) wird im Feld **Letzter Schluessel** oder **In Schluessel** zur Information angezeigt. Wird (same) als Feldinhalt übernommen oder durch Leerzeichen oder NIL-Zeichen ersetzt, sind der erste und letzte Meldungsschlüssel identisch
(**Letzter Schluessel**: (same)) oder die Meldungsschlüssel werden nicht umbenannt (**In Schluessel**: (same)).

(list)

Das Wort (list) wird in den Feldern **Erster Schluessel** und **Letzter Schluessel** angezeigt, wenn in einer Anweisung mehrere Meldungsschlüssel angegeben werden, die nicht in einem Intervall zusammengefaßt werden können. (list) ist rein informativ, es kann nicht in ein Feld eingetragen werden.

Sprache(n)

Der Eintrag in das Feld **Sprache(n)** kennzeichnet die Sprache, in welcher die Texte (Meldungs-, Bedeutungs- und Maßnahmetext) abgefaßt werden. Jede Sprache wird durch einen Kennbuchstaben dargestellt. Das Kürzel D wird für Deutsch und E für Englisch verwendet. Alle weiteren Sprachen (insgesamt acht) können beliebig abgekürzt werden.
Kein Eintrag bedeutet, dass die gesamte Meldungseinheit, einschließlich der Meldungsattribute und der Texte in allen Sprachen, bearbeitet wird.

Meldungsdatei

Dieses Feld enthält den Namen einer Meldungsdatei.
In der Regel erscheint in diesem Feld der Name der aktuell geöffneten Meldungsdatei; in einigen Masken kann diese Vorbelegung überschrieben werden.

8.2.6 Beschreibung der Masken

- Maske MENU - Hauptmaske von MSGMAKER
- Maske MSG-FILE-ATTRIBUTES - Eintragen und Ändern der Meldungsdatei-Attribute
- Maske COPY - Kopieren von Meldungseinheiten
- Maske MOVE - Kopieren und Löschen von Meldungseinheiten
- Maske SHOW - Inhalt einer Meldungsdatei anzeigen
- Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen
- Maske ADD-MSG - Einfügen einer Meldungseinheit
- Maske MODIFY-MSG - Meldungseinheit ändern
- Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes
- Maske MEANING/RESPONSE - Einfügen oder Ändern des Bedeutungs- und Maßnahmetextes
- Maske INSERT-ATTRIBUTES - Einfügen oder Ändern der Insert-Attribute
- Maske DELETE-MSG - Meldungseinheit löschen
- Maske ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen
- Maske MODIFY-DOCUMENTATION - Ändern, Hinzufügen und Löschen von Dokumentationszeilen
- Maske DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen

8.2.6.1 Maske MENU - Hauptmaske von MSGMAKER

Reihenfolge der Masken

/START-MSG-MAKER
|
|

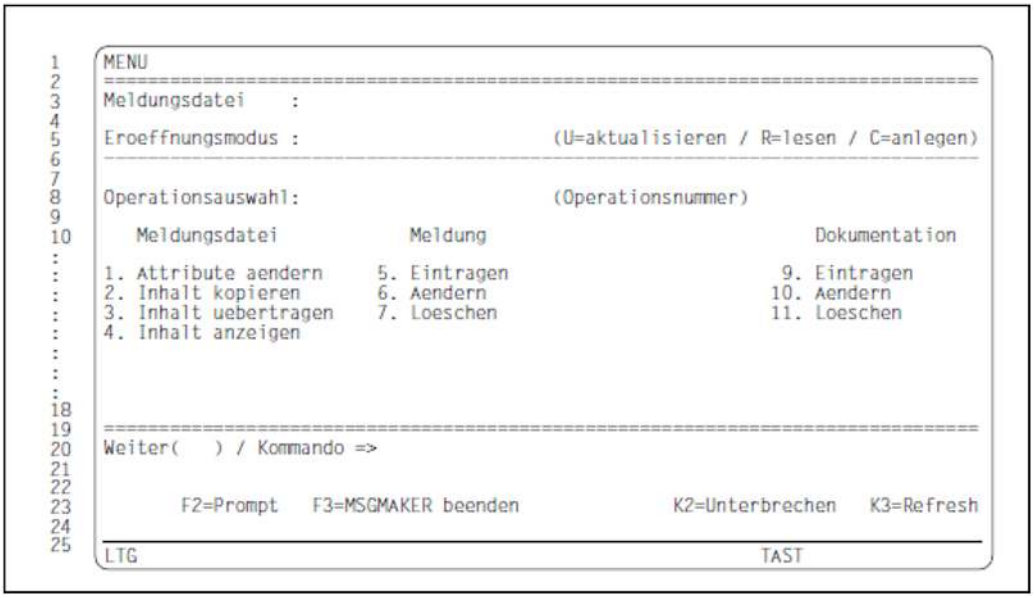
MENU

Diese Maske wird dem Anwender nach Aufruf des Dienstprogramms MSGMAKER angeboten.

Funktion

In dieser Maske wird die zu bearbeitende Meldungsdatei eingegeben. Zugleich kann für diese Datei der als nächstes vorgesehene Bearbeitungsschritt mit einer Nummer ausgewählt werden.
Die Operationen 1, 5, 6, 7, 9, 10 und 11 benötigen die Angabe einer Meldungsdatei, für 2, 3 und 4 muss in die Maske MENU keine Meldungsdatei eingetragen werden.

Maske



Eingabefelder

Meldungsdatei

Name einer Meldungsdatei. Wird die Eingabe mit DUE bestätigt, erscheint der vollqualifizierte Name der Meldungsdatei.

Soll eine neue Meldungsdatei geöffnet werden, muss der Name der bereits geöffneten Datei überschrieben und mit DUE bestätigt werden.

Gültigkeitskriterien:
Datentyp: <filename 1..54>

Eroeffnungsmodus

U (Update):	Eine bereits existierende Meldungsdatei wird eröffnet und soll aktualisiert werden (Voreinstellung /Anzeige).
R (Read):	Die Meldungsdatei wird nur zum Lesen eröffnet.
C (Create):	Eine neue Meldungsdatei wird angelegt; sie ist die aktuelle Arbeitsdatei. Die Maske MSG-FILE-ATTRIBUTES (siehe "Maske MSG-FILE-ATTRIBUTES - Eintragen und Ändern der Meldungsdatei-Attribute") wird automatisch aufgerufen; in ihr können die Attribute der neuen Meldungsdatei angegeben werden.

Wenn der Name einer noch nicht existierenden Datei in Verbindung mit U angegeben wird, gibt MSGMAKER die Fehlermeldung `MSMDJ01` aus.

Operationsauswahl

Durch Eintrag der entsprechenden Nummer kann eine der folgenden Operationen ausgewählt werden.

1 (Meldungsdatei - Attribute aendern)

Aufruf der Maske **MSG-FILE-ATTRIBUTES**.

In dieser Maske können die Attribute der Meldungsdatei verändert werden.

Zu den Datei-Attributen zählen

- die Dateiart. Es werden kundeneigene Meldungsdateien und BS2000-Stan-dard-Meldungsdateien unterschieden.
- der Name des Software-Produkts, für das die Meldungsdatei erzeugt wird.
- die Versionsnummer dieses Software-Produkts.

> ¡ Wird eine Meldungsdatei erzeugt (C=anlegen), wird die Maske MSG-FILE-ATTRIBUTES **automatisch** aufgerufen. Die Ziffer 1 muss eingegeben werden, wenn die Attribute einer bereits bestehenden Meldungsdatei verändert werden sollen.

2 (Meldungsdatei - Inhalt kopieren)

Aufruf der Maske **COPY**.

In dieser Maske werden Bestandteile einer Meldungsdatei über ihren Meldungsschlüssel ausgewählt und von einer Datei in eine andere oder innerhalb derselben Datei kopiert. Die aktuell eröffnete Meldungsdatei braucht weder Sende- noch Empfangsdatei zu sein.

Zu den Bestandteilen einer Meldungsdatei zählen:

- Meldungseinheit(en)
- Dokumentationszeilen
- Komponentenidentifikation und Korrekturinformation (intern)

3 (Meldungsdatei - Inhalt uebertragen)

Aufruf der Maske **MOVE**.

In dieser Maske werden Bestandteile einer Meldungsdatei über ihren Meldungsschlüssel ausgewählt und von einer Datei in eine andere oder innerhalb derselben Datei übertragen. Im Unterschied zur COPY-Funktion, wird der Sendebereich bei der MOVE-Funktion gelöscht. Die aktuell eröffnete Meldungsdatei braucht weder Sende- noch Empfangsdatei zu sein.

Zu den Bestandteilen einer Meldungsdatei zählen:

- Meldungseinheit(en)
- Dokumentationszeilen
- Komponentenidentifikation und Korrekturinformation (intern)

4 (Meldungsdatei - Inhalt anzeigen)

Aufruf der Maske **SHOW**.

In dieser Maske werden Bestandteile einer Meldungsdatei über ihren Meldungsschlüssel ausgewählt und nach SYSOUT oder in eine SYSLST-Datei ausgegeben.

Zu den Bestandteilen einer Meldungsdatei zählen:

- Meldungseinheit(en), aufgeschlüsselt in
  - Meldungsattribute
  - Meldungstext in den definierten Sprachen
  - Bedeutungs- und Maßnahmetext in den definierten Sprachen
  - Inserts und Insert-Attribute
- Dokumentationszeilen
- Komponentenidentifikation und Korrekturinformation (intern)

5 (Meldung - Eintragen)

Aufruf der Maske **ADD-MSG**.

In dieser Maske können neue Meldungseinheiten in die aktuelle Arbeitsdatei eingefügt werden. Über den Meldungsschlüssel werden die neuen Meldungseinheiten in die Datei eingeordnet.

Einer Meldungsdatei können folgende Attribute zugeordnet werden:

- MIP-Zugriffsmethode
- Ausgabeziel
- Berechtigungsschlüssel
- Weight Code
- Garantie
- Kennbuchstabe für die Sprache

Zu jeder definierten Sprache muss ein Meldungstext und kann ein Bedeutungs- und Maßnahmetext geschrieben werden. Ferner können im Meldungstext Inserts definiert werden.

6 (Meldung - Aendern)

Aufruf der Maske **MODIFY-MSG**.

In dieser Maske können Meldungseinheiten der aktuellen Arbeitsdatei verändert werden.

Folgende Attribute einer Meldungsdatei können verändert werden:

- MIP-Zugriffsmethode
- Ausgabeziel
- Berechtigungsschlüssel
- Weight Code
- Garantie
- Kennbuchstabe für die Sprache

In jeder definierten Sprache können die Meldungs-, Bedeutungs- und Maßnahmetexte verändert werden. In gleicher Weise gilt dies für die Inserts und deren Attribute.

7 (Meldung - Loeschen)

Aufruf der Maske **DELETE-MSG**.

In dieser Maske können Meldungseinheiten der aktuellen Meldungsdatei gelöscht werden. Die Meldungseinheiten werden über ihre Meldungsschlüssel ausgewählt.

Es besteht die Möglichkeit

- Texte einer bestimmten Sprache zu löschen
- eine oder mehrere Meldungseinheiten zu löschen
- die Meldungseinheiten in einer weiteren, gleichnamigen Maske ausgeben zu lassen, bevor sie gelöscht werden.

9 (Dokumentation - Eintragen)

Aufruf der Maske **ADD-DOCUMENTATION**.

In dieser Maske können Meldungen dokumentiert werden.

Folgende Angaben sind möglich:

- Meldungsschlüssel/Meldungsintervall
- Name des Meldungsverantwortlichen
- Name des verantwortlichen Teams
- Kommentar

10 (Dokumentation - Ändern)

Aufruf der Maske **MODIFY-DOCUMENTATION**.

In dieser Maske kann die Dokumentation zu Meldungen verändert werden.

Folgende Angaben sind möglich:

- Ändern der Meldungsschlüssel
- Ändern der Meldungsverantwortlichen
- Ändern des verantwortlichen Teams
- Ändern des Kommentars

Veränderte Meldungsschlüssel werden automatisch neu in die Liste einsortiert.

11 (Dokumentation - Loeschen)

Aufruf der Maske **DELETE-DOCUMENTATION**.

In dieser Maske kann die Dokumentation zu einer oder mehreren Meldungen gelöscht werden.

Kommando

Besonderheit

Eine Meldungsdatei, die in das Feld **Meldungsdatei** eingetragen wurde, wird immer vor Ausführung einer eventl. angegebenen Anweisung eröffnet.

> Im Kommandobereich wird ein „+“ angezeigt, wenn eine Standard-Meldungsdatei des Herstellers eröffnet wird. Mit „+“ kann zu einer zweiten MENU-Maske geblättert werden, die Funktionen für den internen Gebrauch anbietet.

Weitere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MENU	-> ?

Operationsnr. 1 - 11: Verzweigen zu den verschiedenen Eingabemasken.

F3 MSGMAKER beenden.

8.2.6.2 Maske MSG-FILE-ATTRIBUTES - Eintragen und Ändern der Meldungsdatei-Attribute

Reihenfolge der Masken

MENU -> 1 -> MSG-FILE-ATTRIBUTES

Funktion

Die Attribute einer Meldungsdatei können eingetragen oder durch Überschreiben verändert werden. Durch Drücken der Taste DUE werden die Eingaben bestätigt und die Maske verlassen.

Ein Wechsel in die Maske MSG-FILE-ATTRIBUTES erfolgt, wenn in der Maske MENU Folgendes angegeben wird:

- eine neue Datei mit **Eroeffnungsmodus**: C oder
- für eine bereits existierende Datei (**Eroeffnungsmodus**: U): **Operationsauswahl**: 1

Der Cursor ist bei Erscheinen der Maske auf dem Feld **Typ** positioniert.

Maske

```
MSG-FILE-ATTRIBUTES
================================================================================



Dateiname: :N:$USER0001.TESTDATEI



Typ      : C                        (C=Kunde: Meldungsdatei eines Kundenprodukts
                                     S=Standard: Meldungsdatei eines bei FTS
                                                 entwickelten Produkts)


Produkt                             (Produkt, zu dem Meldungsdatei gehoert)
  Name    : PROGRAMM                (leer = keine Angabe)
  Version: V1.2A                    (leer = keine Angabe)


================================================================================
Kommando =>


       F2=Prompt                     K1=Abbrechen   K2=Unterbrechen   K3=Refresh

LTG                                                              TAST
```

Ausgabefelder

Dateiname (Name der Meldungsdatei)

Der in die Hauptmaske MENU eingetragene Name wird in die Maske MSG-FILE-ATTRIBUTES übernommen. Der Name kann nicht verändert werden.

Eingabefelder

Typ (Art der Meldungsdatei)

Bezieht sich die Meldungsdatei auf ein Software-Produkt, das beim Hersteller entwickelt worden ist, muss als Dateiart S (Standard) eingetragen werden. Die Voreinstellung
C (Customer) bedeutet, dass die Meldungsdatei für ein kundeneigenes Produkt geschrieben wird.

> Die Umwandlung einer Standard-Meldungsdatei in eine kundeneigene Meldungsdatei durch Abänderung des Eintrags S in C ist **nicht** erlaubt. Es besteht jedoch die Möglichkeit, eine neue Meldungsdatei der Art C zu eröffnen und den Inhalt der Standard-Meldungsdatei in diese zu kopieren. Die Umwandlung einer kundeneigenen Meldungsdatei in eine Standard-Meldungsdatei ist ohne weiteres möglich. Voreinstellung /Anzeige: C (Customer)

Produkt Name (Name des Software-Produkts)

Name des Software-Produkts, für das die Meldungsdatei erstellt wird.

Gültigkeitskriterien:
Datentyp: <structured-name 1..15>
Ein Produktname ist nur erforderlich, wenn eine Versionsnummer angegeben wird.
Der eingetragene Name wird sowohl bei erneutem Aufruf der Maske als auch innerhalb der Meldungsdatei in Großbuchstaben ausgegeben.

Produkt Version (Versionsnummer des Produkts)

Versionsnummer des Software-Produkts, für das die Meldungsdatei erstellt wird.

Gültigkeitskriterien:
Datentyp: <composed-name 3..8> oder <c-string 1..8>
Sind Buchstaben Bestandteil der Versionsnummer, werden diese sowohl bei erneutem Aufruf der Maske als auch innerhalb der Meldungsdatei in Großbuchstaben umgewandelt.Bei Angabe der Versionsnummer ist ein Produktname ebenfalls erforderlich.

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MSG-FILE-ATTRIBUTES	-> ?

DUE	Die Eingaben werden bestätigt; Rückkehr zur Maske MENU.
K1	Die Eingaben werden nicht bestätigt und gehen verloren; Rückkehr zur Maske MENU.

8.2.6.3 Maske COPY - Kopieren von Meldungseinheiten

Reihenfolge der Masken

MENU -> 2 -> COPY

Funktion

In dieser Maske können Bestandteile einer Meldungsdatei durch Angabe des Meldungsschlüssels ausgewählt und in eine andere oder innerhalb derselben Datei kopiert werden. Die aktuelle Meldungsdatei muss weder Sende- noch Empfangsdatei sein.

Die Maske COPY kann, ohne Eintrag des Dateinamens in die Hauptmaske, von MENU aus direkt aufgerufen werden. Wird in MENU eine Meldungsdatei eingetragen, wird der Name in die Felder **Aus Datei** und **In Datei** der Maske COPY übernommen.

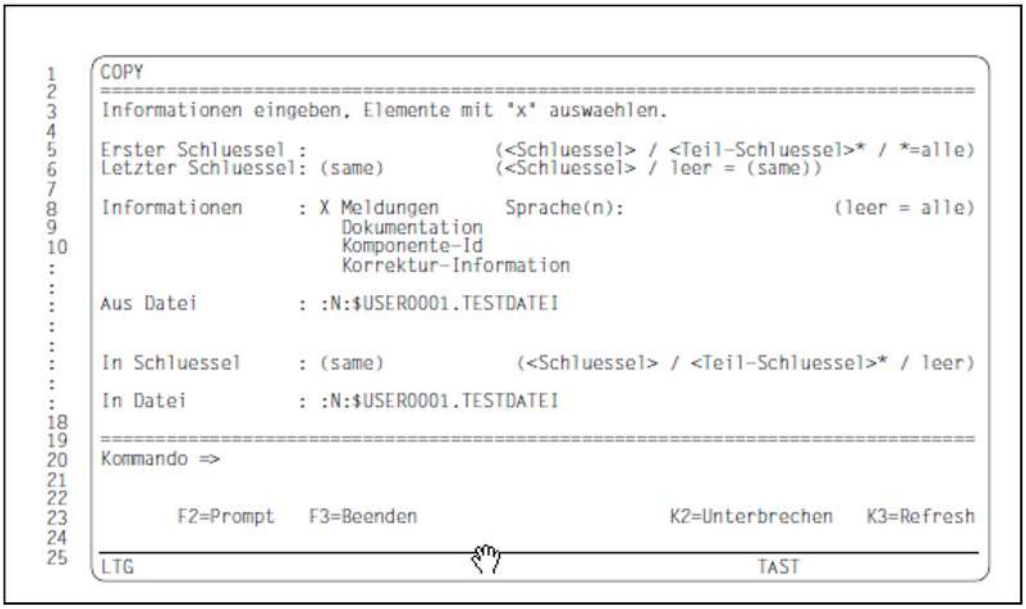
Die COPY-Funktion wird ausgeführt, wenn die Eingaben mit DUE bestätigt werden. Bei erfolgreichem Kopieren gibt MSGMAKER die Meldung `MSMN100` aus.

Besonderheiten des Kopiervorgangs

- Existiert eine angegebene Empfangs-Meldungseinheit noch nicht, wird eine Meldungseinheit mit den Meldungs- und Insertattributen der Sende-Meldungseinheit erzeugt.
- Ist in der Empfangs-Meldungseinheit bereits ein Text unter identischem Meldungsschlüssel und gleichem Kennbuchstaben für die Sprache abgespeichert, wird der Text der Empfangs-Meldungseinheit mit dem Text der Sende-Meldungseinheit überschrieben.
- Unterscheiden sich die Texte nur durch den Kennbuchstaben für die Sprache (z.B. ABC0000D, ABC0000E), wird der Text in der Empfangs-Meldungseinheit angefügt.
- Liegt in der Empfangs-Meldungseinheit durch den Kopiervorgang eine neue, nicht definierte Insert-Nummer vor, werden die entsprechenden Insert-Attribute der Sendeeinheit in die Empfangs-Meldungseinheit kopiert. Im Gegenzug werden Insert-Attribute nicht mehr vorhandener Inserts gelöscht.

> ¡ Soll der Gesamtinhalt einer Meldungsdatei in eine andere Datei kopiert werden, ist die Anweisung //MERGE-MSG-FILE der COPY-Funktion vorzuziehen. Der Kopiervorgang läuft auf diesem Weg viel effizienter ab.

Maske



Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet eine Meldungseinheit oder die erste Meldungseinheit eines Meldungsbereichs.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis sieben Zeichen ersetzen kann.
Die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse).

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> A*, AB*, ABC*, ABC0*, ABC00*, ABC000*

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe im Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

Informationen

Aus dem definierten Meldungsintervall können Meldungseinheiten und
Dokumentationszeilen ausgewählt werden, die in eine andere Datei oder innerhalb derselben Datei kopiert werden können.

Gültigkeitskriterien:
Markieren der Dateiinhalte mit dem Zeichen „X“.

Anzeige: X Meldungen

Meldungen
Meldungseinheiten in den angegebenen Sprachen (siehe Feld **Sprache(n)**), die sich im definierten Meldungsintervall befinden, werden kopiert. Kein Eintrag in das Feld **Sprache(n)** bedeutet, dass alle Meldungseinheiten kopiert werden.

Dokumentation
Alle Dokumentationszeilen, die im angegebenen Meldungsintervall definiert sind, werden kopiert. Überschneiden mehrere definierte Meldungsintervalle den festgelegten Meldungsbereich, werden die Dokumentationszeilen dieser Intervalle ebenfalls kopiert.

Beispiel

Siehe Beschreibung der Anweisung //COPY.

Komponente-Id / Korrektur-Information
Diese Zusatzinformationen sind nur bei BS2000-Standard-Meldungsdateien verfügbar.

Sprache(n) (Kennbuchstabe für die Sprache)

Durch Eingabe des Kennbuchstabens werden die Texte (Meldungs-, Bedeutungs- und Maßnahmetext), die unter diesem Kennbuchstaben eingetragen sind, in die angegebene Empfangs-Meldungseinheit kopiert. Der Empfangsbereich kann entweder erneut in der Sendedatei oder auch in einer weiteren Meldungsdatei (siehe Feld **In Datei**) liegen.

Gültigkeitskriterien:
Die Buchstaben A bis Z können eingetragen werden. Für die Auswahl ist entscheidend, dass die Buchstaben mit den Kennbuchstaben übereinstimmen, welche beim Erstellen der Meldungseinheit vereinbart wurden.

Aus Datei (Sendedatei)

Name der Meldungsdatei, deren Inhalte kopiert werden sollen. Die aktuelle Meldungsdatei, die in der Maske MENU eröffnet und deren Name in das Feld **Aus Datei** der Maske COPY übertragen wurde, kann überschrieben werden. Als Sendedatei kann jede bereits bestehende Meldungsdatei angegeben werden.

Gültigkeitskriterien:
Datentyp: <filename 1..54>

In Schluessel (Meldungsschlüssel der Empfangsdatei)

Der Meldungsbereich, der in der Empfangsdatei reserviert wird, muss mindestens so groß sein wie der in den Feldern **Erster Schluessel** und **Letzter Schluessel** ausgewählte Bereich der Sendedatei.
Wird das Feld **In Schluessel** nicht ausgefüllt, ist dies gleichbedeutend mit der Angabe (same) und bewirkt, dass die Meldungsschlüssel der Sendedatei gleichlautend in die Empfangsdatei übernommen werden.
(same) darf nicht angegeben werden, wenn die Sendedatei zugleich Empfangsdatei ist.(same) **muss** angegeben werden, wenn

- im Feld **Erster Schluessel**
  - das Schlüsselwort (list) steht
  - die Angabe * steht
  - mehrere Meldungsklassen durch den Platzhalter * ausgewählt wurden (Bsp: AB*)
- in den Feldern **Erster Schluessel** und **Letzter Schluessel** ein Meldungsintervall über mehrere Meldungsklassen definiert wurde (Bsp: ABC0000 - ABD9999).

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis vier Zeichen ersetzen kann.

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> ABC*, ABC0*, ABC00*, ABC000*

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

In Datei (Name der Empfangsdatei)

Name der Meldungsdatei, in welche die Dateiinhalte der Sendedatei kopiert werden. Die aktuelle Meldungsdatei, die in der Maske MENU eröffnet und deren Name in das Feld
In Datei übertragen wurde, kann in der Maske COPY überschrieben werden. Als Empfangsdatei kann jede bereits bestehende Meldungsdatei angegeben werden. Existiert unter dem angegebenen Namen noch keine Meldungsdatei, dann wird diese erzeugt. Die Dateiart (Customer oder Standard), sowie Produktname und Versionsnummer werden analog zur Sendedatei festgelegt.

Gültigkeitskriterien:
Datentyp: <filename 1..54>

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

COPY -> ?

DUE Die Funktion COPY wird ausgelöst; die Maske COPY wird in ihrem ursprünglichen Zustand wieder angezeigt. Die Funktion kann erneut aufgerufen werden, indem in die nun leeren „Schluessel“-Felder wieder Werte eingetragen werden.

F3 Die Maske Copy wird verlassen, ohne dass die Funktion COPY ausgeführt wird. Die Ausgangs-Maske MENU wird angezeigt.

8.2.6.4 Maske MOVE - Kopieren und Löschen von Meldungseinheiten

Reihenfolge der Masken

MENU -> 3 -> MOVE

Funktion

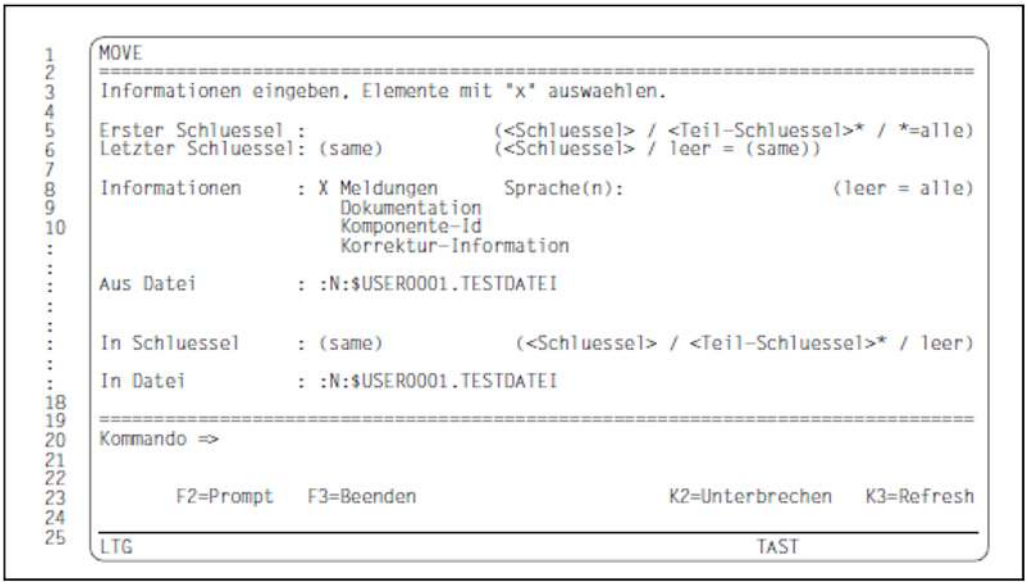
In dieser Maske können Bestandteile einer Meldungsdatei durch Angabe des Meldungsschlüssels ausgewählt und in eine andere oder innerhalb derselben Datei übertragen werden. Im Unterschied zur COPY-Funktion, wird der Sendebereich bei der MOVE-Funktion gelöscht.
Die aktuelle Meldungsdatei muss weder Sende- noch Empfangsdatei sein. Die Maske MOVE kann direkt, ohne einen Eintrag des Dateinamens von MENU aus aufgerufen werden. Wird in MENU eine Meldungsdatei eingetragen, wird der Name in die Felder **Aus Datei** und **In Datei** der Maske MOVE übernommen.

Die MOVE-Funktion wird ausgeführt, wenn die Eingaben mit DUE bestätigt werden. Bei erfolgreichem Übertragen gibt MSGMAKER die Meldung `MSMN100` aus.

Besonderheiten der Übertragung (Kopieren + Löschen)

- Existiert die angegebene Empfangs-Meldungseinheit nicht, wird eine Meldungseinheit mit den Meldungsattributen der Sende-Meldungseinheit erzeugt.
- Ist in der Empfangs-Meldungseinheit bereits ein Text unter identischem Meldungsschlüssel und gleichem Kennbuchstaben für die Sprache abgespeichert, wird der Text der Empfangs-Meldungseinheit mit dem Text der Sende-Meldungseinheit überschrieben.
- Unterscheiden sich die Texte nur durch den Kennbuchstaben für die Sprache (z.B. ABC0000D, ABC0000E), wird der Text in der Empfangs-Meldungseinheit angefügt.
- Liegt in der Empfangs-Meldungseinheit durch die Übertragung eine neue, nicht definierte Insert-Nummer vor, werden die entsprechenden Insert-Attribute der Sendeeinheit in die Empfangs-Meldungseinheit kopiert. Im Gegenzug werden Insert-Attribute nicht mehr vorhandener Inserts gelöscht.
- Sind alle sprachabhängigen Teile einer Meldungseinheit durch die Funktion MOVE aus einer Meldungseinheit entfernt worden, werden die verbliebenen Meldungsattribute und somit die Meldungseinheit automatisch gelöscht.

Maske



Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet eine Meldungseinheit oder die erste Meldungseinheit eines Meldungsbereichs.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis sieben Zeichen ersetzen kann.

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> A*, AB*, ABC*, ABC0*, ABC00*, ABC000*

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe in das Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

Informationen (Auswahl der zu übertragenden Bestandteile einer Meldungsdatei)

Aus dem definierten Meldungsintervall können Meldungseinheiten und Dokumentationszeilen ausgewählt werden, die in eine andere Datei oder innerhalb derselben Datei kopiert werden können. Im Unterschied zur COPY-Funktion, wird der Sendebereich bei der MOVE-Funktion gelöscht.

Gültigkeitskriterien:
Markieren der Dateiinhalte mit dem Zeichen „X“.

Anzeige: X Meldungen

Meldungen
Meldungseinheiten in den angegebenen Sprachen (siehe Feld **Sprache(n)**), die sich im definierten Meldungsintervall befinden, werden in den Empfangsbereich kopiert und im Sendebereich gelöscht. Kein Eintrag im Feld **Sprache(n)** bedeutet, dass alle Meldungseinheiten kopiert werden.

Dokumentation
Alle Dokumentationszeilen, die im angegebenen Meldungsintervall definiert sind, werden kopiert und anschließend im Sendebereich gelöscht. Überschneiden mehrere definierte Meldungsintervalle den festgelegten Meldungsbereich, werden die Dokumentationszeilen dieser Intervalle ebenfalls kopiert und anschließend im Sendebereich gelöscht.

Beispiel

Siehe Beschreibung der Anweisung //MOVE.

Komponente-Id / Korrektur-Information
Diese Zusatzinformationen sind nur bei BS2000-Standard-Meldungsdateien verfügbar.

Sprache(n) (Kennbuchstabe für die Sprache)

Durch Eingabe des Kennbuchstabens werden die Texte (Meldungs-, Bedeutungs- und Maßnahmetext), die unter diesem Kennbuchstaben eingetragen sind, in die angegebene Empfangs-Meldungseinheit kopiert.
Im Anschluss an den Kopiervorgang wird der Sendebereich gelöscht.

Gültigkeitskriterien:
Die Buchstaben A bis Z können eingetragen werden. Für die Auswahl ist entscheidend, dass die Buchstaben mit den Kennbuchstaben übereinstimmen, welche beim Erstellen der Meldungseinheit vereinbart wurden.

Aus Datei (Sendedatei)

Name der Meldungsdatei, deren Inhalte kopiert werden sollen. Der Sendebereich dieser Datei wird anschließend gelöscht. Die aktuelle Meldungsdatei, die in der Maske MENU eröffnet und deren Name in das Feld **Aus Datei** übertragen wurde, kann in der Maske MOVE überschrieben werden. Als Sendedatei kann jede bereits bestehende Meldungsdatei angegeben werden.

Gültigkeitskriterien:
Datentyp: <filename 1..54>

In Schluessel (Meldungsschlüssel der Empfangsdatei)

Der Meldungsbereich, der in der Empfangsdatei reserviert wird, muss mindestens so groß sein wie der in den Feldern **Erster Schluessel** und **Letzter Schluessel** ausgewählte Bereich der Sendedatei.
Wird das Feld **In Schluessel** nicht ausgefüllt, ist dies gleichbedeutend mit der Angabe (same) und bewirkt, dass die Meldungsschlüssel der Sendedatei gleichlautend in die Empfangsdatei übernommen werden.
(same) darf nicht angegeben werden, wenn die Sendedatei zugleich Empfangsdatei ist.(same) **muss** angegeben werden, wenn

- im Feld **Erster Schluessel**
  - das Schlüsselwort (list) steht
  - die Angabe * steht
  - mehrere Meldungsklassen durch den Platzhalter * ausgewählt wurden (Bsp: AB*)
- in den Feldern **Erster Schluessel** und **Letzter Schluessel** ein Meldungsintervall über mehrere Meldungsklassen definiert wurde (Bsp: ABC0000 - ABD9999).

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis vier Zeichen ersetzen kann.

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> ABC*, ABC0*, ABC00*, ABC000*

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

In Datei (Name der Empfangsdatei)

Name der Meldungsdatei, in welche die Dateiinhalte der Sendedatei übertragen werden. Die aktuelle Meldungsdatei, die in der Maske MENU eröffnet und deren Name in das Feld **In Datei** übertragen wurde, kann in der Maske MOVE überschrieben werden. Als Empfangsdatei kann jede bereits bestehende Meldungsdatei angegeben werden. Existiert unter dem angegebenen Namen noch keine Meldungsdatei, dann wird diese erzeugt. Die Dateiart (Customer oder Standard), sowie Produktname und Versionsnummer werden analog zur Sendedatei festgelegt.

Gültigkeitskriterien:
Datentyp: <filename 1..54>

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MOVE -> ?

DUE	Die Funktion MOVE wird ausgelöst; die Maske MOVE wird in ihrem ursprünglichen Zustand wieder angezeigt. Die Funktion kann erneut aufgerufen werden, indem in die nun leeren „Schluessel“-Felder wieder Werte eingetragen werden.
F3	Die Maske MOVE wird verlassen, ohne dass die Funktion MOVE ausgeführt wird. Die Ausgangs-Maske MENU wird angezeigt.

8.2.6.5 Maske SHOW - Inhalt einer Meldungsdatei anzeigen

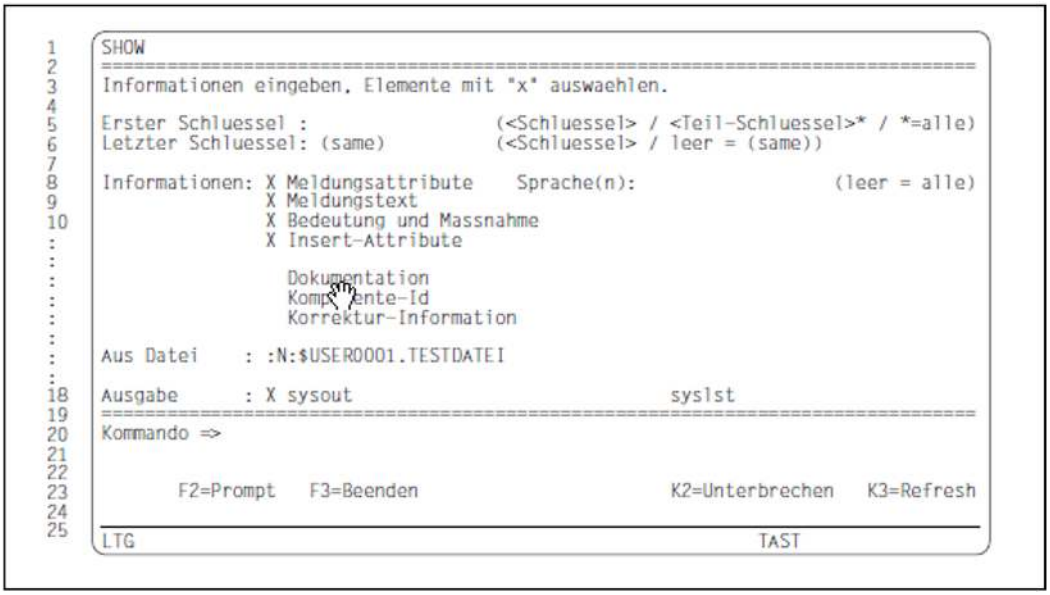
Reihenfolge der Masken

MENU -> 4 -> SHOW

Funktion

In dieser Maske können Bestandteile einer Meldungsdatei durch Angabe des Meldungsschlüssels ausgewählt und nach SYSOUT oder in eine SYSLST Datei ausgegeben werden. Die Meldungseinheiten werden aufgeschlüsselt nach ihren Attributen, dem Meldungs-, Bedeutungs- und Maßnahmetext in den angegebenen Sprachen und den Insert-Attributen angezeigt. Ferner können Dokumentationszeilen zur Anzeige gebracht werden. Diese Bestandteile der Meldungsdatei werden sortiert nach Meldungsklassen ausgegeben. Die Ausgabe nach SYSOUT erfolgt in eine separate Maske (siehe "Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen", Maske SHOW-OUTPUT).

Maske



Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet eine Meldungseinheit oder die erste Meldungseinheit eines Meldungsbereichs.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis sieben Zeichen ersetzen kann.
Die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse).

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> A*, AB*, ABC*, ABC0*, ABC00*, ABC000*

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe im Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

Informationen
(Bestandteile der Meldungsdatei, die nach SYSOUT oder SYSLST ausgegeben werden)

Aus dem definierten Meldungsintervall können Meldungseinheiten und
Dokumentationszeilen ausgewählt werden, die nach SYSOUT oder SYSLST ausgegeben werden.

Gültigkeitskriterien:
Markieren der Dateiinhalte mit dem Zeichen „X“.

Anzeige:

- X Meldungsattribute
- X Meldungstext
- X Bedeutung und Massnahme
- X Insert-Attribute

Meldungsattribute

Die Meldungsattribute der im angegebenen Meldungsintervall definierten Meldungseinheiten werden ausgegeben.
Zu den Meldungsattribute zählen die Zugriffsmethode, das Ausgabeziel, der Berechtigungsschlüssel, die Schreibweise des Textes bei der Ausgabe, der Weight Code und die „Garantie“ der Meldung.

Meldungstext
Ausgabe des Meldungstextes in den angegebenen Sprachen (siehe Feld **Sprache(n)**). Kein Eintrag in das Feld **Sprache(n)** bedeutet, dass die Meldungstexte in allen definierten Sprachen ausgegeben werden.

Bedeutung und Massnahme
Ausgabe von Bedeutungs- und Maßnahmetext in den angegebenen Sprachen. Fehlt eine Sprachangabe im Feld **Sprache(n)**, werden Bedeutungs- und Maßnahmetexte in allen definierten Sprachen ausgegeben.

Insert-Attribute
Ausgabe der Insert-Attribute.

Dokumentation
Alle Dokumentationszeilen, die im angegebenen Meldungsintervall definiert sind, werden ausgegeben. Überschneiden mehrere definierte Meldungsintervalle den festgelegten Meldungsbereich, werden die Dokumentationszeilen dieser Intervalle ebenfalls ausgegeben. Siehe Beschreibung der Anweisung //SHOW.

Komponente-Id / Korrektur-Information

Diese Zusatzinformationen sind nur bei BS2000-Standard-Meldungsdateien verfügbar.

Sprache(n) (Kennbuchstabe für die Sprache)

Durch Eingabe des Kennbuchstabens werden diejenigen Meldungseinheiten ausgewählt, die Texte in dieser Sprache enthalten. Die Ausgabe der Texte erfolgt in der Reihenfolge, in der die Kennbuchstaben eingetragen wurden.
Kein Eintrag bedeutet die Auswahl aller Meldungseinheiten. In diesem Fall werden die Texte in der alphabetischen Reihenfolge der Sprach-Kennbuchstaben ausgegeben.
Wird für die ausgewählten Meldungseinheiten eine nicht definierte Sprache angegeben, werden die Meldungsattribute trotz einer eventuellen Markierung (X Meldungsattribute) nicht angezeigt.

Gültigkeitskriterien:
Die Buchstaben A bis Z können eingetragen werden. Für die Auswahl ist entscheidend, dass die Buchstaben mit den Kennbuchstaben übereinstimmen, welche beim Erstellen der Meldungseinheit vereinbart wurden.

Aus Datei (Meldungsdatei)

Bezeichnet die Meldungsdatei, deren Bestandteile ausgegeben werden sollen. Der aus der Maske MENU übertragene Dateinamen kann überschrieben werden.

Gültigkeitskriterien:
Datentyp: <filename 1..54 without-gen-vers>

Ausgabe (Ausgabeziel)

Die ausgewählten Inhalte der Meldungsdatei können nach SYSOUT und/oder SYSLST ausgegeben werden.

Gültigkeitskriterien:
Markieren mit dem Zeichen „X“.

Anzeige: X sysout

sysout
Die ausgewählten Bestandteile der Meldungsdatei werden in der Maske SHOW-OUTPUT (siehe "Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen") auf der Datensichtstation des Anwenders ausgegeben.

syslst
Die ausgewählten Bestandteile der Meldungsdatei werden in die Systemdatei SYSLST ausgegeben. Der Text wird mit 60 Zeilen pro Seite ausgegeben.

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

SHOW -> ?

DUE Die Funktion SHOW wird ausgelöst; die angeforderten Meldungseinheiten werden ausgegeben.

F3 Die Maske SHOW wird verlassen, ohne dass die Funktion SHOW ausgeführt wird. Die Ausgangs-Maske MENU wird angezeigt.

8.2.6.6 Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen

Funktion

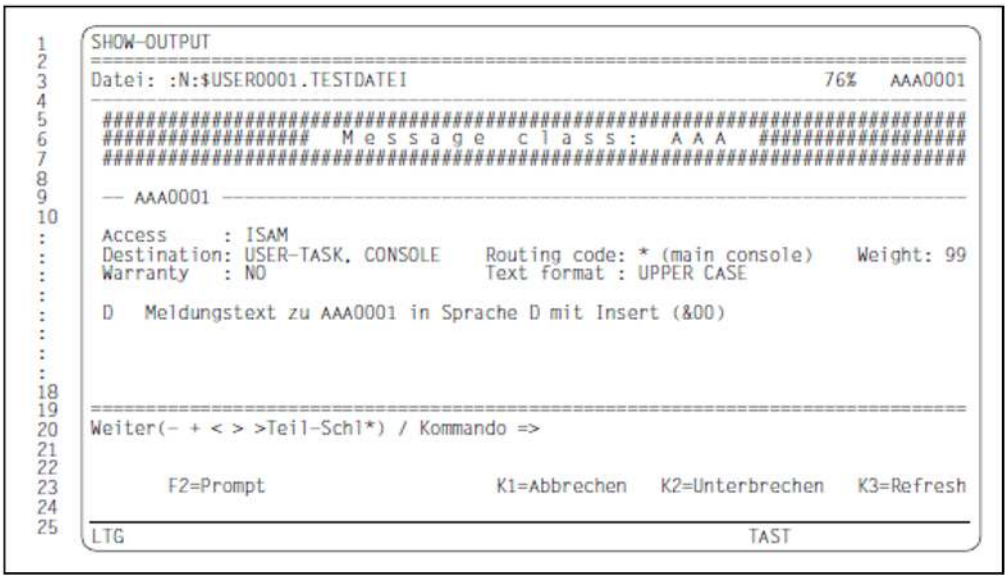
Im Anschluss an die Maske SHOW wird die Maske SHOW-OUTPUT aufgerufen. Die Informationen zu den ausgewählten Meldungseinheiten sowie die Dokumentationszeilen werden in nachstehender Reihenfolge nach SYSOUT (ebenso nach SYSLST) ausgegeben.
Bei Ausgabe nach SYSOUT besteht die Möglichkeit, in der angezeigten Meldungsdatei vor- und rückwärtszublättern (siehe "Maske SHOW-OUTPUT Ausgabe von Meldungseinheiten und Zusatzinformationen").

> ℹ Werden Meldungseinheiten aus mehreren Meldungsklassen angezeigt, erscheinen die Meldungsklassen in alphabetischer Reihenfolge.

1. Meldungsklasse
   Eine neue Meldungsklasse wird durch einen Hinweis z.B. „Message Class: AAA“ angekündigt. Dieser Text wird, umrandet von dem Zeichen „#“, ausgegeben.
2. Komponenten-Identifikation
   Erscheint nur bei Standardmeldungen (intern).
3. Meldungseinheiten
   Der Inhalt einer Meldungseinheit wird in nachstehender Reihenfolge angezeigt:
   - Meldungsattribute
   - Insert-Attribute
     Es werden nur Inserts angezeigt, für die ein Name oder Default-Text vereinbart wurde.
   - Kennbuchstabe für die Sprache
     Meldungstexte (und wenn definiert, Bedeutungs- und Maßnahmetexte) werden zu jeder Sprache angezeigt. Im angezeigten Meldungstext sind die Trennungszeichen „^“ bereits umgesetzt worden, d.h. der Text, der auf ein „^“ folgt, beginnt in einer neuen Zeile.
     Die Reihenfolge der Sprachen richtet sich nach der Eingabe der Kennbuchstaben in die Maske SHOW; ist keine Sprache ausgewählt worden, werden die Kennbuchstaben alphabetisch angezeigt.
4. Dokumentationszeilen
5. Korrektur-Informationen
   Erscheint nur bei Standard-Meldungen (intern).

Maske *(Beispiel)*

In der Meldungsdatei :N:$USER0001.TESTDATEI wurden drei Meldungseinheiten (AAA0001 bis AAA0003) mit Meldungs-, Bedeutungs- und Maßnahmetext definiert.
Die drei Meldungseinheiten sollen auf SYSOUT ausgegeben werden. Die Meldungseinheiten werden für die Ausgabe aufbereitet und die aufbereitete Gesamtinformation auf 100% skaliert. Die nachfolgenden Bildschirmausgabe ist die dritte von insgesamt vier Ausgaben.
Im Informationsbereich des Bildschirms ist zu sehen, dass bereits 76% der aufbereiteten Informationsmenge ausgegeben ist.



Ausgabefelder

Im Informationsbereich

Datei (Name der Meldungsdatei)

Name der Meldungsdatei, deren Inhalt angezeigt wird.

XX% (Informationsmenge in %)

Der ausgewählte Meldungsbereich einer Meldungsdatei wird vor der Ausgabe formatiert. 0% bedeuten den Beginn, 100% das Ende der aufbereiteten Information.

AAA (Meldungsklasse der angezeigten Information)

Die Meldungsklasse wird angezeigt, wenn der erste Meldungsschlüssel innerhalb einer Meldungsklasse oder Dokumentationszeilen ausgegeben werden.

AAAXXXX (Meldungsschlüssel der ersten, im Arbeitsbereich sichtbaren Meldungseinheit)

Ein siebenstelliger Meldungsschlüssel verweist auf die Meldungseinheit, deren Ende im oberen Arbeitsbereich noch sichtbar ist.

Im Arbeitsbereich

Kommt eine neue Meldungsklasse zur Anzeige, wird sie durch einen dreizeiligen Kommentar (siehe „Funktion“) angekündigt.

AAAXXXX (Meldungsschlüssel der angezeigten Information)

Die Buchstaben AAA stehen für die Meldungsklasse, XXXX für die Meldungsnummer.

Access	(MIP-Zugriffsmethode)
Destination	(Ausgabeziele der Meldungen)
Warranty	(Garantie der Meldung)
Routing Code	(Berechtigungsschlüssel)

Text Format	(Schreibweise der Texte, bei Ausgabe durch den Makro MSG7X oder durch das Kommando /HELP-MSG-INFORMATION)
Weight	(Meldungsgewicht)
Insert attributes	(Nummern und Default-Texte der definierten Inserts)

Neben der Insert-Nummer werden im Feld **Name** der Insertname und im Feld **Default Value** der Default-Text, eingeschlossen in Hochkommata, ausgegeben.

Die Hochkommata sind **nicht** Bestandteil des Default-Textes. Eine Eingabe von zwei hintereinanderfolgenden Hochkommata " im Feld **Default Value** entspricht der Angabe DEFAULT-VALUE=*EMPTY-STRING.

Der Insertname wird immer in Großbuchstaben ausgegeben. Dieser Name ist Bestandteil der S-Variablen, die der Anwender deklarieren kann. Der Default-Text bildet den Default-Inhalt dieser S-Variablen. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Im Feld **Automatic Help** wird angegeben, ob die automatische Hilfsfunktion zur Verfügung steht. Wird über ein Insert ein Meldungsschlüssel oder ein Teil davon ausgegeben (z.B. bei DVS-Fehlern wird lediglich die Meldungsnummer ausgegeben), so wird über die automatische Hilfsfunktion MIP dazu veranlasst, zusätzlich zum Meldungsschlüssel automatisch den zugehörigen Meldungstext auszugeben. Nähere Informationen hierzu siehe "ADD-MSG - Einfügen einer Meldungseinheit".

Neben dem Kennbuchstaben für die Sprache erscheint der Meldungstext, in welchem die Inserts **nicht** ersetzt sind.

Es folgen Bedeutungstext (gekennzeichnet durch ?) und Maßnahmetext (gekennzeichnet durch !).

Dokumentation:

AAAXXXX - AAAXXX (Meldungsintervall)

Owner Name	(Für diese Meldungen verantwortliche Person)
Team	(Für diese Meldungen verantwortliches Team)

Die Kommentarzeile schließt die Dokumentation zu jedem Meldungsintervall ab.

Bei Standard-Meldungsdateien des Herstellers wird die Komponenten-Identifikation zusätzlich ausgegeben.

Component	(Komponentenname)
Domain	(Anwendungsbereich)
Version	(Version der Komponente)
Owner	(Person oder Team, verantwortlich für diese Komponente)
Date	(Freigabedatum der Meldungsdatei)

Im Kommandobereich

Weiter (- + <>> Teil-Schl*)

Zum Blättern innerhalb der angezeigten Information sind folgende Aktionen möglich:

- oder +	Wechsel zur vorausgegangenen oder nächsten Maske.

-i oder +i	Bewegt den angezeigten Inhalt um i Zeilen rückwärts oder vorwärts
- oder ++	Blättert zum Beginn oder Ende der formatierten Information
<	Positioniert auf den Beginn einer Information (Meldungseinheit, Dokumentation), wenn gerade ein anderer Bereich dieser Information angezeigt wird. Ist der Beginn der Information bereits erreicht, bewirkt „<“ die Positionierung auf den Beginn der vorausgehenden Information.
>	Positioniert auf den Beginn der nächsten Information.
<<	Positioniert auf den Beginn einer gerade angezeigten Meldungsklasse. Ist der Beginn der Meldungsklasse bereits erreicht, wird auf die vorausgehende Meldungsklasse positioniert.
>>	Positioniert auf den Beginn der nächsten Meldungsklasse oder auf das Ende der formatierten Information, wenn keine weiter Meldungsklasse mehr vorhanden ist.
>Teil-Schl*	*Gilt für Teil-Schl größer 3 Zeichen:* Positioniert auf die erste Meldungseinheit, deren Meldungsschlüssel mit dem angegebenen Teilschlüssel übereinstimmt. Entspricht kein Meldungsschlüssel diesem Teilschlüssel, wird die erste Meldungseinheit, die auf den Teilschlüssel folgt, angezeigt. Diese Meldungseinheit kann auch einer neuen Meldungsklasse angehören. *Gilt für Teil-Schl kleiner gleich 3 Zeichen:* Positioniert auf die erste Meldungseinheit der angegebenen Meldungsklasse (z.B. >TST*) oder auf die folgende Klasse (z.B. TSU). Existiert keine weitere Meldungsklasse mehr, wird das Ende des formatierten Textes angezeigt. ℹ Das Zeichen * ist keine notwendige Eingabe.

Kommando

Wird eine Anweisung in den Kommandobereich eingegeben, um eine angezeigte Meldungseinheit zu ändern (z.B. DELETE-MSG), wird die formatierte Information zu dieser Meldungseinheit **nicht** aktualisiert, d.h. eine Änderung innerhalb der Meldungseinheit erscheint nicht auf dem Bildschirm. Eine aktualisierte Anzeige erfolgt erst bei einem erneuten Aufruf der Maske SHOW.

Weitere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

SHOW-OUTPUT -> ?

+/- DUE	Vor- und Rückwärtsblättern in der formatierten Information (nähere Informationen siehe "Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen")
K1	Die Maske SHOW-OUTPUT wird verlassen; die Ausgangs-Maske SHOW wird angezeigt.

8.2.6.7 Maske ADD-MSG - Einfügen einer Meldungseinheit

Reihenfolge der Masken

MENU -> 5 -> ADD-MSG

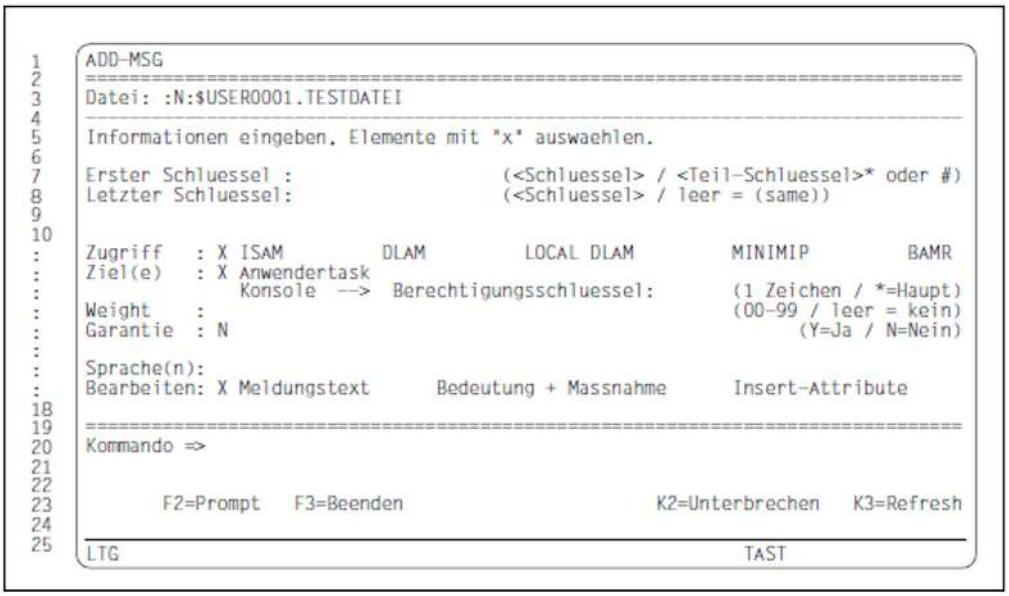
Funktion

Die Maske ADD-MSG ist die erste von insgesamt vier Masken, die zum Einfügen einer neuen Meldungseinheit in die aktuelle Meldungsdatei aufgerufen wird. Der angegebene Meldungsbereich legt die Anzahl der neuen Meldungseinheiten und die Stelle innerhalb der Meldungsdatei fest, an welcher die Meldungseinheiten eingefügt werden.
Die Eingaben in die Maske ADD-MSG müssen mit DUE bestätigt werden; anschließend verzweigt das Programm zu den weiteren Masken.

Eine neue Meldungseinheit wird erst dann abgespeichert, wenn zu allen definierten Sprachen die notwendigen Angaben vorliegen (Meldungstext, Zugriffsmethode, ...).

In eine bereits bestehende Meldungseinheit können Texte in neuen Sprachen nicht eingefügt werden. Die Eingabe eines Meldungsschlüssels, der bereits existiert, wird als Änderungswunsch interpretiert und MSGMAKER wechselt automatisch zur Maske MODIFY-MSG.

Maske



Ausgabefelder

Datei (Name der Meldungsdatei)

Der in der Hauptmaske MENU eingetragene Dateiname wird in die Maske ADD-MSG übernommen. Der Name kann nicht verändert werden.

Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Wird im Feld **Erster Schluessel** ein Meldungsbereich angegeben (Wildcard-Angabe), ermittelt MSGMAKER alle, in diesem Bereich noch nicht definierten Meldungsschlüssel und bringt sie der Reihe nach zur Anzeige. Die Definition eines Meldungsbereichs bietet sich an, wenn für alle Meldungseinheiten dieselben Meldungsattribute gelten sollen. Die Attribute werden zu Beginn des Einfügevorgangs in die Maske ADD-MSG eingetragen und gelten solange für alle weiteren Meldungseinheiten des definierten Bereichs, bis eine Änderung vom Anwender erfolgt. MSGMAKER kehrt nach Durchlaufen der Masken ADD-MSG, MSG-TEXT, MEANING/REPONSE, INSERT-ATTRIBUTES immer wieder zur Ausgangmaske ADD-MSG zurück.

MSGMAKER fährt mit demjenigen Meldungsschlüssel fort, der im definierten Meldungsintervall als nächstes auf den vom Anwender eben eingetragenen Meldungsschlüssel folgt. Die Meldungsattribute sind entweder aufgrund des Eintrags in die erste Maske ADD-MSG für alle weiteren Meldungseinheiten gültig oder können nach Bedarf ebenfalls abgeändert werden.
Jede Meldungseinheit wird sofort in der Meldungsdatei abgespeichert, nicht erst nach Bearbeitung des gesamten Meldungsintervalls.
Sollen zu einem von MSGMAKER angebotenen Meldungsschlüssel keine Angaben erfolgen, kann der Anwender mit K1 diesen Meldungsschlüssel überspringen.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in
welchem der Platzhalter * ein bis sieben Zeichen und der Platzhalter # eine bis vier Ziffern ersetzen kann. Die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse).

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* oder # möglich: A*, AB*, ABC*, ABC0*, ABC00*, ABC000* ABC#, ABC0#, ABC00#, ABC000#

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe im Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Zugriff (MIP-Zugriffsmethoden für die Meldungen)

Bezeichnet die verschiedenen Zugriffsmethoden der Systemkomponente MIP auf die Meldungen. Die Auswahl mehrerer Zugriffsmethoden ist möglich, jedoch dürfen die Methoden ISAM und DLAM bzw. ISAM und LOCAL-DLAM nicht kombiniert werden.

Gültigkeitskriterien:
Markieren der Zugriffsmethoden mit dem Zeichen „X“.

Voreinstellung/Anzeige: X ISAM

ISAM
Die Meldungen werden über den ISAM-Schlüssel gesucht.

DLAM
Diese Zugriffsmethode ist für besonders häufige Meldungen vorgesehen. Wird eine Meldungsdatei, die eine DLAM-Meldung enthält, aktiviert, wird die DLAM-Meldung in den Arbeitsspeicher geladen. Die DLAM-Meldung kann von MIP direkt, ohne Zugriff auf die Meldungsdatei, ausgegeben werden.

LOCAL-DLAM / MINIMIP / BAMR
Diese Zugriffsmethoden sind für den internen Gebrauch beim Hersteller reserviert (siehe DHB-Abschnitt MIP).

Ziel(e) (Ausgabeziele der Meldungen)

Diese Angabe dient dem Anwender zur Dokumentation der Ausgabeziele. Der Makro MSG7X und das Kommando /HELP-MSG-INFORMATION werten diese Angabe **nicht** aus.

Gültigkeitskriterien:
Markieren der Ausgabeziele mit dem Zeichen „X“.

Voreinstellung/Anzeige: X Anwendertask

Anwendertask
Die Meldung hat das Ausgabeziel SYSOUT, SYSLST oder einen benutzereigenen Speicherbereich.

Konsole
Das Ausgabeziel der Meldung ist eine Konsole. Der Berechtigungsschlüssel wird als Zielangabe ausgewertet, wenn neben der Hauptkonsole auch Nebenkonsolen als Meldungsziel verwendet werden.

Berechtigungsschlüssel

Ein Berechtigungsschlüssel muss angegeben werden, wenn im Feld **Ziel(e)** das Ausgabeziel **Konsole** markiert wurde.
Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Gültigkeitskriterien:
Datentyp: <alphanum-name 1..1>

Weight (Meldungsgewicht)

Ein Meldungsgewicht muss angegeben werden, wenn im Feld **Ziel(e)** das Ausgabeziel **Konsole** markiert wurde. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Gültigkeitskriterien:Datentyp: <integer 0..99>

Garantie (Garantie der Meldung)

Das Meldungsattribut „Garantie“ wird von MIP ausgewertet. Die Garantieerklärung besagt, dass bestimmte Teile der Meldung in den kommenden BS2000-Versionen nicht mehr geändert werden.

Folgende Meldungs-Bestandteile werden garantiert:

- Meldungsschlüssel
- Nummerierung und Bedeutung der Inserts

Der Meldungstext wird **nicht** garantiert.

MIP erzeugt für garantierte Meldungen S-Variablen. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Sprache(n) (Kennbuchstabe für die Sprache)

Jede Sprache wird durch einen Kennbuchstaben abgekürzt. Für Deutsch sollte der Buchstabe D, für Englisch E verwendet werden.

Gültigkeitskriterien:
Die Buchstaben „A“ bis „Z“.

Die Reihenfolge der Eingaben bestimmt die Anzeige in den weiteren Masken.

Bearbeiten (Auswahl der Texte)

Für jede Meldungseinheit **muss** zumindest ein Meldungstext definiert werden. Bedeutungs- und Maßnahmetext können nach Bedarf eingegeben werden.

Gültigkeitskriterien:
Markieren der Texte mit dem Zeichen „X“.

Voreinstellung/Anzeige: X Meldungstext

Meldungstext
Im Anschluss an die Maske ADD-MSG wird die Maske MSG-TEXT angezeigt, in welche die Meldungstexte in den verschiedenen Sprachen eingetragen werden können.
Beschreibung der Maske MSG-TEXT siehe "Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes".

Bedeutung + Massnahme
Im Anschluss an die Maske MSG-TEXT wird zur Maske MEANING/RESPONSE verzweigt, in welcher die Bedeutungs- und Maßnahmetexte eingetragen werden können. Beschreibung der Maske MEANING /RESPONSE siehe "Maske MEANING/RESPONSE - Einfügen oder Ändern des Bedeutungs- und Maßnahmetextes".

Insert-Attribute
Im Anschluss an die Masken MSG-TEXT und MEANING/RESPONSE wird zur Maske INSERT-ATTRIBUTES verzweigt, in welcher den Inserts, die im Meldungstext definiert wurden, Namen und Default-Texte zugewiesen

werden. Beschreibung der Maske INSERT-ATTRIBUTES siehe "Maske INSERT-ATTRIBUTES - Einfügen oder Ändern der Insert-Attribute".

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

ADD-MSG -> ?

DUE	Die Funktion ADD-MSG wird gestartet; MSGMAKER verzweigt zu der Maske MSG-TEXT.
K1	Wird ein Meldungsintervall in die Maske ADD-MSG eingetragen, bietet MSGMAKER ab dem zweitem Meldungsschlüssel dieses Intervalls die Funktion K1 = Ueberspringen an. Der Anwender erhält die Möglichkeit, diesen Meldungsschlüssel zu überspringen. Im Anschluss daran erscheint die Maske ADD-MSG mit dem nächsten Meldungsschlüssel.
F3	Die Funktion ADD-MSG wird abgebrochen; MSGMAKER kehrt zur Ausgangsmaske MENU zurück.

8.2.6.8 Maske MODIFY-MSG - Meldungseinheit ändern

Reihenfolge der Masken

MENU	-> 6 ->	MODIFY-MSG

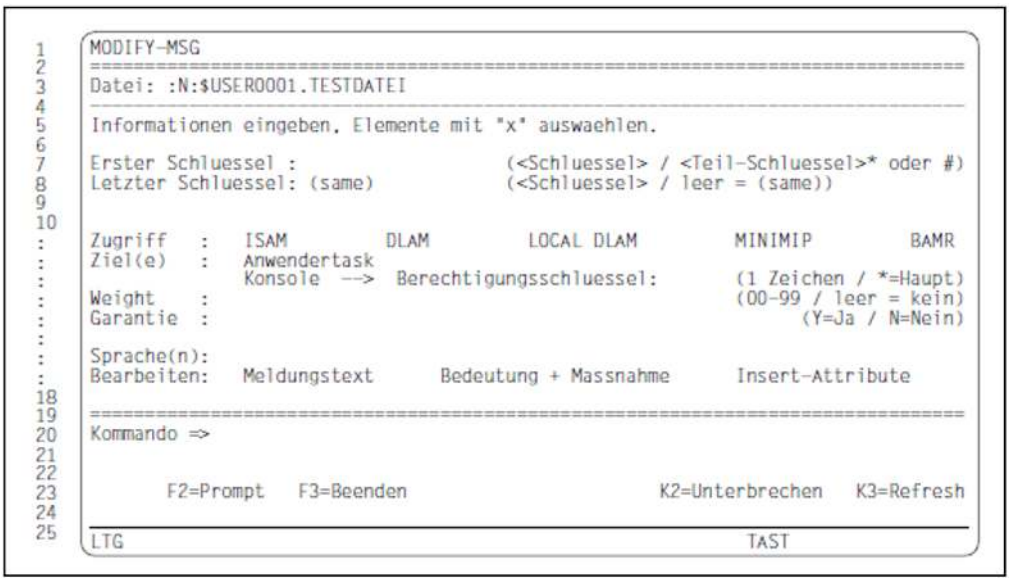
Funktion

Die Maske MODIFY-MSG ist die erste von insgesamt vier Masken, die zum Verändern einer Meldungseinheit in der aktuellen Meldungsdatei aufgerufen werden können.
Nach Eingabe der zu bearbeitenden Meldungsdatei in der Hauptmaske MENU und Auswahl der Funktion 6 (Aendern), wechselt MSGMAKER in die Maske MODIFY-MSG.Der Eintrag eines Meldungsbereichs in die Felder **Erster Schluessel** (eventl. auch **Letzter Schluessel**) und die Bestätigung des Bereichs durch Drücken der Taste DUE bringt die erste Meldungseinheit zur Anzeige.
Erst jetzt können in der Maske MODIFY-MSG die Meldungsattribute der Meldungseinheit geändert werden. Sollen die Texte ebenfalls geändert werden (Einfügen, Anfügen, Löschen, Ersetzen von Textzeilen), so führt das Markieren der folgenden Felder (und abschließendes Drücken von DUE) zu den entsprechenden Masken.

X Meldungstext	Maske MSG-TEXT
X Bedeutung + Massnahme	Maske MEANING / RESPONSE
X Insert-Attribute	Maske INSERT-ATTRIBUTES

Betreffen die Änderungen nur die Meldungsattribute, so werden diese in der Maske MODIFY-MSG mit DUE bestätigt und in die Meldungsdatei übertragen.
Umfangreichere Änderungen innerhalb einer Meldungseinheit (Meldungstext, Inserts, ...) werden erst nach Durchlaufen aller erforderlichen Masken in die Meldungsdatei übertragen.
Soll eine Meldungseinheit innerhalb des ausgewählten Meldungsintervalls nicht geändert werden, so kann der Anwender mit K1 diesen Meldungsschlüssel überspringen. Die Funktionstaste wird in der Maske angeboten, sobald der Anwender mit DUE die Meldungseinheiten in die Meldungdatei eingelesen hat.

Maske



Ausgabefelder

Datei (Name der Meldungsdatei)

Der in die Hauptmaske MENU eingetragene Name wird in die Maske MODIFY-MSG übernommen. Der Name kann nicht verändert werden.

Am rechten Rand des Informationsbereichs wird der ausgewählte Meldungsbereich angezeigt. (*Anm.: In der Maske nicht zu sehen*)

Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Wird in das Feld **Erster Schluessel** ein Meldungsbereich eingetragen und durch Drücken der Taste DUE bestätigt, werden der Reihe nach alle definierten Meldungsschlüssel dieses Meldungsbereichs angezeigt. Die Meldungsschlüssel können nicht verändert werden, der Cursor wird zur Bearbeitung der Meldungseinheit auf das Feld **Zugriff** positioniert.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis sieben Zeichen und der Platzhalter # eine bis vier Ziffern ersetzen kann. Die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse).

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* oder # möglich:
>
> `A*, AB*, ABC*, ABC0*, ABC00*, ABC000*`
>
> `ABC#, ABC0#, ABC00#, ABC000#`

Der Eintrag * wählt alle definierten Meldungsschlüssel aus. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe in das Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Nähere Informationen zum Meldungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

Anzeige: (same)

Zugriff (MIP-Zugriffsmethoden für die Meldungen)

Bezeichnet die verschiedenen Zugriffsmethoden der Systemkomponente MIP auf die Meldungen. Die Auswahl mehrerer Zugriffsmethoden ist möglich, jedoch dürfen die Methoden ISAM und DLAM bzw. ISAM und LOCAL-DLAM nicht kombiniert werden.

Gültigkeitskriterien:
Markieren der Zugriffsmethoden mit dem Zeichen „X“.

Die Auswahl einer Zugriffsmethode kann durch Löschen von „X“ durch ein Leerzeichen oder ein NIL-Zeichen rückgängig gemacht werden. Eine neue Zugriffsmethode muss erneut mit „X“ ausgewählt werden. Sollen keine weiteren Veränderungen bei den Meldungsattributen oder Texten erfolgen, wird die Änderung durch Drücken der Taste DUE bestätigt. Die nächste Meldungseinheit innerhalb des bezeichneten Meldungsintervalls wird angezeigt.

ISAM
Die Meldungen werden über den ISAM-Schlüssel gesucht.

DLAM
Diese Zugriffsmethode ist für besonders häufige Meldungen vorgesehen. Wird eine Meldungsdatei, die eine DLAM-Meldung enthält, aktiviert, wird diese in den Arbeitsspeicher geladen. Die DLAM-Meldung kann von MIP direkt, ohne Zugriff auf die Meldungsdatei, ausgegeben werden.

LOCAL-DLAM / MINIMIP / BAMR

Diese Zugriffsmethoden sind für den internen Gebrauch beim Hersteller reserviert (siehe DHB-Abschnitt MIP).

Ziel(e) (Ausgabeziele der Meldungen)

Diese Angabe dient dem Anwender zur Dokumentation der Ausgabeziele. Der Makro MSG7X und das Kommando /HELP-MSG-INFORMATION werten diese Angabe **nicht** aus.

Gültigkeitskriterien:
Markieren der Ausgabeziele mit dem Zeichen „X“.

Die Einträge können durch Überschreiben der Felder mit Leerzeichen oder NIL-Zeichen gelöscht und mit „X“ neu ausgewählt werden.

Anwendertask
Die Meldung hat das Ausgabeziel SYSOUT, SYSLST oder einen benutzereigenen Speicherbereich.

Konsole
Das Ausgabeziel der Meldung ist eine Konsole. Der Berechtigungsschlüssel wird als Zielangabe ausgewertet.

Berechtigungsschlüssel

Ein Berechtigungsschlüssel muss angegeben werden, wenn im Feld **Ziel(e)** das Ausgabeziel **Konsole** markiert wurde. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Gültigkeitskriterien:
Datentyp: <alphanum-name 1..1>

Weight (Meldungsgewicht)

Ein Meldungsgewicht muss angegeben werden, wenn im Feld **Ziel(e)** das Ausgabeziel **Konsole** markiert wurde. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Gültigkeitskriterien:
Datentyp: <integer 0..99>

Garantie (Garantie der Meldung)

Das Meldungsattribut „Garantie“ wird von MIP ausgewertet.
Die Garantieerklärung besagt, dass bestimmte Teile der Meldung in den kommenden BS2000-Versionen nicht mehr geändert werden.

Folgende Meldungs-Bestandteile werden garantiert:

- Meldungsschlüssel
- Nummerierung und Bedeutung der Inserts

Der Meldungstext wird **nicht** garantiert.

MIP erzeugt für garantierte Meldungen S-Variablen. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Sprache(n) (Kennbuchstabe für die Sprache)

Jede Sprache wird durch einen Kennbuchstaben abgekürzt. Der Buchstabe D steht für Deutsch, E für Englisch. Wird in der Maske MODIFY-MSG ein Kennbuchstabe angezeigt, kann dieser weder durch Überschreiben mit einem anderen Buchstaben geändert noch durch Überschreiben mit einem Leerzeichen gelöscht werden.

Eine Änderung des Kennbuchstabens kann nur in der Maske MSG-TEXT erfolgen (Beschreibung siehe "Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes"). In den Masken MSG-TEXT und DELETE-MSG kann der Kennbuchstabe und somit auch der Meldungstext gelöscht werden.

Gültigkeitskriterien:
Die Buchstaben „A“ bis „Z“.

Bearbeiten (Auswahl der Texte)

Auswahl der Texte, die geändert werden sollen.

Gültigkeitskriterien:
Markieren der Texte mit dem Zeichen „X“.

Meldungstext
Im Anschluss an die Maske MODIFY-MSG wird zur Maske MSG-TEXT verzweigt, in welcher die Meldungstexte in den verschiedenen Sprachen verändert werden können. Beschreibung der Maske MSG-TEXT siehe "Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes".

Bedeutung + Massnahme
Im Anschluss an die Maske MSG-TEXT wird zur Maske MEANING/RESPONSE verzweigt, in welcher die Bedeutungs- und Maßnahmetexte verändert werden können. Beschreibung der Maske MEANING /RESPONSE siehe "Maske MEANING/RESPONSE - Einfügen oder Ändern des Bedeutungs- und Maßnahmetextes".

Insert-Attribute
Im Anschluss an die Masken MSG-TEXT und MEANING/RESPONSE wird zur Maske INSERT-ATTRIBUTES verzweigt, in welcher die Insertattribute verändert werden können. Beschreibung der Maske INSERT-ATTRIBUTES siehe "Maske INSERT-ATTRIBUTES - Einfügen oder Ändern der Insert-Attribute".

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MODIFY-MSG -> ?

DUE Die Funktion MODIFY-MSG wird ausgelöst; Änderungen zu Meldungsattributen werden in der Meldungsdatei abgespeichert; bei Textänderungen verzweigt MSGMAKER zu den ausgewählten Masken.

K1 Wird ein Meldungsintervall in die Maske MODIFY-MSG eingetragen und mit DUE bestätigt, bietet MSGMAKER die Funktion K1 = Ueberspringen an. Der Anwender erhält die Möglichkeit, diesen Meldungsschlüssel zu überspringen. Im Anschluss daran erscheint die Maske MODIFY-MSG mit dem nächsten Meldungsschlüssel.

F3 Die Funktion MODIFY-MSG wird abgebrochen; Änderungen zu vorher editierten Meldungen sind in der Meldungsdatei gespeichert, Änderungen zum aktuellen Meldungsschlüssel gehen verloren. MSGMAKER kehrt zur Ausgangsmaske MENU zurück.

8.2.6.9 Maske MSG-TEXT - Einfügen oder Ändern des Meldungstextes

Reihenfolge der Masken

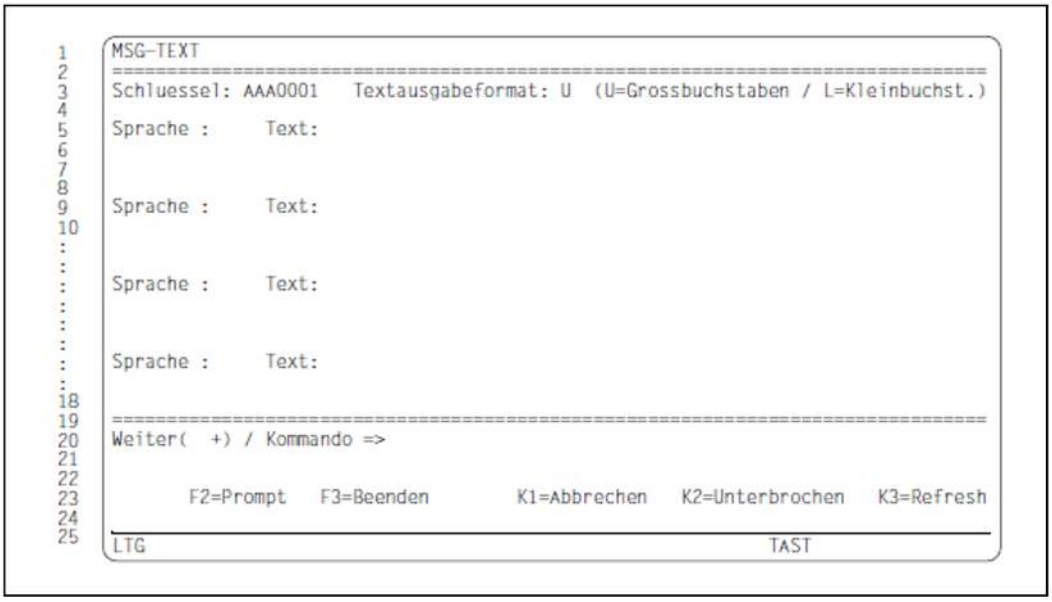
MENU -> 5/6 -> ADD-MSG/MODIFY-MSG -> MSG-TEXT

Funktion

Die Maske MSG-TEXT wird durch Markieren des Feldes „**Bearbeiten**: **X** Meldungstext“ in den Masken ADD-MSG oder MODIFY-MSG aufgerufen. In ihr können Meldungstexte in definierten Sprachen eingegeben, verändert oder gelöscht werden.

Der Meldungsschlüssel der aktuellen Meldungseinheit wird im Feld **Schluessel** angezeigt, kann aber nicht verändert werden. Sind mehrere Sprachen definiert worden, können die zugehörigen Meldungstexte nacheinander in die Maske eingetragen werden. Die Masken MEANING/RESPONSE und INSERT-ATTRIBUTES werden im Anschluss daran aufgerufen, wenn sie in den Masken ADD-MSG oder MODIFY-MSG ebenfalls ausgewählt wurden. Es ist ferner möglich, den Meldungstext in nur einer Sprache in die Maske MSG-TEXT einzugeben und anschließend sofort in die beiden anderen Masken zu verzweigen. In deren Anschluss bringt MSGMAKER automatisch die Maske MSG-TEXT wieder zur Anzeige, in welche die Meldungstexte in den noch fehlenden Sprachen ergänzt werden können.
Die ergänzten oder veränderten Meldungseinheiten werden erst dann in der Meldungsdatei abgespeichert, wenn alle notwendigen Eingaben in den Masken MSG-TEXT, MEANING/RESPONSE oder INSERT-ATTRIBUTES vorhanden sind und die letzte, hierzu notwendige Maske mit DUE bestätigt worden ist.

Maske



Ausgabefelder

Schluessel (Meldungsschlüssel der zu bearbeitenden Meldungseinheit)

Der in den Masken ADD-MSG oder MODIFY-MSG eingetragene Meldungsschlüssel wird in die Maske MSG-TEXT übertragen und kann **nicht** verändert werden.

Eingabefelder

Textausgabeformat (Ausgabeformat des Meldungtextes)

Bezeichnet die Schreibweise des Meldungstextes, der durch den Makro MSG7X oder durch das Kommando /HELP-MSG-INFORMATION ausgegeben wird. Die Default-Texte für Inserts sowie die Texte, die über den Makro MSG7X vorgegeben sind, werden ebenfalls an dieses Format angepaßt.

Gültigkeitskriterien:
Die Buchstaben „U“ oder „L“.
„U“ steht für Upper-case (Ausgabe in Großbuchstaben) und „L“ für Lower-case (Ausgabe des Meldungstextes, so wie dieser in die Maske MSG-TEXT eingetragen wurde).

> ¡ Die Schreibweise des Meldungstextes in der Maske MSG-TEXT entspricht der abgespeicherten Form des Meldungstextes in der Meldungsdatei.

Voreinstellung/Anzeige: U

Sprache (Kennbuchstabe für die Sprache)

Ausgangsmaske ADD-MSG:
Sind Kennbuchstaben in das Feld **Sprache(n)** der Maske ADD-MSG eingetragen, werden diese in die Maske MSG-TEXT übernommen. Andernfalls können die Kennbuchstaben direkt in die Maske MSG-TEXT eingetragen werden.

Ausgangsmaske MODIFY-MSG:
Alle in der Maske MODIFY-MSG angezeigten Sprachen werden in die Maske MSG-TEXT übertragen. Es bestehen nun folgende Möglichkeiten:

- Der Kennbuchstabe bleibt unverändert
  Der Meldungstext kann jedoch verändert werden.
- Der Kennbuchstabe wird durch Überschreiben neu definiert
  Der zugehörige Meldungstext wird folglich unter einer anderen Sprache abgespeichert.
- Der Kennbuchstabe wird mit einem Leerzeichen gelöscht.
  Der Meldungstext muss ebenfalls gelöscht werden. War nur eine Sprache in der Meldungseinheit definiert, fordert MSGMAKER zur erneuten Eingabe einer Sprache (und eines Meldungstextes) auf.

Text (Meldungstext)

Der Meldungstext wird intern nicht dreizeilig (gemäß Bildschirmanzeige), sondern einzeilig abgespeichert. Eine mehrzeilige Ausgabe durch den Makro MSG7X oder durch das Kommando /HELP-MSG-INFORMATION kann mit Hilfe des Trennungszeichens „^“ erreicht werden. NIL-Zeichen innerhalb des Textes werden durch Leerzeichen ersetzt. Nähere Informationen zum Aufbau des Meldungstextes finden Sie im Handbuch „Systembetreuung“ [5].

Gültigkeitskriterien:
Maximal 220 Zeichen reiner Meldungstext, inklusive der nichtersetzten Zeichenfolgen (&00) ... (&29).

Weiter (+ -) Anzeige einer weiteren Maske.

+ Sind mehr als vier Sprachen in der Meldungseinheit gespeichert oder soll eine fünfte Sprache hinzugefügt werden, wird durch Eingabe von „+“ eine zweite Maske MSG-TEXT aufgerufen, in der weitere Texte angezeigt oder eingegeben werden können.

- Mit „-“ wird zur ersten Maske von MSG-TEXT zurückgeblättert.

Gültigkeitskriterien:
Die Zeichen „-“ oder „+“.

Kommando

Weitere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MSG-TEXT -> ?

DUE	Die Eingaben in die Maske MSG-TEXT werden bestätigt und entsprechend der vorher getroffenen Auswahl verzweigt das Programm zu den Masken MEANING/RESPONSE, INSERT-ATTRIBUTES oder kehrt zur Ausgangsmaske ADD-MSG bzw. MODIFY-MSG zurück.
K1	Rückkehr in die Ausgangsmaske ADD-MSG bzw. MODIFY-MSG. Änderungen, die in der Maske MSG-TEXT mit DUE bestätigt wurden, sind gespeichert und weiterverwendbar. Bei Rückkehr in die Maske MSG-TEXT mit DUE, werden die vorher geänderten Daten angezeigt.K1 wirkt in dieser Maske wie K3 = Refresh.
F3	Die Bearbeitung der aktuellen Meldungstexte wird abgebrochen. Änderungen zu vorher editierten Meldungen sind in der Meldungsdatei gespeichert, Änderungen zum aktuellen Meldungsschlüssel gehen verloren. MSGMAKER kehrt zur Ausgangsmaske ADD-MSG bzw. MODIFY-MSG zurück.

8.2.6.10 Maske MEANING/RESPONSE - Einfügen oder Ändern des Bedeutungs- und Maßnahmetextes

Reihenfolge der Masken

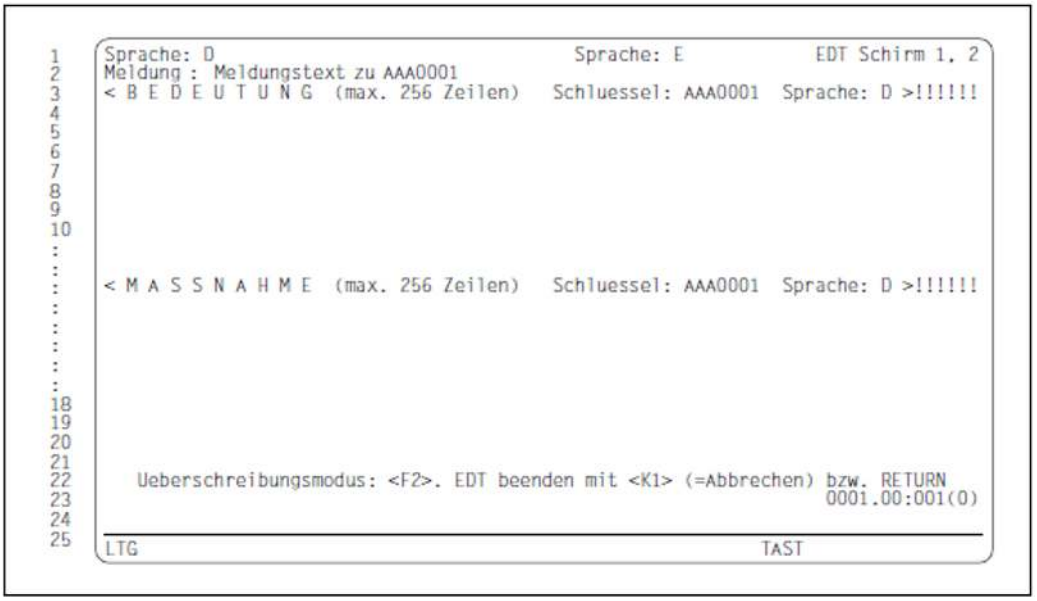
MENU -> 5/6 -> ADD-MSG/MODIFY-MSG -> MEANING/RESPONSE

Funktion

Durch Markieren des Feldes „**Bearbeiten**: **X** Bedeutung + Massnahme“ in den Masken ADD-MSG oder MODIFY-MSG wird in das Unterprogramm EDT verzweigt und ein modifizierter EDT-Arbeitsbildschirm als Maske MEANING /RESPONSE aufgerufen. Ausführliche Informationen zum EDT siehe Handbuch „EDT“ [14].
In einer oder mehreren Masken können Bedeutungs- und Maßnahmetexte eingefügt, verändert oder gelöscht werden.
Zur Erstellung des Bedeutungs- und Maßnahmetextes können alle Kurzanweisungen des EDT verwendet werden, welche in die Markierungsspalte (erste Spalte) des Bildschirms einzugeben sind. Ebenso können EDT-Anweisungen in die letzte Zeile (Anweisungszeile) des Bildschirms eingetragen werden.
Der Bildschirm ist in zwei Bereiche unterteilt; in den oberen Bereich wird der Bedeutungstext, in den unteren der Maßnahmetext eingetragen. Enthält der Bedeutungstext mehr als neun Zeilen, muss der obere Bereich durch Einfügen neuer Zeilen erweitert werden, da jeder Text im unteren Bereich automatisch als Maßnahmetext interpretiert wird.
Die standardmäßig auf dem Bildschirm erscheinenden Informationszeilen können nicht überschrieben werden. Diese Zeilen sind nicht Bestandteil des Bedeutungs- oder Maßnahmetextes.

Die Bedeutungs- und Maßnahmetexte zu allen weiteren, definierten Sprachen können in den EDT-Bildschirmen mit den Nummern 0 bis 7 eingegeben und mit DUE, F1 oder F2 bestätigt werden. Alle EDT-Bildschirme, die nicht zur Texteingabe verwendet werden, stehen als Arbeitsbildschirme zur Verfügung. Werden Texte zu acht Sprachen definiert, können die EDT-Bildschirme 8 und 9 als Arbeitsbildschirme benützt werden.
Die Anweisung @RET[URN] oder RETURN, die in die Anweisungszeile eingetragen wird, beendet das Programm EDT. Alle bearbeiteten Sprachen (= EDT-Bildschirme) werden gemeinsam an das Programm MSGMAKER übergeben. Erst wenn alle, für die Bearbeitung der Meldungseinheit notwendigen Masken durchlaufen sind, werden die erfaßten Bedeutungs- und Maßnahmetexte abgespeichert.

Maske



Ausgabefelder

Sprache	(Kennbuchstabe der aktuellen Sprache)
Sprache	(Kennbuchstaben aller weiteren, definierten Sprachen)
EDT Schirm	(EDT-Bildschirme, in denen die Texte zu allen weiteren Kennbuchstaben eingegeben werden können)
Meldung	(aktueller Meldungstext, zu dem unter **Sprache** eingetragenen Kennbuchstaben)

Eingabefelder

> ¡ Um Text eingeben oder verändern zu können, müssen die angezeigten Zeilen durch die Funktionstaste F2 überschreibbar gesetzt werden. Einzelne Zeilen können durch ein „x“ in der Markierungsspalte ausgewählt und bearbeitet werden.

Das Trennungszeichen „^“

Wird dieses Zeichen innerhalb des Bedeutungs- oder Maßnahmetextes verwendet, wird Text, der auf dieses Zeichen folgt, nach Betätigen der Tasten DUE, F1 und F2 in eine neue Zeile geschrieben. Das Trennungszeichen ist nur im EDT gültig und ist **nicht** Bestandteil des Bedeutungs- und Maßnahmetextes. Es unterscheidet sich folglich vom Trennungszeichen „^“, das im Meldungstext verwendet werden kann.

B E D E U T U N G (max. 256 Zeilen) (Bedeutungstext)

Gültigkeitskriterien:
Der Bedeutungstext darf maximal 256 Zeilen mit je 74 Zeichen lang sein. Zeichen, die in die Spalten 75 bis 80 eingetragen werden, werden nach Drücken der Taste DUE automatisch abgeschnitten und es wird die Meldung EDT2267 ausgegeben. Zeilen, die aus NIL-Zeichen oder Leerzeichen bestehen, werden unterdrückt. NIL-Zeichen innerhalb des Textes werden durch Leerzeichen ersetzt. Soll der Bedeutungstext länger als neun Zeilen sein, muss der obere Bildschirmbereich durch Einfügen neuer Zeilen erweitert werden.

Anzeige: **Schluessel** (aktueller Meldungsschlüssel)

Sprache (Kennbuchstabe der aktuellen Sprache)

Der Meldungsschlüssel der aktuellen Meldungseinheit und der aktuelle Kennbuchstabe (entspricht der Anzeige im Feld **Sprache**) werden angezeigt.

M A S S N A H M E (max. 256 Zeilen) (Maßnahmetext)

Gültigkeitskriterien:
Der Maßnahmetext darf maximal 256 Zeilen mit je 74 Zeichen lang sein. Zeichen, die in die Spalten 75 bis 80 eingetragen werden, werden nach Drücken der Taste DUE automatisch abgeschnitten und es wird die Meldung EDT2267 ausgegeben. Zeilen, die aus NIL-Zeichen oder Leerzeichen bestehen, werden unterdrückt. NIL-Zeichen innerhalb des Textes werden durch Leerzeichen ersetzt.

Anzeige: **Schluessel** (aktueller Meldungsschlüssel)

Sprache (Kennbuchstabe der aktuellen Sprache)

Der Meldungsschlüssel der aktuellen Meldungseinheit und der aktuelle Kennbuchstabe (entspricht der Anzeige im Feld **Sprache**) werden angezeigt.

Kommandobereich

Funktionstasten

Die Tasten F1, F2 und F3 übernehmen im EDT andere Funktionen als im Programm MSGMAKER.

F1 entspricht der Taste DUE.

F2 Wird der Bildschirm mit F2 abgeschickt, werden alle Zeilen auf überschreibbar gestellt, d.h. im oberen und unteren Bildschirmbereich kann Text eingegeben oder verändert werden. Informationszeilen sind vor Überschreiben geschützt.

F3 Wird mit der Anweisung „@ON .... FIND“ eine Suchoperation gestartet, kann mit der Taste F3 zur nächsten oder vorangegangenen Suchzeichenfolge gesprungen werden.

Weitere Informationen zu den Tasten K2, K3 siehe "Allgemeiner Maskenaufbau".

Nähere Informationen zu den Editier-Funktionen siehe Handbuch „EDT“ [14].

Anschließende Operationen:

MEANING/RESPONSE	-> ?

@RET[URN]	Die Eingaben werden bestätigt und entsprechend der vorher getroffenen Auswahl verzweigt das Programm zu den Masken MSG-TEXT, INSERT-ATTRIBUTES oder kehrt zur Ausgangsmaske ADD-MSG bzw. MODIFY-MSG zurück.
K1	bricht die Bildschirmeingabe im EDT ab. Alle Veränderungen (neuer oder geänderter Text) werden nur dann zum Programm MSGMAKER weitergeleitet, wenn die Eingaben durch Drücken der Tasten DUE, F1, F2 oder F3 vorher bestätigt worden sind. Nach Drücken von K1 wird die zuletzt angezeigte Maske im Programm MSGMAKER wieder aufgerufen.

Alle weiteren Operationen siehe oben bei „Funktionstasten“.

8.2.6.11 Maske INSERT-ATTRIBUTES - Einfügen oder Ändern der Insert-Attribute

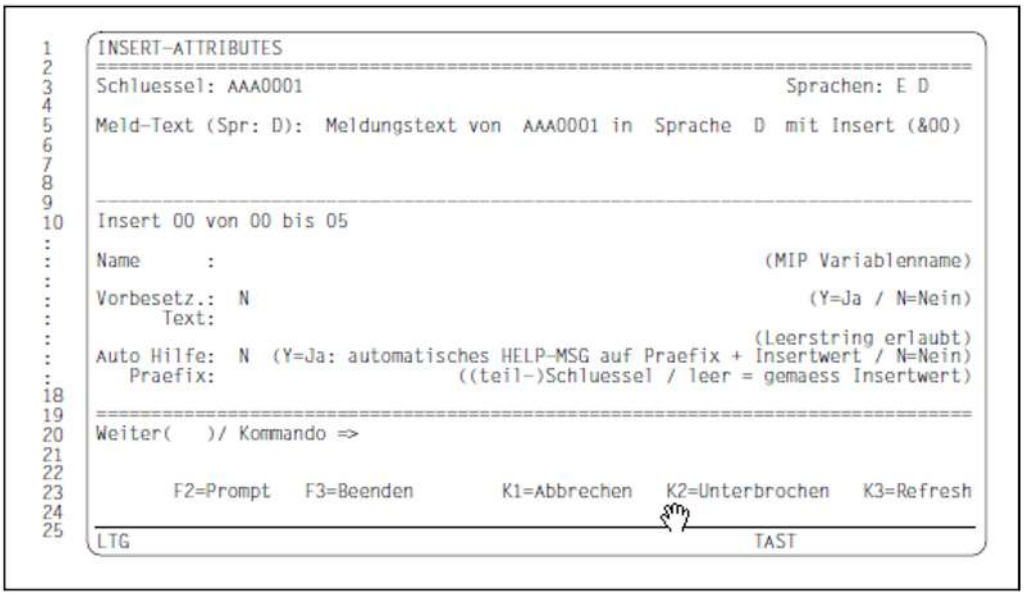
Reihenfolge der Masken

MENU -> 5/6 -> ADD-MSG/MODIFY-MSG -> INSERT-ATTRIBUTES

Funktion

Die Maske INSERT-ATTRIBUTES wird durch Markieren des Feldes „**Bearbeiten**: **X** Insert-Attributes“ in den Masken ADD-MSG oder MODIFY-MSG aufgerufen. In ihr können Namen und Default-Texte, der in den Meldungstexten definierten Inserts eingefügt, verändert oder gelöscht werden. Die Änderungen werden mit DUE bestätigt.
Die Inserts werden in der Meldungseinheit sprachübergreifend definiert.

Maske



Ausgabefelder

Schluessel (Meldungsschlüssel der aktuellen Meldungseinheit)

Sprache(n) (Kennbuchstaben aller definierten Sprachen der Meldungseinheit)

Meld-Text (In der „Sprachreihenfolge“ erster Meldungstext, in welchem Inserts definiert sind)

Als Referenztext wird der Meldungstext angezeigt, in welchem ein oder mehrere Inserts definiert worden sind. Der Meldungstext kann im Gegensatz zum Kennbuchstaben nicht verändert werden. Durch Überschreiben des Buchstaben im Feld **Spr** kann eine neue Sprache für den angezeigten Meldungstext definiert werden.

Eingabefelder

Insert .. von .. bis .. (Insertnummer)

Ändern, Löschen oder Neuaufnehmen von Insertnummern kann nur im Meldungstext selbst erfolgen (siehe Maske MODIFY-MSG, "Maske MODIFY-MSG - Meldungseinheit ändern").

Der Geltungsbereich xx bis yy gibt die niedrigste und die höchste Insertnummer an, die im Meldungstext definiert wurde. Durch Eingabe einer neuen Insertsnummer, wird ein weiteres Insert ausgegeben. Sind einzelne Insertnummern dieses Geltungsbereichs nicht definiert, werden sie nicht ausgegeben.

Name (Insertname)

Namen der im Meldungstext definierten Inserts. Insertnamen werden von MIP ausgewertet und zur Erzeugung von S-Variablen verwendet. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

Vorbesetz. (Default-Text)

Voreingestellter Text (Default-Text), der anstelle des Inserts in den Meldungstext eingefügt wird, falls im Makro MSG7X kein aktueller Text dafür angegeben ist.

Vorbesetzung: Y	Ein Inserttext ist voreingestellt
Vorbesetzung: N	Kein Inserttext voreingestellt

Liegt die ADD-MSG-Funktion zugrunde, kann angegeben werden, ob ein Default-Text angezeigt werden soll oder eine leere Zeichenfolge ausgegeben wird. Zur Eingabe einer leeren Zeichenfolge vergleiche Operandenberschreibung DEFAULT-VALUE = *EMPTY-STRING, "ADD-MSG - Einfügen einer Meldungseinheit". Liegt die MODIFY-Funktion zugrunde, werden bereits definierte Default-Texte angezeigt.

Gültigkeitskriterien:
Datentyp: <c-string 1..54 with-low>

Der Default-Text in der Maske INSERT-ATTRIBUTES wird nicht in Hochkommata eingeschlossen. Hochkommata innerhalb des Default-Textes müssen nicht verdoppelt werden. Der im Eingabefeld eingegebene Text ist der Eingabetext. Der Eingabetext kann auch ein Leerstring oder Leerzeichen sein. Der voreingestellte Text kann durch Überschreiben mit NIL-Zeichen gelöscht werden, nicht mit Leezeichen.

Die Ausgabe des Default-Textes hängt vom Operanden MSG-TEXT-OUTPUT ab.

Auto Hilfe (Automatische Hilfsfunktion)

Wird über ein Insert ein Meldungsschlüssel oder ein Teil davon ausgegeben (z.B. bei DVS-Fehlern wird lediglich die Meldungsnummer ausgegeben), so wird über die automatische Hilfsfunktion MIP dazu veranlaßt, zusätzlich zu diesem Meldungsschlüssel den zugehörigen Meldungstext auszugeben, siehe auch "ADD-MSG - Einfügen einer Meldungseinheit".

Auto Hilfe : Y	Zum Meldungsschlüssel wird der Meldungstext ausgegeben
Praefix :	Meldungsklasse, Meldungsnummer usw., vergleiche Operand PREFIX, "ADD-MSG - Einfügen einer Meldungseinheit"
Auto Hilfe: N	Es wird nur der Meldungsschlüssel ausgegeben. Das Feld Prefix darf nur NIL- oder Leerzeichen enthalten

Durch Eingabe von „+“ bzw. „-“ wird zur nächsten bzw. vorhergehenden Insertnummer geblättert, die für diese Meldung definiert wurde.

Die Eingabe einer neuen Insertnummer hat Vorrang vor der Funktion „Blättern“.

Die Eingabe neuer Insertattribute oder das Modifizieren von Insertattributen hat Vorrang vor der Funktion „Blättern“ oder dem Ändern der Insertnummer.

Kommando

Weitere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

INSERT-ATTRIBUTES	-> ?

DUE	Die Eingaben werden bestätigt und entsprechend der vorher getroffenen Auswahl verzweigt das Programm zu den Masken MSG-TEXT, MEANING/RESPONSE oder kehrt zur Ausgangsmaske ADD-MSG bzw. MODIFY-MSG zurück.
F3	Die Funktion wird abgebrochen. Änderungen zu vorher editierten Meldungen sind in der Meldungsdatei gespeichert, Änderungen zur aktuellen Meldung (Meldungs-/Bedeutungs-/Maßnahmetext) gehen verloren. Das Programm kehrt zur Ausgangsmaske ADD-MSG bzw. MODIFY-MSG zurück, auch wenn noch weitere Meldungseinheiten zur Bearbeitung anstehen.
K1	Rückkehr zu den Masken MSG-TEXT oder MEANING/RESPONSE. Eintragungen in die Maske INSERT-ATTRIBUTES gehen verloren. Bei Rückkehr in die Maske INSERT-ATTRIBUTES müssen erneut Werte eingeben werden.

8.2.6.12 Maske DELETE-MSG - Meldungseinheit löschen

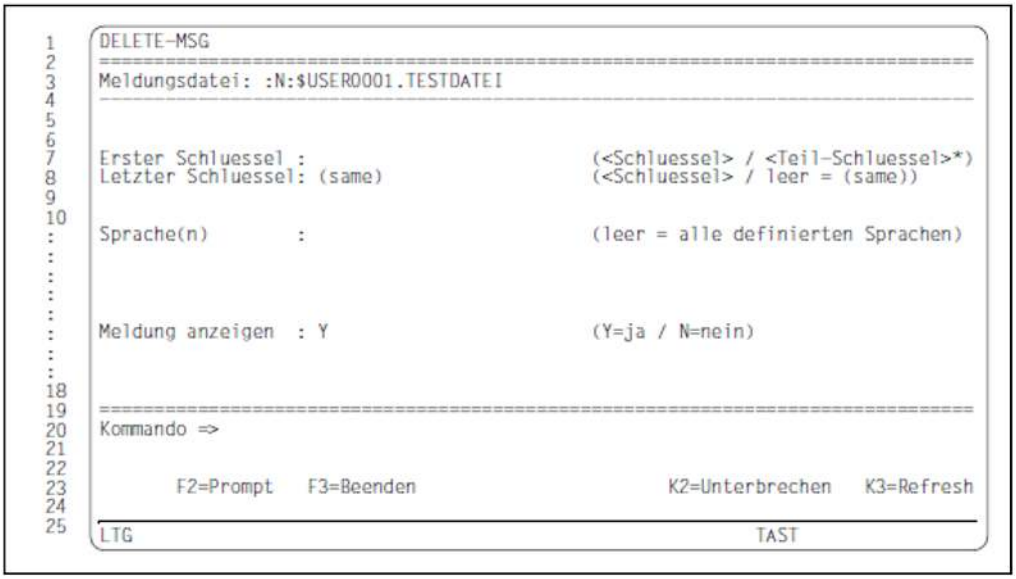
Reihenfolge der Masken

MENU -> 7 -> DELETE-MSG

Funktion

In dieser Maske können zu löschende Meldungseinheiten durch Angabe des Meldungsschlüssels und des Kennbuchstabens ausgewählt werden.
Eine zweite Maske gleichen Namens wird aufgerufen, wenn die Meldungsattribute und Meldungstexte der ausgewählten Meldungseinheiten vor dem Löschen noch einmal angezeigt werden sollen.

Maske 1



Ausgabefelder

Meldungsdatei (aktuelle Meldungsdatei)

Die aktuelle Meldungsdatei wird angezeigt. Sie wurde bereits in der Maske MENU eröffnet. Der Dateiname kann in der Maske DELETE-MSG nicht verändert werden.

Eingabefelder

Erster Schluessel (Erster Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet eine Meldungseinheit oder die erste Meldungseinheit eines Meldungsbereichs.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel oder teilweise definierter Meldungsschlüssel, in welchem der Platzhalter * ein bis vier Zeichen ersetzen kann.
Die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse).

Beispiel

> Folgende Angaben sind für <Teil-Schluessel>* möglich:
> ABC*, ABC0*, ABC00*, ABC000*

Letzter Schluessel (Letzter Meldungsschlüssel des Meldungsintervalls)

Der angegebene Meldungsschlüssel bezeichnet die letzte Meldungseinheit eines Meldungsbereichs. Der Meldungsschlüssel muss derselben Meldungsklasse wie der erste Meldungsschlüssel angehören. Es wird genau eine Meldungseinheit ausgewählt, wenn der Meldungsschlüssel mit der Eingabe im Feld **Erster Schluessel** übereinstimmt oder das Schlüsselwort (same) übernommen wird.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel; ein leeres Feld bedeutet, dass der Meldungsschlüssel des Feldes **Erster Schluessel** übernommen wird. Weitere Informationen zum Meldungsschlüssel siehe "Beschreibung häufig auftretender Felder".

Anzeige: (same)

Sprache(n) (Kennbuchstabe für die Sprache)

Die einzugebenden Kennbuchstaben orientieren sich nach den Kürzeln, welche beim Erzeugen der Meldungseinheit vereinbart wurden.
Wird im Feld **Sprache(n)** keine Sprache eingetragen, werden die Texte in allen Sprachen sowie die Attribute der Meldungseinheit gelöscht.

Sind alle Meldungseinheiten einer Meldungsdatei auf diese Weise entfernt worden, ist die Meldungsdatei aus diesem Grunde nicht gelöscht, da die Attribute der Meldungsdatei und Zusatzinformationen (z.B. Dokumentationszeilen) erhalten bleiben.

Gültigkeitskriterien:
Die Buchstaben „A“ bis „Z“.

Meldung anzeigen (Anzeige der Meldungseinheit)

Durch Eingabe von „Y“ (ja) im Feld **Meldung anzeigen** wird die zweite Maske DELETE-MSG aufgerufen, in welcher die Meldungsattribute und die Meldungstexte der zu löschenden Meldungseinheit angezeigt werden.
Durch Eingabe von „N“ (nein) werden alle bezeichneten Meldungseinheiten oder Meldungstexte einer Meldungseinheit sofort, ohne nochmals angezeigt zu werden, gelöscht.

Gültigkeitskriterien:
Die Buchstaben „N“ und „Y“.

Voreinstellung/Anzeige: Y

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

DELETE-MSG 1	-> ?

DUE Die Eingaben werden bestätigt und

- bei **Meldung anzeigen**: N wird die Meldungseinheit gelöscht
- bei **Meldung anzeigen**: Y wird zur zweiten Maske von DELETE-MSG verzweigt.

F3 Die Funktion DELETE-MSG wird abgebrochen.

Maske 2

Diese Maske wird aufgerufen, wenn in der vorausgehenden Maske DELETE-MSG „**Meldung anzeigen**: Y“ eingegeben wurde.
Im Informationsbereich der Maske werden der Meldungsschlüssel und die Kennbuchstaben der ausgewählten Sprachen angezeigt.
Im Arbeitsbereich der Maske sind die Meldungsattribute und maximal die ersten drei Meldungstexte in den ausgewählten Sprachen dargestellt.

Ausgabefelder

Im Informationsbereich

Schluessel (Meldungsschlüssel der aktuellen Meldungseinheit)

Sprache (Kennbuchstaben für alle, in der vorausgegangenen Maske ausgewählten Sprachen)

Im Arbeitsbereich

Sprache (Kennbuchstabe für die Sprache)

Text (Meldungstext, in der jeweilige Sprache)

Kommando

Weitere Informationen zur Belegung der Funktionstasten K2 und K3 siehe "Allgemeiner Maskenaufbau".

Anschließende Operationen:

DELETE-MSG 2 -> ?

Um mit der Maskenbearbeitung fortzufahren, werden in der letzten Zeile des Arbeitsbereichs drei Möglichkeiten angeboten:

Folgende Aktionen können ausgeführt werden:

DUE	Die angezeigte Meldungseinheit (Meldungstexte in allen ausgewählten Sprachen) wird sofort gelöscht. Ist ein Meldungsintervall definiert worden, wird anschließend die nächste Meldungseinheit angezeigt.
K1	Die angezeigte Meldungseinheit wird nicht gelöscht. Ist ein Meldungsintervall definiert worden, wird die nächste Meldungseinheit angezeigt.
F3	Die DELETE-Funktion wird beendet. Die angezeigte Meldungseinheit und alle weiteren, im Meldungsintervall angegebenen Meldungseinheiten werden nicht gelöscht. Rückkehr zur Maske DELETE-MSG 1.

8.2.6.13 Maske ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen

Reihenfolge der Masken

MENU -> 9 -> ADD-DOCUMENTATION

Funktion

In dieser Maske können Meldungseinheiten dokumentiert werden. Die Zuordnung der Dokumentation zu den Meldungseinheiten erfolgt über den Meldungsschlüssel
(z.B. AAA0001) oder über ein Meldungsintervall (z.B. AAA0002-AAA0015).

Folgende Angaben sind möglich:

- Meldungsschlüssel/Meldungsintervall
- Name des Meldungsverantwortlichen (20 Zeichen)
- Name des verantwortlichen Teams (15 Zeichen)
- Kommentar (60 Zeichen)

Die Meldungsschlüssel können in beliebiger Reihenfolge in die Maske eingetragen werden. Werden die Eingaben mit DUE bestätigt, so erfolgt eine alphabetische Sortierung der Meldungsschlüssel und die Dokumentationszeilen werden in der Meldungsdatei abgespeichert.

Im Unterschied zu MODIFY-DOCUMENTATION müssen in dieser Maske die Dokumentationszeilen an bereits vorhandene Dokumentation angefügt werden. Letztere ist vor Überschreiben geschützt.

> ¡ Die angegebenen Meldungsschlüssel und der Inhalt der Dokumentationszeilen werden nicht auf ihre Korrektheit hin überprüft.

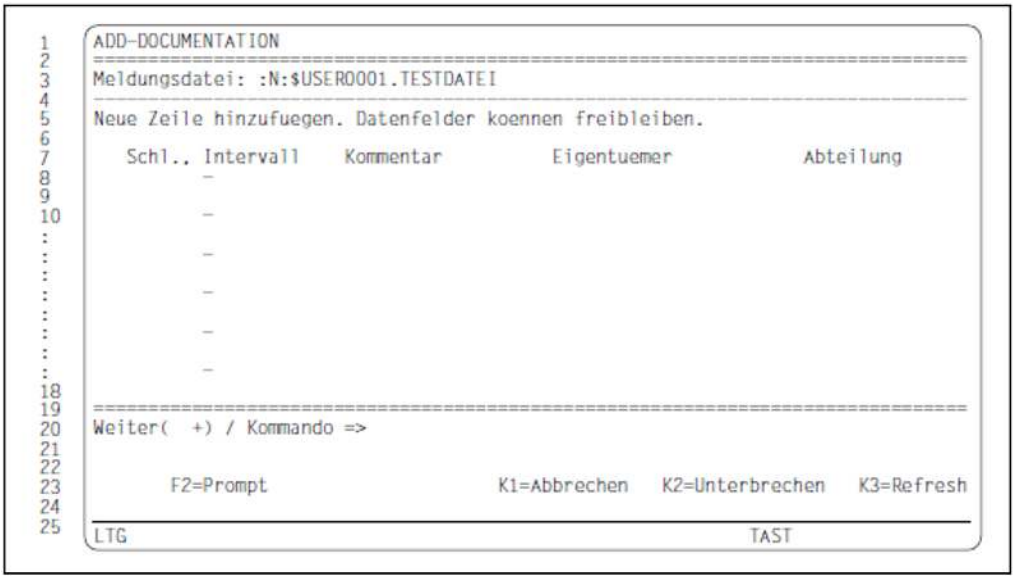
Bei Eingabe eines bereits dokumentierten Meldungsintervalls, gibt MSGMAKER folgende Meldung aus:

```
MSMQL00 DOCUMENTATION LINE '<message range>' ALREADY EXISTS
(COMMENT='<comment>'). OVERWRITE? REPLY (Y=YES; N=NO)? N
```

Wird die Frage mit N beantwortet, wird die Dokumentationszeile noch einmal angezeigt und kann verändert werden.

Mit der COPY- oder MOVE-Funktion können die Dokumentationszeilen innerhalb einer Meldungsdatei oder zwischen zwei Dateien übertragen werden.
Die Ausgabe auf SYSOUT oder SYSLST ist mit der SHOW-Funktion möglich.

Maske



Ausgabefelder

Meldungsdatei

Die aktuelle Meldungsdatei wird angezeigt. Sie wurde bereits in der Maske MENU eröffnet. Der Dateiname kann in der Maske ADD-DOCUMENTATION nicht verändert werden.

Wurden bereits Dokumentationszeilen zu dieser Meldungsdatei verfasst, wird das alphabetisch letzte Meldungsintervall in der ersten Zeile des Dokumentationsbereichs angezeigt. Der Cursor steht am Anfang der nächsten Zeile.
Mit „-“ (in der Kommandozeile) kann auf die bereits dokumentierten Meldungsintervalle zurückgeblättert werden.

Eingabefelder

Schl., Intervall (Meldungsschlüssel/Meldungsintervall)

In dieses Feld wird ein Meldungsschlüssel oder ein Meldungsintervall eingetragen, welcher/welches in den folgenden Feldern dokumentiert wird. Bei Bestätigung der Eingabe mit DUE wird ein einzelner Meldungschlüssel als Bereichsobergrenze wiederholt
(z.B. AAA0001-AAA0001).

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel, die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse). Die Grenzen eines Meldungsintervalls müssen der gleichen Meldungsklasse angehören.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Kommentar (Kommentar zu den Meldungen)

Gültigkeitskriterien:
Maximal 60 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Eigentuemer (Name des Meldungsverantwortlichen)

Gültigkeitskriterien:
Maximal 20 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Abteilung (Bezeichnung des verantwortlichen Teams)

Gültigkeitskriterien:
Maximal 15 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

ADD-DOCUMENTATION	-> ?

DUE	Die Eingaben werden bestätigt und das Programm kehrt zur Maske MENU zurück.
K1	Die Funktion ADD-DOCUMENTATION wird abgebrochen. Die Eingaben gehen verloren, auch wenn vorher mit „+“ und anschließendem DUE zu einer weiteren ADD-DOCUMENTATION-Maske geblättert wurde.

8.2.6.14 Maske MODIFY-DOCUMENTATION - Ändern, Hinzufügen und Löschen von Dokumentationszeilen

Reihenfolge der Masken

MENU	-> 10 ->	MODIFY-DOCUMENTATION

Funktion

In dieser Maske können Dokumentationszeilen, die für eine oder mehrere Meldungseinheiten verfasst wurden, geändert oder gelöscht werden. Ebenso können neue Zeilen hinzugefügt werden. Im Unterschied zu ADD-DOCUMENTATION hat der Anwender auf alle Dokumentationszeilen Zugriff.

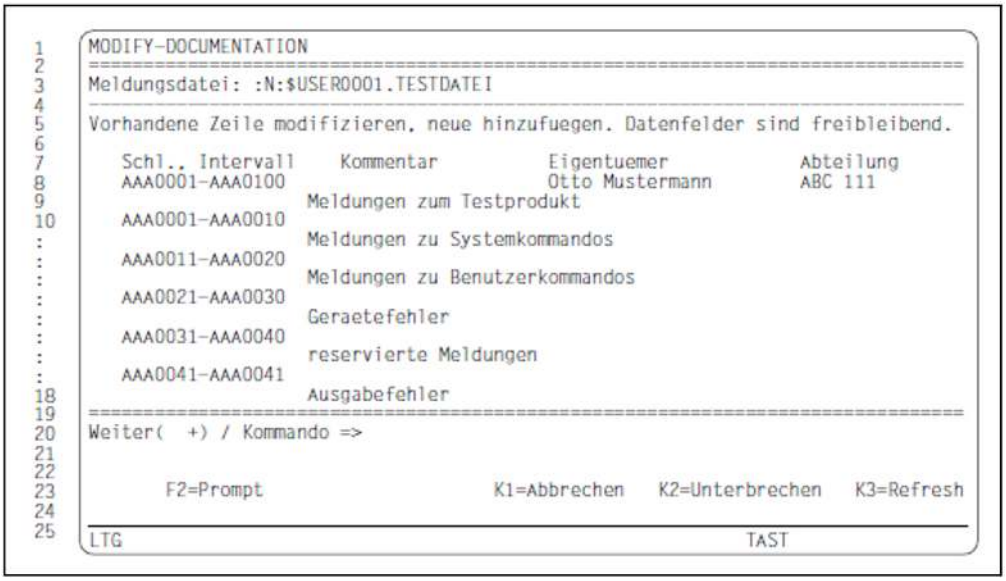
Die Dokumentationszeilen werden in alphabetischer Reihenfolge angezeigt. Jede Zeile kann wie folgt verändert werden:

- Ändern durch Überschreiben
- Löschen durch Leerzeichen oder NIL-Zeichen

Änderungen im Schlüsselfeld werden erst bei endgültiger Quittierung der Eingaben mit DUE in die Meldungsdatei übernommen. Die Meldungsschlüssel bzw. Meldungsintervalle werden dann neu in die Liste einsortiert.

DUE in Verbindung mit „+“ oder „-“ blättert in der Maske vor- oder rückwärts. Änderungen in den Dokumentationszeilen werden hierbei nicht abgespeichert.

Maske



Ausgabefelder

Meldungsdatei

Die aktuelle Meldungsdatei wird angezeigt. Sie wurde bereits in der Maske MENU eröffnet. Der Dateiname kann in der Maske MODIFY-DOCUMENTATION nicht verändert werden.

Eingabefelder

Schl., Intervall (Meldungsschlüssel/Meldungsintervall)

In diesem Feld kann ein Meldungsschlüssel geändert, gelöscht oder eingefügt werden.Wird das Schlüsselfeld gelöscht, ohne dass die Dokumentationszeile ebenfalls gelöscht wird, gibt MSGMAKER die Fehlermeldung `MSMIF01` aus.

Erst wenn die Dokumentationszeile vollständig gelöscht ist, erfolgt keine Fehlermeldung mehr. Ein Meldungsintervall ohne Dokumentation erzeugt keine Fehlermeldung.

Gültigkeitskriterien:
Siebenstelliger Meldungsschlüssel, die ersten drei Zeichen müssen Buchstaben sein (Meldungsklasse). Die Grenzen eines Meldungsintervalls müssen der gleichen Meldungsklasse angehören.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Kommentar (Kommentar zu den Meldungen)

Gültigkeitskriterien:
Maximal 60 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Eigentuemer (Name des Meldungsverantwortlichen)

Gültigkeitskriterien:
Maximal 20 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Abteilung (Bezeichnung des verantwortlichen Teams)

Gültigkeitskriterien:
Maximal 15 Zeichen, auch Leerzeichen.

Zum nächsten Feld kann mit dem Tabulator gesprungen werden.

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

MODIFY-DOCUMENTATION	-> ?

DUE	Die Eingaben werden bestätigt und das Programm kehrt zur Maske MENU zurück.
K1	Die Funktion MODIFY-DOCUMENTATION wird abgebrochen. Die Eingaben gehen verloren, auch wenn vorher mit „+“/“-“ und anschließendem DUE zu einer weiteren MODIFY-DOCUMENTATION-Maske geblättert wurde.

8.2.6.15 Maske DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen

Reihenfolge der Masken

MENU -> 11 -> DELETE-DOCUMENTATION

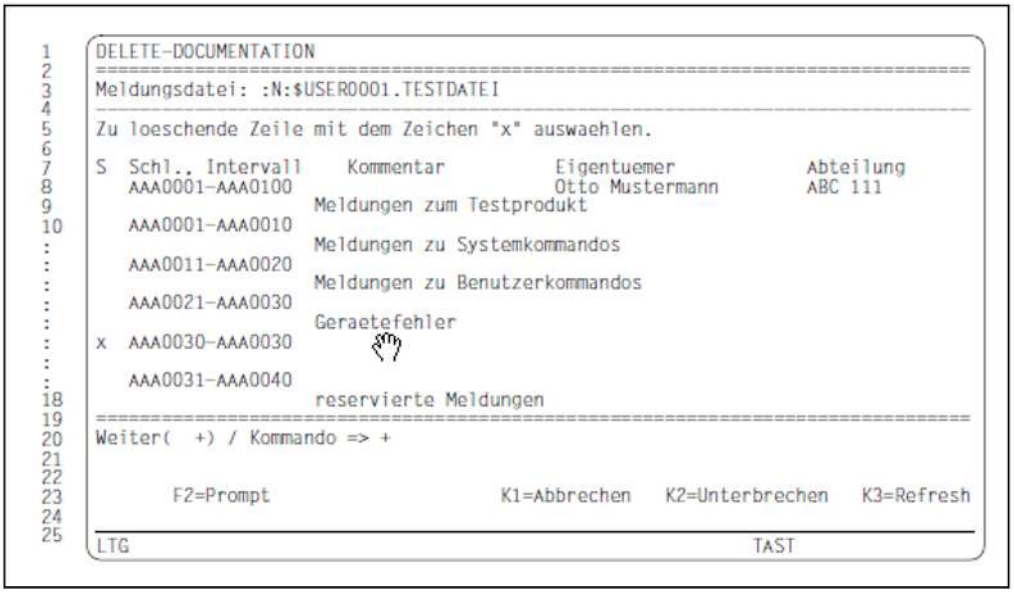
Funktion

In dieser Maske können Dokumentationszeilen, die für eine oder mehrere Meldungseinheiten verfasst wurden, gelöscht werden. Im Unterschied zur Maske MODIFY-DOCUMENTATION können nur die Zeilen gelöscht werden, welche mit einem „x“ in der Spalte S markiert werden. Die Dokumentationszeilen selbst sind gegen Überschreiben geschützt.

Mit DUE werden die ausgewählten Zeilen ohne nochmalige Rückfrage gelöscht.

> DUE in Verbindung mit „+“ oder „-“ blättert in der Maske vor- oder rückwärts. Eine ausgewählte Dokumentationszeile wird **nicht** gelöscht.

Maske



> *Anmerkung zur Maske*
>
> In diesem Fall muss vor dem endgültigen Löschen der Dokumentationszeile AAA0030-AAA0030 das „+“ in der Kommandozeile gelöscht werden. Erst dann löst DUE auch „Löschen der Dokumentationszeile“ aus.

Ausgabefelder

- **Meldungsdatei**
- **Schl., Intervall** (Meldungsschlüssel/Meldungsintervall)
- **Kommentar** (Kommentar zu den Meldungen)
- **Eigentuemer** (Name des Meldungsverantwortlichen)
- **Abteilung** (Bezeichnung des verantwortlichen Teams)

Beschreibung siehe "Maske MODIFY-DOCUMENTATION - Ändern, Hinzufügen und Löschen von Dokumentationszeilen".

Eingabefelder

S (Markierungsspalte)

Gültigkeitskriterien:
Buchstabe 'x'.

Kommando

Nähere Informationen zur

- Eingabe von Anweisungen siehe "Allgemeiner Maskenaufbau"
- Belegung der Funktionstasten siehe "Allgemeiner Maskenaufbau"

Anschließende Operationen:

DELETE-DOCUMENTATION	-> ?

DUE	Die Eingaben werden bestätigt und die Funktion DELETE-DOCUMENTATION ausgelöst; das Programm kehrt zur Maske MENU zurück.
K1	Die Funktion DELETE-DOCUMENTATION wird abgebrochen, das Programm kehrt zur Maske MENU zurück.

8.3 Anweisungen

In Batch-Aufträgen und Prozeduren wird die Meldungsbearbeitung über Anweisungen gesteuert. Die Operanden und ihre möglichen Werte entsprechen in den meisten Fällen den Feldern und deren Eingaben in den Masken. Der Anweisungsname stimmt überwiegend mit dem Maskentitel überein.

Die Meldungsbearbeitung beginnt nach Aufruf von MSGMAKER in der Regel mit der Anweisung //OPEN-MSG-FILE. Die geöffnete Meldungsdatei kann dann mit Hilfe aller weiteren Funktionen bearbeitet werden. Die Funktionen COPY, MOVE, SHOW und MERGE-MSG-FILES können direkt aufgerufen werden, um die Meldungsdatei zu bearbeiten.

8.3.1 Übersicht über die Anweisungen

Nachfolgende Übersicht enthält alle Anweisungen des Dienstprogramms MSGMAKER:

MSGMAKER-Anweisungen	Bedeutung
//ADD-DOCUMENTATION	ermöglicht die Dokumentation (Meldungsverantwortliche, Kommentar) zu Meldungen
//ADD-MSG	fügt eine oder mehrere Meldungseinheiten in die aktuelle Meldungsdatei ein
//COPY	kopiert Bestandteile einer Meldungsdatei (Meldungseinheiten, Dokumentationszeilen) von einer Meldungsdatei in eine andere oder innerhalb einer Meldungsdatei
//DELETE-DOCUMENTATION	löscht die Dokumentation (Meldungsverantwortliche, Kommentar) zu Meldungen
//DELETE-MSG	löscht eine oder mehrere Meldungseinheiten in der aktuellen Meldungsdatei
//END	beendet MSGMAKER; die geöffneten Meldungsdateien werden geschlossen.
//MERGE-MSG-FILES	mischt die Gesamtinhalte mehrerer Meldungsdateien in eine Meldungsdatei zusammen
//MODIFY-DOCUMENTATION	verändert die Dokumentation zu Meldungen
//MODIFY-MSG	verändert eine oder mehrere Meldungseinheiten in der aktuellen Meldungsdatei
//MODIFY-OPTION	legt fest, ob (Teile von) Meldungseinheiten überschrieben werden dürfen
//MOVE	kopiert Bestandteile einer Meldungsdatei (Meldungseinheiten und Dokumentationszeilen) von einer Meldungsdatei in eine andere oder innerhalb derselben Meldungsdatei. Im Unterschied zur Anweisung //COPY wird der Sendebereich gelöscht.
//OPEN-MSG-FILE	eröffnet eine Meldungsdatei
//SHOW	gibt Meldungseinheiten und die zugehörigen Dokumentationszeilen auf SYSOUT oder SYSLST aus

Zusätzlich können die allgemein gültigen SDF-Anweisungen benutzt werden. Diese Anweisungen sind im Handbuch „Dialogschnittstelle SDF“ [20] beschrieben.

8.3.2 Beschreibung der Anweisungen

- ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen
- ADD-MSG - Einfügen einer Meldungseinheit
- COPY - Meldungseinheit kopieren
- DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen
- DELETE-MSG - Meldungseinheit löschen
- END - MSGMAKER beenden
- MERGE-MSG-FILES - Meldungsdateien mischen
- MODIFY-DOCUMENTATION - Verändern und Löschen von Dokumentationszeilen
- MODIFY-MSG - Meldungseinheit verändern
- MODIFY-OPTION - Überschreiben einer Meldungseinheit
- MOVE - Meldungseinheit kopieren und löschen
- OPEN-MSG-FILE - Meldungsdatei eröffnen
- SHOW - Inhalt einer Meldungsdatei anzeigen

8.3.2.1 ADD-DOCUMENTATION - Eintragen von Dokumentationszeilen

Funktion

Mit der Anweisung //ADD-DOCUMENTATION können Meldungseinheiten dokumentiert werden. Die Zuordnung der Dokumentationszeilen zu den Meldungseinheiten erfolgt über den Meldungsschlüssel. Neben den Personen, die sich für die Meldungsdatei verantwortlich zeichnen, kann ein 60 Zeichen umfassender Kommentar eingetragen werden.
Die Richtigkeit der Dokumentation und ihr Bezug zu den Meldungen wird nicht überprüft!

Werden bereits beschriebene Meldungen erneut dokumentiert, entscheidet die Voreinstellung in der Anweisung //MODIFY-OPTION, ob die Dokumentation überschrieben werden darf.

Unterschiede zur Anweisung //ADD-DOCUMENTATION im Menümodus

Die Anweisung //ADD-DOCUMENTATION, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //ADD-DOCUMENTATION in Kommandoprozeduren durch den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

ADD-DOCUMENTATION

```
MSG-ID = *INTERVAL(...) / <name 7..7>

    *INTERVAL(...)

        |   FROM = <name 7..7>

        |   ,TO = <name 7..7>

,OWNER = *NONE / *PARAMETERS(...)

    *PARAMETERS(...)

        |   NAME = *NONE / <c-string 1..20>

        |   ,TEAM = *NONE / <c-string 1..15>

,COMMENTS = *NONE / <c-string 1..60>
```

Operanden

MSG-ID = *INTERVAL(...) / <name 7..7>
Bezeichnet eine oder mehrere Meldungseinheiten, die dokumentiert werden sollen.

MSG-ID = <name 7..7>
Der vollständige Meldungsschlüssel, bestehend aus der dreistelligen Meldungsklasse und der vierstelligen Meldungsnummer, muss eingegeben werden.

MSG-ID = *INTERVAL(...)
Bezeichnet ein Meldungsintervall. Die Grenzen des Intervalls müssen einer Meldungsklasse angehören.

FROM = <name 7..7>
Bezeichnet die untere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

TO = <name 7..7>
Bezeichnet die obere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

OWNER = *NONE / *PARAMETERS(...)
Bezeichnet die Personen, die für die Meldungsdatei verantwortlich sind.

OWNER = *NONE
Er werden keine Angaben zu den verantwortlichen Personen gemacht.

OWNER = *PARAMETERS(...)

NAME = *NONE / <c-string 1..20>
Bezeichnet die Person, die für die Meldungsdatei verantwortlich ist.

TEAM = *NONE / <c-string 1..15>
Bezeichnet das Team (Abteilung), das für die Meldungsdatei verantwortlich ist.

COMMENTS = *NONE / <c-string 1..60>
Maximal 60 Zeichen umfassender Kommentar zu den ausgewählten Meldungen.

8.3.2.2 ADD-MSG - Einfügen einer Meldungseinheit

Funktion

Mit der Anweisung //ADD-MSG wird eine neue Meldungseinheit in die aktuell geöffnete Meldungsdatei eingefügt.

Neben dem Meldungschlüssel, der die Adresse der Meldungseinheit innerhalb der Meldungsdatei darstellt, müssen die Meldungsattribute, Kennbuchstaben der Sprachen und Texte festgelegt werden.

Zu den Meldungsattributen zählen Zugriffsmethode, Ausgabeziel, Berechtigungsschlüssel, Weight Code und „Garantie“ der Meldung. Zu jedem definierten Kennbuchstaben muss ein Meldungstext geschrieben werden, wohingegen Bedeutungs- und Maßnahmetext nach Bedarf eingetragen werden können. Werden im Meldungstext Inserts vereinbart, können ihnen im Operanden INSERT-ATTRIBUTES Namen und Default-Texte zugewiesen werden. Die Meldungsattribute und Insertattribute werden nur einmal definiert und gelten für alle Texte und Sprachen einer Meldungseinheit.

In eine bereits bestehende Meldungseinheit können Texte in einer neuen Sprache nicht eingetragen werden. In diesem Fall würde mit der Anweisung //ADD-MSG die existierende Meldungseinheit überschrieben werden. Im Batch-Betrieb und Dialogprozeduren wird Spin-Off ausgelöst, falls ein Überschreiben der Meldung nicht erlaubt ist (siehe Operand OVERWRITE der Anweisung //ADD-MSG und //MODIFY-OPTION).

Unterschiede zur Anweisung //ADD-MSG im Menümodus

Die Anweisung //ADD-MSG, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //ADD-MSG in Kommandoprozeduren, durch

- den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- zusätzliche Werte des Operanden MSG-ID
- den Operanden OVERWRITE, der im Menümodus nicht verfügbar ist.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

```
ADD-MSG

 MSG-ID = <name 7..7>

 ,ACCESS-METHODS = *ISAM / list-poss(4): *ISAM / *DLAM / *LOCAL-DLAM / *MINIMIP / *BAMR

 ,DESTINATIONS = *USER-TASK / *ALL(...) / list-poss(2): *USER-TASK / *CONSOLE(...)

     *ALL(...)

         |    ROUTING-CODE = <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG

     CONSOLE(...)

         |    ROUTING-CODE = <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG

 ,WEIGHT = *NONE / <integer 0..99>

 ,WARRANTY = *NO / *YES

 ,MSG-TEXT-OUTPUT = *UPPER-CASE / *LOWER-CASE

 ,LANGUAGES = list-poss(8): <name 1..1>(...)

     <name 1..1>(...)

         |    MSG-TEXT = <c-string 1..220 with-low>

         |    ,MEANING = *NONE / list-poss(256): <c-string 1..74 with-low>

         |    ,RESPONSE = *NONE / list-poss(256): <c-string 1..74 with-low>

 ,INSERT-ATTRIBUTES = *NONE / list-poss(30): <integer 0..29>(...)

     <integer 0..29>(...)

         |    NAME = *NONE / <structured-name 1..20>

         |    ,DEFAULT-VALUE = *NONE / <c-string 1..54 with-low> / *EMPTY-STRING

         |    ,AUTOMATIC-HELP = *NO / *YES(...)

         |          *YES(...)

         |                |     PREFIX = *BY-INSERT-VALUE / <name 3...7>

 ,OVERWRITE = *STD / *YES / *NO
```

Operanden

MSG-ID = <name 7..7>
Bezeichnet die Meldungseinheit, die eingefügt werden soll. Der vollständige Meldungsschlüssel, bestehend aus der dreistelligen Meldungsklasse und der vierstelligen Meldungsnummer, muss eingegeben werden.

ACCESS-METHODS = *ISAM / list-poss(4): *ISAM / *DLAM / *LOCAL-DLAM / *MINIMIP / *BAMR
Beschreibt die MIP-Zugriffsmethode für die Meldungen. Die Kombination von maximal drei Methoden ist möglich, wobei ISAM und DLAM bzw. ISAM und LOCAL-DLAM nicht zusammen ausgewählt werden dürfen.

ACCESS-METHODS = *ISAM
Die Meldungen werden über den ISAM-Schlüssel (=Meldungsschlüssel) gesucht.

ACCESS-METHODS = *DLAM
Die Zugriffsmethode DLAM ist für besonders häufig auftretende Meldungen vorgesehen. Wird eine Meldungsdatei, die eine DLAM-Meldung enthält, aktiviert, wird diese in den Arbeitsspeicher geladen. Die DLAM-Meldung kann von MIP direkt, ohne Zugriff auf die Meldungsdatei, ausgegeben werden.

ACCESS-METHODS = *LOCAL-DLAM / *MINIMIP / *BAMR
Diese Zugriffsmethoden sind für den internen Gebrauch beim Hersteller reserviert (siehe DHB-Abschnitt MIP).

DESTINATIONS = *USER-TASK / *ALL(...) / list-poss(2): *USER-TASK / *CONSOLE
Dieser Operand dokumentiert die möglichen Ausgabeziele der Meldung aus der Sicht des Meldungsverfassers. Das Ausgabeziel selbst wird im Operanden DEST (Destination Code) des Makros MSG7X festgelegt.

DESTINATIONS = *USER-TASK
Die Meldung hat das Ausgabeziel SYSOUT, SYSLST oder einen benutzereigenen Speicherbereich.

DESTINATIONS = *ALL(...)
Alle Ausgabeziele sind möglich.

> **ROUTING-CODE = <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG**
> Der einstellige Berechtigungsschlüssel wird bei Konsolausgaben als Zielangabe ausgewertet. Die Bedeutung der Berechtigungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].
>
> **ROUTING-CODE = <alphanum-name 1..1>**
> Als Berechtigungsschlüssel kann jeder Buchstabe, jede Zahl oder die Sonderzeichen #, $, oder @ verwendet werden. Die Bedeutung der schon vorbelegten
> Berechtigungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].
>
> **ROUTING-CODE = *MAIN-CONSOLE**
> Ziel der Meldung ist der besondere Berechtigungsschlüssel *, der stets zumindest der Hauptkonsole zugeordnet ist.
>
> **ROUTING-CODE = *CONSLOG**
> Meldungen, die keine Antwort erfordern, werden nur in der CONSLOG-Datei protokolliert. ROUTING-CODE = *CONSLOG entspricht der Zuweisung ROUTING-CODE = @.

DESTINATION = *CONSOLE(...)
Das Ausgabeziel der Meldung ist eine Konsole. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

> **ROUTING-CODE = <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG**
> Der einstellige Berechtigungsschlüssel wird bei Konsolausgaben als Zielangabe ausgewertet. Die Bedeutung der Berechtigungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].
>
> **ROUTING-CODE = <alphanum-name 1..1>**
> Als Berechtigungsschlüssel kann jeder Buchstabe, jede Zahl oder die Sonderzeichen #, $, oder @ verwendet werden. Die Bedeutung der schon vorbelegten
> Berechtigungsschlüssel finden Sie im Handbuch „Systembetreuung“ [5].

ROUTING-CODE = *MAIN-CONSOLE
Ziel der Meldung ist der besondere Berechtigungsschlüssel *, der stets zumindest der Hauptkonsole zugeordnet ist.

ROUTING-CODE = *CONSLOG
Meldungen, die keine Antwort erfordern, werden nur in der CONSLOG-Datei protokolliert. ROUTING-CODE = *CONSLOG entspricht der Zuweisung ROUTING-CODE = @.

WEIGHT = *NONE / <integer 0..99>
Der Weight Code bezeichnet das Meldungsgewicht.
Ein Gewicht muss nur für eine Meldung vereinbart werden, deren Ausgabeziel eine Konsole ist, andernfalls ist *NONE der voreingestellte Wert.
Nähere Informationen zum Weight Code finden Sie im Handbuch „Systembetreuung“ [5].

WARRANTY = *NO / *YES
Das Meldungsattribut „Garantie“ wird von MIP ausgewertet. Die Garantieerklärung besagt, dass bestimmte Teile der Meldung in den kommenden BS2000-Versionen nicht mehr geändert werden.

Folgende Meldungs-Bestandteile werden garantiert:

- Meldungsschlüssel
- Nummerierung und Bedeutung der Inserts

Der Meldungstext wird **nicht** garantiert.
MIP erzeugt für garantierte Meldungen S-Variablen.
Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

MSG-TEXT-OUTPUT = *UPPER-CASE / *LOWER-CASE
Bezeichnet die Schreibweise des Meldungstextes bei der Ausgabe. Die Default-Texte für Inserts sowie die Texte, die über den Makro MSG7X vorgegeben sind, werden ebenfalls an dieses Format angepasst. Diese Festlegung gilt nicht für den Bedeutungs- und Maßnahmetext.

Folgende Ausgabeziele sind möglich:

- nach SYSOUT
- in eine SYSLST-Datei
- in einen benutzereigenen Speicherbereich
- an eine Konsole

Die Ausgabe erfolgt durch den Makro MSG7X oder durch das Kommando /HELP-MSG-INFORMATION.

MSG-TEXT-OUTPUT = * UPPER-CASE
Der Meldungstext und die Insert-Texte (Default oder aktuell) werden immer in Großbuchstaben umgewandelt.

MSG-TEXT-OUTPUT = *LOWER-CASE
Der Meldungstext und die Insert-Texte (Default oder aktuell) werden gemäß ihrer Schreibweise im Operanden LANGUAGES = ...(MSG-TEXT='...') ausgegeben.

LANGUAGES = list-poss(8): <name 1..1>(...)
Kennzeichnet die Sprache, in der der Meldungs-, Bedeutungs- und Maßnahmetext eingetragen und ausgegeben wird. Die Kennzeichnung besteht aus einem Buchstaben, welcher für Deutsch D und für Englisch E lautet. Für andere Sprachen gibt es keine Konvention.
Die Texte können in maximal acht Sprachen eingegeben werden.

MSG-TEXT = <c-string 1..220 with-low>
Zu jeder Meldung **muss** ein Meldungstext geschrieben werden, der maximal 220 Zeichen lang sein kann. In dieser Länge sind die Zeichenfolgen (&00) bis (&29) enthalten, jedoch nicht die Default-Texte, welche bei Meldungsausgabe diese Zeichenfolgen ersetzen.

Trennungszeichen „^“
Der Meldungstext wird als fortlaufende Textzeile eingegeben. Um den Meldungstext zu strukturieren, kann das Trennungszeichen „^“ an jeder beliebigen Stelle und beliebig oft in den Meldungstext eingefügt werden. Der Text, der auf das Trennungszeichen folgt, wird in der jeweils nächsten Zeile ausgegeben. Die Textausgabe durch den Makro MSG7X oder durch das Kommando /HELP-MSG-INFORMATION kann mit der Anweisung //SHOW überprüft werden.

Beispiel

```
MSG-TEXT ='THIS MESSAGE IS OUTPUT TO ''(&04)''^ IN THE FOLLOWING

LANGUAGES; ''(&00)'', ''(&01)'' AND ''(&03)'''
```

Ausgegeben wird folgender Text:

```
% ABC1234 THIS MESSAGE IS OUTPUT TO 'SYSOUT'

  IN THE FOLLOWING LANGUAGES; 'E', 'D' AND 'F'
```

Nähere Informationen zum Meldungstext finden Sie im Handbuch „Systembetreuung“ [5].

MEANING = *NONE / list-poss(256): <c-string 1..74 with-low>
Zu jeder Meldung **kann** in der entsprechenden Sprache ein Bedeutungstext geschrieben werden, der maximal 256 Zeilen mit je 74 Zeichen umfassen kann.Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

RESPONSE = *NONE / list-poss(256): <c-string 1..74 with-low>
Zu jeder Meldung **kann** in der entsprechenden Sprache ein Maßnahmetext geschrieben werden, der maximal 256 Zeilen mit je 74 Zeichen umfassen kann.Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

INSERT-ATTRIBUTES = *NONE / list-poss(30): insert-number <integer 0..29>(...)
Beschreibt die im Meldungstext definierten Inserts.

INSERT-ATTRIBUTES = *NONE
Den Inserts werden keine Eigenschaften zugeordnet.

INSERT-ATTRIBUTES = list-poss(30): insert-number <integer 0..29>(...) Gibt die Nummer des Inserts an, für das ein Default-Text vereinbart wird. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

NAME = *NONE / <structured-name 1..20>
Bezeichnet den Namen des Inserts. Insertnamen werden von MIP ausgewertet. Sie werden von MIP zur Erzeugung von S-Variablen verwendet. Der im Operanden DEFAULT-VALUE angegebene Wert bzw. der aktuelle Wert (MSG7X) wird zum Inhalt dieser S-Variablen. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].
Die Buchstaben der hier eingetragenen Namen werden immer in Großbuchsstaben umgewandelt.

Beispiel

Die Inserts mit den Nummern 0 und 4 sollen die Namen „LANGUAGE“ und „DESTINATION“ erhalten:

```
INSERT-ATTRIBUTES=(0(NAME=LANGUAGE),4(NAME=DESTINATION))
```

DEFAULT-VALUE = *NONE / <c-string 1..54 with-low> / *EMPTY-STRING
Bezeichnet den voreingestellten Text, der anstelle des Inserts in den Meldungstext eingefügt wird, falls im Makro MSG7X kein aktueller Text dafür angegeben ist. DEFAULT-VALUE = *NONE bedeutet, dass kein Inserttext voreingestellt ist. Mit DEFAULT-VALUE = *EMPTY-STRING wird ein Leerstring als Inserttext vereinbart. Der Operandenwert DEFAULT-VALUE = *NULL, der *EMPTY-STRING entspricht, wird aus Kompatibilitätsgründen in Batch-Aufträgen und Prozeduren weiterhin unterstützt. Die Ausgabe des Default-Textes hängt vom Operanden MSG-TEXT-OUTPUT ab.

Beispiel

Die Inserts mit den Nummern 4, 0 und 1 sollen die Texte „F“, „E“ und „D“ erhalten:

```
INSERT-ATTRIBUTES=(4(DEF='F'),0(DEF='E'),1(DEF='D'))
```

AUTOMATIC-HELP = *NO / *YES(...)
Wird über ein Insert ein Meldungsschlüssel oder ein Teil davon ausgegeben (z.B. bei DVS-Fehlern wird lediglich die Meldungsnummer ausgegeben), so wird über die automatische Hilfsfunktion MIP dazu veranlaßt, zusätzlich zu diesem Meldungsschlüssel den zugehörigen Meldungstext auszugeben.

AUTOMATIC-HELP = *NO
Es wird nur der Meldungsschlüssel ausgegeben.

AUTOMATIC-HELP = *YES(...)
Zum Meldungsschlüssel wird der Meldungstext ausgegeben.

PREFIX = *BY-INSERT-VALUE / <name 3...7>
Zu einem in der Fehlermeldung ausgegebenen Insert oder zu einer explizit angegebenen Meldungsnummer wird der gesamte Meldungstext ausgegeben.

PREFIX = *BY-INSERT-VALUE
Mögliche Ausgaben:

- Meldungsschlüssel mit sieben Zeichen
- DVS-Fehlerschlüssel mit „0...“ am Anfang. Die Fehlerklasse DMS wird durch /HELP-MSG-INFORMATION vorangestellt
- vierstelliger Meldungsschlüssel, der in einen siebenstelligen Meldungsschlüssel umgewandelt wird.

MSGMAKER überprüft nicht, ob der Default-Insertwert einen gültigen Meldungsschlüssel bzw. eine gültige Meldungsnummer enthält. Liegt kein gülter Meldungsschlüssel vor, wird von MIP die Meldung „MESSAGE UNDEFINED“ ausgegeben.

Folgende Regeln gelten für die Länge der Insertwerte bzw. Standardinsertwerte:

Länge < 4:	der Wert wird links bis zu vier Zeichen mit „0“ ergänzt
Länge > 4 und < 7:	der Wert wird bei den 4 letzten Zeichen abgeschnitten
Länge > 7:	der Wert wird bei den 7 letzten Zeichen abgeschnitten

PREFIX = <name 3...7>
Für die Ausgabe im Kommando /HELP-MSG-INFORMATION wird der Meldungsschlüssel aus Präfix und Insertwert oder Default-Inserttext gebildet. Die ersten drei Zeichen geben die Meldungsklasse an und müssen Buchstaben sein.

Entspricht der Präfix einem vollständigen Meldungsschlüssel, wird die automatische Hilfsfunktion direkt darauf angewendet, unabhängig vom Insertwert.

Der Insertwert wird entweder mit „0“ erweitert oder an der linken Seite abgeschnitten, damit das Ergebnis einem Meldungsschlüssel mit 7 Zeichen entspricht.

Beispiel

Präfix	Insertwert	/HELP-MSG-INFORMATION
ABC	12	ABC0012
ABCX	123	ABCX123
ABCX	1234	ABCX234
ABC01	XYZ1234	ABC0134
ABC01	<none>	ABC01 (error)

Ist eine Insertnummer mehrmals in einem Meldungstext definiert, wird die automatische Hilfsfunktion nur einmal auf diese Insertnummer angewendet.

Bei einer Meldung, die eine Antwort erfordert, gibt MIP die Meldung nach Ausführen der automatischen Hilfsfunktion erneut aus. Die automatische Hilfsfunktion wird nicht aktiviert, wenn die Meldung an einen Benutzerpuffer geschickt wird.

OVERWRITE = *STD / *YES / *NO
Gibt an, ob die aktuelle Meldungseinheit von einer bereits bestehenden Meldungseinheit mit identischem Meldungsschlüssel überschrieben werden darf.

OVERWRITE = *STD
Der Defaultwert *STD bezieht sich auf die letzte OVERWRITE-Vereinbarung in der Anweisung //MODIFY-OPTION (siehe "MODIFY-OPTION - Überschreiben einer Meldungseinheit").

OVERWRITE = *YES
Die bestehende Meldungseinheit wird durch die aktuelle Meldungeinheit vollständig überschrieben.

OVERWRITE = *NO
Die bestehende Meldungseinheit wird durch die aktuelle Meldungseinheit nicht überschrieben.

Beispiel

MIP meldet den DVS-Fehlercode:

```
% DMS05F8 DMS error code '0533'. Command processing aborted.
  IN SYSTEM MODE: /HELP-MSG DMS(0533)
```

Mit der automatischen Hilfsfunktion wird der entsprechende Meldungstext automatisch ausgegeben:

```
DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET '(&00)'. COMMAND TERMINATED
```

8.3.2.3 COPY - Meldungseinheit kopieren

Funktion

Mit der Anweisung //COPY können Meldungseinheiten mit oder ohne den zugehörigen Dokumentationszeilen von einer Meldungsdatei in eine andere oder innerhalb derselben Meldungsdatei kopiert werden. Die Meldungseinheit in der Sendedatei wird durch den Kopiervorgang nicht verändert.

Jede Meldungseinheit ist durch ihren Meldungsschlüssel gekennzeichnet. Wird dem Operanden MSG-ID eine Meldungsklasse oder ein Meldungsintervall zugewiesen, können mehrere Meldungseinheiten in einem Vorgang kopiert werden. Wird dem Operanden INFORMATION der Wert DOCUMENTATION zugewiesen, werden die im angegebenen Meldungsbereich definierten Dokumentationszeilen kopiert.

Sind Sende- und Empfangsdatei des Kopiervorgangs identisch, müssen im Operanden TO-MSG-ID neue Meldungsschlüssel vereinbart werden.
Unabhängig von der aktuell geöffneten Datei, können Meldungseinheiten zwischen zwei weiteren Meldungsdateien kopiert werden. Die Operanden FROM-FILE und TO-FILE ermöglichen den Zugriff auf diese Dateien; die aktuelle Meldungsdatei bleibt geöffnet und wird nicht verändert, falls sie weder Sende- noch die Empfangsdatei ist.

Sind im Operanden INFORMATION=MESSAGES(...) explizit Sprachen angegeben, werden nur die Texte (Meldungs-, Bedeutungs- und Maßnahmetext) in diesen Sprachen in die Empfangs-Meldungseinheit kopiert. Existiert diese Meldungseinheit nicht, wird eine Meldungseinheit mit den Meldungsattributen (Zugriffsmethode, Ausgabeziel, ...) der Sende-Meldungseinheit erzeugt.

Ist in der Empfangs-Meldungseinheit bereits ein Text unter identischem Meldungsschlüssel und Kennbuchstaben für die Sprache abgespeichert, wird der Text der Empfangs-Meldungseinheit mit dem Text der Sende-Meldungseinheit überschrieben.
Im **Dialog** wird für diesen Fall eine Fehlermeldung ausgegeben mit der Frage, ob die bestehende Meldungseinheit überschrieben werden soll oder nicht.
In **Batch-Aufträgen** und **Prozeduren** werden die angegebenen Bestandteile der Meldungsdatei nicht kopiert und die Verarbeitung fortgesetzt, vorausgesetzt, in der Anweisung //COPY oder //MODIFY-OPTION wurde nicht OVERWRITE=*YES vereinbart.

Unterscheiden sich die Texte nur durch den Kennbuchstaben für die Sprache, wird der Text in der Empfangs-Meldungseinheit angefügt.

> ¡ Soll der Gesamtinhalt einer Meldungsdatei in eine andere Datei kopiert werden, ist die Anweisung //MERGE-MSG-FILES der Anweisung //COPY vorzuziehen (siehe Anweisung //MERGE-MSG-FILES, "MERGE-MSG-FILES - Meldungsdateien mischen")

Unterschiede zur Anweisung //COPY im Menümodus

Die Anweisung //COPY, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //COPY in Kommandoprozeduren, durch

- den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- einen zusätzlichen Wert des Operanden MSG-ID
- den Operanden OVERWRITE, der im Menümodus nicht verfügbar ist.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

COPY

```
MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
        *CLASS(...)
              |       MSG-CLASS = <name 3..3>
        *INTERVAL(...)
              |       FROM = <name 7..7>
              |       ,TO = <name 7..7>
,INFORMATION = *MESSAGES (...) / *ALL / list-poss(4): *DOCUMENTATION / *COMPONENT-ID /
                      *CORRECTION-INFO / *MESSAGES(...)
        *MESSAGES(...)
              |       LANGUAGES = *ALL / list-poss(8): <name 1..1>
,FROM-FILE = *CURRENT / <filename 1..54>
,TO-MSG-ID = *SAME / *CLASS(...) / <alphanum-name 4..7 with-wild>
        *CLASS(...)
              |       MSG-CLASS = <name 3..3>
,TO-FILE = *CURRENT / <filename 1..54> (...)
        <filename 1..54>(...)
              |       ,FILE-FORMAT = *NEW / *OLD                                  1)
,OVERWRITE = *STD / *YES / *NO
```

1) Der Operand FILE-FORMAT ist obsolet. Er kann aus Kompatibilitätsgründen angegeben werden.

Operanden

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
Bezeichnet die Meldungsschlüssel einer oder mehrerer Meldungseinheiten der eröffneten Meldungsdatei, die kopiert werden sollen.
Durch die Zuweisung INFORMATION = DOCUMENTATION können die zu den Meldungsschlüsseln definierten Dokumentationszeilen ebenfalls kopiert werden.

MSG-ID = *ALL
Alle im Operanden INFORMATION definierten Bestandteile einer Meldung werden kopiert.In diesem Fall muss dem Operanden TO-MSG-ID der Wert *SAME zugewiesen werden.

MSG-ID = *CLASS(...)
Alle Bestandteile der Meldungen innerhalb der angegebenen Meldungsklasse werden kopiert.

MSG-CLASS = <name 3..3>

Kennzeichnet die dreistellige Meldungsklasse.

MSG-ID = *INTERVAL(...)
Alle Bestandteile der Meldungen innerhalb dieses Meldungsintervalls werden kopiert.
Für den Fall, dass das Meldungsintervall mehr als eine Meldungsklasse beinhaltet, muss dem Operanden TO-MSG-ID der Wert *SAME zugewiesen werden.

FROM = <name 7..7>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls.

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssel des Meldungsintervalls.

MSG-ID = list-poss(2000): <name 7..7>
Gibt den vollständigen, siebenstelligen Meldungsschlüssel an. Werden mehrere Meldungsschlüssel als Liste angegeben, muss dem Operanden TO-MSG-ID der Wert *SAME zugewiesen werden.

INFORMATION = *MESSAGES(...) / *ALL / list-poss(3): *MESSAGES(...) / *DOCUMENTATION / *CORRECTION-INFO / *COMPONENT-ID
Bezeichnet die Bestandteile der Meldungsdatei, die kopiert werden sollen.

INFORMATION = *MESSAGES(...)
Bezeichnet Meldungseinheiten, die kopiert werden sollen. Über den Operanden MSG-ID werden die Meldungseinheiten ausgewählt.

LANGUAGES = *ALL / list-poss(8): <name 1..1>
Bezeichnet die Kennbuchstaben der Sprachen. Meldungs-, Bedeutungs- und Maßnahmetexte in diesen Sprachen werden kopiert.

LANGUAGES = *ALL
Die vollständige Meldungseinheit, einschließlich aller Meldungsattribute, Inserts und Insert-Attribute sowie alle Texte werden kopiert.

LANGUAGES = list-poss(8): <name 1..1>
Der Meldungstext und/oder Bedeutungs-, Maßnahmetext in den ausgewählten Sprache(n) werden kopiert. Die Meldungsattribute und Insert-Attribute der Sende-Meldungseinheit werden nur kopiert, wenn die Empfangs-Meldungseinheit nicht existiert. Ansonsten wird die Meldung `MSME108` ausgegeben.

INFORMATION = *ALL
Alle Bestandteile der Meldungsdatei (Meldungseinheiten und Dokumentationszeilen) werden kopiert. Wird dem Operanden MSG-ID = *ALL zugewiesen, wird der gesamte Inhalt der Meldungsdatei kopiert.

INFORMATION = *DOCUMENTATION
Alle Dokumentationszeilen, deren Meldungsschlüssel im ausgewählten Bereich (Operand MSG-ID) liegen, werden kopiert. Überschneiden mehrere definierte Meldungsintervalle den festgelegten Meldungsbereich, werden die Dokumentationszeilen dieser Intervalle ebenfalls kopiert.

Beispiel

MSG-ID = *INTERVAL (TST1500-TST1599).
Die Dokumentationszeilen der Meldungsintervalle (TST1500-TST1510),(TST1000-TST1999), (TST1000-TST1500), TST1510, ... werden kopiert.

INFORMATION = *CORRECTION-INFO / *COMPONENT-ID
Diese Operanden sind für den internen Gebrauch beim Hersteller reserviert.

FROM-FILE = *CURRENT / <filename 1..54>
Name der Meldungsdatei, aus welcher die bezeichneten Bestandteile (Operand MSG-ID) kopiert werden.
Der Operandenwert *CURRENT bezeichnet die aktuelle Meldungsdatei, die mit der Anweisung //OPEN-MSG-FILE zuletzt eröffnet wurde.

TO-MSG-ID = *SAME / *CLASS(...) / <alphanum-name 4..7 with-wild>
Gibt die neuen Meldungsschlüssel der kopierten Meldungsbestandteile im Empfangsbereich an.

TO-MSG-ID = *SAME
Die Meldungsbestandteile behalten beim Kopiervorgang ihren Meldungsschlüssel bei. Der Wert *SAME darf nicht angegeben werden, wenn Meldungsbestandteile innerhalb einer Datei kopiert werden. Der Wert *SAME muss angegeben werden, wenn dem Operanden MSG-ID eine Liste von Meldungsschlüsseln, der Wert *ALL oder Meldungsschlüssel aus mehreren Meldungsklassen zugwiesen wird.

TO-MSG-ID = *CLASS(...)
Bezeichnet eine neue Meldungsklasse für die kopierten Meldungsbestandteile. Die Meldungsnummern bleiben unverändert.

> **MSG-CLASS = <name 3..3>**
> Bezeichnet die neue, dreistellige Meldungsklasse.

TO-MSG-ID = <alphanum-name 4..7 with-wild>
Der neue Meldungsschlüssel wird entweder vollständig angegeben oder über einen Teilnamen mit Platzhalter definiert.
Der Teilname bildet den Beginn des neuen Meldungsschlüssels, die restlichen Stellen, symbolisiert durch *, werden unverändert vom alten Meldungsschlüssel übernommen.

Beispiel

> Alter Meldungsschlüssel MSG-ID = ABC1234;
> TO-MSG-ID = TST0* bewirkt, dass die ersten vier Stellen des neuen Meldungsschlüssels zu TST0 abgeändert, die restlichen drei Stellen aber vom alten Meldungsschlüssel übernommen werden. Ergebnis: TST0234.

TO-FILE = *CURRENT / <filename 1..54>(...)
Bezeichnet die Meldungsdatei, in welche die Meldungsbestandteile kopiert werden.

T**O-FILE = *CURRENT**
Empfangsdatei ist die mit der Anweisung //OPEN-MSG-FILE zuletzt geöffnete Meldungsdatei. Als Empfangsdatei muss sie im UPDATE-Modus geöffnet sein.

TO-FILE = <filename 1..54>
Explizite Angabe der Meldungsdatei, in welche die Meldungsbestandteile kopiert werden. Eine noch nicht bestehende Empfangsdatei wird erzeugt, wobei analog zur Sendedatei die Dateiart (Customer oder Standard), der Produktname und die Produktversion festgelegt werden.

OVERWRITE = *STD / *YES / *NO
Gibt an, ob die Meldungsbestandteile der Sendedatei einen bereits bestehenden Bereich der Empfangsdatei überschreiben dürfen.
Der Default-Wert *STD entspricht dem Wert der letzten OVERWRITE-Vereinbarung in der Anweisung //MODIFY-OPTION.

Beispiel

```
//COPY MSG-ID=TST000,INFORMATION=MESSAGE(LANGUAGE=E),-
  FROM-FILE=SYSMES.TSTFILE,TO-MSG-ID=*CLASS(TTO)
```

8.3.2.4 DELETE-DOCUMENTATION - Löschen von Dokumentationszeilen

Funktion

Mit der Anweisung //DELETE-DOCUMENTATION werden die Dokumentationszeilen zu Meldungseinheiten gelöscht.
Die dokumentierten Meldungen werden über den Meldungsschlüssel ausgewählt.

Unterschiede zur Anweisung //DELETE-DOCUMENTATION im Menümodus

Die Anweisung //DELETE-DOCUMENTATION, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //DELETE-DOCUMENTATION in Kommandoprozeduren, durch den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

```
DELETE-DOCUMENTATION

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / <name 7..7>

    *CLASS(...)

        |    MSG-CLASS = <name 3..3>

    *INTERVAL(...)

        |    FROM = <name 7..7>

        |    ,TO = <name 7..7>

        |    ,DELETE-SUBSETS = *NO / *YES
```

Operanden

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / <name 7..7>
Bezeichnet eine oder mehrere Meldungseinheiten, deren Dokumentation gelöscht werden soll.

MSG-ID = *ALL
Alle Dokumentationszeilen der aktuell geöffneten Meldungsdatei werden gelöscht.

MSG-ID = *CLASS(...)

MSG-CLASS = <name 3..3>
Die Dokumentationszeilen, die für Meldungen der hier bezeichneten Meldungsklasse verfasst worden sind, werden gelöscht.

MSG-ID = *INTERVAL(...)
Die Dokumentationszeilen, die für Meldungen des hier bezeichneten Intervalls verfasst worden sind, werden gelöscht. Die Grenzen des Intervalls müssen einer Meldungsklasse angehören.

FROM = <name 7..7>
Bezeichnet die untere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

TO = <name 7..7>
Bezeichnet die obere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

DELETE-SUBSETS = *NO / *YES
Ermöglicht die genaue Bestimmung der zu löschenden Dokumentationszeilen, die in diesem Meldungsintervall definiert sind.

DELETE-SUBSETS = *NO
Es werden nur die Dokumentationszeilen gelöscht, die exakt für das mit FROM bis TO bezeichnete Meldungsintervall definiert worden sind. Die Dokumentation zu dem Meldungsintervall erfolgte in einer Zuweisung.

Beispiel

```
//ADD-DOCUMENTATION MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0020),
    OWNER=*PARAMETERS(NAME='Smith', TEAM='TST0815'),
    COMMENTS='documentation for messages AAA0001 to AAA0020'
//DELETE-DOCUMENTATION MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0020,
    DELETE-SUBSETS=NO)
```

DELETE-SUBSETS = *YES
Es werden alle Dokumentationszeilen gelöscht, die in das mit FROM bis TO bezeichnete Meldungsintervall fallen. Die Dokumentation zu dem Meldungsintervall erfolgte über mehrere Zuweisungen.

Beispiel

```
//ADD-DOCUMENTATION MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0008),
    OWNER=*PARAMETERS(NAME='Smith', TEAM='TST0815'),
    COMMENTS='documentation for messages AAA0001 to AAA0008'
//ADD-DOCUMENTATION MSG-ID=AAA0010, OWNER=*PARAMETERS(NAME='Smith',
    TEAM='TST0815'), COMMENTS='documentation for message AAA0010'
//ADD-DOCUMENTATION MSG-ID=*INTERVAL(FROM=AAA0015,TO=AAA0019),
    OWNER=*PARAMETERS(NAME='Smith', TEAM='TST0815'),
    COMMENTS='documentation for messages AAA0015 to AAA0019'
//DELETE-DOCUMENTATION MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0020,
    DELETE-SUBSETS=*YES)
```

MSG-ID = <name 7..7>
Die Dokumentation zu dem hier bezeichneten, vollständigen Meldungsschlüssel wird gelöscht.

8.3.2.5 DELETE-MSG - Meldungseinheit löschen

Funktion

Mit der Anweisung //DELETE-MSG können eine oder mehrere Meldungseinheiten der geöffneten Meldungsdatei gelöscht werden.
Sprachabhängige Meldungsteile, wie Meldungs-, Bedeutungs- und Maßnahmetext können ebenfalls entfernt werden, jedoch mit der Einschränkung, dass beim Entfernen aller Meldungstexte die gesamte Meldungseinheit automatisch gelöscht wird.

Unterschiede zur Anweisung //DELETE-MSG im Menümodus

Die Anweisung //DELETE-MSG, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //DELETE-MSG in Kommandoprozeduren, durch

- den Operanden *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- einen zusätzlichen Wert des Operanden MSG-ID.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

```
DELETE-MSG

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>

        *CLASS(...)

                |       MSG-CLASS = <name 3..3>

        *INTERVAL(...)

                |       FROM = <name 7..7>

                |       ,TO = <name 7..7>

,LANGUAGES = *ALL / list-poss(8): <name 1..1>
```

Operanden

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
Bezeichnet eine oder mehrere Meldungseinheiten, die verändert werden sollen.

MSG-ID = *ALL
Alle Meldungseinheiten der aktuellen Meldungsdatei werden gelöscht.

MSG-ID = *CLASS(...)
Jede Meldungseinheit, deren Meldungsschlüssel mit der angegebenen Meldungklasse beginnt, wird gelöscht.

MSG-CLASS = <name 3..3>
Gibt die dreistellige Meldungsklasse an.

MSG-ID = *INTERVAL(...)
Jede Meldungseinheit, deren Meldungsschlüssel im definierten Meldungsintervall liegt, wird gelöscht. Die angegebenen Intervallgrenzen müssen einer Meldungsklasse angehören.

FROM = <name 7..7>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls.

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssel des Meldungsintervalls.

MSG-ID = list-poss(2000): <name 7..7>
Die zu löschende Meldungseinheit wird durch einen vollständigen Meldungsschlüssel beschrieben. Mehrere Meldungsschlüssel können als Liste angegeben werden.

LANGUAGES = *ALL / list-poss(8): <name 1..1>
Bezeichnet die Kennbuchstaben der definierten Sprachen. Meldungstexte sowie Bedeutungs- und Maßnahmetexte in den angegebenen Sprachen werden gelöscht.

LANGUAGES = *ALL
Die Meldungseinheit wird vollständig, einschließlich aller Meldungsattribute, Insertattribute und Texte gelöscht.

LANGUAGES = list-poss(8): <name 8..8>
Es können maximal acht Kennbuchstaben angegeben werden; die Texte in den entsprechenden Sprachen werden gelöscht.

Beispiel

```
//DELETE-MSG MSG-ID=*CLASS(TST),LANGUAGE=E
```

8.3.2.6 END - MSGMAKER beenden

Funktion

Die Anweisung //END beendet MSGMAKER.

Unterschiede zur Anweisung //END im Menümodus

Die Anweisung //END kann ohne Unterschiede im Kommandobereich der Bildschirmmaske und in Kommandoprozeduren eingesetzt werden.

Wird MSGMAKER jedoch im Menümodus mit der Anweisung //END beendet, ist zu beachten, dass die aktuelle Bildschirmfunktion abgebrochen wird.

Format

END

8.3.2.7 MERGE-MSG-FILES - Meldungsdateien mischen

Funktion

Mit der Anweisung //MERGE-MSG-FILES können die Gesamtinhalte mehrerer Meldungsdateien in eine neue Meldungsdatei zusammengemischt werden, die leer oder nicht katalogisiert sein sollte. Gesamtinhalt bedeutet, dass alle Meldungen inklusive der Dokumentationszeilen in den Mischvorgang miteinbezogen werden. Über den Operanden LANGUAGES kann das Mischen auf Meldungen bestimmter Sprachen beschränkt werden.

Besonderheiten des Mischvorgangs:

Existieren in den Meldungsdateien Meldungen mit identischen Meldungsschlüsseln, erkennt MSGMAKER diese Übereinstimmung. Es erscheint folgende Meldung:

```
% MSMQI00 IDENTICAL OBJECTS IN FILES 1='filename 1' AND 2='filename 2'.
  PRIORITY? REPLY (1;2;T=TERMINATE).
```

In dieser Abfrage entscheidet der Anwender, welche Meldungsdatei bezüglich der Meldungsschlüssel mit Priorität behandelt werden soll. Jede auftretende Übereinstimmung wird auf diese Weise geprüft; die Abfrage erfolgt auch, wenn Meldungen aus verschiedenen Dateien mit demselben Meldungsschlüssel aber in unterschiedlichen Sprachen abgespeichert sind.
Die identischen Meldungen der nicht-priorisierten Datei werden nicht in die Zieldatei übernommen.
In Batch-Aufträgen und Prozeduren erhält die im Operanden FILE-NAMES zuerst genannte Meldungsdatei die Priorität, falls eine Übereinstimmung der Meldungsschlüssel auftreten sollte. Eine Meldung informiert den Anwender über dieses Ereignis.

Für identische Dokumentationszeilen wird dasselbe Prüfverfahren durchgeführt und anschließend nach der vergebenen Priorität entschieden.

> ¡ Soll der Gesamtinhalt mehrerer Meldungsdateien in eine neue Meldungsdatei zusammengemischt werden oder soll eine Meldungsdatei in eine andere Datei kopiert werden, ist die Anweisung //MERGE-MSG-FILE immer der COPY-Funktion vorzuziehen. Die zusammengemischte Meldungsdatei benötigt überdies weniger Speicherplatz, als eine mit COPY erzeugte Datei.
> Es steht keine Bildschirmmaske zur Verfügung, die der Anweisung //MERGE-MSG-FILES entspricht. Die Anweisung kann jedoch in die Kommandozeile der Maske eingetragen und mit DUE oder F2 gestartet werden.

Format

MERGE-MSG-FILES
FILE-NAMES = list-poss(2000): *CURRENT / <filename 1..80 with-wild> /   <partial-filename 2..79 with-wild> ,LANGUAGES = *ALL / list-poss(8): <name 1..1> ,TO-FILE = *CURRENT / <filename 1..54>

Operanden

FILE-NAMES = list-poss(2000): *CURRENT / <filename 1..80 with-wild> /<partial-filename 2..79 with-wild>
Bezeichnet eine oder mehrere Meldungsdateien, die zusammengemischt werden sollen. Wird nur ein Name angegeben, ist dies gleichbedeutend mit einer Kopie dieser Datei in die mit TO-FILE bezeichnete Datei.

Mit *CURRENT wird die mit der Anweisung //OPEN-MSG-FILE zuletzt geöffnete Datei angegeben.

Wird ein Dateiname mehrmals genannt, so werden alle Nennungen bis auf die erste von MSGMAKER ignoriert.

LANGUAGES = *ALL / list-poss(8): <name 1..1>
Bezeichnet die Sprachen, in denen die Meldungen abgespeichert sind.

LANGUAGES = *ALL
Die Meldungsdatei soll vollständig, d. h. alle Meldungen in allen Sprachen, für den Mischvorgang verwendet werden.

LANGUAGES = list-poss(8): <name 1..1>
Von der Meldungsdatei sollen nur Meldungen in den angegebenen Sprachen für den Mischvorgang verwendet werden.

TO-FILE = *CURRENT / <filename 1..54>
Ausgabedatei, in der die Inhalte der Eingabedateien zusammengemischt werden.*CURRENT bezeichnet die in MSGMAKER zuletzt geöffnete Datei.

Die Ausgabedatei muss leer oder noch nicht katalogisiert sein. Für die Dateiattribute gelten folgende Regeln:

- Der Dateityp der Ausgabedatei richtet sich nach dem Dateityp der Eingabedatei. Ist mindestens eine Eingabedatei vom Typ Standard, so erhält auch die Ausgabedatei diesen Dateityp.
  Eingabedateien vom Typ Customer ergeben auch wieder eine Ausgabedatei vom Typ Customer. Wird der Ausgabedatei mit der Anweisung //OPEN-MSG-FILE der Typ Customer zugewiesen und ist mindestens eine Eingabedatei vom Typ Standard, so gibt MSGMAKER die Warnung `MSMWJ01` aus.
- Gibt es mehrere Eingabedateien, werden die Dateiattribute Produktname und Produktversion der Ausgabedatei nicht bestimmt.
  Existiert nur eine Eingabedatei, werden die Dateiattribute dieser Datei in die Ausgabedatei übertragen. In diesem Fall ist die Ausgabedatei die Kopie der Eingabedatei.

Beispiel

```
//OPEN-MSG-FILE FILE-NAME=SYSMES.EKP.112, MODE=CREATE(TYPE=STANDARD,
      PRODUCT=BS2000(V200))
//MERGE-MSG-FILES FILE-NAMES=(SYSMES.INT*.200,SYSMES.COMP*.///,
      SYSMES.SPECIAL.123), TO-FILE=*CURRENT
```

8.3.2.8 MODIFY-DOCUMENTATION - Verändern und Löschen von Dokumentationszeilen

Funktion

Mit der Anweisung //MODIFY-DOCUMENTATION können dokumentierte Meldungen folgenderweise bearbeitet werden:

- Ändern von Dokumentationszeilen
- Löschen von Dokumentationsangaben (nur die variablen Angaben werden gelöscht)

Die dokumentierten Meldungen werden über den Meldungsschlüssel ausgewählt.

Unterschiede zur Anweisung //MODIFY-DOCUMENTATION im Menümodus

Die Anweisung //MODIFY-DOCUMENTATION, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //MODIFY-DOCUMENTATION in Kommandoprozeduren, durch den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

```
MODIFY-DOCUMENTATION

MSG-ID = *INTERVAL(...) / <name 7..7>

    *INTERVAL(...)

        |   FROM = <name 7..7>

        |   ,TO = <name 7..7>

,OWNER = *UNCHANGED / *NONE / *PARAMETERS(...)

    *PARAMETERS(...)

        |   NAME = *UNCHANGED / *NONE / <c-string 1..20>

        |   ,TEAM = *UNCHANGED / *NONE / <c-string 1..15>

,COMMENTS = *UNCHANGED / *NONE / <c-string 1..60>
```

Operanden

MSG-ID = *INTERVAL(...) / <name 7..7>
Bezeichnet eine oder mehrere Meldungseinheiten, die verändert werden sollen.

MSG-ID = <name 7..7>
Der vollständige Meldungsschlüssel, bestehend aus der dreistelligen Meldungsklasse und der vierstelligen Meldungsnummer, muss eingegeben werden.

MSG-ID = *INTERVAL(...)
Bezeichnet ein Meldungsintervall. Die Grenzen des Intervalls müssen einer Meldungsklasse angehören.

FROM = <name 7..7>
Bezeichnet die untere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

TO = <name 7..7>

Bezeichnet die obere Grenze des Meldungsintervalls (vollständiger Meldungsschlüssel).

OWNER = *UNCHANGED / *NONE / *PARAMETERS(...)
Bezeichnet die Personen, die für die Meldungsdatei verantwortlich sind.

OWNER = *UNCHANGED
Die Angaben zu den verantwortlichen Personen werden nicht verändert.

OWNER = *NONE
Er werden keine Angaben zu den verantwortlichen Personen gemacht. Bestehende Angaben werden gelöscht.

OWNER = *PARAMETERS(...)

NAME = *UNCHANGED / *NONE / <c-string 1..20>
Bezeichnet die Person, die für die Meldungsdatei verantwortlich ist.

NAME = *UNCHANGED
Die Angaben zur Person werden nicht verändert.

NAME = *NONE
Die Angaben zur Person werden gelöscht.

NAME = <c-string 1..20>
Die Angaben zur Person werden verändert.

TEAM = *UNCHANGED / *NONE / <c-string 1..15>
Bezeichnet das Team (Abteilung), das für die Meldungsdatei verantwortlich ist.

TEAM = *UNCHANGED
Die Angaben zum Team werden nicht verändert.

TEAM = *NONE
Die Angaben zum Team werden gelöscht.

TEAM = <c-string 1..15>
Die Angaben zum Team werden verändert.

COMMENTS = *UNCHANGED / *NONE / <c-string 1..60>
Bezeichnet den Kommentar zu den ausgewählten Meldungen.

COMMENTS = *UNCHANGED
Der Kommentar wird nicht verändert.

COMMENTS = *NONE
Der Kommentar wird gelöscht.

COMMENTS = <c-string 1..60>
Der Kommentar wird verändert.

Beispiel

Die Meldungsintervalle (AAA0001-AAA0010) und (AAA0021-AAA0030) werden folgendermaßen dokumentiert:

```
//ADD-DOCUMENTATION -
   MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0010),-
   OWNER=*PARAMETERS(NAME='John Miller',TEAM='ABC 0001'),-
   COMMENTS='messages for test product'
//ADD-DOCUMENTATION -
   MSG-ID=*INTERVAL(FROM=AAA0021,TO=AAA0030),-
   OWNER=*PARAMETERS(TEAM='ABC')
```

Ausgabe nach SYSOUT oder SYSLST:

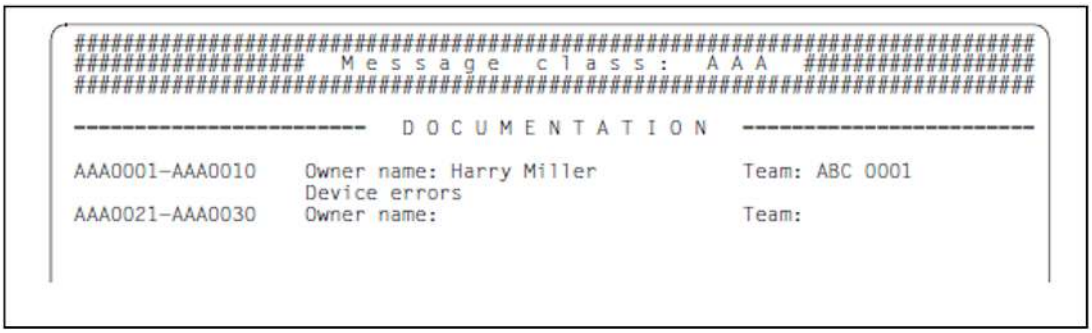
```
################################################################################
##################  M e s s a g e  c l a s s :  A A A  ##################
################################################################################

-----------------------  D O C U M E N T A T I O N  -----------------------

AAA0001-AAA0010    Owner name: John Miller             Team: ABC 0001
                   Messages for test product
AAA0021-AAA0030    Owner name:                         Team: ABC
```

Die Dokumentation soll nun verändert werden:

- AAA0001-AAA0010:
  Änderung des Namens in Harry Miller;
  Änderung des Kommentars in „device errors“
- AAA0021-AAA0030:
  Löschen der Angabe zum Team

```
//MODIFY-DOCUMENTATION -
   MSG-ID=*INTERVAL(FROM=AAA0001,TO=AAA0010),-
   OWNER=*PARAMETERS(NAME='Harry Miller',TEAM=*UNCHANGED),-
   COMMENTS='device errors'
//MODIFY-DOCUMENTATION -
   MSG-ID=*INTERVAL(FROM=AAA0021,TO=AAA0030),-
   OWNER=*PARAMETERS(TEAM=*NONE)
```

Ausgabe nach SYSOUT oder SYSLST:


Message class: AAA
DOCUMENTATION
AAA0001–AAA0010 Owner name: Harry Miller Team: ABC 0001
Device errors
AAA0021–AAA0030 Owner name: Team:

8.3.2.9 MODIFY-MSG - Meldungseinheit verändern

Funktion

Mit der Anweisung //MODIFY-MSG wird eine Meldungseinheit der aktuell geöffneten Meldungsdatei verändert. Durch die Angabe eines Meldungsschlüssels oder eines Meldungsbereichs kann auf eine oder mehrere Meldungseinheiten zugegriffen werden. Neben den Meldungsattributen können Meldungs-, Bedeutungs- und Maßnahmetext, sowie Inserts und deren Attribute verändert werden. Eine Veränderung der Meldungseinheit bedeutet nicht nur eine Änderung bereits bestehender Operandenwerte, sondern beinhaltet ebenfalls die Möglichkeit, den Operanden neue Werte zuzuweisen oder Operandenwerte zu löschen.

Unterschiede zur Anweisung //MODIFY-MSG im Menümodus

Die Anweisung //MODIFY-MSG, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //MODIFY-MSG in Kommandoprozeduren, durch

- den Operanden *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- einen zusätzlichen Wert des Operanden MSG-ID
- einen zusätzlichen Wert des Operanden LANGUAGES

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

MODIFY-MSG

```
MSG-ID = *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
    *CLASS(...)
        | MSG-CLASS = <name 3..3>
    *INTERVAL(...)
        | FROM = <name 7..7>
        | ,TO = <name 7..7>
,ACCESS-METHODS = *UNCHANGED / list-poss(4): *ISAM / *DLAM / *LOCAL-DLAM / *MINIMIP / *BAMR
,DESTINATIONS = *UNCHANGED (...) / *ALL(...) / list-poss(2): *USER-TASK / *CONSOLE(...)
    *UNCHANGED(...)
        | ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG
    *ALL(...)
        | ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG
    CONSOLE(...)
        | ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG
,WEIGHT = *UNCHANGED / *NONE / <integer 0..99>
```

,WARRANTY = *UNCHANGED / *NO / *YES

,MSG-TEXT-OUTPUT = *UNCHANGED / *UPPER-CASE / *LOWER-CASE

```
,LANGUAGES = *UNCHANCHED / list-poss(8): <name 1..1>(...)
    <name>(...)
        |   MSG-TEXT = *UNCHANGED / <c-string 1..220 with-low>
        |   ,MEANING = *UNCHANGED / list-poss(2000): *ADD(...) / *INSERT(...) /
        |                          *REPLACE(...) / *REMOVE(...)
        |       *ADD(...)
        |           |   TEXT = list-poss(2000): <c-string 1..74 with-low>
        |       *INSERT(...)
        |           |   LINE-NUMBER = <integer 1..256>
        |           |   ,TEXT = list-poss(2000): <c-string 1..74 with-low>
        |       *REPLACE(...)
        |           |   LINE-NUMBER = <integer 1..256>
        |           |   ,TEXT = <c-string 1..74 with-low>
        |       *REMOVE(...)
        |           |   LINE-NUMBERS = *ALL / list-poss(2000): <integer 1..256>
        |   ,RESPONSE = *UNCHANGED / list-poss(2000): *ADD(...) / *INSERT(...) /
        |                            *REPLACE(...) / *REMOVE(...)
        |       *ADD(...)
        |           |   TEXT = list-poss(2000): <c-string 1..74 with-low>
        |       *INSERT(...)
        |           |   LINE-NUMBER = <integer 1..256>
        |           |   ,TEXT = list-poss(2000): <c-string 1..74 with-low>
        |       *REPLACE(...)
        |           |   LINE-NUMBER = <integer 1..256>
        |           |   ,TEXT = <c-string 1..74 with-low>
        |       *REMOVE(...)
        |           |   LINE-NUMBERS = *ALL / list-poss(2000): <integer 1..256>
```

```
,INSERT-ATTRIBUTES = *UNCHANGED / *NONE / list-poss(30): <integer 0..29>(...)

    <integer 0..29>(...)

        |   NAME = *UNCHANGED / *NONE / <structured-name 1..20>

        |   ,DEFAULT-VALUE = *UNCHANGED / *NONE / <c-string 1..54 with-low> / *EMPTY-STRING

        |   ,AUTOMATIC-HELP = *UNCHANGED / *NO / *YES(...)

        |           *YES(...)

        |                   |       PREFIX = *BY-INSERT-VALUE / <name 3...7>
```

Operanden

MSG-ID = *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
Bezeichnet die Meldungsschlüssel einer oder mehrerer Meldungseinheiten der eröffneten Meldungsdatei, die verändert werden sollen.

MSG-ID = *CLASS(...)
Jede Meldungseinheit, deren Meldungsschlüssel mit der angegebenen Meldungsklasse beginnt, kann verändert werden.

MSG-CLASS = <name 3..3>
Kennzeichnet die dreistellige Meldungsklasse.

MSG-ID = *INTERVAL(...)
Jede Meldungseinheit innerhalb dieses Meldungsintervalls wird verändert.

FROM = <name 7..7>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls.

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssels des Meldungsintervalls.

MSG-ID = list-poss(2000): <name 7..7>
Gibt den vollständigen Meldungsschlüssel einer Meldungseinheit an, die verändert werden soll. Mehrere Meldungsschlüssel können als Liste angegeben werden.

ACCESS-METHODS = *UNCHANGED / list-poss(4): *ISAM / *DLAM / *LOCAL-DLAM / *MINIMIP / *BAMR
Eine neue MIP-Zugriffsmethode für die Meldungen kann vereinbart werden. Alle früheren Vereinbarungen werden bei einer Änderung gelöscht.

ACCESS-METHODS = *UNCHANGED
Die Zugriffsmethoden werden nicht verändert.

ACCESS-METHODS = *ISAM
Die Meldungen werden über den ISAM-Schlüssel gesucht.

ACCESS-METHODS = *DLAM
Die Zugriffsmethode DLAM ist für besonders häufig auftretende Meldungen vorgesehen. Wird eine Meldungsdatei, die eine DLAM-Meldung enthält, aktiviert, wird diese in den Arbeitsspeicher geladen. Die DLAM-Meldung kann von MIP direkt, ohne Zugriff auf die Meldungsdatei, ausgegeben werden.

ACCESS-METHODS = *LOCAL-DLAM / *MINIMIP / *BAMR
Diese Zugriffsmethoden sind für den internen Gebrauch beim Hersteller reserviert (siehe DHB-Abschnitt MIP).

DESTINATIONS = *UNCHANGED(...) / *ALL(...) / list-poss(2): *USER-TASK / *CONSOLE(...)
Dieser Operand dokumentiert die neuen, möglichen Ausgabeziele der Meldung aus der Sicht des Meldungsverfassers. Die Angabe dient nur Dokumentationszwecken. Das Ausgabeziel selbst wird im Operanden DEST (Destination code) des Makros MSG7X festgelegt.

DESTINATIONS = *UNCHANGED(...)
Das Ausgabeziel der Meldung bleibt unverändert.

ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG
Der einstellige Berechtigungsschlüssel wird bei Konsolausgaben als Zielangabe ausgewertet. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

ROUTING-CODE = *UNCHANGED
Der Berechtigungsschlüssel bleibt unverändert.

ROUTING-CODE = <alphanum-name 1..1>
Als Berechtigungsschlüssel kann jeder Buchstabe, jede Zahl oder die Sonderzeichen #, $, oder @, mit Ausnahme von * verwendet werden. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

ROUTING-CODE = *MAIN-CONSOLE
Ziel der Meldung ist der besondere Berechtigungsschlüssel *, der stets zumindest der Hauptkonsole zugeordnet ist.

ROUTING-CODE = *CONSLOG
Meldungen, die keine Antwort erfordern, werden nur in der CONSLOG-Datei protokolliert. ROUTING-CODE = *CONSLOG entspricht der Zuweisung ROUTING-CODE = @.

DESTINATIONS = *ALL(...)
Alle Ausgabeziele sind für die Meldungen möglich.

ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG
Der einstellige Berechtigungsschlüssel wird bei Konsolausgaben als Zielangabe ausgewertet. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

ROUTING-CODE = *UNCHANGED
Der Berechtigungsschlüssel bleibt unverändert.

ROUTING-CODE = <alphanum-name 1..1>
Als Berechtigungsschlüssel kann jeder Buchstabe, jede Zahl oder die Sonderzeichen #, $, oder @, mit Ausnahme von * verwendet werden. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

ROUTING-CODE = *MAIN-CONSOLE
Ziel der Meldung ist der besondere Berechtigungsschlüssel *, der stets zumindest der Hauptkonsole zugeordnet ist.

ROUTING-CODE = *CONSLOG
Meldungen, die keine Antwort erfordern, werden nur in der CONSLOG-Datei protokolliert. ROUTING-CODE = *CONSLOG entspricht der Zuweisung ROUTING-CODE = @.

DESTINATIONS = *USER-TASK
Die Meldung hat als neues Ausgabeziel SYSOUT, SYSLST oder einen benutzereigenen Speicherbereich.

DESTINATIONS = *CONSOLE(...)
Das Ausgabeziel der Meldung ist eine Konsole. Wird CONSOLE als neuer Operandenwert vereinbart, **muss** ein Berechtigungsschlüssel definiert werden. Wird CONSOLE durch andere Ausgabeziele ersetzt, wird der zugehörige Berechtigungsschlüssel automatisch unterdrückt.

> **ROUTING-CODE = *UNCHANGED / <alphanum-name 1..1> / *MAIN-CONSOLE / *CONSLOG**
> Der einstellige Berechtigungsschlüssel wird bei Konsolausgaben als Zielangabe ausgewertet. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].
>
> **ROUTING-CODE = *UNCHANGED**
> Der Berechtigungsschlüssel bleibt unverändert.
>
> **ROUTING-CODE = <alphanum-name 1..1>**
> Als Berechtigungsschlüssel kann jeder Buchstabe, jede Zahl oder die Sonderzeichen #, $, oder @, mit Ausnahme von * verwendet werden. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].
>
> **ROUTING-CODE = *MAIN-CONSOLE**
> Ziel der Meldung ist der besondere Berechtigungsschlüssel *, der stets zumindest der Hauptkonsole zugeordnet ist.
>
> **ROUTING-CODE = *CONSLOG**
> Meldungen, die keine Antwort erfordern, werden nur in der CONSLOG-Datei protokolliert. ROUTING-CODE = *CONSLOG entspricht der Zuweisung ROUTING-CODE = @.

WEIGHT = *UNCHANGED / *NONE / <integer 0..99>
Der Weight Code bezeichnet das Meldungsgewicht. Ein Gewicht muss für eine Meldung vereinbart werden, deren Ausgabeziel eine Konsole ist (Zuweisung
DESTINATION = *CONSOLE(...)/*ALL) .
Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

WEIGHT = *UNCHANGED
Der Weight Code bleibt unverändert.

WEIGHT = *NONE
Die Meldung erhält kein Gewicht.

WEIGHT = <integer 0..99>
Der Meldung wird ein Wert von 0 bis 99 zugewiesen, wobei 99 die höchste Priorität für eine Meldung darstellt.

WARRANTY = *UNCHANGED / *NO / *YES
Das Meldungsattribut „Garantie“ wird von MIP ausgewertet.

Folgende Meldungs-Bestandteile werden garantiert:

- Meldungsschlüssel
- Nummerierung und Bedeutung der Inserts

Der Meldungstext wird **nicht** garantiert.
MIP erzeugt für garantierte Meldungen S-Variablen. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

MSG-TEXT-OUTPUT = *UNCHANGED / *UPPER-CASE / *LOWER-CASE
Bezeichnet die Schreibweise des Meldungstextes bei der Ausgabe nach SYSOUT, in eine SYSLST-Datei, an eine Konsole oder in einen benutzereigenen Speicher durch den Makro MSG7X oder das Kommando /HELP-MSG-INFORMATION. Die Default-Texte für Inserts sowie die Texte, die über den Makro MSG7X vorgegeben sind, werden ebenfalls an dieses Format angepasst.

MSG-TEXT-OUTPUT = *UNCHANGED
Die Schreibweise des Meldungstextes wird nicht verändert.

MSG-TEXT-OUTPUT = *UPPER-CASE
Der eingetragene Meldungstext wird in Großbuchstaben ausgegeben.

MSG-TEXT-OUTPUT = *LOWER-CASE
Der Meldungstext wird gemäß seiner Schreibweise im Operanden LANGUAGES = ...(MSG-TEXT = '...') ausgegeben.

LANGUAGES = *UNCHANGED / list-poss(8): <name 1..1>(...)
Bezeichnet Kennbuchstaben bereits definierter Sprachen. Meldungs-, Bedeutungs- und Maßnahmetexte in diesen Sprachen können verändert werden.

LANGUAGES = *UNCHANGED
Es werden keine Kennnbuchstaben angegeben; die Meldungs-, Bedeutungs- und Maßnahmetexte sollen nicht verändert werden.

LANGUAGES = list-poss(8): <name 1..1>(...)
Die Meldungs-, Bedeutungs- und Maßnahmetexte in den durch die Kennnbuchstaben bezeichneten Sprachen sollen verändert werden.

> **MSG-TEXT = *UNCHANGED / <c-string 1..220 with-low>**
> Bezeichnet den Meldungstext, welcher in der angegebenen Sprache verfasst wurde.
>
> **MSG-TEXT = *UNCHANGED**
> Der Meldungstext bleibt unverändert.
>
> **MSG-TEXT = <c-string 1..220 with-low>**
> Bezeichnet einen neuen Meldungstext für die angegebene Sprache.
>
> **MEANING = *UNCHANGED / list-poss: *ADD(...) / *INSERT(...) / *REPLACE(...) / *REMOVE(...)**
> Bezeichnet den Bedeutungstext, welcher in der angegebenen Sprache verfasst wurde und durch das Kommando /HELP-MSG-INFORMATION zur Anzeige kommt.
> Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].
> Jede der insgesamt 256 Zeilen des Bedeutungstextes kann verändert oder gelöscht werden, des weiteren können neue Zeile angefügt oder eingefügt werden. Die Zeilen des Bedeutungstextes sind intern von 1 bis maximal 256 durchnummeriert und können auf diese Weise einzeln angesprochen werden.
>
> **MEANING = *UNCHANGED**
> Der Bedeutungstext wird nicht verändert.
>
> **MEANING = *ADD(...)**
> Am Ende des Bedeutungstextes können eine oder mehrere neue Zeilen angefügt werden.
>
> > **TEXT = list-poss: <c-string 1..74 with-low>**
> > Bedeutungstext, der in einer oder mehreren Zeilen steht und an den bestehenden Text angefügt werden soll.

MEANING = *INSERT(...)
In den Bedeutungstext können eine oder mehrere neue Zeilen eingefügt werden. Alle folgenden Zeilen werden nach Abschluss aller Veränderungen neu durchnummeriert.

LINE-NUMBER = <integer 1..256>
Der neue Text wird zwischen der hier angegebenen Zeilennummer und der vorhergehenden Zeilennummer eingefügt.

TEXT = list-poss: <c-string 1..74 with-low>
Bezeichnet den neuen Bedeutungstext.

MEANING = *REPLACE(...)
Der Bedeutungstext in der angegebenen Zeile wird durch den eingetragenen Text ersetzt.

LINE-NUMBER = <integer 1..256>
Bezeichnet die Zeilennummer, in welcher der Text steht, der ersetzt werden soll.

TEXT = <c-string 1..74 with-low>
Bezeichnet den Text, welcher den Bedeutungstext in der angegebenen Zeile ersetzen soll.

MEANING = *REMOVE(...)
Zeilen des Bedeutungstextes können entfernt werden. Alle folgenden Zeilen werden nach Abschluss aller Änderungen verschoben und neu durchnummeriert.

LINE-NUMBERS = *ALL / list-poss: <integer 1..256>
Kennzeichnet die Zeilennummern.

Beispiele zum Operanden MEANING

Der Bedeutungstext sei:
Zeile 1: `'texttext 1'`
Zeile 2: `'texttext 2'`
Zeile 3: `'texttext 3'`

Beispiel 1

In den oben genannten Bedeutungstext wird eine neue Zeile eingefügt, eine Zeile durch eine andere ersetzt und eine Zeile am Ende des Bedeutungstextes angefügt.

```
MEANING=(*INSERT(2,'texttext 1 ext'),*REPLACE(2,'texttext 2 new'),
         *ADD('texttext 4')
```

Bedeutungstext, mit neuer Zeilennummerierung:
Zeile 1: `'texttext 1'`
Zeile 2: `'texttext 1 ext'`
Zeile 3: `'texttext 2 new'`
Zeile 4: `'texttext 3'`
Zeile 5: `'texttext 4'`

Beispiel 2

Im ursprünglichen Bedeutungstext wird die erste Zeile gelöscht und die zweite Zeile durch eine neue Zeile ersetzt.

```
MEANING=(*REMOVE(1),*REPLACE(2,'texttext' 2 new) )
```

Bedeutungstext, mit neuer Zeilennummerierung:
Zeile 1: `'texttext 2 new'`
Zeile 2: `'texttext 3'`

RESPONSE = *UNCHANGED / list-poss: *ADD(...) / *INSERT(...) / *REPLACE(...) / *REMOVE(...)

Bezeichnet den Maßnahmetext, welcher in der angegebenen Sprache verfasst wurde und durch das Kommando /HELP-MSG-INFORMATION zur Anzeige kommt. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].
Jede der insgesamt 256 Zeilen des Maßnahmetexts kann verändert oder gelöscht werden. Weiterhin können neue Zeilen angefügt oder eingefügt werden. Die Zeilen des Maßnahmetexts sind intern von 1 bis maximal 256 durchnummeriert und können auf diese Weise einzeln angesprochen werden. Beispiele siehe Operand MEANING.

RESPONSE = *UNCHANGED
Der Maßnahmetext wird nicht verändert.

RESPONSE = *ADD(...)
Am Ende des Maßnahmetexts können eine oder mehrere neue Zeilen angefügt werden.

TEXT = list-poss: <c-string 1..74 with-low>
Maßnahmetext, der in einer oder mehreren Zeilen steht und an den bestehenden Text angefügt werden soll.

RESPONSE = *INSERT(...)
In den Maßnahmetext können eine oder mehrere neue Zeilen eingefügt werden.
Alle folgenden Zeilen werden nach Abschluss aller Änderungen neu durchnummeriert.

LINE-NUMBER = <integer 1..256>
Der neue Text wird zwischen der hier angegebenen Zeilennummer und der vorhergehenden Zeilennummer eingefügt.

TEXT = list-poss: <c-string 1..74 with-low>
Bezeichnet den neuen Maßnahmetext.

RESPONSE = *REPLACE(...)
Der Maßnahmetext in der angegebenen Zeile wird durch den eingetragenen Text ersetzt.

LINE-NUMBER = <integer 1..256>
Bezeichnet die Zeilennummer, in welcher der Text steht, der ersetzt werden soll.

TEXT = <c-string 1..74 with-low>
Bezeichnet den Text, welcher den Maßnahmetext in der angegebenen Zeile ersetzen soll.

RESPONSE = *REMOVE(...)
Zeilen des Maßnahmetextes können entfernt werden. Alle folgenden Zeilen werden nach Abschluss aller Änderungen verschoben und neu durchnummeriert.

LINE-NUMBERS = *ALL / list-poss: <integer 1..256>
Kennzeichnet die Zeilennummern.

INSERT-ATTRIBUTES = *UNCHANGED / *NONE / list-poss(30): <integer 0..29>(...)
Die Attribute der angegebenen Inserts werden hinzugefügt, verändert oder gelöscht.

INSERT-ATTRIBUTES = *UNCHANGED
Die Attribute der definierten Inserts werden nicht verändert.

INSERT-ATTRIBUTES = *NONE
Alle Attribute der definierten Inserts werden gelöscht.

INSERT-ATTRIBUTES = <integer 0..29>(...)
Die Attribute der angegebenen Inserts (Nummern von 00 bis 29) können hinzugefügt, verändert oder gelöscht werden, siehe auch Anweisung //ADD-MSG.

NAME = *UNCHANGED / *NONE / <structured-name 1..20>
Der Namen des Inserts soll verändert werden. Insertnamen werden von MIP ausgewertet.

DEFAULT-VALUE = *UNCHANGED / *NONE / <c-string 1..54 with-low> / *EMPTY-STRING
Änderung des voreingestellten Textes, der anstelle des Inserts in den Meldungstext eingefügt wird, falls im Makro MSG7X kein aktueller Text vereinbart wurde.

DEFAULT-VALUE = *UNCHANGED
Der voreingestellte Text wird nicht verändert.

DEFAULT-VALUE = *NONE
Der voreingestellte Text wird gelöscht.

Beispiel

```
INSERT-ATTRIBUTES=(0(NAME=*IDENTIFIER),4(NAME=*NONE),
2(DEFAULT-VALUE=*NONE),1(AUTO-HELP=*YES(CMD)))
```

DEFAULT-VALUE = <c-string 1..54 with-low>
Bezeichnet einen neuen Inserttext. Die Längenbeschränkungen bei Meldungstexten sind zu beachten. Nähere Informationen hierzu finden Sie im Handbuch „Systembetreuung“ [5].

DEFAULT-VALUE = *EMPTY-STRING
Vereinbart einen Leerstring. Aus Kompatibilitätsgründen wird der Operandenwert DEFAULT-VALUE = *NULL in Batch-Aufträgen und Prozeduren weiterhin unterstützt.

Beispiel

```
INSERT-ATTRIBUTES=(0(DEFAULT-VALUE = *EMPTY-STRING),
1(DEFAULT-VALUE ='$TSOS'),4(DEFAULT-VALUE=*NONE))
```

AUTOMATIC-HELP = *UNCHANGED / *NO / *YES(...)
Wird über ein Insert ein Meldungsschlüssel oder ein Teil davon ausgegeben (z.B. bei DVS-Fehlern wird lediglich die Meldungsnummer ausgegeben), so wird über die automatische Hilfsfunktion MIP dazu veranlaßt, zusätzlich zu diesem Meldungsschlüssel den zugehörigen Meldungstext auszugeben, siehe "ADD-MSG - Einfügen einer Meldungseinheit".

AUTOMATIC-HELP = *NO
Es wird nur der Meldungsschlüssel ausgegeben.

AUTOMATIC-HELP = *YES(...)
Zum Meldungsschlüssel wird der Meldungstext ausgegeben

PREFIX = *BY-INSERT-VALUE / <name 3...7>
Zu einem in der Fehlermeldung ausgegebenen Insert oder zu einer explizit angegebenen Meldungsnummer wird der gesamte Meldungstext ausgegeben.

8.3.2.10 MODIFY-OPTION - Überschreiben einer Meldungseinheit

Funktion

Die Anweisung //MODIFY-OPTION ist nur in Batch-Aufträgen und Prozeduren verfügbar.

Mit der Anweisung //MODIFY-OPTION wird beim Start von MSGMAKER festgelegt, ob (Teile von) Meldungseinheiten überschrieben werden dürfen. Diese Einstellung ist für alle nachfolgenden Anweisungen //ADD-DOCUMENTATION, //ADD-MSG, //COPY und
//MOVE solange gültig, bis eine neue Anweisung //MODIFY-OPTION eingegeben wird.

Format

MODIFY-OPTION
OVERWRITE = *UNCHANGED / *YES / *NO

Operanden

OVERWRITE = *UNCHANGED / *YES / *NO
Gibt an, ob (Teile von) Meldungseinheiten oder Dokumentationszeilen überschrieben werden dürfen. Bei Programmstart wird dem Operanden OVERWRITE der Wert *NO zugewiesen.

8.3.2.11 MOVE - Meldungseinheit kopieren und löschen

Funktion

Mit der Anweisung //MOVE können Meldungseinheiten mit oder ohne den zugehörigen Dokumentationszeilen von einer Meldungsdatei in eine andere oder innerhalb derselben Datei kopiert werden. Im Unterschied zur COPY-Funktion wird der Sendebereich bei der MOVE-Funktion gelöscht.

Jede Meldungseinheit ist durch ihren Meldungsschlüssel gekennzeichnet. Wird dem Operanden MSG-ID eine Meldungsklasse oder ein Meldungsintervall zugewiesen, können mehrere Meldungseinheiten in einem Vorgang übertragen werden. Wird dem Operanden INFORMATION der Wert DOCUMENTATION zugewiesen, werden die im angegebenen Meldungsbereich definierten Dokumentationszeilen ebenfalls übertragen.

Sind Sende- und Empfangsdatei des MOVE-Vorgangs identisch, müssen im Operanden TO-MSG-ID neue Meldungsschlüssel vereinbart werden.
Unabhängig von der aktuell geöffneten Datei, können Meldungseinheiten zwischen zwei weiteren Meldungsdateien übertragen werden. Die Operanden FROM-FILE und TO-FILE ermöglichen den Zugriff auf diese Dateien; die aktuelle Meldungsdatei bleibt geöffnet und wird nicht verändert, falls sie weder die Sende- noch die Empfangsdatei ist.

Sind im Operanden INFORMATION=MESSAGES(...) explizit Sprachen angegeben, werden nur die Texte (Meldungs-, Bedeutungs- und Maßnahmetext) in diesen Sprachen in die Empfangs-Meldungseinheit übertragen. Existiert diese Meldungseinheit nicht, wird eine Meldungseinheit mit den Meldungsattributen (Zugriffsmethode, Ausgabeziel, ...) der Sende-Meldungseinheit erzeugt. Ist in der Empfangs-Meldungseinheit bereits ein Text unter identischem Meldungsschlüssel und gleichem Kennbuchstaben für die Sprache abgespeichert, wird der Text der Empfangs-Meldungseinheit mit dem Text der Sende-Meldungseinheit überschrieben.
Im **Dialog** wird für diesen Fall eine Fehlermeldung ausgegeben mit der Frage, ob die bestehende Meldungseinheit überschrieben werden soll oder nicht.
In **Batch-Aufträgen** und **Prozeduren** werden die angegebenen Bestandteile der Meldungsdatei nicht übertragen und die Verarbeitung fortgesetzt, vorausgesetzt, in der Anweisung //MOVE oder //MODIFY-OPTION wurde nicht OVERWRITE=*YES vereinbart.

Unterscheiden sich die Texte nur durch die Sprache, wird der Text in der Empfangs-Meldungseinheit angefügt.

Sind alle sprachabhängigen Teile einer Meldungseinheit durch die Anweisung //MOVE aus einer Meldungseinheit entfernt worden, werden die verbliebenen Meldungsattribute, und somit die Meldungseinheit, automatisch gelöscht.

Unterschiede zur Anweisung //MOVE im Menümodus

Die Anweisung //MOVE, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //MOVE in Kommandoprozeduren, durch

- den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- einen zusätzlichen Wert des Operanden MSG-ID
- den Operanden OVERWRITE, der im Menümodus nicht verfügbar ist.

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

MOVE

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>

*CLASS(...)

| MSG-CLASS = <name 3..3>

*INTERVAL(...)

| FROM = <name 7..7>

| ,TO = <name 7..7>

,INFORMATION = *MESSAGES(...) / *ALL / list-poss(4): *DOCUMENTATION / *COMPONENT-ID /

*CORRECTION-INFO / *MESSAGES(...)

*MESSAGES(...)

| LANGUAGES = *ALL / list-poss(8): <name 1..1>

,FROM-FILE = *CURRENT / <filename 1..54>

,TO-MSG-ID = *SAME / *CLASS(...) / <alphanum-name 4..7 with-wild>

*CLASS(...)

| MSG-CLASS = <name 3..3>

,TO-FILE = *CURRENT / <filename 1..54>

,OVERWRITE = *STD / *YES / *NO

Operanden

MSG-ID = *ALL / *CLASS(...) / *INTERVAL(...) / list-poss(2000): <name 7..7>
Bezeichnet die Meldungsschlüssel einer oder mehrerer Meldungseinheiten der eröffneten Meldungsdatei, die kopiert und anschließend in der Sendedatei gelöscht werden sollen.Durch die Zuweisung INFORMATION=DOCUMENTATION können die zu den Meldungsschlüsseln definierten Dokumentationszeilen ebenfalls übertragen werden.

MSG-ID = *ALL
Alle im Operanden INFORMATION definierten Bestandteile einer Meldung werden übertragen. Diese Angabe ist nur zulässig, wenn dem Operanden TO-MSG-ID der Wert *SAME zugewiesen wird.

MSG-ID = *CLASS(...)
Alle Meldungsbestandteile der angegebenen Meldungsklasse werden übertragen.

MSG-CLASS = <name 3..3>
Gibt die dreistellige Meldungsklasse an.

MSG-ID = *INTERVAL(...)
Alle Meldungsbestandteile innerhalb dieses Meldungsintervalls werden übertragen. Für den Fall, dass das Meldungsintervall mehr als eine Meldungsklasse beinhaltet, muss dem Operanden TO-MSG-ID der Wert *SAME zugewiesen werden.

FROM = <name 7..7>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls.

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssel des Meldungsintervalls.

MSG-ID = list-poss(2000): <name 7..7>
Bezeichnet den vollständigen Meldungsschlüssel eines Meldungsbestandteils, das übertragen werden soll. Werden mehrere Meldungsschlüssel als Liste angegeben, muss dem Operanden TO-MSG-ID der Wert *SAME zugewiesen werden.
Mehrere Meldungsschlüssel können eine Liste bilden.

INFORMATION = *MESSAGES(...) / *ALL / list-poss(2000): *MESSAGES(...) /*DOCUMENTATION / *COMPONENT-ID / *CORRECTION-INFO
Bezeichnet die Meldungsbestandteile, die übertragen werden sollen.

INFORMATION = *MESSAGES(...)
Bezeichnet Meldungseinheiten, die übertragen werden sollen. Über den Operanden MSG-ID werden die Meldungseinheiten ausgewählt.

LANGUAGES = *ALL / list-poss(8): <name 1..1>
Gibt an, in welchen Sprachen der Meldungs-, Bedeutungs- und/oder Maßnahmetext übertragen werden soll.

LANGUAGES = *ALL
Die vollständige Meldungseinheit, einschließlich aller Meldungsattribute, Inserts und Insert-Attribute, sowie alle Texte werden übertragen.

LANGUAGES = list-poss(8): <name 1..1>
Der Meldungstext und/oder Bedeutungs-, Maßnahmetext in den ausgewählten Sprache(n) werden übertragen. Die Meldungsattribute und Insert-Attribute der Sende-Meldungseinheit werden nur in die Empfangs-Meldungseinheit übertragen, wenn letztere nicht existiert. Ansonsten wird die Meldung `MSME108` ausgegeben.

INFORMATION = *ALL
Alle Meldungsbestandteile (Meldungseinheiten und Dokumentationszeilen) werden übertragen. Wird dem Operanden MSG-ID = *ALL zugewiesen, wird der gesamte Inhalt der Meldungsdatei übertragen.

INFORMATION = *DOCUMENTATION
Alle Dokumentationszeilen, deren Meldungsschlüssel im ausgewählten Bereich (Operand MSG-ID) liegen, werden übertragen. Überschneiden mehrere definierte Meldungsintervalle den festgelegten Meldungsbereich, werden die Dokumentationszeilen dieser Intervalle ebenfalls übertragen.

Beispiel

MSG-ID = *INTERVAL (TST1500-TST1599).
Die Dokumentationszeilen der Meldungsintervalle (TST1500-TST1510), (TST1000-TST1999), (TST1000-TST1500), TST1510, ... werden übertragen.

MESSAGES = *CORRECTION-INFO / *COMPONENT-ID
Diese Operanden sind für den internen Gebrauch beim Hersteller reserviert.

FROM-FILE = *CURRENT / <filename 1..54 without-gen-vers>
Name der Meldungsdatei, aus welcher die bezeichneten Meldungsbestandteile (Operand MSG-ID) übertragen werden.

Der Operandenwert *CURRENT bezeichnet die aktuelle Meldungsdatei, die mit der Anweisung //OPEN-MSG-FILE zuletzt eröffnet wurde. Die Datei muss im UPDATE-Modus eröffnet worden sein.

TO-MSG-ID = *SAME / *CLASS(...) / <alphanum-name 4..7 with-wild>
Bezeichnet die neuen Meldungsschlüssel der übertragenen Meldungsbestandteile im Empfangsbereich.

TO-MSG-ID = *SAME
Die Meldungsbestandteile behalten beim Übertragungsvorgang ihren Meldungsschlüssel bei. Der Wert *SAME darf nicht angegeben werden, wenn Meldungsbestandteile innerhalb derselben Meldungsdatei übertragen werden. Der Wert *SAME muss angegeben werden, wenn dem Operanden MSG-ID eine Liste von Meldungsschlüsseln, der Wert *ALL oder Meldungsschlüssel aus mehreren Meldungsklassen zugewiesen wird.

TO-MSG-ID = *CLASS(...)
Bezeichnet eine neue Meldungsklasse für die zu übertragenden Meldungsbestandteile. Die Meldungsnummern bleiben unverändert.

> **MSG-CLASS = <name 3..3>**
> Bezeichnet die neue, dreistellige Meldungsklasse.

TO-MSG-ID = <alphanum-name 4..7 with-wild>
Der neue Meldungsschlüssel wird entweder vollständig angegeben oder über einen Teilnamen mit Platzhalter definiert.
Der Teilname bildet den Beginn des neuen Meldungsschlüssels, die restlichen Stellen, symbolisiert durch *, werden unverändert vom alten Meldungsschlüssel übernommen.

Beispiel

> Siehe Anweisung //COPY, Operand TO-MSG-ID und nachfolgende Beispiele.

TO-FILE = *CURRENT / <filename 1..54 without-gen-vers>
Bezeichnet die Meldungsdatei, in welche die Meldungsbestandteile übertragen werden.

TO-FILE = *CURRENT
Die Zuweisung TO-FILE = *CURRENT legt die mit der Anweisung //OPEN-MSG-FILE zuletzt eröffnete Meldungsdatei als Empfangsdatei fest. Als Empfangsdatei muss sie im UPDATE-Modus eröffnet worden sein.

TO-FILE = <filename 1..54 without-gen-vers>
Explizite Angabe der Meldungsdatei, in welche die Meldungsbestandteile übertragen werden.

OVERWRITE = *STD / *YES / *NO
Gibt an, ob der Meldungsbestandteil der Sendedatei einen bereits bestehenden Bereich der Empfangsdatei überschreiben darf.
Der Default-Wert *STD entspricht dem Wert der letzten OVERWRITE-Vereinbarung in der Anweisung //MODIFY-OPTION.

Beispiel

```
//MOVE MSG-ID=*INTERVAL(TST000,TST0009),-
   INFORMATION=(MESSAGE,DOCUMENTATION), TO-MSG-ID=TTT*, OVERWRITE=*NO
```

Beispiele zu den Operanden MSG-ID und TO-MSG-ID

Beispiel A

Wirkung der Anweisung:

```
//MOVE MSG-ID=*INTERVAL(CCCA000 ,CCCI999 ), TO-MSG-ID=DDD0*
```

CCCA001, CCCA002, ...	->	DDD0001, DDD0002, ...
CCCBAAA, CCCBAAB, ...	->	DDD1AAA, DDD1AAB, ...
.		.
.		
CCCE100, CCCE101, ...	->	DDD4100, DDD4101, ...
.		.
.		
CCCI998, CCCI999	->	DDD8998, DDD8999

Die im TO-MSG-ID-Operanden definierte Meldungsklasse oder der teilqualifizierte Meldungsschlüssel wird in den neuen Meldungsschlüssel übernommen, für den Platzhalter * erscheint der noch verbliebene Teil des alten Meldungsschlüssels.

Beispiel B

Wirkung der Anweisung:

```
//MOVE MSG-ID = *INTERVAL(CCCA000,CCCI999), TO-MSG-ID=CCC0*
```

Die vier Meldungschlüssel <CCCA010, CCCA800, CCCD033, CCCH000> werden umbenannt in <CCC0010, CCC0800, CCC3033, CCC7000> und **nicht** in <CCC0010, CCC0800, CCC1033, CCC2000>.

Der Teil der Meldungsnummer, der zum festgesetzten Teil des neuen Meldungsschlüssels gehört (bei „CCC0*“ entspricht dies „0“), wird, falls notwendig, automatisch um die vorgegebenen Schrittweite erhöht. Die Meldungsklasse wird nicht inkrementiert.

Beispiel C

Folgende Kombinationsmöglichkeiten sind vorhanden:

MSG-ID	TO-MSG-ID	Bedeutung	Ungenügende Bereichsangabe
*CLASS(CCC)	*CLASS(DDD)/ DDD*	Neue Meldungsklasse Beibehalten der Meldungsnummern	DDD0*
(CCCH403, CCCH503, CCCH603)	*SAME	*SAME	alle Angaben außer *SAME
*INT(CCC0000-CCC4999)	*CLASS(DDD)/ DDD* CCC5* CCC50*	DDD0000, ... ,DDD4999 DDD0000, ... ,DDD4999 CCC5000, ... ,CCC9999 CCC5000, ... ,CCC9999	CCC6* / CCC51*
*INT(CCC0000, CCC0599)	*CLASS(DDD)/ DDD* DDD5* DDD55* DDD555*	DDD0000, ... ,DDD0599 DDD0000, ... ,DDD0599 DDD5000, ... ,DDD5599 DDD5500, ... ,DDD5599 DDD5600, ... ,DDD5599 DDD6A00, ... ,DDD6A99 DDD5550, ... ,DDD5599 DDD56A0, ... ,DDD5999 DDD6AA0, ... ,DDD6BE9	DDD95*

Der Meldungsbereich der Empfangsdatei muss größer oder gleich dem der Sendedatei sein.

8.3.2.12 OPEN-MSG-FILE - Meldungsdatei eröffnen

Funktion

Mit der Anweisung //OPEN-MSG-FILE wird eine Meldungsdatei zur Bearbeitung eröffnet.

Die Meldungsdatei bleibt solange eröffnet, bis eine andere Datei eröffnet oder MSGMAKER beendet wird.

Die Anweisung //OPEN-MSG-FILE muss nicht vor jeder Dateibearbeitung angegeben werden. In den Anweisungen //COPY, //MOVE und //SHOW kann über den Operanden FROM-FILE, bei //MERGE-MSG-FILES über die Operanden FILE-NAMES und TO-FILE direkt auf eine Meldungsdatei zugegriffen werden.

Unterschiede zum Menümodus

Die Eingabe dieser Anweisung ist im Menümodus nicht möglich, da nur eine Meldungsdatei zur gleichen Zeit geöffnet sein kann.

Format

```
OPEN-MSG-FILE

 FILE-NAME = <filename 1..54>

 ,MODE = *UPDATE(...) / *CREATE(...) / *READ

     UPDATE(...)

         |    TYPE = *UNCHANGED / *CUSTOMER / *STANDARD

         |    ,PRODUCT = *UNCHANGED (...) / <structured-name 1..15>(...) / *NONE

         |        *UNCHANGED(...)

         |            |    VERSION = *UNCHANGED / <composed-name 3..8> / <c-string 1..8> / *NONE

         |        <structured-name 1..15>(...)

         |            |    VERSION = *UNCHANGED / <composed-name 3..8> / <c-string 1..8> / *NONE

     CREATE(...)

         |    TYPE = *CUSTOMER / *STANDARD

         |    ,PRODUCT = *NONE / <structured-name 1..15>(...)

         |        <structured-name 1..15>(...)

         |            |    VERSION = *NONE / <composed-name 3..8> / <c-string 1..8>
```

Operanden

FILE-NAME = <filename 1..54>
Gibt den Namen der Meldungsdatei an, die zur Bearbeitung eröffnet werden soll. Diese Datei ist solange aktuelle Datei, bis eine andere Datei geöffnet wird.

MODE = *UPDATE(...) / *CREATE(...) / *READ
Gibt den Modus an, in dem die Meldungsdatei eröffnet werden soll.

MODE = *UPDATE(...)
Die Meldungsdatei ist bereits katalogisiert und soll aktualisiert werden.

TYPE = *UNCHANGED / *CUSTOMER / *STANDARD
Gibt die Art einer bereits bestehenden Meldungsdatei an.

TYPE = *UNCHANGED
Die Art der Meldungsdatei wird nicht verändert.

> Die Umwandlung einer Standard-Meldungsdatei in eine kundeneigene Meldungsdatei durch Angabe von MODE=UPDATE(TYPE=CUSTOMER) ist nicht möglich. Die Umwandlung einer kundeneigenen Meldungsdatei in eine Standard-Meldungsdatei ist ohne weiteres möglich.

TYPE = *CUSTOMER
Die Meldungsdatei wird einem Kundenprodukt zugeordnet.

TYPE = *STANDARD
Die Meldungsdatei bezieht sich auf ein Software-Produkt, das beim Hersteller entwickelt wurde.

PRODUCT = *UNCHANGED(...) / <structured-name 1..15>(...) / *NONE
Name und Version des Software-Produkts, auf das sich die aktuelle Meldungsdatei bezieht. Die eingegebenen Buchstaben werden immer in Großbuchstaben umgewandelt.

PRODUCT = *UNCHANGED(...)
Der Name des Software-Produkts wird nicht verändert.

VERSION = *UNCHANGED / <composed-name 3..8> / <c-string 1..8> / *NONE
Version des Produkts. Die Angabe wird immer in Großbuchstaben umgewandelt.

PRODUCT = <structured-name 1..15>(...)
Neuer Name des Produkts, auf das sich die aktuelle Meldungsdatei bezieht.

VERSION = *UNCHANGED / <composed-name 3..8> / <c-string 1..8> / *NONE
Version des Produkts. Die Angabe wird immer in Großbuchstaben umgewandelt.

PRODUCT = *NONE
Dem Produkt wird weder ein Name, noch eine Version zugewiesen.

MODE = *CREATE(...)
Eine neue Meldungsdatei wird katalogisiert und zur aktuellen Arbeitsdatei.

TYPE = *CUSTOMER / *STANDARD
Legt fest, ob die Meldungsdatei zu einem kundeneigenen Produkt oder zu einem BS2000-Produkt gehört.

TYPE = *CUSTOMER
Die Meldungsdatei bezieht sich auf ein kundeneigenes Produkt.

TYPE = *STANDARD
Die Meldungsdatei bezieht sich auf ein Software-Produkt, das beim Hersteller entwickelt wurde.

PRODUCT = *NONE / <structured-name 1..15>(...)
Bezeichnet Name und Version des Produkts, auf welches die aktuelle Meldungsdatei Bezug nimmt.

PRODUCT = *NONE
Dem Produkt wird weder ein Name, noch eine Version zugewiesen.

PRODUCT = <structured-name 1..15>(...)
Name des Produkts, auf das sich die Meldungsdatei bezieht.

VERSION = *NONE / <composed-name 3..8> / <c-string 1..8>
Version des Produkts. Die eingegebenen Buchstaben werden immer in Großbuchstaben umgewandelt.

MODE = *READ
Die Meldungsdatei ist nur zum Lesen geöffnet; es können keine Änderungen vorgenommen werden.

Beispiel

Meldungsdatei erzeugen:

```
//OPEN-MSG-FILE FILE-NAME=SYSMES.TSTFILE,MODE=CREATE(TYPE=STANDARD,-
   PRODUCT=TSTPROD(VERSION=V01.0A10))
```

Meldungsdatei aktualisieren:

```
//OPEN-MSG-FILE SYSMES.TSTFILE,(PRODUCT=*UNCHANGED('V1.0A10'))
```

8.3.2.13 SHOW - Inhalt einer Meldungsdatei anzeigen

Funktion

Mit der Anweisung //SHOW kann der Inhalt einer Meldungsdatei auf Bildschirm oder Drucker ausgegeben werden. Die Meldungseinheiten werden, sortiert nach Meldungsklassen, mit oder ohne Dokumentationszeilen nach SYSOUT oder SYSLST übertragen.
Näheres zum Ausgabeformat der Meldungsdatei, siehe "Maske SHOW-OUTPUT - Ausgabe von Meldungseinheiten und Zusatzinformationen".

Unterschiede zur Anweisung //SHOW im Menümodus

Die Anweisung //SHOW, die in den Kommandobereich der Bildschirmmaske eingetragen werden kann, unterscheidet sich von der Anweisung //SHOW in Kommandoprozeduren, durch

- den Operandenwert *PANEL-REQUEST, der im Menümodus jedem Operanden zugewiesen werden kann
- einen zusätzlichen Wert des Operanden MSG-ID

Näheres im Abschnitt „Besonderheiten der Anweisungen im Menümodus“.

Format

```
SHOW

MSG-ID = *ALL / list-poss(2000): *CLASS(...) / *INTERVAL(...) / <name 7..7>

    *CLASS(...)

        |    MSG-CLASS = <name 3..3>

    *INTERVAL(...)

        |    FROM = <name 7..7>

        |    ,TO = <name 7..7>

,INFORMATION = *MESSAGES (...) / *ALL / list-poss(4): *MESSAGES(...) / *DOCUMENTATION /

                    *COMPONENT-ID / *CORRECTION-INFO

    MESSAGES(...)

        |    LANGUAGES = *ALL / list-poss(8): <name 1..1>

        |    ,ELEMENTS = *ALL / list-poss(2000): *ATTRIBUTES / *MSG-TEXT /

        |                *MEANING-RESPONSE / *INSERT-ATTRIBUTES

,FROM-FILE = *CURRENT / <filename 1..54>

,OUTPUT = *SYSOUT / *ALL / list-poss(2): *SYSOUT / *SYSLST
```

Operanden

MSG-ID = *ALL / list-poss(2000): *CLASS(...) / *INTERVAL(...) / <name 7..7>
Bezeichnet die Meldungsschlüssel einer oder mehrerer Meldungseinheiten der eröffneten Meldungsdatei, die

ausgegeben werden sollen.
Durch die Zuweisung INFORMATION=DOCUMENTATION können die zu den Meldungsschlüsseln definierten Dokumentationszeilen ebenfalls ausgegeben werden.

MSG-ID = *ALL
Alle im Operanden INFORMATION definierten Meldungsbestandteile werden ausgegeben.

MSG-ID = *CLASS(...)
Jede Meldungseinheit, deren Meldungsschlüssel mit der angegebenen Meldungsklasse beginnt, wird mit oder ohne Dokumentationszeilen (gemäß den Vereinbarungen imOperanden INFORMATION) angezeigt.

MSG-CLASS = <name 3..3>
Gibt die dreistellige Meldungsklasse an.

MSG-ID = *INTERVAL(...)
Jede Meldungseinheit, deren Meldungsschlüssel innerhalb des angegebenen Meldungsintervalls liegt, wird mit oder ohne Dokumentationszeilen (gemäß den Vereinbarungen im Operanden INFORMATION) angezeigt.

FROM = <name 7..7>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls.

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssel des Meldungsintervalls.

MSG-ID = <name 7..7>
Bezeichnet den vollständigen Meldungsschlüssel einer Meldungseinheit, die mit oder ohne Dokumentationszeilen (gemäß den Vereinbarungen im Operanden INFORMATION) angezeigt wird.

INFORMATION = *MESSAGES(...) / *ALL / list-poss: *MESSAGES(...) / *DOCUMENTATION / *COMPONENT-ID / *CORRECTION-INFO
Bezeichnet die Bestandteile einer Meldungdatei, die angezeigt werden sollen.

INFORMATION = *MESSAGES(...)
Eine oder mehrere Meldungseinheiten sollen angezeigt werden.

LANGUAGES = *ALL / list-poss(8): <name 1..1>
Kennbuchstaben für die Sprachen, in welchen die Meldungs-, Bedeutungs- und Maßnahmetexte definiert worden sind.

LANGUAGES = *ALL
Alle sprachabhängigen Teile einer Meldungseinheit werden in alphabetischer Reihenfolge angezeigt.

LANGUAGES = list-poss(8): <name 1..1>
Es können bis zu acht Kennbuchstaben angegeben werden. Die Teile der Meldungseinheit, die in diesen Sprachen definiert wurden, werden mit der hier festgelegten Reihenfolge der Kennbuchstaben angezeigt.

ELEMENTS = *ALL / list-poss: *ATTRIBUTES / *MSG-TEXT / *MEANING-RESPONSE / *INSERT-ATTRIBUTES
Bezeichnet Teile einer Meldungseinheit. Die Operandenwerte ATTRIBUTES und INSERT-ATTRIBUTES sind unabhängig von der Definition einer Sprache.

ELEMENTS = *ALL
Alle definierten Teile einer Meldungseinheit, wie Meldungsattribute, Insert-Attribute und Meldungs-, Bedeutungs- und Maßnahmetexte werden ausgegeben.

ELEMENTS = *ATTRIBUTES
Alle definierten Meldungsattribute der Meldungseinheit werden ausgegeben.

ELEMENTS = *MSG-TEXT
Ausgabe des Meldungstextes.

ELEMENTS = *MEANING-RESPONSE
Ausgabe des Bedeutungs- und Maßnahmetextes.

ELEMENTS = *INSERT-ATTRIBUTES
Alle definierten Insert-Attribute werden ausgegeben.

INFORMATION = *ALL
Alle, im Meldungsbereich (Operand MSG-ID) definierten Bestandteile einer Meldungsdatei werden ausgegeben. Ist dem Operanden MSG-ID der Wert *ALL zugewiesen, wird der gesamte Inhalt der Meldungsdatei angezeigt.

INFORMATION = *DOCUMENTATION
Ausgabe von Dokumentationszeilen, die im angegebenen Meldungsbereich (Operand MSG-ID) liegen. Ferner werden auch solche Dokumentationszeilen angezeigt, deren Meldungsbereich nur einen Teil des hier definierten Bereichs überschneidet.

Beispiel

Wird MSG-ID = *INTERVAL(TST1500-TST1599) angegeben, werden alle Dokumentationszeilen der definierten Meldungsbereiche (TST1500-TST1510), (TST1000-TST1999), (TST1000-TST1500) oder TST1510 angezeigt.

INFORMATION = *CORRECTION-INFO / *COMPONENT-ID
Diese Operanden sind für den internen Gebrauch beim Hersteller reserviert.

FROM-FILE = *CURRENT / <filename 1..54 without-gen-vers>
Name der Meldungsdatei, deren Meldungsbestandteile angezeigt werden sollen. Der Operandenwert *CURRENT bezeichnet die aktuell geöffnete Meldungsdatei.

OUTPUT = *SYSOUT / *ALL / list-poss: *SYSOUT / *SYSLST
Gibt das Ausgabeziel für die oben festgelegten Bestandteile der Meldungsdatei an.

OUTPUT = *SYSOUT
Die Bestandteile der Meldungsdatei werden in die Systemdatei SYSOUT ausgegeben.

OUTPUT = *ALL
Die Bestandteile der Meldungsdatei werden auf SYSOUT und SYSLST ausgegeben.

OUTPUT = *SYSLST
Die Bestandteile der Meldungsdatei werden auf SYSLST ausgegeben. Nach Beenden von MSGMAKER kann mit dem BS2000-Kommando /PRINT-DOCUMENT die SYSLST-Datei ausgedruckt werden. Der Text wird mit 60 Zeilen pro Seite ausgegeben.

Beispiel

```
//SHOW MSG-ID=*INTERVAL (TSTAAA,TTT9999),
    INFORMATION=MESSAGES(LANGUAGES=(E,F),ELEMENTS=(ATTIBUTES,MSG-TEXT)),
    FROM-FILE=SYSMES.TSTFILE.OUTPUT=*SYSLST
```

8.3.3 Besonderheiten der Anweisungen im Menümodus

Im Menümodus werden Anweisungen in den dreizeiligen Kommandobereich der Maske eingegeben. Die Operanden können fortlaufend, ohne Fortsetzungszeichen in die Zeilen eingetragen werden. Die Eingabe wird durch Drücken der Taste DUE bestätigt. Bei vollständiger und fehlerloser Eingabe der Operanden wird die Funktion im „Hintergrund“ ausgeführt, wobei die Felder der aktuelle Maske unverändert bleiben.

Ist die Anweisung syntaktisch fehlerhaft, kann im SDF-geführten Dialog die Eingabe korrigiert werden.
Ebenso wird bei fehlenden Operanden in den SDF-geführten Dialog oder zu den entsprechenden Masken verzweigt, um die notwendigen Werte ergänzen zu können. Nach korrekter Funktionsausführung wird in die aufrufende Maske zurückgekehrt.

Anstelle einer direkten Eingabe von Operanden in den Kommandobereich, kann auf den eingetragenen Anweisungsnamen die Prompt-Funktion ausgeführt werden. D.h. durch Drücken der Taste F2 im Anschluss an die Eingabe im Kommandobereich wird die der Anweisung entsprechende Maske aufgerufen. In der Maske können nun die fehlenden Operanden ergänzt werden. Mit DUE werden die Eingaben bestätigt und die Anweisung ausgeführt. Nähere Informationen zur Prompt-Funktion siehe "Allgemeiner Maskenaufbau".
Die Felder der Ausgangsmaske sind durch Aufruf der Maske und Ausführung der Anweisung nicht verändert worden.

Anweisungen, die im Kommandobereich der Bildschirmmaske eingetragen werden können, unterscheiden sich in einigen Operanden bzw. Operandenwerten von den Anweisungen, die in Batch-Aufträgen und Prozeduren möglich sind.

*Hinweise zum Operandenwert *PANEL-REQUEST*

Grundsätzlich kann allen Operanden der Anweisungen, die im Kommandobereich der Bildschirmmaske eingegeben werden, der Operandenwert ***PANEL-REQUEST** zugewiesen werden.
*PANEL-REQUEST bewirkt den Wechsel zur Maske, deren „panel-id“ in den meisten Fällen dem Anweisungsnamen entspricht. Der aktuelle Wert des Operanden wird in der Maske angezeigt und kann durch weitere Anweisungen oder Bildschirmfunktionen verändert werden.
*PANEL-REQUEST ist immer dann Default-Wert, wenn für die Operanden, die im Abschnitt„Anweisungen“ beschrieben sind, kein Default-Wert (unterstrichener Wert) angegeben ist.

Beispiel

Durch Eingabe folgender Anweisung wird die Maske MODIFY-MSG aufgerufen:

```
//MODIFY-MSG MSG-ID = TST0001, LAN=*PANEL-REQUEST
```

In ihr werden alle zum Meldungsschlüssel TST0001 vorhandenen Meldungsattribute, Kennbuchstaben für die Sprache und Texte angezeigt, welche im nächsten Arbeitsschritt verändert werden können.

In folgender Zusammenstellung sind nur solche Anweisungen aufgeführt, deren Operanden im Menümodus andere oder zusätzliche Operandenwerte erhalten.

- Steht ein Operandenwert zusätzlich zur Verfügung, ist dies auf folgende Weise gekennzeichnet: z.B. `... / <alphanum-name 1..7 with-wild> / ...` Alle nicht angeführten Operanden (angedeutet durch ...) stehen sowohl im Menümodus als auch in Batch-Aufträgen und Prozeduren zur Verfügung.
- Ist ein Operandenwert im Menümodus grundsätzlich von dem in Batch-Aufträgen und Prozeduren verschieden, ist dies in der Operandenbeschreibung ausdrücklich vermerkt.

Einschränkungen

- Die Anweisungen //MODIFY-OPTION und //OPEN-MSG-FILE sind nur in Kommandoprozeduren möglich.
- Mit der Eingabe eines Fragezeichens und DUE in den Kommandobereich einer Maske liefert SDF eine Auswahl aller verfügbaren Anweisungen.

  Wenn eine Standard-Meldungsdatei des Herstellers geöffnet ist, befinden sich hierunter auch die folgenden Anweisungen, die für den internen Gebrauch reserviert sind:

  - //ADD-COMPONENT-ID
  - //DELETE-COMPONENT-ID
  - //MODIFY-COMPONENT-ID
  - //ADD-CORRECTION-INFORMATION
  - //DELETE-CORRECTION-INFORMATION
  - //MODIFY-CORRECTION-INFORMATION
  - //GENERATE-CSECT

8.3.3.1 ADD-DOCUMENTATION

Die Anweisung //ADD-DOCUMENTATION ist bis auf den Operandenwert *PANEL-REQUEST im Menümodus und in Kommandoprozeduren identisch.

8.3.3.2 ADD-MSG

Soll im Menümodus mit der Anweisung //ADD-MSG eine bereits bestehende Meldungseinheit eingefügt werden, wechselt MSGMAKER automatisch in die Maske MODIFY-MSG und gibt eine Warnung aus.

Format (Operanden im Menümodus)

```
ADD-MSG

MSG-ID = *CLASS(...) / *INTERVAL(...) / <alphanum-name 1..7 with-wild> / list-poss(2000): <name 7..7>

    *CLASS(...)

        |    MSG-CLASS = <name 3..3>

    *INTERVAL(...)

        |    FROM = <alphanum-name 1..7 with-wild>

        |    ,TO = <name 7..7>
```

Der Operand OVERWRITE entfällt.

Operanden

MSG-ID = *CLASS(...) / *INTERVAL(...) / <alphanum-name 1..7 with-wild> /list-poss(2000): <name 7..7>
Bezeichnet den Meldungsschlüssel der Meldungseinheit.
Im Gegensatz zur Anweisung //ADD-MSG in Batch-Aufträgen und Prozeduren kann eine Meldungsklasse, ein Meldungsintervall oder ein teilweise definierter Meldungsschlüssel angegeben werden. Ferner ist es möglich, Meldungsschlüssel in einer Liste anzugeben.

MSG-ID = *CLASS(...)
Definiert eine Meldungsklasse.

MSG-CLASS = <name 3..3>
Gibt die dreistellige Meldungsklasse an.

MSG-ID = *INTERVAL(...)
Definiert ein Meldungsintervall.

FROM = <alphanum-name 1..7 with-wild>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls. Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert werden, die ersten drei Stellen sollten immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen, der Platzhalter # ein bis vier Ziffern ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 1..7 with-wild> möglich:
, A, AB*, ABC*, ABC0*, ABC00*, ABC000*
ABC#, ABC0#, ABC00#, ABC000#

TO = <name 7..7>
Bezeichnet den letzten Meldungsschlüssel des Meldungsintervalls. Der vollständige, siebenstellige Meldungsschlüssel muss angegeben werden.

MSG-ID = <alphanum-name 1..7 with-wild>
Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert werden, die ersten drei Stellen sollten immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen, der Platzhalter # ein bis vier Ziffern ersetzen.

Beispiel

Siehe Operand MSG-ID=*INTERVAL(FROM=<alphanum-name 1..7 with-wild>)

MSG-ID = list-poss(2000): <name 7..7>
Es können vollständige Meldungsschlüssel als Liste angegeben werden.

8.3.3.3 COPY

Format (Operanden im Menümodus)

```
COPY

 MSG-ID = ... / <alphanum-name 1..7 with-wild> / ...

     *INTERVAL(...)

         |    FROM = <alphanum-name 1..7 with-wild>
```

Der Operand OVERWRITE entfällt.

Operanden

MSG-ID = <alphanum-name 1..7 with-wild>
Ergänzend zu den Werten für den Operanden MSG-ID in Batch-Aufträgen und Prozeduren kann der Meldungschlüssel im Menümodus vollständig oder teilweise definiert werden. Die ersten drei Stellen des siebenstelligen Meldungsschlüssels (Meldungsklasse) sollten hierbei immer Buchstaben sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 1..7 with-wild> möglich:
, A, AB*, ABC*, ABC0*, ABC00*, ABC000*

MSG-ID = *INTERVAL(...)
Definiert ein Meldungsintervall.

FROM = <alphanum-name 1..7 with-wild>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls. Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert werden, die ersten drei Stellen sollten immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.
Unterschied zum Operanden in Batch-Aufträgen und Prozeduren; dort kann der Meldungsschlüssel nur vollständig (FROM = <name 7..7>) angegeben werden.

Beispiel

Siehe Operand MSG-ID=<alphanum-name 1..7 with-wild>

8.3.3.4 DELETE-DOCUMENTATION

Die Anweisung //DELETE-DOCUMENTATION ist bis auf den Operandenwert *PANEL-REQUEST im Menümodus und in Kommandoprozeduren identisch.

8.3.3.5 DELETE-MSG

> ¡ Dem Operanden MSG-ID kann im Menümodus der Wert *ALL **nicht** zugewiesen werden.

Format (Operanden im Menümodus)

DELETE-MSG
MSG-ID = ... / <alphanum-name 4..7 with-wild> / ...

Operanden

MSG-ID = <alphanum-name 4..7 with-wild>
Ergänzend zu den Werten für den Operanden MSG-ID in Batch-Aufträgen und Prozeduren (mit Ausnahme von *ALL) kann der Meldungschlüssel im Menümodus vollständig oder teilweise definiert werden. Der Platzhalter * kann ein bis vier Zeichen ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 4..7 with-wild> möglich:
ABC*, ABC0*, ABC00*, ABC000*

8.3.3.6 END

Die Anweisung //END ist im Menümodus und in Kommandoprozeduren identisch.

8.3.3.7 GO-TO - Verzweigen zu einer angegebenen Maske

Die Anweisung //GO-TO kann nur im Menümodus eingegeben werden.

Mit der Anweisung //GO-TO kann der Anwender den von MSGMAKER vorgeschriebenen Pfad verlassen (Masken-Übersicht) und eine gewünschte Maske direkt ansteuern.
Die Anweisung //GO-TO kann in den Kommandobereich jeder Maske eingegeben und mit DUE gestartet werden. GO-TO wird sofort ausgeführt, wobei die Funktion der aktuellen Bildschirmmaske nicht mehr ausgeführt wird. Daten, die in diese Maske eingegeben worden sind, gehen verloren.

Ausnahme: Maske MENU

> Wird eine Meldungsdatei in die Maske MENU eingetragen und nicht mit DUE bestätigt, wird die Meldungsdatei trotzdem zuerst geöffnet und anschließend in die bei GO-TO angegebene Maske verzweigt.

Format (Operanden im Menümodus)

GO-TO / GOTO
PANEL-ID = *MENU / *MSG-FILE-ATTRIBUTES / *ADD-MSG / *MODIFY-MSG / *DELETE-MSG / *COPY / *MOVE / *SHOW / *ADD-DOCUMENTATION / *MODIFY-DOCUMENTATION / *DELETE-DOCUMENTATION

Operanden

PANEL-ID = *MENU / *MSG-FILE-ATTRIBUTES / *ADD-MSG / MODIFY-MSG / *DELETE-MSG / *COPY / *MOVE / *SHOW / *ADD-DOCUMENTATION /*MODIFY-DOCUMENTATION / *DELETE-DOCUMENTATION
Name der Maske, zu der verzweigt werden soll. Der Name kann gemäß der SDF-Konventionen abgekürzt werden.

Einschränkung

> Die Masken MSG-TEXT, MEANING/RESPONSE und INSERT-ATTRIBUTES können mit der Anweisung //GO-TO nicht angesteuert werden.

Beispiel

> GO-TO **add-msg** entspricht GO-TO **a-m**

8.3.3.8 MERGE-MSG-FILES

Die Anweisung //MERGE-MSG-FILES ist im Menümodus und in Kommandoprozeduren identisch.

Zum gegenwärtigen Zeitpunkt existiert noch keine Maske, in der die Funktion MERGE-MSG-FILES ausgeführt werden kann. Demzufolge ist der Operandenwert *PANEL-REQUEST im Menümodus nicht verfügbar.

Die Anweisung //MERGE-MSG-FILES kann im Menümodus entweder mit F2 oder mit DUE gestartet werden.

8.3.3.9 MODIFY-DOCUMENTATION

Die Anweisung //MODIFY-DOCUMENTATION ist bis auf den Operandenwert *PANEL-REQUEST im Menümodus und in Kommandoprozeduren identisch.

8.3.3.10 MODIFY-MSG

Format (Operanden im Menümodus)

```
MODIFY-MSG

 MSG-ID = ... / <alphanum-name 1..7 with-wild> / ...

     *INTERVAL(...)

           FROM = <alphanum-name 1..7 with-wild>

,LANGUAGES = ... / *ANY(...) / ...

     *ANY(...)

        |  MSG-TEXT = *UNCHANGED

        |  ,MEANING = *UNCHANGED

        |  ,RESPONSE = *UNCHANGED
```

Operanden

MSG-ID = <alphanum-name 1..7 with-wild>
Ergänzend zu den Werten für den Operanden MSG-ID in Batch-Aufträgen und Prozeduren kann der Meldungschlüssel im Menümodus vollständig oder teilweise definiert werden.

Die ersten drei Stellen des siebenstelligen Meldungsschlüssels (Meldungsklasse) sollten hierbei immer Buchstaben sein. Der Platzhalter * kann ein bis sieben Zeichen, der Platzhalter # ein bis vier Ziffern ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 1..7 with-wild> möglich:
, A, AB*, ABC*, ABC0*, ABC00*, ABC000*
ABC#, ABC0#, ABC00#, ABC000#

MSG-ID = *INTERVAL(...)
Definiert ein Meldungsintervall.

FROM = <alphanum-name 1..7 with-wild>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls. Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert werden, die ersten drei Stellen sollten immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen, der Platzhalter # ein bis vier Ziffern ersetzen.
Unterschied zum Operanden in Batch-Aufträgen und Prozeduren; dort kann der Meldungsschlüssel nur vollständig (FROM = <name 7..7>) angegeben werden.

Beispiel

Siehe Operand MSG-ID=<alphanum-name 1..7 with-wild>

LANGUAGES = *ANY(...)
Ergänzend zu den Werten des Operanden LANGUAGES in Batch-Aufträgen und Prozeduren kann hier der

Operandenwert *ANY angegeben werden. Innerhalb der Struktur, die von *ANY eingeleitet wird, können den Operanden MSG-TEXT, MEANING und RESPONSE nur die Werte *UNCHANGED oder *PANEL-REQUEST zugewiesen werden. Bei LANGUAGES = *ANY können keine Texte eingegeben werden.

MSG-TEXT = *UNCHANGED
Der Meldungstext soll nicht verändert werden. Der Inhalt der Maske MSG-TEXT kann nur verändert werden, wenn der Wert *PANEL-REQUEST zugewiesen ist.

MEANING = *UNCHANGED
Der Bedeutungstext soll nicht verändert werden. Der Inhalt der Maske MEANING/RESPONSE kann nur verändert werden, wenn der Wert *PANEL-REQUEST zugewiesen ist.

RESPONSE = *UNCHANGED
Der Maßnahmetext soll nicht verändert werden. Der Inhalt der Maske MEANING/RESPONSE kann nur verändert werden, wenn der Wert *PANEL-REQUEST zugewiesen ist.

8.3.3.11 MOVE

Format (Operanden im Menümodus)

MOVE
MSG-ID = ... / <alphanum-name 1..7 with-wild> / ... *INTERVAL(...) | FROM = <alphanum-name 1..7 with-wild>

Der Operand OVERWRITE entfällt.

Operanden

MSG-ID = <alphanum-name 1..7 with-wild>
Ergänzend zu den Werten für den Operanden MSG-ID in Batch-Aufträgen und Prozeduren kann der Meldungschlüssel im Menümodus vollständig oder teilweise definiert werden. Die ersten drei Stellen des siebenstelligen Meldungsschlüssels (Meldungsklasse) sollten hierbei immer Buchstaben sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 1..7 with-wild> möglich:
, A, AB*, ABC*, ABC0*, ABC00*, ABC000*

MSG-ID = *INTERVAL(...)
Definiert ein Meldungsintervall.

FROM = <alphanum-name 1..7 with-wild>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls. Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert sein, die ersten drei Stellen sollten jedoch immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.
Unterschied zum Operanden in Batch-Aufträgen und Prozeduren; dort kann der Meldungsschlüssel nur vollständig (FROM = <name 7..7>) angegeben werden.

Beispiel

Siehe Operand MSG-ID=<alphanum-name 1..7 with-wild>

8.3.3.12 SHOW

Format (Operanden im Menümodus)

```
SHOW

 MSG-ID = ... / <alphanum-name 1..7 with-wild> / ...

     *INTERVAL(...)

         |    FROM = <alphanum-name 1..7 with-wild>
```

Operanden

MSG-ID = <alphanum-name 1..7 with-wild>
Ergänzend zu den Werten für den Operanden MSG-ID in Batch-Aufträgen und Prozeduren kann der Meldungschlüssel im Menümodus vollständig oder teilweise definiert werden. Die ersten drei Stellen des siebenstelligen Meldungsschlüssels (Meldungsklasse) sollten hierbei immer Buchstaben sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.

Beispiel

Folgende Angaben sind für <alphanum-name 1..7 with-wild> möglich:
, A, AB*, ABC*, ABC0*, ABC00*, ABC000*

MSG-ID = *INTERVAL(...)
Definiert ein Meldungsintervall.

FROM = <alphanum-name 1..7 with-wild>
Bezeichnet den ersten Meldungsschlüssel des Meldungsintervalls. Der siebenstellige Meldungsschlüssel kann vollständig oder teilweise definiert sein, die ersten drei Stellen sollten jedoch immer Buchstaben (Meldungsklasse) sein. Der Platzhalter * kann ein bis sieben Zeichen ersetzen.
Unterschied zum Operanden in Batch-Aufträgen und Prozeduren; dort kann der Meldungsschlüssel nur vollständig (FROM = <name 7..7>) angegeben werden.

Beispiel

Siehe Operand MSG-ID=<alphanum-name 1..7 with-wild>

8.3.4 Beispiel

In der Prozedur TEST.MSG wird mit dem Programm MSGMAKER zunächst eine Meldungsdatei erzeugt und eine Meldungseinheit eingetragen. Im Anschluss daran werden Meldungen aus der Meldungsdatei SYSMES.MSG.010 in die neue Meldungsdatei kopiert und verändert. Der Inhalt der neu erzeugten und anschließend veränderten Meldungsdatei SYSMES.TSTFILE wird in eine SYSOUT- und SYSLST-Datei ausgegeben. Das SYSOUT-Protokoll finden Sie im Anschluss an die Beschreibung der einzelnen Prozedurschritte.

Prozedur TEST.MSG

```
/BEGIN-PROCEDURE
/ASSIGN-SYSLST TO=TEST.MSG.SYSLST
/ASSIGN-SYSOUT TO=TEST.MSG.SYSOUT
/ASSIGN-SYSDTA *SYSCMD
/REMARK ************************************************************
/REMARK ** Part 1 : Create a new message file
/REMARK ************************************************************
/START-MSGMAKER ──────────────────────────────────────────────────── (1)
//REMARK
//REMARK ************************ OPEN-MSG-FILE *************************
//REMARK
//OPEN-MSG-FILE FILE-NAME=SYSMES.TSTFILE,-
// MODE=CREATE(TYPE=C,PRODUCT=TSTPROD(VERSION=V01.0A10)) ── (2)
//REMARK
//REMARK ************************ ADD-MSG *********************************
//REMARK
//ADD-MSG MSG-ID=TST0000,- ─────────────────────────────────────────── (3)
// ACCESS-METHODS=ISAM,-
// DESTINATIONS=CONSOLE(ROUTING-CODE=A),-
// WEIGHT=30,-
// LANGUAGES=(E(-
// MSG-TEXT='Text in english with inserts ''(&&00)'' and ''(&&01)''',-
// MEANING=('First meaning line',-
// 'second meaning line'),-
// RESPONSE='Response line'-
// ),-
// D(-
// MSG-TEXT='Text in Deutsch mit Inserts ''(&&01)'' und ''(&&00)''',-
// MEANING=('Erste Bedeutungszeile',-
// 'Zweite Bedeutungszeile'),-
// RESPONSE=('Erste Massnahmeszeile',-
// 'Zweite Massnahmeszeile')-
// )-
// ),-
// INSERT-ATTRIBUTES=(0(NAME=NAM0,DEFAULT-VALUE='default0'),-
// 1(NAME=NAM1,DEFAULT-VALUE='default1'))
//REMARK
//REMARK ************************ SHOW ************************************
//REMARK
//SHOW MSG-ID=*ALL,OUTPUT=*ALL ─────────────────────────────────────── (4)
//END
/REMARK ************************************************************
/REMARK ** Part 2 : Messages from another message file
/REMARK ************************************************************
/REMARK
/START-MSGMAKER ──────────────────────────────────────────────────── (5)
//REMARK
//REMARK ************************ SHOW (other file) ************************
```

```
//REMARK
//SHOW MSG-ID=(PEP0001,PEP0002,PEP0004),- ———————————————————— (6)
// INFORMATION=(MESSAGES(LANGUAGES=E,ELEMENTS=(MSG-TEXT,MEAN-RESP)),-
// DOCUMENTATION),-
// FROM-FILE=SYSMES.MSG.010,-
// OUTPUT=*ALL
//REMARK
//REMARK ************************ OPEN-MSG-FILE ***************************
//REMARK
//OPEN-MSG-FILE SYSMES.TSTFILE,(PRODUCT=*UNCHANGED('V01.1A10')) ———— (7)
//REMARK
//REMARK ***** COPY (from the other file) *********************************
//REMARK
//COPY MSG-ID=(PEP0001,PEP0002,PEP0004),- ———————————————————— (8)
// INFORMATION=*ALL,-
// FROM-FILE=SYSES.MSG.010
//REMARK
//REMARK ************************ ADD-MSG *********************************
//REMARK
//ADD-MSG MSG-ID=PEP0005,- ———————————————————————————————— (9)
// LANGUAGES=(E(-
// MSG-TEXT='File ''(&&05)'' does not exist',-
// ),-
// D(-
// MSG-TEXT='Datei ''(&&05)'' nicht vorhanden',-
// )-
// )
//REMARK
//REMARK ************************ MODIFY-MSG ******************************
//REMARK
//REMARK +++++ Modify the message attributes +++++
//REMARK
//MODIFY-MSG MSG-ID=(PEP0001,PEP0002),- ———————————————————— (10)
// ACCESS-METHODS=DLAM,-
// DESTINATIONS=(USER-TASK,-
// CONSOLE(ROUTING-CODE=*MAIN-CONSOLE)),-
// INSERT-ATTRIBUTES=1(DEFAULT-VALUE= 'E')
//REMARK
//REMARK +++++ Modify the texts +++++
//REMARK
//MODIFY-MSG - ———————————————————————————————————————— (11)
// MSG-ID=PEP0001,-
// LANG=E(-
// MEANING=(-
// *REPLACE(-
// LINE-NUMBER=1,-
// TEXT='For more detailed information about the DMS error code-
// enter /HELP-MSG in'),-
// *ADD('system mode or see the BS2000 manual ''System-
// Messages''.')-
// )-
// )
//REMARK
//REMARK ************************ MOVE (rename a message) ******************
//REMARK
//MOVE MSG-ID=PEP0004,- ———————————————————————————————— (12)
// INFORMATION=(MESSAGES(LANGUAGES=E)),-
// TO-MSG-ID=TST*,-
// OVERWRITE=NO
```

```
//REMARK
//REMARK ************************ SHOW (new message file) ******************
//REMARK
//SHOW MSG-ID=(*CLASS(PEP),- ──────────────────────────────────────────── (13)
// TST0004),-
// INFORMATION=(MESSAGES,DOCUMENTATION),-
// OUTPUT=*ALL
//REMARK
//REMARK ************************ DELETE-MSG ******************************
//REMARK
//DELETE-MSG MSG-ID=*CLASS(TST),LANGUAGES=(D,E) ──────────────────────── (14)
//END
/END-PROCEDURE
```

Erläuterung

1. Aufruf von MSGMAKER
2. Eröffnen der Meldungsdatei SYSMES.TSTFILE für das Produkt TSTPROD mit der Version V01.0A10.
3. Mit dem Meldungsschlüssel TST0000 wird eine Meldung in die neu eröffnete Meldungsdatei eingetragen. Neben Meldungsattributen (Zugriffsmethode, Ausgabeziel, Weight Code) werden Meldungs-, Bedeutungs- und Maßnahmetext in Englisch und Deutsch eingetragen. Die Meldungstexte enthalten die Inserts (&00) und (&01). Sollen die Inserts bei Meldungsausgabe in Hochkommata eingeschlossen werden, müssen die Hochkommata in der Anweisung verdoppelt werden. Bitte beachten Sie auch die Verdoppelung des Zeichens &.
4. Der Inhalt der Meldungsdatei SYSMES.TSTFILE wird nach SYSOUT und SYSLST ausgegeben.
5. Erneuter Aufruf von MSGMAKER.
6. Drei Meldungen aus der Meldungsdatei SYSMES.MSG.010 werden nach SYSOUT und SYSLST ausgegeben. Die Ausgabe umfaßt die englischen Meldungs-, Bedeutungs- und Maßnahmetexte, sowie die Dokumentationszeilen.
7. Die Meldungsdatei SYSMES.TSTFILE wird erneut, mit einer geänderten Versionsangabe (V01.1A10) geöffnet.
8. Aus der Meldungsdatei SYSMES.MSG.010 werden drei Meldungen inkl. Dokumentation in die Datei SYSMES.TSTFILE kopiert.
9. Die neue Meldung PEP0005 wird in die Meldungsdatei SYSMES.TSTFILE eingetragen.
10. Für die zwei Meldungen PEP0001 und PEP0002 wird die Zugriffsmethode, das Ausgabeziel und der Default-Text für Insert (&01) verändert.
11. Für die Meldung PEP0001 wird der englische Bedeutungstext verändert. Die erste Zeile wird durch einen neuen Text ersetzt, eine zweite Zeile wird angefügt.
12. Alle englischen Texte der Meldung PEP0004 werden unter dem Meldungschlüssel TST0004 abgespeichert.
13. Die Meldungen der Meldungsklasse PEP und die Meldung TST0004 werden nach SYSOUT und SYSLST ausgegeben.
14. Die englischen und deutschen Texte der Meldung TST0004 werden gelöscht.

Ausgabe nach SYSOUT

zu Punkt (4):

```
################################################################################
###################   M e s s a g e   c l a s s :   T S T   ###################
################################################################################
-- TST0000 -------------------------------------------------------------
Access     : ISAM
Destination: CONSOLE              Routing code: A                Weight: 30
Warranty   : NO                   Text format : UPPER CASE
Insert attributes:
(&00)  Name: NAM0
       Default value: 'default0'
(&01)  Name: NAM1
       Default value: 'default1'
D   Text in Deutsch mit Inserts '(&01)' und '(&00)'
  ? Erste Bedeutungszeile
    Zweite Bedeutungszeile
  ! Erste Massnahmezeile
    Zweite Massnahmezeile
E   Text in english with inserts '(&00)' and '(&01)'
  ? First meaning line
    second meaning line
  ! Response line
%  MSMN600 MSGMAKER TERMINATED NORMALLY
```

zu Punkt (6):

```
################################################################################
###################   M e s s a g e   c l a s s :   P E P   ###################
################################################################################
-- PEP0001 -------------------------------------------------------------
E   MESSAGE WITH INSERT (&01)
  ? meaning for pep0001
  ! response for pep0001
-- PEP0002 -------------------------------------------------------------
E   MESSAGE WITH INSERT (&01)
  ? meaning for pep0002
  ! response for pep0002
-- PEP0004 -------------------------------------------------------------
E   MESSAGE WITH INSERT (&01)
  ? meaning for pep0004
  ! response for pep0004
=======================   D O C U M E N T A T I O N   ====================
PE0001-PEP0004     Owner name: MAIER                 Team: TEAM1
                   product = DMS-ERR
```

zu Punkt (13):

```
################################################################################
###################   M e s s a g e   c l a s s :   P E P   ###################
################################################################################
-- PEP0001 ---------------------------------------------------------------------
Access     : DLAM
Destination: USER-TASK, CONSOLE  Routing code: * (main console)   Weight: 99
Warranty   : NO                  Text format : UPPER CASE
Insert attributes:
(&01)  Default value: 'E'
E   MESSAGE WITH INSERT (&01)
  ? For more detailed information about the DMS error code enter /HELP-MSG
    in system mode or see the BS2000 manual 'System Messages'.
  ! response for pep0001
-- PEP0002 ---------------------------------------------------------------------
Access     : DLAM
Destination: USER-TASK, CONSOLE   Routing code: * (main console)  Weight: 99
Warranty   : NO                   Text format : UPPER CASE
Insert attributes:
(&01)  Default value: 'E'
E   MESSAGE WITH INSERT (&01)
  ? meaning for pep0002
  ! response for pep0002
-- PEP0005 ---------------------------------------------------------------------
Access     : ISAM
Destination: USER-TASK           Routing code:                    Weight: 99
Warranty   : NO                  Text format : UPPER CASE
D   Datei '(&05)' nicht vorhanden
E   File '(&05)' does not exist
```

```
=========================   D O C U M E N T A T I O N   =====================
PEP0001-PEP0004    Owner name: MAIER                  Team: TEAM1
                  product = DMS-ERR
################################################################################
###################   M e s s a g e   c l a s s :   T S T   ###################
################################################################################
-- TST0004 ---------------------------------------------------------------------
Access     : ISAM
Destination: USER-TASK            Routing code:                    Weight: 99
Warranty   : NO                   Text format : UPPER CASE
E   MESSAGE WITH INSERT (&01)
  ? meaning for pep0004
  ! response for pep0004
```

Ausgabe nach SYSLST

Die Ausgabe des Meldungsdateiinhalte nach SYSLST stimmt nahezu mit der nach SYSOUT überein.

So werden die Meldungen und Dokumentationszeilen identisch ausgegeben. Um die Ausgabe auf Drucker zu steuern, ist die Spalte 1 des SYSLST-Protokolls jedoch für EBCDIC-Steuerzeichen reserviert. Die neue Seite wird mit einer Informationszeile eingeleitet, die Datum, Uhrzeit, Produktname, Produktversion und Seitenzahl enthält.

9 PAMCONV Konvertieren von Dateiformaten

Versionsstand: PAMCONV V21.0C

Das Dienstprogramm PAMCONV dient zur Konvertierung von Dateien im K-Format in das NK-Format oder umgekehrt.

Es werden auch mit Crypto-Kennwort verschlüsselte Dateien unterstützt (siehe "Funktionalität von PAMCONV").

Im **K-Format** wird DVS-Verwaltungsinformation in einem PAM-Schlüssel (PAM-Key) abgelegt, welcher den Datenblöcken vorangestellt ist. Dieses Dateiformat wird als Key-Format, kurz K-Format, bezeichnet.

Im **NK-Format**, das den PAM-Key nicht verwendet, ist die DVS-Verwaltungsinformation entweder in den Datenblock integriert oder sie entfällt. Dieses Dateiformat heißt Non-Key-Format oder kurz NK-Format.

Der Begriff NK-Datei dient als Oberbegriff für NK2- und NK4-Dateien.
Die Dateiformate (K, NK2 und NK4) wurden geschaffen, damit das Datenverwaltungssystem die bestehenden Plattenformate optimal nutzen kann.

Die minimale Transfer-Unit (TU) zwischen Platte und Hauptspeicher sowie die Größe der kleinsten Datei (minimale Allokierungseinheit = min. AU) sind abhängig vom Dateiformat.

Auf NK4-Platten können nur Dateien im NK4-Format liegen. PAMCONV bietet die Möglichkeit, K- oder NK2-Dateien in das NK4-Format zu konvertieren.

Plattenformate

Ein Plattenformat wird durch die Kriterien „mit/ohne PAM-Key“, „minimale Allokierungseinheit (min. AU)“ und „minimale Transferunit (min. TU)“ festgelegt.

Das Plattenformat in einem Pubset ist homogen (außer bei SM-Pubsets).
Für Privatplatten werden nur die Formate K- und NK2-Platte mit einer minimalen Allokierungseinheit von 6 KByte unterstützt.

Folgende Plattenformate werden unterstützt:

Platten-Format	PAM-Key		min. AU			min. TU	
	mit	ohne	6 KB	8 KB	64 KB	2 KB	4 KB
K-Platte	x		x			x	
NK2-Platte		x	x			x	
NK2-Platte		x		x		x	
NK2-Platte		x			x	x	
NK4-Platte		x		x			x
NK4-Platte		x			x		x

Folgendes Schema zeigt, welche Dateiformate auf den unterstützten Plattenformaten abgespeichert werden können, ohne dass vorher eine Konvertierung der Dateiformate mit PAMCONV durchgeführt werden muss.

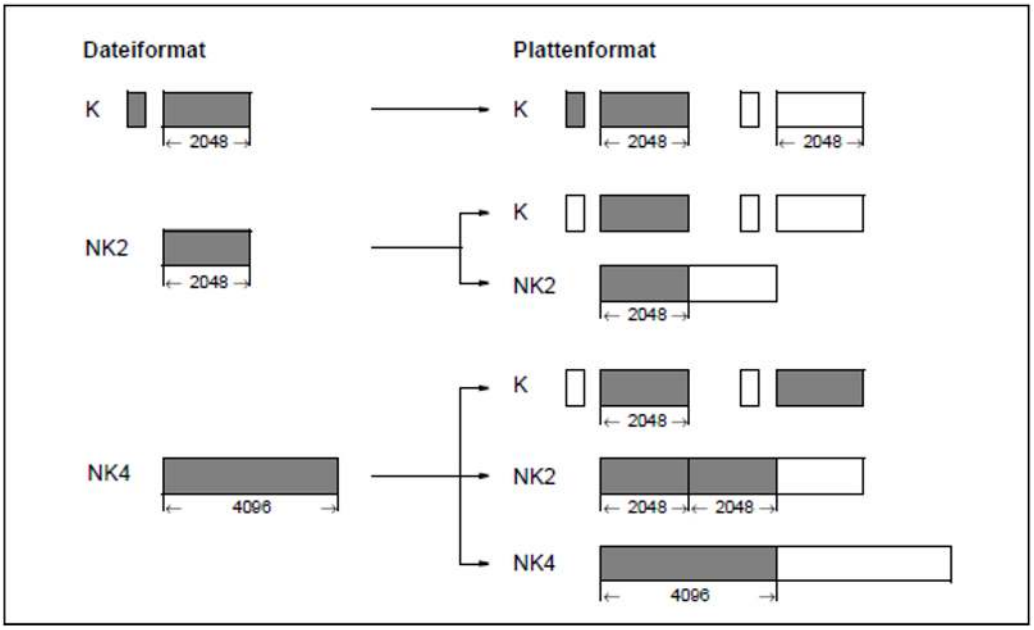
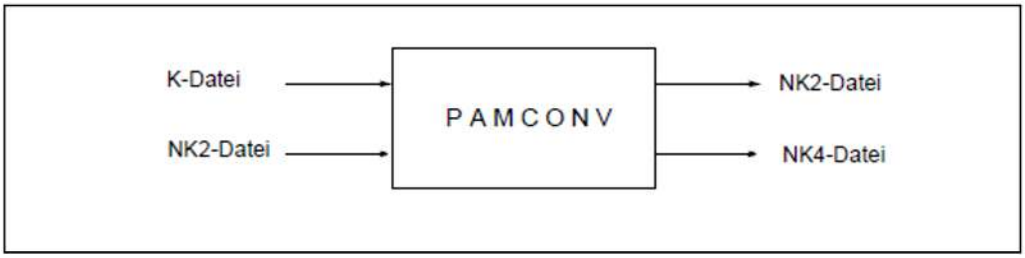


Bild 12: Dateiformate, die ohne Konvertierung auf bestimmte Plattenformate abgespeichert werden können

PAMCONV dient in erster Linie dazu, K-Dateien in NK-Dateien umzuwandeln, um diese auf NK-Pubsets ablegen zu können.



Konvertierungsmöglichkeiten mit PAMCONV

Die Konvertierungsmöglichkeiten, die mit PAMCONV für die einzelnen Dateistrukturen bestehen, sind in nachfolgender Tabelle dargestellt.

Konvertierungsmöglichkeiten mit PAMCONV

Dateistruktur	Dateiformat1 <--> Dateiformat2
ISAM	K-ISAM <---> NK2-ISAM K-ISAM <---> NK4-ISAM NK2-ISAM <---> NK4-ISAM
SAM	K-SAM <---> NK2-SAM K-SAM <---> NK4-SAM NK2-SAM <---> NK4-SAM
PAM	K-PAM <---> NK-PAM K-PAM <---> NK4-PAM NK2-PAM <---> NK4-PAM
Phase	K-Phase <---> NK2-Phase K-Phase <---> NK4-Phase NK2-Phase <---> NK4-Phase

9.1 Programmlauf starten

Das Programm PAMCONV wird gestartet mit `/START-PAMCONV.`

START-PAMCONV	Alias: **PAMCONV**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Nach Programmaufruf können die Steueranweisungen des Programms eingegeben werden.

Mit der Anweisung END wird PAMCONV beendet.

Alternativ ist der Aufruf wie folgt möglich:

`/START-EXECUTABLE-PROGRAM FROM-FILE=$PAMCONV`

9.2 Funktionalität von PAMCONV

Mit dem Dienstprogramm PAMCONV werden dem Anwender zwei grundlegende Funktionen zur Verfügung gestellt, die ihm ermöglichen, seine Dateien dem vorhandenen Plattenformat anzupassen.

- Konvertierung des Dateiformats

  Mit PAMCONV können Dateien vom K-Format in das NK-Format konvertiert werden (und umgekehrt). Über Anweisungen, im Speziellen die Anweisung CONVERT-FILE, wird die Konvertierung des Dateiformats abgewickelt.
- Umblockung

  Mit PAMCONV kann der Blockungsfaktor einer Datei von ungeradzahlig in geradzahlig umgewandelt werden. Nur Dateien mit geradzahligem Blockungsfaktor können auf einer NK4-Platte abgespeichert werden.

Funktionalität von PAMCONV

Die nachfolgenden Tabellen geben einen Überblick über die Konvertierungsmöglichkeiten für ISAM-, SAM-, PAM- und Phasen-Dateien.

Ferner sind zu jeder Dateistruktur die grundlegenden Konvertierungsrichtungen schematisch dargestellt. Die Blockstruktur der Datei wird hierbei vor und nach der Konvertierung gezeigt.

ISAM-Datei

Eine ISAM-Datei, die mit PAMCONV konvertiert werden soll, kann in den folgenden drei Blockstrukturen vorliegen.

- **PAMKEY**: eine K-ISAM-Datei liegt vor.
- **DATA2K**: eine NK2-ISAM-Datei liegt vor; der Blockungsfaktor n kann geradzahlig oder ungeradzahlig sein.
- **DATA4K**: eine NK4-ISAM-Datei liegt vor; der Blockungsfaktor n ist geradzahlig.

Die nachfolgende Tabelle fasst alle Konvertierungsmöglichkeiten für eine ISAM-Datei zusammen:

Zusammenstellung der Konvertierungsmöglichkeiten für eine ISAM-Datei

Quelle	Richtung	Ziel
PAMKEY	------>	DATA2K
PAMKEY	------>	DATA4K
DATA2K	------>	PAMKEY
DATA2K	------>	DATA4K
DATA4K	------>	PAMKEY
DATA2K	<------>	DATA2K
DATA4K	<------>	DATA4K
DATA4K	------>	DATA2K

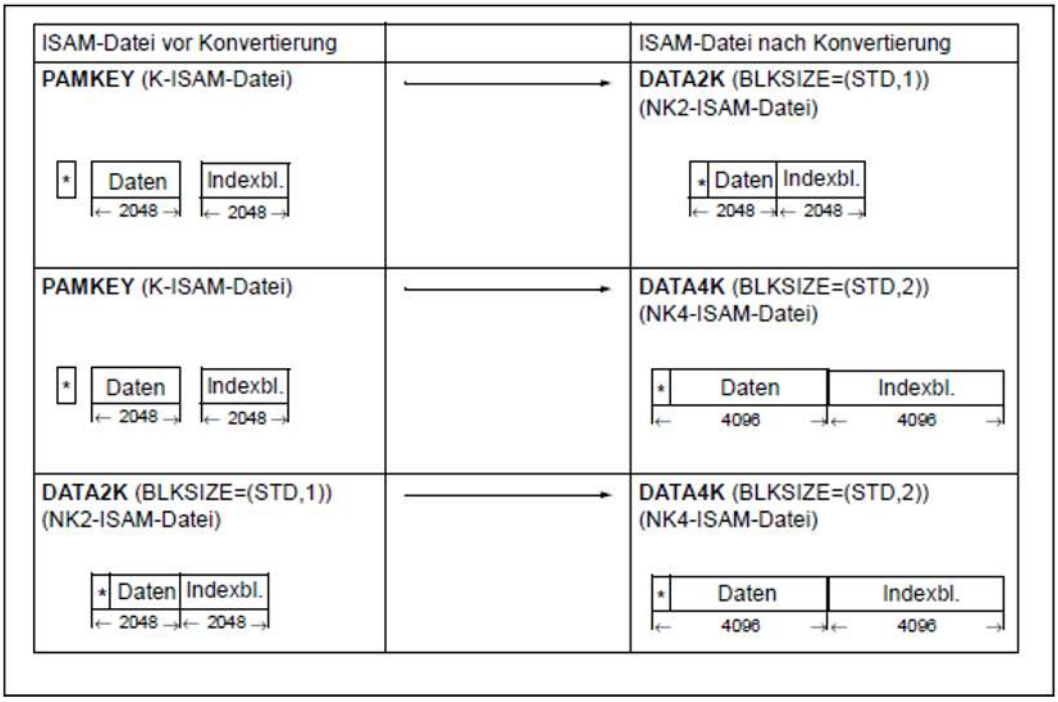
MLU-Makrobibliotheken sind ISAM-Dateien und sind wie solche zu konvertieren.

Für folgende Konvertierungsrichtungen soll anschaulich gemacht werden, welche Blockstruktur eine ISAM-Datei vor und nach der Dateiformatkonvertierung aufweist:

Quelle	Richtung	Ziel
PAMKEY	------>	DATA2K
PAMKEY	------>	DATA4K
DATA2K	------>	DATA4K

Es bedeuten:

*	16 Byte Blockverwaltungsinformation
BLKSIZE	logische Blocklänge
Indexbl.	Indexblock



SAM-Datei

Eine SAM-Datei, die mit PAMCONV konvertiert werden soll, kann in den folgenden zwei Blockstrukturen vorliegen:

- **PAMKEY**: eine K-SAM-Datei liegt vor.
- **DATA**: eine NK2-SAM-Datei liegt vor, wenn der Blockungsfaktor n ungeradzahlig ist; eine NK4-SAM-Datei, wenn n geradzahlig ist.

Die nachfolgende Tabelle fasst alle Konvertierungsmöglichkeiten für eine SAM-Datei zusammen:

Zusammenstellung der Konvertierungsmöglichkeiten für eine SAM-Datei

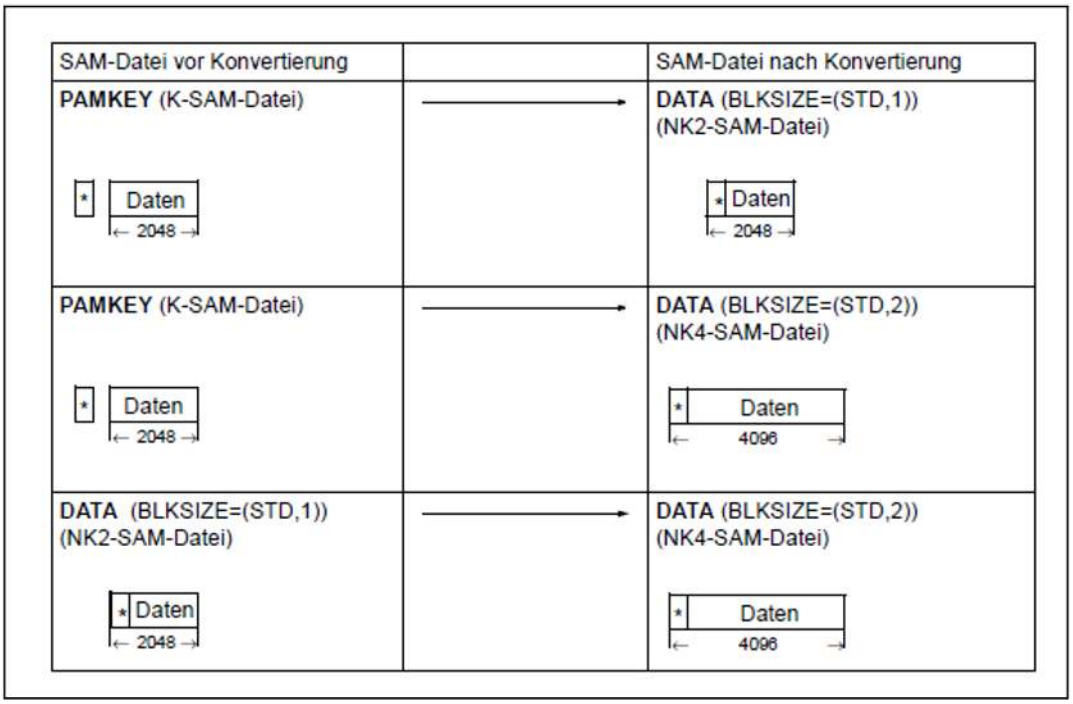
Quelle	Richtung	Ziel
PAMKEY	------>	DATA
DATA	------>	PAMKEY
DATA	<------>	DATA

Für folgende Konvertierungsrichtung soll anschaulich gemacht werden, welche Blockstruktur eine SAM-Datei vor und nach der Dateiformatkonvertierung aufweist:

Quelle	Richtung	Ziel
PAMKEY	------>	DATA

Es bedeuten:

*	16 Byte Blockverwaltungsinformation
BLKSIZE	logische Blocklänge; bei NK-SAM-Dateien werden die 16 Byte nur einmal pro logischer Blocklänge abgezogen



PAM-Datei

Eine PAM-Datei, die mit PAMCONV konvertiert werden soll, kann in den folgenden drei Blockstrukturen vorliegen:

- **PAMKEY**: eine K-PAM-Datei liegt vor.
- **DATA**: eine NK2-PAM-Datei liegt vor, wenn der Blockungsfaktor n ungeradzahlig ist; eine NK4-PAM-Datei, wenn n geradzahlig ist.
- **NO**: eine NK2-PAM-Datei liegt vor, wenn der Blockungsfaktor n ungeradzahlig ist; eine NK4-PAM-Datei, wenn n geradzahlig ist. Es wird keine Blockverwaltungsinformation abgespeichert.

Die nachfolgende Tabelle fasst alle Konvertierungsmöglichkeiten für eine PAM-Datei zusammen.

Zusammenstellung der Konvertierungsmöglichkeiten für eine PAM-Datei

Quelle	Richtung	Ziel
PAMKEY	------>	NO
NO	------>	PAMKEY
NO	<------>	NO
DATA	<------>	DATA

PAM-DATA-Dateien können nicht in das K-Format konvertiert werden. Wird trotzdem die Konvertierungsrichtung NONKEY-TO-KEY gewählt, wird die Verarbeitung mit der Meldung `PEA2212` abgewiesen.

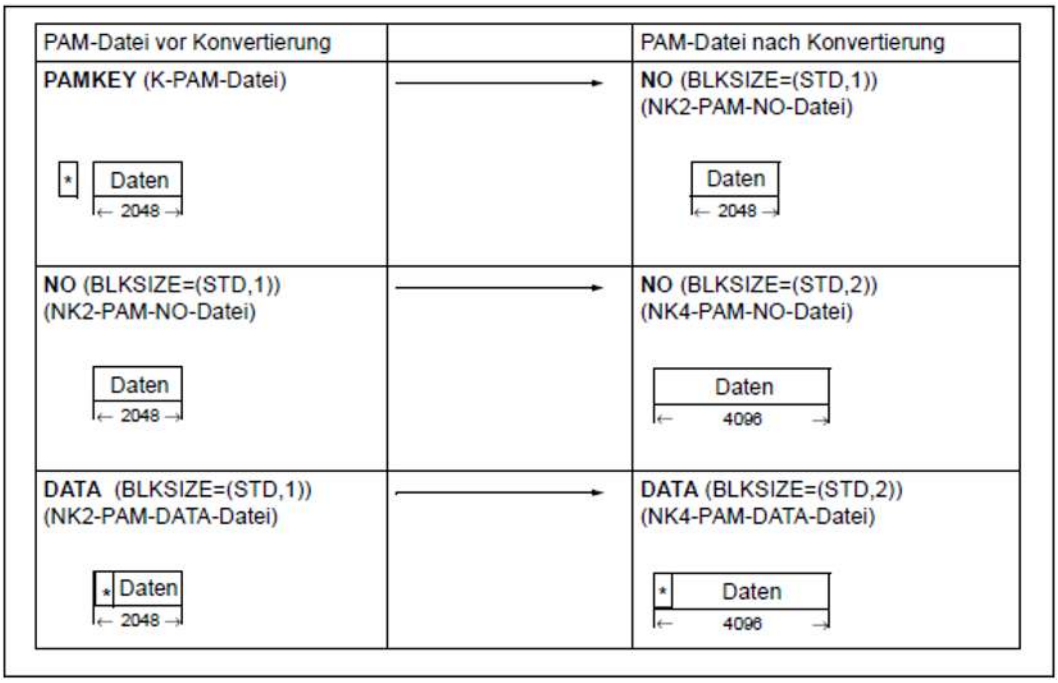
PLAM-Bibliotheken sind PAM-Dateien, die den PAM-Schlüssel nicht verwenden. Sie sind konvertierbar.

Für folgende Konvertierungsrichtungen soll anschaulich gemacht werden, welche Blockstruktur eine PAM-Datei vor und nach der Dateiformatkonvertierung aufweist:

Quelle	Richtung	Ziel
PAMKEY	------>	NO
NO	------>	NO
DATA	------>	DATA

Es bedeuten:

*	16 Byte Blockverwaltungsinformation
BLKSIZE	logische Blocklänge



Phasen-Datei

Phasen (Lademodule) sind spezielle PAM-Dateien. In den PAM-Dateinamen (K-PAM-Datei oder NK-PAM-Datei) wird die Angabe „PAM“ durch „Phase“ ersetzt. Somit unterscheidet man K-, NK2- und NK4-Phasen-Dateien.

NK2-Phasen-Dateien besitzen das Dateiformat mit der Blockstruktur NO, d.h. es wird keine Blockkontrollinformation abgespeichert. Die logische Blocklänge beträgt (STD,1), also 2048 Bytes.
Bei NK4-Phasen-Dateien beträgt die logische Blocklänge (STD,2), also 4096 Bytes. Alle anders lautenden Angaben zur Blocklänge werden mit einer Fehlermeldung abgewiesen.

Konvertierungsmöglichkeiten für eine Phasen-Datei

Quelle	Richtung	Ziel
PAMKEY	<------>	NO
NO	<------>	NO

Mit Crypto-Kennwort verschlüsselte Quelldateien

Es werden mit Crypto-Kennwort verschlüsselte Dateien unterstützt.

Die Quelldatei wird zunächst auf Verschlüsselung geprüft. Ist sie verschlüsselt, wird von PAMCONV weiter geprüft, ob das entsprechende Crypto-Kennwort in der tasklokalen Crypto-Kennwort-Tabelle eingetragen ist. (Der Eintrag erfolgt mit `/ADD-CRYPTO-PASSWORD`.) Wenn nicht, beendet PAMCONV die Konvertierung mit Fehlermeldung.

Ist die Quelldatei verschlüsselt und das entsprechende Kennwort in der Crypto-Kennwort-Tabelle eingetragen, erhält die Zieldatei eine identische Verschlüsselung.

In folgenden Fällen findet keine Übernahme der Verschlüsselung für die Zieldatei statt:

- Die Zieldatei liegt unter der Benutzerkennung TSOS auf dem Home-Pubset.
- Die Quelldatei ist eine einzelne Dateigeneration.
- Die Konvertierung erfolgt über eine Zwischendatei (siehe „Konvertierung über Zwischendatei“ (Konvertierungsarten)).

9.3 Konvertieren von Dateiformaten

- Konvertierungsarten
- Anforderungen an die Systemumgebung
- Angabe der Quelldateien und Zieldateien

9.3.1 Konvertierungsarten

Normalkonvertierung

Es wird davon ausgegangen, dass sich Quell- und Zieldatei bei einer Konvertierung auf gemeinschaftlichen Datenträgern befinden.

Diesen Fall könnte man als Regelfall bezeichnen. Zur Konvertierung muss der Platz für beide Dateien verfügbar sein. Ist dies nicht der Fall, so muss eine Konvertierung über eine Zwischendatei vorgenommen werden.

Konvertierung über Zwischendatei

- Allgemeines

  Für die Konvertierung einer Datei gilt, dass die Zieldatei den etwa gleichen Speicherbedarf auf Magnetplatte benötigt wie die Quelldatei. Eine „in-sich“- Konvertierung, die ohne zusätzlichen Platzbedarf auskommt, ist mit PAMCONV nicht möglich. Das bedeutet, dass für die Zieldatei noch genügend Plattenspeicherplatz zur Verfügung stehen muss. Ist dies aber nicht der Fall, ist die Konvertierung unter Verwendung einer Zwischendatei auf Magnetband oder Privatplatte möglich.

  Der Operand FILE-DISPOSAL der Anweisung CONVERT-FILE (gibt an, wie mit der erzeugten Datei nach der Konvertierung verfahren werden soll) wird bei Konvertierung über Zwischendatei nicht berücksichtigt.

  Bei dieser Konvertierungsart wird bereits beim Erstellen der Zwischendatei geprüft, ob die Datei konvertierbar ist. Dies soll verhindern, dass sich eine Zwischendatei nicht mehr in die Zieldatei umsetzen lässt.

- Konvertierung über Zwischendatei in zwei Schritten

  Diese Art der Konvertierung ist explizit durch zwei CONVERT-FILE-Anweisungen vorzunehmen.

  Die beiden Anweisungen können im selben Programmlauf oder in getrennten Programmläufen gegeben werden.

  - Erster Schritt
    Konvertierung der Quell-(Magnetplatten)-Datei auf Band oder Privatplatte.
  - Zweiter Schritt
    Konvertierung der Zwischendatei von Magnetband oder Privatplatte auf die Zieldatei.

- Konvertierung über Zwischendatei in einem Schritt

  Die oben angeführte Dateikonvertierung in zwei Schritten wird zu einer Aufgabe zusammengefasst und nur durch eine CONVERT-FILE-Anweisung ausgeführt. Dies kann erreicht werden, wenn für den Operanden DEVICE-FOR-TEMPFILE der Wert DISK oder TAPE angegeben wird.

  Nach der erfolgreichen Umsetzung der Quell- auf die Zwischendatei wird die Quelldatei gelöscht und somit Platz für die Zieldatei geschaffen.

  Für diese Zwischendatei wird ein Dateiname mit folgendem Aufbau gebildet:

  `SYSTMP.<tsn>.PAMCONV.<ss>.<cpusec>`

  Die Zwischendatei wird nach erfolgreicher Umsetzung auf die Zieldatei gelöscht, andernfalls kann die Zwischendatei über diesen Namen für eine weitere Verarbeitung angesprochen werden.

  > **i** Programmintern wird die Konvertierung in den oben erwähnten zwei Schritten durchgeführt. Der Benutzer wird jedoch erst wieder zur Eingabe von Anweisungen aufgefordert, wenn beide Konvertierungsschritte durchgeführt worden sind.

Spezifikation der Konvertierungsvarianten

Die Auswahl der Konvertierungsvarianten wird auf Grund der Spezifikation der Quell- und Zieldatei und der Angabe im Operanden DEVICE-FOR-TEMPFILE der Anweisung CONVERT-FILE vorgenommen:

- DEVICE-FOR-TEMPFILE=*NONE

  Es wird keine Zwischendatei auf einem privaten Datenträger erzeugt, außer sie wird durch `/ADD-FILE-LINK` vereinbart.

  Es sind folgende Varianten möglich:

  - Die Quelldatei ist als Magnetplattendatei spezifiziert *und* für die Zieldatei existiert weder ein Katalogeintrag noch ist sie durch `/ADD-FILE-LINK` spezifiziert oder die Zieldatei ist als Magnetplattendatei angegeben:

    Konvertierung von Quelldatei auf Magnetplatte nach Zieldatei auf Magnetplatte. (Normalkonvertierung)
  - Die Quelldatei ist als Magnetplattendatei und die Zieldatei ist als Magnetbanddatei angegeben:

    Konvertierung von Quelldatei auf Magnetplatte nach Zwischendatei auf Magnetband. (Konvertierung über Zwischendatei auf Magnetband in zwei Schritten: Erster Schritt.)
  - Die Quelldatei ist als Magnetbanddatei spezifiziert *und* für die Zieldatei existiert weder ein Katalogeintrag noch ist sie durch `/ADD-FILE-LINK` spezifiziert oder die Zieldatei ist als Magnetplattendatei angegeben:

    Konvertierung von Quelldatei auf Magnetband (muss eine von PAMCONV erzeugte Zwischendatei sein) nach Zieldatei auf Magnetplatte.
    (Konvertierung über Zwischendatei auf Magnetband in zwei Schritten: Zweiter Schritt.)

  Die Spezifizierung von Quell- und Zieldatei als Magnetbanddatei ist nicht zulässig.

  > **i** Dies bedeutet auch, dass eine Datei auf Magnetband immer eine PAMCONV-Zwischendatei sein muss, andernfalls wird die Konvertierung abgewiesen.

- DEVICE-FOR-TEMPFILE=*TAPE

  Es wird in jedem Fall eine Zwischendatei auf einem Magnetband erzeugt. Eine gleichzeitige Spezifizierung der Quell- und/oder Zieldatei durch `/ADD-FILE-LINK` als Magnetbanddatei ist nicht zulässig.

  Es sind folgende Varianten möglich:

  - Die Quelldatei ist als Magnetplattendatei spezifiziert *und* für die Zieldatei existiert weder ein Katalogeintrag noch ist sie durch `/ADD-FILE-LINK` spezifiziert oder
  - die Zieldatei ist als Magnetplattendatei spezifiziert:

    Konvertierung von Quelldatei auf Magnetplatte nach Zieldatei auf Magnetplatte. (Konvertierung über Zwischendatei auf Magnetband in einem Schritt.)

- DEVICE-FOR-TEMPFILE=*DISK

  Es wird in jedem Fall eine Zwischendatei auf einer Privatplatte erzeugt. Eine gleichzeitige Spezifizierung der Quell- und/oder Zieldatei durch `/ADD-FILE-LINK` als Magnetbanddatei ist nicht zulässig. Wird die Quell- und /oder Zieldatei durch `/ADD-FILE-LINK` (auf Privatplatte) spezifiziert, darf der in diesem Kommando angegebene Datenträger nicht gleich mit dem im Operanden DEVICE-FOR-TEMPFILE angegebenen Datenträger der Privatplatte sein.

  Es sind folgende Varianten möglich:

- Die Quelldatei ist als Magnetplattendatei spezifiziert *und* für die Zieldatei existiert weder ein Katalogeintrag noch ist sie durch `/ADD-FILE-LINK` spezifiziert oder
- die Zieldatei ist als Magnetplattendatei spezifiziert:

  Konvertierung von Quelldatei auf Magnetplatte nach Zieldatei auf Magnetplatte. (Konvertierung über Zwischendatei auf Magnetplatte in einem Schritt.)

Format der Zwischendatei auf Magnetband

Die Quell-(Magnetplatten)-Dateien können SAM-, ISAM-, oder PAM-Dateien sein.

Die Zugriffsmethode ISAM ist für Bänder nicht definiert. Daher wird gleich ein einheitliches Format für die Zwischendatei auf Band verwendet. Sie ist eine SAM-Datei und enthält die Datensätze der Quelldatei (ISAM: aufsteigend, nach Schlüsseln sortiert). Ein Satz besteht bei allgemeinen PAM-Dateien und Phasen aus einem 8 Byte langen Feld mit dem User-Teil des PAM-Schlüssels und einem 2048 Byte langen Feld mit dem PAM-Block.

Zusätzliche Dateimerkmale werden in einem eigenen Benutzerkennsatz (UHL) abgelegt.

Eine solche Zwischendatei ist nur als Zwischendatei für eine Konvertierung gedacht. Es müssen Magnetbänder mit Standardkennsätzen verwendet werden.

- Konvertierung über Zwischendatei auf Band in zwei Schritten

  Sollen mehrere Dateien gleichzeitig konvertiert werden, so muss `/ADD-FILE-LINK` mit SUPPORT=TAPE(FILE-SEQUENCE =...) eingegeben werden, da sonst die Zwischendatei auf Band überschrieben wird.
- Dateiattribute

  Eine Zwischendatei auf Magnetband hat folgende Eigenschaften:

```
FCBTYPE = SAM
BLKSIZE = (STD,16)
RECFORM = VARIABLE
LABEL   = STD
BLKCTRL = PAMKEY
```

  Im Übrigen gelten die Standardwerte aus dem Makro FCB.

  Wie aus folgendem Diagramm ersichtlich, ist eine Zwischendatei auf Magnetband immer eine K-SAM-Datei, unabhängig vom Key-Format oder FCBTYPE der Quell- und Zieldatei.

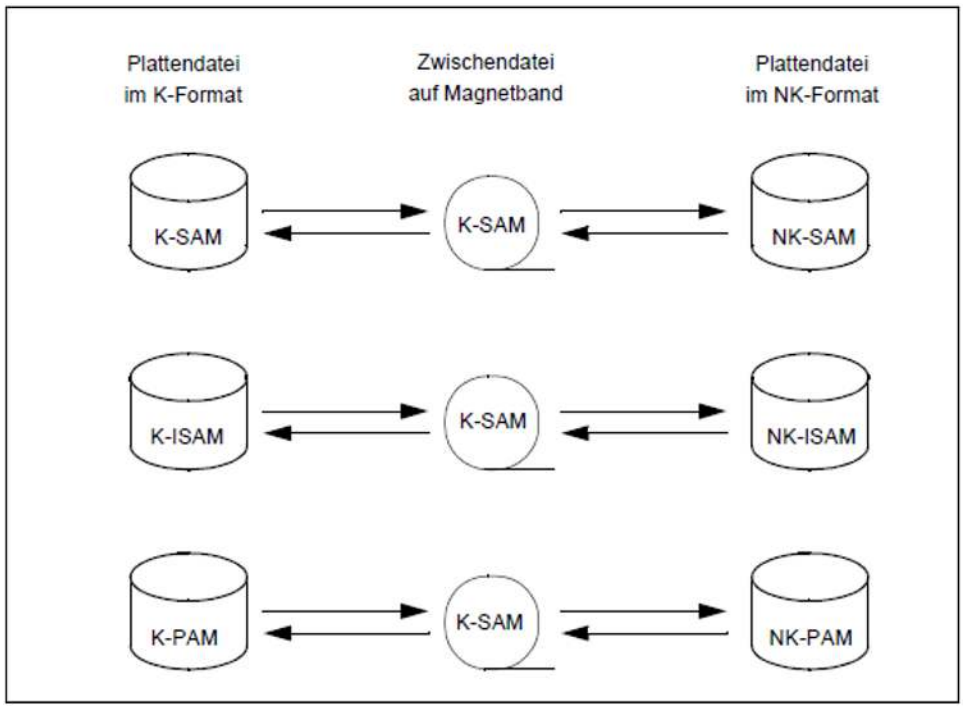


Bild 13: Zwischendatei auf Magnetband

- Benutzerkennsätze

  Für die Speicherung der Dateieigenschaften der Quelldatei wird der Benutzerkennsatz UHL1 mit folgendem Aufbau verwendet:

Stellen	Inhalt	Bedeutung
1..4	UHL1	Kennzeichen des Benutzerkennsatzes
5..12	PAMELA-I	Kennzeichen: ISAM-Zwischendatei
5..12	PAMELA-S	Kennzeichen: SAM-Zwischendatei
5..12	PAMELA-P	Kennzeichen: PAM-Zwischendatei
13..66	CL54	Dateiname der Quelldatei
67	AL1(b)	BLKSIZE=(STD,b)
68	X'02'	RECFORM=VARIABLE
68	X'04'	RECFORM=FIXED
69..70	AL2(r)	RECSIZE=r
71..72	AL2(p)	KEYPOS=p
73	AL1(k)	KEYLEN=k
74	X'00'	VALPROP=MIN
74	X'01'	VALPROP=MAX

75	AL1(f1)	LOGLEN=f1
76	AL1(f2)	VALLEN=f2
77	X'80'	DUPEKY=YES
77	X'00'	DUPEKY=NO
78	AL1	Druckersteuerzeichen (IDCEDMS)
79	CL1	Bibliothekstyp ('P': PLAM)
80	AL1	Datei-Indikator (X'40': FGG)

Format der Zwischendatei auf Privatplatte

Eine Zwischendatei auf Privatplatte besitzt (im Gegensatz zur Zwischendatei auf Magnetband) kein eigenes Format. Sie ist immer eine Kopie der Quell- bzw. Zieldatei, abhängig von der Konvertierungsrichtung. Die Zwischendatei wird immer als NK-Datei erzeugt. Daher ist man vom Key-Format der Privatplatte unabhängig.

> **i** Bei SAM-Dateien kann sich die Kopie (Zwischendatei) von der NK-Quelldatei im RECSIZE-Wert unterscheiden. Wenn der RECSIZE-Wert der NK-Quelldatei null ist, wird für die Zwischendatei der maximal mögliche Wert eingesetzt.

Aus folgendem Diagramm ist ersichtlich, in welchem Format die Zwischendatei auf der Privatplatte eingerichtet wird und von welcher Datei (abhängig von der Konvertierungsrichtung) die Kopie erstellt wird.

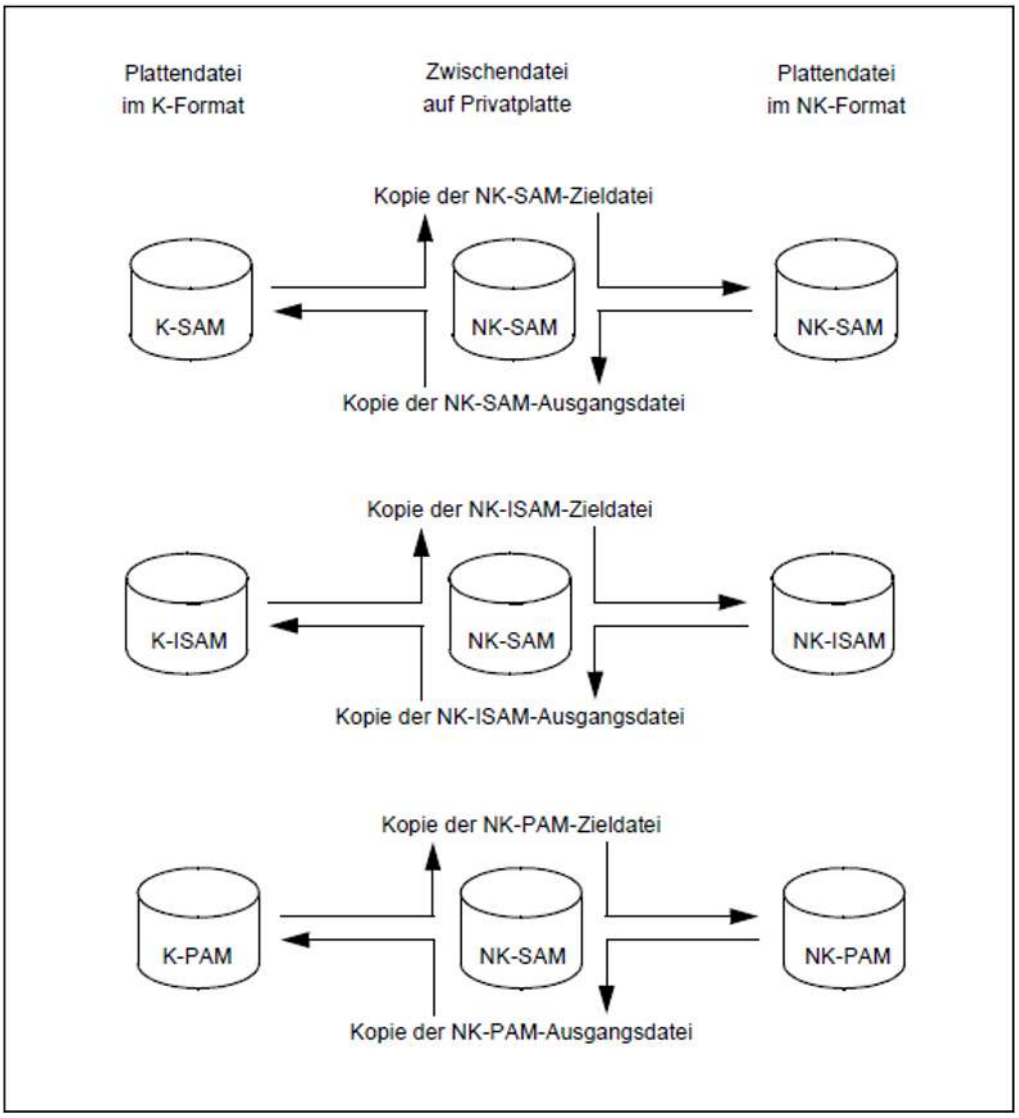


Bild 14: Zwischendatei auf Privatplatte

9.3.2 Anforderungen an die Systemumgebung

Bandperipherie

Magnetbänder, die Zwischendateien aufnehmen sollen, müssen mit Standardetiketten versehen sein.

Plattenperipherie

Ist in einem System eine reine NK-Plattenperipherie (ohne K-Platten) vorhanden, so ist die Erzeugung von K-Plattendateien nicht möglich. Das bedeutet auch, dass die Konvertierung von NONKEY-TO-KEY in solchen Systemen nicht möglich ist.

Sollen ganz allgemein Quell- und Zieldatei innerhalb eines Pubsets liegen, so ist darauf zu achten, dass Platz für beide Dateien während der Konvertierung vorhanden ist. Die Zieldatei kann mehr Platz als die Quelldatei benötigen. Ist diese Voraussetzung nicht gegeben, so sollte die Konvertierung über eine Zwischendatei durchgeführt werden.

ACS (Alias Catalog Service)

Bei Angabe von Quell- und Zieldateien für die Konvertierung können Aliasnamen verwendet werden. Diese Namen müssen exakt so angegeben werden, wie sie im Alias-Katalog eingetragen sind. Nur so kann die Verbindung zum richtigen Dateinamen hergestellt werden.

Diagnoseunterlagen

Treten in einem PAMCONV-Lauf Probleme auf (unvorhergesehene Meldungen, Dump, ...), werden folgende Unterlagen für die Diagnose benötigt:

- SYSLST-Protokoll
  Das Protokoll soll das Umfeld des PAMCONV-Laufs zeigen (z.B. Eingaben, die unmittelbar vor dem Fehlerfall durchgeführt wurden).
- Eingabedatei, bei der der Fehler auftrat. Die Entwicklung hat damit die Möglichkeit, die Ursache des Problems zu finden, die eventuell erst viel später zum Abbruch führt.

9.3.3 Angabe der Quelldateien und Zieldateien

Quelldatei und zugehörige Zieldatei können jeweils auf drei Arten spezifiziert werden:

Angabe durch vollqualifizierten Dateinamen

Quell- und Zieldatei können durch Angabe eines vollqualifizierten Dateinamens ausgewählt werden. Auf diese Art kann auch eine einzelne Dateigeneration angegeben werden.

Bei der Quelldatei werden in diesem Falle die Dateimerkmale programmintern dem Katalogeintrag entnommen.

Bei der Zieldatei sind die Dateimerkmale der Quelldatei auch für die Zieldatei maßgebend.

Es werden dabei folgende Dateimerkmale erfasst (vergleiche `/ADD-FILE-LINK`):

- Zugriffsmethode
- Satzformat
- Satzlänge
- Datenblocklänge
- Schlüssellänge in ISAM-Dateien
- Schlüsselposition in ISAM-Dateien
- Mehrfachschlüssel in ISAM-Dateien
- Länge der logischen Markierung in ISAM-Dateien
- Länge der Wertmarkierung in ISAM-Dateien
- Zeichensatz

Angabe durch Auswahlkriterien

Die Quelldatei wird teilqualifiziert mit oder ohne zusätzliche Auswahlkriterien (Operand SELECT=*BY-ATTRIBUTES (...) der Anweisung CONVERT-FILE mit CREATION-DATE, LAST-ACCESS-DATE, SIZE, FILE-STRUCTURE, BLKSIZE, BLKCTRL) angegeben.

Aus diesen Angaben werden alle umzusetzenden Dateien ermittelt (vergleiche `/SHOW-FILE-ATTRIBUTES`, Handbuch „Kommandos“ [1]).

Für die Zieldateien ist die Benennung nach den Bildungsregeln mit Musterzeichen möglich (siehe `/SHOW-FILE-ATTRIBUTES`)

Für die übrigen Dateimerkmale der Zieldatei sind die Dateimerkmale der Quelldatei maßgebend.

> **i** Wird bei der Angabe der Auswahlkriterien bezüglich PAM-Dateien nicht darauf geachtet, dass keine LMR-Modulbibliotheken ausgewählt werden, wäre mit der Zerstörung der Modulbibliotheken zu rechnen.
> Um dies zu vermeiden, wird vor der Konvertierung einer PAM-Datei eine Prüfung durchgeführt. LMR-Modulbibliotheken werden dann erkannt und nicht konvertiert. LMR-Modulbibliotheken können mit dem ISP-LMS (/EXEC $LMS) umgesetzt werden.
>
> PAM-Dateien, die Phasen sind, werden als solche erkannt und in das entsprechende Phasendateiformat konvertiert.

Angabe einer Referenz über einen Dateikettungsnamen

Die Referenz wird hergestellt über den Dateikettungsnamen zu einem vorher eingegebenen `/ADD-FILE-LINK`, das die Datei spezifiziert.

Der Dateikettungsname wird im Operanden *LINK(LINK-NAME=...) der Anweisung CONVERT-FILE angegeben.

Über eine Dateikettungsnamen kann auch eine einzelne Dateigeneration ausgewählt werden.

Die Angabe eines Dateikettungsnamen ist notwendig, wenn für die Zieldatei bestimmte Eigenschaften spezifiziert werden sollen, die nicht aus dem Katalogeintrag der Quelldatei bestimmt werden können. Dies sind insbesondere:

- SPACE-Angaben
- Index-/Daten-Trennung (nur für K-ISAM Dateien)
- PAD-Faktor (standardmäßig wird Defaultwert verwendet)
- RETPD-Angabe
- WROUT=NO (standardmäßig WROUT=YES verwendet)
- WRCHK=YES (standardmäßig WRCHK=NO verwendet)

9.4 Umblockung

Auf NK4-Platten können nur Dateien mit geradzahligem Blockungsfaktor abgelegt werden. Um folglich Dateien mit ungeradzahligem Blockungsfaktor auf NK4-Platten speichern zu können, bietet PAMCONV die Funktion „Umblockung“. Im Operanden TO-FILE-BLKSIZE in den Anweisungen CONVERT-FILE, MODIFY-CONVERT-FILE-DEFAULTS und SHOW-CONVERT-FILE-DEFAULTS ist diese Funktion realisiert.

9.4.1 Explizite Umblockung

Die Umblockung wird vom Anwender im Operanden TO-FILE-BLKSIZE veranlasst. Er legt hiermit die logische Blockgröße der Zieldatei fest. Ist die Blockgröße der Quelldatei ungerade und wird die Blockgröße der Zieldatei explizit angegeben, kommt es bei Eröffnung der Datei auf einer NK4-Platte zur Ausgabe einer Fehlermeldung.

9.4.2 Implizite Umblockung

Durch die Angabe TO-FILE-BLKSIZE=*STD/*NK4 übernimmt PAMCONV selbst die Umblockung.

- Implizites Heraufsetzen des Blockungsfaktors bei TO-FILE-BLKSIZE=*STD

  PAMCONV richtet sich nach den Gegebenheiten des Zielpubsets und führt nötigenfalls eine implizite Erhöhung des Blockungsfaktors durch.
  Bei NK2-Pubsets bleibt die Blockgröße unverändert, bei NK4-Pubsets wird bei ungeradzahligem Blockungsfaktor der Quelldatei der Blockungsfaktor der Zieldatei erhöht.

- Implizites Heraufsetzen des Blockungsfaktors bei TO-FILE-BLKSIZE=*NK4

  Unabhängig vom Zielpubset wird bei ungeradzahligem Blockungsfaktor der Quelldatei der Blockungsfaktor der Zieldatei erhöht.

Implizites Herabsetzen des Blockungsfaktors wird von PAMCONV nicht vorgenommen.

9.4.3 Umblockung von PAM-DATA-Dateien ohne Änderung des Dateiformats

Eine weitere Funktion von PAMCONV ist die Umblockung von NK-PAM-DATA-Dateien ohne Änderung des Datenformats, d.h. Quell- und Zieldatei besitzen die Eigenschaft BLKCTRL=DATA. Umblocken ohne zu konvertieren wird durch die Angabe DIRECTION=*TO-NONKEY und durch den Operanden TO-FILE-BLKSIZE unterstützt.

Ein **Heraufsetzen des Blockungsfaktors** bei PAM-DATA-Dateien erfolgt nur, wenn die Division der Blockgröße der Zieldatei (TO-FILE-BLKSIZE) durch die Blockgröße der Quelldatei keinen Rest ergibt. Also Modulo(Ziel-BLKSIZE / Quell-BLKSIZE) = 0. Ist der Divisionsrest ungleich null, wird die Umblockung mit einer Fehlermeldung abgebrochen.

Beim Heraufsetzen des Blockungsfaktors von PAM-DATA-Dateien wird nur das Blockkontrollfeld des jeweils ersten Blockes innerhalb eines logischen Blockes in die Zieldatei geschrieben. Alle anderen Kontrollfelder, die im selben logischen Block liegen, werden mit X'00' in der Länge von 12 Byte belegt.

Ein **Herabsetzen des Blockungsfaktors** bei PAM-DATA-Dateien erfolgt nur, wenn die Division der Blockgröße der Quelldatei durch die Blockgröße der Zieldatei keinen Rest ergibt. Also Modulo(Quell-BLKSIZE / Ziel-BLKSIZE) = 0. Ist der Divisionsrest ungleich null wird die Umblockung mit einer Fehlermeldung abgebrochen.

Beim Herabsetzen des Blockungsfaktors wird geprüft, ob die Kontrollfelder den Inhalt X'00' besitzen. Wenn nicht, wird die Umblockung mit einer Fehlermeldung abgebrochen.
Der Blockungsfaktor wird nur herabgesetzt, wenn der Blockungsfaktor der Datei schon einmal erhöht wurde.

Besitzt die PAM-DATA-Datei als Quelldatei die Blockgröße BLKSIZE=(STD,1), führt PAMCONV eine implizite Umblockung durch. Es wird eine Warnung ausgegeben, da die Dateistruktur geändert wird.

Einschränkungen

- NK2-PAM-DATA-Dateien mit BLKSIZE größer als (STD,8) können von PAMCONV nicht umgesetzt werden.
- Die Funktion „Umblockung“ von PAM-DATA-Dateien wird nur für die Konvertierungsrichtung TO-NONKEY durchgeführt. Bei Angabe von NONKEY-TO-KEY wird mit der Meldung PEA2212 abgebrochen, da PAM-DATA-Dateien nicht nach KEY konvertiert werden können.

9.4.4 Schwierigkeit beim Herabsetzen des Blockungsfaktors

Die Satzlänge (RECSIZE) der Quelldatei kann durch Herabsetzen des Blockungsfaktors die Blockgröße der Zieldatei überschreiten.

- Feste Satzlänge (RECFORM=F)

  PAMCONV prüft die Satzlänge der Quelldatei und vergleicht diese mit der Satzlänge der Zieldatei, die aus der Blockgröße der Zieldatei berechnet wurde. Ist die Satzlänge der Quelldatei größer als die der Zieldatei, wird die Verarbeitung mit einer Meldung abgebrochen.
- Variable oder unbestimmte Satzlänge (RECFORM=V/U)

  PAMCONV nimmt an, dass die Satzlänge der Zieldatei nicht überschritten wird, und startet die Verarbeitung. Sollte die Satzlänge der Zieldatei trotzdem überschritten werden, wird PAMCONV vom DVS darüber informiert und die Verarbeitung wird mit einer Meldung abgebrochen.

Beim Heraufsetzen des Blockungsfaktors treten keine Schwierigkeiten mit der Satzlänge der Zieldatei auf.

9.5 Durchführung der Konvertierung und Umblockung

Die Konvertierung und Umblockung von Dateien wird über die Anweisung CONVERT-FILE gesteuert. Weitere PAMCONV-Anweisungen dienen der Einstellung oder Abfrage der vom Benutzer definierten PAMCONV-Umgebung.

Im Folgenden wird auf Besonderheiten bei der Konvertierung der Dateistrukturen ISAM und PAM und des Konvertierungsablaufs hingewiesen.

9.5.1 Besonderheiten bei der Konvertierung

Besonderheiten bei der Konvertierung von ISAM-Dateien

Über den Operanden TO-FILE-BLKCTRL der Anweisungen CONVERT-FILE oder MODIFY-CONVERT-FILE-DEFAULTS kann für ISAM-Dateien das Dateiformat festgelegt werden, wenn die Konvertierungsrichtung TO-NONKEY lautet.
Für diesen speziellen Fall sind einige Einschränkungen bezüglich der Verträglichkeit zwischen der logischen Blockgröße (TO-FILE-BLKSIZE) und dem Dateiformat (TO-FILE-BLKCTRL) einer ISAM-Zieldatei zu beachten.

Die Zuweisung TO-FILE-BLKCTRL=*STD bedeutet, dass die Blockkontrollinformation entsprechend dem Zielpubset gesetzt wird. Bei NK2-Pubsets wird das Dateiformat mit DATA2K, bei NK4-Pubsets mit DATA4K festgelegt.
Bei TO-FILE-BLKCTRL=*NK4 erhält die Blockkontrollinformation den Wert DATA4K.

Mit der Zuweisung TO-FILE-BLKSIZE wird die logische Blockgröße der Zieldatei festgelegt:

- TO-FILE-BLKSIZE=*STD:
  die logische Blockgröße wird nach den Gegebenheiten des Zielpubsets festgesetzt.
- TO-FILE-BLKSIZE=*NK4:
  die logische Blockgröße wird so gesteuert, dass sie auf jeden Fall geradzahlig ist.
- TO-FILE-BLKSIZE=<integer 1..16>:
  die Zieldatei wird mit der logische Blockgröße gleich dem angegebenen Wert erzeugt.

Folgende Übersicht zeigt nun die Verträglichkeit zwischen der logischen Blockgröße TO-FILE-BLKSIZE und dem Datenformat TO-FILE-BLKCTRL einer ISAM-Zieldatei.

	TO-FILE-BLKSIZE																	
TO-FILE-BLKCTRL	STD	NK4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
STD	X [1)]	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
NK4		X		X		X		X		X		X		X		X		X

[1)] X bedeutet in dieser Tabelle, dass beide Angaben unterstützt werden, andernfalls kommt es zu einer Fehlermeldung

Besonderheiten bei der Konvertierung von PAM-Dateien (Nicht-Phasen)

Um unterscheiden zu können, ob eine PAM-Datei mit PAM-Schlüssel diesen auch verwendet, wird der User-Teil des PAM-Schlüssels geprüft.

Ist dieser in jedem PAM-Block mit 8 Bytes X'00' belegt, wird die Datei als konvertierbar eingestuft.

Die Umsetzung in das NK-Format besteht im Verzicht auf die („nicht verwendete“) PAM-Schlüssel-Information, das Feld BLKCTRL-Indikator (siehe Abschnitt „Festlegung der Blockkontrollinformation bei der Konvertierung“) erhält den Wert NO.

Ist der User-Teil nicht mit X'00' belegt, so wird davon ausgegangen, dass diese Datei den PAM-Schlüssel verwendet. Eine solche Datei wird als nicht konvertierbar klassifiziert, die Konvertierung wird abgebrochen.

Eine Ausnahme bilden PAM-Dateien, deren PAM-Schlüssel-User-Teil sonst null ist, jedoch im ersten Byte des User-Teils mit X'01' oder X'80' belegt ist. Bei solchen Dateien wird angenommen, dass der PAM-Schlüssel nicht

verwendet wird, der User-Teil aber durch einen DVS-Fehler irrtümlich mit den oben genannten Werten versorgt wurde. Daher werden solche Dateien als konvertierbar klassifiziert, und eine Konvertierung in das NONKEY-Format ist möglich. Bei der Konvertierung wird durch eine Meldung auf diesen Sonderfall aufmerksam gemacht.

Enthält eine mit PAM-Schlüssel versehene PAM-Datei Dateilücken, so werden diese in der PAM-Datei ohne PAM-Schlüssel bei Konvertierung durch „Nullblöcke“ ersetzt (2048 Bytes X'00').

Konvertieren von Dateigenerationen

Die in einer Dateigenerationsgruppe befindlichen Dateigenerationen können konvertiert werden, indem für jede Dateigeneration die Anweisung CONVERT-FILE gegeben wird.

Die vollautomatische Konvertierung einer Dateigenerationsgruppe und der darin enthaltenen Dateigenerationen ist nicht möglich und wird durch eine Fehlermeldung abgewiesen.

Festlegung der Blockkontrollinformation bei der Konvertierung

Bei der Konvertierung vom K-Format in das NK-Format, wird für die Zieldatei die Blockkontrollinformation in Abhängigkeit von der jeweiligen Zugriffsmethode gesetzt und zwar auf BLKCTRL = DATA für ISAM- und SAM-Dateien und auf BLKCTRL = NO für PAM-Dateien.

Wird der Dateiname per Dateikettungsnamen angegeben und dabei der Parameter BLKCTRL spezifiziert, so muss dieser zutreffend angegeben werden.

Konvertierung einer NK-Datei in das NK-Format

Die Konvertierung einer Datei in das NK-Format, wenn die Quelldatei bereits NK-Format besitzt, wird aus Kompatibilitätsgründen weiterhin akzeptiert. Die Funktion kopiert aber lediglich die Quell- in die Zieldatei.

Übernahme der Dateischutzattribute nach der Konvertierung

Nach erfolgreicher Konvertierung können die Dateischutzattribute einer Quelldatei in die Zieldatei übernommen werden. Hierzu muss in der Anweisung CONVERT-FILE der Operand PROTECTION mit dem Wert *SAME angegeben werden. Gibt der Benutzer diesen Wert nicht an, wird die Zieldatei ohne die Merkmale der Quelldatei bezüglich Dateischutz und Dateisicherung erstellt.

Die Übernahme von Dateischutzattributen entspricht der Angabe PROTECTION=*SAME bei `/COPY-FILE` (siehe Handbuch „Kommandos“ [1]).
Um eine geschützte Datei konvertieren zu können, muss der Benutzer vor dem PAMCONV-Lauf sämtliche Zugriffsberechtigungen (z.B. Absetzen von Kennwörtern) mit den entsprechenden Kommandos spezifizieren. Der eigentliche Konvertierungsalgorithmus läuft anschließend unverändert ab. Am Ende einer erfolgreichen Konvertierung werden die Schutzattribute der Quelldatei in die Zieldatei übertragen.

- Übernahme der Dateischutzattribute in Abhängigkeit der Benutzerkennung

  Folgender Aufzählung ist die direkte Konvertierung (ohne Zwischenmedium) von Platte auf Platte zu Grunde gelegt.

  - Konvertieren innerhalb einer beliebigen Benutzerkennung

    Folgende Schutzattribute werden übernommen:

Schutzattribut	Beschreibung
ACCESS	Standardzugriffskontrolle; gibt an, ob für die Datei Schreibzugriff (impliziter Lesezugriff) oder nur Lesezugriff erlaubt ist.

BACKUP-CLASS	gibt die Häufigkeit der automatischen Dateisicherung mit dem Sicherungssystem ARCHIVE bzw. HSMS an.
NUM-OF-BACKUP-VERS	gibt an, ob die Datei am Versions-Backup teilnimmt bzw. wie viele Datei-Versionen maximal im HSMS Versions-Backup-Archiv gesichert werden sollen.
BASIC-ACL	Einfache Zugriffskontroll-Liste; die Zugriffskontrolle für die Datei erfolgt über einen BASIC-ACL-Eintrag. Die Zugriffsrechte Lesen, Schreiben und Ausführen können an verschiedene Benutzergruppen verteilt werden.
DESTROY-BY-DELETE	Nicht mehr benötigte Daten werden mit X'00' überschrieben. Dies erhöht den Datenschutz.
ENCRYPTION	Kennwortverschlüsselung
GUARDS	Zugriffskontrolle über GUARDS; GUARDS ist eine Funktionseinheit des Software-Produktes SECOS.
LARGE	Umfang der automatischen Dateisicherung mit dem Sicherungssystem ARCHIVE bzw. HSMS.
MIGRATE	Verdrängung von Dateien auf eine andere Speicherebene, wenn auf die Dateien längere Zeit nicht mehr zugegriffen wurde.
OPNBACK	gibt an, ob Datenbankdateien auch im geöffneten Zustand mit ARCHIVE gesichert werden können.
RETENTION-PERIOD	legt eine Schutzfrist fest, während der die Datei nur gelesen werden darf, d.h. sie darf weder geändert noch gelöscht werden.
USER-ACCESS	regelt den Zugriff anderer Benutzerkennungen auf die Datei.

Kennwörter können bei Konvertierung innerhalb einer beliebigen Benutzerkennung nicht übernommen werden. Dies ist nur unter der Kennung der Systemverwaltung (TSOS) möglich.

- Konvertieren unter der Kennung der Systemverwaltung (TSOS)

  Lese- (READ-PASSWORD), Schreib- (WRITE-PASSWORD) und Ausführungs-Kennwörter (EXEC-PASSWORD) sowie alle oben beschriebenen Schutzattribute werden übernommen.

- Konvertieren einer Quelldatei von einer fremden Benutzerkennung in die eigene Kennung

  Die Dateisicherungsattribute LARGE, BACKUP, NUM-OF-BACKUP-VERS, MIGRATE, OPNBACK sowie die Dateisicherheitsattribute DESTROY-BY-DELETE, RETENTION-PERIOD, READ-PASSWORD, WRITE-PASSWORD, EXEC-PASSWORD und ENCRYPTION werden übernommen.
  BASIC-ACL oder GUARDS-Einträge in der Quelldatei werden nicht in die Zieldatei übertragen. Diese Einträge werden in der Zieldatei ebenso wie das Dateischutzattribut ACCESS mit Standardwerten versorgt. Verfügt eine bereits existierende Zieldatei schon über Schutzeinträge, werden diese vor der Übernahme der Quelldatei-Einträge zurückgesetzt.

- Konvertieren einer Quelldatei von eigener Benutzerkennung auf eine fremde Benutzerkennung

  Die Schutzattribute der Quelldatei werden nicht in die Zieldatei übernommen, auch wenn der Anwender berechtigt ist, die Zieldatei zu erzeugen.

- Einschränkungen bei der Übernahme von Dateischutzattribute bzgl. Konvertierung über ein Zwischenmedium

Die Übernahme der Dateischutzattribute wird nur bei der „Konvertierung über Zwischendatei in einem Schritt“ unterstützt.
Bei der „Konvertierung über Zwischendatei in zwei Schritten“ werden die Dateischutzattribute nicht übernommen. Die Angabe PROTECTION=*SAME in der Anweisung CONVERT-FILE wird ignoriert.

- Konvertierung über Zwischenmedium in einem Schritt

  Die Schutzattribute der Quelldatei werden übernommen, da weder der PAMCONV-Lauf, noch die aktuelle laufende Konvertierungsanweisung CONVERT-FILE unterbrochen wird. Folgende Verarbeitungsschritte werden ausgeführt:

  - Konvertierung von gemeinschaftlicher Platte auf Zwischenmedium
  - Freigabe des Speicherplatzes der Quelldatei
  - interner Transfer von Zwischenmedium auf gemeinschaftliche Platte
  - Setzen der Schutzattribute
  - Löschen der Quelldatei

- Konvertierung über Zwischenmedium in zwei Schritten

  Bei dieser Konvertierungsart kann sowohl der PAMCONV-Lauf als auch die Konvertierungsanweisung unterbrochen werden. Bei Ausgabe der Zwischendatei auf Magnetband oder Privatplatte können die Dateischutzattribute aus folgenden Gründen nicht übernommen werden:

  - Der zweite Konvertierungsschritt (von Magnetband oder Privatplatte auf die Zieldatei) könnte zu einem beliebigen, späteren Zeitpunkt an einem beliebigen System erfolgen. Hierbei kann es zu Inkompatibilitäten kommen, da die Übernahme der NK-Datei vom Zwischenmedium auf gemeinschaftlichen Datenträger nicht mit PAMCONV durchgeführt werden muss. Andere Übernahmeprogramme erhalten keine Informationen zur Übernahme der Dateischutzattribute.
  - Der Benutzerkennsatz des Magnetbandes, das eine Zwischendatei aufnehmen soll, muss aus Kompatibilitätsgründen beibehalten werden; da weitere Dateieigenschaften aus Platzgründen nicht mehr aufgenommen werden können, ist die Übernahme der Schutzattribute nicht möglich.

- Verfahrensweise mit der Zieldatei nach der Konvertierung

  Das Verfahren mit der Zieldatei nach der Konvertierung wird mit dem Operanden FILE-DISPOSAL in der Anweisung CONVERT-FILE geregelt. Die Übernahme der Dateischutzattribute wird bei Angabe der folgenden Operandenwerte wie folgt durchgeführt.

FILE-DISPOSAL =	Bedeutung für die Übernahme der Dateischutzattribute
KEEP	Voreinstellung. Quell- und Zieldatei bleiben erhalten. Die Dateischutzattribute werden ohne Problem übernommen.
RENAME	Die Quelldatei wird nach der Konvertierung gelöscht. Die Schutzattribute werden vor dem Löschen übernommen. Falls bei der Übernahme ein Fehler auftritt, wird die Quelldatei nicht gelöscht und mit einem Fehler abgebrochen. Die Dateischutzattribute bleiben dadurch erhalten.
REPLACE	Die Quelldatei wird nach der Konvertierung gelöscht und die Zieldatei wird gemäß der Quelldatei umkatalogisiert. Die Schutzattribute werden vor dem Löschen übernommen. Falls bei der Übernahme ein Fehler auftritt, wird die Quelldatei nicht gelöscht und mit einem Fehler abgebrochen. Die Dateischutzattribute bleiben dadurch erhalten.
INPLACE	Nach der Konvertierung wird die Quelldatei mit der Zieldatei überschrieben. Die Dateischutzattribute werden übernommen.

9.5.2 Weitere Hinweise zur Konvertierung

Als Entscheidungshilfe für die Konvertierung bietet PAMCONV die Anweisung CLASSIFY-FILE an, die Dateien auf ihre Konvertierbarkeit hin überprüft.

9.6 Anweisungen

PAMCONV liest jeweils eine Anweisung ein und führt diese Anweisung sofort aus. Wurden für die Konvertierung Linknamen spezifiziert, werden die zugehörigen TFT-Einträge nach Beendigung der Konvertierung nicht gelöscht, um eine weitere Verwendung dieser Linknamen zu ermöglichen.

9.6.1 Übersicht über die Anweisungen von PAMCONV

Anweisung	Bedeutung
CHANGE-TO-SYSTEM-MODE	Wechsel in den Systemmodus
CLASSIFY-FILE	Dateien nach Konvertierbarkeit klassifizieren
CONVERT-FILE	Konvertieren von ISAM-, SAM- oder PAM-Dateien sowie Phasendateien
END	Beenden des Programms PAMCONV
MODIFY-CONVERT-FILE-DEFAULTS	Ändern der eingestellten Standardwerte für die Anweisung CONVERT-FILE
MODIFY-LOGGING-OPTIONS	Ändern der Optionen für die Protokollierung
SHOW-CONVERT-FILE-DEFAULTS	Auflisten der eingestellten Standardwerte für die Anweisung CONVERT-FILE
SHOW-LOGGING-OPTIONS	Auflisten der eingestellten Protokoll-Optionen

Weitere zulässige Anweisungen

Außerdem sind die von SDF standardmäßig angebotenen Anweisungen (z.B. STEP) zulässig. Sie haben allgemeine Bedeutung im Zusammenhang mit SDF. Sie werden im Handbuch „Dialogschnittstelle SDF“ [20] beschrieben.

9.6.2 Beschreibung der Anweisungen

- CHANGE-TO-SYSTEM-MODE - Wechsel in den System-Modus
- CLASSIFY-FILE - Dateien nach Konvertierbarkeit klassifizieren
- CONVERT-FILE - Konvertieren von Dateien
- END - PAMCONV beenden
- MODIFY-CONVERT-FILE-DEFAULTS - Standardwerte für CONVERT-FILE einstellen
- MODIFY-LOGGING-OPTIONS - Protokollierungswerte einstellen
- SHOW-CONVERT-FILE-DEFAULTS - Standardwerte für CONVERT-FILE auflisten
- SHOW-LOGGING-OPTIONS - Eingestellte Protokoll-Optionen auflisten

9.6.2.1 CHANGE-TO-SYSTEM-MODE - Wechsel in den System-Modus

Die Anweisung CHANGE-TO-SYSTEM-MODE bewirkt den Wechsel in den Systemmodus von BS2000. Anschließend können BS2000-Kommandos eingegeben werden. Wenn PAMCONV nicht entladen wurde (z.B. durch `/START-EXECUTABLE-PROGRAM` oder `/LOAD-EXECUTABLE-PROGRAM`), kann danach der PAMCONV-Lauf mit `/RESUME-PROGRAM` fortgesetzt werden.

Format

CHANGE-TO-SYSTEM-MODE

Diese Anweisung besitzt keine Operanden.

Um in den Systemmodus zu wechseln, können auch die Anweisungen SYSTEM oder SYS verwendet werden, diese können allerdings nicht in gekürzter Form eingegeben werden.

9.6.2.2 CLASSIFY-FILE - Dateien nach Konvertierbarkeit klassifizieren

Es werden zu jeder ausgewählten Datei Aussagen angefordert darüber, ob sie konvertierbar ist und ob sich Inkompatibilitäten ergeben.

> ¡ Die durch die Anweisung MODIFY-CONVERT-FILE-DEFAULTS eingestellten Standardwerte werden bei dieser Anweisung nicht berücksichtigt. Das Ergebnis der Anweisung CLASSIFY-FILE beschreibt die Konvertierbarkeit einer Quelldatei. Für Aussagen zur Zieldatei muss der Abschnitt „Umblockung“ berücksichtigt werden.

Format

```
CLASSIFY-FILE

 DIRECTION = *TO-NONKEY / *NONKEY-TO-KEY

 ,FROM-FILE = *LINK(...) / *ALL / <partial-filename 2..53 with-wild(79)> / <filename 1..54>

     *LINK(...)

         |    LINK-NAME = <filename 1..8 without-gen>

 ,SELECT = *ALL / *BY-ATTRIBUTES(...)

     BY-ATTRIBUTES(...)

         |    CREATION-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)

         |          *INTERVAL(...)

         |                |      FROM = 0000-01-01 / <date> / *YESTERDAY

         |                |      ,TO = *TODAY / <date> / *YESTERDAY

         |    ,LAST-ACCESS-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)

         |          *INTERVAL(...)

         |                |      FROM = 0000-01-01 / <date> / *YESTERDAY

         |                |      ,TO = *TODAY / <date> / *YESTERDAY

         |    ,SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)

         |          *INTERVAL(...)

         |                |      FROM = 0 / <integer 0..16777215>

         |                |      ,TO = 16777215 / <integer 0..16777215>

         |    ,FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM

         |    ,BLKSIZE = *ANY / <integer 1..16>

         |    ,BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K
```

Operanden

DIRECTION = *TO-NONKEY / *NONKEY-TO-KEY
Bezeichnet die Richtung der Dateikonvertierung, die vom Anwender vorgesehen ist. Da die Einteilung je nach Konvertierungsrichtung unterschiedlich ist, ist diese hier zu spezifizieren.

DIRECTION = *TO-NONKEY
Die Konvertierung von Dateien soll ins NK-Format erfolgen.

DIRECTION = *NONKEY-TO-KEY
Die Konvertierung von Dateien soll vom NK-Format ins K-Format erfolgen.

FROM-FILE = <filename 1..54> / *LINK(...) / <partial-name 2..79 with-wild> / *ALL
Bezeichnet die Dateien, die geprüft werden sollen.

FROM-FILE = <filename 1..54>
Bezeichnet den vollqualifizierten Dateinamen. Die Angabe einer Dateigeneration ist zulässig.

FROM-FILE = *LINK(...)
Bezeichnet die Dateien über einen Dateikettungsnamen.

> **LINK-NAME = <filename 1..8 without-gen>**
> Bezeichnet den Dateikettungsnamen.

FROM-FILE = <partial-filename 2..53 with-wild(79)>
Bezeichnet den teilqualifizierten Dateinamen mit Wildcardsyntax.

FROM-FILE = *ALL
Alle Dateien der Benutzerkennung sollen geprüft werden.

SELECT = *ALL / *BY-ATTRIBUTES(...)
Gibt an, ob die Dateien, die klassifiziert werden sollen, über bestimmte Auswahlkriterien zusätzlich zum teilqualifizierten Dateinamen ausgewählt werden sollen.

SELECT = *ALL
Für die Quelldateien werden keine zusätzlichen Auswahlkriterien getroffen.

SELECT = *BY-ATTRIBUTES(...)
Bestimmt die Auswahlkriterien für die zu klassifizierenden Dateien.

> **CREATION-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY**
> Bezeichnet das Erstellungsdatum als Auswahlkriterium.
>
> **CREATION-DATE = *ANY**
> Das Erstellungsdatum wird nicht als Auswahlkriterium herangezogen. Alle Dateien werden bei der Auswahl berücksichtigt.
>
> **CREATION-DATE = *INTERVAL(...)**
> Dateien mit Erstellungsdatum innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.
>
> > **FROM = 0000-01-01 / *YESTERDAY / <date>**
> > Dateien mit Erstellungsdatum gleich oder später dem angegebenen Grenzwert werden selektiert.
> >
> > **FROM = 0000-01-01**
> > Der untere Grenzwert ist das früheste mögliche Datum.

FROM = *YESTERDAY
Der untere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum >= dem gestrigen Tagesdatum werden selektiert.

FROM = <date>
Der untere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum >= dem angegebenen Wert werden selektiert.

TO = *TODAY / *YESTERDAY / <date>
Dateien mit Erstellungsdatum gleich oder früher dem angegebenen Grenzwert werden selektiert.

TO = *TODAY
Der obere Grenzwert ist das aktuelle Tagesdatum. Dateien mit Erstellungsdatum <= dem aktuellen Tagesdatum werden selektiert.

TO = *YESTERDAY
Der obere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum <= dem gestrigen Tagesdatum werden selektiert.

TO = <date>
Der obere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum <= dem angegebenen Wert werden selektiert.

LAST-ACCESS-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY
Bezeichnet das Datum des letzten Dateizugriffs als Auswahlkriterium.

Erklärung der Operanden ANY, INTERVAL(...), <date>, TODAY und YESTERDAY siehe Operand CREATION-DATE.

SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)
Bezeichnet die Dateigröße als Auswahlkriterium.

SIZE = *ANY
Die Dateigröße wird nicht als Auswahlkriterium herangezogen.

SIZE = <integer 0..16777215>
Dateien mit einer Dateigröße gleich dem angegebenen Wert werden selektiert.

SIZE = *INTERVAL(...)
Dateien mit einer Dateigröße innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.

FROM = 0 / <integer 0..16777215>
Dateien mit einer Dateigröße >= dem angegebenen Grenzwert werden selektiert.

FROM = 0
Der untere Grenzwert ist die kleinste mögliche Größe.

FROM = <integer 0..16777215>
Der untere Grenzwert ist die angegebene Größe.

TO = 16777215 / <integer 0..16777215>
Dateien mit einer Dateigröße <= dem angegebenen Grenzwert werden selektiert.

TO = 16777215
Der obere Grenzwert ist die größte mögliche Größe.

TO = <integer 0..16777215>
Der obere Grenzwert ist die angegebene Größe.

FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM
Bezeichnet die Zugriffsmethode als Auswahlkriterium.

FILE-STRUCTURE = *ANY
Die Zugriffsmethode wird nicht als Auswahlkriterium herangezogen.

FILE-STRUCTURE = *SAM
Dateien mit Zugriffsmethode SAM werden selektiert.

FILE-STRUCTURE = *ISAM
Dateien mit Zugriffsmethode ISAM werden selektiert.

FILE-STRUCTURE = *PAM
Dateien mit Zugriffsmethode PAM werden selektiert.

BLKSIZE = *ANY / <integer 1..16>
Bezeichnet die Blockgröße als Auswahlkriterium.

BLKSIZE = *ANY
Die Blockgröße wird nicht als Auswahlkriterium herangezogen.

BLKSIZE = <integer 1..16>
Dateien mit der Blockgröße gleich dem angegebenen Wert werden selektiert.

BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K
Bezeichnet die Blockkontrolleigenschaft als Auswahlkriterium.

BLKCTRL = *ANY
Die Blockkontrolleigenschaft wird nicht als Auswahlkriterium herangezogen.

BLKCTRL = *PAMKEY
Dateien mit der Blockkontrolleigenschaft PAMKEY werden selektiert.

BLKCTRL = *NO
Dateien mit der Blockkontrolleigenschaft NO werden selektiert.

BLKCTRL = *DATA
Dateien mit der Blockkontrolleigenschaft DATA werden selektiert.

BLKCTRL = *DATA2K
Dateien mit der Blockkontrolleigenschaft DATA2K werden selektiert.

BLKCTRL = *DATA4K
Dateien mit der Blockkontrolleigenschaft DATA4K werden selektiert.

Das Ausgabeziel wird durch den Operanden OUTPUT der Anweisung MODIFY-LOGGING-OPTIONS bestimmt.

Die Prüfergebnisse werden in folgender Form nach SYSLST ausgegeben (Zeilenlänge max. 132 Zeichen):

```
%//CLASSIFY-FILE   FROM-FILE=>from-file<
% CLASSIFY-FILE    >from-file<    DIRECTION = >direction<
%
% FILENAME                                        !  PAM-  !FCB- !CONVER-!
INCOMPATIBILITIES  !
%                                                 !  PAGES !TYPE !TIBLE
```

```
!                     !
%
-------------------------------------------------------------------------------------------
```

Die Prüfergebnisse werden in folgender Form nach SYSOUT ausgegeben (Zeilenlänge max. 80 Zeichen):

```
% CLASSIFY-FILE   FROM-FILE=>from-file<
% CLASSIFY-FILE   >from-file<    DIRECTION = >direction<
%
% FILENAME                                                    ! CONVER- !
%                                                             ! TIBLE   !
% ------------------------------------------------------------------------
% :CATID:$USERID.>filename 1< ..............................  ! >yesno< !
% :CATID:$USERID.>filename 2< ..............................  ! >yesno< !
% :CATID:$USERID.>filename 3< ..............................  ! >yesno< !
%
%
% :CATID:$USERID.>filename n< ..............................  ! >yesno< !
% ------------------------------------------------------------------------
%            >n<       FILE(S) LISTED
```

Bedeutung der Ausgabefelder:

>from-file<	In der Anweisung CLASSIFY-FILE angegebene Dateinamen.
>direction<	In der Anweisung CLASSIFY-FILE angegebene vorgesehene Konvertierungsrichtung
>filename<	Dateiname der geprüften Datei
>size<	Größe der geprüften Datei
>fcb<	FCB-Typ der geprüften Datei
>n<	Gesamtzahl der geprüften Dateien
>yesno<	Aussage, ob die Datei konvertierbar ist. Mögliche Werte: YES — Datei ist konvertierbar. NO — Datei ist nicht konvertierbar. NK2 — Datei ist nur auf NK2-Pubsets mit Standard-Blksize konvertierbar (PLAM-Bibliotheken). NK4 ... — Datei ist nur auf NK4-Pubsets konvertierbar (z.B. SAM-Dateien mit BLKSIZE=RECSIZE und Standard-Blksize).
>reason<	Grund der Inkompatibilität. Mögliche Werte: NONE keine Inkompatibilitäten, Datei ist konvertierbar.

RECSIZE EXCEEDS MAXIMUM

Die Satzlänge überschreitet den durch BLKSIZE bestimmten Maximalwert. Die Datei kann durch Heraufsetzen des Blockungsfaktors konvertiert werden.

RECSIZE EXCEEDS MAX(NK2)

Die Satzlänge würde auf NK2-Pubsets den durch BLKSIZE bestimmten Maximalwert überschreiten. Eine Konvertierung auf NK4-Pubsets ist möglich.

PLAM(NK4) NO KONVERT

PLAM-Bibliotheken mit BLKSIZE > 2 werden von PAMCONV nicht konvertiert.

FILE ALREADY IN KEY-FORMAT

Die Datei liegt bereits im K-Format vor. Die Datei im K-Format kann nicht ins K-Format konvertiert werden.

FILESIZE INCREASES

Durch die Bildung von OVERFLOW-Blöcken wird die Zieldatei größer, Datei ist konvertierbar.

KEYPOS IN OVERFLOW-BLOCK

Der ISAM-Schlüssel würde in einem OVERFLOW-Block abgelegt werden müssen. Die Datei kann durch Heraufsetzen des Blockungsfaktors konvertiert werden.

LMR-LIBRARY

Datei ist eine LMR-Bibliothek, sie ist nicht konvertierbar.

PAMKEY IS USED

Datei verwendet den Userteil des PAM-Schlüssels, Datei ist nicht konvertierbar.

PAMKEY CONTAINS SPECIAL FLAG

Datei enthält im ersten Byte des Userteils vom PAM-Schlüssel nur das Zeichen X'01' oder X'80'. Die Datei ist konvertierbar.

9.6.2.3 CONVERT-FILE - Konvertieren von Dateien

Die Anweisung CONVERT-FILE dient zur Konvertierung von Dateien aus einem Format, bei dem der PAM-Schlüssel für die Datendarstellung verwendet wird, in ein Format, bei dem der PAM-Schlüssel nicht verwendet wird, oder zur Konvertierung in entgegengesetzter Richtung.

Des Weiteren dient die Anweisung CONVERT-FILE zur Umblockung (siehe Abschnitt „Umblockung“).

Die Anweisung CONVERT-FILE bietet drei Möglichkeiten:

- **Konvertieren**: Änderung des Dateiformats, d.h. Umwandlung von K-Format in NK-Format oder umgekehrt.
- **Umblocken:** Änderung der logischen Blockgröße. Keine Änderung im Dateiformat.
- **Konvertieren und Umblocken**

Die Standardwerte für die Anweisung CONVERT-FILE werden mit der Anweisung MODIFY-CONVERT-FILE-DEFAULTS eingestellt.

Die durch Unterstreichung gekennzeichneten Standardwerte gelten nur, wenn keine anderen Werte eingestellt wurden.

Format

CONVERT-FILE

```
DIRECTION = *TO-NONKEY / *NONKEY-TO-KEY
,FROM-FILE = <filename 1..54> / *LINK(...) / <partial-filename 2..53 with-wild(79)> / *ALL
    *LINK(...)
        | LINK-NAME = <filename 1..8 without-gen>
,SELECT = *ALL / *BY-ATTRIBUTES(...)
    BY-ATTRIBUTES(...)
        | CREATION-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)
        |     *INTERVAL(...)
        |         | FROM = 0000-01-01 / <date> / *YESTERDAY
        |         | ,TO = *TODAY / <date> / *YESTERDAY
        | ,LAST-ACCESS-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)
        |     *INTERVAL(...)
        |         | FROM = 0000-01-01 / <date> / *YESTERDAY
        |         | ,TO = *TODAY / <date> / *YESTERDAY
        | ,SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)
        |     *INTERVAL(...)
        |         | FROM = 0 / <integer 0..16777215>
        |         | ,TO = 16777215 / <integer 0..16777215>
        | ,FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM
        | ,BLKSIZE = *ANY / <integer 1..16>
        | ,BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K
```

```
,TO-FILE = <filename 1..54> / *LINK(...) / <partial-filename 2..53 with-wild(79)>
      *LINK(...)
           |     LINK-NAME = <filename 1..8 without-gen>
,TO-FILE-BLKSIZE = *STD / *NK4 / <integer 1..16>
,TO-FILE-BLKCTRL = *STD / *NK4
,REPLACE-OLD-FILES = *NO / *YES / *DIALOG
,FILE-DISPOSAL = *KEEP / *REPLACE / *INPLACE / *RENAME
,PROTECTION = *STD / *SAME
,DEVICE-FOR-TEMPFILE = *NONE / *TAPE(...) / *DISK(...)
      *TAPE(...)
           |     VOLUME = list-poss(100): <alphanum-name 1..6>
           |     ,DEVICE-TYPE = <device>
      *DISK(...)
           |     VOLUME = list-poss(100): <alphanum-name 1..6>
           |     ,DEVICE-TYPE = <device>
```

Operanden

DIRECTION = *TO-NONKEY / *NONKEY-TO-KEY
Bezeichnet die Richtung der Dateikonvertierung.

DIRECTION = *TO-NONKEY
Die Dateikonvertierung soll ins NK-Format erfolgen. Die Quelldatei kann im K-Format oder im NK-Format vorliegen.

DIRECTION = *NONKEY-TO-KEY
Die Dateikonvertierung soll vom NK-Format ins K-Format erfolgen.

FROM-FILE = <filename 1..54> / *LINK(...) / <partial-filename 2..53 with-wild(79)> / *ALL
Bezeichnet die Dateien, die konvertiert werden sollen.

FROM-FILE = <filename 1..54>
Bezeichnet die Datei, die konvertiert werden soll. Die Angabe einer Dateigeneration ist zulässig.

FROM-FILE = *LINK(...)
Die zu konvertierende Datei wurde durch ein vorher eingegebenes `/ADD-FILE-LINK` spezifiziert, der dort angegebene Dateikettungsname muss mit dem hier angegebenen übereinstimmen.

LINK-NAME = <filename 1..8 without-gen>
Bezeichnet den Dateikettungsnamen.

FROM-FILE = <partial-filename 2..53 with-wild(79)>
Bedeutet, dass alle Dateien, die der angegebenen Wildcardsyntax und den spezifizierten Auswahlkriterien entsprechen, konvertiert werden sollen.

FROM-FILE = *ALL
Bedeutet, dass alle Dateien, die den spezifizierten Auswahlkriterien entsprechen, konvertiert werden sollen.

SELECT = *ALL / *BY-ATTRIBUTES(...)
Gibt an, ob die Dateien, die konvertiert werden sollen, zusätzlich zum teilqualifizierten Dateinamen über bestimmte Auswahlkriterien ausgewählt werden sollen.

SELECT = *ALL
Für die Quelldateien werden keine zusätzlichen Auswahlkriterien festgelegt.

SELECT = *BY-ATTRIBUTES(...)
Bestimmt die Auswahlkriterien für die zu konvertierenden Dateien.

CREATION-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY
Bezeichnet das Erstellungsdatum als Auswahlkriterium.

CREATION-DATE = *ANY
Das Erstellungsdatum wird nicht als Auswahlkriterium herangezogen. Alle Dateien werden bei der Auswahl berücksichtigt.

CREATION-DATE = INTERVAL(...)
Dateien mit Erstellungsdatum innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.

FROM = 0000-01-01 / *YESTERDAY / <date>
Dateien mit Erstellungsdatum gleich oder später dem angegebenen Grenzwert werden selektiert.

FROM = 0000-01-01
Der untere Grenzwert ist das früheste mögliche Datum.

FROM = *YESTERDAY
Der untere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum >= dem gestrigen Tagesdatum werden selektiert.

FROM = <date>
Der untere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum >= dem angegebenen Wert werden selektiert.

TO = TODAY / *YESTERDAY / <date>
Dateien mit Erstellungsdatum gleich oder früher dem angegebenen Grenzwert werden selektiert.

TO = *TODAY
Der obere Grenzwert ist das aktuelle Tagesdatum. Dateien mit Erstellungsdatum <= dem aktuellen Tagesdatum werden selektiert.

TO = *YESTERDAY
Der obere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum <= dem gestrigen Tagesdatum werden selektiert.

TO = <date>
Der obere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum <= dem angegebenen Wert werden selektiert.

LAST-ACCESS-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY
Bezeichnet das Datum des letzten Dateizugriffs als Auswahlkriterium.

Erklärung der Operanden ANY, INTERVAL(...), <date>, TODAY und YESTERDAY siehe Operand CREATION-DATE.

SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)
Bezeichnet die Dateigröße als Auswahlkriterium.

SIZE = *ANY
Die Dateigröße wird nicht als Auswahlkriterium herangezogen.

SIZE = <integer 0..16777215>
Dateien mit einer Dateigröße gleich dem angegebenen Wert werden selektiert.

SIZE = *INTERVAL(...)
Dateien mit einer Dateigröße innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.

FROM = 0 / <integer 0..16777215>
Dateien mit einer Dateigröße >= dem angegebenen Grenzwert werden selektiert.

FROM = 0
Der untere Grenzwert ist die kleinste mögliche Größe.

FROM = <integer 0..16777215>
Der untere Grenzwert ist die angegebene Größe.

TO = 16777215 / <integer 0..16777215>
Dateien mit einer Dateigröße <= dem angegebenen Grenzwert werden selektiert.

TO = 16777215
Der obere Grenzwert ist die größte mögliche Größe.

TO = <integer 0..16777215>
Der obere Grenzwert ist die angegebene Größe.

FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM
Bezeichnet die Zugriffsmethode als Auswahlkriterium.

FILE-STRUCTURE = *ANY
Die Zugriffsmethode wird nicht als Auswahlkriterium herangezogen.

FILE-STRUCTURE = *SAM
Dateien mit Zugriffsmethode SAM werden selektiert.

FILE-STRUCTURE = *ISAM
Dateien mit Zugriffsmethode ISAM werden selektiert.

FILE-STRUCTURE = *PAM
Dateien mit Zugriffsmethode PAM werden selektiert.

BLKSIZE = *ANY / <integer 1..16>
Bezeichnet die Blockgröße als Auswahlkriterium.

BLKSIZE = *ANY
Die Blockgröße wird nicht als Auswahlkriterium herangezogen.

BLKSIZE = <integer 1..16>
Dateien mit der Blockgröße gleich dem angegebenen Wert werden selektiert.

BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K
Bezeichnet die Blockkontrolleigenschaft als Auswahlkriterium.

BLKCTRL = *ANY
Die Blockkontrolleigenschaft wird nicht als Auswahlkriterium herangezogen.

BLKCTRL = *PAMKEY
Dateien mit der Blockkontrolleigenschaft PAMKEY werden selektiert.

BLKCTRL = *NO
Dateien mit der Blockkontrolleigenschaft NO werden selektiert.

BLKCTRL = *DATA
Dateien mit der Blockkontrolleigenschaft DATA werden selektiert.

BLKCTRL = *DATA2K
Dateien mit der Blockkontrolleigenschaft DATA2K werden selektiert.

BLKCTRL = *DATA4K
Dateien mit der Blockkontrolleigenschaft DATA4K werden selektiert.

TO-FILE = <filename 1..54> / *LINK(...) / <partial-filename 2..53 with-wild(79)>
Bezeichnet die Dateien, die durch die Konvertierung erzeugt werden sollen.

TO-FILE = <filename 1..54>
Bezeichnet die Datei, die durch die Konvertierung erzeugt werden soll. Die Angabe einer Dateigeneration ist zulässig.

TO-FILE = *LINK(...)
Die zu erzeugende Datei wurde durch ein vorher eingegebenes `/ADD-FILE-LINK` spezifiziert, der dort angegebene Dateikettungsname muss mit dem hier angegebenen übereinstimmen.

LINK-NAME = <filename 1..8 without-gen>
Bezeichnet den Dateikettungsnamen.

TO-FILE = <partial-filename 2..53 with-wild(79)>
Bezeichnet die Dateien, die durch die Konvertierung erzeugt werden sollen, teilqualifiziert mit Wildcardsyntax.

TO-FILE-BLKSIZE = *STD / *NK4 / <integer 1..16>
Bezeichnet die logische Blockgröße der Zieldatei.

TO-FILE-BLKSIZE = *STD
Die logische Blockgröße der Zieldatei wird nicht vom Benutzer definiert. PAMCONV richtet sich nach den Gegebenheiten des Ziel-Pubsets und setzt, wenn notwendig, den Blockungsfaktor herauf. Es wird intern maximal um eins heraufgeblockt.

TO-FILE-BLKSIZE = *NK4
Die logische Blockgröße der Zieldatei wird so gesteuert, dass sie auf jeden Fall geradzahlig ist, d.h. die Zieldatei kann auf einem NK4-Pubset abgelegt werden. Es wird intern maximal um eins heraufgeblockt.

TO-FILE-BLKSIZE = <integer 1..16>
Die Zieldatei wird mit der Blockgröße gleich dem angegebenen Wert erzeugt, wenn die Angabe mit den übrigen Bedingungen der Umblockung verträglich ist (siehe Abschnitt „Umblockung“).

TO-FILE-BLKCTRL = *STD / *NK4
Bezeichnet die Blockkontroll-Information der Zieldatei. Dieser Operand ist nur für die Konvertierungsrichtung TO-NONKEY und ausschließlich für ISAM-Dateien relevant.

TO-FILE-BLKCTRL = *STD
Die Blockkontroll-Information wird entsprechend dem Zielpubset gesetzt. Bei NK2-Pubsets wird das Datenformat mit DATA2K, bei NK4-Pubsets mit DATA4K festgelegt.

TO-FILE-BLKCTRL = *NK4
Die Blockkontroll-Information erhält den Wert DATA4K.

REPLACE-OLD-FILES = *NO / *YES / *DIALOG
Gibt an, ob unter diesem Namen bereits existierende Dateien überschrieben werden sollen.

REPLACE-OLD-FILES = *NO
Dateien sollen nicht überschrieben werden, die Konvertierung dieser Datei wird nicht durchgeführt.

REPLACE-OLD-FILES = *YES
Dateien sollen in jedem Fall überschrieben werden, sofern nicht ein zusätzlicher Schutz (Passwort, ACCESS=*READ,..) vorhanden ist.

REPLACE-OLD-FILES = *DIALOG
Der Benutzer wird im Dialog aufgefordert, die Verfahrensweise für bereits existierende Dateien zu bestimmen. Nur im Dialogbetrieb möglich.

FILE-DISPOSAL = *KEEP / *RENAME / *REPLACE / *INPLACE
Gibt an, wie mit der Quell- und Zieldatei nach der Konvertierung verfahren werden soll.

FILE-DISPOSAL = *KEEP
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Die Zieldateien existieren neben den Quelldateien.

FILE-DISPOSAL = *RENAME
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung sollen die Quelldateien gelöscht werden.

FILE-DISPOSAL = *REPLACE
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung sollen die Quelldateien gelöscht werden und die Zieldateien die Namen der Quelldateien erhalten. Effektiv wird die Quelldatei durch die Zieldatei ersetzt.

FILE-DISPOSAL = *INPLACE
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung soll versucht werden, die Quelldatei mit der Zieldatei zu überschreiben und der Zieldatei den Namen der Quelldatei zu geben. Dies bewirkt, dass die Zieldatei auch physikalisch in etwa den gleichen Platz wie die Quelldatei einnimmt. Effektiv wird die Quelldatei durch die Zieldatei ersetzt.

PROTECTION = *STD / *SAME
Gibt an, ob die Dateischutzattribute der Quelldatei in die Zieldatei übernommen werden sollen.

PROTECTION = *STD
Die Dateischutzattribute werden nicht übernommen.

PROTECTION = *SAME
Die Dateischutzattribute werden in die Zieldatei übernommen. Näheres siehe Abschnitt „Übernahme der Dateischutzattribute nach der Konvertierung“ (Besonderheiten bei der Konvertierung).

DEVICE-FOR-TEMPFILE = *NONE / *TAPE(...) / *DISK(...)
Bezeichnet das Speichermedium, auf dem die erzeugte Zwischendatei abgelegt werden soll.

DEVICE-FOR-TEMPFILE = *NONE
Es soll keine Zwischendatei auf einem privaten Datenträger abgelegt werden.

DEVICE-FOR-TEMPFILE = *TAPE(...)
Die erzeugte Zwischendatei soll auf Magnetband ablegt werden.

VOLUME = list-poss(100): <alphanum-name 1..6>
Bezeichnet die VSN des Bandes bzw. der Bänder, falls mehrere VSNs in Form einer Liste angegeben werden, das bzw. die als Speichermedium dienen soll(en).

DEVICE-TYPE = <device>
Bezeichnet den Bandgerätetyp, der verwendet werden soll.

DEVICE-FOR-TEMPFILE = *DISK(...)
Die erzeugte Zwischendatei soll auf einer Privatplatte abgelegt werden.

VOLUME = list-poss(100): <alphanum-name 1..6>
Bezeichnet die VSN der Privatplatte bzw. der Platten, falls mehrere VSNs in Form einer Liste angegeben werden, die als Speichermedium dienen soll(en).

DEVICE-TYPE = <device>
Bezeichnet den Plattengerätetyp, der verwendet werden soll.

9.6.2.4 END - PAMCONV beenden

Die Anweisung END bewirkt die Beendigung des Programms PAMCONV.

Format

END

Diese Anweisung besitzt keine Operanden.

Standardwerte CONVERT-FILE einstellen

Mit dieser Anweisung werden die Standardwerte für die Anweisung CONVERT-FILE eingestellt. Diese gelten in einem Programmlauf bis zur nächsten MODIFY-CONVERT-FILE-DEFAULTS-Anweisung.

Bei Eingabe der Anweisung ohne Operanden behalten die eingestellten Standardwerte ihre Gültigkeit.

Die aktuell gültigen Werte können durch die Anweisung SHOW-CONVERT-FILE-DEFAULTS abgefragt werden.

> ¡ Die hier eingestellten Standardwerte gelten **nur** für die Anweisung CONVERT-FILE. Die Anweisung CLASSIFY-FILE ist davon nicht beeinflusst.

Format

```
MODIFY-CONVERT-FILE-DEFAULTS

 DIRECTION = *UNCHANGED / *TO-NONKEY / *NONKEY-TO-KEY

 ,SELECT = *UNCHANGED / *ALL / *BY-ATTRIBUTES(...)

     *BY-ATTRIBUTES(...)

         |    CREATION-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)

         |           *INTERVAL(...)

         |                |      FROM = 0000-01-01 / <date> / *YESTERDAY

         |                |      ,TO = *TODAY / <date> / *YESTERDAY

         |    ,LAST-ACCESS-DATE = *ANY / <date> / *TODAY / *YESTERDAY / *INTERVAL(...)

         |           *INTERVAL(...)

         |                |      FROM = 0000-01-01 / <date> / *YESTERDAY

         |                |      ,TO = *TODAY / <date> / *YESTERDAY

         |    ,SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)

         |           *INTERVAL(...)

         |                |      FROM = 0 / <integer 0..16777215>

         |                |      ,TO = 16777215 / <integer 0..16777215>

         |    ,FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM

         |    ,BLKSIZE = *ANY / <integer 1..16>

         |    ,BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K

 ,TO-FILE-BLKSIZE = *UNCHANGED / *STD / *NK4 / <integer 1..16>
```

```
,TO-FILE-BLKCTRL = *UNCHANGED / *STD / *NK4

,REPLACE-OLD-FILES = *UNCHANGED / *NO / *YES / *DIALOG

,FILE-DISPOSAL = *UNCHANGED / *KEEP / *REPLACE / *INPLACE / *RENAME

,PROTECTION = *UNCHANGED / *STD / *SAME

,DEVICE-FOR-TEMPFILE = *UNCHANGED / *NONE / *TAPE(...) / *DISK(...)

    *TAPE(...)

        |   VOLUME = list-poss(100): <alphanum-name 1..6>

        |   ,DEVICE-TYPE = <text 1..20>

    *DISK(...)

        |   VOLUME = list-poss(100): <alphanum-name 1..6>

        |   ,DEVICE-TYPE = <text 1..20>
```

Operanden

DIRECTION = *UNCHANGED / *ONKEY / *NONKEY-TO-KEY
Bezeichnet die Richtung der Dateikonvertierung und die gewünschte Voreinstellung für die Richtung der Dateikonvertierung.

DIRECTION = *UNCHANGED
Der aktuell gültige Standardwert für DIRECTION soll nicht geändert werden.

DIRECTION = *TO-NONKEY
Die Dateikonvertierung soll ins NK-Format erfolgen.

DIRECTION = *NONKEY-TO-KEY
Die Dateikonvertierung soll vom NK-Format ins K-Format erfolgen.

SELECT = *UNCHANGED / *ALL / *BY-ATTRIBUTES(...)
Gibt an, ob die Dateien, die konvertiert werden sollen, über bestimmte Auswahlkriterien zusätzlich zum teilqualifizierten Dateinamen ausgewählt werden sollen.

SELECT = *UNCHANGED
Der aktuell gültige Standardwert für SELECT soll nicht geändert werden.

SELECT = *ALL
Für die Quelldateien werden keine Auswahlkriterien eingestellt.

SELECT = *BY-ATTRIBUTES(...)
Bestimmt für die zu konvertierenden Dateien die Auswahlkriterien, die anschließend als Standardwerte für die Anweisung CONVERT-FILE gelten.

CREATION-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY
Bezeichnet das Erstellungsdatum als Auswahlkriterium.

CREATION-DATE = *ANY
Das Erstellungsdatum wird nicht als Auswahlkriterium herangezogen. Alle Dateien werden bei der Auswahl berücksichtigt.

CREATION-DATE = *INTERVAL(...)
Dateien mit Erstellungsdatum innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.

FROM = 0000-01-01 / *YESTERDAY / <date>
Dateien mit Erstellungsdatum gleich oder später dem angegebenen Grenzwert werden selektiert.

FROM = 0000-01-01
Der untere Grenzwert ist das früheste mögliche Datum.

FROM = *YESTERDAY
Der untere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum >= dem gestrigen Tagesdatum werden selektiert.

FROM = <date>
Der untere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum >= dem angegebenen Wert werden selektiert.

TO = *TODAY / *YESTERDAY / <date>
Dateien mit Erstellungsdatum gleich oder früher dem angegebenen Grenzwert werden selektiert.

TO = *TODAY
Der obere Grenzwert ist das aktuelle Tagesdatum. Dateien mit Erstellungsdatum <= dem aktuellen Tagesdatum werden selektiert.

TO = *YESTERDAY
Der obere Grenzwert ist das gestrige Tagesdatum. Dateien mit Erstellungsdatum <= dem gestrigen Tagesdatum werden selektiert.

TO = <date>
Der obere Grenzwert ist das angegebene Datum. Dateien mit Erstellungsdatum <= dem angegebenen Wert werden selektiert.

LAST-ACCESS-DATE = *ANY / *INTERVAL(...) / <date> / *TODAY / *YESTERDAY
Bezeichnet das Datum des letzten Dateizugriffs als Auswahlkriterium.

Erklärung der Operanden ANY, INTERVAL(...), <date>, TODAY und YESTERDAY siehe Operand CREATION-DATE.

SIZE = *ANY / <integer 0..16777215> / *INTERVAL(...)
Bezeichnet die Dateigröße als Auswahlkriterium.

SIZE = *ANY
Die Dateigröße wird nicht als Auswahlkriterium herangezogen.

SIZE = <integer 0..16777215>
Dateien mit einer Dateigröße gleich dem angegebenen Wert werden selektiert.

SIZE = *INTERVAL(...)
Dateien mit einer Dateigröße innerhalb des angegebenen Intervalls werden selektiert. Die Grenzwerte für das Intervall legen die Operanden FROM und TO fest.

FROM = 0 / <integer 0..16777215>
Dateien mit einer Dateigröße >= dem angegebenen Grenzwert werden selektiert.

FROM = 0
Der untere Grenzwert ist die kleinste mögliche Größe.

FROM = <integer 0..16777215>
Der untere Grenzwert ist die angegebene Größe.

TO = 16777215 / <integer 0..16777215>
Dateien mit einer Dateigröße <= dem angegebenen Grenzwert werden selektiert.

TO = 16777215
Der obere Grenzwert ist die größte mögliche Größe.

TO = <integer 0..16777215>
Der obere Grenzwert ist die angegebene Größe.

FILE-STRUCTURE = *ANY / list-poss(3): *SAM / *ISAM / *PAM
Bezeichnet die Zugriffsmethode als Auswahlkriterium.

FILE-STRUCTURE = *ANY
Die Zugriffsmethode wird nicht als Auswahlkriterium herangezogen.

FILE-STRUCTURE = *SAM
Dateien mit Zugriffsmethode SAM werden selektiert.

FILE-STRUCTURE = *ISAM
Dateien mit Zugriffsmethode ISAM werden selektiert.

FILE-STRUCTURE = *PAM
Dateien mit Zugriffsmethode PAM werden selektiert.

BLKSIZE = *ANY / <integer 1..16>
Bezeichnet die Blockgröße als Auswahlkriterium.

BLKSIZE = *ANY
Die Blockgröße wird nicht als Auswahlkriterium herangezogen.

BLKSIZE = <integer 1..16>
Dateien mit der Blockgröße gleich dem angegebenen Wert werden selektiert.

BLKCTRL = *ANY / *PAMKEY / *NO / *DATA / *DATA2K / *DATA4K
Bezeichnet die Blockkontrolleigenschaft als Auswahlkriterium.

BLKCTRL = *ANY
Die Blockkontrolleigenschaft wird nicht als Auswahlkriterium herangezogen.

BLKCTRL = *PAMKEY
Dateien mit der Blockkontrolleigenschaft PAMKEY werden selektiert.

BLKCTRL = *NO
Dateien mit der Blockkontrolleigenschaft NO werden selektiert.

BLKCTRL = *DATA
Dateien mit der Blockkontrolleigenschaft DATA werden selektiert.

BLKCTRL = *DATA2K
Dateien mit der Blockkontrolleigenschaft DATA2K werden selektiert.

BLKCTRL = *DATA4K
Dateien mit der Blockkontrolleigenschaft DATA4K werden selektiert.

TO-FILE-BLKSIZE = *UNCHANGED / *STD / *NK4 / <integer 1..16>
Bezeichnet die logische Blockgröße der Zieldatei.

TO-FILE-BLKSIZE = *UNCHANGED
Der aktuell gültige Standardwert für TO-FILE-BLKSIZE soll nicht geändert werden.

TO-FILE-BLKSIZE = *STD
Die logische Blockgröße der Zieldatei wird nicht vom Benutzer definiert. PAMCONV richtet sich nach den Gegebenheiten des Ziel-Pubsets und setzt, wenn notwendig, den Blockungsfaktor herauf. Es wird intern maximal um eins heraufgeblockt.

TO-FILE-BLKSIZE = *NK4
Die logische Blockgröße der Zieldatei wird so gesteuert, dass sie auf jeden Fall geradzahlig ist, d.h. die Zieldatei kann auf einem NK4-Pubset abgelegt werden. Es wird intern maximal um eins heraufgeblockt.

TO-FILE-BLKSIZE = <integer 1..16>
Die Zieldatei wird mit der Blockgröße gleich dem angegebenen Wert erzeugt, wenn die Angabe mit den übrigen Bedingungen der Umblockung verträglich ist (siehe Abschnitt „Umblockung“).

TO-FILE-BLKCTRL = *UNCHANGED / *STD / *NK4
Bezeichnet die Blockkontroll-Information der Zieldatei. Dieser Operand ist nur für die Konvertierungsrichtung TO-NONKEY und ausschließlich für ISAM-Dateien relevant.

TO-FILE-BLKCTRL = *UNCHANGED
Der aktuell gültige Standardwert für TO-FILE-BLKCTRL wird nicht geändert.

TO-FILE-BLKCTRL = *STD
Die Blockkontroll-Information wird entsprechend dem Zielpubset gesetzt. Bei NK2-Pubsets wird das Datenformat mit DATA2K, bei NK4-Pubsets mit DATA4K festgelegt.

TO-FILE-BLKCTRL = *NK4
Die Blockkontroll-Information erhält den Wert DATA4K.

REPLACE-OLD-FILES = *UNCHANGED / *NO / *YES / *DIALOG
Gibt an, ob unter diesem Namen bereits existierende Dateien überschrieben werden sollen.

REPLACE-OLD-FILES = *UNCHANGED
Der aktuell gültige Standardwert für REPLACE-OLD-FILES soll nicht geändert werden.

REPLACE-OLD-FILES = *NO
Dateien sollen nicht überschrieben werden, die Konvertierung dieser Datei soll nicht durchgeführt werden (Standardwert).

REPLACE-OLD-FILES = *YES
Dateien sollen in jedem Fall überschrieben werden, sofern nicht ein zusätzlicher Schutz (Passwort, ACCESS=*READ,..) vorhanden ist.

REPLACE-OLD-FILES = *DIALOG
Der Benutzer wird im Dialog aufgefordert, die Verfahrensweise für bereits existierende Dateien zu bestimmen. Nur im Dialogbetrieb möglich.

FILE-DISPOSAL = *UNCHANGED / *KEEP / *RENAME / *REPLACE / *INPLACE
Gibt an, wie mit der Quell- und Zieldatei nach der Konvertierung verfahren werden soll.

FILE-DISPOSAL = *UNCHANGED
Der aktuell gültige Standardwert für FILE-DISPOSAL soll unverändert bleiben.

FILE-DISPOSAL = *KEEP
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Die Zieldateien existieren neben den Quelldateien.

FILE-DISPOSAL = *RENAME
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung sollen die Quelldateien gelöscht werden.

FILE-DISPOSAL = *REPLACE
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung sollen die Quelldateien gelöscht werden und die Zieldateien die Namen der Quelldateien erhalten. Effektiv wird die Quelldatei durch die Zieldatei ersetzt.

FILE-DISPOSAL = *INPLACE
Die Zieldateien sollen mit den Namen erzeugt werden, die für sie in der Konvertierungsanweisung angegeben sind. Nach erfolgreicher Konvertierung soll die Quelldatei mit der Zieldatei überschrieben und der Zieldatei der Name der Quelldatei gegeben werden. Dies bewirkt, dass die Zieldatei auch physikalisch in etwa den gleichen Platz wie die Quelldatei einnimmt. Effektiv wird die Quelldatei durch die Zieldatei ersetzt.

PROTECTION = *UNCHANGED / *STD / *SAME
Gibt an, ob die Dateischutzattribute der Quelldatei in die Zieldatei übernommen werden sollen.

PROTECTION = *UNCHANGED
Der aktuell gültige Standardwert für PROTECTION soll unverändert bleiben.

PROTECTION = *STD
Die Dateischutzattribute werden nicht übernommen.

PROTECTION = *SAME
Die Dateischutzattribute werden in die Zieldatei übernommen. Näheres siehe Abschnitt „Übernahme der Dateischutzattribute nach der Konvertierung“ (Besonderheiten bei der Konvertierung).

DEVICE-FOR-TEMPFILE = *UNCHANGED / *NONE / *TAPE(...) / *DISK(...)
Bezeichnet das Speichermedium, auf dem die erzeugte Zwischendatei abgelegt werden soll.

DEVICE-FOR-TEMPFILE = *UNCHANGED
Der aktuell gültige Standardwert für DEVICE-FOR-TEMPFILE soll unverändert bleiben.

DEVICE-FOR-TEMPFILE = *NONE
Es soll keine Zwischendatei auf einem privaten Datenträger abgelegt werden.

DEVICE-FOR-TEMPFILE = *TAPE(...)
Die erzeugte Zwischendatei soll auf Magnetband abgelegt werden.

VOLUME = list-poss(100): <alphanum-name 1..6>
Bezeichnet die VSN des Bandes bzw. der Bänder, falls mehrere VSNs in Form einer Liste angegeben werden, das bzw. die als Speichermedium dienen soll(en).

DEVICE-TYPE = <text 1..20>
Bezeichnet den Gerätetyp, der verwendet werden soll.

DEVICE-FOR-TEMPFILE = *DISK(...)
Die erzeugte Zwischendatei soll auf einer Privatplatte abgelegt werden soll.

VOLUME = list-poss(100): <alphanum-name 1..6>
Bezeichnet die VSN der Privatplatte bzw. der Platten, falls mehrere VSNs in Form einer Liste angegeben werden, die als Speichermedium dienen soll(en).

DEVICE-TYPE = <text 1..20>
Bezeichnet den Gerätetyp, der verwendet werden soll.

ℹ Der im Operanden DEVICE-FOR-TEMPFILE angegebene Datenträger wird erst während der Ausführung der Anweisung CONVERT-FILE reserviert bzw. freigegeben.

9.6.2.6 MODIFY-LOGGING-OPTIONS - Protokollierungswerte einstellen

Mit dieser Anweisung werden für PAMCONV Protokollierungs-Werte eingestellt, die global für alle Funktionen des Programmes gültig sind. Wird die Anweisung ohne Operanden eingegeben, behalten die eingestellten Werte ihre Gültigkeit. Die aktuell gültigen Werte können durch die Anweisung SHOW-LOGGING-OPTIONS abgefragt werden.

Format

MODIFY-LOGGING-OPTIONS
INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM ,OUTPUT = *UNCHANGED / list-poss(2): *SYSOUT / *SYSLST

Operanden

INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM
Steuert den Umfang des von PAMCONV erzeugten Protokolls.

INFORMATION = *UNCHANGED
Der aktuell gültige Standardwert für INFORMATION wird nicht geändert.

INFORMATION = *MEDIUM
Anweisungen werden nur im Fehlerfall protokolliert. Erfolgsmeldungen (Meldungsklasse 5) werden protokolliert (additiv zu MINIMUM).

INFORMATION = *MINIMUM
Es werden nur Fehlermeldungen, Schlussmeldungen und Nichterfolgsmeldungen protokolliert (alle Meldungsklassen außer 5).

INFORMATION = *MAXIMUM
Das erzeugte Protokoll besteht aus: Anweisungen, Erfolgsmeldungen, Nichterfolgsmeldungen, Fehler- und Schlussmeldungen.

OUTPUT = *UNCHANGED / list-poss(2): *SYSOUT / *SYSLST
Definiert das Ausgabemedium für die von PAMCONV erzeugten Protokolle.

OUTPUT = *UNCHANGED
Der aktuell gültige Standardwert für OUTPUT wird nicht geändert.

OUTPUT = list-poss(2): *SYSOUT / *SYSLST

Es ist möglich, eine Liste bestehend aus 2 Ausgabemedien SYSOUT und SYSLST anzugeben oder nur ein Ausgabemedium.

Ist SYSOUT angegeben, wird in die Systemdatei SYSOUT geschrieben (im Dialog ist dies standardmäßig die Datensichtstation). Ist SYSLST angegeben, werden Protokolle in die Systemdatei SYSLST geschrieben. Ist eine Liste angegeben, gehen die Ausgaben der Protokolle in beide Systemdateien.

> ¡ Folgende Werte sind voreingestellt:
> `INFORMATION=*MEDIUM, OUTPUT=*SYSOUT`

9.6.2.7 SHOW-CONVERT-FILE-DEFAULTS - Standardwerte für CONVERT-FILE auflisten

Durch die Anweisung SHOW-CONVERT-FILE-DEFAULTS werden die aktuell gültigen Standardwerte zur Anweisung CONVERT-FILE aufgelistet.

Format

SHOW-CONVERT-FILE-DEFAULTS

Diese Anweisung besitzt keine Operanden.

Das Ausgabeziel wird durch den Operanden OUTPUT der Anweisung MODIFY-LOGGING-OPTIONS bestimmt. Die eingestellten Werte werden in folgender Form aufgelistet:

```
%//SHOW-CONVERT-FILE-DEFAULTS
% CURRENT CONVERT-FILE DEFAULTS
%   DIRECTION                 : <wert>
%   SELECT                    : <wert>
%   TO-FILE-BLKSIZE           : <wert>
%   TO-FILE-BLKCTRL           : <wert>
%   REPLACE-OLD-FILES         : <wert>
%   FILE-DISPOSAL             : <wert>
%   PROTECTION                : <wert>
%   DEVICE-FOR-TEMPFILE       : <wert>
```

<wert> ... mögliche Werte siehe MODIFY-CONVERT-FILE-DEFAULTS.

> Die oben aufgeführten Textsegmente `CURRENT CONVERT-FILE DEFAULTS`, ..., `DEVICE-FOR-TEMPFILE` sind in der Meldungsdatei definiert und werden dieser entnommen. Sollen diese Textsegmente anders lauten, können sie in der Meldungsdatei neu definiert werden.

9.6.2.8 SHOW-LOGGING-OPTIONS - Eingestellte Protokoll-Optionen auflisten

Es wird das Auflisten der aktuell gültigen Werte für die Protokollierung angefordert.

Format

SHOW-LOGGING-OPTIONS

Diese Anweisung besitzt keine Operanden.

Das Ausgabeziel wird durch den Operanden OUTPUT der Anweisung MODIFY-LOGGING-OPTIONS bestimmt.

Die eingestellten Werte werden in folgender Form aufgelistet:

```
%//SHOW-LOGGING-OPTIONS
%CURRENT LOGGING OPTIONS
%  INFORMATION : <wert>
%  OUTPUT      : <wert>
```

<wert> ... mögliche Werte siehe MODIFY-LOGGING-OPTIONS.

> Die Textsegmente `CURRENT LOGGING OPTIONS`, `INFORMATION` und `OUTPUT` sind in der Meldungsdatei definiert und werden dieser entnommen. Sollen diese Textsegmente anders lauten, können sie in der Meldungsdatei von der Systemverwaltung neu definiert werden.

9.7 PAMCONV-Programmablauf

Das Programm liest die Steueranweisungen über SYSDTA ein. Meldungen gibt es über SYSOUT oder/und SYSLST aus, abhängig von den eingestellten LOGGING-OPTIONS (siehe Anweisungen MODIFY-LOGGING-OPTIONS, SHOW-LOGGING-OPTIONS).

Beispiel

```
/START-EXECUTABLE-PROGRAM FROM-FILE=PAMCONV                                  1.
%  BLS0523 ELEMENT 'PAMCONV', VERSION '<version>', TYPE 'L' FROM LIBRARY
':1OSH:$TSOS.SYSLNK.PAMCONV.<version>' IN PROCESS
%  BLS0524 LLM 'PAMCONV', VERSION '<version>' OF '<date> <time>' LOADED
%  PEA7000 <time> PAMCONV VERSION <version> STARTED IN BS2000 <version>
%  PEA7001 PLEASE ENTER PAMCONV STATEMENTS
%//CONVERT-FILE FROM-FILE=DAT*,SELECT=BY-ATTRIBUTES(CREATION-DATE=
   INTERVAL(,YESTERDAY),FILE-STRUCTURE=SAM),TO-FILE=NK.DAT*                   2.
%  PEA5000 CONVERSION TO NON-KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.DATEI1'; TARGET FILE: ':N:$USER0001.NK.DATEI1'
%  PEA5000 CONVERSION TO NON-KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.DATEI2'; TARGET FILE: ':N:$USER0001.NK.DATEI2'
%  PEA5000 CONVERSION TO NON-KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.DATEI3'; TARGET FILE: ':N:$USER0001.NK.DATEI3'
%  PEA5000 CONVERSION TO NON-KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.DATEI4'; TARGET FILE: ':N:$USER0001.NK.DATEI4'               3.
%//CONVERT-FILE FROM-FILE=DAT*,TO-FILE=NK.DAT*                               4.
%  PEA5000 CONVERSION TO NON-KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.DATEI5'; TARGET FILE: ':N:$USER0001.NK.DATEI5'               5.
%  PEA2103 'TO-FILE' ALREADY EXISTS. TARGET FILE: 'NK.DATEI1'
%  PEA2103 'TO-FILE' ALREADY EXISTS. TARGET FILE: 'NK.DATEI2'
%  PEA2103 'TO-FILE' ALREADY EXISTS. TARGET FILE: 'NK.DATEI3'
%  PEA2103 'TO-FILE' ALREADY EXISTS. TARGET FILE: 'NK.DATEI4'                 6.
%//MODIFY-CONVERT-FILE-DEFAULTS SELECT=BY-ATTRIBUTES(FILE-STRUCTURE=ISAM)    7.
%//SHOW-CONVERT-FILE-DEFAULTS                                                8.
% CURRENT   CONVERT-FILE   DEFAULTS
%   DIRECTION                  : TO-NONKEY
%   SELECT                     : BY-ATTRIBUTES( )
%      CREATION-DATE           : ANY
%      LAST-ACCESS-DATE        : ANY
%      SIZE                    : ANY
%      ACCESS-METHOD           : ISAM
%      BLKSIZE                 : ANY
%      BLKCTRL                 : ANY
%   TO-FILE-BLKSIZE            : STD
%   TO-FILE-BLKCTRL            : STD
%   REPLACE-OLD-FILES          : NO
%   FILE-DISPOSAL              : KEEP
%   PROTECTION                 : STD
%   DEVICE-FOR-TEMPFILE        : NONE                                          9.
%//CONVERT-FILE DIRECTION=NONKEY-TO-KEY,FROM-FILE=NK.DAT*,TO-FILE=K.DAT*     10.
%  PEA5001 CONVERSION FROM NON-KEY TO KEY FORMAT COMPLETED. SOURCE FILE:
   ':N:$USER0001.NK.DATEI5'; TARGET FILE: ':N:$USER0001.K.DATEI5'            11.
%//CLASSIFY-FILE FROM-FILE=*DAT*                                             12.
% CLASSIFY-FILE   *DAT*  DIRECTION = KEY-TO-NONKEY
%
% FILENAME                                                    ! CONVER- !
%                                                             ! TIBLE   !
% :N:$USER0001.DATEI1 ..................................      !   YES   !
% :N:$USER0001.DATEI2 ..................................      !   YES   !
```

```
% :N:$USER0001.DATEI3 ..............................         !   YES   !
% :N:$USER0001.DATEI4 ..............................         !   YES   !
% :N:$USER0001.DATEI5 ..............................         !   YES   !
% :N:$USER0001.K.DATEI5 ............................         !   YES   !
% :N:$USER0001.NK.DATEI1 ............................         !   YES   !
% :N:$USER0001.NK.DATEI2 ............................         !   YES   !
% :N:$USER0001.NK.DATEI3 ............................         !   YES   !
% :N:$USER0001.NK.DATEI4 ............................         !   YES   !
% :N:$USER0001.NK.DATEI5 ............................         !   YES   !
%            11      FILE(S)LISTED                                    13.
%//END                                                                14.
%  PEA7003 10:01:09/1.5323 PAMCONV TERMINATED ABNORMALLY              15.
```

1. Aufruf des Programms PAMCONV.
2. Eingabe der Anweisung CONVERT-FILE mit partiellen Dateinamen und Auswahlkriterien.
3. Quittung über erfolgte Dateikonvertierung.
4. Eingabe der Anweisung CONVERT-FILE mit partiellen Dateinamen, ohne weitere Auswahlkriterien.
5. Quittungen über erfolgte Dateikonvertierungen.
6. Datei existiert bereits, die Konvertierung dieser Datei wird daher nicht durchgeführt.
7. Einstellen von Auswahlkriterien für weitere CONVERT-FILE-Anweisungen.
8. Eingabe der Anweisung SHOW-CONVERT-FILE-DEFAULTS.
9. Ausgabe der mit der Anweisung SHOW-CONVERT-FILE-DEFAULTS angeforderten Werte. Die ausgegebenen Werte sind die Standardwerte für nachfolgende CONVERT-FILE-Anweisungen.
10. Eingabe der Anweisung CONVERT-FILE mit partiellen Dateinamen, gewünschte Konvertierungsrichtung NONKEY-TO-KEY.
11. Quittungen über erfolgte Dateikonvertierungen.
12. Eingabe der Anweisung CLASSIFY-FILE mit partiellen Dateinamen.
13. Ausgabe der mit der Anweisung CLASSIFY-FILE angeforderten Werte. Klassifizierung der Eingabedateien nach ihrer Konvertierbarkeit.
14. Eingabe der Anweisung END.
15. Programm PAMCONV wird abnormal beendet, da im PAMCONV-Lauf ein Fehler aufgetreten ist.

9.8 Fehlerbehandlung

Fehler bei den Steueranweisungen im Dialog-Betrieb

- Syntaktische Fehler: Fehlerhafte Anweisungen an das Programm werden mit entsprechenden Fehlermeldungen abgewiesen.
- Semantische Fehler : Mit dem Benutzer wird ein Fehlerdialog geführt.

Fehler bei den Steueranweisungen im Batch-Betrieb

Syntaktische oder/und semantische Fehler: Nach Erkennen einer fehlerhaften Anweisung werden alle Anweisungen bis zur nächsten STEP- oder END-Anweisung übergangen.

DVS-Fehlermeldungen, Fehlercode

Fehlermeldungen des DVS bekommt der Benutzer (fehlende Passwörter, ACCESS=*READ, Crypto-Kennwort, ...). Der gemeldete DVS-Fehlercode wird teilweise mit zusätzlicher PAMCONV-spezifischer Information ausgegeben.

Fehler während der Durchführung der Konvertierung

Treten während der Bearbeitung einer CONVERT-FILE-Anweisung Fehler auf, bevor die Zieldatei ohne Fehler abgeschlossen ist, wird die unvollständig erzeugte Zieldatei (evtl. Zwischendatei) gelöscht, die Quelldatei bleibt unabhängig vom explizit oder durch Voreinstellung spezifizierten Operanden FILE-DISPOSAL der Anweisung MODIFY-CONVERT-FILE-DEFAULTS erhalten. Das erfolgreiche Schließen der Zieldatei wird durch eine entsprechende Meldung quittiert. Für Fehler nach erfolgreichem Schließen der Zieldatei (evtl. als Zwischendatei) sind folgende Fälle möglich:

- Die Quelldatei lässt sich nicht korrekt schließen. Die Konvertierung ist erfolgreich durchgeführt, es werden aber die dem explizit oder durch Voreinstellung spezifizierten Operanden FILE-DISPOSAL entsprechenden Aktivitäten nicht mehr durchgeführt.
- Quell- und Zieldatei sind korrekt abgeschlossen, gemäß dem für die Konvertierung gültigen Wert des Operanden FILE-DISPOSAL soll die Quelldatei durch die Zieldatei ersetzt werden (REPLACE oder RENAME), das Löschen der Quelldatei lässt sich nicht durchführen. Es wird wie im zuvor beschriebenen Fall verfahren.
- Die Quelldatei soll durch die Zieldatei ersetzt werden, die Quelldatei wurde mit Erfolg gelöscht, aber die gemäß dem für die Konvertierung gültigen Wert REPLACE des Operanden FILE-DISPOSAL notwendige Umbenennung lässt sich nicht mehr durchführen. In diesem Fall ist das Endergebnis wie bei Spezifikation von RENAME im Operanden FILE-DISPOSAL.

Anschließend wird die nächste Anweisung ausgeführt.

Inkonsistente Dateien

Ist die Quelldatei eine ISAM- oder SAM-Datei oder eine Phase, wird sie während der Konvertierung einer strengen Prüfung auf formale Konsistenz unterworfen. Bei festgestellten Inkonsistenzen sind zwei Fälle zu unterscheiden:

- Die Inkonsistenz ist dergestalt, dass bei Fortsetzung der Konvertierung mit Datenverlust zu rechnen ist (z.B. defekte Indexeinträge). Die Konvertierung wird mit einer entsprechenden Meldung abgebrochen.
- Es liegt zwar eine formale Inkonsistenz vor, Datenverlust ist aber auf Grund der Inkonsistenz ausgeschlossen (z. B. reservierte Felder oder freie Blöcke haben nicht den Inhalt „binär null“). In diesem Fall wird lediglich eine entsprechende Warnung ausgegeben und die Konvertierung wird fortgesetzt.

Anschließend wird die nächste Anweisung ausgeführt.

Inkompatible Dateien

Die Formate ohne PAM-Schlüssel sind nicht voll kompatibel zu bedienen. Daher ergeben sich auch für die Konvertierung in das NK-Format verschiedene Inkompatibilitäten. Siehe dazu PAM-Key-Eliminierung im Handbuch „Einführung in das DVS“ [4].

Werden Inkompatibilitäten festgestellt, die eine Fortsetzung der Konvertierung verhindern, so wird die Konvertierung mit einer entsprechenden Meldung abgebrochen (Harte Inkompatibilität).

Werden Inkompatibilitäten festgestellt, die lediglich Performanceverschlechterungen bewirken, aber die Konvertierung nicht grundsätzlich verhindern, wird lediglich eine entsprechende Warnung ausgegeben und die Konvertierung wird fortgesetzt (Weiche Inkompatibilität).

Anschließend wird die nächste Anweisung ausgeführt.

9.9 Meldungen von PAMCONV

Die Meldungen des Dienstprogramms PAMCONV haben die Meldungsklasse PEA.

Siehe auch den Abschnitt „Meldungen und ihre Bedeutung“ (Darstellungsmittel).

10 PASSWORD Verschlüsseln von Kennwörtern

Versionsstand: PASSWORD V21.0C

In diesem Kapitel ist das Kennwortverschlüsselungsprogramm PASSWORD beschrieben.

Das Programm PASSWORD verschlüsselt Kennwörter in einem System ohne automatische Kennwortverschlüsselung, damit Dateien eingelesen werden können, die in einem System mit Kennwortverschlüsselung erstellt wurden.

Das Programm PASSWORD ist ein Hilfsprogramm für den Export von Dateien von einem System in ein anderes (z. B. mit dem Programm ARCHIVE oder HSMS). PASSWORD wird nur dann gebraucht, wenn Dateien in einem System mit Kennwortverschlüsselung erstellt und geschützt wurden und in einem System ohne Kennwortverschlüsselung verarbeitet werden sollen. PASSWORD verschlüsselt angegebene Kennwörter und trägt sie in die PASSWORD-Tabelle des Auftrags ein.

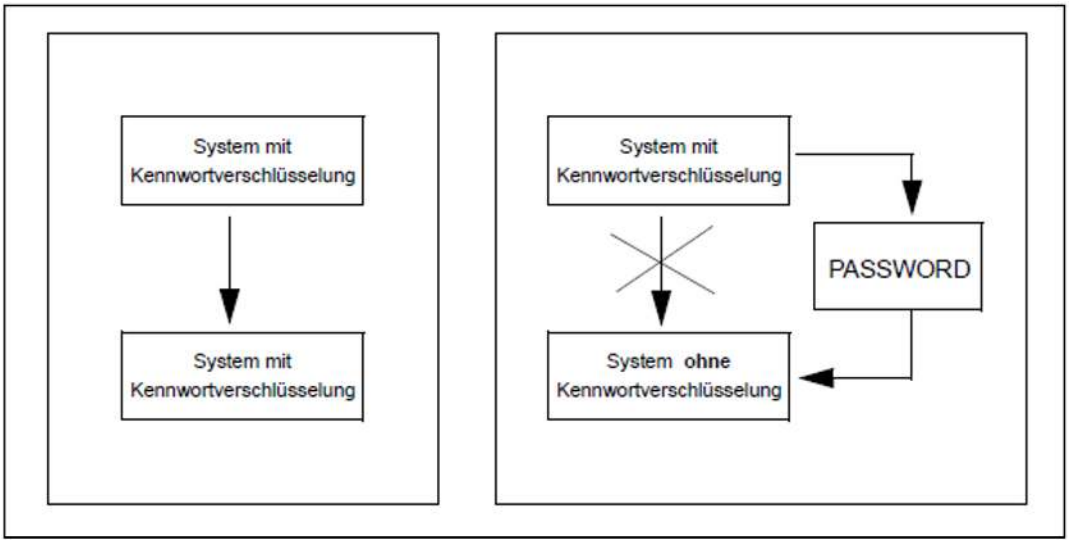


Bild 15: Kennwortverarbeitung in Systemen mit und ohne Kennwortverschlüsselung

10.1 Arbeitsweise und Programmablauf

Wenn ein Benutzer ein Datei- oder Logon-Kennwort angibt, wird dieses Kennwort in den Katalogeintrag der Datei oder - bei Logon-Kennwörtern - in den Benutzerkatalog geschrieben. In Systemen, die mit Kennwortverschlüsselung arbeiten, wird das vom Anwender eingegebene Kennwort nach einem internen Code verschlüsselt und so in den Katalogeintrag oder in den Benutzerkatalog übernommen. Der Anwender muss bei jedem Dateiaufruf oder /SET-LOGON-PARAMETERS das Kennwort in der unverschlüsselten Form angeben, da das System intern die Verschlüsselung vornimmt.
Zweck dieses Verfahrens ist es, zu verhindern, dass bei einem Speicherabzug zufällig Teile des Katalogs oder Teile des Benutzerkatalogs mit ausgegeben werden und dadurch Unbefugte Kenntnis von Kennwörtern erhalten.

Werden Dateien, deren Kennwörter verschlüsselt sind, in ein System ohne Kennwortverschlüsselung gebracht und gibt der Benutzer beim Dateizugriff das unverschlüsselte Kennwort an, so erhält er keinen Dateizugriff, da das Kennwort im Katalog verschlüsselt ist. Aufgabe von PASSWORD ist es, die unverschlüsselten Kennwörter in entsprechend codierte umzusetzen und damit den Zugriff auf eine Datei sicherzustellen.

10.1.1 Kennwörter

PASSWORD verarbeitet Dateikennwörter und Logon-Kennwörter.

Dateikennwörter schützen eine Datei

- vor unberechtigtem Schreiben (Schreibkennwort, Operand WRITE-PASSWORD von /MODIFY-FILE-ATTRIBUTES)
- vor unberechtigtem Lesen (Lesekennwort, Operand READ-PASSWORD von /MODIFY-FILE-ATTRIBUTES)
- vor unberechtigtem Ablauf bei Programm- und Prozedurdateien (Ausführungskennwort, Operand EXEC-PASSWORD von /MODIFY-FILE-ATTRIBUTES).

Die Stärke dieser Dateikennwörter ist hinsichtlich des Zugriffs gewichtet. Ein Schreibkennwort ist das stärkste mögliche Kennwort, darauf folgt das Lesekennwort, dann das Ausführungskennwort. Ein schwaches Kennwort muss nicht ausdrücklich angegeben werden, wenn ein stärkeres Kennwort bereits angegeben ist.

Dateikennwörter werden für eine oder mehrere Dateien fest vereinbart. Ob die richtigen Dateikennwörter vorhanden sind, wird vor jeder Ausführung der Systemkommandos /START-EXECUTABLE-PROGRAM, /LOAD-EXECUTABLE-PROGRAM, /MODIFY-FILE-ATTRIBUTES, /DELETE-FILE, /CREATE-FILE, /ADD-FILE-LINK, /COPY-FILE, /SHOW-FILE und außerdem bei der Dateieröffnung überprüft.

Um zu verhindern, dass Dateikennwörter bei Systemkommandos immer wieder
angegeben werden müssen, können sie während des Auftragsablaufs einmal in die PASSWORD-Tabelle geschrieben werden. Diese Tabelle wird immer dann nach dem passenden Kennwort durchsucht, wenn eine Datei mit Kennwortschutz bearbeitet werden soll.

Das **Logon-Kennwort** ist das Kennwort, das mit /SET-LOGON-PARAMETERS eingegeben werden muss, damit nur derjenige Benutzer unter einer Kennung arbeiten kann, der dazu berechtigt ist.

Im Zusammenhang mit ENTER-Jobs werden Logon-Kennwörter nur dann ausgewertet, wenn der Aufruf des ENTER-Jobs von einer Konsole erfolgt. Andernfalls werden alle eventuell angegebenen LOGON-Operanden, einschließlich Logon-Kennwort, im ENTER-Job ignoriert.

Formatvorschriften für Kennwörter

Dateikennwörter und *Jobvariablenkennwörter* dürfen maximal je 4 Bytes lang sein. Sie werden dargestellt in der Form:

C'x'	x = 1..4 alphanumerische Zeichen und Sonderzeichen.
X'n'	n = 1..8 Sedezimalziffern.
d	d ist eine Dezimalzahl mit höchstens 8 Ziffern und eventuell einem positiven oder negativen Vorzeichen, deren Wert in einen Binärwert umgesetzt wird.

Logon-Kennwörter dürfen 1 bis 32 Zeichen lang sein. Die Kennwörter werden in folgender Form dargestellt:

C'x' x = 1..32 alphanumerische Zeichen und Sonderzeichen, oder

X'n' n = 1..16 Sedezimalziffern.

> PASSWORD verarbeitet nur Logon-Kennwörter mit 1..8 Zeichen. Längere Logon-Kennwörter müssen zuvor auf 8 Zeichen verkürzt werden. Das Softwareprodukt SDF-P ermöglicht mit der vordefinierten Funktion HASH-STRING() diese Konvertierung (Beispiel siehe "ENCRYPTJ - Angegebenes Logon-Kennwort verschlüsseln").Weitere Details zu „langen Kennwörtern“ siehe auch `/MODIFY-USER-PROTECTION` im Handbuch „Kommandos“ [1]).

10.1.2 Programmablauf

Das Programm PASSWORD wird gestartet mit `/START-PASSWORD.`

START- PASSWORD
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>

Danach können Sie PASSWORD-Anweisungen eingeben. Jede Anweisung an PASSWORD wird sofort ausgeführt. Evtl. Ausgaben werden an der Datensichtstation (SYSOUT) angezeigt.

Nach vollständiger Ausführung einer Anweisung kann entweder eine neue PASSWORD-Anweisung eingegeben oder das Programm mit der END-Anweisung beendet werden.

10.2 Anweisungen

- Übersicht über die Anweisungen von PASSWORD
- Beschreibung der Anweisungen
  - CONVERT - Stärkstes Kennwort einer Datei verschlüsseln
  - ENCPASS - Dateikennwort verschlüsseln und in PASSWORD-Tabelle eintragen
  - ENCRYPTD - Angegebenes Dateikennwort verschlüsseln
  - ENCRYPTJ - Angegebenes Logon-Kennwort verschlüsseln
  - END - PASSWORD beenden
  - HELP - Hilfe zu PASSWORD ausgeben lassen
  - JVCONV - Kennwort für Jobvariable verschlüsseln
  - MODE - Verschlüsselungsroutine auswählen
  - PASSWORD - Dateikennwort verschlüsseln

10.2.1 Übersicht über die Anweisungen von PASSWORD

Die Anweisungen für PASSWORD werden über SYSDTA eingegeben.

Anweisung	Kurzbeschreibung
CONVERT	Für die angegebene Datei wird das stärkste Kennwort verschlüsselt in die PASSWORD-Tabelle eingetragen. Für die Datei wird `/MODIFY-FILE-ATTRIBUTES` mit den Kennwörtern eingegeben.
ENCPASS	Verschlüsselt ein angegebenes Kennwort und trägt die Verschlüsselung in die PASSWORD-Tabelle ein.
ENCPRYPTD	Verschlüsselt ein angegebenes Dateikennwort und gibt die Verschlüsselung nach SYSOUT aus.
ENCRYPTJ	Verschlüsselt ein angegebenes Logon-Kennwort und gibt die Verschlüsselung nach SYSOUT aus.
END	Beendet den PASSWORD-Programmlauf.
HELP	Gibt eine Kurzbeschreibung aller Anweisungen oder der angegebenen Anweisung aus.
JVCONV	Für die angegebene Jobvariable wird das stärkste Kennwort verschlüsselt in die PASSWORD-Tabelle eingetragen. Dann wird für die Jobvariable `/MODIFY-JV-ATTRIBUTES` mit den Kennwörtern eingegeben.
MODE	Bestimmt den Verschlüsselungsalgorithmus
PASSWORD	Verschlüsselt ein angegebenes Dateikennwort und trägt sowohl das Kennwort als auch die Verschlüsselung in die PASSWORD-Tabelle ein.

10.2.2 Beschreibung der Anweisungen

- CONVERT - Stärkstes Kennwort einer Datei verschlüsseln
- ENCPASS - Dateikennwort verschlüsseln und in PASSWORD-Tabelle eintragen
- ENCRYPTD - Angegebenes Dateikennwort verschlüsseln
- ENCRYPTJ - Angegebenes Logon-Kennwort verschlüsseln
- END - PASSWORD beenden
- HELP - Hilfe zu PASSWORD ausgeben lassen
- JVCONV - Kennwort für Jobvariable verschlüsseln
- MODE - Verschlüsselungsroutine auswählen
- PASSWORD - Dateikennwort verschlüsseln

10.2.2.1 CONVERT - Stärkstes Kennwort einer Datei verschlüsseln

Mit der CONVERT-Anweisung können ein bis drei Kennwörter einer Datei angegeben werden. Die CONVERT-Anweisung verschlüsselt das stärkste Kennwort mit `/ADD-PASSWORD` und gibt dann `/MODIFY-FILE-ATTRIBUTES` mit allen angegebenen Kennwörtern ein.

Nach der Ausführung des Kommandos muss für einen nachfolgenden Zugriff das Kennwort mit `/ADD-PASSWORD` in die PASSWORD-Tabelle eingetragen werden.

Format

CONVERT
filename { ,WRPASS = fpassword / ,RDPASS = fpassword / ,EXPASS = fpassword }

Operanden

filename
Vollqualifizierter Name der Datei, deren Kennwörter verschlüsselt werden.

WRPASS = fpassword
gibt ein Schreibschutzkennwort an, das verschlüsselt wird.

RDPASS = fpassword
gibt ein Lesekennwort an, das verschlüsselt wird.

EXPASS = fpassword
gibt ein Ausführungskennwort an, das verschlüsselt wird.

> fpassword ist ein 1 bis 4 Bytes langes Dateikennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“). Es muss mindestens eines der drei Kennwörter angegeben werden.

Beispiele

1. PASSWORD-Lauf mit Angabe eines Kennworts:

```
/start-password
ENTER COMMAND NOW :
*convert anne.3,rdpass=c'susi'
PASSWORD X'9A41632A'
CAT     ANNE.3,RDPASS=C'SUSI',STATE=U
*end
PASSWORD : NORMAL   END
/
```

2. Gezeigt wird, welches Kennwort bei mehreren Angaben verschlüsselt wird:

```
/start-password
ENTER COMMAND NOW :
*convert pass.test,wrpass=c'susi',rdpass=c'anne',expass=c'otto'
PASSWORD X'9A41632A'
CAT     PASS.TEST,WRPASS=C'SUSI',RDPASS=C'ANNE',EXPASS=C'OTTO',STATE=U
*convert pass.test,expass=c'otto',rdpass=c'anne'
PASSWORD X'7A08FCC2'
CAT     PASS.TEST,EXPASS=C'OTTO',RDPASS=C'ANNE',STATE=U
*convert pass.test,expass=c'otto'
PASSWORD X'DE28062F'
CAT     PASS.TEST,EXPASS=C'OTTO',STATE=U
*end
PASSWORD : NORMAL   END
/
```

10.2.2.2 ENCPASS - Dateikennwort verschlüsseln und in PASSWORD-Tabelle eintragen

Die ENCPASS-Anweisung verschlüsselt ein angegebenes Dateikennwort und trägt die Verschlüsselung in die PASSWORD-Tabelle des Prozesses ein. Das verschlüsselte Kennwort wird zusätzlich nach SYSOUT ausgegeben.

Zweck der Anweisung

Eine geschützte Datei wurde aus einem System mit Kennwortverschlüsselung in ein System ohne Kennwortverschlüsselung gebracht (z.B. mit ARCHIVE). Im Katalogeintrag der Datei steht ein verschlüsseltes Kennwort. Auf die Datei kann also nur derjenige zugreifen, der das verschlüsselte Kennwort angibt.

Mit ENCPASS wird das angegebene Kennwort verschlüsselt in die PASSWORD-Tabelle eingetragen und der Anwender erhält somit auf die mit dem angegebenen Kennwort geschützte Datei Zugriff.

Format

ENCPASS / EP
fpassword

Operanden

fpassword
1 bis 4 Byte langes Dateikennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“).

Beispiel

```
/start-password
ENTER COMMAND NOW :
*encpass c'susi'
PASSWORD X'9A41632A'
*end
PASSWORD : NORMAL   END
/
```

10.2.2.3 ENCRYPTD - Angegebenes Dateikennwort verschlüsseln

Die ENCRYPTD-Anweisung verschlüsselt ein angegebenes Dateikennwort und gibt die Verschlüsselung nach SYSOUT aus.

Zweck der Anweisung

Eine geschützte Datei wurde aus einem System mit Kennwortverschlüsselung in ein System ohne Kennwortverschlüsselung gebracht (z.B. mit ARCHIVE). Im Katalogeintrag der Datei steht ein verschlüsseltes Kennwort. Auf die Datei kann also nur derjenige zugreifen, der das verschlüsselte Kennwort angibt.

Die ENCRYPTD-Anweisung verschlüsselt das Ausgangskennwort. Im Gegensatz zur ENCPASS-Anweisung erhält der Anwender erst durch die Eingabe des verschlüsselten Kennworts, z.B. mit ADD-PASSWORD, wieder Zugriff auf die Datei.

Format

ENCRYPTD / ED
fpassword

Operanden

fpassword
1 bis 4 Byte langes Dateikennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“).

Beispiel

```
/start-password
ENTER COMMAND NOW :
*encryptd c'susi'
PASSWORD ENCRYPTED IS = X'9A41632A'
*end
PASSWORD : NORMAL   END
/
```

10.2.2.4 ENCRYPTJ - Angegebenes Logon-Kennwort verschlüsseln

Die ENCRYPTJ-Anweisung verschlüsselt ein angegebenes Logon-Kennwort und gibt die Verschlüsselung über SYSOUT aus.

Format

ENCRYPTJ / EJ
lpassword

Operanden

lpassword
1 bis 8 Byte langes Logon-Kennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“).

> **i** In BS2000 können auch längere Logon-Kennwörter mit maximal 32 Zeichen vereinbart werden. Um solche Kennwörter mit PASSWORD verarbeiten zu können, müssen sie auf Systemebene auf 8 Zeichen verkürzt werden.
> Das Softwareprodukt SDF-P ermöglicht mit der vordefinierten Funktion HASH-STRING() diese Konvertierung:
>
> ```
> (TO-X-LIT(HASH-STRING('<long password>',8)))
> ```

Beispiele

1. Verschlüsselung eines Kennworts, das 4 Zeichen lang ist.

```
ENTER COMMAND NOW :
*encryptj c'susi'
PASSWORD ENCRYPTED IS = X'6B0537211705D615'
*end
PASSWORD : NORMAL   END
```

2. Verschlüsselung eines Kennworts, das 14 Zeichen lang ist.

```
/A=(TO-X-LIT(HASH-STRING('longpassword',8)))
/show-var a
A = X'523E146036CED784'
/start-password
ENTER COMMAND NOW :
*encryptj x'523E146036CED784'
PASSWORD ENCRYPTED IS = X'73FE57A922EA780D'
*end
PASSWORD : NORMAL   END
```

10.2.2.5 END - PASSWORD beenden

Die END-Anweisung beendet den Programmlauf von PASSWORD.

Format

END

Diese Anweisung besitzt keine Operanden.

Die normale Programmbeendigung zeigt das Programm an mit der Meldung

```
PASSWORD : NORMAL END
```

10.2.2.6 HELP - Hilfe zu PASSWORD ausgeben lassen

Die HELP-Anweisung gibt entweder alle Anweisungen des Programms PASSWORD mit ihren zulässigen Operanden aus oder gibt zu einer angegebenen Anweisung diese Anweisung mit allen zusätzlichen Operanden aus.

Format

HELP / H
anweisung

Operanden

anweisung
Name einer PASSWORD-Anweisung in voller Länge: CONVERT, JVCONV, ENCRYPTD, ENCRYPTJ, ENCPASS, MODE, PASSWORD oder END

Es ist aber auch die Angabe von Kurzformen der Anweisungen erlaubt, wie sie beim Format der Anweisungen beschrieben sind.

10.2.2.7 JVCONV - Kennwort für Jobvariable verschlüsseln

Die JVCONV-Anweisung verschlüsselt für eine angegebene Jobvariable die Kennwörter. Zusätzlich zum Jobvariablen-Namen muss mindestens ein Kennwort angegeben werden. Die JVCONV-Anweisung verschlüsselt das stärkste Kennwort und setzt dann `/MODIFY-JV-ATTRIBUTES` mit allen angegebenen Kennwörtern ab.

Nach der Ausführung des Kommandos muss für einen nachfolgenden Zugriff das Kennwort mit `/ADD-PASSWORD` in die PASSWORD-Tabelle eingetragen werden.

Format

JVCONV
jvname { ,WRPASS = jpassword / ,RDPASS = jpassword }

Operanden

jvname
1 bis 4 Byte langes Jobvariablenkennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“). Es muss mindestens eines der Kennwörter angegeben werden.

WRPASS=jpassword
gibt ein Schreibschutzkennwort an, das verschlüsselt wird.

RDPASS=jpassword
gibt ein Lesekennwort an, das verschlüsselt wird.

Beispiel

```
/start-password
ENTER COMMAND NOW :
*jvconv hugo,wrpass=c'susi'
PASSWORD X'9A41632A'
CATJV   HUGO,WRPASS=C'SUSI',STATE=U
*end
PASSWORD : NORMAL   END
```

10.2.2.8 MODE - Verschlüsselungsroutine auswählen

Die MODE-Anweisung wählt die Verschlüsselungsroutine, die in BS2000 verwendet werden soll. Zur Auswahl stehen die „alte“ und die „SCA85“ Verschlüsselungsroutine.

Format

MODE / M
OLD / SCA

Operanden

OLD / O
Es soll die standardmäßige Voreinstellung gewählt werden.

SCA
Es soll die Verschlüsselungsroutine „SCA85“ gewählt werden.

10.2.2.9 PASSWORD - Dateikennwort verschlüsseln

Die PASSWORD-Anweisung verschlüsselt ein angegebenes Dateikennwort. Die Verschlüsselung und das unverschlüsselte Kennwort werden in der PASSWORD-Tabelle des Auftrags eingetragen. Beide Kennwörter werden zusätzlich nach SYSOUT ausgegeben. Ein Dateikennwort, das verschlüsselt im Katalogeintrag steht, kann also unverschlüsselt beim Dateizugriff angegeben werden, ohne dass der Dateizugriff wegen Nichtbefugnis zurückgewiesen wird.

Format

PASSWORD
fpassword

Operanden

fpassword
1 bis 4 Bytes langes Dateikennwort (Format siehe Abschnitt „Formatvorschriften für Kennwörter“).

Beispiel

```
/start-password
ENTER COMMAND NOW :
*password c'susi'
PASSWORD X'9A41632A'
PASSWORD C'SUSI'
*end
PASSWORD : NORMAL   END
```

11 PVSREN Umbenennen von Pubsets und Volume-Sets

Versionsstand: PVSREN V21.0C

PVSREN V21.0A ist ab BS2000 V21.0A ablauffähig und unterstützt SF- und SM-Pubsets.

Das Dienstprogramm PVSREN (Public Volume Sets REName) bietet folgende Funktionen an:

- Umbenennen von Pubsets und Volume-Sets innerhalb einer Notationsart
- Konvertieren von einer in die andere Notationsart
- Erzeugen neuer, eigenständiger Pubsets aus Spiegel-Pubsets, die mit den lokalen Replikationsfunktionen externer Plattenspeichersysteme erstellt wurden

Pubsets und Volume-Sets können innerhalb einer Notationsart umbenannt oder in die jeweils andere Notation konvertiert werden (von PUB-Notation in Punkt-Notation und umgekehrt), ohne dass ein erneutes Initialisieren notwendig ist.

Bei der Umbenennung von Pubsets und bei der Konvertierung von Pubset-Notationen gelten bestimmte Einschränkungen hinsichtlich der systemweit eindeutigen Adressierbarkeit von Pubsets und der maximalen Dateinamenslänge.

PVSREN unterstützt die Replikationsfunktionen externer Plattenspeichersysteme zum Erstellen von eigenständigen Pubsets, indem Spiegel-Pubsets umbenannt werden. Bei SM-Pubsets werden die Umbennungsregeln für die Volume-Sets in einer Parameterdatei abgelegt. In einigen Fällen erzeugt PVSREN die einzelnen Spiegel-Pubsets selbst, in anderen nutzt es die mit SHC-OSD erstellte Spiegel-Pubsets (siehe Anweisung CREATE-PUBSET-FROM-MIRROR bzw. Handbuch „SHC-OSD“ [22]).

Auch bereits abgetrennte Spiegel-Pubsets, deren VSNs der sog. „Doppelpunkt-Notation“ genügen, kann PVSREN zu eigenständigen Pubsets machen.

> ¡ In diesem Kapitel werden Kopien von Pubsets, die mit Replikationsfunktionen externer Plattenspeichersysteme erzeugt wurden, kurz **Spiegel-Pubsets** genannt.

Mit PVSREN können Spiegel-Pubsets, die in Doppelpunkt-Notation erstellt wurden (SPECIAL-VSN), auf ihren ursprünglichen Namen zurückbenannt werden. Die Rückbenennung ist in Einheiten von SM-/SF-Pubsets und Volume-Sets möglich.

Zum Aufbau von SF- und SM-Pubsets siehe Handbuch „Systembetreuung“ [5].

11.1 Voraussetzungen für PVSREN

- Voraussetzungen für den Ablauf von PVSREN
- Voraussetzung für eine Umbenennung
- Voraussetzungen für das Rückbenennen von Spiegel-Pubsets
- Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets

11.1.1 Voraussetzungen für den Ablauf von PVSREN

PVSREN benennt Pubsets um oder konvertiert Pubset-Notationen. Im Folgenden wird zur Vereinfachung nur von „Umbenennen von Pubsets“ gesprochen.

Folgende grundlegenden Voraussetzungen gelten für den Ablauf von PVSREN:

- PVSREN ist nur unter der Benutzerkennung TSOS ablauffähig.
- Die Standard-Katalogkennung kann nur mit dem Systemprivileg USER-ADMINISTRATION geändert werden.
- PVSREN nutzt den SVC 79, deshalb muss der Systemparameter SVC79 auf 0 oder 1 gesetzt sein.

11.1.2 Voraussetzung für eine Umbenennung

Wenn eine der folgenden Bedingungen nicht erfüllt ist, wird der Pubset nicht umbenannt.

- Der Pubset muss den Status INACCESSIBLE (unerreichbar) haben, da sonst während der Umbenennung unerwünschte Zugriffe auf den Katalog möglich wären.

  Wenn ein Volume-Set umbenannt werden soll, muss zuerst der zugehörige SM-Pubset exportiert werden. Es ist nicht möglich, einen Volume-Set zu exportieren.
- Alle benötigten Platten müssen an dem System, an dem die Umbenennung durchgeführt wird, verfügbar sein.

  Beim Umbenennen eines SM-Pubsets oder seines Control Volume Sets werden alle Platten des Control-Volume-Sets und zusätzlich die Volres aller zum SM-Pubset gehörenden Volume-Sets benötigt.

  Beim Umbenennen eines „normalen“ Volume-Sets werden alle Platten des Volume-Sets und zusätzlich noch alle Platten des Control-Volume-Sets benötigt.
- PVSREN muss alle benötigten Platten exklusiv belegen können.

  Beim Umbenennen eines SM-Pubsets darf nicht gleichzeitig ein Volume-Set dieses Pubsets umbenannt werden.

  Das Umbenennen eines Volume-Sets ist nur dann möglich, wenn nicht gleichzeitig ein anderer Volume-Set oder der SM-Pubset, zu dem der Volume-Set gehört, umbenannt werden.
- Es dürfen keine Platten mit der Ziel-VSN vorhanden sein.

  Beispiel: Umbenennung von „ABC“ auf „X“

  Der Pubset ABC besteht aus den Platten ABC.00 bis ABC.05:
  In diesem Fall dürfen die Platten PUBX00 bis PUBX05 nicht vorhanden sein.
- Wenn ein MRSCAT-Eintrag mit der Ziel-Katalogkennung vorhanden ist, darf er nicht den Zustand ACCESSIBLE haben.
  Hat dieser MRSCAT-Eintrag den Zustand INACCESSIBLE, erfolgt die Umbenennung nur nach Rückfrage über die Meldung `PVR0205`.
- Beim Umbenennen eines SM-Pubsets oder eines Control-Volume-Sets darf der SM-Pubset keinen Volume-Set mit dem Status DEFECT enthalten.
- Das Umbenennen eines Volume-Sets mit dem Status DEFECT ist nicht möglich.
- Die Länge von Datei-, Dateigenerationsgruppen- und Jobvariablennamen darf die Maximallänge nicht überschreiten. Die Anweisung CHECK-FILENAME-LENGTH führt die Längenprüfung durch. Diese Prüfung ist nur dann notwendig, wenn sich die Länge der Katalogkennung erhöht.

  Für Volume-Sets ist die Längenprüfung nicht notwendig, da die Volume-Set-Kennung nicht in den Dateinamen eingeht.

Weiter sollten vor einer Umbenennung folgende Bedingungen beachtet werden, um einen reibungslosen Ablauf sicherzustellen:

- Wenn ein MRSCAT-Eintrag mit der Ziel-Katalogkennung vorhanden ist, sollte eine Umbenennung auf diese Katalogkennung nur dann durchgeführt werden, wenn dieser MRSCAT-Eintrag nicht mehr benötigt wird. PVSREN löscht diesen MRSCAT-Eintrag und legt ihn neu an.
- Alle Privatdateien, die auch nach der Umbenennung genutzt werden, sollten einen Katalog-Eintrag im Katalog dieses Pubsets haben, da nur dann die Längenprüfung für diese Dateinamen durchgeführt wird.
- Es ist darauf zu achten, dass sich keine Enterjobs im Jobpool befinden, da diese nach der Umbenennung möglicherweise nicht mehr ablauffähig sind.

- Vor einer Umbenennung sollten alle Druckaufträge für Dateien, die sich auf dem umzubenennenden Pubset befinden, erledigt sein. Diese Aufträge würden sonst in der Warteschlange bleiben: der benötigte Pubset kann nicht mehr importiert werden, da er umbenannt wurde.
- Wird ein Home-Pubset umbenannt, muss Folgendes berücksichtigt werden:
  - Beim Startup werden in vielen Rechenzentren automatisch Enterjobs gestartet, in denen die Katalogkennung definiert ist (z.B. ENTER zum Laden des SPEEDCAT). Diese nicht mehr gültige Katalogkennung muss durch die neue ersetzt werden.

    Falls die von IMON erstellte Quelldatei des Subsystemkatalogs <subsystemkatalog>.SRC verwendet wird, muss auch in dieser die nicht mehr gültige Katalogkennung durch die neue ersetzt werden.
  - Ein FAST-Startup wird mittels einer Parameterdatei durchgeführt. Teile, die sich auf die VSN beziehen, müssen geändert werden.
- Eine Konvertierung von PUB-Notation in Punktnotation ist immer möglich. Eine Konvertierung von Punktnotation in PUB-Notation ist nur dann möglich, wenn die Anzahl der Volumes nicht größer als 32 ist.

11.1.3 Voraussetzungen für das Rückbenennen von Spiegel-Pubsets

Bei der Rückbenennung von Spiegel-Pubsets müssen folgende Voraussetzungen erfüllt sein:

- Alle benötigten Platten müssen an dem System verfügbar sein, an dem die Umbenennung durchgeführt werden soll.
- PVSREN muss alle benötigten Platten exklusiv belegen können.

11.1.4 Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets

Die folgenden Bedingungen müssen erfüllt sein, damit der Pubset erzeugt werden kann:

- Damit die Platten des neuen Pubset einen konsistenten Stand haben, muss der Pubset den Status INACCESSIBLE (unerreichbar) haben, wenn die Replikation aktiv ist.

  Wenn die Spiegel-Pubsets bereits zugreifbar sind (also getrennt bzw. aktiviert) und die VSNs der zugreifbaren Kopien der Doppelpunkt-Notation genügen, spielt der Status des Pubsets keine Rolle.
- Sowohl SF- als auch SM-Pubsets werden als Ganzes bearbeitet. Für SM-Pubsets muss die Umbenennungsvorschrift für jeden Volume-Set in einer Parameterdatei hinterlegt werden (siehe Anweisung SET-NAME-OF-NEW-VOLUMESET). Es dürfen auch Volume-Sets im Zustand DEFINED-ONLY vorhanden sein.
- Alle benötigten Platten müssen an dem System, an dem der Pubset erzeugt werden soll, verfügbar sein.
  - Falls die Spiegel-Paare nicht bereits getrennt zugreifbar sind, handelt es sich dabei um alle Platten des Ausgangs-Pubsets mit den entsprechenden Plattenkopien; beim Arbeiten mit Snap-Funktionen muss die ensprechende Anzahl freier virtueller Platten verfügbar sein.
  - Falls die Spiegel-Paare bereits getrennt zugreifbar sind, werden nur die Kopien aller Platten des Pubsets mit VSNs in Doppelpunkt-Notation benötigt.
- PVSREN muss alle benötigten Platten exklusiv belegen können.
- Es dürfen keine Platten mit den Ziel-VSNs vorhanden sein.

  Beispiel: Aus dem Spiegel-Pubset des Pubsets ABC soll Pubset XYZ erzeugt werden.

  Der Pubset ABC besteht aus den Platten ABC.00 bis ABC.05.
  In diesem Fall dürfen die Platten XYZ.00 bis XYZ.05 nicht bereits vorhanden sein
- Wenn ein oder mehrere MRSCAT-Einträge mit der oder den Ziel-Katalogkennungen vorhanden sind, darf keiner von ihnen den Zustand ACCESSIBLE haben. Haben all diese MRSCAT-Einträge den Zustand INACCESSIBLE, erfolgt die Umbenennung nur nach Rückfrage über die Meldung `PVR0205`.
- Beim Erzeugen von Pubsets aus Spiegel-Pubsets darf die Kopie noch nicht abgetrennt bzw. aktiviert sein oder sie muss in Doppelpunkt-Notation benannt sein. Abgetrennte Kopien mit der Katalogkennung des ursprünglichen Pubsets oder mit einer neuen Katalogkennung verarbeitet PVSREN nicht.
- Beim Erzeugen von SM-Pubsets aus Spiegel-Pubsets darf der ursprüngliche SM-Pubset keinen Volume-Set mit dem Status DEFECT enthalten.
- Die Länge der Katalogkennung von Ausgangs-Pubset und neu zu erzeugendem Pubset muss gleich sein.
- Wenn MRSCAT-Einträge mit Katalogkennungen des neu zu erzeugenden Pubsets und/oder von einem oder mehreren seiner Volume-Sets bereits vorhanden sind, sollte die Erzeugung eines Pubsets mit diesen Katalogkennungen nur dann durchgeführt werden, wenn die entsprechenden MRSCAT-Einträge nicht mehr benötigt werden. PVSREN löscht diese MRSCAT-Einträge und legt sie neu an.

11.2 Ablauf von PVSREN

PVSREN benennt Pubsets oder Volume-Sets um oder konvertiert Pubset-Notationen. Im Folgenden wird zur Vereinfachung nur von „Umbenennen von Pubsets“ gesprochen. Außerdem kann PVSREN aus Spiegel-Pubsets eines SF- oder SM-Pubsets neue, eigenständige Pubsets erzeugen.

Wird sowohl auf Dateien, Dateigenerationsgruppen und Jobvariablen Bezug genommen, so wird allgemein von „Objekt“ gesprochen.

Das Umbenennen erfolgt mit der Anweisung RENAME-PUBSET-OR-VOLUME-SET. Diese Anweisung unterstützt SF-Pubsets, SM-Pubsets und Volume-Sets. Für das Umbenennen eines Pubsets oder Volume-Sets kann zwischen zwei verschiedenen Modi gewählt werden, die über den Operanden MODE gesteuert werden:

1. Vollständiges Umbenennen eines Pubsets oder Volume-Sets.
   In diesem Fall werden die Katalogkennung, Benutzerkataloge und das SCI angepasst.
2. Umbenennung eines Pubsets oder Volume-Sets nach einer Trennung durch SHC-OSD. In diesem Fall werden die Benutzerkataloge und das SCI angepasst.

Eine vollständige Umbenennung sollte erst durchgeführt werden, wenn alle Voraussetzungen geprüft wurden (siehe Abschnitt „Voraussetzungen für den Ablauf von PVSREN“). Es sind auch die „Einschränkungen und Nacharbeiten“ zu beachten.

Neue Pubsets aus Spiegel-Pubsets werden mit der Anweisung CREATE-PUBSET-FROM-MIRROR erzeugt. Diese Anweisung unterstützt SF- und SM-Pubsets.
Für SM-Pubsets muss in der Parameterdatei angegeben werden, wie die einzelnen Volume-Sets benannt werden sollen. Dies erfolgt für jeden einzelnen Volume-Set mit der Anweisung SET-NAME-OF-NEW-VOLUME-SET.
Bevor ein Pubset erzeugt wird, sollten die im Abschnitt „Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets“ (Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel- Pubsets) genannten Voraussetzungen geprüft werden.

Umbenennen eines SF-Pubsets

1. Es wird eine Längenprüfung der Objektnamen durchgeführt, auch wenn vorher die Anweisung CHECK-FILENAME-LENGTH abgearbeitet wurde. Dabei werden alle Objekte aufgelistet, deren Name die maximal zulässige Länge überschreitet. Die Umbenennung wird erst durchgeführt, wenn keine zu langen Objektnamen mehr vorhanden sind.
2. In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen wird die VSN geändert.
3. Im SVL aller zum Pubset gehörenden Platten wird die VSN geändert.
4. Im OLC (Online Catalog auf der Pubres) wird die VSN aller zum Pubset gehörenden Platten geändert.
5. Für den SF-Pubset wird der alte MRSCAT-Eintrag gelöscht und ein neuer mit der neuen Katalogkennung angelegt.
6. Die Standard-Katalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubset und des umbenannten Pubsets wird geändert. Diese Aktion wird im Dialogbetrieb nur nach Rückfrage und im Batchbetrieb standardmäßig durchgeführt. Dazu wird der SF-Pubset importiert.
   Die Änderung der Standardkatalogkennung erfolgt mit `/MODIFY-USER-ATTRIBUTES`.
7. Ändern der betroffenen Einträge im IMON-SCI auf dem Daten-Pubset und (nach Rückfrage) auch auf dem Home-Pubset.
8. Der Anwender kann entscheiden, ob der Pubset verfügbar werden bzw. bleiben soll.

Umbenennen eines SM-Pubsets

1. Wenn zum MRSCAT-Eintrag eines SM-Pubsets auch eine Platte mit der entsprechenden VSN (z.B. ABC.00 bei catid=ABC) existiert, wird die Umbenennung abgebrochen, da nicht eindeutig ist, ob der SM-Pubset oder der gleichzeitig vorhandene SF-Pubset umbenannt werden soll.
2. Es wird eine Längenprüfung der Objektnamen durchgeführt, auch wenn vorher die Anweisung CHECK-FILENAME-LENGTH ausgeführt wurde. Dabei werden alle Objekte aufgelistet, deren Name die maximal zulässige Länge überschreitet. Die Umbenennung wird erst durchgeführt, wenn keine zu langen Objektnamen mehr vorhanden sind.
3. Die Katalogkennung des SM-Pubsets wird in der Pubset-Konfigurationsdatei auf dem Control-Volume-Set geändert.
4. Im SVL jeder Volres aller zum SM-Pubset gehörenden Volume-Sets wird die Katalogkennung des SM-Pubsets geändert.
5. Die MRSCAT-Einträge aller zum SM-Pubset gehörenden Volume-Sets werden gelöscht. Sie werden beim Importieren des SM-Pubsets wieder neu angelegt.
6. Für den SM-Pubset wird der alte MRSCAT-Eintrag gelöscht und ein neuer mit der neuen Katalogkennung angelegt.
7. Die Standard-Katalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubsets und des umbenannten Pubsets wird geändert. Diese Aktion wird im Dialogbetrieb nur nach Rückfrage und im Batchbetrieb standardmäßig durchgeführt. Dazu wird der SM-Pubset importiert.
   Die Änderung der Standardkatalogkennung erfolgt mit `/MODIFY-USER-ATTRIBUTES`.
8. Ändern der betroffenen Einträge im IMON-SCI auf dem Daten-Pubset und nach Rückfrage auch auf dem Home-Pubset.
9. Der Anwender kann entscheiden, ob der Pubset verfügbar werden bzw. bleiben soll.

Umbenennen eines Volume-Sets

1. Die Katalogkennung des Volume-Sets wird in der Pubset-Konfigurationsdatei auf dem Control-Volume-Set geändert.
2. In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen wird die VSN geändert.
3. Im SVL aller zum Volume-Set gehörenden Platten wird die VSN geändert.
4. Im OLC (Online Catalog auf der Volres) wird die VSN aller zum Volume-Set gehörenden Platten geändert.
5. Eine Längenprüfung der Objektnamen ist nicht notwendig, da die Katalogkennung des Volume-Sets nicht in den Dateinamen eingeht.
6. Die Katalogkennung des Volume-Sets kann nicht als Standardkatalogkennung verwendet werden, deshalb ist eine Änderung der Standardkatalogkennung nicht notwendig. Nach dem Umbenennen wird der zugehörige SM-Pubset nicht importiert. Diese Aktion muss vom Anwender durchgeführt werden.

Umbenennen eines Control-Volume-Sets

1. Die Katalogkennung des Control-Volume-Sets wird in der Pubset-Konfigurationsdatei auf dem Control-Volume-Set geändert.
2. In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen wird die VSN geändert.
3. Im SVL aller zum Control-Volume-Set gehörenden Platten wird die VSN geändert.
4. Im OLC (Online Catalog auf der Volres) wird die VSN aller zum Control-Volume-Set gehörenden Platten geändert.

5. Im SVL der Hauptplatte aller zum entsprechenden SM-Pubset gehörenden Volume-Sets wird die Katalogkennung des Control-Volume-Sets geändert.
6. Die MRSCAT-Einträge aller zum entsprechenden SM-Pubset gehörenden Volume-Sets werden gelöscht. Sie werden beim Importieren des SM-Pubsets wieder neu angelegt
7. Für den entsprechenden SM-Pubset wird der MRSCAT-Eintrag gelöscht und mit der neuen Katalogkennung des Control-Volume-Sets neu angelegt.
8. Eine Längenprüfung der Objektnamen ist nicht notwendig, da die Katalogkennung des Control-Volume-Sets nicht in den Dateinamen eingeht.
9. Die Katalogkennung des Control-Volume-Sets kann nicht als Standardkatalogkennung verwendet werden, deshalb ist eine Änderung der Standardkatalogkennung nicht notwendig. Nach dem Umbenennen wird der zugehörige SM-Pubset nicht importiert. Diese Aktion muss vom Anwender durchgeführt werden.

Erzeugen eines SF-Pubsets

1. Falls die Spiegel-Platten des Pubsets nicht bereits getrennt sind, werden sie getrennt und direkt mit der neuen VSN erzeugt. Dazu muss der Original-Pubset exportiert sein.
   Falls die Spiegel-Platten des Pubsets bereits getrennt und in Doppelpunkt-Notation benannt sind, wird in ihrem SVL die VSN von der Doppelpunkt-Notation auf die entsprechende VSN des neu zu erzeugenden Pubsets geändert.
2. Im OLC (Online Catalog auf der Pubres) des Spiegel-Pubsets wird die VSN aller zum Pubset gehörenden Platten geändert.
3. In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen wird die VSN geändert.
4. Die Standard-Katalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubsets und des neu erzeugten Pubsets wird geändert. Diese Aktion wird im Dialogbetrieb nur nach Rückfrage und im Batchbetrieb standardmäßig durchgeführt. Dazu wird der SF-Pubset importiert. Die Änderung der Standardkatalogkennung erfolgt mit `/MODIFY-USER-ATTRIBUTES`.
5. Der Anwender kann entscheiden, ob der Pubset verfügbar werden bzw. bleiben soll.

Erzeugen eines SM-Pubsets

1. Falls die Spiegel-Platten des Pubsets nicht bereits getrennt sind, werden sie getrennt bzw. aktiviert. Dazu muss der Original-Pubset exportiert sein.
2. Für den SM-Pubset wird ein neuer MRSCAT-Eintrag mit der neuen Katalogkennung angelegt.
3. Im SVL der Volres des Control-Volume-Sets wird die Katalogkennung des SM-Pubsets geändert. Im SVL aller zum Control-Volume-Set gehörenden Platten wird die VSN geändert.
4. Im OLC (Online Catalog auf der Volres) des Control-Volume-Sets wird die VSN aller zum Control-Volume-Set gehörenden Platten geändert.
5. In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen des Control-Volume-Sets wird die VSN geändert.
6. Die Katalogkennungen des SM-Pubsets, des Control-Volume-Sets und aller weiteren Volume-Sets werden in der Pubset-Konfigurationsdatei auf dem Control-Volume-Set geändert.
7. Nach Anpassung des Control-Volume-Sets werden für alle weiteren Volume-Sets folgende Anpassungen durchgeführt:

- Im SVL der Volres des Volume-Sets wird die Katalogkennung des SM-Pubsets und die Katalogkennung des Control-Volume-Sets geändert. Im SVL aller zum Volume-Set gehörenden Platten wird die VSN, wie in der ensprechenden Anweisung SET-NAME-OF-NEW-VOLUME-SET angegeben, geändert.
- Im OLC (Online Catalog auf der Volres) des Control-Volume-Sets wird die VSN aller zum Volume-Set gehörenden Platten geändert.
- In der Datenträgerliste des Katalogeintrages von Dateien und Dateigenerationsgruppen des Volume-Sets wird die VSN geändert.

8. Die Standard-Katalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubsets und des umbenannten Pubsets wird geändert. Diese Aktion wird im Dialogbetrieb nur nach Rückfrage und im Batchbetrieb standardmäßig durchgeführt. Dazu wird der SM-Pubset importiert. Die Änderung der Standardkatalogkennung erfolgt mit `/MODIFY-USER-ATTRIBUTES`.
9. Der Anwender kann entscheiden, ob der Pubset verfügbar werden bzw. bleiben soll.

11.3 Einschränkungen und Nacharbeiten

Dateien auf Net-Storage beim Umbenennen von SF- oder SM-Pubsets

Wenn es in einem Pubset Dateien gibt, die auf einem Standard-Net-Storage-Volume liegen, dann werden die zugehörigen Metadaten der Datei im lokalen Pubset und auf dem Net-Storage-Volume an den neuen Pubset-Namen angepasst.

Wenn es in einem Pubset Dateien gibt, die auf einem Net-Storage-Volume mit einer benutzer-definierten VSN liegen, dann werden die zugehörigen Metadaten der Datei auf dem Net-Storage-Volume an den neuen Pubset-Namen angepasst.

In beiden Fällen kann nach der Umbenennung ohne Einschränkung auf die Net-Storage-Dateien zugegriffen werden.

Die Verwaltung von Net-Storage ist im Handbuch „Systembetreuung“ [5] beschrieben.

Dateien auf Net-Storage beim Erzeugen von Spiegel-Pubsets

Wenn es in einem abgetrennten Pubset Dateien gibt, die auf einem Standard-Net-Storage-Volume liegen, dann werden die zugehörigen Metadaten der Datei nur im lokalen Pubset an den neuen Pubset-Namen angepasst.

Wenn es in einem Pubset Dateien gibt, die auf einem Net-Storage-Volume mit einer benutzer-definierten VSN liegen, dann werden die zugehörigen Metadaten der Datei nicht angepasst.

In beiden Fällen kann nach der Abtrennung auf die Net-Storage-Dateien nicht zugegriffen werden. Gegebenfalls müssen die Metadaten aus dem abgetrennten Pubset entfernt werden.

Ändern der Standardkatalogkennung

Die Standardkatalogkennung kann nur dann geändert werden, wenn die Benutzerkennung das Systemprivileg USER-ADMINISTRATION hat.

Besitzt die Benutzerkennung dieses Privileg nicht, wird über die Meldung `PVR0200` entschieden, ob die Umbenennung trotzdem durchgeführt werden soll. In diesem Fall kann die Standardkatalogkennung nachträglich mit der Anweisung MODIFY-JOINFILE geändert werden, sobald die Benutzerkennung das Privileg besitzt.

Maximale Anzahl von Volumes

Eine Umbenennung von Punkt- nach PUB-Notation kann nur dann durchgeführt werden, wenn die Anzahl der Volumes nicht größer als 32 ist.

Eine Umbenennung von Punkt- nach Punktnotation ist nicht möglich, wenn die neue Katalogkennung vierstellig und die Anzahl der Volumes größer als 36 ist, da nur eine Stelle für die Folgenummer zur Verfügung steht.

Dateischutz mit GUARDS

Bei einem Guard kann man auch das Programm definieren, mit dem der Zugriff auf die Datei erfolgen darf (z.B. $TSOS.EDT). Dabei wird im GUARDS-Katalog auch die Katalogkennung gespeichert.

Wenn der Pubset umbenannt wird, auf dem das Programm liegt, ist die Zugriffsbedingung nicht mehr erfüllt: die Katalogkennung im GUARDS-Katalog und die Katalogkennung des Pubsets, auf dem das Programm jetzt liegt, stimmen nicht mehr überein.

In diesem Fall müssen die Zugriffsbedingungen für alle Guards, die diese Katalogkennung enthalten, geändert werden. Betroffen sind nicht nur die Guards auf dem umbenannten und dem Home-Pubset, sondern auch die Guards auf allen anderen Pubsets.

PVSREN bietet für dieses Problem keine Unterstützung.

Katalogkennung in Benutzerdateien

PVSREN ändert nicht die Katalogkennung in Benutzerdateien.

Enthält eine Benutzerdatei die Katalogkennung eines Pubsets, der umbenannt wird (z.B. in Prozeduren), muss diese Katalogkennung vom Dateieigentümer geändert werden, falls dies notwendig ist.

Katalogkennung in IMON-Dateien

Nach einer Software-Installation mit dem Installations-Monitor (IMON) wird der Ablageort der Installations-Items im Software-Configuration-Inventory (SCI) vermerkt. Dabei wird auch die Katalogkennung gespeichert.

Wird ein Pubset umbenannt, auf dem Software mit IMON installiert wurde, müssen alle SCIs geändert werden, in denen diese Software registriert ist. Das SCI des HOME- und des umbenannten Pubsets ändert PVSREN. Das SCI weiterer Pubsets muss mit `/MODIFY-IMON-SCI` manuell geändert werden. Dabei muss für OLD-NAME der Name des Pubsets vor der Umbenennung und bei NEW-NAME der Name des Pubsets nach der Umbenennung angegeben werden.

Shared-Pubset

PVSREN kann die MRSCAT-Einträge nur auf dem System ändern, auf dem die Umbenennung durchgeführt wird.

Da es bei einem Shared-Pubset auf allen Systemen, die diesen Pubset verwenden, einen MRSCAT-Eintrag für diesen Pubset gibt, müssen die MRSCAT-Einträge auf allen Remote-Systemen von der Systemverwaltung angepasst werden.

- Bei Pubsets muss der alte MRSCAT-Eintrag gelöscht und ein MRSCAT-Eintrag mit der neuen Katalogkennung angelegt werden. Für SM-Pubsets ist dabei der (ggf. ebenfalls neue) Name des Control-Volume-Sets anzugeben.

  Die MRSCAT-Einträge für die Volume-Sets eines SM-Pubsets müssen zusammen mit dem Pubset-Eintrag gelöscht werden (voreingestellt beim Kommando REMOVE-MASTER-CATALOG-ENTRY). Sie werden automatisch neu erzeugt. Siehe auch den Hinweis unten.
- Beim Umbenennen des Control-Volume-Sets muss der MRSCAT-Eintrag des Control-Volume-Sets gelöscht werden. Der Eintrag für den neuen Control-Volume-Set wird beim Importieren des SM-Pubsets automatisch neu angelegt. Siehe auch den Hinweis unten.

  Im MRSCAT-Eintrag des SM-Pubsets, zu dem der Control-Volume-Set gehört, muss zusätzlich mit /MODIFY-MASTER-CATALOG-ENTRY die neue Katalogkennung des Control-Volume-Sets eingetragen werden.
- Beim Umbenennen eines „normalen“ Volume-Sets muss nur der MRSCAT-Eintrag dieses Volume-Sets gelöscht werden. Er muss nicht neu angelegt werden, da dies beim Importieren des SM-Pubsets erfolgt. Siehe auch den folgenden Hinweis.

> **i** Ein Volume-Set kann dann nicht automatisch in den MRSCAT eingetragen werden, wenn dort bereits ein gleichnamiger SF-Pubset oder ein gleichnamiger Volume-Set eines anderen SM-Pubsets eingetragen ist. In diesem Fall muss der vorhandene Eintrag zuvor manuell aus dem MRSCAT gelöscht werden.

Storage-Klassen

Auf allen SM-Pubsets existieren Kataloge für Storage-Klassen und Volume-Set-Listen. Storage-Klassen haben keine direkten Bezüge zu Katalogkennungen. Volume-Set-Listen hingegen ordnen einer Storageklasse eine oder mehrere Voume-Sets zu. Die Einträge des Katalogs der Volume-Set-Listen enthalten daher ein oder mehrere Katalogkennungen von Volume-Sets des SM-Pubset.

Bei der Umbenennung eines Volume-Sets wird diese Umbennung auch im Volume-Set-Katalog nachgezogen, damit die Allokierung von Speicherplatz für Dateien über Storage-Klassen (explizit angegebene oder Default Storage-Klassen im Benutzerkatalog) auf dem Pubset durch die Umbenennung nicht beeinflusst wird.

Beim Erzeugen eines Pubsets aus einem Spiegel-Pubset können die Storage-Klassen des Ausgangs-Pubset wahlweise übernommen oder gelöscht werden. Dazu wird die beantwortbare Meldung `PVR0207` ausgegeben. Bei Übernahme der Storage-Klassen werden die Volume-Set-Listen (wie bei der Umbenennung) an die Katalogkennungen der Volume-Sets des Ziel-Pubsets angepasst.

Startup-Parameterdatei

Beim Umbenennen eines Home- oder Paging-Pubsets muss ggf. die Startup-Parameterdatei angepasst werden. Sie kann neben den VSNs der Paging-Platten vollständige Dateinamen mit Katalogkennung enthalten.

SDF-Parameterdatei

Mit `/MODIFY-SDF-PARAMETERS` kann eine Syntaxdatei aktiviert und ihr Dateiname mit der Katalogkennung angegeben werden.

Mit `/SHOW-SDF-PARAMETERS` kann festgestellt werden, ob sich eine Syntaxdatei auf dem umbenannten Pubset befindet. Ist das der Fall, muss diese mit `/MODIFY-SDF-PARAMETERS` deaktiviert und mit dem neuen Dateinamen wieder aktiviert werden.

MIP-Parameterdatei

Mit `/MODIFY-MIP-PARAMETERS` kann eine Meldungsdatei aktiviert und ihr Dateiname mit der Katalogkennung angegeben werden.

Mit `/SHOW-MIP-PARAMETERS` kann festgestellt werden, ob sich eine Meldungsdatei auf dem umbenannten Pubset befindet. Ist das der Fall, muss diese mit `/MODIFY-MIP-PARAMETERS` deaktiviert und mit dem neuen Dateinamen wieder aktiviert werden.

SLED-Parameterdatei

Enthält der umbenannte Pubset eine SLEDFILE, die in der SLED-Parameterdatei referenziert wird, dann muss diese angepasst werden.

REPEAT-Jobs

REPEAT-Jobs, bei denen sich die Enterdatei auf dem umbenannten Pubset befindet, müssen neu gestartet werden.

Druckaufträge

Druckaufträge, die sich auf Dateien auf dem umbenannten Pubset beziehen, können nach der Umbenennung nicht mehr ausgeführt werden. Sie müssen gelöscht und neu gestartet werden.

Migrierte Dateien auf einem Pubset

Wenn ein SF-Pubset oder ein SM-Pubset umbenannt wird, auf dem migrierte Dateien vorhanden sind, muss das ARCHIVE-Directory mit dem Tool DIRCONV bearbeitet werden. PVSREN gibt keine Meldungen aus, die darauf hinweisen, dass auf dem Pubset migrierte Dateien vorhanden sind.

Beispiel

```
/START-DIRCONV
//RENAME-CATID DIRECTORY-NAME = <directory name>
               OLD-CATID = *BY-SPECIFICATION(CATID = <old catid>),
               NEW-CATID = <new catid>,
               NEW-DIRECTORY-NAME = <new directory name>
//END
```

Danach muss das neu erzeugte Directory mit dem alten Directory vertauscht werden.

Wenn ein HSMS-CONTROLLED-Volume-Set (S1-Ebene) umbenannt wird, muss der S1-Volume-Set mit der HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS geändert werden.

Fehlerbehandlung beim Erzeugen von Pubsets aus Spiegel-Pubsets

Wird von der PVSREN-Verarbeitung während Erzeugens eines Pubsets aus einem Spiegel-Pubset ein Fehler erkannt, so werden alle bereits durchgeführten Schritte zurückgesetzt.

Im Einzelnen:

- Für Snap- und Clone-Units wird die Sychronisation nicht wieder automatisch aufgenommen. Die Volumes behalten ihren aktuellen Zustand bei.
- Wenn PVSREN eine Snap-Session erzeugt hat, so wird diese wieder abgebaut.

Bei einer Systembeendigung während des Erzeugens eines Pubset können sich die Volumes des neu zu erzeugenden Pubsets in einem inkonsistenten Zustand befinden.

Zum Wiederaufsetzen ist es erforderlich,

- Clone- bzw. Snap-Sessions mit den SHC-OSD-Kommandos `/START-CLONE-SESSION` bzw. `/START-SNAP-SESSION` neu zu starten.

Falls Pubsets aus bereits abgetrennten Clone-Units oder Snap-Units erzeugt werden sollen, ist der Ausgangszustand auf diese Weise nicht mehr rekonstruierbar. In diesem Fall sollte vor dem Erzeugen der neuen Pubsets eine Sicherung des Stands der Clone-Units oder Snap-Units durchgeführt werden.

Eine Restart-Funktion, wie beim Umbenennen von Pubsets, wird nicht angeboten.

Anpassung der Speicherebenen S1 und S2 beim Erzeugen von Pubsets aus Spiegel-Pubsets

Die Backup-Archive für den Pubset werden neu initialisiert, da es sich um einen neuen Pubset handelt.

Migrationsarchive werden von PVSREN an die neue, erweiterte Umgebung angepasst. Für SF-Pubsets existiert entweder ein globales Migrationsarchiv oder, bei der empfohlenen „dezentralen“ Organisation, ein pubsetspezifisches Migrationsarchiv.
SM-Pubsets haben stets eigene Migrationsarchive, deren Directories im SM-Pubset liegen, d.h. bei Migration nach S1 liegt auch das entsprechende Migrationsarchiv im Pubset.

Trotz der Unterschiede zwischen SF- und SM-Pubsets kann die Anpassung der Migrationsarchive nach einer Pubset-Erzeugung aus Spiegel-Pubsets im Wesentlichen auf die gleiche Weise erfolgen.

Die Anpassung erfolgt in folgenden Schritten:

1. Das Migrationsdirectory für den Ausgangs-Pubset wird als Datei kopiert. In dieser Datei wird mit DIRCONV die Katalogkennung des Ausgangspubset in die Katalogkennung des neuen Pubset umbenannt. (Es wird vorausgesetzt, dass diese Katalogkennung nicht im Directory vorhanden ist.)
2. Als vorübergehende Hilfskonstruktion wird das kopierte Directory mit der umbenannten Katalogkennung als Directory des Archivs SYSBACKUP eingesetzt.
3. Für den neuen Pubset wird nun, falls es sich um einen SM-Pubset oder um einen SF-Pubset mit dezentraler Organisation handelt, ein neues Migrationsarchiv definiert. An dieser Stelle sollten auch nicht mehr benötigte, migrierte Dateien gelöscht werden.
4. Für den neuen Pubset wird die HSMS-Anweisung REPAIR-CATALOG-BY-RESTORE aufgerufen. Dort wird für alle migrierten Dateien geprüft, ob der Verweis im Katalog gültig ist; dies ist hier aber nicht der Fall. Dann wird SYSBACKUP ins Migrationsarchiv kopiert und der Verweis im Katalogeintrag wird aktualisiert.
5. Das als Hilfskonstruktion genutzte Directory (siehe Schritt 2) wird gelöscht. Seine Nutzung als SYSBACKUP wird beendet.

11.4 Starten und Beenden von PVSREN

Das Programm PVSREN wird unter der Kennung TSOS gestartet mit: `/START-PVSREN.`

START-PVSREN	Alias: **PVSREN**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Mit der Anweisung MODIFY-LOGGING-OPTIONS können Umfang und Ausgabeziel der Protokollierung festgelegt werden (siehe "MODIFY-LOGGING-OPTIONS - Ändern voreingestellter Protokollierungswerte").

Ein Start über `/START-EXECUTABLE-PROGRAM FROM-FILE=PVSREN` wird nur noch aus Kompatibilitätsgründen unterstützt.

Mit der Anweisung **END** wird PVSREN beendet.

Format

END

Diese Anweisung besitzt keine Operanden.

11.5 Meldungen von PVSREN

Die Meldungen des Dienstprogramms PVSREN haben die Meldungsklasse PVR.

Ihr Ausgabeziel wird mit der Anweisung MODIFY-LOGGING-OPTIONS festgelegt. Standardmäßig werden sie nach SYSOUT ausgegeben.

Siehe auch den Abschnitt „Meldungen und ihre Bedeutung“ (Darstellungsmittel).

11.6 Anweisungen von PVSREN

- Übersicht über die Anweisungen von PVSREN
- Beschreibung der Anweisungen
  - CHECK-FILENAME-LENGTH - Prüfen der Länge von Datei- und Jobvariablennamen
  - CREATE-PUBSET-FROM-MIRROR - Neuen Pubset aus Spiegel-Platten eines Pubset erzeugen
  - MODIFY-JOINFILE - Ändern der Standardkatalogkennung im Benutzerkatalog
  - MODIFY-LOGGING-OPTIONS - Ändern voreingestellter Protokollierungswerte
  - RENAME-PUBSET-OR-VOLUME-SET - Konvertieren der Pubset-Notation oder Umbenennen eines SF- oder SM-Pubsets oder Volume-Sets
  - RESTART-RENAMING - Neustart der Konvertierung oder Umbenennung
  - RESTORE-LABELS-OF-PUBSET - Pubsets aus Spiegel-Pubsets mit Doppelpunkt-Notation rückbenennen
  - SET-NAME-OF-NEW-VOLUME-SET - Volume-Set-Name beim Erzeugen eines SM-Pubsets aus Spiegel-Platten festlegen
  - SHOW-LOGGING-OPTIONS - Anzeigen der gültigen Werte für die Protokollierung

11.6.1 Übersicht über die Anweisungen von PVSREN

Anweisung	Bedeutung
CHECK-FILENAME-LENGTH	Prüfen der Länge von Datei-, Dateigenerationsgruppen- und Jobvariablennamen im Katalog eines Pubsets
CREATE-PUBSET-FROM-MIRROR	Neuen Pubset aus den Spiegel-Platten eines Pubsets erzeugen
MODIFY-JOINFILE	Ändern der voreingestellten Katalogkennung im Benutzerkatalog eines Pubsets
MODIFY-LOGGING-OPTIONS	Ändern der voreingestellten, für alle Funktionen des Programms gültigen Protokollierungswerte
RENAME-PUBSET-OR-VOLUME-SET	Konvertieren der Pubset-Notation oder Umbenennen eines SF- oder SM-Pubsets oder Volume-Sets
RESTART-RENAMING	Neustart einer abgebrochenen Konvertierung oder Umbenennung
RESTORE-LABELS-OF-PUBSET	Rückbenennen von Spiegel-Pubsets mit VSNs in Doppelpunkt-Notation auf ihre ursprüngliche VSN
SET-NAME-OF-NEW-VOLUME-SET	Festlegen des neuen Namens eines Volume-Sets beim Erzeugen eines SM-Pubsets aus Spiegel-Platten
SHOW-LOGGING-OPTIONS	Auflisten der gültigen Werte für die Protokollierung

Außerdem unterstützt PVSREN die Ausführung der SDF-Standardanweisungen (siehe Handbuch „Dialogschnittstelle SDF“ [20]).

11.6.2 Beschreibung der Anweisungen

- CHECK-FILENAME-LENGTH - Prüfen der Länge von Datei- und Jobvariablennamen
- CREATE-PUBSET-FROM-MIRROR - Neuen Pubset aus Spiegel-Platten eines Pubset erzeugen
- MODIFY-JOINFILE - Ändern der Standardkatalogkennung im Benutzerkatalog
- MODIFY-LOGGING-OPTIONS - Ändern voreingestellter Protokollierungswerte
- RENAME-PUBSET-OR-VOLUME-SET - Konvertieren der Pubset-Notation oder Umbenennen eines SF- oder SM-Pubsets oder Volume-Sets
- RESTART-RENAMING - Neustart der Konvertierung oder Umbenennung
- RESTORE-LABELS-OF-PUBSET - Pubsets aus Spiegel-Pubsets mit Doppelpunkt-Notation rückbenennen
- SET-NAME-OF-NEW-VOLUME-SET - Volume-Set-Name beim Erzeugen eines SM-Pubsets aus Spiegel-Platten festlegen
- SHOW-LOGGING-OPTIONS - Anzeigen der gültigen Werte für die Protokollierung

11.6.2.1 CHECK-FILENAME-LENGTH - Prüfen der Länge von Datei- und Jobvariablennamen

Mit der Anweisung CHECK-FILENAME-LENGTH wird die Prüfung der Länge aller Datei-, Dateigenerationsgruppen- und Jobvariablennamen im Katalog des Pubsets veranlasst. Die Namen dürfen eine bestimmte Maximallänge (siehe unten) nicht überschreiten.

Diese Funktion ist nur sinnvoll, wenn eine Umbenennung gewünscht wird, bei der sich die Länge der Katalogkennung (catid) vergrößert. Die Anweisung wird nur ausgeführt, wenn der angegebene Pubset mit `/EXPORT-PUBSET` auf „inaccessible“ gesetzt wurde.

Falls erforderlich, sollte die Längenprüfung immer vor der Umbenennung durchgeführt werden. Erkennt das Programm bei der Prüfung Namen, deren Länge die maximal zulässige Länge überschreiten, werden diese protokolliert und PVSREN läuft auf die nächste STEP-Anweisung. Die betroffenen Datei- bzw. Jobvariablennamen müssen mit `/MODIFY-FILE-ATTRIBUTES` bzw. `/MODIFY-JV-ATTRIBUTES` der zulässigen Länge angepasst werden.

Das Programm prüft auch, ob die Systemkennung (sysid) im SVL der SF-Pubres initialisiert ist. Ist das nicht der Fall, wird eine entsprechende Meldung ausgegeben.

Da SM-Pubsets und Volume-Sets nicht als Home-Pubset eingesetzt werden können, hat die Systemkennung bei ihnen keine Bedeutung und wird auch nicht geprüft.

Da die Volume-Set-Kennung nicht in den Dateinamen eingeht, wird die Anweisung CHECK-FILENAME-LENGTH für Volume-Sets abgewiesen.

Maximallänge der Datei- und Jobvariablennamen

Die maximale Länge für einen Pfadnamen (mit Katalog- und Benutzerkennung) beträgt bei Dateien und Jobvariablen 54 Zeichen, bei Dateigenerationsgruppen 47 Zeichen.

Aufbau des Pfadnamens:

`:catid:$userid.<filename / fgg-name / jv-name>`

Die maximale Länge des eigentlichen Namens (Datei-, Dateigenerationsgruppen- bzw. Jobvariablen-Name) errechnet sich wie folgt:

```
max. Pfadnamenlänge (54 oder 47)
   minus Anzahl der Trennzeichen (immer 4 (::$.))
   minus Länge der Katalogkennung (1..4)
   minus Länge der Benutzerkennung (1..8)
```

Format

CHECK-FILENAME-LENGTH
CATID = <catid 1..4> ,NEW-CATID = <catid 1...4>

Operanden

CATID = <catid 1..4>
Die vorliegende Katalogkennung.

NEW-CATID = <catid 1..4>
Die gewünschte Katalogkennung.

Beispiel

```
(IN)  /CHECK-FILENAME-LENGTH CATID=A,NEW-CATID=ABC                    1.
(OUT) %PVR0134 The following object names exceed the maximum length    2.
(NL)  :<catid>:<userid>.<jv-name>    (JV)                             3.
(NL) *:<catid>:<userid>.<filename> (FGG)                              4.
(NL)  :<catid>:<userid>.<filename> (FGG)                              5.
(NL) *:<catid>:<userid>.<filename>                                    6.
(NL)  :<catid>:<userid>.<filename>                                    7.
```

1. Die neue Katalogkennung hat 3 Zeichen, daraus ergibt sich (bei einer Benutzerkennung von 8 Zeichen) eine Maximallänge für Datei- und Jobvariablennamen von 39 Zeichen, für Dateigenerationsgruppen von 32 Zeichen.
2. Wenn Namen erkannt werden, deren Länge die Maximallänge überschreiten, wird diese Meldung ausgegeben.
3. Der ausgegebene Name repräsentiert ... eine Jobvariable.
4. ... eine private Dateigenerationsgruppe. (* = privat)
5. ... eine gemeinschaftliche Dateigenerationsgruppe.
6. ... eine private Datei. (* = privat)
7. ... eine gemeinschaftliche Datei.

11.6.2.2 CREATE-PUBSET-FROM-MIRROR - Neuen Pubset aus Spiegel-Platten eines Pubset erzeugen

Die Anweisung CREATE-PUBSET-FROM-MIRROR erzeugt aus den Spiegel-Platten (Clone-Units, Snap-Units) eines SF- oder SM-Pubsets einen neuen SF-Pubset mit neuer Katalogkennung bzw. bei SM-Pubsets entsprechend neue Volume-Sets mit neuen Katalogkennungen.

Mit dem Operanden MIRRORING-METHOD = *SPLIT-MIRROR werden die Spiegelplatten eines Pubsets, die bereits vor dem PVSREN-Aufruf abgetrennt wurden, in einen freien Pubset mit neuem Namen umbenannt. Die abgetrennten Spiegelplatten genügen dabei der Doppelpunkt-Notation.

Mit dem Operanden MIRRORING-METHOD = *SNAP(...) wird eine neue Snap-Session zu einem Pubset eingerichtet. Alle Snap-Units dieser Snap-Session werden aktiviert und in einen freien Pubset mit neuem Namen umbenannt. Dabei ist es erlaubt, dass parallel ein oder mehrere Snapsets zu dem Pubset vorhanden sind.

Mit dem Operanden MIRRORING-METHOD = *CLONE(...) werden alle Clone-Units einer bereits eingerichteten Clone-Session zu einem Pubset aktiviert und in einen freien Pubset unter neuem Namen umbenannt.

Voraussetzungen

- Wenn PVSREN die Spiegel-Pubsets abtrennt bzw. aktiviert, muss der Pubset exportiert sein. Nur so kann PVSREN die Abtrennung konsistent durchführen.
- Die VSNs bereits abgetrennter Spiegel-Pubsets müssen der Doppelpunkt-Notation entsprechen.
- Die neue Katalogkennung muss die gleiche Länge wie die Katalogkennung des Ausgangs-Pubsets haben.
- Die Katalogkennungen der neuen Pubsets bzw. Volume-Sets dürfen nicht bereits im System vorhanden sein.
- Bei SM-Pubsets ist für alle Volume-Sets eine Umbenennungsvorschrift anzugeben. Diese Umbennungsvorschrift muss in einer Datei abgelegt sein. Dabei ist zu beachten, dass die Länge der neuen Katalogkennungen jeweils so zu wählen ist, dass alle Volumes des jeweiligen Volume-Sets mit der entsprechenden VSN benannt werden können: In PUB-Notation können max. 32 Volumes, in Punktnotation max. 255 Volumes (bei 2- und 3-stelligen Katalogkennungen) bzw. max. 36 Volumes (bei 4-stelligen Katalogkennungen) adressiert werden. Die Umbenennung erfolgt mit der Anweisung SET-NAME-OF-NEW-VOLUME-SET pro Volume-Set (siehe "SET-NAME-OF-NEW-VOLUME-SET - Volume-Set-Name beim Erzeugen eines SM-Pubsets aus Spiegel- Platten festlegen").

Siehe auch Abschnitt „Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets“.

Für eine ausführliche Beschreibung des Ablaufs einschließlich aller evtl. notwendiger Nacharbeiten siehe "Ablauf von PVSREN" und "Einschränkungen und Nacharbeiten".

Format

```
CREATE-PUBSET-FROM-MIRROR

 SOURCE-PUBSET = <catid 1..4>

 ,NEW-PUBSET = <catid 1..4>

 ,PUBSET-TYPE = *SF / *SM(...)

      *SM(...)

           |     CONTROL-VOLUME-SET = <catid 1..4>

           |     ,NEW-VOLUME-SET-NAMES = *BY-FILE(...)

           |          *BY-FILE(...)

           |               |     FILE-NAME = <filename 1..54>

 ,MIRRORING-METHOD = *CLONE / *SPLIT-MIRROR / *SNAP(...)









      *SNAP(...)

           |     SELECT = *STD / *TARGET-UNIT



      *CLONE(...)

           |     SELECT = *STD / *TARGET-UNIT
```

Operanden

SOURCE-PUBSET = <catid 1..4>
Gibt die Katalogkennung des Pubsets an, aus dessen Spiegel-Platten ein neuer Pubset erzeugt werden soll.

NEW-PUBSET = <catid 1..4>
Gibt die Katalogkennung des neuen Pubsets an.

PUBSET-TYPE =
Gibt an, um welchen Pubset-Typ es sich handelt.

PUBSET-TYPE = *SF

Es handelt sich um einen SF-Pubset.

PUBSET-TYPE = *SM(...)
Es handelt sich um einen SM-Pubset.

CONTROL-VOLUME-SET = <catid 1..4>
Gibt die Katalogkennung des Control-Volume-Sets des SM-Pubsets an, aus dessen Spiegel-Pubset ein neuer Pubset erzeugt werden soll.

NEW-VOLUME-SET-NAMES = *BY-FILE(...)
Ordnet den Katalogkennungen der Volume-Sets des Ausgang-Pubsets neue Katalogkennungen für den zu erzeugenden SM-Pubset zu. Die Vorschrift für die Umsetzung der Katalogkennungen der Volume-Sets ist in einer Datei abgelegt.

FILE-NAME = <filename 1..54>
Name der Datei, in der die Umbenennungsvorschrift für die Volume-Sets des neu zu erzeugenden SM-Pubsets abgelegt ist.

MIRRORING-METHOD =
Spezifiziert die Art der Spiegel-Platten. Außerdem wird festgelegt, ob und ggf. wie PVSREN die Spiegel-Platten abtrennt.

MIRRORING-METHOD = *SPLIT-MIRROR
Die Spiegel-Platten sind bereits abgetrennte Clone- oder Snap-Units. Ihre VSNs müssen der Doppelpunkt-Notation genügen.

MIRRORING-METHOD = *SNAP(...)

Die Spiegel-Platten sind Snap-Units, die durch SHC-OSD administriert werden.

SELECT =

Gibt an, welche Gruppe von Snap-Units einer Remote-Spiegel-Konfiguration mit Source- und Target-Units abgetrennt werden soll. Falls keine Remote-Spiegelung erfolgt, darf dieser Parameter nur den Wert *STD haben.

SELECT = *STD

Die Snap-Units auf der Source-Seite einer Remote-Spiegel-Konfiguration werden abgetrennt.

SELECT = *TARGET-UNIT

Die Snap-Units auf der Target-Seite einer Remote-Spiegel-Konfiguration werden abgetrennt.

MIRRORING-METHOD = *CLONE(...)

Die Spiegel-Platten sind Clone-Units, die durch SHC-OSD administriert werden.

SELECT =

Gibt an, welche Gruppe von Clone-Units einer Remote-Spiegel-Konfiguration mit Source- und Target-Units abgetrennt werden soll. Falls keine Remote-Spiegelung erfolgt, darf dieser Parameter nur den Wert *STD haben.

SELECT = *STD

Die Clone-Units auf der Source-Seite einer Remote-Spiegel-Konfiguration werden abgetrennt.

SELECT = *TARGET-UNIT

Die Clone-Units auf der Target-Seite einer Remote-Spiegel-Konfiguration werden abgetrennt.

11.6.2.3 MODIFY-JOINFILE - Ändern der Standardkatalogkennung im Benutzerkatalog

Mit der Anweisung MODIFY-JOINFILE kann für alle Benutzerkennungen die voreingestellte Katalogkennung (Standardkatalogkennung) im Benutzerkatalog eines SF-Pubsets oder SM-Pubsets geändert werden. Eine Ausnahme ist dabei zu beachten: Für die Kennung TSOS kann eine Änderung nur dann erfolgen, wenn die Operandenwerte bei PUBSET und NEW-DEFAULT-CATID gleich sind.

Für die Ausführung dieser Anweisung muss der Pubset importiert sein.

Format

MODIFY-JOINFILE
PUBSET = <catid 1..4> ,DEFAULT-CATID = <catid 1..4> ,NEW-DEFAULT-CATID = <catid 1..4>

Operanden

PUBSET = <catid 1..4>
Gibt die Katalogkennung des Pubsets an, auf dem sich der zu ändernde Benutzerkatalog befindet.

DEFAULT-CATID = <catid 1..4>
Gibt die voreingestellte Katalogkennung in dem Benutzerkatalog an, der geändert werden soll.

NEW-DEFAULT-CATID = <catid 1..4>
Gibt die Katalogkennung an, die die alte ersetzt und nun als neue Standardkatalogkennung gelten soll.

11.6.2.4 MODIFY-LOGGING-OPTIONS - Ändern voreingestellter Protokollierungswerte

Mit der Anweisung MODIFY-LOGGING-OPTIONS können voreingestellte Protokollierungswerte für PVSREN global für alle Funktionen des Programms geändert werden.

Will sich der Benutzer über die aktuell gültigen Werte informieren, muss er diese mit der Anweisung SHOW-LOGGING-OPTIONS abfragen.

Format

MODIFY-LOGGING-OPTIONS
INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM ,OUTPUT = *UNCHANGED / list-poss(2): *SYSOUT / *SYSLST

Operanden

INFORMATION =

Steuert den Umfang des von PVSREN erzeugten Protokolls.

INFORMATION = *UNCHANGED
Der voreingestellte Protokollierungswert behält unverändert seine Gültigkeit. Voreingestellter Wert ist INFORMATION=*MEDIUM.

INFORMATION = *MEDIUM
Außer Fehlermeldungen sollen noch Erfolgsmeldungen protokolliert werden, die für den Benutzer von entscheidender Bedeutung sind.

INFORMATION = *MINIMUM
Nur Fehlermeldungen werden protokolliert.

INFORMATION = *MAXIMUM
Alle Meldungen von PVSREN werden protokolliert.

OUTPUT =
Definiert das Ausgabemedium für die von PVSREN erzeugten Protokolle.

OUTPUT = *UNCHANGED
Der voreingestellte Wert für das Ausgabemedium soll unverändert gültig bleiben. Voreingestellter Wert ist OUTPUT=*SYSOUT.

OUTPUT = *SYSOUT
Die Ausgabe soll im Dialog über die Datensichtstation, im Batch-Betrieb in die Systemdatei SYSOUT erfolgen.

OUTPUT = *SYSLST
Ausgabemedium für die Protokolle soll die Systemdatei SYSLST sein.

Werden SYSOUT und SYSLST als Liste angegeben, so werden die Protokolle in die Systemdateien SYSLST und SYSOUT (im Dialog über die Datensichtstation) ausgegeben.

11.6.2.5 RENAME-PUBSET-OR-VOLUME-SET - Konvertieren der Pubset-Notation oder Umbenennen eines SF- oder SM-Pubsets oder Volume-Sets

Volume-Sets) umbenannt oder ihre Pubset-Notation konvertiert werden.
Die Anweisung wird nur ausgeführt, wenn der angegebene Pubset oder Volume-Set den Status INACCESSIBLE hat.

Die durchgeführten Aktionen sind vom jeweiligen Modus abhängig.
Bei MODE=*NORMAL-RENAME werden nacheinander durch RENAME-PUBSET-OR- VOLUME-SET folgende Schritte ausgeführt:

1. Prüfung der Namenslängen von Dateien, Dateigenerationsgruppen und Jobvariablen (nicht bei Volume-Sets). Wird die Maximallänge überschritten, so wird die Anweisung abgebrochen. Es sollte daher immer zuerst die Anweisung CHECK-FILENAME-LENGTH aufgerufen werden.
2. Ändern der Katalogkennung des Pubsets oder Volume-Sets
   - im Dateikatalog TSOSCAT (nicht bei SM-Pubsets)
   - im Pfadnamen von Systemdateien (Paging-Bereich, TSOSCAT.<catid>)
   - im SVL aller zum SF-Pubset oder Volume-Set gehörenden Platten
   - in der Pubset-Konfigurationsdatei auf dem Control-Volume-Set des SM-Pubsets
   - im SVL aller Hauptplatten der zum entsprechenden SM-Pubset gehörenden Volume-Sets
3. Löschen der alten und Anlegen aller neuen MRSCAT-Einträge
4. (nach Rückfrage): Ändern der Standardkatalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubsets und des geänderten Pubsets
5. Ändern der betroffenen Einträge im IMON-SCI auf dem Daten-Pubset und (nach Rückfrage) auch auf dem Home-Pubset.

Bei MODE=*COMPLETE-SHC-RENAME werden nur die Aktionen 4 und 5 ausgeführt, d.h.:

1. (nach Rückfrage): Ändern der Standardkatalogkennung im Benutzerkatalog (SYSSRPM) des Home-Pubsets und des geänderten Pubsets
2. Ändern der betroffenen Einträge im IMON-SCI auf dem Daten-Pubset und (nach Rückfrage) auch auf dem Home-Pubset.

Für eine ausführliche Beschreibung des Ablaufs einschließlich aller evtl. notwendiger Nacharbeiten siehe "Ablauf von PVSREN" und "Einschränkungen und Nacharbeiten".

Format

RENAME-PUBSET-OR-VOLUME-SET
NAME = <catid 1..4> ,NEW-NAME = <catid 1..4> ,SYSID = *STD / *SAME / <integer 65..192> ,MODE = *NORMAL-RENAME / *COMPLETE-SHC-RENAME ,PUBSET-PARAMETERS = *STD / *SAME

Operanden

NAME = <catid 1..4>
Gibt die Katalogkennung des zu ändernden Pubsets oder Volume-Sets an (Ausgangs-Katalogkennung).

NEW-NAME = <catid 1..4>
Bestimmt die neue Katalogkennung, die der Pubset oder Volume-Set nach der Konvertierung oder Umbenennung erhalten soll (Ziel-Katalogkennung).

SYSID =
Die Systemkennung ist nur für Home-Pubsets von Bedeutung und wird deshalb nur für SF-Pubsets ausgewertet. Die Systemkennung wird in das SVL des SF-Pubsets eingetragen und identifiziert das System, das diesen Home-Pubset verwendet, in einem Shared-Pubset-Verbund.
Für SM-Pubsets und Volume-Sets wird der Operand SYSID ignoriert.

SYSID = *STD
In Abhängigkeit von Ausgangs- und Ziel-Katalogkennung wird die Systemkennung folgendermaßen bestimmt: ,

catid-alt	catid-neu	Berechnung der Systemkennung
einstellig	einstellig	sysid = catid
mehrstellig	einstellig	sysid = catid
einstellig	mehrstellig	sysid wird umgesetzt: A=65 ... Z=90, 0=91 ... 9=100
mehrstellig	mehrstellig	sysid wird übernommen

SYSID = *SAME
Die Systemkennung wird vom Ausgangs-Pubset übernommen. Diese Angabe ist nur möglich, wenn die Ausgangs- und die Ziel-Katalogkennung mehrstellig sind.

SYSID = <integer 65..192>
Diese Angabe ist nur möglich, wenn die Ziel-Katalogkennung mehrstellig ist.
Bei einstelliger Ziel-Katalogkennung ist die Systemkennung gleich der Katalogkennung und kann nicht definiert werden.

MODE =
Durch diesen Operanden wird der Verarbeitungsmodus und damit die auszuführenden Einzelfunktionen festgelegt.

MODE = *NORMAL-RENAME
Der angegebene Pubset oder Volume-Set wird vollständig umbenannt, d.h. Katalogkennung, Benutzerkatalog und SCI werden angepasst.

MODE = *COMPLETE-SHC-RENAME
Eine durch SHC-OSD bei der Abtrennung bereits teilweise durchgeführte Umbenennung eines Pubsets, d.h. die VSNs sind bereits umbenannt, nicht aber die Verweise im IMON-SCI und im Benutzerkatalog (SYSSRPM), wird durch PVSREN vervollständigt. Dabei werden die Benutzerkataloge und das IMON-SCI angepasst.
Die Anpassungen werden nur nach Rückfrage durchgeführt. Wurde der Home-Pubset von SHC-OSD abgetrennt, werden die folgenden Funktionen nicht angeboten:

- Umbenennung der Benutzereinträge auf dem Home-Pubset
- Umbenennung des IMON-SCI auf dem Home-Pubset

Bei diesem Modus ist zu beachten, dass der Operand NAME mit der Katalogkennung des ursprünglichen Pubsets und der Operand NEW-NAME mit der Katalogkennung des durch SHC-OSD erstellten Pubsets anzugeben ist.

PUBSET-PARAMETERS =
Dieser Operand legt fest, welche Einstellungen aus dem MRSCAT des ursprünglichen Pubsets in den MRSCAT des umbenannten Pubsets übernommen werden. Der Operand wird nur beim Umbenennen eines SF- oder SM-Pubsets ausgewertet, nicht aber beim Umbenennen eines Volume-Sets.

PUBSET-PARAMETERS = *STD
Folgende MRSCAT-Einstellungen des ursprünglichen Pubsets werden übernommen:

- Pubset-Typ
- Gerätetyp der Pubres-Platte
- Einstellungen zur Rechner-lokalen Mehrbenutzbarkeit
- BCAM-Name des Partner-Systems

PUBSET-PARAMETERS = *SAME
Neben den MRSCAT-Einstellungen bei PUBSET-PARAMETERS=*STD werden folgende MRSCAT-Einstellungen des ursprünglichen Pubsets zusätzlich übernommen:

- Pufferanzahl
- Puffer-Ablageort
- Wartezeiten für Batch- und Dialogaufträge
- Zugriffsbeschränkung für Benutzer mit etwaiger Sondererlaubnis
- XCS-Pubset-Eigenschaft
- Remote-Import-Möglichkeiten
- mnemotechnische Gerätebezeichnung der Pubres-Platte
- EAM-Parameter

Bei SF-Pubsets werden außerdem übernommen:

- Einstellungen zur SPEEDCAT-Nutzung
- Direktallokierungsbeschränkungen
- spezifische Allokierungsparameter

> **i** Falls das Umbenennen mit PUBSET-PARAMETERS = *SAME mit der Meldung PVR0142 fehlschlägt, versucht PVSREN automatisch den Lauf mit dem Operanden PUBSET-PARAMETERS = *STD zu wiederholen.

11.6.2.6 RESTART-RENAMING - Neustart der Konvertierung oder Umbenennung

Mit dieser Anweisung wird eine abnormal beendete oder unterbrochene Konvertierung von Pubset-Notationen oder Umbenennung von Pubsets oder Volume-Sets neu gestartet und abgeschlossen.

Kommt es während der Abarbeitung der Anweisung RENAME-PUBSET-OR-VOLUME-SET zu einer abnormalen Beendigung, hat der Pubset oder Volume-Set einen inkonsistenten Zustand. Ein Teil der Daten (z.B. der Dateikatalog) wurde bereits geändert, der andere Teil (z.B. die VSN im SVL) hat noch den alten Inhalt.

Ein solcher inkonsistenter Zustand kann mit der Anweisung RESTART-RENAMING behoben werden. Dabei wird die angeforderte Konvertierung oder Umbenennung nochmals mit dem Ziel durchgeführt, dass der Pubset die angegebene neue Katalogkennung erhält.

Voraussetzung für einen erfolgreichen Neustart ist, dass die ursprüngliche Plattenkonfiguration nicht verändert wurde. Weiter wird eine so genannte Loggingdatei benötigt, die PVSREN selbstständig am Beginn einer Konvertierung oder Umbenennung anlegt und erst bei normaler Beendigung derselben wieder löscht. Diese Datei enthält alle Daten, die für einen Neustart der abgebrochenen Funktion notwendig sind.

Die Loggingdatei wird standardmäßig unter der Benutzerkennung TSOS angelegt und hat den Namen SYSLOG.PVSREN.<catid-alt>.<catid-neu>. Sie ist mit einem Schreibpasswort geschützt und darf nicht gelöscht, umbenannt oder verändert werden.

Format

RESTART-RENAMING
FROM-FILE = <filename 1..54>

Operanden

FROM-FILE = <filename 1..54>
Name der Loggingdatei, die für die abnormal beendete oder unterbrochene Konvertierung oder Umbenennung angelegt wurde.

Der Dateiname lautet SYSLOG.PVSREN.<catid-alt>.<catid-neu> und wird immer unter der Benutzerkennung TSOS abgelegt.

11.6.2.7 RESTORE-LABELS-OF-PUBSET - Pubsets aus Spiegel-Pubsets mit Doppelpunkt-Notation rückbenennen

Nach der Restaurierung eines Pubset aus einem Spiegel-Pubset in „Doppelpunkt-Notation“ (SPECIAL-VSN), werden mit dieser Anweisung die VSNs des Pubset auf die entsprechende Punkt- bzw. PUB-Notation zurückbenannt.

Hierzu werden nacheinander alle Platten, die zu einer Einheit gehören, belegt und die Doppelpunkt-Notation durch die Punkt- bzw. PUB-Notation ausgetauscht.

Ein Wiederaufsetzen bei einer abgebrochenen Rückbenennung ist jederzeit möglich. In diesem Fall wird für jede Platte, die bereits umbenannt wurde, eine Warnung ausgegeben, dass diese Platte nicht in der Doppelpunkt-Notation vorliegt. Liegt eine benötigte Platte weder in der Doppelpunkt-Notation noch in der Standard-Notation vor, wird der Lauf abgebrochen. In diesem Fall muss die fehlende Platte zur Verfügung gestellt und die Rückbenennung erneut gestartet werden.

Format

RESTORE-LABELS-OF-PUBSET
CATID = <catid 1..4>

Operanden

CATID = <catid 1..4>
Katalogkennung des SM-/SF-Pubsets.

11.6.2.8 SET-NAME-OF-NEW-VOLUME-SET - Volume-Set-Name beim Erzeugen eines SM-Pubsets aus Spiegel- Platten festlegen

Beim Erzeugen eines neuen SM-Pubsets aus Spiegel-Platten (Clone-Units, Snap-Units) wird mit der Anweisung SET-NAME-OF-NEW-VOLUME-SET der neue Name von Volume-Sets festgelegt.

Beim Erzeugen eines neuen SM-Pubsets aus den Spiegel-Platten eines SM-Pubsets müssen die Volume-Sets des neuen Pubsets neue Katalogkennungen erhalten (Umbenennungsvorschrift). Für jeden Volume-Set des Pubsets wird die Abbildung der Katalogkennung eines Volume-Sets des Ausgangs-Pubsets auf die Katalogkennung des neuen Pubsets explizit mit der Anweisung SET-NAME-OF-NEW-VOLUME-SET spezifiziert, also eine Anweisung pro Volume-Set. Die Anweisungen werden in einer Datei hinterlegt, auf die beim Erzeugen eines SM-Pubsets in der Anweisung CREATE-PUBSET-FROM-MIRROR Bezug genommen wird.

Voraussetzungen

- Die Anweisung kann ausschließlich in einer Datei (als Eingabe für die Anweisung CREATE-PUBSET-FROM-MIRROR) gegeben werden. Die Angabe in einer anderen Umgebung wird abgewiesen.
- Für jeden Volume-Set ist genau eine Abbildungsvorschrift für die Katalogkennung erforderlich.
- Die Katalogkennungen der neuen Volume-Sets dürfen nicht bereits im System vorhanden sein.

Siehe auch Abschnitt „Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel-Pubsets“ (Voraussetzungen für das Erzeugen neuer Pubsets aus Spiegel- Pubsets).

Für eine ausführliche Beschreibung des Ablaufs einschließlich aller evtl. notwendiger Nacharbeiten siehe "Ablauf von PVSREN" und "Einschränkungen und Nacharbeiten".

Format

SET-NAME-OF-NEW-VOLUME-SET
NAME = <catid 1..4> ,NEW-NAME = <catid 1..4>

Operanden

NAME = <catid 1..4>
Gibt die Katalogkennung eines Volume-Sets des Source-Pubsets an, aus dessen Spiegel-Platten ein neuer SM-Pubset erzeugt werden soll.

NEW-NAME = <catid 1..4>)
Gibt die Katalogkennung des entsprechenden Volume-Sets des neuen, aus den Spiegel-Platten zu erzeugenden SM-Pubsets an.

11.6.2.9 SHOW-LOGGING-OPTIONS - Anzeigen der gültigen Werte für die Protokollierung

Mit der Anweisung SHOW-LOGGING-OPTIONS kann die Auflistung gültiger Werte für die Protokollierung angefordert werden. Die Werte werden mit der Anweisung MODIFY-LOGGING-OPTIONS festgelegt.

Format

SHOW-LOGGING-OPTIONS

Diese Anweisung besitzt keine Operanden.

Die eingestellten Werte werden wie folgt aufgelistet:

```
%CURRENT LOGGING OPTIONS:
%  INFORMATION:<value>
%  OUTPUT:<value>
```

12 RMS - Rep-Montage-System

Versionsstand: **RMS V7.1G**

Privilegierung: **TSOS**

Das **R**ep-**M**ontage-**S**ystem (RMS) dient der Verwaltung, Dokumentation, Lieferung und Montage von Repmengen und Vorab-Korrekturen. Das Produkt wird sowohl in internen Dienststellen (Anwendergruppen 1 und 2, s.u.), als auch von externen Kunden (Anwendergruppen 3 und 4, s.u.) eingesetzt.

RMS ist als vierstufige Anwendung für die folgenden vier Anwendergruppen konzipiert (siehe auch "Anwendergruppen und Aktionen"):

- Anwendergruppe 1: Entwicklung oder vergleichbare Instanz
- Anwendergruppe 2: Support Center bzw. Service Center
- Anwendergruppe 3: Kunde
- Anwendergruppe 4: Kundeninternes Data Center

Allen Anwendergruppen steht der gleiche Funktionsumfang von RMS zur Verfügung. Die Handhabung und Durchführung aller notwendigen Aktionen werden dadurch vereinheitlicht und vereinfacht. Beim Kunden ist kein zusätzlicher Aufwand erforderlich.
Jede Anwendergruppe hat eine eigene Datei (Depot), in die anwendergruppenspezifisch folgende Informationen eingetragen sind:

- alle Rep-Korrekturen und Beschreibungen
- deren Herkunft und Produktzugehörigkeit
- Umfang und Zeitpunkt aller Aktionen
- Laderdefinition und Geschichte

Alle Funktionen sind im Dialogbetrieb einsetzbar und menügesteuert. Einige Funktionen können zusätzlich im Batchbetrieb verwendet werden. Dies ermöglicht den automatischen Ablauf aller wesentlichen Funktionen - von der ersten bis zur vierten Anwendergruppe - ohne manuellen Eingriff unter Berücksichtigung aller anwenderspezifischen Belange.

> ¡ Im (deutschen) RMS-Dialog wird zuweilen „Anlagenlader“ als Synonym zu „Lader“ (= Subsystem-Lader) verwendet.

Einsatz von RMS

Das folgende Bild verdeutlicht den zeitlichen Ablauf im Einsatz von RMS bzgl. der vier Anwendergruppen.

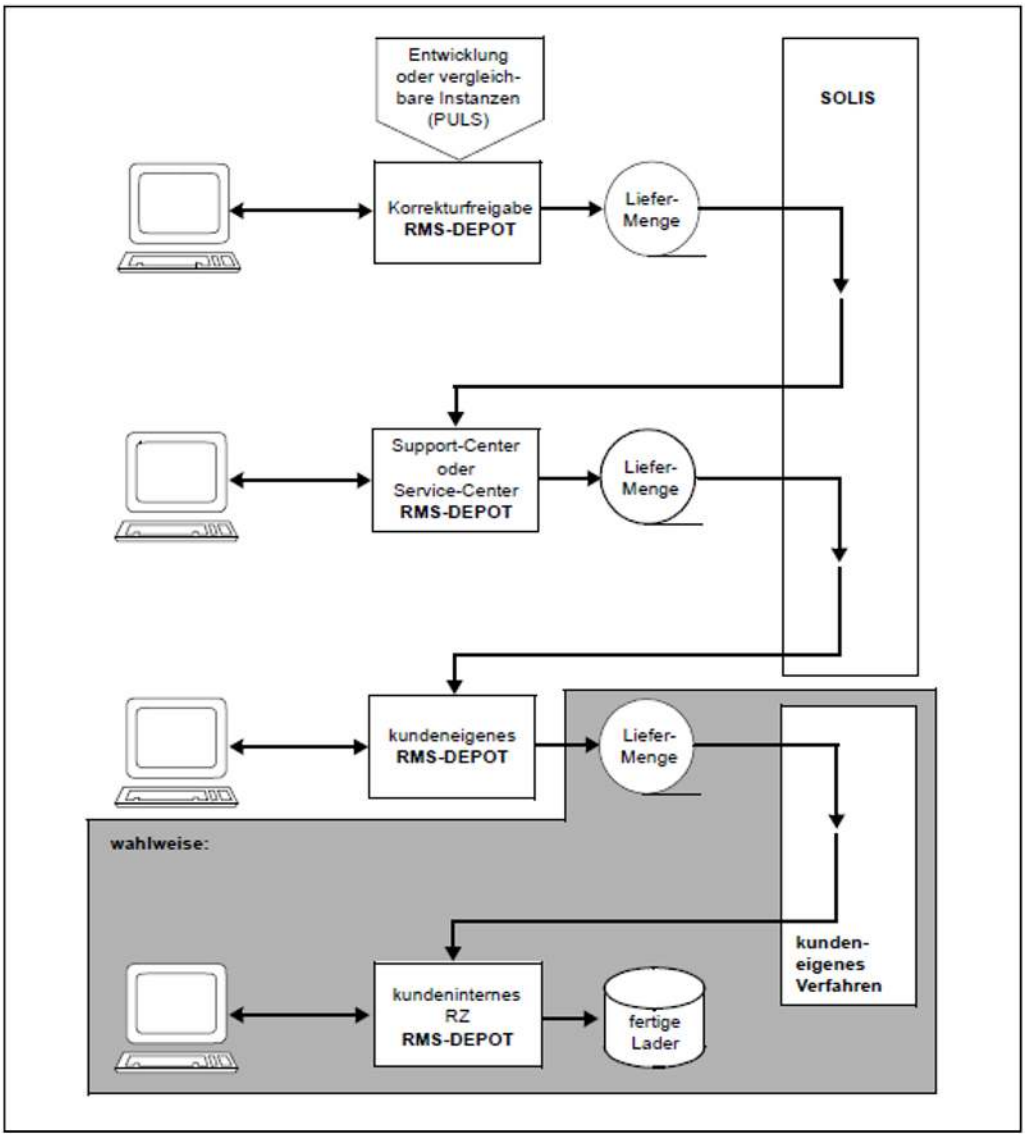


Bild 16: Einsatz von RMS

Hinweise

1. Aus dem Verfahren PULS wird produktversionsspezifisch (d.h. für eine bestimmte Version eines Produktes) eine Menge von Software-Korrekturen (Reps) erzeugt. Diese Menge ist in einer Rep-Datei abgelegt.
2. Diese Rep-Datei (RMS-Liefermenge) wird durch RMS entgegengenommen und in ein Depot eingebracht. In diesem Depot sind zu allen Software-Produkten versionsspezifisch Rep-Korrekturen, Kommentare zu diesen Korrekturen, Status (aktiv, nicht aktiviert) usw. eingetragen. Die neu eingefahrene RMS-Liefermenge kann über RMS im Depot weiter bearbeitet werden, z.B. durch eigene, weitere (Rep-)Korrekturen ergänzt werden oder es können (Rep)-Korrekturen aktiviert, deaktiviert oder mit Kommentaren versehen werden.
3. Aus dem Depot kann nun entweder wiederum eine RMS-Liefermenge erzeugt werden, um sie an eine nächste Anwendergruppe zu übergeben. Die Übergabe erfolgt über SOLIS oder kundeneigene Verfahren. Die nächste Anwendergruppe bringt dann mittels RMS diese RMS-Liefermenge in ihr Depot ein und bearbeitet sie ggf. weiter.

Es kann aber auch aus dem Depot ein Lader erzeugt werden. Ein Lader ist diejenige Rep-Datei, deren Inhalt beim Starten eines Produkts angewandt wird. Der Lader wird von RMS als anwendergruppen-spezifische Auswahl von Rep-Korrekturen zu einer bestimmten Produktversion erzeugt und unter einem fixen, produktspezifischen Namen abgespeichert.

4. Punkt 3 kann durch die Anwendergruppen 1, 2 und 3 so durchgeführt werden. Anwendergruppe 4 kann lediglich Lader erzeugen.

12.1 Anwendergruppen und Aktionen

In der Regel können die vier Anwendergruppen und Aktionen zwischen diesen Anwendergruppen wie folgt definiert werden:

Anwendergruppe 1 - Entwicklung oder vergleichbare Instanz

Betreut werden alle für Kunden freigegebenen Produkte und solche Produkte, die sich in einer Pilotierung befinden.

Alle Korrekturen werden einer fachlichen und formalen Qualitätskontrolle unterzogen. Für freigegebene Produkte wird in zeitlichen Intervallen, je nach Gewichtung des Produktes und der behobenen Fehler, eine fest definierte Korrekturmenge an die nächste Anwendergruppe übergeben.

Produkte, die sich in der Pilotierung befinden, werden nicht über den SOLIS2-Weg weitergegeben.

Anwendergruppe 2 - Support Center bzw. Service Center

Es werden alle Lieferungen von Anwendergruppe 1 in das eigene Depot übernommen.

Auf Grund von Problemmeldungen oder Vorabkorrekturen können sich Änderungen zu den angelieferten Korrekturmengen ergeben. Diese Modifikationen werden in das eigene Depot eingetragen. Dadurch sind die Korrekturmengen immer auf einem aktuellen Stand.

Bei der Übergabe von Lieferungen oder Korrekturauslieferungen an den Kunden werden diese Modifikationen automatisch berücksichtigt.

Anwendergruppe 3 - Kunde

Es werden alle Lieferungen von Anwendergruppe 2 in das eigene Depot übernommen.

Auf Grund von Problemmeldungen oder Vorabkorrekturen können sich Änderungen zu den angelieferten Korrekturmengen ergeben. Diese Modifikationen werden in das eigene Depot eingetragen. Dadurch sind die Korrekturmengen immer auf einem aktuellen Stand.

Bei der Übergabe von Lieferungen oder Korrekturauslieferungen an das kundeninterne Data Center werden diese Modifikationen automatisch berücksichtigt.

Anwendergruppe 4 - Kunden-internes Data Center

Es werden die Lieferungen von der Anwendergruppe 3 übernommen. Im kundeneigenen Data Center fällt kein zusätzlicher Aufwand an.

12.2 RMS-Depot

Das RMS-Depot ist eine BS2000-Datei und stellt den zentralen Datenpool der RMS-Anwendung dar. Im Depot werden alle Korrekturen (Reps) nach Produkt, Fehlernummer und Datum spezifisch eingetragen. Jede Anwendergruppe hat ein *eigenes* Depot, in dem nur die Daten enthalten sind, die die jeweilige Anwendergruppe benötigt. Alle produktspezifischen Daten für eine effiziente und übersichtliche Korrekturverwaltung sind somit in einer Datei enthalten. Alle notwendigen Aktionen lassen sich produktspezifisch automatisch oder manuell durchführen.

Alle Anwenderaktionen werden mit Art und Zeitpunkt in das Depot eingetragen. Transparenz bei der Korrekturverwaltung ist damit sichergestellt. Man kann sich über alle durchgeführten Aktionen informieren. Jede Aktion ist zu jedem Zeitpunkt reproduzierbar. Da alle Aktionen eindeutig in das Depot eingetragen werden, hat jeder Anwender die Möglichkeit, sich über alle durchgeführten Aktionen zu informieren und sie nach seinen Belangen zu beeinflussen. Dadurch werden u.a. Revisionssicherheit und Datenschutz gewährleistet.

Zusätzlich werden eventuelle Servicetätigkeiten vereinfacht und für alle Anwendergruppen transparent gemacht.

Sämtliche Texte werden nach Textebenen-Namen und Textebenen-Schlüssel organisiert.

Logischer Aufbau des Depots:

- Produktspezifischer Teil
  - Reps
  - Korrekturbeschreibungen
  - Lieferinformation

  Abhängig vom Produktumfang dieses Anwenders werden die jeweiligen Korrekturmengen in der vorliegenden Anwendergruppe erstellt und in dieses Depot eingebracht.
- Anwenderspezifischer Teil
  - Modifikation von Korrekturmengen
  - Eigene bzw. Vorab-Korrekturen
  - Zurückgewiesene Korrekturen
  - Korrekturdokumentation
  - Dokumentation aller Art
  - Daten zu Laderbau-Verwaltung und -Generierung
  - Auswahl optionaler Korrekturen
  - Daten für automatische oder manuelle Ladergenerierung/Installation

12.3 Flexibler Einsatz von RMS

RMS bietet dem Anwender mehrere Möglichkeiten, seine Anwendungen nach eigenem Bedarf zu konfigurieren.

- Mittels eine Menüdefinitions-Datei (Standarddateiname `SYSDAT.RMS.071`) kann jeder gewünschte Menüaufbau dargestellt werden.
- Durch Zusammenstellen aller Info-Module in einer getrennten Modulbibliothek kann ein reines *Info-RMS* zur Verfügung gestellt werden.
- Anwendereigene Funktionsmodule können in der RMS-Laderbibliothek und in der Menüdefinitions-Datei eingetragen werden, um den Funktionsumfang von RMS zu erweitern.
- Alle Aufgaben, die automatisch ablaufen sollen, können als Anweisungen in einer Prozedur oder einem Enter-Job gespeichert werden.

Durch Auswahl von entsprechenden Funktionen und Erstellen entsprechender Batchjobs lassen sich z.B. folgende RMS-Systeme für spezifische Aufgabenstellungen erzeugen:

Die einfache Anwendung

Die für RMS notwendigen Dateien sind in das System zu übernehmen. Dazu gehört auch das Depot, das von der vorhergehenden Anwendergruppe initialisiert wurde.
Bei jeder Korrekturlieferung werden alle Korrekturen automatisch in das Depot übernommen. Alle notwendigen Lader werden automatisch generiert und installiert. Erst bei einem Produkt- oder Betriebssystemwechsel sind eventuell manuelle Eingriffe notwendig. Der automatische Ablauf wird über (in der Regel mitgelieferte) Batchprozeduren erreicht, die durch die zum Software-Produkt gehörenden Installationsprozeduren aufgerufen werden. Beim Kunden (Anwendergruppe 3 und 4) findet im Allgemeinen dieses Verfahren Anwendung. Beim Einsatz neuer Produkte, deren Korrekturen in den BS2000-Systemlader eingemischt werden müssen, ist die Produktauswahl neu zu definieren. Diese werden dann bei jeder Korrekturlieferung berücksichtigt.

Die Standardanwendung

Der Umfang der Lader (Produkte und optionale Korrekturen) wird selbstständig definiert. Bei eventuell auftretenden Fehlern werden auf Anweisung Vorab-Korrekturen in das Depot eingebracht. Alle weiteren Aktionen laufen wie bei der einfachen Anwendung automatisch ab.

Die komplexe Anwendung

Es wird der volle Funktionsumfang genutzt. Der Automatisierungsgrad der Abläufe wird nach eigenen Belangen definiert.

12.4 Bedienung von RMS

- Starten und Beenden von RMS
- Eingaben an RMS

12.4.1 Starten und Beenden von RMS

Sie starten RMS über eine Startprozedur mittels des BS2000-Kommandos

```
/CALL-PROCEDURE <procedure_name>
```

Die mitgelieferte Standard-Startprozedur heißt SYSPRC.RMS.071. In dieser sind die Basisparameter für RMS vereinbart. Eine Beschreibung der Prozedur finden Sie im Abschnitt „Startprozedur und Basisparameter“.

Sie beenden RMS durch Drücken von K1 auf der obersten Menüstufe. Beantworten Sie die anschließende Frage, ob Sie RMS beenden wollen, mit „Ja“.

12.4.2 Eingaben an RMS

Die Funktionen von RMS stehen Ihnen im Dialog menügeführt zur Verfügung. Die Steuerung erfolgt über K-, F-Tasten.

Mit der Taste MAR treffen Sie eine Auswahl bzw. machen eine getroffene Auswahl wieder rückgängig. Die Auswahl oder Eingaben in einer Maske müssen mit der Taste DUE bestätigt werden. Mit der Taste K1 brechen Sie eine Funktion ab und kehren in die nächst höhere Menüstufe zurück.

Mit Eingabe der Zeichen „/“ bzw. „x“ im Eingabefeld vor einem Eintrag markieren Sie die gewünschten Einträge in einer Auswahlliste. Mit Löschen des Auswahlzeichens oder Überschreiben mit Leerzeichen machen Sie eine getroffene Auswahl wieder rückgängig. Die Auswahl oder Eingaben in einer Maske müssen mit der Taste DUE bestätigt werden. Mit der Taste K1 brechen Sie eine Funktion ab und kehren in die nächst höhere Menüstufe zurück.

Zwischen verschiedenen Listenseiten können Sie vor- und zurückblättern indem Sie „nächste Seite“ oder „vorige Seite“ mit der Taste MAR markieren und dies mit der Taste DUE bestätigen.

Hinweise zur Steuerung finden Sie in jedem Funktionsschirm im unteren Bereich. Über die Taste K9 bzw. ESC + ? können Sie Hilfeinformation zu den einzelnen Bildschirmmasken anfordern. Eine Auflistung der Funktion aller Funktionstasten finden Sie im Abschnitt „Verwendung der Funktionstasten“.

Der Gesamtumfang von RMS und damit die Funktionalität, die in den einzelnen Bildschirmmasken angeboten wird, ist über die einzelnen Funktionsmodule geregelt (siehe "Modulliste") und steht somit fest. Sie können aber mittels einer Funktionsdatei den Inhalt der Menüs und damit die darin angebotenen Funktionen sowie die Menütitel und Bezeichnung der Funktionen selbst zusammenstellen und benennen (siehe "Funktionsumfang definieren"). So ist es z.B.

möglich, Funktionen in mehreren Menüs anzubieten. Im Abschnitt „Übersicht über dieFunktionen des Standardmenüs“ wird auf das Standardmenü von RMS bezug genommen, das jedoch vom konkret vorliegenden abweichen kann.

Im Batchbetrieb stehen die Funktionen von RMS über eine Anweisungsschnittstelle zur Verfügung. Beispiele finden Sie im Abschnitt „Typische Einsatzfälle für RMS“, eine Beschreibung aller Anweisungen im Abschnitt „Anweisungen“.

In den RMS-Menüs und der vorliegenden Dokumentation zu RMS werden folgende Begriffe und Abkürzungen verwendet:

Begriff / Abkürzung	Bedeutung
RMS-Liefermenge	Zum einen die aus PULS gewonnene REP-Datei zu einer Produktversion, die durch RMS ins Depot eingefahren werden. Zum anderen die aus RMS erzeugte Übergabemenge, die an die nächste Anwendergruppe übergeben und dort durch RMS ins Depot eingefahren wird.
Lader /Anlagenlader	Diejenige Rep-Datei, deren Inhalt beim Starten eines Produkts angewandt wird. Der Lader wird von RMS als anwendergruppen-spezifische Auswahl von Repkorrekturen zu einer bestimmten Produktversion erzeugt und unter einem fixen, produktspezifischen Namen abgespeichert.
Rep-Datei	Datei, die alle Reps zu einer Produktversion enthält. Ist diese Datei unter einem fixen, produktspezifischen Namen abgespeichert, fungiert sie als Lader.
SUBNO	Korrekturnummer oder Fehlermeldungsnummer zu einer Produktversion im Verfahren PULS
jd	Datum nach dem julianischen Kalender, zur Darstellung siehe auch "Darstellung des JD"
LB	Lieferbestandteil

Tabelle 14: Begriffe und Abkürzungen zu RMS

12.5 Funktionsübersicht

Jede SOLIS2-Lieferung beinhaltet eine für SOLIS2 zusammengestellte RMS-Version.
Es ist deswegen nicht notwendig, dass Sie vorher RMS installiert haben. Das Depot wird automatisch verwaltet.

Der folgende Abschnitt gibt eine kurze Übersicht über die in RMS angebotenen Funktionen. Am Ende des Abschnitts finden Sie eine Auflistung der Funktionen des ausgelieferten Standardmenüs. Da diese Einstellungen jedoch über die Definitionsdatei geändert werden können, kann das Menü im konkreten Fall vom beschriebenen abweichen. Über die Zuordnung zu den Modulen ist aber jederzeit der aktuelle Bezug herstellbar.

Eine Beschreibung der Funktionensmasken im Dialog finden Sie ab Abschnitt „Transfervon Korrekturmengen“, die entsprechenden Anweisungen, sofern verfügbar, sind beschrieben im Abschnitt „Anweisungen“.

12.5.1 Transfer von Korrekturmengen

Der Korrekturtransfer ist die Schnittstelle zwischen den Anwendergruppen. Die dafür vorhandenen Funktionen können im Dialog- und Batchbetrieb verwendet werden. Durch die Batchfähigkeit wird ein vollautomatischer Ablauf ermöglicht.

Folgende Funktionen stehen zur Verfügung:

- Liefern
  Lieferung definierter Korrekturmengen und ausgewählter Dokumentation, im festen Format, mit automatischer Versionsnummernvergabe.
- Einfahren
  Mit dieser Funktion werden folgende Aktionen angestoßen:
  - Übernahme neuer Korrekturmengen und Dokumentation
  - Automatischer Abgleich vorhandener Notizbuchaktionen
  - Erstellen einer Differenzliste

12.5.2 Laderdienste

Für individuelle Einstellungen und Abläufe stehen Funktionen im Dialog und Batch zur Verfügung.

- Definieren
  Mit dieser Funktion wird der Umfang und Inhalt eines Anlagenladers definiert. Neben der Produktauswahl kann auch die Verknüpfung mit OPT-REP Einstellungen und/oder Ladermodifikationen durchgeführt werden.
- Installieren
  Bei der Installation werden Anlagenlader generiert und unter dem vereinbarten Installationsnamen katalogisiert. Die dafür benötigten Parameter werden der jeweiligen Anlagenladerdefinition entnommen.
- Erstellen
  Die Erstellung läuft wie die Installation bis auf das Katalogisieren gleich ab. Sollen die erstellten Anlagenlader eingesetzt werden, müssen die Dateien manuell auf den gültigen Installationsnamen umbenannt werden

Die Funktionen können teilweise automatisch zum Ablauf gebracht bzw. so eingestellt werden, dass ein Eingriff des Anwenders nicht notwendig ist.

Die **Anlagenladerdefinition** legt die folgenden Einstellungen fest:

- Installationseigenschaften
  Einstellung kundenspezifischer Installationsmerkmale.
- Produkt- oder Produktversionsauswahl
  Auswahl der Produkte bzw. Produktversionen, die zu diesem Anlagenlader gehören.
- OPT-REP-Einstellungen
  Auswahl optionaler Korrekturen, die, falls vorhanden, bei jeder Ladergenerierung berücksichtigt werden sollen.
- Ladermodifikationen
  Auswahl von Korrekturen, die bei jeder Ladergenerierung unabhängig von ihrem Status berücksichtigt werden sollen.

Die Anlagenladerdefinition kann unter einem frei wählbaren Namen gespeichert werden.

12.5.3 Korrekturverwaltung - Notizbuch

In der Korrekturverwaltung können mit der Notizbucherfassung Veränderungen gegenüber übernommener Korrekturmengen durchgeführt werden. Folgende Funktionen stehen zur Verfügung:

- Übernehmen eigener und Vorab-Korrekturen
- Zurückweisen fehlerhafter Korrekturen
- Austausch von Korrekturen
- Zurücksetzen von Notizbuchaktionen
- Informationsmöglichkeiten

Die eingetragenen Notizbuchaktionen werden bei der Übernahme neuer Korrekturmengen (Einfahren), beim Korrekturtransfer (Liefern) und bei der Lader-Generierung automatisch berücksichtigt.

12.5.4 Informationsdienste

Im Dialog stehen umfangreiche produkt- und korrekturspezifische Informationsmöglichkeiten zur Verfügung.

Produktspezifische Information

- Übersicht aller im Depot befindlichen Produkte (Module RMSST00)
- Erstellungs-, Übernahme- und Freigabe-Datum (Module MATRIX)
- Hinweis auf noch nicht gelieferte Notizbuchaktionen (Module NBKINF)

Korrekturspezifische Information

- Zuordnung zum Produkt, Einfahrmengen und Lieferungen (Module INFODATA)
- Überblicksliste der Zuordnung zum Produkt, Einfahrmengen und Liefermengen (Module MATRIX)
- Vergleich zwischen Einfahr-, Liefermengen und Anlagenladerdateien, um mögliche Unterschiede festzustellen (Module VGL)
- Notizbuchaktionen (Module NBKINF)
- Korrekturinhalt (Module INFODATA)
- Korrekturarten (Module INFODATA)
- Korrekturspezifische Texte (Module INFODATA)

Das Modul INFODATA kann über die Funktionsauswahlmaske oder mit der Taste K12 aufgerufen werden.

Mit der Taste K12 wird eine andere laufende Funktion nicht beendet. Nach Beendigung dieser Dialoginformationsfunktion wird zur unterbrochenen Funktion zurückgekehrt.

Für die meisten Funktionsgruppen werden zusätzliche Dialog- und Listen-Informationsmöglichkeiten angeboten.

12.5.5 Textverarbeitung

Es können Texte für Korrekturen oder Produkte eingetragen, geändert oder gelöscht werden. Diese Texte können durch eine frei definierbare hierarchische Struktur als passive oder aktive Informationen hinterlegt werden.

12.5.6 RMS-Verwaltung

Zur Administration gehören die folgenden Tätigkeiten und Funktionen, die zum Betrieb von RMS notwendig sind:

- Installation
- Definieren Gesamt-Funktionsumfang
- Definieren Funktionsumfang und Zugriffsberechtigung für Benutzer
- Depotpflege und Wartung (EXPORT/IMPORT)

12.5.7 Übersicht über die Funktionen des Standardmenüs

Menüpunkt	Sub-Menüpunkt	Modul	Details[1)] siehe
Rep-Verwaltung			
	Korrekturverwaltung - Notizbuch	RMSNOTIZ	
	Liefermengen übernehmen	RMSEINF	
	Ladererstellung und Installation	RMSLGEN	
	Laderverwaltung und Information	RMSLVWIN	Hilfe
	Textverarbeitung (Produktnotizen, Textebenen)	RMSEDIT	
	Liefermengen erzeugen	RMSLIEF	
	markierte Anlagenlader erstellen	MONTAGE	2)
	EDT aufrufen	CALLEDT	Handbuch EDT
Informations-Dienste			
	Informationsdienste	INFODATA	
	Zeichenorientiertes Suchen	RMSFIND	
	Vergleich von Rep-Mengen (neues Format)	VGL	
	Vergleich von Repmengen (altes Format)	VGLSTART	Hilfe
	EDT aufrufen	CALLEDT	Handbuch EDT
Übersichts-Listen			
	Korrektur - Gesamtübersicht	Matrix	
	Notizbuch - Dialoginformation	NBKINFO	
	Notizbuch - Gesamtübersicht	NOTBLIST	
	Depot-Übersichtsliste	RMSST00	2)
	EDT aufrufen	CALLEDT	Handbuch EDT
Administration			
	DEPOT-Pflege (Export - Import)	RMSEXIM	
	LB Versionen reorganisieren	RMSREORG	Hilfe
	RMS Optionen einstellen	RMSOPTS	Hilfe
	Version eines Produktes ändern	RMSCNGPR	Hilfe
	Liefermengen erzeugen RMS V6.0 Format	LIEFERN	Hilfe

	EDT aufrufen	CALLEDT	Handbuch EDT

[1] Detailinformationen zu den einzelnen Funktionsmasken erhalten Sie auch über die Hilfefunktion; falls in der Spalte „Hilfe“ eingetragen ist, nur über die Hilfefunktion.

[2] Es wird ohne weitere Funktionsmaske eine Ausgabedatei erzeugt.

12.6 Dialogorientierte Anwendung

Im folgenden Abschnitt werden die RMS-Funktionen aus Sicht des menügeführten Dialogs beschrieben. Zu jeder Funktionsmaske können Sie über die Tasten K9 bzw ESC+? Hilfeinformation anfordern.

12.6.1 Transfer von Korrekturmengen

Liefermengen erzeugen

Mit dieser Funktion können produktspezifische Korrekturmengen (Lieferbestandteile) erstellt werden. Diese dienen als Eingabe für das Depot der nächsten Anwendergruppe. Jeder Lieferbestandteil (LB) wird durch eine eindeutige Versionsnummer (LB-Vers) gekennzeichnet, die in den Anwendergruppen 1, 2 und 3 automatisch vergeben und in das Depot als gelieferte Menge eingetragen wird.

In der Anwendergruppe 4 (Kundeninternes Rechnenzentrum) können nur die in das Depot übernommenen Liefermengen wieder als Datei erstellt werden.

Erstellung von aktuellen Liefermengen

Diese Funktion kann nur von den Anwendergruppen 1, 2 und 3 verwendet werden.

Um aktuelle Mengen, d.h. LB mit Berücksichtigung des Notizbuches zu liefern, ist die entsprechende LB-Version einzugeben bzw. in der Auswahlmaske zu markieren.

Beim Erstellen einer Liefermenge der Anwendergruppe 1 kann in der Bestätigungsmaske der von RMS vorgeschlagene Laderbuchstabe geändert werden. In der folgenden Maske kann man sich zwischen einer kompletten Lieferung aller Texteinträge oder einer Differenzmenge zu einer bestimmten LB-Version entscheiden.

Bevor einer Lieferung erfolgt, wird überprüft, ob für alle optionale Korrekturen eine Beschreibung in der Textebene SK-OPT BESCHREIBUNG existiert. Fehlt eine Beschreibung, so wird die SUBNO der fehlenden Texte in der Protokolldatei aufgelistet.

Bei fehlenden Texten wird die Lieferung abgebrochen und die SUBNO der fehlenden Texte in der Protokolldatei aufgelistet.

Für die restliche Korrekturen werden *alle* Textebenen durchsucht und der erste Eintrag in die Ausgabedatei geschrieben.

Jede Lieferung kann für Test- oder Kontrollzwecke geprüft werden. Dabei muss das Feld `Prüfe Lieferung` markiert werden. Die gewünschte Lieferung wird in allen Einzelheiten geprüft.

Ausgabedateinamen:
Der Ausgabedateiname kann in der Anweisungszeile definiert werden. Ansonsten wird folgender Standarddateiname erzeugt:

`SYSRMS.<product>.<version>.<lb-version>`

Erstellung von Originalliefermengen

Diese Funktion steht allen 4 Anwendergruppen zur Verfügung. Jeder Anwender kann diese Aktion durchführen, da keine Einträge und Veränderungen im Depot durchgeführt werden.

Um Originalmengen, d.h. übernommene oder schon gelieferte Mengen zu erstellen, muss das Feld `Original Lieferung` markiert werden.

Lieferbestandteil-Versionsnummer

Eine Versionsnummer hat folgendes Aussehen: B01.03

Die ersten drei Stellen werden vom zentralen Service, die letzten beiden Stellen vom regionalen Service automatisch vergeben.

Der Laderbuchstabe (erste Stelle) wurde nur aus historischen Gründen beibehalten. Er legt fest, in welcher Reihenfolge die Liefermengen vom zentralen Service erstellt wurden.
(B00.00 bevor C00.00 usw.).

Die numerischen Werte werden bei einer aktuellen Lieferung nur dann erhöht, wenn im jeweiligen Depot Änderungen (Notizbucheinträge) gegenüber der letzten Lieferung vorgenommen wurden.

Im Falle einer kundeninternen Lieferung wird der Laderbuchstabe von Großschreibung in Kleinschreibung geändert (z.B. B --> b).

Liefermengen übernehmen

Mit dieser Funktion können Lieferbestandteile (Rep- und Text-Mengen und Software Konfigurationen) in das eigene Depot übernommen werden, die von einer vorherliegenden Anwendergruppe erstellt wurden.

Jeder zu übernehmende aktuelle Lieferbestandteil beinhaltet alle zum Zeitpunkt der Erstellung gültigen Korrekturen des Lieferdepots.

Es besteht folgende Einfahrlogik:

- KORREKTUREN (REP)
  Eine Korrektur wird pro Produkt nur einmal physikalisch im Depot gespeichert. Bei künftigen Übernahmevorgängen wird nur eine logische Zuordnung erstellt.
- TEXTE
  Jeder Text wird übernommen, d.h. schon im Depot vorhandene Texte werden bei gleichem Textebenen-Namen und Text-Schlüssel überschrieben.
- SW-KONF
  Jeder Produktgruppe wird mit den schon im Depot gespeicherten Gruppen verglichen und falls nicht im Depot vorhanden, wird der SW-Konfiguration um diese Produktgruppe ergänzt. Die Eigenschaften werden immer gespeichert aber werden nur aktiv benutzt wenn im Depot keine Veränderungen der SW-Konfigurations-Eigenschaften vorgenommen wurden.

Eine Liefermenge kann aus mehreren Lieferbestandteilen (LB), Textebenen (TE) und Software Konfigurationen (SW-K) bestehen. Ein Einfahrpaket kann aus mehreren Liefermengen bestehen.

Die produktspezifische Software-Konfiguration ist ein fester Bestandteil der Lieferdatei. Bei jeder Übernahme einer Lieferung werden die im Depot befindlichen Konfigurationen automatisch aktualisiert.

Jeder LB und jede TE ist durch eine eigene Steuerzeile gekennzeichnet. Diese werden bei einer Lieferung automatisch erzeugt. Die Steuerzeilen dürfen nicht verändert werden.

Die Steuerzeile für eine Textebene (TE) kann jeder Anwender selbst erstellen, um eigene TE einzurichten. Es stehen 32767 TE zur Verfügung. Beim Einfahren kann der Inhalt von bestehenden TE verändert werden. Das Erstellen und Ändern von Textebenen ist auch mit dem Texteditor möglich.

Das Format des Inputs für Textebenen sieht folgendermaßen aus:

Format Steuerzeile

```
KKKTTTTTTTTTTTTTTTTTTTTT*********tt/mm/jj*********************************
```

Erklärung:

KKK = Kennzeichen für Textebene
TTT...TTT = Name der Textebene (22 Stellen)
tt/mm/jj = Datum

Beispiel:

```
*T*SK-REP BESCHREIBUNG *********<date>*****************************
```

Format Textzeile

`PPPPPPPPBBBBBtexttext...`

Erklärung:

PPPPPPPP = PM 8-stellig
BBBBB = 5 Blanks oder ggf. -xxx- wobei xxx das jd ist
`texttext...` = PATCH-DISCRIPTION aus PULS

Beispiel:

```
*T*SK-REP BESCHREIBUNG   *********<date>*****************************
A0326168     FUNKTIONS-NAME: Name der OPT-Korrektur
A0346168     Jetzt folgt die PATCH-DESCRIPTION aus PULS!
```

> ¡ Bei optionalen Korrekturen muss der Text in beiden Textebenen stehen (SK-REP und SK-OPT). Die Texte zu optionalen Korrekturen müssen mit „FUNKTIONS-NAME:'BLANK' “ beginnen.

Die Einfahrmenge kann durch den LINK-Namen „EINDAT“ zugeordnet werden, ansonsten wird im Dialog nach dem Dateinamen gefragt.

Für jedes Produkt, das in das Depot übernommen wird, wird in der Protokolldatei eine Vergleichsliste erstellt.

Ausgabedatei (druckaufbereitet): EINFAHR.LISTE.hhmmss

Dieses Protokoll enthält einen Überblick über die Differenz zur letzten eingefahrenen Menge des Produktes und der durchgeführten Notizbuchabgleiche. Sollte die neue Menge identisch sein, besteht im Dialog die Möglichkeit, über die Aufnahme dieser Menge zu entscheiden.

12.6.2 Laderdienste

RMS bietet eine abgeschlossene Laderdienst-Funktion zur Pflege aller Komponenten eines Anlagenladers.

Das zentrale Element ist eine Anlagenladerdefinition; hier werden alle Einstellungen gespeichert, die notwendig sind, um eine Anlagenladerdatei zu erstellen bzw. zu installieren.

Inhalt einer Anlagenladerdefinition

- Welche Produkte dazugehören.
  Im Subsystem-Lader ist normalerweise nur ein Produkt vorhanden, in einem so genannten „BS2000-Lader“ können mehrere Produkte zusammengemischt und nach REP-Klasse sortiert werden.
- Wie die Installationsmerkmale einzustellen sind.
  Eine Anlagenladerdatei wird unter dem in der Anlagenladerdefinition definierten Pfadnamen installiert und bekommt die dort definierten Inhalte und Einstellungen.
- Ob und welche OPT-Rep-Pakete zu benutzen sind.
  Ein OPT-Rep-Paket ist eine Auswahl von Korrekturen, die bei jeder Lader-Generierung berücksichtigt werden sollen. Ist eine Korrektur nicht in der ausgewählten LB-Version vorhanden, führt dies *nicht* zu einem Fehler.
- Ob und welche Lader-Modifikations-Pakete zu benutzen sind.
  Ein Lader-Modifikations-Paket ist eine Auswahl von Korrekturen, die bei jeder Lader-Generierung, unabhängig von ihrem Status, berücksichtigt werden sollen. Hiermit ist eine anlagen- bzw. aufgabenspezifische Einstellung möglich - z.B.
  hardwarespezifische Korrekturen, die immer in der Anlagenladerdatei vorhanden sein müssen.

Umfang und Inhalt einer neuen Anlagenladerdefinition festlegen

Der Definitionsvorgang besteht aus 7 Schritten:

1. Laderdefinitionsname eingeben und „Administration“ markieren
2. SW-Konfiguration zuordnen (wahlweise)
3. Produktauswahl (Pflicht)
4. Ladermodifikationspaket(e) zuordnen (wahlweise)
5. OPT-REP-Einstellung(en) zuordnen (wahlweise)
6. Installationseigenschaften anpassen (wahlweise)
7. Bestätigung der Einstellungen und Speichern der Anlagenladerdefinition im Depot

Im Folgenden werden die Schritte 2 bis 7 näher erläutert.

SW-Konfiguration zuordnen

Eine Übersicht zeigt alle SW-Konfigurationen, die in diesem Depot enthalten sind.
Eine SW-Konfiguration kann ausgewählt werden um die Produktauswahl zu vereinfachen oder um die Standard-Einstellungen der Lader zu übernehmen.

Produktauswahl

Wenn keine SW-Konfiguration ausgewählt wurde, werden alle im Depot gespeicherte Produkte zur Auswahl angeboten. Jede beliebige Zusammenstellung ist möglich.

Wenn die Zusammenstellung keiner SW-Konfiguration entspricht, kann *nie* eine Anlagenladerdatei erstellt werden!

Die gewünschten Produkte bzw. LB-Versionen sind zu markieren. Wird nur das Produkt ausgewählt, wird bei jeder Anlagenladergenerierung immer die aktuelle LB Menge verwendet.

Ladermodifikationspakete zuordnen

Übersicht aller im Depot befindlichen Ladermodifikationen, aus denen eine Auswahl getroffen werden kann.

Die aktuellen Einstellungen in den Ladermodifikationspaketen werden bei jeder Anlagenladergenerierung berücksichtigt.

OPT-REP-Einstellung zuordnen

Übersicht aller im Depot befindlichen OPT-REP-Einstellungen, aus denen eine Auswahl getroffen werden kann.

Die aktuellen Einstellungen in den OPT-Rep-Paketen werden bei jeder Anlagenladergenerierung berücksichtigt.

Installationseigenschaften anpassen

Angezeigt werden

- Die schon angepassten Eigenschaften aus einer existierenden Anlagenladerdefinition
- Eine leere Einstellung, falls keine SW-Konfiguration zugeordnet wurde
- Die Standardeinstellungen für die zugeordnete SW-Konfiguration.

Diese können speziell für diese Anlagenladerdefinition verändert werden. Eine Beschreibung der Installationseigenschaften finden Sie in der nachstehenden Tabelle.

Beschreibung der Installationseigenschaften:

Eigenschaft	Wert	Bemerkung
Passwort		Die Eigenschaften können durch ein Passwort vor dem Überschreiben geschützt werden. Das Passwort besteht aus max. 8 Zeichen. 8 x Blank wird ignoriert. Um ein Passwort zu löschen, ist *NONE einzugeben. Falls eine Passwort vergessen wurde, kann unter TSOS das Passwort zurückgesetzt werden.
Share	ja nein	Definieren der Zugriffsberechtigung.
Sicherung	ja nein	Sicherung der Anlagenladerdatei, die bei einer Installation überschrieben wird. Anlagenladerdateien im BS2LOADER-Format werden immer gesichert.
Installation	ja nein	Es soll eine Installation unter dem Installationspfadnamen durchgeführt werden.
Information	Max	Folgende Informationen werden bei Einsatz des Anlagenladers auf der Konsole ausgegeben: Name des Anlagenladers und die Generierungsversion der im Anlagenlader enthaltenen Produkte und dessen verwendete LB-Version, alle Korrekturnummern für Notizbucheinträge, die bei der Erstellung aktiv waren, die durchgeführten Anlagenlader-Modifikationenalle aktivierten OPT-Reps.
Information	Med	Folgende Information wird bei Einsatz des Anlagenladers auf der Konsole ausgegeben: Name des Anlagenladers und die Generierungsversion, im Anlagenlader enthaltene Produkte und dessen verwendete LB-Version.
Information	Min	Folgende Information wird bei Einsatz des Anlagenladers auf der Konsole ausgegeben: Name des Anlagenladers und die Generierungsversion.
Funktions-Name	ja nein	Zu den Subnos der aktivierten OPT-Reps werden die Funktionsnamen ausgegeben. Nur die PM-Nr s werden ausgegeben.
PUBLIC oder PRIVATE		Die Ausgabedatei kann durch die Angabe von Volume und Device auf PRIVAT erfolgen.
Installationspfadname		Kann geändert werden, eine Benutzerkennung *OWN bedeutet, dass immer die Benutzerkennung des Aufrufenden benutzt wird.Durch Eingabe von *STD im Dateinamen-Feld werden die mit dem letzten LB übernommenen Werte eingesetzt.
LOADER-FORMAT	BS2000 SUB-SYSTEM	BS2000-Format wird beim Systemladen benötigt, Subsystem wird von DSSM bearbeitet.

Bestätigung und Speichern

Die Einstellungen für die Anlagenladerdefinition sind nun vollständig durchgeführt aber noch nicht im Depot gespeichert. Folgende Möglichkeiten werden angeboten:

DUE	Es wird einer Übersicht der durchgeführten Einstellungen zur weiteren Entscheidung ausgegeben
F1	Änderungsvorgang wiederholen
F2	Eine Speicherung erfolgt
F3	Eine Speicherung erfolgt und anschließend wird eine Überprüfung der Anlagenlader durchgeführt
K1	Der Vorgang wird abgebrochen

Ladererstellung und Installation

Bei der Anlagenladergenerierung wird eine einsatzfähige Laderdatei erzeugt. Die Installation katalogisiert diese Laderdatei unter dem in der Anlagenladerdefinition vorgegebenen Dateinamen und ergänzt die Datei mit Informationen über ihre Vorgänger.

Die für den Ablauf benötigten Angaben wie Inhalt, Umfang und Installationseigenschaften sind in der Anlagenladerdefinition enthalten.

Bei der Auswahl der zu bauenden Anlagenladerdefinitionen kann bestimmt werden, ob:

1. Alle Anlagenlader installiert werden, unabhängig von deren Definition.
2. Kein Anlagenlader installiert werden soll, auch unabhängig von deren Definition
3. Die Anlagenlader generiert bzw. installiert werden sollen, abhängig von den Einstellungen in deren Definitionen.

Ein weitere Möglichkeit ist die Angabe einer alternativen Katalogkennung und/oder Benutzerkennung, ggf. mit einem Präfix. Die Auswirkungen sind:

1. Der Pfadname der zu erzeugende bzw. zu installierenden Laderdatei wird dementsprechend modifiziert.
2. Der Parameter „Sicherung“ wird ignoriert, falls
   - Ein Präfix angegeben ist oder
   - Eine Benutzerkennung angegeben ist und sie ungleich der Benutzerkennung in der Anlagenladerdefinition ist oder
   - *OWN als Benutzerkennung in der Anlagenladerdefinition eingetragen ist und die angegebene Benutzerkennung ungleich der Benutzerkennung des Aufrufers ist.

Anlagenladergenerierung

Eine installationsfähige Laderdatei mit folgendem Dateinamen wird unter der aufrufenden Benutzerkennung erstellt: `laderdefname.laderversion`

Anlagenladerinstallation

Bei einer Anlagenladerinstallation wird der ausgewählte Anlagenlader erstellt und unter dem in der Anlagenladerdefinition enthaltenen Installationsnamen katalogisiert.

Dieser Installationsname entspricht, wenn er nicht manuell verändert wurde, dem Laderdateinamen, den das jeweilige Produkt benötigt. Ist bereits eine Datei gleichen Namens im System vorhanden, wird diese Datei nur überschrieben, wenn es eine SAM-Datei ist und es sich um eine Laderdatei mit gleichem Produkt und Produktversion handelt. Bei einem Versionswechsel ist eine solche Überprüfung unerwünscht und kann durch Einschalten einer RMS-Option unterdrückt werden. Im Dialog geschieht dies durch Markieren von OFF im Punkt „Letzte Produkt/Version in Laderdatei pruefen“ in der Funktionsmaske RMS-Optionen einstellen. Im Batch geschieht dies durch NO-PRODUCT-CHECK=YES in der Anweisung CREATE-RMS-OPTIONS.

Bei BS2000-Systemladerdateien und Anlagenlader, für die in der Anlagenladerdefinition *Sicherung=ja* gesetzt ist, wird eine von RMS erstellte zu überschreibende Datei immer unter dem Generierungsnamen gesichert. Andere Dateien werden immer unter deren Dateinamen plus dem Suffix `.RMS.ttmmjj.hhmmss` gesichert. Sollte diese Konstruktion zu lang für BS2000 sein, wird der Standarddateiname `LADER.SICHERUNG.RMS.ttmmjj.hhmmss` benutzt. Dieser Dateiname wird in die neue Repfile als Kommentar mit aufgenommen und beim Startup (Repverarbeitung) ausgegeben, um im Notfall mit der alten Repfile neu starten zu können.

Die Laderdateien werden standardmäßig mit vordefinierten Dateischutzkriterien angelegt.

Wiederholen von Anlagenladergenerierungen

1. Auf der Ladergenerierungsmaske muss das Feld `Anlagenlader` markiert werden.
2. Die gewünschte Auswahl ist durch Markieren zu treffen.
   Zusätzlich ist das `Feld Alte Version` zu markieren.
3. Anschließend wird die Übersicht aller Anlagenlader mit Erstellungsdatum ausgegeben, die mit diesem Namen generiert wurden. Nach dem Markieren der gewünschten Laderversion wird der gleiche Anlagenlader erzeugt wie bei der Erstgenerierung.

OPT-Rep Einstellung

Hier können diejenigen auswählbaren Korrekturen zusammengestellt werden, die beim Anlagenladerbau berücksichtigt werden sollen. Eine auswählbare Korrektur kann in einer der folgenden Klassen eingruppiert werden:

- Typ: O oder U (eine Erweiterung der BS2000-Funktionalität)
- Typ: O oder U (die Wiederherstellung eines alten Verhaltens)
- Typ: T (schaltet eine Überwachung (Trace) ein oder aus)
- Typ: D (erzeugt Unterlagen, um ein Problem näher analysieren zu können)

Um Probleme beim Versionswechsel zu vermeiden, wird die Auswahl der optionalen (funktionalen) Korrekturen immer „familienbezogen“ durchgeführt, d.h. BS2000-GA wird angeboten, aber ohne Versionsbezeichnung.

Nur die Korrekturen, die in der neuesten LB-Version für jedes Mitglied eine Familie vorkommen, werden angeboten.

Ladermodifikationen

Ein Ladermodifikationspaket bietet den Möglichkeit, Anlagenlader-spezifische Aktionen zu definieren, unabhängig vom REP-Typ. Ein Beispiel wäre ein Hardware-abhängiges REP, das nur bei bestimmten Anlagen eingesetzt werden darf.

Änderungsmöglichkeiten

Elemente der Anlagenladerfunktionalität können geändert, dupliziert, umbenannt und gelöscht werden.

- Alles ändern (Standard)
  Alle Einstellungen bzw. Verknüpfungen, die im Depot unter dem Namen abgespeichert sind, können geändert werden.
- Eigenschaften ändern
  Dieses Feld wird nur für Anlagenladerdefinitionen bzw. SW-Konfigurationen ausgegeben. Wird dieses Feld markiert, werden nur die Einstellungen der Eigenschaften zum Bearbeiten angeboten.
- Löschen
  Durch Markieren und Abschicken mit der F1-Taste wird der ausgewählte Namen gelöscht. Handelt es sich um die Verwaltungsgruppe OPT-Rep Einstellung oder Anlagenlader-Modifikation und wurden diese schon mit einem Anlagenlader verknüpft, erfolgt bei allen Anlagenladerdefinitionen, in dem dieser Namen vorkommt, der Eintrag DEL.
- Umbenennen
  Der ausgewählte Namen kann anders benannt werden. Dadurch werden alle Verknüpfungen mit dem neuen Namen versorgt. Der alte Namen steht nicht mehr zur Verfügung.
- Kopieren
  Der ausgewählte Namen kann unter einem neuen Namen kopiert werden. Dieser steht dann für weitere Bearbeitungen zur Verfügung. Der alte Namen wird nicht verändert. Ein Umbenennen oder Kopieren kann nur auf einen im Depot für die
  Verwaltungsgruppe noch nicht vorhandenen Namen erfolgen.
- Passwort
  Dieses Feld wird nur ausgegeben, wenn der ausgewählte Namen mit einem Passwort geschützt ist. Für weitere Aktionen muss das Passwort eingegeben werden.

Software-Konfiguration

Eine Software-Konfiguration beinhaltet die Standardzuordnung von Produkten zur Installationsdatei (Anlagenlader). Dadurch werden fehlerhafte Mischvorgänge verhindert und evtl. Abhängigkeiten aufgezeigt.

Die produktspezifische Software-Konfiguration ist ein fester Bestandteil der Lieferdatei. Bei jeder Übernahme einer Lieferung werden die im Depot befindlichen Konfigurationen automatisch aktualisiert.

Ändern Installationseigenschaften

> ¡ Die Einstellungen, außer Dateiname, werden beim Transfer (Liefern) einer Repmenge an die nächste Anwenderstufe weitergegeben.

Die vom Anwender geänderten Werte werden beim nächsten Übernahmevorgang (Einfahren) der SW-Konfiguration nicht verändert. Eine Beschreibung der Installationseigenschaften finden Sie in der Tabelle 15 im Abschnitt „Laderdienste“.

Der Anwender kann für jede SW-Konfiguration die mitgelieferten Installationseigenschaften mit der Funktion Administration SW-Konfiguration abändern und damit eigene Standards festlegen.

12.6.3 Korrekturverwaltung - Notizbuch

Das Notizbuch dient zum Modifizieren des Depotinhaltes. Eine Modifikation ist die Änderung des Status einer Korrektur im Depot (Deaktivierung, Aktivierung, Eintragung neuer Korrektur).

Das Notizbuch enthält alle Abweichungen gegenüber den übernommenen Korrekturmengen und hat folgende Eigenschaften:

- Jede Aktion gilt solange, bis in einem neu übernommenen LB die angegebene Aktion realisiert ist oder eine weitere Notizbuchaktion zu dieser Korrektur erfolgt.
- Alle Aktionen gelten immer produktversionsspezifisch.
- Jede Aktion wird dokumentiert.
- Alle Einträge werden bei jedem Zugriff berücksichtigt (Ausnahme: man verlangt bewusst Originalmengen).

Eine Notizbuchsitzung beginnt mit der Eintragung der Benutzeridentifikation und wird mit K1 beendet. Die Benutzeridentifikation wird für jede Modifikation eingetragen.

> ¡ Bei hardware-spezifischen Korrekturen werden Identifikationen benutzt, bei denen der erste PULS bekannte Buchstabe zu S (bei /390-Servern) oder K (bei x86-Servern) wird. Wenn eine PULS-Identifikation angegeben ist und die ursprüngliche Identifikation nicht auffindbar ist, sucht RMS bei folgenden Aktionen nach einer entsprechenden hardware-spezifischen Version:
>
> - Übernahme von Vorab-Korrekturen (ADD mit Angabe einer Datei)
> - Aktivieren Depot-Korrekturen (ADD)
> - Korrekturen ungültig setzen (REJ)

Alle Notizbuchaktionen können im Dialog und auf Datei ausgegeben werden.

Die Bildschirmmaske ist bei allen Funktionen gleich. Im unteren Teil der Maske werden nach einer internen Konsistenzprüfung Hinweise für mögliche Aktionen angeboten. Erst nach Bestätigung einer vorgeschlagenen Aktion wird der Eintrag übernommen.

Die Korrekturen im Depot können in folgende Weise klassifiziert werden:

- Quelle der Korrektur
  - Lieferweg (SOLIS2 oder PULS)
  - ADD-Aktion im Notizbuch.
- Art der Korrektur
  - Normale Korrektur
  - Korrektur für Selectable-Unit
    - Optionale Korrektur
    - Vorab Korrektur
    - Korrektur zu Erstellung von Diagnose-Unterlagen
    - Korrektur um Verfolgungs (Trace)- oder Überwachungsroutinen ein- bzw. auszuschalten

- Korrekturstatus in RMS
  - Normal - d.h. es wurden kein Notizbuchaktionen für diese Korrektur vorgenommen.
  - ADD mit Zuordnung zu einer Teil die LBs - Teil-ADD
  - ADD mit Zuordnung zu alle LBs - Voll-ADD
  - REJ (zurückgewiesen).

Eine Notizbuchaktion ist von dem Korrekturstatus und ggf. der Quelle der Korrektur abhängig, aber nicht von der Art der Korrektur.

Die möglichen Notizbuchaktionen sind in der folgenden Übersicht aufgeführt und werden im weiteren Verlauf näher erklärt:

Aktion	Beschreibung
ADD/CMD /SIS aus Datei	Eine neue Korrektur ist bekannt gemacht worden (über FM2000, SIS oder ein Telefonat), und es scheint sinnvoll, diese Korrektur zu benutzen und ggf. per Lieferung weiterzugeben.
ADD aus Depot	Es wurde festgestellt, dass es keine aktuelle Korrektur zu diesem Problem gibt, aber dass eine frühere Korrektur gut genug ist, um eingesetzt zu werden.
REJ	Es wurde festgestellt dass eine Korrektur fehlerhaft ist und nicht mehr zur Erstellung von Lieferungen bzw. Anlagenladerdateien hinzugezogen werden soll. Diese Aktion bleibt aktuell, bis die Korrektur in einer zukünftigen Lieferung fehlt.
RST	Es wurde irrtümlicherweise Notizbuch Aktionen für eine SUBNO durchgeführt. Mit diese Aktion werden *alle* Aktionen rückgängig gemacht.

Tabelle 16: Übersicht über die möglichen Notizbuchaktionen

Übernahme von Vorab-Korrekturen - ADD mit Angabe einer Datei

Durch eine ADD-Aktion kann eine Korrektur aus einer Datei ausgewählt, in das Depot übernommen und einem Produkt zugeordnet werden.

Diese Korrektur wird allen Lieferbestandteilen des ausgewählten Produktes zugeordnet und bleibt solange gültig, bis eine neue Korrektur zu dieser SUBNO übernommen bzw. eine erneute Notizbuchaktion (ADD, REJ, CMD, SIS oder RST) durchgeführt wird. Wenn für eine SUBNO und ein Produkt bereits eine Korrektur im Depot vorhanden ist, wird diese als ungültig gekennzeichnet (implizite REJ-Aktion).

Pflichtangaben sind:

```
subno jd 'ADD' kommentar produkt version dateiname
```

Übernahme von mehreren Vorab-Korrekturen - CMD mit Angabe einer Datei

Bei der CMD-Aktion werden alle Korrekturen aus der angegebenen Datei als einzelne ADD-Aktionen zur Übernahme in das Depot für das angegebene Produkt angeboten.

Soll eine angebotene Korrektur in das Depot übernommen werden, muss ein Kommentar eingegeben werden, anschließend DUE. Soll eine Korrektur nicht übernommen werden, nur DUE.

Pflichtangaben sind:

```
'CMD' produkt version dateiname
```

Automatische Übernahme von Aktionen aus SIS - SIS mit Angabe einer Datei

Bei der SIS-Aktion werden alle Korrekturen aus der angegebenen Datei als einzelne ADD- bzw. REJ-Aktionen zur Übernahme in das Depot für alle in der Datei enthaltenen Produkte automatisch abgearbeitet.

Diese Aktion wird allen Lieferbestandteilen des ausgewählten Produktes zugeordnet und bleibt solange gültig, bis eine neue Korrektur zu dieser SUBNO übernommen bzw. eine erneute Notizbuchaktion (ADD, REJ, CMD, SIS oder RST) durchgeführt wird.

Pflichtangaben (nur durch die Anwendergruppe 2 zu machen) sind:

```
'SIS' dateiname
```

Aktivieren Depot-Korrekturen - ADD

Durch eine ADD-Aktion ohne Dateiangabe wird die angegebene Depotkorrektur für alle LB des angegebenen Produktes aktiviert.

Pflichtangaben sind:

```
subno jd 'ADD' kommentar produkt version
```

Korrekturen ungültig setzen - REJ

Mit der REJ-Aktion wird eine Korrektur zurückgewiesen. Dadurch wird sichergestellt, dass diese Korrektur bei jeder weiteren Lieferung bzw. Erstellen eines Anlagenladers nicht mehr enthalten ist.

Pflichtangaben sind:

```
subno jd 'REJ' kommentar
```

Wahlweise können Produkt und Version angegeben werden. Unabhängig davon werden alle Produkte ausgegeben, in denen sich diese Korrektur befindet. Dadurch besteht die Möglichkeit, den REJ nur für das angegebene oder für alle Produkte durchzuführen.

Sollte diese Korrektur noch nicht im Depot sein, kann die Aktion vorgemerkt werden. In so einem Falle wird die Aktion erst durchgeführt, wenn diese Korrektur in einer Liefermenge vorhanden ist.

Diese Aktion bleibt solange gültig, bis eine erneute Notizbuchaktion (ADD, CMD, SIS oder RST) durchgeführt wird.

Zurücksetzen von Notizbuchaktionen - RST

Die RST-Aktion setzt alle Notizbucheinträge für eine SUBNO zurück. Damit wird wieder der Originalzustand (Einfahrmenge) im Depot für diese Korrektur hergestellt.

Pflichtangaben sind:

```
subno 'RST' kommentar
```

Wahlweise können Produkt und Version angegeben werden. Unabhängig davon werden alle Produkte ausgegeben, in denen sich diese Korrektur befindet. Dadurch besteht die Möglichkeit, den RST nur für das angegebene oder für alle Produkte durchzuführen.

12.6.4 Informationsdienste

Die Informationsdienste bestehen aus folgende unabhängigen Funktionen

- Korrektur - Dialoginformation
- Korrektur - Gesamtübersicht
- Notizbuch - Dialoginformation
- Notizbuch - Gesamtübersicht
- Vergleich von REP-Mengen
- Zeichenorientiertes Suchen

12.6.4.1 Korrektur - Dialoginformation

Suche nach Korrekturnummer (PM-Nr)

Die zu suchende PM-Nummer (Subno) ist in der Maske einzutragen. Um den Suchvorgang einzuschränken, kann eine Produktfamilie bzw. eine Produktversion eingetragen werden.

Als Ergebnis wird eine Übersicht ausgegeben. Durch Markieren der einzelnen Felder stehen weitere Informationsmöglichkeiten zur Verfügung:

Korrekturzuordnung	Durch Markierung einer Produktversion wird die Korrekturübersicht ausgegeben, die die Originalzuordnung „*“ der JD zur jeweiligen eingefahrenen LB-Version zeigt.
Korrekturinhalt und -texte	Durch Markieren des JD wird der Korrekturinhalt mit evtl. Notizbucheinträgen oder ausgewählten Texten ausgegeben.

Produkt LB-Information

Durch Eingabe einer der folgenden Werte im Feld „Produkt“ und/oder „Version“ wird eine Produkttransferinformationen ausgegeben.

*	Alle im Depot gespeicherten Produkte werden (alphabetisch) angezeigt. Die angezeigte Information ist die eingefahrene Lieferung, das Solis-Datum, das Einfahr-Datum und der Lieferbestandteil.
?	Eine Übersicht aller im Depot gespeicherten Produkte wird angeboten und durch Markieren kann eine Auswahl getroffen werden, um Informationen über spezifische Produkte auszugeben.
Produkt Name	Transferinformation für alle Versionen dieses Produktes wird ausgegeben.
Nur Version	Transferinformationen für alle Produkte mit dieser Version werden ausgegeben
Produkt und Version	Transferinformation für dieses Produkt mit dieser Version wird ausgegeben.

Als Erstes wird die Übernahmeinformation angezeigt, - d.h. alle LB-Versionen werden zusammen mit deren Erstellungs- und Übernahmedatum aufgelistet.

12.6.4.2 Korrektur - Gesamtübersicht

In der Übersichtsliste werden alle Korrekturnummern für ein oder mehrere Produkte und deren Zugehörigkeit zu den einzelnen LB-Versionen aufgelistet. Es werden bis maximal 16 LB-Versionen aufgelistet.

Ein Protokoll aller eingefahrenen Lieferungen mit dem dazu gehörigen SOLIS2-Datum wird am Anfang der Protokoll-Datei geschrieben.

Beschreibung der markierbaren Felder

„Einfahr-Korrekturmenge“
„Liefer-Korrekturmenge“

> Durch Markieren eines dieser Felder kann der einfahr- bzw. lieferspezifische Korrekturstand aufgelistet werden. Wird keines der Felder markiert, wird der aktuelle Korrekturstand aufgelistet, d.h. Einfahrmenge plus Notizbuchaktionen.

„eine auszuwählende Menge“

> Wird dieses Feld markiert, besteht die Möglichkeit, den Produktumfang der Liste zu bestimmen.

„Auswahl der ersten LB-Version der Liste“

> Wird *ja* markiert, besteht auf dem Folgeschirm die Möglichkeit, die erste LB-Version der Auswertung zu bestimmen.

„Listing“

> Durch Markieren des Feldes „Summary“ wird eine Liste erstellt mit einer Zusammenfassung, wie viel NEU, CNG oder DEL Korrekturen in jeder LB-Version gezählt wurden.

„Eine modulspezifische Untermenge“

> Es besteht die Möglichkeit, die Liste nur für bestimmte Module bzw. Module plus Displacement zu erstellen. Auf einer Folgemaske kann das gewünschte Modul ggf. plus Displacement angegeben werden.

12.6.4.3 Notizbuch - Dialoginformation

Die Vorgehensweise (Suchkriterien) bei dieser Funktion ist die gleiche wie bei der Suche nach Korrekturnummer.

Als Ausgabe erfolgen die gewünschten Notizbucheinträge:

- Produktspezifische Information

  Es sind das Feld „Datei“ und die gewünschten Produkte zu markieren.

  Ausgabedatei: `NBK.ALLE.EINTRAEGE.produkt.vers`
- JD-spezifische Information

  Es sind das Feld „Datei“ und das gewünschte *jd* zu markieren.

  Ausgabedatei: `NBK.subno.jd.produkt.vers`

12.6.4.4 Notizbuch - Gesamtübersicht

Diese Liste enthält alle Einfahr- und Notizbuchaktionen in der Reihenfolge ihrer Durchführung. Ausgabedatei (druckaufbereitet): `#NOTBUCH.LIST.hhmmss`. Verwendete Begriffe:

VORMERK	Die REJ-Aktion kann noch nicht durchgeführt werden, da die Korrektur noch nicht im Depot ist. Der Eintrag wird erst berücksichtigt, wenn diese Korrektur in einem neu übernommenen LB vorhanden ist.
AKTIV	Die angegebene Korrektur ist im Depot vorhanden. Diese Aktion wird solange berücksichtigt, bis sie in einem neu übernommenen LB realisiert ist, oder durch eigene Notizbuchaktionen abgeändert wurde.
ERLEDIGT	Der Eintrag ist durch eine neue Notizbuchaktion oder einem neu übernommenen LB erledigt worden. Damit hat dieser Eintrag keine Wirkung auf neu angelieferte LB.

12.6.4.5 Vergleich von Rep-Mengen

Mit dieser Funktion können zwei Rep-Mengen verglichen werden. Dabei werden alle neuen, fehlenden und fachlich bzw. formal geänderten Korrekturen ermittelt. Die Daten werden druckaufbereitet in eine temporäre Datei geschrieben.

Ausgabedatei (druckaufbereitet): `#hh-mm-ss.LSTVGL`

Als Eingabemengen können die Folgenden benutzt werden:

- Anlagenladerdatei
- Sonstige REP-Dateien
- Eine im Depot übernommene LB-Version
- Eine im Depot übernommene LB-Version plus aller Notizbuchaktionen
- Eine aus dem Depot gelieferte LB-Version

Auf dem Bildschirm können die zu vergleichenden Rep-Mengen definiert werden. Zusätzlich kann durch Markieren der Felder „short Info“ und/oder „ohne Reps“ der Listenumfang bestimmt werden.

short Info	Gleiche Korrekturen werden in der Übersichtsliste nicht angezeigt.
ohne Reps	Der Inhalt von fehlenden, neuen oder geänderten Korrekturen wird nicht ausgegeben.

Bei jedem Vergleich wird eine Kurzinformation über die Anzahl der neuen (New), gelöschten (Del) und geänderten (Cng) Korrekturen auf dem Bildschirm ausgegeben.

12.6.4.6 Zeichenorientiertes Suchen

Mit dieser Funktion kann man im Depot nach einem oder mehreren Strings suchen. Dabei werden ausgewählte oder alle Produkte und/oder Textebenen durchsucht.

Es kann eine zeichen- oder eine modulorientierte Suche ausgewählt werden.

Pro Produkt oder Textebene wird eine Übersicht der Korrekturen oder Texteinträge ausgegeben, in denen der oder die angegebenen Suchstrings vorkommen. Durch Markieren der gewünschten Einträge werden die Detailinformationen (Korrekturen oder Text) ausgegeben.

Bei folgender (Standard) Einstellung wird das gesamte Depot durchsucht:

gleicher Umfang	aus
Produkte	ein
Texte	ein

Durch die Standardeinstellungen *Lower aus* und *Case ein* wird der eingegebene Suchstring sowie die internen Texte in Großbuchstaben umgesetzt und verglichen. Die Ausgabe erfolgt im Original.

Durch Markieren der folgenden Begriffe kann „aus“ auf „ein“ bzw. „ein“ auf „aus“ geschaltet werden. Dadurch kann die zu durchsuchende Menge bestimmt bzw. beeinflusst werden:

Umfang definieren	Produktauswahl-Schirm folgt und es kann durch Markieren von Produkten die Suche auf diese Produkte begrenzt werden. *gleicher Umfang* wird automatisch eingeschaltet.
gleicher Umfang ein	Die vorher getroffene Produktauswahl bleibt bestehen.
Texte ein	Textebenen werden durchsucht.
Produkte ein	Korrekturen und Notizbucheinträge werden durchsucht.
Lower ein	Eingegebener Suchstring wird nicht in Großbuchstaben umgesetzt.
Case ein	Kleinbuchstaben im Depot werden zur Suche in Großbuchstaben umgesetzt.

12.6.5 Textverarbeitung

Die Texte können eingetragen werden als

Global	PRODUKT ist leer, SUBNO ist leer, JD ist leer
Produkt-spezifisch	SUBNO ist leer, JD ist leer
SUBNO-spezifisch	JD ist leer
SUBNO-JD-spezifisch	Alles ist definiert.

Textebenen

Zugegriffen wird auf einzelne Textebenen-Schlüssel in den verschiedenen Textebenen. Zum Bearbeiten von Textebenen muss das Feld „Textebene“ markiert werden. Anschließend müssen auf dem Folgebildschirm die Textebenen ausgewählt werden, die bearbeitet werden sollen.

Produktnotizen

Diese Notizen dienen nur zum Aufmerksam machen. Bei der Übernahme eines neuen Lieferpakets werden diese bei Übereinstimmung des Produktes und ggf. SUBNO/JD im Einfahrprotokoll ausgegeben.

Zum Bearbeiten von Produktnotizen muss das Produkt markiert werden, für welches eine Notiz hinterlegt werden soll.

Texteditor

Der Texteditor dient zum Bearbeiten (eintragen, löschen und ändern von Texten) von Textebenen und Produktnotizen. Es wird zwischen SUBNO-orientierten und globalen Texten unterschieden.

Beispiel

> In der Textebene „SK-OPT BESCHREIBUNG“ soll der Text für die SUBNO 41000077 geändert werden. Zuerst wird das Feld „Textebene“ markiert, dann auf dem Folgebildschirm die Textebene „SK-OPT BESCHREIBUNG“ ausgewählt. Nun ist der Texteditor aktiviert. Nach Eingabe der SUBNO wird der Text zum Ändern angeboten.

Erläuterung des Editierbildschirm

Das Bildschirm ist dreigeteilt,

1. Ein Kontrollbereich mit den im Depot befindlichen Texten. Dieses Datenfenster kann nicht überschrieben werden.
2. Ein Änderungsbereich; in diesem Datenfenster wird der ursprüngliche Text zum Ändern angeboten.
3. Eine Markierungsspalte. Durch Eingeben bestimmter Zeichen in der Markierungsspalte können Zeilen des Arbeitsfensters für bestimmte Anwendungsfälle gekennzeichnet werden.

   d Die mit *d* markierten Sätze werden gelöscht.

   n Anzahl der einzufügenden Zeilen (1-9).
   Die Zeilen werden vor der mit *n* markierten Zeile eingefügt.

Aktionsfelder und Bedienungshinweise

Low OFF /ON	Low legt fest, ob der Texteditor eingegebene Kleinbuchstaben in Großbuchstaben umsetzt oder nicht. Durch Markieren des Feldes kann die Standardeinstellung OFF auf ON geschaltet werden. Beachten Sie: Im Gegensatz zu EDT wird die Low-Einstellung ausgewertet, bevor der Text bearbeitet wird. OFF: Der Editor setzt eingegebene Kleinbuchstaben in Großbuchstaben um. ON: Zeichen werden verarbeitet wie eingegeben.
Loesche (+F1)	Durch Markieren des Feldes und Abschicken mit der F1-Taste wird der Text für die angezeigte SUBNO aus dem Depot gelöscht.
F2	Der geänderte Text wird im Depot gespeichert.
F3	Die Änderung für die angezeigte SUBNO wird ignoriert.
K1	Der Texteditor wird beendet.

Nach dem Bearbeiten einer SUBNO wird die nächst höhere SUBNO zum Ändern angeboten.

12.6.6 RMS-Verwaltung

- Installation
- Startprozedur und Basisparameter
- Funktionsumfang definieren
- Beispiel einer Definitionsdatei
- DEPOT-Pflege

12.6.6.1 Installation

Zum Ablauf von RMS werden folgende Dateien benötigt:

`SYSPRC.RMS.071` bzw. wahlfreier Name	Prozedur zum Aufrufen des Programmes RMS. In dieser Startprozedur sind die Basisparameter für RMS zu vereinbaren.
`SYSOML.RMS.071` bzw. wahlfreier Name	Bindemodulbibliothek, die alle Module, die für den Programmablauf notwendig sind, enthält.
`SYSDAT.RMS.071` bzw. wahlfreier Name	Definiert Funktionsumfang und Zugriffsberechtigung. Zusätzlich besteht die Möglichkeit, den Lademodus der Module sowie den Aufbau und Textinhalt der Menübildschirme zu bestimmen.
`RMS.DEPOT` bzw. RMS.DEPOT.*suffix*	Standardmäßig greift RMS auf die Datei RMS.DEPOT zu. Mit der Angabe von *.suffix* ist es möglich, mehrere Depots unter eine Benutzerkennung zu halten. *.suffix* ist wahlfrei. Diese Art des Dateinamens kann durch Eingabe des Basisparameters 'FAM' in der Startprozedur verwendet werden. Ist die erwartete Datei nicht vorhanden, wird ein Depot mit dem definierten Namen eingerichtet.

12.6.6.2 Startprozedur und Basisparameter

Im Folgenden wird die Standard-Startprozedur erklärt, die zum Lieferumfang von RMS gehört. Diese kann nach eigenen Belangen modifiziert werden. Wichtig sind dabei die Basisparameter, mit denen RZ- und anwenderspezifische Funktionen eingestellt werden können.

Minimaler Umfang der Startprozedur

```
/PROC N
/     OPTION DUMP=YES
/     FILE SYSOML.RMS.071,LINK=RMSOML
/     SYSFILE SYSDTA=(SYSCMD)
/     EXEC (RMS,SYSOML.RMS.071)
TEMPF=#
/      ENDP
```

Maximaler Umfang der Startprozedur (wird mit RMS geliefert)

```
/PROC N,(&ID=TSOS,/
        &DEF=TSOS,/
        &DEPOT=TSOS,/
        &FAM=*NONE,/
        &K1OPT=JA,/
        &LCASE=JA,/
        &LMAX=60,/
        &MODE=DIALOG,/
        &V7SYNTAX=JA,/
        &LAN=D,/
        &PTASTEN=JA,/
        &TEMPID=#),/
        SUBDTA=&
/        REMARK ALL SETTINGS ARE RMS DEFAULT SETTINGS
/        OPTION DUMP=YES                                          1.
/        FILE $&DEF..SYSDAT.RMS.071,LINK=DEF                      2.
/        FILE $&ID..SYSOML.RMS.071,LINK=RMSOML                    3.
/        SYSFILE SYSDTA=(SYSCMD)
/        SKIP  .&MODE
/.DIALOG REMARK ** CALL RMS IN INTERACTIVE MODE (MENU-DRIVEN)
/.DIA    EXEC (RMS,$&ID..SYSOML.RMS.071)                          4.
MODE=DIALOG/&LAN,DEPOT-USER=&DEPOT,FAM=&FAM,K1OPT=&K1OPT,-        5.
LCASE=&LCASE,PTASTEN=&PTASTEN,TEMPF=&TEMPID,LMAX=&LMAX
/        ENDP
/.SOLIS  REMARK ** RMS ACTS LIKE SOLRMS **
/.BATCH  REMARK ** RMS IS STARTED UP IN BATCH MODE **
/        EXEC (RMS,$&ID..SYSOML.RMS.071),IDA=NO
MODE=&MODE/&LAN,DEPOT-USER=&DEPOT,FAM=&FAM,LCASE=&LCASE,
TEMPF=&TEMPID,LMAX=&LMAX,V7SYNTAX=&V7SYNTAX
/BREAK
/ENDP-RESUME
```

1. Bei abnormaler Beendigung von RMS wird ein Userdump erstellt
2. Zuweisen der Definitionsdatei, falls diese unter einer anderen Benutzerkennung steht. Diese Angabe ist optional. Wird kein FILE-Kommando angegeben oder die Datei unter dem Link-Namen nicht gefunden, greift RMS auf die lokale Datei SYSDAT.RMS.071 zu. Ist diese ebenfalls nicht vorhanden, gibt RMS eine Fehlermeldung aus.

3. Zuweisen der Modulbibliothek, aus der der Modul geladen werden soll.
4. Laden der Großmodule RMS aus der Modulbibliothek
5. Übergabe der zum Laden von RMS nötigen Basisparameter.

Beschreibung der Basisparameter

DEPOT-USER	Benutzerkennung, unter der das DEPOT katalogisiert ist, ggf. mit Katalogkennung. Wahlfreier Parameter, Standardeinstellung ist *OWN (aufrufende Benutzerkennung).
FAM	Es kann ein String definiert werden, der dem Standardnamen RMS.DEPOT angefügt wird. Dieser erweiterte DEPOT-Dateinamen wird dann verwendet. FAM=*NONE bedeutet Standardname RMS.DEPOT. Wahlfreier Parameter, Standardeinstellung ist *NONE
K1OPT	JA / YES Beim Beenden von RMS durch K1-Taste wird abgefragt, ob RMS wirklich beendet werden soll. NEIN / NO RMS wird ohne zusätzliche Abfrage beendet. Wahlfreier Parameter, Standardeinstellung ist JA
LCASE	JA / YES Druckausgabe Groß/Kleinschreibung NEIN / NO Druckausgabe nur Großschreibung Wahlfreier Parameter, Standardeinstellung ist JA.
LMAX	Bestimmt die Anzahl der Zeilen einer von RMS erzeugten Protokoll- bzw. Listenseite. Es ist ein Wert von 20-99 möglich. Wahlfreier Parameter, Standardeinstellung ist 60
MODE	BATCH Batchbetrieb wird simuliert. RMS liest die Eingaben mit RDATA von SYSDTA. COMMAND Batchbetrieb wird simuliert. RMS liest die Eingaben mit RDATA von SYSDTA. DIALOG RMS wird als maskenorientiertes Dialogprogramm aufgerufen SOLIS Batchbetrieb wird simuliert. RMS liest die Eingaben mit RDATA von SYSDTA. Wenn RMS mit MODE=SOLIS aufgerufen wurde, wird das Erzeugen der Laderdateien bei der FUNKTION=LADERBAU (SELECT-REPFILE-TO-BUILD) unterdrückt. Das Erzeugen dieser Laderdateien erfolgt durch Aufruf der FUNKTION=BAULADER (BUILD-REPFILE).

	IMON Wie bei SOLIS. Bei der Funktion SELECT-REPFILE-TO-BUILD können die Operanden RUNAME, RUVERS und SCI zusätzlich angegeben werden. Damit wird der IMON-Makro SETINSP für den generierten Lader unterstützt (siehe Handbuch „IMON“ [15]). TEST Batchbetrieb wird simuliert aber die Anweisungen werden nur überprüft und nicht durchgeführt. Dieser Parameter ist nur in Verbindung mit V7SYNTAX=YES zu verwenden. Durch Angabe von /D bzw. /E nach dem MODE-Parameter werden alle Texte in deutscher oder englischer Sprache ausgegeben. Wahlfreier Parameter, Standardeinstellung DIALOG/D
PTASTEN PKEYS	JA / YES die P-Tasten werden geladen NEIN / NO die P-Tasten werden nicht geladen. Wahlfreier Parameter, Standardeinstellung NEIN
TEMPF	Zeichen für temporäre Dateien. Zugelassen sind die vom Betriebssystem unterstützten Sonderzeichen für temporäre Dateien. Wahlfreier Parameter, Standardeinstellung ist #.
V7 SYNTAX	JA / YES alle Anweisungen werden nach der V7-Syntaxregeln geprüft und ggf. durchgeführt. NEIN / NO die aus Version 6 stammende Syntaxregeln werden benutzt. Wahlfreier Parameter, Standardeinstellung ist JA

12.6.6.3 Funktionsumfang definieren

Die Funktionsdatei wird zum Laden von RMS benötigt. Sie kann mit einem FILE-Kommando (LINK=DEF) in der Startprozedur definiert werden. Erfolgt keine Zuweisung, wird der Standarddateiname SYSDAT.RMS.071 verwendet. Ein Beispiel einer Definitionsdatei finden Sie im Abschnitt „Beispiel einer Definitionsdatei“.

Folgende Funktionsanweisungen stehen zur Verfügung:

MODE MEISTER	Mit dieser Anweisung wird die Schreibberechtigung für das Depot definiert.
LANGUAGE	Sprachauswahl für die Texte - nur wirksam bei fehlende Sprachkennzeichen beim MODE-Parameter.
MENU	Der Inhalt und die Anzahl der Menübildschirme kann den Anwenderbelangen angepasst werden.
MOD	Definieren des Gesamt-Funktionsumfanges von RMS. Zusätzlich kann der Lademodus der Funktionsmodule und deren Titel für die Menübildschirme definiert werden.
ENDMENU	Endanweisung für Menü Anweisung.
kommentar	Für Dokumentationszwecke.

Beschreibung der Anweisungen

MODE MEISTER

Diese Anweisung erlaubt alle mit MOD definierten Funktionen auszuführen und muss immer die erste Anweisung in der Funktionsdatei sein. Bei fehlender Anweisung kann nur lesend auf das Depot zugegriffen werden, unabhängig vom definierten Funktionsumfang.

LANGUAGE=DEUTSCH oder ENGLISH

Die Texte werden in deutscher oder englischer Sprache ausgegeben.

Dieser Anweisung wird nur dann ausgewertet, wenn in der Startprozedur keine Sprache im MODE-Parameter definiert ist.

MENU,TEXT='menu-titel max. 48 Zeichen'

Der 'menue-titel' erscheint als Titelzeile des Menü-Bildschirmes oder als Funktionstitel mit der Kennzeichnung als Sub-Menü. Ein Sub-Menü ist immer ein eigener Menü-Bildschirm.

Diese Anweisung muss vor der ersten MOD Anweisung stehen.

MOD=modulname[,TEMPORARY / RESIDENT],TEXT='funktionsbeschreibung'

Der `modulname` ist der Entry/Modulname der Funktion und muss groß geschrieben werden.

Die `funktionsbezeichnung` erscheint auf dem Menü-Schirm als Beschreibung für diese Funktion.

Man hat die Möglichkeit zu bestimmen, wann ein Modul geladen wird:

- `TEMPORARY`: Das Modul wird erst bei Aufruf der Funktion geladen und nach Beendigung wieder entladen.
- `RESIDENT`: Das Modul wird beim Starten von RMS geladen und bleibt über die ganze RMS-Session resident.

Ist keine der beiden Angaben vorhanden, wird das Funktionsmodul beim ersten Aufruf geladen und bleibt über die ganze Session resident.

Der bei der TEXT Anweisung stehende Text darf bis zu 48 Zeichen lang sein und erscheint auf dem Menü-Schirm als Überschrift für diese Funktion.

Liste der Modulnamen siehe Anhang A.

ENDMENU

Endekriterium für ein Menü (Menü-Bildschirm). Steht vor einer ENDMENU Anweisung erneut eine MENU Anweisung, werden alle dazwischenliegenden MOD Anweisungen (Funktionen) in einem untergeordneten Menü (Sub-Menü) angezeigt.

kommentar

Ist das erste Zeichen eines Satzes ein '*', wird der Satz als Kommentar erkannt und überlesen. Der Kommentar darf maximal 99 Zeichen lang sein.

12.6.6.4 Beispiel einer Definitionsdatei

Dieses Beispiel zeigt den Aufbau einer Definitionsdatei mit maximaler Funktionalität. Es zeigt unter anderem die Bedeutung der MENU- und MOD Anweisungen und anschließend deren Auswirkung auf die Menü- bzw. Sub-Menü-Bildschirme.

Mehrfachdefinitionen (gleicher Modul) ist möglich. Siehe in diesem Beispiel EDT. Dadurch erscheint diese Funktion auf allen drei Menü-Schirmen.

EDT und LMS werden von RMS nicht benötigt. Sie können aber definiert werden, um evtl. Programmwechsel zu vermeiden.

```
MODE MEISTER
LANGUAGE=DEUTSCH
MENU,TEXT='RMS - Produktions System'
   MOD=INFODATA,TEXT='DIALOG-INFORMATION'
   MOD=NOTIZERF,TEMPORARY,TEXT='KORREKTURVERWALTUNG - NOTIZBUCH'
   MOD=RMSLGEN,TEXT='LADERERSTELLUNG und INSTALLATION'
   MOD=RMSLVWIN,TEXT='LADERVERWALTUNG und INFORMATION'
   MOD=EINFAHR,TEMPORARY,TEXT='LIEFERMENGEN UEBERNEHMEN'
   MOD=LIEFERN,TEXT='LIEFERMENGEN ERSTELLEN' MOD=CALLEDT,TEXT='EDT'
   MENU,TEXT='INFORMATIONSDIENSTE und LISTEN'
      MOD=INFODATA,RESIDENT,TEXT='DIALOG-INFORMATION'
      MOD=RMSFIND,TEXT='ZEICHENORIENTIERTES SUCHEN'
      MOD=NBKINFO,TEXT='NOTIZBUCH-INFORMATION (produktspezifisch)'
      MOD=VGLSTART,TEMPORARY,TEXT='VERGLEICH VON REPMENGEN'
      MOD=CALLEDT,TEXT='EDT'
      ENDMENU
   MENU,TEXT='UEBERSICHTSLISTEN'
      MOD=MATRIX,TEXT='KORREKTUR GESAMTUEBERSICHT (Matrix-Liste)'
      MOD=NBKINFO,TEMPORARY,TEXT='AUSGEWAEHLTE NOTIZBUCHEINTREAGE'
      MOD=NOTBLIST,TEXT='GESAMTE NOTIZBUCHLISTE'
      MOD=CALLEDT,TEXT='EDT'
      ENDMENU
   MENU,TEXT='VERWALTER-FUNKTIONEN'
      MOD=RMSEXIM,TEMPORARY,TEXT='DEPOT-PFLEGE (Export - Import)'
      MOD=RMSEDIT,TEXT='TEXTVERARBEITUNG (Notizen, Textebenen)'
      MOD=CALLEDT,TEXT='EDT'
      ENDMENU
   ENDMENU
```

Das Einrücken ist nicht vorgeschrieben, macht aber die Liste leserlicher.

Das Modul INFODATA wird immer als Bestandteil des residenten RMS geladen und die Angabe von TEMPORARY ist wirkungslos.

12.6.6.5 DEPOT-Pflege

EXPORT-Funktion - Überblick

Die Export-Funktion dient zur Depotpflege. Es können Produkte, Textebenen sowie die Lader-Definitionen für Anlagenlader, Ladermodifikation und OPT-Rep Einstellung aus dem verwendeten Depot gelöscht werden. Die zu löschenden Daten (ausgenommen Textebenen) werden immer in ein Backup-Depot geschrieben, welches bei jeder Export-Funktion neu erstellt wird. Es können auch spezielle Backup-Depots erstellt werden, ohne Löschung im Original-Depot.

Ausgabedateiname des Backup-Depots: `RMS.DEPOT.jjmmtt.hhmmss`

Dieses Backup-Depot kann

- als normales Depot mit RMS verarbeitet werden,
- als Eingabe-Depot für die IMPORT-Funktion benutzt werden.

Ausgabedateiname Textebenen: `RMS.TEXTEBENE.name`

Diese Datei kann nur mit der RMS-Funktion „Einfahr“ in ein Depot übernommen werden.

IMPORT-Funktion - Überblick

Mit der IMPORT-Funktion können Bestandteile (Produkte, Anlagenladerdefinitionen, Lader Modifikationen und OPT-Rep Einstellungen) aus einem anderen Depot in das eröffnete Depot übernommen werden. Es können nur Bestandteile übernommen werden, die im eröffneten Depot noch nicht vorhanden sind.

Importieren von Textebenen ist nicht möglich.

Ablauf

Die gewünschten Aktionen sind durch Markieren der einzelnen Felder auszuwählen. Zum Selektieren innerhalb der vorhandenen Lieferbestandteile werden Folgebildschirme ausgegeben.

Die Funktion IMPORT wird nur durch die Eingabe eines Depotdateinamen aktiviert.

Beschreibung der markierbaren Felder

Produkte	Die auf dem Folgeschirm ausgewählten Produkte werden inklusiv der dazugehörigen Notizbucheinträge und Lieferungen exportiert bzw. Importiert. Diese Möglichkeit wird angeboten, um ein nicht mehr benötigtes Produkt zu entfernen.
Anlagenlader	Die auf dem Folgeschirm ausgewählten Anlagenladerdefinitionen werden inklusiv der dazugehörigen Lader-Modifikationen, OPT-Rep Einstellungen und SW-Konfiguration exportiert bzw. importiert.
Lader Modifikation	Die auf dem Folgeschirm ausgewählten Lader Modifikationen werden exportiert bzw. importiert.
OPT-REP Einstellung	Die auf dem Folgeschirm ausgewählten OPT-Rep Einstellungen werden exportiert bzw. importiert.
Textebene	Die auf dem Folgeschirm ausgewählten Textebenen werden als SAM-Dateien mit den Namen RMS.TEXTEBENE.name ausgegeben, falls <name> der Name der Textebene ist. Das Importieren von Textebenen ist nicht möglich.
löschen	Bestandteile nach dem Exportieren sollen gelöscht werden, RMS überprüft eventuelle Abhängigkeiten zu den Anlagenlader-Definitionen und gibt diese dann auf dem Bildschirm aus. Hier besteht nun die Möglichkeit, über den Löschvorgang zu entscheiden. Zu diesem Zeitpunkt ist das Backup-Depot bereits erstellt. F1 Alle ausgewählten Mengen werden gelöscht. DUE Es wird keine der ausgewählten Mengen gelöscht. Der Inhalt des Bildschirmes kann durch Überschreiben des Feldes '?' mit einem Zeichen ungleich Leerzeichen (Blank) und Absenden mit der DUE-Taste nach SYSLST ausgegeben werden.

12.7 Typische Einsatzfälle für RMS

Im Folgenden soll stichpunktartig der Ablauf von einigen typischen Einsatzfällen für RMS beschrieben werden.

12.7.1 Depot aufbauen

Aufgabe

Es soll erstmalig ein Depot für eine Anlagenkonfiguration aufgebaut werden. Es wird immer dann ein neues Depot von RMS angelegt, wenn RMS gestartet wird und versucht, ein nicht vorhandenes Depot zu adressieren. Der Name des neu erstellten Depots ist
RMS.DEPOT.<familie>, wobei familie der Parameter der RMS-Startprozedur ist.

Vorgehen über die Anweisungsschnittstelle

Rufen Sie RMS mit folgendem BS2000-Kommando auf und geben die RMS-Anweisungen ein, so wird ein Depot mit dem Namen RMS.DEPOT.TEST erzeugt. Beachten Sie, dass die Parameter ID, DEF und DEPOT standardmäßig TSOS zugeordnet sind.

```
CALL-PROC SYSPRC.RMS.071,(FAM=TEST,V7SYNTAX=YES,MODE=BATCH)
   INPUT-DELIVERY-PACKET INPUT=<filename>
   END
```

Der Typ der erstellten Depots ist abhängig von der Version des übernommenen Lieferpakets, Es gilt:

Version des Lieferpakets	Typ des erstellten Depots
#nn.nn	PULS-Depot
Ann.00	Support-Center oder Service-Center-Depot
Ann.xx	Kunden Level 1 Depot
ann.nn	Kunden Level 2 Depot
Erklärung: nn = 00 .. 99 xx = 01 .. 99	

12.7.2 Neue Lieferpakete ins Depot einfahren

Aufgabe

Es soll neue Lieferpakete (Korrekturmengen) ins Depot eingefahren werden, um danach einen neuen Lader für eine Anwendung zu bauen.

Vorgehen über die Dialogschnittstelle

Im Dialog können mehrere Lieferpakete gleichzeitig ins Depot übernommen werden. Tragen Sie dazu in der Funktionsmaske „Liefermengen übernehmen“ den Dateinamen ein, der das Lieferpaket enthält. Durch Angabe eines Wildcards können in einem Folgeschirm mehrere Dateien markiert werden und somit gleichzeitig mehrere Lieferpakete eingefahren werden.

12.7.3 Optionale Reps einschalten/aktivieren

Aufgabe

Es sollen optionale Reps für den Lader einer SPOOL-Version aktiviert werden. Optionale Reps sind von Hause aus immer deaktiviert und müssen vor der Erstellung eines Lader erst im Depot aktiviert werden.

Vorgehen über die Anweisungsschnittstelle

Zuerst muss das gewünschte optionale Rep-Paket erstellt werden. Starten Sie dazu RMS wie folgt und erzeugen Sie das Paket über das entsprechende RMS-Anweisung:

```
CALL-PROC SYSPRC.RMS.071,(FAM=TEST,V7SYNTAX=YES,MODE=BATCH)
   CREATE-OPT-PACKET NAME='OPT.SPOOL.049A.2',INPUT=*SYSDTA
   ADD=(A1111111,A2222222)
   ADD=A3333333
   END
   END
```

Jetzt müssen Sie das optionale Rep-Paket mit dem Lader verbinden.

```
MODIFY-REPFILE-DEFINITION NAME='LADER.SPOOL.049.2',-
                          ADDOPTS='OPT.SPOOL.049A.2'
END
```

12.7.4 Lader bauen im Dialog

Aufgabe

Für ein Produkt sollen neue Lader generiert und installiert werden. Vorausgesetzt wird, dass sowohl Lader als auch Software-Konfiguration im Depot definiert sind.

Vorgehen über die Anweisungsschnittstelle

Starten Sie RMS und wählen Sie das Produkt aus, für das Lader erzeugt werden sollen:

```
CALL-PROC SYSPRC.RMS.071,(FAM=TEST,V7SYNTAX=YES,MODE=BATCH)
```

```
SELECT-REPFILE-TO-BUILD PROD=SPOOL V4.9A
```

Alle Lader, die die SPOOL-Version V4.9A in ihrer SW-Konfiguration haben, werden ausgewählt. Jetzt können die ausgewählten Lader erzeugt werden.

```
BUILD-REPFILE
```

```
END
```

12.7.5 Lader bauen über Batchdatei

Folgendes Beispiel zeigt den Bau eines Laders über eine Batchdatei. Der Ablauf lässt sich durch Nutzung der Anweisungsschnittstelle somit vollständig automatisieren.

```
/ BEGIN-PROCEDURE
/          SET-PROC-OPTIONS LOGGING=YES,DATA-ESCAPE-CHAR='&&'
/          BEGIN-PARAMETER-DECLARATION
/          DECL-PARAM NAME=RMS        (TYPE=STRING ,INIT='RMS')
/          DECL-PARAM NAME=RMSOML     (TYPE=STRING ,INIT='SYSOML.RMS.071')
/          DECL-PARAM NAME=RMSFUNC    (TYPE=STRING ,INIT='SYSDAT.RMS.071')
/          END-PARAMETER-DECLARATION
/          MODIFY-JOB-SWITCHES OFF=*ALL
/          &*
/          &* RMS CALL
/          &* --------
/          &*
/          ADD-FILE-LINK LINK-NAME=RMSOML,FILE-NAME=&RMSOML
/          ADD-FILE-LINK LINK-NAME=DEF,FILE-NAME=&RMSFUNC
/          ASSIGN-SYSDTA TO-FILE=*SYSCMD
/          START-EXECUTABLE-PROGRAM *MOD(ELEMENT=&RMS,LIBRARY=&RMSOML)
LMAX=55,FAM=EX,V7SYNTAX=JA,MODE=BATCH
CREATE-RMS-OPTIONS PROTOCOL=EX.PROT,ERROR=ON(30)                          1.
CREATE-SW-CONF NAME='SYSREP.SPOOL.049',ADD=SPOOL V4.9A,ERROR=ON(30)       2.
CREATE-REPFILE-DEFINITION NAME='LADER.SPOOL.049.1',PASS='SPOOL',-         3.
          ADDPROD=SPOOL V4.9A,ERROR=ON(30)
SELECT-REPFILE-TO-BUILD PROD=SPOOL V4.9A,-                                4.
             USERID=$RMSD,ERROR=ON(30)
BUILD-REPFILE ERROR=ON(30)                                                5.
END
/          ASSIGN-SYSDTA TO-FILE=*SYSCMD
/          SKIP-COMMANDS TO-LABEL=TCERR,IF=JOB-SWITCHES(ON=30)
/          SKIP-COMMANDS TO-LABEL=TCOK
/.TCERR    REMARK
/          REMARK **** ERROR PROCESSING ****
/          SKIP-COMMANDS TO-LABEL=ENDP
/.TCOK     REMARK
/          REMARK **** OK PROCESSING ****
/.ENDP     REMARK
/          MODIFY-JOB-SWITCH OFF=*ALL
/EXIT-PROCEDURE
```

1. Setzen der RMS Optionen. Die Datei EX.PROT wird als Protokolldatei genutzt.
2. Definition einer Software-Konfiguration SYSREP.SPOOL.049 für das Produkt SPOOL V4.9A. Diese Definition ist Voraussetzung für das Erzeugen eines Laders, da jeder Lader einer Software-Konfiguration zugeordnet sein muss.
3. Definition des Laders für das Software-Produkt SPOOL V4.9A. Das Produkt wird automatisch der Software-Konfiguration SYSREP.SPOOL.049 zugeordnet.
4. Auswahl des zu erzeugenden Laders. Alle Lader, die das Produkt SPOOL V4.9A in ihrer Definition enthalten, werden ausgewählt.
5. Erzeugen der ausgewählten Lader. Alle Lader werden so erzeugt oder installiert, wie es in den Optionen bei ihrer Definition ausgewählt wurde. Die Erzeugung und Installation läuft unter der in der SELECT-REPFILE-TO-BUILD Anweisung angegebenen Benutzerkennung $RMSD ab.

12.7.6 Übergabe einer REP-Menge an die nachfolgende Anwendergruppe

Aufgabe

Einer nachfolgenden Anwendergruppe soll eine REP-Menge zu SPOOL V4.9A und Liefermenge A00.01übergeben werden (gilt für die Anwendergruppen 1 bis 3). Dazu muss eine entsprechende Liefermenge erzeugt werden.

Vorgehen über die Anweisungsschnittstelle

Starten Sie RMS mit dem Kommando

```
CALL-PROCEDURE SYSPRC.RMS.071,(FAM=TEST,V7SYNTAX=YES,MODE=BATCH)
```

Jetzt kann die Liefermenge erzeugt werden mit der Anweisung

```
CREATE-DELIVERY-PACKET PRODUCT=SPOOL V4.9A/A00.01
```

Haben Sie in einer RMS-Sitzung mehrere Lieferpakete erhalten und wollen sie direkt an eine nachfolgende Anwendergruppe weiterreichen, so können Sie dies mit der folgenden Anweisung bewirken:

```
CREATE-DELIVERY-PACKET PROD=*PRODSAVE
```

Vorgehen über die Dialogschnittstelle

Wählen Sie im Menü „Repverwaltung“ die Funktion „Liefermengen erzeugen“ aus. In der folgenden Funktionsmaske können Sie den Lieferbestandteil auswählen, für den Sie eine Liefermenge erzeugen wollen.

12.7.7 Zurückweisung einer falschen Rep-Korrektur

Aufgabe

Es soll eine falsche Korrektur (hier der REP A9999999/280 zu SPOOL V4.9A), den Sie mit einer Liefermenge erhalten haben, zurückgewiesen werden, um die Generierung eines Laders oder eine Liefermenge mit diesem REP zu vermeiden. Dies geschieht über die Notizbuch-Funktion.

Vorgehen über die Anweisungsschnittstelle

Starten Sie RMS mit dem Kommando

```
CALL-PROCEDURE SYSPRC.RMS.071,(FAM=TEST,V7SYNTAX=YES,MODE=BATCH)
```

Geben nun folgende RMS-Anweisung ein, um eine entsprechende Notiz zu erzeigen:

```
MODIFY-DELIVERY-PACKET IDENT='myname',REJ,A9999999/280,SPOOL V4.9A
```

Vorgehen über die Dialogschnittstelle

Wählen Sie im Menü „Repverwaltung“ die Funktion „Korrekturverwaltung Notizbuch“ aus. In der folgenden Funktionsmaske können Sie den Rep eintragen, den Sie zurückweisen wollen. Ausführliche Information zu den einzelnen Feldern der Maske erhalten Sie in der Online-Hilfe zur Funktionsmaske (Eingabe von ESC + ? oder K9).

12.8 Anweisungen

Die in diesem Kapitel beschriebenen Anweisungen ersetzen alle Anweisungen aus den RMS-Versionen vor V7. Aus Kompatibilitätsgründen können die alten Anweisungen noch verwendet werden. Eine Beschreibung der alten Anweisungen befindet sich im Abschnitt „Anweisungen von RMS vor V7.0“.

Um die neuen Anweisungen zu benutzen, ist der folgender RMS-Einleitungsparameter anzugeben:

V7SYNTAX = JA / YES

Die gleichzeitige Benutzung von alten und neuen Anweisungen ist nicht möglich.

Die Metasyntax der Anweisungen ist die für ISP-Kommandos und ist beschrieben im Abschnitt „Darstellungsmittel“.

Ablauf

Die Anweisungen werden zuerst syntaktisch überprüft und in eine Zwischendatei geschrieben. Abhängig vom MODE-Operanden wird entweder die Anweisung aus der Zwischendatei abgearbeitet (Normalfall) oder im Falle MODE=TEST der Lauf beendet. Beim Feststellung eines Syntaxfehlers wird der Lauf abnormal beendet, ohne eine Aktion durchzuführen.

Bei fehlenden Produkten, OPT- oder MOD-Paketen werden gelöschte Einträge während der zweiten Bearbeitungsphase ins Depot eingetragen, d.h. es findet keine Kontextüberprüfung statt.

12.8.1 Erzeugen und Bearbeiten einer Software-Konfiguration

- CREATE-SW-CONF - Software-Konfiguration neu erstellen
- MODIFY-SW-CONF - Software-Konfiguration ändern
- DELETE-SW-CONF - Software-Konfiguration löschen

12.8.1.1 CREATE-SW-CONF - Software-Konfiguration neu erstellen

Eine Software-Konfiguration wird neu erstellt, eine bereits vorhandene Software-Konfiguration wird überschrieben. Alle bisher in diesem Eintrag vorhandenen Einstellungen gehen verloren. Als Abschluss werden alle Anlagenladerdefinitionen überprüft. Falls eine Standard-Einstellung festgestellt wird, werden die Einstellungen aus der neu eingetragenen Software-Konfiguration übernommen.

Format

```
CREATE-SW-CONF

NAME = 'swkonfname'

[,PASS = 'password']

,ADD = productversion / (productversion,...)

[,SETTINGS = (

      ,SHARE = NO / YES

      ,SAVE = NO / YES

      ,INSTALL = NO / YES

      ,INFORMATION = MAX / MED / MIN

      ,FUNCTION-NAME = NO / YES

      [,VOLUME = vol,DEVICE = dev]

      [,PATH = [:catid:][$userid.]filename]

      ,LOADER-FORMAT = SUB-SYSTEM / BS2000 )]

[,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]
```

Operanden

NAME = 'swkonfname'
Name der Software-Konfiguration, die neu erstellt werden soll.

PASS = 'password'
Passwort, mit der die Software-Konfiguration geschützt wird.

ADD = productversion / (productversion, ...)
Produkt und ggf. Produktversion, das in die Software-Konfiguration aufgenommen werden soll. Der Produktname kann maximal 15 Zeichen lang sein. Die Produktversion ist im Format V[n]n.nn[nn] einzugeben.

SETTINGS = (...)
Bestimmt die Eigenschaften des Laders in Bezug zur Software-Konfiguration. Der erste genannte Operand kann ohne führendes Komma angegeben werden, die Reihenfolge ist beliebig.

SHARE = NO / YES
Definieren der Zugriffsberechtigung

SAVE = YES / NO
Sicherung der Anlagenladerdatei, die bei einer Installation überschrieben wird. Anlagenladerdateien im BS2LOADER-Format werden immer gesichert.

INSTALL = YES / NO
Es soll eine Installation unter dem Installationspfadnamen durchgeführt werden

INFORMATION = MED / MAX / MIN
Folgende Informationen werden mit INFORMATION=MAX bei Einsatz des Anlagenladers auf der Konsole ausgegeben:

- Name des Anlagenladers und die Generierungsversion der im Anlagenlader enthaltenen Produkte und dessen verwendete LB-Version,
- alle Korrekturnummern für Notizbucheinträge, die bei der Erstellung aktiv waren,
- die durchgeführten Anlagenlader-Modifikationen
- alle aktivierten OPT-Reps.

Informationen zur Ausgabe bei MIN/MED siehe "Laderdienste".

FUNCTION-NAME = NO / YES
Zu den Subnos der aktivierten OPT-Reps werden die Funktionsnamen ausgegeben. Bei NO werden nur die PM-Nr s ausgegeben.

VOLUME = vol, DEVIVE = dev
Datenträger und Gerät, unter dem der entsprechende Lader abgelegt wird.

PATH = :catid:$userid.filename
Pfadname, unter dem der entsprechende Lader abgelegt wird.

LOADER-FORMAT = SUB-SYSTEM / BS2000
BS2000-Format wird beim Systemladen benötigt, Sub-System wird von DSSM bearbeitet.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.1.2 MODIFY-SW-CONF - Software-Konfiguration ändern

Eine bestehende Software-Konfiguration wird geändert. Ist die Software-Konfiguration nicht im Depot vorhanden, wird *keine* Software-Konfiguration angelegt.

Format

MODIFY-SW-CONF
NAME = 'swkonfname' [,NEW-NAME = 'swkonfname'] [,PASS = 'password'][,NEW-PASS = 'password'] [,{ ADD / DELL } = productversion / (productversion,...)] [,SETTINGS = (,SHARE = NO / YES ,SAVE = NO / YES ,INSTALL = NO / YES ,INFORMATION = MAX / MED / MIN ,FUNCTION-NAME = NO / YES [,VOLUME = vol,DEVICE = dev] [,SHARE = [:catid:][$userid.]filename] ,LOADER-FORMAT = SUB-SYSTEM / BS2000)] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*swkonfname*'
Name einer bestehenden Software-Konfiguration, die geändert werden soll.

NEW-NAME = '*swkonfname*'
Neuer Name der Software-Konfiguration.

PASS = '*password*', NEW-PASS = '*password*'
Passwort, mit der die Software-Konfiguration geschützt ist. Falls die Software-Konfiguration mit einem Passwort geschützt ist, muss dieses erst angegeben werden, bevor irgend ein Attribut geändert werden kann. Mit NEW-PASS='password' kann ein neues Passwort vergeben werden.

ADD / DEL = productversion/ (productversion, ...)
Produkt und ggf. Produktversion, das in die Software-Konfiguration aufgenommen oder dort gelöscht werden soll. Der Produktname kann maximal 15 Zeichen lang sein. Die Produktversion ist im Format V[n]n.nn[nn] einzugeben.

SETTINGS = (...)
Beschreibung des SETTINGS-Operanden siehe Anweisung CREATE-SW-CONF - Software-Konfiguration neu erstellen (dprog V21.0C, #333).

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.1.3 DELETE-SW-CONF - Software-Konfiguration löschen

Eine bestehende Software-Konfiguration wird gelöscht.

Format

DELETE-SW-CONF
NAME = 'swkonfname' [,PASS = 'password'] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*swkonfname*'
Name einer bestehenden Software-Konfiguration, die gelöscht werden soll.

PASS = '*password*'
Passwort, mit der die Software-Konfiguration geschützt ist. Falls die Software-Konfiguration mit einem Passwort geschützt ist, muss dieses erst angegeben werden, bevor sie gelöscht werden kann.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.2 Optionale Rep-Auswahl

- CREATE-OPT-PACKET - Optionales REP-Auswahlpaket erstellen
- MODIFY-OPT-PACKET - Optionales REP-Auswahlpaket erweitern
- DELETE-OPT-PACKET - Optionales Paket löschen

12.8.2.1 CREATE-OPT-PACKET - Optionales REP-Auswahlpaket erstellen

Im Depot wird ein neues Element unter dem angegebenen `optpacketname` angelegt. Ein bestehendes Element wird überschrieben.

Format

CREATE-OPT-PACKET
NAME = 'optpacketname' ,INPUT = *SYSDTA / filename [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*optpacketname*'
Name des optionalen REP-Auswahlpaketes, das neu erstellt werden soll. Der Name kann maximal 22 Zeichen lang sein.

INPUT = *SYSDTA / filename
Die folgenden ADD-Anweisungen werden von SYSDTA gelesen oder stehen in der angegebenen Datei, wobei Dateiname ein vollqualifizierter Name sein kann.

Die *SYSDTA-Eingabe bzw. die Eingabedatei hat folgenden Aufbau:

```
ADD = subno / (subno, ...) [,produktfamilie]
...
ADD = subno / (subno, ...) [,produktfamilie]
END
```

Dabei ist `produktfamilie` der Produktname ohne Versionsbezeichnung.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.2.2 MODIFY-OPT-PACKET - Optionales REP-Auswahlpaket erweitern

Ein im Depot vorhandenes Element wird verändert. Ist `optpacketname` nicht im Depot vorhanden, wird ein neues Element angelegt.

Format

MODIFY-OPT-PACKET
NAME = 'optpacketname' [,NEW-NAME = 'optpacketname'] ,INPUT = *SYSDTA / filename [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*optpacketname*'
Name des optionalen REP-Auswahlpaketes, das verändert werden soll. Der Name kann maximal 22 Zeichen lang sein.

NEW-NAME = '*optpacketname*'
Neuer Name des optionalen REP-Auswahlpaketes, das verändert werden soll. Der Name kann maximal 22 Zeichen lang sein.

INPUT = *SYSDTA / filename
Die folgenden ADD-Anweisungen werden von SYSDTA gelesen oder stehen in der angegebenen Datei, wobei Dateiname ein vollqualifizierter Name sein kann.

Die*SYSDTA-Eingabe bzw. die Eingabedatei hat folgenden Aufbau:

```
ADD = subno / (subno, ...) [,produktfamilie]
...
ADD = subno / (subno, ...) [,produktfamilie]
DEL = subno / (subno, ...) [,produktfamilie]
END
```

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.2.3 DELETE-OPT-PACKET - Optionales Paket löschen

Das angegebene Element wird gelöscht.

Format

DELETE-OPT-PACKET
NAME = 'optpacketname' [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = 'optpacketname'
Name des optionalen REP-Auswahlpaketes, das gelöscht werden soll. Der Name kann maximal 22 Zeichen lang sein.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.3 Ladermodifikationen

- CREATE-MOD-PACKET - Ladermodifikations-Element neu erstellen
- MODIFY-MOD-PACKET - Ladermodifikations-Element verändern
- DELETE-MOD-PACKET - Ladermodifikations-Element löschen

12.8.3.1 CREATE-MOD-PACKET - Ladermodifikations-Element neu erstellen

Im Depot wird ein neues Element `modpacketname` angelegt. Ein bestehendes Element wird überschrieben.

Format

CREATE-MOD-PACKET
NAME = 'modpacketname' ,INPUT = *SYSDTA / filename [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*modpacketname*'
Name des Ladermodifikations-Elements, das neu erstellt werden soll. Der Name kann maximal 22 Zeichen lang sein.

INPUT = *SYSDTA / filename
Die folgenden ADD-/DEL-Anweisungen werden von SYSDTA gelesen oder stehen in der angegebenen Datei, wobei Dateiname ein vollqualifizierter Name sein kann.

Die *SYSDTA-Eingabe bzw. die Eingabedatei hat folgenden Aufbau:

```
ADD / DEL = subno[-jd],productversion,TEXT='begründung'
...
ADD / DEL = subno[-jd],productversion,TEXT='begründung'
END
```

> Bei ADD muss das Datum -jd mit angegeben werden.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.3.2 MODIFY-MOD-PACKET - Ladermodifikations-Element verändern

Ein vorhandenes Element wird verändert. Ist `modpacketname` nicht im Depot vorhanden, wird ein neues Element angelegt.

Format

MODIFY-MOD-PACKET
NAME = 'modpacketname' [,NEW-NAME = 'modpacketname'] ,INPUT = *SYSDTA / filename [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*modpacketname*'
Name des Ladermodifikations-Elements, das verändert werden soll. Der Name kann maximal 22 Zeichen lang sein.

NEW-NAME = '*modpacketname*'
Neuer Name der Laderelements. Der Name kann maximal 22 Zeichen lang sein.

INPUT = *SYSDTA / filename
Die folgenden ADD-/DEL-Anweisungen werden von SYSDTA gelesen oder stehen in der angegebenen Datei, wobei Dateiname ein vollqualifizierter Name sein kann.

Die*SYSDTA-Eingabe bzw. die Eingabedatei hat folgenden Aufbau:

```
ADD / DEL = subno[-jd],productversion,TEXT='begründung'
...
ADD / DEL = subno[-jd],productversion,TEXT='begründung'
END
```

> Bei ADD muss das Datum -jd mit angegeben werden.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.3.3 DELETE-MOD-PACKET - Ladermodifikations-Element löschen

Das angegebene Element wird gelöscht.

Format

DELETE-MOD-PACKET
NAME = 'modpacketname' [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = 'modpacketname'
Name des Ladermodifikations-Elements, das gelöscht werden soll. Der Name kann maximal 22 Zeichen lang sein.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.4 REPFILE-Definitionen

- CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen
- MODIFY-REPFILE-DEFINITION - Laderdefinitionen ändern
- DELETE-REPFILE-DEFINITION - Laderdefinitions-Element löschen

12.8.4.1 CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen

Die gewünschten Laderdefinitionen werden unter dem Element `loaderdefname` neu angelegt. Ein bestehendes Element wird überschrieben.

Es wird versucht, der Produktzusammenstellung eine Software-Konfiguration zuzuordnen. Wenn dies nicht möglich ist, wird der Lauf abgebrochen. Ist die Definition unvollständig, wird eine Warnung ausgegeben und der Lauf fortgesetzt.

Format

CREATE-REPFILE-DEFINITION
NAME = 'laderdefname' [,PASS = 'password'] ,ADDPROD = *PRODSAVE / productversion / (productversion,...) [,ADDMODS = 'modpacketname' / ('modpacketname',...) ,...] [,ADDOPTS = 'optpacketname' / ('optpacketname',...) ,...] [,SETTINGS = (,SHARE = NO / YES ,SAVE = NO / YES ,INSTALL = NO / YES ,INFORMATION = MED / MAX / MIN ,FUNCTION-NAME = NO / YES [,VOLUME = vol,DEVICE = dev] [,PATH = [:catid:][$userid.]filename] ,LOADER-FORMAT = SUB-SYSTEM / BS2000)] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*loaderdefname*'
Name der Laderdefinition, die neu erstellt werden soll.

PASS = '*password*'
Passwort, mit der die Laderdefinition geschützt werden soll.

ADDPROD = *PRODSAVE / productversion / (productversion, ...)
Es werden diejenigen Produkte definiert, die in der Rep-Datei enthalten sein müssen.

*PRODSAVE bedeutet, dass auf eine Liste von Produkten zugegriffen wird, die in einer vorangegangenen Anweisung angegeben wurde. RMS merkt sich diese Liste und benutzt sie hier wieder.

In der Rep-Datei kann ein Produkt oder eine Reihe von Produkten (maximal 10) enthalten sein.

ADDMODS = '**modpacketname**' **/ (**'**modpacketname**'**, ...)**
Es werden diejenigen Ladermodifikations-Elemente definiert, die in der Rep-Datei enthalten sein müssen. Die Ladermodifikations-Elemente werden erzeugt oder bearbeitet durch die Anweisungen CREATE- oder MODIFY-MOD-PACKET.

In der Rep-Datei kann ein Element oder eine Reihe von Elementen (maximal 10) enthalten sein.

ADDOPTS = 'optpacketname' / ('optpacketname', ...)
Es werden diejenigen optionalen REP-Auswahlpakete definiert, die in der Rep-Datei enthalten sind. Die optionalen REP-Auswahlpakete werden erzeugt oder bearbeitet durch die Anweisungen CREATE- oder MODIFY-OPT-PACKET.

In der Rep-Datei kann ein Paket oder eine Reihe von Paketen (maximal 10) enthalten sein.

> ¡ Die Summe der angegebenen ADDMODS / ADDOPTS Operanden muss >= 9 sein.

SETTINGS = (...)
Beschreibung des SETTINGS-Operanden, siehe Anweisung CREATE-SW-CONF.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.4.2 MODIFY-REPFILE-DEFINITION - Laderdefinitionen ändern

Die spezifizierten Eigenschaften des ausgewählten Elements `loaderdefname` werden geändert. Ist die Laderdefinition nicht im Depot vorhanden, wird ein neues Element angelegt.

Ein Versuch, für eine nicht existierende Laderdefinition ein DELPROD, DELOPTS oder DELMODS durchzuführen, wird erst bei der Durchführung der Aktions bemerkt und mit einer Warnung zurückgewiesen. Ebenso verhält sich RMS, wenn die Produktversion/modpacketname/optpacketname des jeweiligen DEL-Operanden nicht im Depot zu finden ist.

Beim Anlegen einer neuen Laderdefinition wird versucht, eine eindeutige Zuordnung zu einer im Depot gespeicherten Software-Konfiguration zu treffen. Falls erfolgreich, werden die Voreinstellwerte aus der Software-Konfiguration übernommen und ggf. durch die Operandenwerte in der Anweisung überschrieben.

Format

MODIFY-REPFILE-DEFINITION
NAME = 'loaderdefname'[,NEW-NAME = 'loaderdefname'] [,PASS = 'password'[,NEW-PASS = 'password']] [,{ ADDPROD / DELPROD } = *PRODSAVE / productversion / (productversion,...)] [{ { ADDMODS / DELMODS } = 'modpacketname' / 'modpacketname',...) / { ADDOPTS / DELOPTS } = 'optpacketname' / 'optpacketname',...) } ,...] [,SETTINGS = (,SHARE = NO / YES ,SAVE = NO / YES ,INSTALL = NO / YES ,INFORMATION = MED / MAX / MIN ,FUNCTION-NAME = NO / YES [,VOLUME = vol,DEVICE = dev] [,PATH = [:catid:][$userid.]filename] ,LOADER-FORMAT = SUB-SYSTEM / BS2000)] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = 'loaderdefname'
Name einer bestehenden Laderdefinition, die geändert werden soll

NEW-NAME = 'loaderdefname'
Neuer Name der Laderdefinition.

PASS = 'password', NEW-PASS = 'password'
Passwort, mit der die Laderdefinition geschützt werden soll. Wird ein neues Passwort vergeben, muss erst das Alte, und dann unter NEW-PASS das neue Passwort angegeben werden.

ADDPROD / DELPROD = *PRODSAVE / productversion / (productversion, ..)
siehe CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen (dprog V21.0C, #345)

ADDMODS / DELMODS = 'modpacketname' / ('modpacketname',...)
siehe CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen (dprog V21.0C, #345)

ADDOPTS / DELOPTS = 'optpacketname' / ('optpacketname',...)
siehe CREATE-REPFILE-DEFINITION - Laderdefinitionen neu erstellen (dprog V21.0C, #345)

> Die Summe der angegebenen ADDMODS / DELMODS / ADDOPTS / DELOPTS Operanden muss <= 9 sein.

SETTINGS = (...)
Beschreibung des SETTINGS-Operanden, siehe Anweisung CREATE-SW-CONF - Software-Konfiguration neu erstellen (dprog V21.0C, #333).

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.4.3 DELETE-REPFILE-DEFINITION - Laderdefinitions-Element löschen

Das ausgewählte Element wird gelöscht.

Format

DELETE-REPFILE-DEFINITION
NAME = 'loaderdefname' [,PASS = 'password'] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*loaderdefname*'
Name der Laderdefinition, die gelöscht werden soll.

PASS = '*password*'
Passwort, mit der die Laderdefinition geschützt ist.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.5 Anlagenladerdatei erstellen

- SELECT-REPFILE-TO-BUILD - Anlagenladerdefinitionen auswählen
- BUILD-REPFILE - Anlagenlader erstellen bzw. installieren

12.8.5.1 SELECT-REPFILE-TO-BUILD - Anlagenladerdefinitionen auswählen

Die Elemente für die einsatzfähigen Lader werden ausgewählt. Die Erstellung der Anlagenladerdefinition erfolgt erst bei der nächsten BUILD-REPFILE-Anweisung.

Format

SELECT-REPFILE-TO-BUILD
{ NAME = ’laderdefname’ / (’laderdefname’,...) /
,PRODUCT = *PRODSAVE / productversion / (productversion,...) }
,INSTALL = DEF / ABS / NO
[,USERID = [:catid:][$userid.]]
[,PREFIX = prefix]
[,RUNAME = releaseunitname]
[,RUVERS = releaseunitversion]
[,SCI = imoninventoryname]
[,ADDOPTS = 'optpacketname' / ('optpacketname',...)]
[,OVERWRITE = YES / NO]
[,SETTINGS = STD / SWK]
[,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

NAME = '*loaderdefname*'
Name der Anlagenladerdefinition, die ausgewählt werden soll. Es kann eine Liste von bis zu 30 Namen angegeben werden.

Falls NAME nicht angegeben wird, muss PRODUCT angegeben werden. In diesem Fall werden alle Lader, die das Produkt in ihrer Definition enthalten, ausgewählt.

Falls NAME und PRODUCT angegeben sind, werden die angegebenen Lader nur dann ausgewählt, wenn sie eines der unter PRODUCT angegebenen Produkte beinhalten.

PROD[UCT] = *PRODSAVE / productversion / (productversion, ...)

Definiert die Produkte, die zur Erzeugung des Laders herangezogen werden sollen.

*PRODSAVE bedeutet, dass auf eine Liste von Produkten zugegriffen wird, die in einer vorangegangenen Anweisung angegeben wurde. RMS merkt sich diese Liste und benutzt sie hier wieder.

Es kann ein Produkt oder eine Reihe von Produkten (maximal 10) angegeben werden.

Falls mehrere Produkte angegeben werden, wird jedes Einzelne unabhängig vom jeweils anderen herangezogen und alle Lader ausgewählt, die das jeweilige Produkt enthalten.

INSTALL = DEF / ABS / NO
Legt fest, wie der Wert für INSTALL in der Laderdefinition ausgewertet werden soll.

INSTALL = DEF
Die Installation wird in Abhängigkeit des Wertes für INSTALL in der Laderdefinition durchgeführt.

INSTALL = ABS
Die Installation wird immer durchgeführt, unabhängig vom Wert für INSTALL in der Laderdefinition.

INSTALL = NO
Die Installation wird nicht durchgeführt, unabhängig vom Wert für INSTALL in der Laderdefinition.

USERID = :catid:$userid
Legt fest, wo der generierte Lader gespeichert werden soll. Ist keine USERID angegeben, dann wird der Lader unter der USERID des RMS-Depots bereitgestellt.

PREFIX = prefix
Gibt das Prefix an, mit dem die Laderdatei gespeichert wird.

RUNAME = releaseunitname
Legt den Release-Unit-Namen für IMON fest (siehe Handbuch „IMON“ [15]). Der Name darf maximal 30 Zeichen lang sein. Der Operand darf nur dann angegeben werden, wenn beim Aufruf der Startprozedur MODE=IMON angegeben wurde.

RUVERS = releaseunitversion
Legt den Release-Unit-Version für IMON fest (siehe Handbuch „IMON“ [15]). Die Versionsangabe darf maximal 8 Zeichen lang sein. Der Operand darf nur dann angegeben werden, wenn beim Aufruf der Startprozedur MODE=IMON angegeben wurde.

SCI = imoninventoryname
Legt das Software-Configuration-Inventory für IMON fest (siehe Handbuch „IMON“ [15]). Der Operand darf nur dann angegeben werden, wenn beim Aufruf der Startprozedur MODE=IMON angegeben wurde.

ADDOPTS = 'optpacktename' / ('optpacketname',...)
siehe Anweisung CREATE-REPFILE-DEFINITION.

Die Summe der angegebenen ADDOPTS-Operanden muss <= 9 sein.

OVERWRITE = YES / NO
Option für IMON (siehe Handbuch „IMON“ [15]). Falls die zu erzeugende SYSREP-Datei bereits existiert, muss sie bei OVERWRITE=NO als SYSREP.<nnn> katalogisiert werden, wobei n = 0...9.

SETTINGS = STD / SWK
Gibt an, welche Einstellungen bei der Installation verwendet werden sollen.

SETTINGS = STD
Die Einstellungen aus den Laderdefinitionen werden verwendet (siehe Anweisung CREATE-REPFILE-DEFINITION).

SETTINGS = SWK
Die Einstellungen aus den Software-Konfigurationen werden verwendet (siehe Anweisung CREATE-SW-CONF).

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.5.2 BUILD-REPFILE - Anlagenlader erstellen bzw. installieren

Aus allen mit SELECT-REPFILE-TO-BUILD ausgewählte Elementen wird die REPFILE erzeugt.

Format

BUILD-REPFILE
[,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)

Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.6 Transport von Korrekturliefermengen

- INPUT-DELIVERY-PACKET - Lieferpaket übernehmen
- MODIFY-DELIVERY-PACKET - Abweichungen erfassen (Notizbuch)
- CHECK-DELIVERY-PACKET - Lieferpaket prüfen
- CREATE-DELIVERY-PACKET - Lieferpaket erzeugen

12.8.6.1 INPUT-DELIVERY-PACKET - Lieferpaket übernehmen

Alle im Paket enthaltene REP-Pakete, Texte, Software-Konfigurationen und ggf. OPT-Einstellungen werden geprüft und, falls einwandfrei, ins Depot übernommen.

Als Abschluss werden alle Anlagenladerdefinitionen überprüft. Falls eine Standard-Einstellung festgestellt wird, werden die Einstellungen aus der neu eingetragenen Software-Konfiguration übernommen (gilt für die Anwendergruppen 2, 3 und 4).

Format

INPUT-DELIVERY-PACKET
INPUT = filename [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

INPUT = filename
legt den Dateinamen fest, der die Reps enthält, die ins Depot übernommen werden sollen.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.6.2 MODIFY-DELIVERY-PACKET - Abweichungen erfassen (Notizbuch)

Vorab-Korrekturen werden hinzugefügt, fehlerhafte Korrekturen ungültig gesetzt und frühere Aktionen rückgängig gemacht. Jede Aktion wird im Depot notiert.

Format

MODIFY-DELIVERY-PACKET
IDENT = 'userid' { { ADD,subno/jd,productversion[,INPUT=filename] / REJ,subno/jd,productversion / RST,subno } ,REASON=’text’ / SIS,INPUT=filename } [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

IDENT = 'userid'
legt den Benutzernamen fest, unter dem alle Abweichungen protokolliert werden. Der Name darf maximal 44 Zeichen lang sein.

ADD / REJ / RST ..., REASON = 'text'
Eine ausführliche Beschreibung dieser Operanden finden im Abschnitt „Korrekturverwaltung - Notizbuch“.
'text' darf maximal 44 Zeichen lang sein.

> ¡ In den Anweisungen ADD, REJ und SIS ist der Schrägstrich zwischen subno und jd Bestandteil der Eingabe und muss mit eingegeben werden!

SIS, INPUT=filename
Datei, die die Reps enthält, die ins Depot übernommen werden sollen. Dies ist nur für die Anwendergruppe 2 möglich.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.6.3 CHECK-DELIVERY-PACKET - Lieferpaket prüfen

Eine Produktversion oder Lieferbestandteil wird auf Vollständigkeit überprüft.

Format

CHECK-DELIVERY-PACKET
PRODUCT = *PRODSAVE / productversion / lb [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

PROD[UCT] = *PRODSAVE / productversion / lb
Die Produktversion oder der Lieferbestandteil, der auf Vollständigkeit überprüft wird.

*PRODSAVE bedeutet, dass auf eine Liste von Produkten zugegriffen wird, die in einer vorangegangenen Anweisung angegeben wurde. RMS merkt sich diese Liste und benutzt sie hier wieder.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.6.4 CREATE-DELIVERY-PACKET - Lieferpaket erzeugen

Ein Lieferpaket wird erzeugt und im Depot notiert. Diese Funktion ist bis zur ersten Kundenebene verfügbar.

Format

CREATE-DELIVERY-PACKET
PRODUCT = *PRODSAVE / productversion / lb [,TEXT = *ALL / *NONE / 'textname' / ('textname',...)] [,START = *ALL / lb] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

PROD[UCT] = *PRODSAVE / productversion / lb
Die Produktversion mit dem Lieferbestandteil, die zur Erzeugung des Lieferpakets herangezogen werden.

*PRODSAVE bedeutet, dass auf eine Liste von Produkten zugegriffen wird, die in einer vorangegangenen Anweisung angegeben wurde. RMS merkt sich diese Liste und benutzt sie hier wieder.

TEXT = *ALL / *NONE / 'textname' / ('textname',...)
Steuert die Liste, der in die Lieferung einzubeziehenden Textebenen. Unabhängig von der Angabe ist die Textebene „SK-OPT BESCHREIBUNG“ immer enthalten.

TEXT = *ALL
Es sollen alle Textebenen einbezogen werden.

TEXT = *NONE
Es soll keine Textebene einbezogen werden.

TEXT = 'textname' / ('textname',...)
Die angegebene Textebene bzw. die angebene Liste von Textebenen soll einbezogen werden.

START = *ALL / lb
Gibt an, ob alle Textebenen (*ALL) oder nur die Textebene (lb), die seit einer bestimmten Lieferkomponente eingeführt wurde, einbezogen werden sollen.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.7 Informationsfunktionen

- CREATE-MATRIX-LIST - MATRIX-Liste erstellen
- COMPARE-REPFILE - Repmengen vergleichen
- CREATE-RMS-OPTIONS - RMS-Einstellungen ändern
- CALL-FUNCTION - Modul aufrufen

12.8.7.1 CREATE-MATRIX-LIST - MATRIX-Liste erstellen

Eine MATRIX-Liste für ein Produkt oder mehrere Produkte wird erstellt.

Format

CREATE-MATRIX-LIST
ACTUAL / DELIVERED / ORIGINAL ,PRODUCT = *PRODSAVE / productversion / (productversion,...) ,LIST = SUMMARY / DETAIL [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

ACTUAL / DELIVERED / ORIGINAL
Legt fest, wie mit den Reps verfahren werden soll.

ACTUAL
es werden die Reps ausgegeben, wie sie an RMS übergeben wurden incl. der Notizbuchaktionen.

DELIVERED
es werden die Reps ausgegeben, die ausgeliefert wurden.

ORIGINAL
es werden die Reps ausgegeben, wie sie an RMS übergeben wurden.

PROD[UCT] = *PRODSAVE / productversion
Die Produktversion oder der Lieferbestandteil, für die eine Matrix-Liste erstellt wird.

*PRODSAVE bedeutet, dass auf eine Liste von Produkten zugegriffen wird, die in einer vorangegangenen Anweisung angegeben wurde. RMS merkt sich diese Liste und benutzt sie hier wieder.

LIST = SUMMARY / DETAIL
legt fest, welche Informationen zu jedem Rep und den dazugehörigen Fehlermeldungen ausgegeben wird

LIST = SUMMARY
Es wird zu jeder Version eines Lieferbestandteils ausgegeben:

- die Anzahl der Rep-Karten
- die Anzahl der Fehlermeldungen
- die Anzahl neuer, fehlender, gleicher oder geänderter Fehlermeldungen
- die Anzahl fehlender Fehlermeldungen

LIST = DETAIL
Zusätzlich zu einer Summary wird die Entwicklung einer Fehlermeldung in aufeinanderfolgenden Versionen eines Lieferbestandteils ausgegeben.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.7.2 COMPARE-REPFILE - Repmengen vergleichen

Zwei REPFILEs/Produkte/Lieferungen werden verglichen und eine Differenzliste erstellt.

Format

COMPARE-REPFILE
{ { PRODUCT = productversion / lb } / { DELIVERY = productversion / lb } / { LOADER = 'loadername' [Vnnnn] / FILE = filename } WITH { { PRODUCT = { *SAME / productversion } / lb} / DELIVERY = {*SAME / productversion } / lb } / LOADER = 'loadername' [Vnnnn] / FILE = filename } [WITH [REPS] [LONG INFO] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

PRODUCT = productversion / lb
Legt die Produktversion und Lieferbestandteil fest, die zum Vergleich herangezogen werden soll.

DELIVERY = productversion / lb
Es werden die gelieferten Reps zum Vergleich herangezogen.

LOADER = 'loadername' Vnnnn
Es wird ein Lader zum Vergleich herangezogen. Der Lader kann zusammen mit der Version (von 1 ... 9999) in der Form V1 ... V9999 angegeben werden.

FILE = filename
Name der Rep-Datei (vollqualifizierter Dateiname), die zum Vergleich herangezogen werden soll.

WITH ...
Der erste WITH-Teil bestimmt die Vergleichsbasis (Produkt, Lieferung, Lader oder Rep-Datei; s.o.). *SAME bedeutet, dass die gleiche Produktversion bzw. Lieferung, die zuvor angegeben wurde, zum Vergleich herangezogen wird.

WITH [REPS] [LONG INFO]
Der zweite WITH-Teil steuert, welche Informationen in der erzeugten Liste enthalten sind. Standardmäßig sind in der Liste nur die in der neuen Rep-Auswahl geänderten Reps enthalten. Mit der Angabe LONG INFO sind auch die nicht geänderten Reps enthalten. Mit der Angabe REPS werden auch die Rep-Definitionen aufgelistet.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.7.3 CREATE-RMS-OPTIONS - RMS-Einstellungen ändern

Eine Protokolldatei kann spezifiziert und bestimmte Einstellungen für *diesen* RMS-Lauf können geändert werden.

Format

CREATE-RMS-OPTIONS
{ { PROTOCOL = { *NONE / filename[/**EXT**END] } / ALL-OPTS-ON = { NO/ YES } / NO-PRODUCT-CHECK = { NO/ YES } / NO-TYPIO = { NO/ YES } / LARGE= { NO/ YES } / BUILD-UNCHANGED= { NO/ YES } / PRODSAVE = { NO / YES / CLEAR } } [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

PROTOCOL = *NONE / filename / EXTEND

Legt fest, ob ein Protokoll des RMS-Laufs erstellt werden soll und in welche Datei das Protokoll geschrieben wird. Wenn der Dateiname mit /EXTEND erweitert wird, wird die Datei erweitert.

ALL-OPTS-ON = NO / YES
Legt fest, ob alle optionale Korrekturen, abhängig von ihrem Status, aber unabhängig von einer OPT-REP-Einstellung, in den Anlagenlader aufgenommen werden sollen.

NO-PRODUCT-CHECK = NO / YES
Legt fest, ob bei der Installation eines Anlagenladers eine Überprüfung der Produktversion von existierenden Dateien durchgeführt werden soll.

NO-TYPIO = NO / YES
Legt fest, ob bei erfolgreicher Installation einen Anlagenlader eine Meldung an der Systemkonsole ausgegeben wird.

LARGE = NO / YES
Legt fest, ob in einer Einfahraktion mehrere Korrekturmengen in das Depot übernommen werden sollen.

BUILD-UNCHANGED = NO / YES
Legt fest, ob in einem SOLIS-Lauf ein Anlagenlader erstellt und installiert wird, auch dann, wenn keine neuen Korrekturen in dem Depot aufgenommen wurden.

PRODSAVE = ON / OFF / CLEAR
Kontrolliert das Verhalten der Kommandos, die den Operanden PROD[UCT]=*PRODSAVE verwenden.

PRODSAVE = ON
Legt fest, dass RMS die Namen aller Produkte aufzeichnen soll, auf die in den folgenden Anweisungen zugegriffen wird, und diese Liste in Anweisungen weiterverwenden soll, bei denen der Operand PROD[UCT]=*PRODSAVE angegeben ist.

PRODSAVE = OFF
Schaltet die Aufzeichnung der Produkte aus, auf die von RMS-Anweisungen zugegriffen wird.

PRODSAVE = CLEAR
Löscht die Liste der Produkte, die von RMS aufgezeichnet wurde.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.7.4 CALL-FUNCTION - Modul aufrufen

Jeder in der Datei SYSOML.RMS.071 eingetragene Modul kann mit dieser Anweisung aufgerufen werden.

Format

CALL-FUNCTION
modulename [,INPUT = filename] [,ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)]

Operanden

modulename
name des Moduls in der Modulbibliothek SYSOML.RMS.071, der aufgerufen werden soll.

INPUT = filename
Name der Datei, die den Input für den aufgerufenen Modul enthält.

ERROR = ON(<integer 0..31>) / OFF(<integer 0..31>)
Im Fehlerfall wird der angegebene Auftragsschalter gesetzt (ON) oder zurückgesetzt (OFF).

12.8.8 Steueranweisungen

- END - Eingaben abschließen oder RMS im Batch-Mode beenden

12.8.8.1 END - Eingaben abschließen oder RMS im Batch-Mode beenden

Die Anweisung END beendet Eingaben von RMS-Anweisungen über *SYSDTA oder Datei und veranlasst die Ausführung der vorangegangenen Anweisungen, wenn RMS mit MODE=BATCH gestartet wurde.

Als letzte Anweisung in einer Batchprozedur beendet END das Programm RMS.

Format

END

Diese Anweisung besitzt keine Operanden.

12.9 Referenzinformation

- Verwendung der Funktionstasten
- Verwendung der P-Tasten
- Darstellung des JD
- Namenskonventionen für Eingabedateien
- Ausgabedateien
- Modulliste

12.9.1 Verwendung der Funktionstasten

Zur Bedienung von RMS werden die Funktionstasten folgendermaßen verwendet:

Taste	Bedeutung
K1	Rückzug um eine logische Stufe.
K2	BREAK.
K3	Neuaufbau des Schirms mit gleichem Inhalt.
K4	
K5 oder ESC+W	Textausgabe wechselt zwischen Deutsch und Englisch.
K6	
K7	
K8	
K9 oder ESC+?	HELP-Funktion - Online-Dokumentation zum Schirm, kann von jedem Schirm angefordert werden, mit Rückkehr zur aktuellen Bearbeitung.
K10	
K11 oder ESC+=	Die Sortierreihenfolge der Auswahlbildschirme kann in absteigend bzw. aufsteigend geändert werden.
K12 oder ESC+<	Es wird die Funktion Dialoginformation angesprungen und die laufende Funktion wird unterbrochen. Rückkehr mit K1-Taste.
K13	
K14 oder ESC+<	beendet die laufende Funktion oder Programm.
K15	
F1	Diese Taste wird funktionsspezifisch verwendet und ist auf den jeweiligen Bildschirmen beschrieben.
F2	Diese Taste wird funktionsspezifisch verwendet und ist auf den jeweiligen Bildschirmen beschrieben.
F3	Diese Taste wird funktionsspezifisch verwendet und ist auf den jeweiligen Bildschirmen beschrieben.
F4 oder ESC+^	Diese Taste wird funktionsspezifisch verwendet und ist auf den jeweiligen Bildschirmen beschrieben.

12.9.2 Verwendung der P-Tasten

Zur Erleichterung der Funktionsauswahl können die P-Tasten geladen und wahlweise verwendet werden:

P1 ---> Auswahl Funktion 1

. . .

P12 ---> Auswahl Funktion 12

P13 ---> positioniert auf das letzte markierbare Feld

P18 ---> Hardcopy des gesamten Bildschirm (falls Drucker-Anschluss vorhanden)

12.9.3 Darstellung des JD

***** vor JD	Diese Korrektur wurde mit der Notizbuchaktion ADD aus einer externen Datei in das Depot übernommen.
Blank vor JD	Diese Korrektur wurde über die Funktion Einfahren in das Depot übernommen.
- nach JD	Diese Korrektur ist auf Grund einer Notizbuchaktion (REJ) für keinen LB mehr gültig.
+ nach JD	Diese Korrektur ist auf Grund einer Notizbuchaktion (ADD) für alle LB gültig.
Blank nach JD	Keine Notizbuchaktion. Diese Korrektur ist für die Lieferbestandteile gültig, mit denen sie eingefahren wurde.

12.9.4 Namenskonventionen für Eingabedateien

Die Namen für die Eingabedateien entsprechen den BS2000-Konventionen.
Für alle Bildschirme, auf denen eine Eingabedatei verlangt wird, kann der Dateiname in der Wildcard-Syntax angegeben werden. Diese Syntax entspricht dem SDF-Datentyp *filename* mit dem Zusatz *with-wild* (siehe SDF-Metasyntax im Handbuch „Kommandos“ [1]).

Gibt es mehrere Dateien, die der angegebenen Auswahl entsprechen, wird eine Übersicht ausgegeben. Die gewünschte Datei ist in der Übersicht zu markieren.

Entspricht nur eine Datei den Kriterien, wird automatisch diese Datei verarbeitet.

12.9.5 Ausgabedateien

Folgende Funktionen erstellen Dateien mit den angegebenen Namen:

Funktion	Dateiname
DIALOG-INFORMATION, Ausgabe Textinhalt	TXT.subno.text-ebenen-namen
DIALOG-INFORMATION, Ausgabe Korrekturinhalt	REP.subno.jd.produkt.version
EXPORT - Produkte/Laderdefinitionen	RMS.DEPOT.jjmmtt.hhmmss
EXPORT - Textebene	RMS.TEXTEBENE.name
LADERGENERIERUNG - Installieren	installationsname
LADERGENERIERUNG - Generieren	anlagenlader-definition.version
LIEFERUNGEN ERSTELLEN	SYSRMS.produkt.version.lb-vers
LIEFERUNGEN ERSTELLEN, bei fehlenden Texten	TEXTNR.FEHL.produkt.version.lb-vers
LIEFERUNGEN ERSTELLEN, Originallieferung	LIEFER.produkt.version.lb-vers.ORG
NOTIZBUCHLISTE, ausgewählte Produkte	NBK.ALLE.EINTRAEGE.produkt.vers
VERGLEICH - Erstellen aktuelle Repmenge	produkt.version.lb-vers
KORREKTUR-GESAMTUEBERSICHT (MATRIX)	#MATRIX.LIST.hhmmss
LIEFERUNGEN übernehmen	EINFAHR.LISTE.hhmmss
NOTIZBUCHLISTE - Gesamtliste	#NOTBUCH.LIST.hhmmss
NOTIZBUCHLISTE, ausgewählte Korrekturnummer	NBK.subno.jd.produkt.vers
VERGLEICH - Menge mit Menge	#hh-mm-ss.LSTVGL

12.9.6 Modulliste

Zentrale Module

RMS	Enthält Basisroutinen, interaktive Informationsfunktion, Pascal-XT Laufzeitsystem und FHS-Laufzeitsystem.
RMSCONV	Routine, um ein Depot ab V3.20 auf den jetzigen Stand umzusetzen.
RMSHELP	Enthält Hintergrundinformationen über Bildschirme in Deutsch.
RMSHELPE	Enthält Hintergrundinformationen über Bildschirme in Englisch.
RMSINIT	Führt alle Initialisierungstätigkeiten durch und ruft, falls notwendig, RMSCONV bzw. RMSIN2 auf.
RMSIN2	Wird aufgerufen, falls ein Depot-Update (Einfahren, Notizbucherfassung oder Import) nicht ordnungsgemäß abgeschlossen wurde. Der Zustand vor Aufruf der Funktion wird wiederhergestellt.
RMSMONT	Enthält alle Laderbau-Routinen.
RMSSYN	Enthält alle Batch/SOLIS2-spezifischen Routinen.
RMS7SYN	Enthält alle Batch/SOLIS2-spezifischen Routinen für die V7-Syntax.

Funktionsmodule, die den MODE MEISTER benötigen

EINFAHR	Übernahme SOLIS2- bzw. PULS-Repmengen.
LIEFERN	Generierung Lieferbestandteile für SOLIS2.
LIEFER7	Generierung Lieferbestandteile für SOLIS2 (neue Funktion).
MATRIX	Erstellung Querverweisliste der Korrekturen.
MONTAGE	Generierung bzw. Installation aller zu bauenden Anlagenlader.
NOTIZERF	Notizbucherfassung.
NBKINFO	Notizbuchinformationen ausgeben.
NOTBLIST	Gesamtliste alle Notizbucheinträge.
RMSEDIT	Einrichten neuer Textebene, Einrichten, Ändern und Löschen Texteinträge in Textebenen bzw. Produktnotizen.

RMSEXIM	Export (plus ggf. Löschen) von Mengen aus dem Depot.Mengen sind Produkte, Laderdefinitionen, OPT-Rep Einstellungen, Ladermodifikationen und Textebenen. Die gesamte Information wird in einem neuen DEPOT gespeichert. *Import von Mengen aus einem anderen DEPOT* Für Produkte, besteht die Möglichkeit, • alles zu übernehmen • alles ab dem zuletzt eingefahrenen und gelieferten LB • alle eingefahrenen LBs und den neuesten Lieferkorrekturstand
RMSFIND	Zeichenorientiertes Durchsuchen des Depots, modulorientiertes Durchsuchen aller Korrekturen.
RMSLGEN	Anschluss an RMSMONT für Laderauswahl
RMSLVWIN	Anschluss an RMSMONT für Lader-Administration und Information.
RMSOPTS	Einstellung Optionen für dieser Lauf.
RMSST00	Gibt eine Statistik über den Depotinhalt aus.
VGL	REP-Vergleiche durchführen - neues Format.
VGLSTART	REP-Vergleiche durchführen.

Dienstprogrammanschlussmodule

Diese Module beinhalten nur den Anschluss der Dienstprogramme. Der Anwender muss die entsprechende Bibliothek installiert haben.

CALLEDOR	Ruft EDOR auf.
CALLEDT	Ruft EDT in Abhängigkeit vom Menüparameter (TEMPORARY oder RESIDENT) auf.
CALLLMS	Ruft LMS auf.

Sonderfunktionsmodule

DEFKUDEP	Konvertiert ein Service-Depot in ein Kundendepot.
RMSCNGPR	Ändert einen Produktversion (bei Kundenfreigabe).
RMSQM001	Erstellt ein OPT-REP-Paket aus einer Definitionsdatei.

12.10 Anweisungen von RMS vor V7.0

Dialogfunktion		Anweisungsfunktion
DIALOG-INFORMATION		Nein
FUNKTIONALE-INFORMATIONEN und ÜBERSICHTEN		1
KORREKTURVERWALTUNG		
	Übernahme von Vorab-Korrekturen	2
	Aktivieren Depot-Korrekturen	2
	Korrekturen löschen	2
	Zurücksetzen von Notizbuchaktionen	2
	Informationsmöglichkeiten	2
LADERVERWALTUNG		
	Definieren Anlagenlader	3
	OPT-Rep Einstellung	4
	Automatische OPT-Rep Einstellung	Nein
	Lokale Lader-Modifikation	4
	Ändern und löschen von Definitionen	Nein
LADERGENERIERUNG und INSTALLATION		
	Lader-Generierung	Ja
	Erstellung von alten Ladern	Nein
	Laderinstallation	Ja
ERSTELLUNG von LIEFERUNGEN		
	Erstellung von aktuellen Lieferungen	Ja
	Erstellung von Originallieferungen	Ja
	Lieferbestandteil-Versionsnummer	5
ÜBERNAHME von LIEFERUNGEN		Ja
TEXTVERARBEITUNG		
	Textebenen	6
	Produktnotizen	6

	Korrekturdokumentation	6
	Texteditor	Nein
INSTALLATION und DEPOTVERWALTUNG		
	EXPORT-Funktion	Nein
	IMPORT-Funktion	Nein

Erklärung:

Ja	Diese Funktion ist als Anweisung uneingeschränkt möglich.
Nein	Diese Funktion ist nur im Dialog anwendbar.
1	Bei den Anweisungsfunktionen werden spezielle Informations- und Übersichtslisten ausgegeben.
2	Die im Dialog eingegebenen Aktionen werden von allen Anweisungen berücksichtigt.
3	Die im Dialog definierten Einstellungen werden von der Anweisung LADERBAU berücksichtigt. Sind keine Definitionen vorhanden und wurde zusätzlich der Parameter AUTODEF=YES angegeben, wird automatisch eine Standarddefinition angelegt.
4	Die im Dialog definierten Einstellungen werden von der Funktion LADERBAU berücksichtigt.
5	In der Anwendergruppe 2 und 3 wird die Versionsnummer einer Lieferung im Dialog- und Anweisungs-Modus automatisch vergeben.
6	Im Anweisungs-Modus ist die Übernahme und Lieferung von Textebenen möglich.

12.10.1 Beispiele

Die Funktionen sind auch im Dialog (Line-Modus) möglich. Dazu muss die Standard-Startprozedur wie folgt aufgerufen werden:

`/CALL-PROC SYSPRC.RMS.071,(MODE=BATCH)`

Durch die Kopplung einzelner Anweisungen können Jobs erstellt werden, die bestimmte Anwendungen automatisch durchführen.

Beispiel 1

Alle Daten der `LIEFERDATEI` werden in das Depot übernommen. Notizbucheinträge werden berücksichtigt.

Die angegebene `LISTDATEI` wird um das Einfahrprotokoll erweitert.

Im Fehlerfall wird der Auftragsschalter 30 auf ON gesetzt.

```
FUNKTION=EINFAHR,-
  INPUT=LIEFERDATEI,-
  LIST=LISTDATEI/EXT,-
  ERROR=ON(30)
```

Für alle Produkte, die in das Depot übernommen werden, werden einsatzfähige Lader generiert. Vorhandene Anlagendefinitionen werden berücksichtigt.

Bei fehlenden Definitionen werden Standarddefinitionen verwendet und eingetragen.

Im Fehlerfall wird der Auftragsschalter 31 auf ON gesetzt.

```
FUNKTION=LADERBAU,-
INSTALLATION=DEF,-
  AUTODEF=YES,-
  ERROR=ON(31)
ENDE
```

Beispiel 2

Es werden die Lader generiert, wie diese in den Anlagenladerdefinitionen mit den Namen `S090REP ZE1` und `S090REP ZE2` definiert wurden.

```
FUNKTION=LADERBAU,-
  NAME='S090REP ZE1',-
  INSTALLATION=DEF
FUNKTION=LADERBAU,-
  NAME='S090REP ZE2',-
  INSTALLATION=DEF
```

Für alle Anlagenladerdefinitionen, in denen SPOOL V4.9A definiert ist, werden Lader generiert und ggf. installiert.

Bei fehlenden Definitionen wird eine Standarddefinition verwendet und eingetragen (AUTODEF=YES).

```
FUNKTION=LADERBAU,-
  PROD=SPOOL,VERS=4.9A,-
  AUTODEF=YES,-
  INSTALLATION=DEF
ENDE
```

12.10.2 Beschreibung der Anweisungen

Allgemeine Definitionen

pathname	[:catid:][$userid.]filename
prodname	Produktfamilienname aus PULS (maximal 15-stellig)
prodvers	Version des Produkts, ohne V und ohne die führende 0, aber mit Angabe des Punktes
loaderdefname	Name der Anlagenladerdefinition (maximal 22-stellig)

RMS-PARAMETER-Anweisung (ab V7.0: CREATE-RMS-OPTIONS)

In die Protokolldatei werden die angegebenen Anweisungen und Ausgabelisten der aufgerufenen Funktionen druckaufbereitet geschrieben.

FUNKTION=RMS-PARAMETER
,PROT[OKOLL]= { *NONE / filename }[/EXT[END]

PROTOKOLL
Ein Protokolldatei mit dem Namen `filename` soll erstellt werden.

EXTEND
Die Datei soll nicht neu erstellt, sondern fortgeschrieben werden.

***NONE**
Schließt die eröffnete Protokolldatei

EINFAHR-Anweisung (ab V7.0: INPUT-DELIVERY-PACKET)

FUNKTION=EINFAHR
,{ DATEI / INPUT / FILE }=pathname[,L[IST]=pathname[/EXT[END]]]

LIST
Es soll eine von der Protokolldatei getrennte Liste des Einfahr-Vorgangs erstellt werden.

EXTEND
Die Datei soll nicht neu erstellt, sondern fortgeschrieben werden.

LIEFER-Anweisung (ab V7.0: CREATE-DELIVERY-PACKET)

FUNKTION=LIEFER
[,PROD=prodname,VERS=prodvers [,LB={ * / - / libvers }]][,{ DATEI / OUTPUT / FILE }={ *STD / *SAME / filename }] [,TEXT= { *NONE / 'textname' } =,UMFANG= { ALL / lbvers0 }]

LADERBAU-Anweisung (ab V7.0: SELECT-REPFILE-TO-BUILD)

FUNKTION=LADERBAU
[,PROD=proname,VERS=prodvers][,NAME='loaderdefname'] [,AUTODEF=YES[,AUTOOPT='optpacketname']] [,INSTALL[ATION]={ ABS / DEF / NO }][,USERID= { *OWN / userid }][,PRE[FIX]= { *NONE / prefix }]

BAULADER-Anweisung (ab V7.0: BUILD-REPFILE)

FUNKTION=BAULADER

ENDE-Anweisung (ab V7.0: END)

E[NDE] / H[ALT]

13 SIR Pubsets einrichten

Versionsstand:	**SIR V21.0C** (<ver>=210)
Privilegierung:	**STD-PROCESSING** (für nicht-privilegierte Funktionen) **TSOS** (für privilegierte Funktionen))

Das Dienstprogramm SIR (System Install and Restore) dient zum Einrichten eines Pubsets.

Außerdem bietet SIR die Möglichkeit, Platten zu initialisieren (Dienstprogramm VOLIN).

Produktstruktur

Die Produktstruktur von SIR ist zweiteilig. Der Funktionsumfang, den der Anwender nutzen kann, wird vom Privileg TSOS bestimmt.

- Läuft das Dienstprogramm SIR unter der Benutzerkennung TSOS, steht dem Anwender die vollständige Funktionsbreite von SIR zur Verfügung, Pubsets einzurichten und zu erweitern, sowie die Funktionen zur Software-Installation. Im Weiteren wird dieser Funktionsumfang „privilegierter SIR“ genannt.
- Läuft das Dienstprogramm SIR **nicht** unter der Benutzerkennung TSOS, stehen dem Anwender nur die Funktionen zum Übertragen von Dateien zwischen Pubsets und Privatplatten zur Verfügung.
  Im Weiteren wird dieser Funktionsumfang „nicht privilegierter SIR“ genannt.

Funktionen

Die wesentlichen Funktionen für den Anwender des privilegierten SIR sind:

- Initialisierung von Platten (vollständige VOLIN-Funktion) Diese Funktion wird im Abschnitt „Volume-Sets“ beschrieben.
- Einrichten beliebiger Volume-Sets
- Erweitern bestehender Volume-Sets
- Einrichten beliebiger Pubsets
- Erweitern eines bestehenden Pubset
- Einrichten des Home-Pubsets bei der Installation von BS2000
- Übertragen von Dateien zwischen Pubsets und Privatplatten

Mit dem Dienstprogramm SIR können Home-Pubsets für den Ablauf unter dem aktuellen BS2000 (Zielversion) eingerichtet und ein Versionswechsel auf die aktuelle Version von BS2000 (Zielversion) aus unterschiedlichen Ablaufversionen von SIR durchgeführt werden.

Generell ist eine SIR-Version an eine bestimmte Version von BS2000 gekoppelt. Das bedeutet, dass stets gelten muss: Ablaufversion <= Zielversion.

Mit SIR V21.0C kann ein Home-Pubset für BS2000 V21.0C eingerichtet und ein Versionswechsel auf BS2000 V21.0C (Zielversion) aus den **Ablaufversionen BS2000 V21.0A/B/C** ausgeführt werden. Zusätzlich kann SIR V21.0C Urladeplatten für BS2000 V21.0A/B neu erstellen bzw. aktualisieren.

Mit SIR V21.0C kann auf x86-Servern ein (weiteres) Home-Pubset für BS2000 auf x86-Servern eingerichtet werden.

13.1 Einrichten von Pubsets mit SIR

Ein **SINGLE-FEATURE-Pubset** (SF-Pubset) besitzt eine Pubset-Id (syntaktisch gleich der Katalogkennung) und besteht aus maximal 255 Platten. Das Format aller Platten muss identisch sein bezüglich Allocation Unit, Formatierung, PAM-Key-Nutzung usw. Dem SF-Pubset werden bestimmte Eigenschaften, z.B. hoch verfügbar, zugeordnet, die für alle Dateien innerhalb des Pubset gelten.

Mit dem privilegierten SIR kann auch ein **SYSTEM-MANAGED-Pubset** (SM-Pubset) oder ein Volume-Set eingerichtet werden. Für den Benutzer erscheint ein SM-Pubset nahezu identisch wie ein SF-Pubset. Die Unterschiede der beiden Pubsettypen sind vor allem von der Systemverwaltung zu berücksichtigen.

Im Gegensatz zum SF-Pubset besteht ein SM-Pubset aus mehreren Volume-Sets. Jedes Volume-Set besitzt wie ein SF-Pubset ein bestimmtes Format und bestimmte Eigenschaften. Die Formate und Eigenschaften der verschiedenen Volume-Sets, die zu einem SM-Pubset gehören, können unterschiedlich sein. Das SM-Pubset wird vom Benutzer wie ein SF-Pubset nur über eine Katalogkennung angesprochen. Auf welches Volume-Set eine Datei abgelegt wird, wird nun durch Angabe bestimmter Dateiattribute erreicht, so dass in einem SM-Pubset Dateien mit verschiedenen Anforderungen bezüglich Verfügbarkeit und Performance angelegt werden können.

SM-Pubsets können nicht als Home-Pubset verwendet werden. Deshalb werden alle Anweisungen des SIR, die ausschließlich zum Einrichten eines Home-Pubset benötigt werden, bei SM-Pubsets abgewiesen.

Dynamische Pubset-Rekonfiguration

Die beim Einrichten festgelegte Pubset-Konfiguration kann während der „Lebensdauer“ des Pubset geändert (rekonfiguriert) werden.

Über verschiedene Kommandoschnittstellen der Pubset-Rekonfiguration können Konfigurationsdaten sowie Platten- und Volume-Set-Konfiguration eines Pubset dynamisch rekonfiguriert werden, d.h. während der Pubset in Betrieb ist. Näheres zur dynamischen Pubset-Rekonfiguration siehe Handbuch „Systembetreuung“ [5].

13.1.1 Volume-Sets

Einrichten eines Volume-Set

Zunächst werden die Platten für das Volume-Set initialisiert (VOLIN-Funktion) und in der Volres (Systemplatte des Volume-Set) eingetragen. Anschließend wird ein Katalogextent auf der Volres eingerichtet.

Folgende Anweisungen stehen für die Funktion „Einrichten eines Volume-Set“ zur Verfügung:

```
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*INSTALL
CREATE-VOLUME ...
CREATE-CATALOG 0,nnnn
END-VOLUME-SET-DECLARATION
```

Die Anweisung BEGIN-VOLUME-SET-DECLARATION leitet die Deklaration des einzurichtenden Volume-Set ein. Mit END-VOLUME-SET-DECLARATION wird die Deklaration beendet. Sollen in einem SIR-Lauf mehrere Volume-Sets eingerichtet werden, kann dieser Anweisungsblock für jedes einzurichtende Volume-Set wiederholt werden.

Die Anweisung CREATE-VOLUME muss für jede zum Volume-Set gehörende Platte angegeben werden, damit diese im SVL der Volres hinterlegt werden kann. Für die Volres kann eine CREATE-CATALOG-Anweisung gegeben werden. Ohne die Anweisung wird standardmäßig ein Dateikatalog mit 2000 PAM-Seiten angelegt, wobei seine Größe auf ein Vielfaches der Allocation Unit aufgerundet wird.

Die Funktion unterstützt das Einrichten von Volume-Sets mit Platten, deren Blockgröße 4K beträgt.

Das Einrichten und Erweitern von „freien“ Volume-Sets, d.h. Volume-Sets, die noch nicht zu einem SM-Pubset gehören, wird im Wesentlichen für die Pubset-Rekonfiguration benötigt.

Beispiel

```
/START-SIR
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=AB,
        ACTION=*INSTALL(FORMAT=*NK(*4K(8)))                              1.
//CREATE-VOLUME DISK-NUMBER=000(OVERWRITE-DISK=*ANY,UNIT=MA),
        FORMAT=*YES                                                      2.
//CREATE-CATALOG DISK-NUMBER=000,FILE-SIZE=160                           3.
//END-VOLUME-SET-DECLARATION                                             4.
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=X,
        ACTION=*INSTALL(FORMAT=*K,AVAILABILITY=*NORMAL)                  5.
//CREATE-VOLUME DISK-NUMBER=*RANGE(00,04,OVERWRITE-DISK=*ANY,
        UNIT=(MN00,MN01,MN02,MN03,MN04)),
        FORMAT=*NO                                                       6.
//CREATE-CATALOG DISK-NUMBER=00,FILE-SIZE=500                            7.
//END-VOLUME-SET-DECLARATION                                             8.
//END
```

1. Ein Volume-Set mit der Kennung AB und dem Plattenformat NK4 mit einer Allokierungseinheit von 8 KB soll eingerichtet werden.
2. Es wird die erste Platte des Volume-Sets AB mit der VSN AB.000 ohne vorherige VSN-Prüfung initialisiert und mit einem bestimmten Bitmuster überschrieben.
3. Für die eben initialisierte Platte 000 wird ein Katalogabschnitt eingerichtet. Er hat eine Größe von 160 PAM-Seiten.

4. Die Deklaration für den Volume-Set AB wird beendet. Ein solcher BEGIN-/END-Anweisungsblock kann in einem SIR-Lauf beliebig oft wiederholt werden.
5. Ein weiterer Volume-Set mit der Kennung X und dem Plattenformat K (d.h. mit einer Allokierungseinheit von 6 KB) soll eingerichtet werden. Der Volume-Set soll ohne Spiegelplatten eingerichtet werden (normale Verfügbarkeit).
6. Es werden fünf Platten für den Volume-Set X ohne vorherige VSN-Prüfung initialisiert. Diese Platten bekommen die VSN PUBX00 bis PUBX04.
7. Für die eben initialisierte Platte 00 wird ein Katalogabschnitt eingerichtet. Er hat eine Größe von 500 PAM-Seiten.
8. Die Deklaration für den Volume-Set X wird beendet.

Erweitern eines Volume-Set

Ein „freies“ Volume-Set, d.h. es gehört noch keinem SM-Pubset an, kann um zusätzliche Platten erweitert werden.

Folgende Anweisungen stehen für die Funktion „Erweitern eines Volume-Set“ zur Verfügung:

```
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*EXTEND
CREATE-VOLUME ...
END-VOLUME-SET-DECLARATION
```

Die Anweisung CREATE-VOLUME muss für jede neu zum Volume-Set hinzukommende Platte angegeben werden, damit diese im SVL der Volres hinterlegt werden kann.

Die Anweisung CREATE-CATALOG ist für diese Funktion nicht zugelassen.

Das Erweitern eines Volume-Set, das bereits zu einem SM-Pubset gehört, ist mit der Funktion „Erweitern eines SM-Pubset“ (siehe "SM-Pubsets") möglich.

Beispiel

```
/START-SIR
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=AB,ACTION=*EXTEND            1.
//CREATE-VOLUME DISK-NUMBER=*RANGE(001,002,OVERWRITE-DISK=*ANY,
        UNIT=(MP(SUBUNIT=MQ),MR(SUBUNIT=MS))),
        FORMAT=*YES                                                   2.
//END-VOLUME-SET-DECLARATION
//END
```

1. Der Volume-Set mit der Kennung AB soll erweitert werden.
2. Zu diesem Volume-Set gehören nun auch eine zweite und dritte Platte mit den VSNs AB.001 und AB.002.

13.1.2 SM-Pubsets

Einrichten eines SM-Pubset

Volume-Sets werden mit dieser Funktion zu einem SM-Pubset zusammengefasst.

Das Einrichten erfolgt SIR-intern in zwei Schritten:

1. Für jedes Volume-Set, das im selben SIR-Lauf neu eingerichtet werden soll, werden zuerst die entsprechenden Platten initialisiert und formatiert. Danach werden in die SVLs der Volres der einzelnen Volume-Sets die zugehörigen Platten eingetragen und dadurch die Volume-Sets gebildet. Anschließend wird auf den Volres ein Dateikatalog eingerichtet. Auf dem Control-Volume-Set, das das ausgezeichnete Volume-Set des SM-Pubset ist, werden weitere Spezialdateien eingerichtet.
2. Aus den im ersten Schritt eingerichteten Volume-Sets oder auch aus Volume-Sets, die bereits in früheren SIR-Läufen eingerichtet wurden, wird ein SM-Pubset eingerichtet.

   Das Volume-Set, das zum Control-Volume-Set werden soll, muss immer im selben SIR-Lauf neu eingerichtet werden, da sich seine Struktur von der Struktur eines einfachen Volume-Set unterscheidet.

Folgende Anweisungen stehen für die Funktion „Einrichten eines SM-Pubset“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-M(ACTION=*INSTALL)
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*ADD / *INSTALL
CREATE-VOLUME ...
CREATE-CATALOG ...
 ...
END-VOLUME-SET-DECLARATION
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*ADD / *INSTALL
 ...
END-VOLUME-SET-DECLARATION
COPY ...
 ...
END
```

Für die Deklaration der einzelnen Volume-Sets gelten beim Einrichten eines SM-Pubset die gleichen Bedingungen wie beim „Einrichten eines Volume-Set“. Jedoch können beim Einrichten eines SM-Pubset auch Katalogextents einzelner Volume-Sets auf Platten ungleich der Volres eingerichtet werden.
Wenn für ein einzurichtendes Volume-Set keine CREATE-CATALOG-Anweisung angegeben wird, wird automatisch auf der Volres ein Katalogextent von 2000 PAM-Seiten angelegt, wobei auf ein Vielfaches der Allocation Unit aufgerundet wird.

Bei der Deklaration eines Control-Volume-Set muss in der BEGIN-VOLUME-SET-DECLARATION-Anweisung ACTION=*INSTALL angegeben werden.

COPY-Anweisungen sind nur außerhalb des Anweisungsblocks BEGIN-/END-VOLUME-SET-DECLARATION erlaubt. Die Lage der zu kopierenden Dateien innerhalb des SM-Pubset wird durch die Performance-Attribute der einzelnen Quell-Dateien festgelegt.

Folgende Anweisungen sind beim Einrichten eines SM-Pubset nicht zugelassen, da ein SM-Pubset nicht als Home-Pubset verwendet werden darf:

`CREATE-SNAP-FILE, CREATE-IPL-VOLUME, MODIFY-IPL-VOLUME`

Beispiel

```
/START-SIR
//DECLARE-PUBSET PUBSET-TYPE=*SYSTEM-MANAGED(PUBSET=SMP1,
        ACTION=*INSTALL(CONTROL-VOLUME-SET=XYZ))                        1.
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=AB,
        ACTION=*ADD(VOLUME-SET-ATTR=*PAR(
          USAGE=*STD,AVAILABILITY=*HIGH,PERFORMANCE-ATTR=*PAR(
            PERFORMANCE=*HIGH,WRITE-CONSISTENCY=*IMMEDIATE)))           2.
//END-VOLUME-SET-DECLARATION
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=XYZ,
        ACTION=*INSTALL(FORMAT=*K)                                      3.
//CREATE-VOLUME DISK-NUMBER=00(OVERWRITE-DISK=*ANY,UNIT=XY(SUBUNIT=XZ)),
        FORMAT=*YES                                                     4.
//CREATE-CATALOG DISK-NUMBER=00,FILE-SIZE=96                            4.
//END-VOLUME-SET-DECLARATION
//END
```

1. Es wird ein SM-Pubset mit der Katalogkennung (catid) SMP1 definiert. Als Control-Volume-Set wird der Volume-Set mit der Kennung XYZ benannt.
2. Der (bereits in einem früheren SIR-Lauf) vordefinierte Volume-Set mit der Kennung AB soll in das SM-Pubset mit der Kennung SMP1 aufgenommen werden.
   Folgende logische Eigenschaften werden für ihn definiert: Er soll als Ablageort für Standarddateien genutzt werden sowie hohe Ausfallsicherheit und Performance bieten. Die Konsistenz der Schreibzugriffe soll unverzüglich hergestellt werden.
3. Danach wird ein Volume-Set mit der Kennung XYZ eingerichtet.
   Aus (1) ist bekannt, dass dieser Volume-Set der Control-Volume-Set sein soll.
4. Es wird eine Platte bestimmt (VSN XYZ.00) und ein Bereich für den Katalog festgelegt.

Erweitern eines SM-Pubset

Ein SM-Pubset bzw. Volume-Sets, die zu einem SM-Pubset gehören, können um Volume-Sets bzw. Platten erweitert werden.

Das Erweitern eines SM-Pubset erfolgt wie das „Einrichten eines SM-Pubset“ SIR-intern in zwei Schritten:

1. Die angegebenen Volume-Sets werden gemäß den CREATE-VOLUME-Anweisungen um Platten erweitert. Dazu werden die neu hinzukommenden Platten zunächst initialisiert und formatiert und anschließend in der Volres der einzelnen Volume-Sets eingetragen. Das Anlegen von Katalogextents wird dabei nicht unterstützt.
2. Im zweiten Schritt werden zusätzliche Volume-Sets zum SM-Pubset hinzugenommen und die COPY-Anweisungen ausgeführt.

Folgende Anweisungen stehen für die Funktion „Erweitern eines SM-Pubset“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-M(ACTION=*EXTEND)
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*ADD / *INSTALL / *EXTEND
CREATE-VOLUME ...
 ...
END-VOLUME-SET-DECLARATION
BEGIN-VOLUME-SET-DECLARATION ...,ACTION=*ADD / *INSTALL / *EXTEND
 ...
END-VOLUME-SET-DECLARATION
COPY ...
 ...
END
```

Hinweise

- Volume-Sets, die mit ACTION=*ADD / *INSTALL bearbeitet werden sollen, dürfen nicht im Pubset existieren.
- Volume-Sets, die mit ACTION=*ADD bearbeitet werden sollen, dürfen keinem anderen Pubset angehören.
- Bei ACTION=*EXTEND muss das Volume-Set bereits Bestandteil des zu bearbeitenden Pubset sein.
- Ein Volume-Set kann nur erweitert werden, wenn es sich in einem konsistenten Zustand befindet.

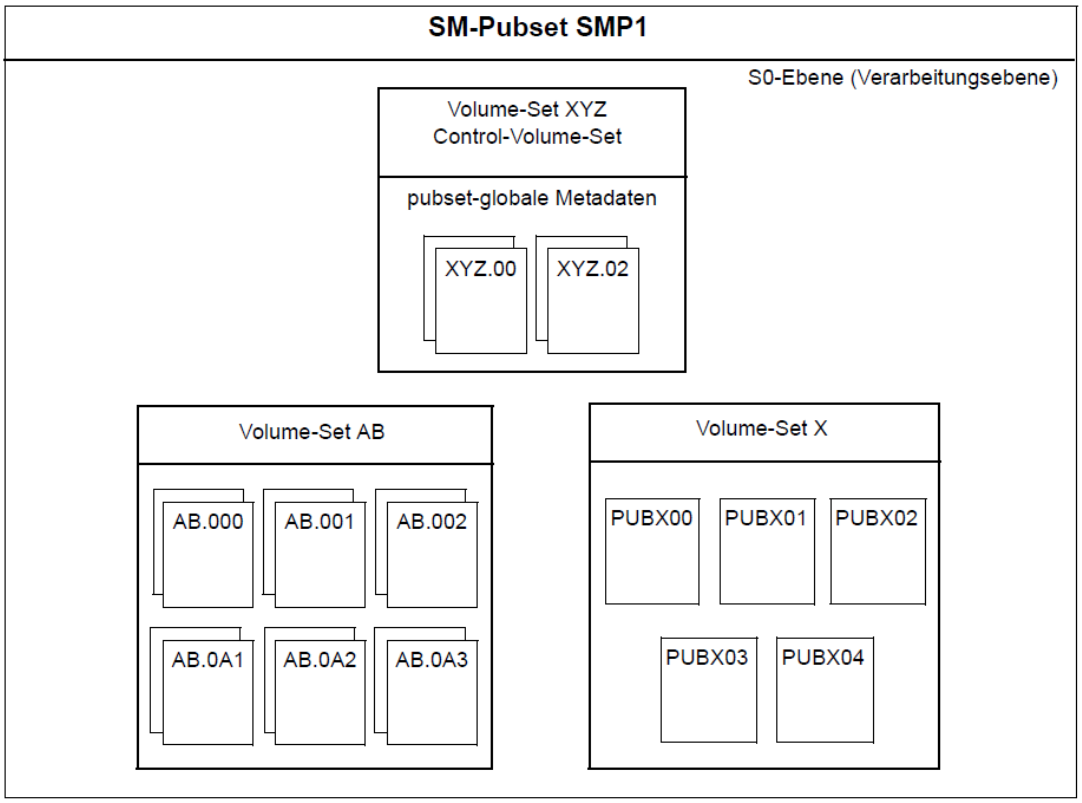
Beispiel

```
/START-SIR
//DECLARE-PUBSET PUBSET-TYPE=*SYSTEM-MANAGED(PUBSET=SMP1,
        ACTION=*EXTEND(CONTROL-VOLUME-SET=XYZ))                          1.
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=X,
        ACTION=*ADD(VOLUME-SET-ATTR=*PAR(PERFORMANCE-ATTR=*PAR(
          PERFORMANCE=*VERY-HIGH,WRITE-CONSISTENCY=*BY-CLOSE)))          2.
//END-VOLUME-SET-DECLARATION
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=AB,
        ACTION=*EXTEND                                                   3.
//CREATE-VOLUME DISK-NUMBER=*RANGE(0A1,0A3,OVERWRITE-DISK=*ANY,
        UNIT=(MT(SUBUNIT=MU),MV(SUBUNIT=MW),MX(SUBUNIT=MY))),
        FORMAT=*YES                                                      4.
//END-VOLUME-SET-DECLARATION
//BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=XYZ,
        ACTION=*EXTEND                                                   5,
//CREATE-VOLUME DISK-NUMBER=02(OVERWRITE-DISK=*ANY,UNIT=Y1(SUBUNIT=Y2)),
        FORMAT=*YES                                                      6.
//END-VOLUME-SET-DECLARATION
:
//COPY FILE=:SF01:SYSMES.(NEW=SYSMES.)                                   7.
//COPY FILE=ARCHIV.(NEW=*PREFIX(REMOVE=ARCHIV)),FROM=*PUBSET             8.
//END
```

1. Der SM-Pubset SMP1 soll erweitert werden.
2. Der (bereits vordefinierte) Volume-Set X (siehe "Volume-Sets") soll in den SM-Pubset aufgenommen werden. Seine logischen Eigenschaften werden wie folgt definiert: Er soll als Ablageort für Standarddateien genutzt werden, keine erhöhte Ausfallsicherheit, aber eine sehr hohe Performance bieten. Die Konsistenz der Schreibzugriffe soll erst beim Schließen der Datei hergestellt werden.
3. Der Volume-Set AB soll erweitert werden.

4. Zu diesem Volume-Set gehören nun auch eine 4., 5. und 6. Platte mit den VSNs AB.0A1, AB.0A2 und AB.0A3.
5. Der Control-Volume-Set XYZ soll erweitert werden.
6. Zu diesem Volume-Set gehört nun auch eine 2. Platte mit der VSN XYZ.02.
7. Alle Dateien des Pubsets SF01, deren Dateiname mit dem Teilnamen „SYSMES“ beginnt, werden unter Beibehaltung ihres Dateinamens auf Platten des bei der DECLARE-PUBSET-Anweisung (1) angegebenen SM-Pubsets SMP1 kopiert.
8. Aus dem Default-Pubset sollen alle Dateien, deren Namen mit dem Teilnamen „ARCHIV“ beginnen, auf SMP1 kopiert werden. Genau dieser Präfix „ARCHIV“ soll dabei entfallen.

Konfiguration des SM-Pubsets SMP1 nach Abarbeitung der Beispiele



Ein von SIR generierter Pubset bietet der Systembetreuung die Basis für weitere Steuerungsmaßnahmen. Ihre Reihenfolge und zeitlicher Ablauf ist nicht fest vorgegeben. Sie müssen nicht zwingend beim Einrichten des Pubsets erfolgen, sondern können auch erst bei Bedarf im Rahmen der Pubset-Pflege vorgenommen werden. Um die bei der Generierung nicht beeinflussbaren Voreinstellungen (z.B. Nutzungseinschränkungen, Cache-Zuordnungen, Sättigungsschwellwerte) zu ändern, ist es empfehlenswert, den Pubset unmittelbar im Anschluss nachzubehandeln. Dabei kann auch der Teil der Pubset-Umgebung erzeugt werden, der bei der Generierung unberücksichtigt bleibt (Storage-Klassen, Management-Klassen, GUARDS-Profile, HSMS-Umgebung, usw.). Für die Nachbehandlung muss der Pubset in Betrieb sein. Eine detaillierte Beschreibung zur Nachbehandlung von neu generierten SM-Pubsets ist im Handbuch „SMS“ [8] zu finden.

13.1.3 SF-Pubsets

Einrichten eines SF-Pubset

Ein SF-Pubset (SINGLE-FEATURE-Pubset) wird so eingerichtet, wie ein Pubset bisher eingerichtet wurde. D.h., die Platten werden mit der VSN initialisiert die Archivnummern und Gerätetypen der zum Pubset gehörenden Platten werden im SVL der Systemplatte des Pubsets (Pubres) verankert und ein Dateikatalog für den Pubset festgelegt.

Folgende Anweisungen stehen für die Funktion „Einrichten eines SF-Pubset“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(...,ACTION=*INSTALL)
CREATE-VOLUME
CREATE-CATALOG
CREATE-PAGING-FILE
```

Wenn keine CREATE-CATALOG-Anweisung angegeben wird, wird standardmäßig auf der Pubres ein Dateikatalog in der Größe von mindestens 2000 PAM-Seiten angelegt, wobei auf ein Vielfaches der Allocation-Unit-Größe aufgerundet wird.

Die Anweisung CREATE-VOLUME muss für jede Platte des einzurichtenden Pubset angegeben werden, da alle Platten im SVL der Pubres verankert werden müssen.

Die Funktion unterstützt das Einrichten von Pubsets mit Platten, deren physikalische Blockgröße 4K beträgt.

Das Einrichten einer urladefähigen Platte mit physikalischer Blockgröße 4K wird abgelehnt.

Erweitern eines SF-Pubset

In ein bereits eingerichtetes SF-Pubset können weitere Platten aufgenommen werden. Ist der Pubset importiert, erfolgt die Erweiterung dynamisch, d.h. der Pubset muss nicht vorher exportiert werden.

Folgende Anweisungen stehen für die Funktion „Erweitern eines SF-Pubset“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(...,ACTION=*EXTEND)
CREATE-VOLUME
CREATE-PAGING-FILE
COPY                      (nur erforderlich bei Erweiterung zum Home-Pubset)
CREATE-IPL-VOLUME                          - " -
MODIFY-IPL-VOLUME                          - " -
CREATE-SNAP-FILE                           - " -
```

Für jede Platte, um die der Pubset erweitert werden soll, muss eine CREATE-VOLUME-Anweisung angegeben werden (siehe Abschnitt „SM-Pubsets“). Die angegebene Plattennummer darf im Pubset noch nicht vorhanden sein.

Die beim Erweitern eines Pubset dazukommenden Platten müssen dieselben Plattenformate (K/NK, NK2/NK4) haben wie die bereits vorhandenen. Die Allocation-Unit muss innerhalb eines Pubset ebenfalls homogen sein. Deshalb werden die Platten von SIR automatisch initialisiert.

Einrichten eines Home-Pubset

Soll ein Betriebssystem BS2000 installiert und sein Home-Pubset eingerichtet werden, so müssen alle Dateien, die für das Betriebssystem benötigt werden (BS2000 V21.0 und die dazu gehörenden systemnahen Software-

Produkte), auf den Pubset gebracht werden. Mindestens eine Platte muss urladefähig gemacht werden (siehe nächsten Abschnitt). Für die Systemeinleitung muss eine ausreichend große Paging-Datei eingerichtet werden (>= 200 MB).

Das Einrichten einer Snapshot-Datei empfiehlt sich, wenn der SNAP-Dump bereits nach der Systemeinleitung zur Verfügung stehen soll. Siehe den Parameter SNAP-ACTIVE-SWITCH im Startup-Parameterservice, Handbuch „Systembetreuung“ [5].

Folgende Anweisungen stehen für das Einrichten eines Home-Pubset zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(...,ACTION=*INSTALL)
CREATE-VOLUME
CREATE-CATALOG
COPY
CREATE-IPL-VOLUME
CREATE-PAGING-FILE
CREATE-SNAP-FILE
```

Es gelten die Bedingungen wie im Abschnitt „SM-Pubsets“.

Einrichten einer Urladeplatte

Mit SIR können gemeinschaftliche oder private Platten urladefähig, also zu IPL-Platten gemacht werden.

Es können gemeinschaftliche Platten eines bereits bestehenden SF-Pubset als Urladeplatten der neuen BS2000-Version eingerichtet werden.
Beim Neuerstellen eines SF-Pubset können mehrere Urladeplatten für verschiedene BS2000-Versionen eingerichtet werden, wobei für Versionen ungleich der aktuellen Version von BS2000 die entsprechende SIR-Version eingesetzt werden muss. Von einer Urladeplatte kann immer nur eine Betriebssystemversion geladen werden. Auch mehrere Urladeplatten der gleichen Version können eingerichtet werden.

Platten eines Pubsets mit dem Attribut `LARGE-FILES-ALLOWED=*YES` können ebenfalls urladefähig gemacht werden. Platten eines Pubset mit Standard-SYS-ID „250“ können **nicht** urladefähig gemacht werden.

Eine Platte ist urladefähig, wenn der Operator von ihr den Betriebssystemlauf (STARTUP) einleiten kann. Dazu muss die Platte folgende Eigenschaften besitzen:

- Auf ihr befinden sich
  die IPL-Phase (SYSPRG.IPL.DSKxxx),

  sowie die Sicherstellungsdateien

  BOOTSAVE (SYSPRG.BOOT.DSKxxx.SAVE),
  SLEDSAVE (SYSPRG.SLED.DSKxxx.SAVE) und
  STARTUP-Konfigurationsdaten (SYSDAT.IPL-CONF.DSKxxx)

  und, falls benötigt, die

  IPL-Rep-Datei (SYSREP.IPL.DSKxxx) und
  SLED-Rep-Datei (SYSREP.SLED.DSKxxx)

  Wenn die Rep-Dateien für IPL und SLED nicht vorhanden sind, wird bei der Verankerung eine entsprechende Meldung ausgegeben und die Bearbeitung fortgesetzt.

  Bei Privatplatten wird <.DSKxxx> durch <.vsn> ersetzt.
- Der Standard-Datenträger-Kennsatz SVL enthält Verweise auf diese Dateien.

Im Rahmen der Funktion „Einrichten einer Urladeplatte“ werden die BOOTSAVE-, die SLEDSAVE- und die IPL-CONF-Datei eingerichtet. Die Dateien werden im Boot- bzw. SVL-Block der Platte verankert.

Diese Dateien müssen vor Aufruf von SIR nicht auf der Platte liegen, die urladefähig gemacht werden soll. SIR sorgt durch internes Kopieren dafür, dass diese Bedingung bei Verankerung der Urladedateien im SVL erfüllt ist: SIR kopiert die Ausgangsdateien unter einem plattenspezifischen Namen auf die Zielplatte. Die Ausgangsdateien werden weder verankert noch modifiziert. Es sind dies: SYSPRG/SKMPRG.IPL.<ver>, SYSREP.IPL.<ver> und SYSREP.SLED.<ver>

Die verankerten Dateien werden bei einer logischen Sicherung nicht berücksichtigt, wohl aber ihre Ausgangsdateien.

Folgende Anweisungen stehen für die Funktion „Einrichten einer Urladeplatte“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(...,ACTION=*IPL)
CREATE-IPL-VOLUME
MODIFY-IPL-VOLUME
       (kann zusätzlich oder ersatzweise gegeben werden)
```

Beim Ablauf werden alle CREATE-IPL-VOLUME-Anweisungen vor den MODIFY-IPL-VOLUME-Anweisungen ausgeführt.

Sind lediglich Privat-Platten urladefähig zu machen, so darf keine Anweisung DECLARE-PUBSET oder BEGIN-VOLUME-SET-DECLARATION gegeben werden.

Befinden sich die IPL-Dateien nicht im Pubset, so sind sie vorher dorthin zu übertragen. Dazu ist zu den oben stehenden Anweisungen die Anweisung COPY hinzuzufügen.

Bei Privatplatte müssen die Urladedateien im Home-Pubset in einem zur Privatplatte kompatiblen Format sein (d.h. in der Regel NK-Format).

Beim Einrichten der Urladeplatte ist zu beachten, dass ein Verändern der Ausgangsdateien sich nicht auf die verankerten Dateien auswirkt. Sollen die verankerten Dateien verändert werden, ist SIR erneut aufzurufen.

Alle anderen Dateien, die für die Systemeinleitung von BS2000 benötigt werden (STARTUP-Dateien, siehe Handbuch „Systembetreuung“ [5]), werden über den Dateikatalog gesucht. Sie können auf beliebigen Platten des Home-Pubset liegen, müssen also nicht unbedingt auf der Urladeplatte stehen.

Modifizieren einer Urladeplatte

Mit dieser Funktion wird eine urladefähige Platte modifiziert. Dabei dürfen die angegebenen Ausgangsdateien (IPL, IPL-Rep, SLED-Rep), anders als beim Einrichten der Urladeplatte, auch vom Standard abweichende Namen haben. Die Dateien werden dann unter einem plattenspezifischen Namen auf die Urladeplatte kopiert und neu im Boot- bzw. SVL-Block verankert. Beim Verankern von IPL werden die BOOTSAVE-, die SLEDSAVE- und die IPL-CONF-Datei neu eingerichtet. Die Verankerungen der restlichen Dateien, die für die Urladefähigkeit notwendig sind, bleiben erhalten.

Werden alle notwendigen Dateien angegeben, entspricht diese Funktion der Funktion „Einrichten einer Urladeplatte “. In diesem Falle muss die betreffende Platte nicht urladefähig gewesen sein.

Folgende Anweisungen stehen für die Funktion „Modifizieren einer Urladeplatte“ zur Verfügung:

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(...,ACTION=*IPL)
```

`MODIFY-IPL-VOLUME`

Für das „Modifizieren einer Urladeplatte“ ist das Gleiche zu beachten wie im Abschnitt „Einrichten einer Urladeplatte“ beschrieben.

13.2 Pubset-Zustände

Die Funktionen des Dienstprogramms SIR erzeugen und verändern Pubsets und damit auch deren Zustand. Dabei sind gewisse Randbedingungen zu beachten, um Hindernisse im Ablauf zu vermeiden. In diesem Abschnitt wird in zusammenfassender Weise die Wirkung der SIR-Aktionen auf die beteiligten Pubsets beschrieben.

Über den Operanden ACTION der Anweisung DECLARE-PUBSET wird der Bearbeitungsmodus eines Pubset festgelegt.

ACTION=*INSTALL (Pubsets)

- Die gewünschte Pubset-Id darf bereits im MRS-Katalog des Home-Pubset eingetragen sein. In diesem Fall findet eine Überprüfung des Gerätetyps statt. Bei Inkonsistenz wird das Einrichten abgebrochen. Fehlt ein MRS-Katalogeintrag, wird dieser von SIR angelegt.
- Wenn der Pubset in der vorhandenen BS2000-Umgebung eingerichtet werden soll, darf kein Pubset mit gleicher Pubset-Id zum Ablaufzeitpunkt importiert sein.
- Werden während des Einrichtens eines SM-Pubset neue Volume-Sets eingerichtet oder bestehende Volume-Sets hinzugenommen, so darf das Volume-Set nicht gerade benutzt werden oder im MRS-Katalog eingetragen sein.
- SIR führt beim Einrichten eines SM-Pubset immer und bei SF-Pubsets dann /IMPORT-PUBSET durch, wenn eine der folgenden Anweisungen angegeben wurde:

```
CREATE-CATALOG DISK-NUMBER ungleich 00
CREATE-PAGING-FILE
CREATE-SNAP-FILE
COPY
CREATE-IPL-VOLUME
MODIFY-IPL-VOLUME
```

ACTION=*EXTEND (Pubsets)

- Der Pubset muss im MRS-Katalog des Home-Pubset eingetragen sein.
- Wenn der Pubset bereits importiert ist, dann sollte er exklusiv importiert sein. Das Erweitern von Shared-Pubsets ist zwar möglich, wird nur dann sicher funktionieren, wenn das lokale System der einzige Sharer ist.
- Wenn der Pubset noch nicht importiert ist, dann muss es möglich sein, ihn mit /IMPORT-PUBSET ACTUAL-JOIN=*STD zu importieren. Dazu muss vorher bereits ein Importieren mit ACTUAL-JOIN=*FIRST erfolgt sein (siehe den entsprechenden Hinweis bei ACTION=*INSTALL).
- Werden während der Erweiterung eines SM-Pubset Volume-Sets eingerichtet oder hinzugenommen, so gelten für diese die gleichen Voraussetzungen wie für das Einrichten eines SM-Pubset.

ACTION=*IPL (nur SF-Pubset)

- Der Pubset muss im MRS-Katalog des Home-Pubset eingetragen sein.
- Der Pubset muss entweder vollständig importiert oder exportiert sein. Zwischenzustände wie sie im Verlauf des /IMPORT-PUBSET oder /EXPORT-PUBSET auftreten können, sind nicht erlaubt und die SIR-Funktion wird abgebrochen.

Für die Belegung des Pubset gegenüber anderen Systemen oder Tasks gilt:

- Die Platten sind bis Abschluss der Verarbeitung mit CREATE-VOLUME exklusiv belegt.
- Die Platten werden bei der Durchführung nachfolgender Anweisungen stets mehrbenutzbar belegt. In diesen Phasen ist es prinzipiell möglich, dass andere Systeme oder Tasks ebenfalls mit dem Pubset mehrbenutzbar arbeiten. Fremdsystembelegungen können zu Inkonsistenzen führen. Diese Fremdbelegungen können durch SIR nicht ausgeschlossen werden. Dazu stehen keine geeigneten System-übergreifenden Funktionen zur Verfügung. Die Systemverwaltung sollte durch organisatorische Maßnahmen dafür sorgen, dass während eines SIR-Laufs nicht von anderen Nutzern auf den Pubset zugegriffen wird.

> ¡ Das nachträgliche Einrichten von Paging-Dateien wird außerhalb von SIR durch Kommandos unterstützt.

„Große“ Platten und Dateien

Ein Pubset besitzt zwei Attribute, die anzeigen, ob für diesen Pubset Platten und Dateien größer als 32 GByte (kurz „große Platten / Dateien“) zugelassen sind.

Große Dateien (ohne PAM-Key) können auf NK- und K-Platten eingerichtet werden (große Dateien mit PAM-Key sind nicht vorgesehen).

Die Attribute werden von SIR beim Einrichten eines Pubset mit den Operanden `LARGE-DISKS-ALLOWED` und `LARGE-FILES-ALLOWED` in der Anweisung `//DECLARE-PUBSET` gesetzt. Große Platten / Dateien können bei Bedarf nachträglich mit dem BS2000-Kommado `/SET-PUBSET-ATTRIBUTES` erlaubt werden.

Wird beim Einrichten oder Erweitern eines Pubset eine große Platte angegeben, obwohl das Attribut dies nicht erlaubt, wird die Platte mit einer entsprechenden Meldung abgewiesen. Soll der Pubset dennoch um diese Platte erweitert werden, so muss bei `ACTION=*INSTALL` der Operand `LARGE-DISKS-ALLOWED=*YES` angegeben werden und bei `ACTION=*EXTEND` zuvor das Attribut mit dem BS2000-Kommando `/SET-PUBSET-ATTRIBUTES` geändert werden.

> ¡ Ein Home-Pubset darf große Platten und großen Dateien enthalten.
> Die Einstellung, dass für ein Pubset große Platten / Dateien zugelassen sind, kann nicht wieder rückgängig gemacht werden.
>
> Eine Platte / Datei mit weniger als 32 GByte („kleine“ Platte) kann, unabhängig vom Attribut „große Platten / Dateien“, in jeden Pubset aufgenommen werden.

13.3 Übersicht der SIR-Funktionen und -Anweisungen

SF-Pubset

Anweisungen / Funktion	Einrichten Pubset	Erweitern Pubset	Einrichten IPL-Platte	Modifizieren IPL-Platte
DECLARE-PUBSET TYPE=*S-F	X	X	X [2)]	X [2)]
CREATE-VOLUME	X	X		
CREATE-CATALOG	(X)			
CREATE-PAGING-FILE	(X)	(X)	(X) [2)]	(X) [2)]
CREATE-SNAP-FILE	(X)	(X)		
COPY	(X)	(X)	(X)	(X)
CREATE-IPL-VOLUME	(X)	(X)	X [1)]	
MODIFY-IPL-VOLUME	(X)	(X)	(X)	X
DELETE-IPL-FACILITY	(X)	(X)	(X)	(X)
INITIALIZE-PRIVATE-VOLUME	(X)	(X)	(X)	(X)
INITIALIZE-PUBLIC-VOLUME	(X)	(X)	(X)	(X)

Zeichenerklärung:

- X Die Anweisung muss angegeben werden.
- (X) Die Anweisung kann angegeben werden.
- 1) Die Anweisung kann durch eine MODIFY-IPL-VOLUME-Anweisung ersetzt werden.
- 2) Die Anweisung muss entfallen, wenn nur Privatplatten urladefähig gemacht werden sollen.

SM-Pubset

Anweisungen außerhalb des Anweisungsblocks BEGIN-VOLUME-SET-DECLARATION und END-VOLUME-SET-DECLARATION:

Anweisungen / Funktion	Einrichten Pubset	Erweitern Pubset
DECLARE-PUBSET TYPE=*S-M	X	X
BEGIN-VOLUME-SET-DECLARATION ACTION=*INSTALL	X	(X)
BEGIN-VOLUME-SET-DECLARATION ACTION=*ADD	(X)	(X)
BEGIN-VOLUME-SET-DECLARATION ACTION=*EXTEND		(X)
END-VOLUME-SET-DECLARATION	X	X
DELETE-IPL-FACILITY	(X)	(X)
INITIALIZE-PRIVATE-VOLUME	(X)	(X)
INITIALIZE-PUBLIC-VOLUME	(X)	(X)
COPY	(X)	(X)

Anweisungen innerhalb des Anweisungsblocks BEGIN-VOLUME-SET-DECLARATION und END-VOLUME-SET-DECLARATION:

Anweisungen / Funktion	Einrichten Volume-Set	Erweitern Volume-Set	Einfügen Volume-Set
CREATE-VOLUME	X	X	
CREATE-CATALOG	(X)		
CREATE-PAGING-FILE	(X)	(X)	
DELETE-IPL-FACILITY	(X)	(X)	(X)
INITIALIZE-PRIVATE-VOLUME	(X)	(X)	(X)
INITIALIZE-PUBLIC-VOLUME	(X)	(X)	(X)

Zeichenerklärung:

- X Die Anweisung muss angegeben werden.
- (X) Die Anweisung kann angegeben werden.

Eine detaillierte Beschreibung der Anweisungen ist im Abschnitt „Anweisungen“ zu finden.

Aufbereitung der Angaben zu den Pubsets/Volume-Sets

Mit der Editoranweisung START-EXECUTION wird die Eingabefolge der Anweisungen zum Einrichten bzw. Erweitern von Pubsets/Volume-Sets beendet und die Abarbeitung eingeleitet. Nach einer logischen Prüfung (siehe "Anweisungs-Editor") werden die Angaben zu den Pubsets in aufbereiteter Form ausgegeben:

Beispiel einer SF-Pubset-Übersicht

```
*****************************************************************
* PUBSET-TYPE: SINGLE-FEATURE                                   *
* LARGE-DISKS-ALLOWED: NO                                       *
* LARGE-FILES-ALLOWED: NO                                       *
*                                                               *
*     P U B S E T - D I S K  T A B L E        (ACTION:  *INSTALL) *
*                                                               *
*   PVS-ID  SYS-ID FORMAT  PHYSICAL-   ALLOCATION-  AVAILABILITY  *
*                          BLOCK-SIZE  UNIT                     *
*                                                               *
*     DEMO     65      K        2K           6         NORMAL     *
*                                                               *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
* DEMO.0 D3435 I                     OLD.00 1000         48    Y  *
* DEMO.1 D3435 I                     OLD.01 1000   150          Y  *
* DEMO.2 D3435 I                     OLD.02 1000   150             *
*                                                               *
*****************************************************************
*****************************************************************
*           C R E A T E - I P L - V O L U M E  T A B L E        *
*                                                               *
* PUBLIC DISKS                                                  *
*                                                               *
* VSN:DEMO.0    DEVICE:D3435                                    *
*                                  HSI:390                      *
*                                                               *
* VSN:DEMO.1    DEVICE:D3435                                    *
*                                  HSI:X86                      *
*                                                               *
*****************************************************************
```

Beispiel einer Übersicht bei der Bearbeitung eines SM-Pubset

```
*****************************************************************
* PUBSET-TYPE:        SYSTEM-MANAGED                            *
* PUBSET:             DEMO                                      *
* CONTROL-VOLUME-SET: CTL                                       *
* LARGE-DISKS-ALLOWED: NO                                       *
* LARGE-FILES-ALLOWED: NO                                       *
*                                                               *
* VOLUME- ACTION  FORMAT  PHYSICAL-  ALLOCATION- AVAILABILITY   *
* SET                     BLOCK-SIZE UNIT                       *
*                                                               *
*  CTL    INSTALL   NK        2K          8          DRV        *
*                                                               *
* LOGICAL VOLUME-SET ATTRIBUTES:                                *
*                                                               *
* PERFORMANCE WRITE-            USAGE       AVAILABILITY        *
*             CONSISTENCY                                       *
*                                                               *
*    STD       BY-CLOSE          STD           STD              *
*                                                               *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
* CTL.00 D3435 I                    OLD.00  2500                *
* CTL.01 D3435 I                    OLD.01  2500                *
* CTL.01 D3435 I                    OLD.01  2500                *
*                                                               *
* VOLUME- ACTION  FORMAT  PHYSICAL-  ALLOCATION- AVAILABILITY   *
* SET                     BLOCK-SIZE UNIT                       *
*                                                               *
*  VOL1   INSTALL   K         2K          6         NORMAL       *
*                                                               *
* LOGICAL VOLUME-SET ATTRIBUTES:                                *
*                                                               *
* PERFORMANCE WRITE-            USAGE       AVAILABILITY        *
*             CONSISTENCY                                       *
*                                                               *
*    STD       BY-CLOSE          STD           STD              *
*                                                               *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
*                                                               *
* VOL1.0 D3435 I                    OLD.02  2500                *
* VOL1.1 D3435 I                    OLD1.1                      *
*                                                               *
*****************************************************************
```

Beispiel einer Übersicht bei der Bearbeitung freier Volume-Sets

```
*****************************************************************
*                                                               *
* VOLUME- ACTION  FORMAT  PHYSICAL-  ALLOCATION- AVAILABILITY   *
* SET                     BLOCK-SIZE UNIT                       *
*                                                               *
*  VOL1   INSTALL   NK        2K         64         NORMAL      *
*                                                               *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
*                                                               *
* VOL1.0 D3435 I                    OLD.00  2500                *
* VOL1.1 D3435 I                    OLD.01                      *
*                                                               *
*****************************************************************
```

In beiden Übersichten wird der Abschnitt zur Beschreibung eines Volume-Set für jedes zu bearbeitende Volume-Set wiederholt.

13.4 Installation und Betrieb von SIR

Versionsabhängigkeiten und Systemeinbettung

Für die Einrichtung von Pubsets sind grundsätzlich zwei Varianten von Versionsabhängigkeiten zu beachten: Zielversionsabhängigkeit und Ablaufversionsabhängigkeit.

Mit SIR V21.0C kann aus den **Ablaufversionen BS2000 V21.0A/B/C** ein Home-Pubset für BS2000 V21.0A/B/C eingerichtet und urladefähig gemacht werden. Hierzu kann beim Aufruf von SIR die gewünschte Zielversion über den Parameter **TARGET-VERSION** angegeben werden.

Dabei ist zu beachten, dass nur Gerätetypen unterstützt werden, die sowohl auf der Ablaufversion als auch in der höchsten möglichen Zielversion zum BS2000-Gerätespektrum gehören.

Zudem muss stets gelten: Ablaufversion <= Zielversion.

Dies bedeutet, dass Rückumstiege nicht möglich sind, z.B. das Einrichten eines Pubset mit versionsspezifischen Merkmalen von BS2000 V21.0A mit einem SIR V21.0C, der auf BS2000 V21.0C abläuft.
Für einen eventuellen Rückumstieg wird empfohlen, einen geeigneten Home-Pubset oder eine geeignete Urladeplatte zu behalten.

Versionsumstieg aus BS2000 V21.0A/B

In den Altversionen BS2000 V21.0A und BS2000 V21.0B ist SIR V21.0C nur über BS2GA.MIGRATE V21.0 installierbar und aktivierbar. Der Versionswechsel selbst ist ausführlich im Handbuch "Systeminstallation" [7] beschrieben.

Privilegierung von SIR

Bei der Nutzung von SIR-Funktionen, die strukturelle Veränderung von Platteninhalten verursachen, muss der Benutzer entsprechend privilegiert sein (Privilegien TSOS und STD-PROCESSING). Ferner sind die auszuführenden Operationen auf den Platten nur unter Verwendung privilegierter Systemfunktionen möglich. Aus diesem Grund läuft die SIR-Funktion nach Abwicklung der Parametereingabe als DSSM-Subsystem hinter einem SVC ab. Die gleichzeitige Verwendung der gleichen Pubset-Id durch parallel laufende Tasks wird abgelehnt.

Aufruf von SIR

Das Dienstprogramm SIR kann im Dialog- oder Batch-Betrieb aufgerufen werden mit dem Kommando `/START-SIR` oder `/SIR`.

START-SIR	Alias: **SIR**
VERSION = *STD / <product-version> **,TARGET-VERSION**= ***STD** | <product-version_mandatory-manual-release_without-correction-state> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Bei Angabe von ***STD** für **TARGET-VERSION** wird die höchste mögliche Zielversion **BS2000 V21.0C** gewählt.

Die Anweisungen werden aus der Systemdatei SYSDTA gelesen. Fehlerhafte Eingaben können im Dialog sofort korrigiert werden, wenn SYSDTA nicht umgewiesen wurde. Andernfalls wird bei der ersten ungültigen Anweisung abgebrochen.
Meldungen werden nach SYSOUT ausgegeben.

Beispiel für den Ablauf von SIR

Ein Pubset wird um eine Platte erweitert, die urladefähig gemacht wird.

```
/start-sir
%  SIRLOAD PROGRAM SIR, VERSION <version> OF <date> LOADED FROM FILE
          :CAM1:$TSOS.SYSLNK.SIR.<version>
%  SIR0000 PROGRAM SIR (VERSION <version>) READY FOR INPUT
declare-pubset pubset-type=*single-feature(pubset=9701,action=*extend)
create-volume disk-number=2(overwrite-disk=c0002r,unit=A901)
create-ipl-volume disk-number=2
list-statements
*********************************************************************
*                                                                   *
* OPTION AND TASK ORDERS GENERATED :                                *
*                                                                   *
*    STATEMENTS GENERATED :                                         *
*      NUMBER  STATEMENT                                            *
*          1   //DECLARE-PUBSET PUBSET-TYPE=*SINGLE-FEATURE(PUBS- *
*              ET=9701,ACTION=*EXTEND,SYS-ID=*CAT-ID)               *
*          2   //CREATE-VOLUME DISK-NUMBER=2(OVERWRITE-DISK=C000- *
*              2R,UNIT=A901(SUBUNIT=*NO)),DEVICE-TYPE=*BY-UNIT,F- *
*              ORMAT=*NO                                            *
*          3   //CREATE-IPL-VOLUME DISK-NUMBER=2,HSI-TYPE=*BY-OWN *
*              -HSI                                                 *
*                                                                   *
*********************************************************************
%  SIR7054 ENTER LIST SUBFUNCTION OR 'END'. REPLY ('+'=SCROLL FORWARD;
          '-'=SCROLL BACK; END=TERMINATE LST FUNCTION)
end
create-paging-file disk-number=2,file-size=150
start-execution
*********************************************************************
* PUBSET-TYPE: SINGLE-FEATURE                                       *
* LARGE-DISKS-ALLOWED: NO                                           *
* LARGE-FILES-ALLOWED: NO                                           *
*                                                                   *
*     P U B S E T - D I S K  T A B L E        (ACTION:  *EXTEND)    *
*                                                                   *
*   PVS-ID  SYS-ID FORMAT  PHYSICAL-   ALLOCATION- AVAILABILITY     *
*                          BLOCK-SIZE  UNIT                         *
*                                                                   *
*     9701     65     K         2K           6        NORMAL       *
*                                                                   *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
* 9701.2 D3435      I    A901        C0002R        150         Y   *
*                                                                   *
*********************************************************************
*********************************************************************
*           C R E A T E - I P L - V O L U M E  T A B L E            *
*                                                                   *
```

```
* PUBLIC DISKS                                                        *
*                                                                    *
* VSN:9701.2    DEVICE:D3435                                         *
*                                  HSI:390                           *
*                                                                    *
**********************************************************************
%  SIR0040 DO YOU WANT TO CORRECT INPUT? REPLY (Y=YES; N=NO)?
n
%  SIR0300 'CREATE-VOLUME' FUNCTION STARTED
%  NVL0000 VOLIN VERSION <version> READY
%  NVL0031 INITIALIZATION STARTED FOR VOLUME '9701.2' ON UNIT 'A901' IN
           FORMAT 'K(A-U=6)'
%  NVL0017 INITIALIZATION OF VOLUME '9701.2' ON UNIT 'A901' COMPLETED.
           VOLUME FORMAT: 'K(A-U=6)'
%  SIR0301 'CREATE-VOLUME' FUNCTION TERMINATED NORMALLY
%  SIR0169 IMCAT TASK STARTED
%  SIR0400 CREATION OF PAGING FILES STARTED
%  SIR0401 CREATION OF PAGING FILES TERMINATED NORMALLY
%  SIR0523 //CREATE-IPL-VOLUME: FILE CHECK STARTED
%  SIR0524 //CREATE-IPL-VOLUME: CHECK OF FILES TERMINATED
%  SIR0500 //CREATE-IPL-VOLUME: PROCESSING STARTED
%  SIR0501 //CREATE-IPL-VOLUME: PROCESSING TERMINATED NORMALLY
**********************************************************************
* PUBSET-TYPE: SINGLE-FEATURE                                        *
* LARGE-DISKS-ALLOWED: NO                                            *
* LARGE-FILES-ALLOWED: NO                                            *
*                                                                    *
*     P U B S E T - D I S K  T A B L E        (ACTION:  *EXTEND)     *
*                                                                    *
*   PVS-ID  SYS-ID FORMAT  PHYSICAL-   ALLOCATION- AVAILABILITY      *
*                          BLOCK-SIZE  UNIT                          *
*                                                                    *
*     9701     65     K         2K           6        NORMAL        *
*                                                                    *
*   VSN  DEVICE-  VOLIN- MAIN- SUB-  CHECK- CAT-  PAG-  SNAP- IPL-*
*        TYPE     ACTION UNIT  UNIT  VSN    SIZE  SIZE  SIZE  VOL *
* 9701.2 D3435      I    A901        C0002R        150         Y  *
*                                                                    *
**********************************************************************
**********************************************************************
*           C R E A T E - I P L - V O L U M E  T A B L E             *
*                                                                    *
* PUBLIC DISKS                                                       *
*                                                                    *
* VSN:9701.2    DEVICE:D3435                                         *
*                                  HSI:390                           *
*                                                                    *
**********************************************************************
end
%  SIR1010 'SIR' TERMINATED
```

13.5 Übertragen von Dateien mit SIR

Das Dienstprogramm kann unter beliebiger Benutzerkennung mit dem Privileg STD-PROCESSING eingesetzt werden. Aufgerufen wird es im Dialog- oder im Batch-Betrieb mit dem Kommando `/START-SIR` oder `/SIR` .

Meldungen werden nach SYSOUT ausgegeben. Anweisungen werden von SYSDTA gelesen. Fehlerhafte Eingaben können im Dialog und, falls SYSDTA=(PRIMARY) gilt, sofort korrigiert werden. Im Batch-Betrieb wird bei der ersten ungültigen Anweisung abgebrochen.

Anweisungen an den „nicht privilegierten SIR“ - Übersicht

Anweisung	Kurzbeschreibung
COPY	Übertragen von Dateien
END	Beenden einer Eingabefolge

13.6 Anweisungen

- Übersicht über die Anweisungen von SIR
- Eingaberegeln
- Anweisungs-Editor
- Beschreibung der Anweisungen
  - BEGIN-VOLUME-SET-DECLARATION - Einrichten eines Volume-Set
  - COPY - Dateien übertragen
  - CREATE-CATALOG - Dateikatalog festlegen
  - CREATE-IPL-VOLUME - Platte urladefähig machen
  - CREATE-PAGING-FILE - Paging-Datei einrichten
  - CREATE-SNAP-FILE - Snapshot-Datei einrichten
  - CREATE-VOLUME - Platten initialisieren
  - DECLARE-PUBSET - Pubset und Verarbeitungsart definieren
  - DELETE-IPL-FACILITY - Urladefähigkeit und Urladedateien einer Platte löschen
  - END - SIR beenden
  - END-VOLUME-SET-DECLARATION - Anweisungsfolge für Volume-Set beenden
  - INITIALIZE-PRIVATE-VOLUME - Privatplatten initialisieren
  - INITIALIZE-PUBLIC-VOLUME - Public-Platten initialisieren
  - MODIFY-IPL-VOLUME - Modifizieren einer Urladeplatte

13.6.1 Übersicht über die Anweisungen von SIR

Übersicht der Anweisungen an den „privilegierten SIR“

Anweisung	Kurzbeschreibung
DECLARE-PUBSET	Definieren globaler Parameter für das zu bearbeitende Pubset
BEGIN-VOLUME-SET-DECLARATION	Definieren globaler Parameter für ein zu bearbeitendes Volume-Set (leitet den Anweisungsblock ein)
END-VOLUME-SET-DECLARATION	Beenden des Volume-Set-Anweisungsblocks
CREATE-VOLUME	Platten initialisieren
CREATE-CATALOG	Einrichten eines Dateikatalogs
CREATE-PAGING-FILE	Einrichten von Paging-Dateien
CREATE-SNAP-FILE	Einrichten der Snapshot-Datei zur Aufnahme von Snapshot-Dumps
COPY	Übertragen von Dateien
CREATE-IPL-VOLUME	Einrichten einer Urladeplatte (nur SF-Pubset)
MODIFY-IPL-VOLUME	Modifizieren einer Urladeplatte (nur SF-Pubset)
DELETE-IPL-FACILITY	Entfernen Urladefähigkeit und Urladedateien einer Platte (nur SF-Pubset)
INITIALIZE-PRIVATE-VOLUME	Initialisieren Privatplatte
INITIALIZE-PUBLIC-VOLUME	Initialisieren einer Platte, ohne sie zu einem Pubset hinzuzufügen
START-EXECUTION	Ausführen der Anweisungsfolge (siehe "Anweisungs-Editor")
END	Beenden des SIR

13.6.2 Eingaberegeln

Weil in einem SIR-Lauf mehr als ein Volume-Set generiert werden kann, müssen die Anweisungen eindeutig einem Volume-Set zuzuordnen sein. Dazu dient der Anweisungsblock zum Einrichten eines Volume-Set, der mit den Anweisungen BEGIN-VOLUME-SET-DECLARATION eingeleitet und mit END-VOLUME-SET-DECLARATION beendet wird. Innerhalb eines Gültigkeitsbereichs können die Anweisungen in beliebiger Reihenfolge angeordnet werden. Werden mehrere Anweisungen der gleichen Art eingegeben, die sich widersprechen, so ist die letzte Anweisung gültig.

Während der Eingabe einer Anweisungsfolge werden folgende Anweisungen sofort verarbeitet:

DELETE-IPL-FACILITY
INITIALIZE-PRIVATE-VOLUME
INITIALIZE-PUBLIC-VOLUME
START-EXECUTION
END

Wird eine dieser Anweisungen innerhalb einer Anweisungsfolge eingegeben, so bleibt die Anweisungsfolge von der sofortigen Ausführung der Funktion unbeeinflusst. Die Eingabe der Anweisungsfolge kann danach fortgesetzt werden.
Ausnahmen sind die beiden zuletzt aufgeführten Anweisungen. START-EXECUTION startet die Abarbeitung der eingegebenen Anweisungsfolge. Nach Ausführung der Anweisung wird die aktuelle Anweisungsfolge gelöscht.
Wird die Anweisung END eingelesen, beendet sich SIR.

Korrektur der Anweisungen

Ein Korrekturdialog für falsch eingegebene Anweisungen ist nicht möglich. Die entsprechende Anweisung kann jedoch gelöscht und/oder durch eine Anweisung gleicher Art ersetzt werden, vorausgesetzt, man befindet sich noch im selben Gültigkeitsbereich.

Beispiel

```
X  DECLARE-PUBSET  PUBSET-TYPE=*S-F(PUBSET=A)
X  CREATE-VOLUME   02,D3435,FORMAT=YES
   CREATE-VOLUME   03,D3435,FORMAT=YES
X  CREATE-VOLUME   03,D3435,FORMAT=NO
```

oder

```
o  BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=A
o  CREATE-VOLUME 02,D3435
o  END-VOLUME-SET-DECLARATION
X  BEGIN-VOLUME-SET-DECLARATION VOLUME-SET=A
X  CREATE-VOLUME 00,D3435
X  END-VOLUME-SET-DECLARATION
X  START-EXECUTION
```

Gültig sind die angekreuzten Anweisungen. Die mit o gekennzeichneten Anweisungen müssen explizit gelöscht werden.

Falsche Anweisungen können explizit mit dem Editor-Kommando DELETE-STATEMENTS aus der Anweisungsfolge gelöscht werden. Über das Kommando LIST-STATEMENTS erhält man die Nummer der Anweisung in der Anweisungsfolge. Anschließend kann mit dem Kommando INSERT-STATEMENTS eine neue an Stelle der falschen Anweisung eingefügt werden.

Erst mit der Editoranweisung START-EXECUTION wird eine Eingabefolge abgeschlossen. Danach wird die logische Überprüfung aller Anweisungen gestartet.

Die Anweisungen DELETE-IPL-FACILITY und INITIALIZE-PRIVATE/PUBLIC-VOLUME können an jeder Stelle des Eingabevorgangs vor END eingegeben werden. Sie werden dann sofort ausgeführt.

Die Initialisierungen können nicht parallel bearbeitet werden. Die gesamte Bearbeitung des Pubset/Volume-Set wird seriell in einer Task durchgeführt. Zur Vermeidung einer seriellen Abarbeitung können die betreffenden Platten vor der Pubset/Volume-Set-Generierung in parallelen Tasks mit der Anweisung INITIALIZE-PUBLIC-VOLUME mit einem bestimmten Bitmuster initialisiert werden.
Bei der nachfolgenden Pubset/Volume-Set-Generierung ist für diese Platten die Angabe FORMAT=*NO (Standard) in der CREATE-VOLUME-Anweisung hinreichend.

Ist durch den Ablauf von SIR die Bearbeitung eines Pubset oder Volume-Set mit bestimmter Id in einer Benutzertask aktiv, so ist diese Id für andere Tasks gesperrt. Aufrufe von SIR mit gleicher Pubset-Id in parallel ablaufenden Benutzertasks werden mit einer entsprechenden Meldung abgebrochen.

> Maximal 32.767 SIR-Anweisungen können angegeben werden.
> Maximale Anzahl an Platten pro SF-Pubset oder Volume-Set:
>
> - 255 bei zwei- oder dreistelliger Cat-Id und einer Allocation-Unit-Größe ungleich 6 KB
> - 36 bei vierstelliger Cat-Id und Allocation Unit > 6KB
> - 32 in allen anderen Pubsets
>
> Maximale Anzahl an Volume-Sets in einem SM-Pubset: 255

Für die Verwendung von Stellungsparametern, Abkürzungen und Fortsetzungszeilen gelten die allgemeinen SDF-Regeln.

13.6.3 Anweisungs-Editor

Nach dem Start des SIR wird der Anweisungs-Editor aufgerufen. Er ermöglicht dem Benutzer, beim Ablauf interaktiv Anweisungen einzugeben oder Änderungen an vorhandenen Anweisungen vorzunehmen.

Der Anweisungs-Editor kennt dabei folgende Editor-Kommandos:

DELETE-STATEMENTS STATEMENT-NUMBER=*ALL / 1..32767

Eine Anweisung oder die gesamte Anweisungsfolge wird gelöscht.

INSERT-STATEMENTS STATEMENT-NUMBER=*APPEND / 1..32767

Die folgenden Anweisungen sollen an der angegebenen Anweisungsnummer eingefügt werden.
Mit *APPEND werden die folgenden Anweisungen an das Ende der Anweisungsfolge angefügt.
Ist die angegebene Anweisungsnummer größer als die der letzten Anweisung, wird ebenfalls an das Ende der Anweisungsfolge angefügt.

RENUMBER-STATEMENTS

Die Anweisungen werden in aufsteigender Reihenfolge neu nummeriert.

LIST-STATEMENTS

Alle Anweisungen der aktuellen Anweisungsfolge werden aufgelistet.

Der Anweisungs-Editor bleibt während des SIR-Laufs aktiv. Vom Editor aus leitet der Benutzer die Abarbeitung der Anweisungsfolge mit START-EXECUTION ein.

Nach START-EXECUTION wird die Anweisungsfolge einer logischen Prüfung unterzogen (d.h. die gegebene Anweisungsfolge wird auf Vollständigkeit und Konsistenz überprüft). Anschließend wird die Eingabe des Benutzers in aufbereiteter Form ausgegeben (siehe Ausgabebeispiel unter „Aufbereitung der Angaben zu den Pubsets /Volume-Sets“ (Übersicht der SIR-Funktionen und -Anweisungen)).

Gibt die Prüfung Anlass zu Beanstandungen, so wird dem Benutzer Gelegenheit gegeben, die Anweisungsfolge zu korrigieren, indem wieder in den Anweisungs-Editor verzweigt wird. Andernfalls werden die Funktionen gestartet.

Ist die Anweisungsfolge abgearbeitet, kehrt SIR in den Anweisungs-Editor zurück. Anschließend kann der Benutzer eine weitere Anweisungsfolge oder die Anweisung END zum Verlassen von SIR angeben.

13.6.4 Beschreibung der Anweisungen

- BEGIN-VOLUME-SET-DECLARATION - Einrichten eines Volume-Set
- COPY - Dateien übertragen
- CREATE-CATALOG - Dateikatalog festlegen
- CREATE-IPL-VOLUME - Platte urladefähig machen
- CREATE-PAGING-FILE - Paging-Datei einrichten
- CREATE-SNAP-FILE - Snapshot-Datei einrichten
- CREATE-VOLUME - Platten initialisieren
- DECLARE-PUBSET - Pubset und Verarbeitungsart definieren
- DELETE-IPL-FACILITY - Urladefähigkeit und Urladedateien einer Platte löschen
- END - SIR beenden
- END-VOLUME-SET-DECLARATION - Anweisungsfolge für Volume-Set beenden
- INITIALIZE-PRIVATE-VOLUME - Privatplatten initialisieren
- INITIALIZE-PUBLIC-VOLUME - Public-Platten initialisieren
- MODIFY-IPL-VOLUME - Modifizieren einer Urladeplatte

13.6.4.1 BEGIN-VOLUME-SET-DECLARATION - Einrichten eines Volume-Set

Mit dieser Anweisung wird die Deklaration des einzurichtenden Volume-Set eingeleitet. Zusammen mit der Anweisung END-VOLUME-SET-DECLARATION, die die Deklaration beendet, umschließt sie einen Anweisungsblock, der für jedes einzurichtende Volume-Set wiederholt werden muss (siehe dazu auch die Funktion „Einrichten eines Volume-Set“ (Volume-Sets)).

Hinweise

- Beim Festlegen der Struktur des Volume-Set ist zu berücksichtigen, dass diese Einstellungen später nicht ohne ein Neueinrichten des Volume-Set geändert werden können. Hierzu zählen alle Angaben, die über den Operanden FORMAT angegeben werden.
- Alle logischen Eigenschaften des Volume-Set, die über den Operanden VOLUME-SET-ATTR angegeben werden, können während des laufenden Betriebes mit dem BS2000-Kommando /MODIFY-PUBSET-DEFINITION-FILE geändert werden.

BEGIN-**VOL**UME-**SET-DECL**ARATION

VOLUME-**SET**=volume-set-id

,**ACT**ION=**<u>INST</u>**ALL(...) / ***ADD**(...) / ***EXT**END

<u>INST</u>ALL(...)

| **FORM**AT=**<u>K</u>** / ***NK**(...)

| ***NK**(...)

| | **PHYS**ICAL-**BLOCK-SIZE**=**<u>2K</u>**(...) / ***4K**(...)

| | **<u>2K</u>**(...)

| | | **ALLOC**ATION-**UNIT**=**<u>6</u>** / **8** / **64**

| | ***4K**(...)

| | | **ALLOC**ATION-**UNIT**=**<u>8</u>** / **64**

| ,**AVAIL**ABILITY=**<u>NORM</u>**AL / ***DRV**

| ,**VOL**UME-**SET-ATTR**=***PAR**AMETERS(...)

| ***PAR**AMETERS(...)

| | **USAGE**=**<u>STD</u>** / ***WORK** / ***HSMS-CONTR**OLLED

| | ,**AVAIL**ABILITY=**<u>STD</u>** / ***HIGH**

| | ,**PERF**ORMANCE-**ATTR**=**<u>STD</u>** / ***PAR**AMETERS(...)

| | ***PAR**AMETERS(...)

| | | **PERF**ORMANCE=list-poss(3): **<u>STD</u>** / ***HIGH** / ***VERY-HIGH**

| | | ,**WRITE-CONS**ISTENCY=**<u>BY-CLOSE</u>** / ***IMMED**IATE

***ADD**(...)

| **VOL**UME-**SET-ATTR**=***PAR**AMETERS(...)

| ***PAR**AMETERS(...)

| | **USAGE**=**<u>STD</u>** / ***WORK** / ***HSMS-CONTR**OLLED

| | ,**AVAIL**ABILITY=**<u>STD</u>** / ***HIGH**

| | ,**PERF**ORMANCE-**ATTR**=**<u>STD</u>** / ***PAR**AMETERS(...)

| | ***PAR**AMETERS(...)

| | | **PERF**ORMANCE=list-poss(3): **<u>STD</u>** / ***HIGH** / ***VERY-HIGH**

| | | ,**WRITE-CONS**ISTENCY=**<u>BY-CLOSE</u>** / ***IMMED**IATE

VOLUME-SET=volume-set-id
Identifier, der das Volume-Set kennzeichnet. Er ist syntaktischer Bestandteil in der VSN aller zum Volume-Set gehörenden Platten. Der Volume-Set-Identifier hat zwei Formate:

volume-set-id1:

Der Identifier ist 1 Zeichen lang (alphanumerisch) und steht als 4. Zeichen in der VSN jeder zum Volume-Set gehörenden Platte.
Aufbau der VSN:
PUB**p**xx mit

p = volume-set-id1

xx = Nummer der Platte im Volume-Set
Wertebereich:: 00 - 31

volume-set-id2:

Der Identifier ist 2 bis 4 Zeichen lang (alphanumerisch) und steht am Beginn der VSN. Aufbau der VSN: **pp**.xyz oder **ppp**.yz oder **pppp**.z mit

p = Zeichen aus volume-set-id2

x,y,z = Nummer der Platte im Volume-Set

Wertebereich bei Allocation-Unit > 6
x,y,z: 0 - 9, A - Z.

Wertebereich bei Allocation-Unit = 6

x,y: Die Stellen für x und y sind mit 0 belegt;

z: 0 - 9, A - V.

Da maximal 32 Platten im Volume-Set möglich sind, sind für z die Werte W bis Z nicht zulässig.

volume-set-id2 darf nicht mit der Zeichenfolge PUB beginnen.

ACTION
Gibt den Bearbeitungsmodus des Volume-Set an.

ACTION=*INSTALL(...)
Das mit der Volume-Set-Id angegebene Volume-Set soll eingerichtet werden.

FORMAT
Legt das Plattenformat und damit die Betriebsart des Volume-Set bezüglich der PAM-Key-Verwendung fest.

FORMAT=*K
Der PAM-Key wird bereitgestellt. Das Volume-Set besteht aus K-Platten. K-Dateien mit BLKCTRL=PAMKEY sind erlaubt.

FORMAT=*NK(...)
Die Platten des Volume-Set sind ohne PAM-Key formatiert (NK-Platten). Nur NK-Dateien mit BLKCTRL=DATA / NO sind möglich.

PHYSICAL-BLOCK-SIZE
Legt die physikalische Blockgröße auf den NK-Platten des Volume-Set fest.

PHYSICAL-BLOCK-SIZE=*2K(...)
NK2-Platte (Standardwert)
Die minimale Transferunit zwischen Platte und Speicher beträgt 2 Kbyte. D.h., durch einen Schreib/Lese-Auftrag werden immer 2 Kbyte (oder ein Vielfaches) bewegt.

ALLOCATION-UNIT=6 / 8 / 64
Legt die Größe der Einheit fest, mit der das DVS Platz für Dateien auf den Platten des Volume-Set belegt. 6 KByte ist der Standardwert für NK2-Platten.

PHYSICAL-BLOCK-SIZE=*4K(...)
NK4-Platte. Die minimale Transferunit beträgt 4 Kbyte.

ALLOCATION-UNIT=8 / 64
Legt die Größe der Einheit fest, mit der das DVS Platz für Dateien auf den Platten des Volume-Set belegt. 8 Kbyte ist Standardwert für NK4-Platten.

AVAILABILITY
Legt die Verfügbarkeit des Volume-Set fest.

AVAILABILITY=*NORMAL
Es werden keine Spiegelplatten eingerichtet (übliche Verfügbarkeit).

AVAILABILITY=*DRV
Ein Volume-Set mit DUAL-RECORDING wird eingerichtet.

VOLUME-SET-ATTR=*PARAMETERS(...)
Die logischen Eigenschaften des Volume-Set werden über Parameter festgelegt.

USAGE
Bestimmt die Nutzungsart des Volume-Set als Dateiablageort.

USAGE=*STD
Das Volume-Set soll als Ablageort für Standarddateien genutzt werden.

USAGE=*WORK
Das Volume-Set soll als Ablageort für Arbeitsdateien genutzt werden.

USAGE=*HSMS-CONTROLLED
Das Volume-Set dient dem Subsystem HSMS zur Realisierung der S1-Speicherebene und des Backup-Volume-Set eines SM-Pubset.

AVAILABILITY
Bestimmt den Grad der Verfügbarkeit von Dateien.

AVAILABILITY=*STD
Das Volume-Set bietet keine erhöhte Ausfallsicherheit.

AVAILABILITY=*HIGH
Das Volume-Set bietet eine hohe Ausfallsicherheit.

PERFORMANCE-ATTR
Legt die Performance-Eigenschaften für das Volume-Set fest.

PERFORMANCE-ATTR=*STD
Das Volume-Set hat die Standard-Performance-Eigenschaften.

PERFORMANCE-ATTR=*PARAMETERS(...)
Die Performance-Eigenschaften des Volume-Set werden über Parameter festgelegt.

PERFORMANCE=list-poss(3):
Die Angabe mehrerer Performance-Eigenschaften (maximal 3) ist möglich.

PERFORMANCE=*STD
Das Volume-Set bietet keine erhöhte I/O-Performance.

PERFORMANCE=*HIGH
Das Volume-Set bietet erhöhte I/O-Performance.

PERFORMANCE=*VERY-HIGH
Das Volume-Set bietet sehr hohe I/O-Performance.

WRITE-CONSISTENCY
Legt den Zeitpunkt der Konsistenz für schreibende Zugriffe auf eine Platte des Volume-Set fest.

WRITE-CONSISTENCY=*BY-CLOSE
Die Konsistenz der Schreibzugriffe wird erst beim Schließen der Datei hergestellt.

WRITE-CONSISTENCY=*IMMEDIATE
Die Konsistenz der Schreibzugriffe wird unverzüglich hergestellt.

ACTION=*ADD(...)
Ein „freies“ Volume-Set, d.h. es ist leer und gehört noch keinem Pubset an, soll unverändert in ein SM-Pubset aufgenommen werden.

Die Operanden der Unterstrukturen zu ACTION=*ADD sind beim Operanden ACTION= *INSTALL beschrieben.

ACTION=*EXTEND
Das mit der Volume-Set-Id angegebene Volume-Set soll erweitert werden.

13.6.4.2 COPY - Dateien übertragen

Mit dieser Anweisung können Dateien zwischen Privatplatte und Pubset übertragen werden. Bei der Beschreibung der einzelnen Operanden wird darauf hingewiesen, wenn der Operand das TSOS-Privileg benötigt.

COPY

FILE=name(...)

name(...)

| **NEW**= ***SAME** / name / ***PREFIX**(...)

| ***PREFIX**(...)

| | **ADD**= ***NO** / pfx

| | ,**REM**OVE= ***NO** / pfx

,**FROM**= ***PUB**SET / ***DEV**ICE(...)

***DEV**ICE(...)

| **VOL**UME=vsn

| ,**DEV**ICE**-TYPE**=device-type / **STDDISK**

,**TO**= ***PUB**SET(...) / vsn(...)

***PUB**SET(...)

| **DISK-NUM**BER= ***ANY** / <integer x..y>

vsn(...)

| **DEV**ICE**-TYPE**=device-type / **STDDISK**

,**REPL**ACE**-OLD-F**ILES= ***DIALOG**UE / ***YES** / ***NO**

FILE
Gibt die Dateien an, die übertragen werden sollen.

FILE=name(...)
Gibt die Dateien, die übertragen werden sollen, durch einen teil- oder vollqualifizierten Namen an.
Falls keine Pubset-Id angegeben ist, gilt:
Die Pubset-Id wird aus der Anweisung DECLARE-PUBSET übernommen, wenn diese Anweisung gegeben wurde, oder aus dem Home-Pubset.

NEW
Gibt die neuen Namen an, die die übertragenen Dateien erhalten sollen.

NEW=*SAME
Die Dateien werden unter dem gleichen Namen katalogisiert, wie im Operanden FILE angegeben ist.
*SAME darf nicht zusammen mit FROM=*PUBSET angegeben werden.

Beispiel

Soll die Dateifolge A.B. aus dem aktuellen BS2000-Pubset in das Ziel-Pubset (Pubset-Id aus der DECLARE-PUBSET-Anweisung) kopiert werden, so ist folgende Anweisung zu verwenden:
`COPY FILE=:<quell-pubset>:A.B.(NEW=A.B.)`

NEW=name
Voll-/teilqualifizierter Name, unter dem die mit FILE=name angegebene Datei katalogisiert werden soll. Der mit FILE angegebene Name muss ebenfalls voll/teilqualifiziert angegeben werden.

NEW=*PREFIX(...)
Steuert das Vorketten bzw. Abstreifen der Präfixe bei der Erzeugung der Zieldateinamen.

ADD=*NO
Dem Zieldateinamen wird kein Präfix vorangestellt.

ADD=pfx
Dem Zieldateinamen wird der Präfix pfx vorangestellt:
Quelldatei: ABC, Präfix: V21.0 wird zu
Zieldatei: V21.0.ABC

REMOVE=*NO
Bei der Bildung des Zieldateinamens wird ein vorhandener Präfix nicht abgestreift.

REMOVE=pfx
Bei der Bildung des Zieldateinamens wird ein vorhandener Präfix entfernt.

Quelldatei: ZN.ABC, Präfix: LG wird zu

Zieldatei: ZN.ABC

Quelldatei: ZN.ABC, Präfix: ZN wird zu

Zieldatei: ABC

pfx darf maximal 10 Zeichen lang sein. Die Katalog- oder Benutzerkennung darf für pfx nicht verwendet werden.

Wenn beide Operanden, ADD und REMOVE, in einer Anweisung angegeben werden, so wird zunächst REMOVE und dann ADD verarbeitet:

```
PREFIX ADD=V21.0,REMOVE=ZN
```

Quelldatei: ZN.ABC wird zu

Zieldatei: V21.0.ABC

FROM
Gibt an, von wo aus die Dateien übertragen werden sollen.

FROM=*PUBSET
Die Dateien sollen aus den Pubsets, die gerade importiert sind, übertragen werden.

FROM=*DEVICE(...)

VOLUME
Die Dateien sollen vom privaten Datenträgern übertragen werden.

VOLUME=vsn
VSN des privaten Datenträgers

DEVICE-TYPE=device-type / STDDISK
Bei Angabe von VOLUME muss auch der Gerätetyp bzw. der Standard-Gerätetyp (STDDISK) mit angegeben werden.

TO
Gibt das Ziel an, wohin die Dateien übertragen werden sollen.

TO=*PUBSET(...)
Wenn DECLARE-PUBSET angegeben ist, wird auf diesen Pubset übertragen. Sonst auf den Standard-Pubset des Benutzers.

DISK-NUMBER
Dieser Operand darf nur mit TSOS-Privileg angegeben werden.

Er gibt die Platte des SF-Pubset an, auf die übertragen werden soll.
Bei Platten eines SM-Pubset wird der Operand ignoriert.

DISK-NUMBER=*ANY
Die Dateien sollen auf beliebige Platten innerhalb des SF-Pubset übertragen werden.

DISK-NUMBER=n
Die Dateien sollen auf die mit n angegebene Platte übertragen werden.
(zu „Nummer der Platte im Pubset“ siehe Anweisung CREATE-VOLUME).

TO=vsn(...)
VSN der Privatplatte, auf die übertragen werden soll.

DEVICE-TYPE=device-type
Gerätetyp der Privatplatte.

DEVICE-TYPE=STDDISK
Standard-Gerätetyp der Privatplatte.

REPLACE-OLD-FILES
Gibt an, wie verfahren werden soll, wenn eine Datei unter gleichem Namen bereits katalogisiert ist.

REPLACE-OLD-FILES=*DIALOGUE
Der Benutzer erhält im Dialog eine Meldung und kann entscheiden, ob die Datei überschrieben werden soll oder nicht. In Ablaufmodi, in denen kein Dialog mit dem Benutzer stattfinden kann (z.B. Batch-Verarbeitung), wirkt der Wert *DIALOGUE wie *NO.

REPLACE-OLD-FILES=*YES
Eine bereits vorhandene Datei wird überschrieben (bei gleichem FCBTYP an die gleiche Stelle).

REPLACE-OLD-FILES=*NO
Eine bereits vorhandene Datei wird nicht überschrieben.

Hinweise

1. Mit der COPY-Anweisung müssen alle für die Installation benötigten Dateien übertragen werden.
2. Jobvariablen und Dateigenerationsgruppen werden übergangen.
3. Die Dateiattribute werden von der „Quelldatei“ übernommen.
4. Bei Plattenfehlern wird die Bearbeitung der aktuell zu übertragenden Datei abgebrochen und ggf. mit der nächsten Datei fortgefahren.
5. Eine Konvertierung des Dateiformats ist nicht möglich.
6. Soll beim Übertragen eine Datei auf eine andere Kennung eingespielt werden (nur unter TSOS möglich), so ist diese mit dem FILE-Operanden NEW=name anzugeben.

13.6.4.3 CREATE-CATALOG - Dateikatalog festlegen

Mit dieser Anweisung wird der Anteil des Dateikatalogs für ein Pubset oder ein Volume-Set festgelegt, der auf der angegebenen Platte bzw. dem angegebenen Bereich eingerichtet werden soll.
Die Größe des TSOSCAT des Pubset oder Volume-Set ergibt sich aus der Summe der Bereiche, die für die einzelnen Platten festgelegt wurden.

```
CREATE-CATALOG

  DISK-NUMBER=list-poss(32): n / *RANGE(...)

        *RANGE(...)

              |       FROM=n

              |       ,TO=m

  ,FILE-SIZE=2000 / <integer 60..64016>
```

DISK-NUMBER
Gibt die Platten an, auf denen ein Katalogabschnitt eingerichtet werden soll. Abhängig vom Format des Pubset- oder Volume-Set-Identifier muss die Nummer der Platte innerhalb des Pubset oder Volume-Set angegeben werden.

DISK-NUMBER=n
Ein-, zwei- oder dreistellige Nummer der Platte innerhalb des Pubset oder Volume-Set (siehe "CREATE-VOLUME - Platten initialisieren"). Es ist eine Liste mit maximal 32 Elementen zugelassen.

DISK-NUMBER=*RANGE
Mehrere Platten innerhalb eines Bereichs sollen bearbeitet werden. Der Bereich wird durch FROM und TO begrenzt.

FROM=n
Beginn des Bereichs.

TO=m
Ende des Bereichs.

Auf jeder Platte dieses Bereichs wird ein Katalogabschnitt in der Größe eingerichtet, wie sie durch den Operanden FILE-SIZE angegeben wird.
Es muss gelten: n < m.

FILE-SIZE=<integer 60..64016>
Gibt die Größe des Katalogabschnitts pro Platte in PAM-Seiten an.
Die Größe wird intern auf eine volle UNIT-Anzahl aufgerundet.

FILE-SIZE=2000
Standardwert für die Größe des Katalogabschnitts ist 2000 PAM-Seiten.

Hinweise

1. Gibt die Systembetreuung keine CREATE-CATALOG-Anweisung für die Systemplatte des Pubset/Volume-Set (Pubres/Volres) an, wird dort standardmäßg ein Katalogabschnitt von 2000 PAM-Seiten eingerichtet.
2. Ist bei DECLARE-PUBSET ACTION=*INSTALL die Summe der FILE-SIZE-Operandenwerte pro Platte größer als die maximal zulässige Kataloggröße, werden nacheinander so viele Katalog-Extents eingerichtet, bis die maximale Größe erreicht ist. Beim letzten Katalog-Extent wird ggf. die Größe reduziert.
3. Es wird nur der TSOSCAT-Typ EXTRA-LARGE unterstützt.
4. SIR bricht mit einer Fehlermeldung ab, wenn eine Kataloggröße angegeben wird, die größer als die Plattenkapazität ist.

13.6.4.4 CREATE-IPL-VOLUME - Platte urladefähig machen

Mit dieser Anweisung wird eine Platte urladefähig gemacht.
Die Urladefähigkeit einer Platte ist im Abschnitt „SF-Pubsets“ ausführlich beschrieben.

CREATE-**IPL-VOL**UME
DISK-NUMBER=n / ***PRIV**ATE(...) ***PRIV**ATE(...) | **DISK**=vsn | ,**DEV**ICE-**TYPE**= **STDDISK** / device-type ,**HSI-TYPE**= ***BY-OWN-HSI** / ***390** / ***X86** [1)]

[1)] Der Operandenwert *7500 kann aus Kompatibilitätsgründen noch angegeben werden.

DISK-NUMBER
Gibt die Platte an, die urladefähig gemacht werden soll. Abhängig vom Format der Pubset-Id (siehe Anweisung DECLARE-PUBSET) muss die Nummer der Platte innerhalb des Pubset angegeben werden.

DISK-NUMBER=n
Ein-, zwei- oder dreistellige Nummer der Platte innerhalb des Pubset (siehe "CREATE-VOLUME - Platten initialisieren").

DISK-NUMBER=*PRIVATE(...)
Gibt eine Privatplatte an, die urladefähig gemacht werden soll.

DISK=vsn
VSN der Privatplatte

DEVICE-TYPE=STDDISK
Standard-Gerätetyp der Privatplatte.

DEVICE-TYPE=device-type
Gerätetyp der Privatplatte.

HSI-TYPE
Legt das HSI fest, für das die Urladefähigkeit hergestellt werden soll.

HSI-TYPE=*BY-OWN-HSI
Die Urladefähigkeit wird für das HSI hergestellt, auf dem SIR läuft.

HSI-TYPE=*390
Die Urladefähigkeit wird für das /390-HSI hergestellt (/390-Server).

HSI-TYPE=*X86
Die Urladefähigkeit wird für das X86-HSI hergestellt (x86-Server).

Hinweise

1. Die Urladedateien müssen im Pubset katalogisiert sein, wenn DISK-NUMBER=n angegeben wird. Urladedateien sind die IPL-Phase (SYSPRG.IPL.<ver> bei HSI=*390 und SKMPRG.IPL.<ver> bei HSI=*X86), sowie die Rep-Dateien für IPL und SLED (SYSREP.IPL.<ver> und SYSREP.SLED.<ver>), wenn diese vorhanden sind.
2. Die Urladefähigkeit erfordert, wie gerade erwähnt, das Vorhandensein der Daten der Urladedateien auf der Platte. CREATE-IPL-VOLUME führt diesen Zustand durch Kopieren der Dateien auf eine Dateikopie herbei.
3. Bei DISK-NUMBER=*PRIVATE müssen die Urladedateien bereits im Home-Pubset sein, und zwar in einem mit der Privatplatte verträglichen Format . Eine Anweisung DECLARE-PUBSET darf nicht gegeben werden, wenn in diesem SIR-Lauf nur eine oder mehrere Privatplatten urladefähig gemacht werden sollen.
   Wenn die Rep-Dateien für IPL und SLED nicht vorhanden sind, wird eine entsprechende Meldung ausgegeben und die Bearbeitung fortgesetzt.
4. Eine Platte kann immer nur für ein HSI urladefähig sein. D.h. wird mit CREATE-IPL-VOLUME die Urladefähigkeit für HSI-TYPE=*X86 festgelegt, so geht die ggf. vorher vorhandene Urladefähigkeit für ein /390-HSI verloren.
5. Beim Herstellen der Urladefähigkeit einer Platte werden zur Aufnahme von Speicherdaten, die beim Laden von BOOT bzw. SLED überschrieben werden, sowie von Startup-Konfigurationsdaten folgende Sicherungsdateien eingerichtet:
   - BOOT-datei.<x>.SAVE
   - SLED-datei.<x>.SAVE
   - SYSDAT.IPL-CONF.<x> (24 2K-Seiten)

   Dabei bedeuten BOOT-datei: Dateiname der BOOT-Phase
   SLED-datei: Dateiname der SLED-Phase
   <x>: DSKxxx oder vsn

   Die Dateien werden mit BLOCKSIZE=(STD,2) angelegt.

   Sind bereits Dateien mit diesen Namen vorhanden, wird wie folgt verfahren:

   - Die Datei ist in der richtigen Größe auf der richtigen Platte.
     Es ist keine weitere Aktion notwendig.
   - Die Datei ist in der falschen Größe auf der richtigen Platte.
     Die Größe der Datei wird angepasst.
   - Die Datei ist auf einer anderen Platte innerhalb des Pubset. Die Datei wird auf der entsprechenden Platte neu angelegt und auf der anderen Platte gelöscht.

Beispiel

Standard-IPL	SYSPRG.IPL.<ver>
Standard-IPL-Rep	SYSREP.IPL.<ver>
Standard-SLED-Rep	SYSREP.SLED.<ver>

`CREATE-IPL-VOLUME DISK-NUMBER=*PRIVATE(PRIV00),HSI-TYPE=*390` erzeugt folgende Dateien auf der Urladeplatte:

IPL	SYSPRG.IPL.PRIV00
BOOTSAVE	SYSPRG.BOOT.PRIV00.SAVE
SLEDSAVE	SYSPRG.SLED.PRIV00.SAVE
IPL-CONF	SYSDAT.IPL-CONF.PRIV00
IPL-REP	SYSREP.IPL.PRIV00
SLED-REP	SYSREP.SLED.PRIV00

6. Zur Sicherstellung von Systemkorrekturen während der Systemeinleitung wird von SIR die Datei SYS.NSI.SAVEREP angelegt, wenn die Urladefähigkeit mindestens einer Platte eines Pubsets hergestellt wird.
7. Alle verankerten Dateien erhalten die Dateiattribute BACKUP=E, DESTROY=*YES, ACCESS=*READ und MIGRATION=*FORBIDDEN.
8. Dateischutz mit BASIC-ACL für Urladedateien wird von SIR nicht unterstützt. Hat der Benutzer eine dieser Dateien mit BASIC-ACL geschützt, muss er den Schutz aufheben, damit SIR die Urladedateien modifizieren kann.
9. Die Dateinamen für die Ablage der Urladedateien werden implizit gebildet. Es ist darauf zu achten, dass sich die Urladedateien unter dem Standardnamen im Pubset befinden. Andernfalls wird eine Meldung ausgegeben und die Anweisung ignoriert.
10. Der Operand HSI-TYPE muss nur dann angegeben werden, wenn die Urladefähigkeit der Platte nicht für das HSI hergestellt werden soll, auf dem SIR aktuell abläuft.
11. Die Anweisung wird abgewiesen für Platten von Pubsets, die nicht als Home-Pubset eingesetzt werden können:
    - Pubsets mit `SYS-ID=*STD`
    - Pubsets mit `PHYSICAL-BLOCK-SIZE=*4K`
    - SM-Pubsets

13.6.4.5 CREATE-PAGING-FILE - Paging-Datei einrichten

Mit dieser Anweisung werden die Paging-Dateien eingerichtet.

CREATE-**PAG**ING- **FILE**
DISK-NUMBER=list-poss(32): n / ***RANGE**(...) ***RANGE**(...) | **FROM**=n | ,**TO**=m ,**F**ILE-**SIZE**=<integer 1..32767>

DISK-NUMBER
Gibt an, auf welchen Platten des Pubset eine Paging-Datei eingerichtet werden soll. Abhängig vom Format der Pubset-Id (siehe Anweisung DECLARE-PUBSET) muss die Nummer der Platte innerhalb des Pubset angegeben werden.

DISK-NUMBER=n
Ein-, zwei- oder dreistellige Nummer der Platte innerhalb des Pubset (siehe "CREATE-VOLUME - Platten initialisieren"). Es ist eine Liste zugelassen mit maximal 32 Elementen.

DISK-NUMBER=*RANGE(...)
Mehrere Platten innerhalb eines Bereichs sollen bearbeitet werden. Der Bereich wird durch die Operanden FROM und TO begrenzt.

FROM=n
Beginn des Bereichs des Pubset.

TO=m
Ende des Bereichs des Pubset.

Auf jeder Platte wird eine Paging-Datei in der Größe eingerichtet, die durch den Operanden FILE-SIZE angegeben wird.
Es muss gelten: n < m.

FILE-SIZE=<integer 1..32767>
Gibt die Größe der Paging-Datei pro Platte in Megabytes an.
Ist die Plattenkapazität geringer als im Operanden FILE-SIZE angegeben, so wird eine Paging-Datei in der Größe des maximal zulässigen Plattenwertes eingerichtet.

Hinweise

1. Das System benötigt zur Ablaufzeit eine Paging-Datei, die sich jedoch nicht auf dem Home-Pubset befinden muss. Daher können Pubsets ohne Paging-Datei eingerichtet werden.
2. Wurde beim Einrichten eines Pubset (ACTION=*INSTALL) keine Anweisung CREATE-PAGING-FILE gegeben, so wird eine Meldung ausgegeben.

3. Paging-Dateien erhalten die gleichen Dateiattribute, die mit dem Kommando /CREATE-PAGING-FILE vergeben werden.
4. Weitere Informationen zu Paging-Dateien sind im „Performance Handbuch“ [9] beschrieben.

13.6.4.6 CREATE-SNAP-FILE - Snapshot-Datei einrichten

Mit dieser Anweisung wird eine Snapshot-Datei eingerichtet. Pro Pubset ist nur eine Snapshot-Datei zugelassen. Sie wird unter dem Dateinamen $TSOS.SNAPFILE katalogisiert.

CREATE-**SNAP-** **FILE**
DISK-NUMBER=n ,**F**ILE-**SIZE**=<integer 16..1024>

DISK-NUMBER
Gibt die Platte des Pubset an, auf der die Snapshot-Datei eingerichtet werden soll. Abhängig vom Format der Pubset-Id (siehe Anweisung DECLARE-PUBSET) muss die Nummer der Platte innerhalb des Pubset angegeben werden.

DISK-NUMBER=n
Ein-, zwei- oder dreistellige Nummer der Platte innerhalb des Pubset (siehe "CREATE-VOLUME - Platten initialisieren").

FILE-SIZE=<integer 16..1024>
Gibt die Mindestgröße der Snapshot-Datei in Megabyte an.
Der Wert wird auf die Größe der Allocation-Unit aufgerundet.
Die Datei muss in ihrer vollen Größe als ein Extent eingerichtet werden können.
Ist die Plattenkapazität geringer als im Operanden FILE-SIZE angegeben, wird die Snapshot-Datei in der Größe des maximalen Plattenwertes eingerichtet. Informationen zur Größe der Snapshot-Datei finden Sie im „Diagnosehandbuch“ [2].

Hinweise

1. Wenn auf der selben Platte eine Paging-Datei eingerichtet wird, wird die Snapshot-Datei vom Ende der Platte her und die Paging-Datei um die Mitte zentriert eingerichtet. Sollten sich die Bereiche auf der Platte überschneiden, wird die Snapshot-Datei nicht eingerichtet.
2. Das Einrichten einer Snapshot-Datei empfiehlt sich, wenn der SNAP-Dump bereits nach der Systemeinleitung zur Verfügung stehen soll. Siehe den Parameter SNAP-ACTIVE-SWITCH im Startup-Parameterservice, Handbuch „Systembetreuung“ [5]. Beim Einrichten eines Pubset mit `DECLARE-PUBSET PUBSET-TYPE=*S-F(ACTION= *INSTALL)` wird eine Warnung ausgegeben, wenn keine CREATE-SNAP-FILE-Anweisung angegeben wird.
3. Eine Snapshot-Datei kann auch nachträglich bei einer Aktivierung des SNAP-Dump mit dem Kommando /ACTIVATE-SNAPSHOT eingerichtet bzw. geändert werden.

13.6.4.7 CREATE-VOLUME - Platten initialisieren

Mit dieser Anweisung werden Platten oder Bereiche eines Pubset oder Volume-Set initialisiert. Dies entspricht der VOLIN-Funktion.

```
CREATE-VOLUME

 DISK-NUMBER=list-poss(32): n(...) / *RANGE(...)

     n(...)

         |  OVERWRITE-DISK=*ANY / vsn

         |  ,UNIT=*NO / mn(...)

         |        mn(...)

         |              |  SUBUNIT=*NO / mn

     *RANGE(...)

         |  FROM=n

         |  ,TO=m

         |  ,OVERWRITE-DISK=*ANY / list-poss(32): vsn

         |  ,UNIT=*NO / list-poss(32): mn(...)

         |        mn(...)

         |              |  SUBUNIT=*NO / mn

 ,DEVICE-TYPE=device-type / *BY-UNIT [1)]

 ,FORMAT=*NO / *YES
```

[1] Die Angabe des Operanden DEVICE-TYPE wird nicht mehr empfohlen. Der DEVICE-TYPE soll implizit durch die UNIT bestimmt werden. Der Operand wird im geführten Dialog nicht mehr angezeigt.

DISK-NUMBER
Gibt an, welche Platte oder welcher Bereich eines Pubset oder Volume-Set initialisiert werden soll.

DISK-NUMBER=n(...)
Nummer der Platte innerhalb des Pubset oder Volume-Set. Der zulässige Bereich hängt ab von der Wahl des Formats der Pubset- oder Volume-Set-Id.

pubset-id1, volumeset-id1: (Der Identifier ist 1 Zeichen lang.)

> Die Nummer n der Platte im Pubset muss zweistellig sein. Sie geht als Zeichen 5 und 6 in die neue VSN der zu bearbeitenden Platte ein.
> Wert: 00 <= n <= 31
>
> Damit hat die VSN der Platte folgendes Aussehen: **PUBpnn** wobei **p** der einstellige Pubset- oder Volume-Set-Identifier ist.

pubset-id2, volumeset-id2: (Der Identifier ist 2 bis 4 Zeichen lang.)

Die Nummer der Platte im Pubset hängt ab von der Allocation-Unit:

ALLOCATION-UNIT = 6 :

Die Nummer der Platte ist einstellig. Wertebereich für z: 0 - 9, A - V

Damit kann die VSN der Platte folgendes Aussehen haben:

pp.00z oder **ppp.0z** oder **pppp.z**
wobei **p** der zwei- bis vierstellige Pubset- oder Volume-Set-Identifier ist.

ALLOCATION-UNIT > 6 :

Die Nummer der Platte ist maximal dreistellig. Wertebereich für x,y,z: 0 - 9, A - Z, wobei gilt:

1) 0 < 1 <...< 9 < A < B...< Z < 10

2) 10<11...<99<AA<BB...<ZZ<100

3) 100<101...<999<AAA<BBB...<ZZZ

Wenn weniger als drei Stellen angegeben werden, wird die Nummer der Platte linksbündig mit Nullen aufgefüllt.

Damit kann die VSN der Platte folgendes Aussehen haben:

1) **pppp.z** oder 2) **ppp.yz** oder 3) **pp.xyz**
wobei **p** der zwei- bis vierstellige Pubset- oder Volume-Set-Identifier ist.

Es ist eine Liste zugelassen. Sie darf maximal 32 Elemente umfassen. Der Operand OVERWRITE-DISK bezieht sich auf die jeweils zugehörige n-Angabe.

OVERWRITE-DISK
Gibt an, ob die VSN vor dem Initialisieren noch überprüft werden soll.

OVERWRITE-DISK=*ANY
Die Initialisierung findet ohne vorherige VSN-Prüfung durch SIR statt. In diesem Fall gibt VOLIN später die Frage NVL0024 aus. D.h. die VSN-Prüfung erfolgt manuell, im Dialog durch den Aufrufer, im Batch durch den Operator.

OVERWRITE-DISK=vsn
VSN der zu montierenden Platte. VOLIN initialisiert diese nur, wenn der Wert `vsn` mit der VSN der Platte übereinstimmt.

UNIT
Leitet eine Liste von Plattengeräten ein.

UNIT=*NO
Es sollen keine Platten-Mnemoniks vorgegeben werden.

UNIT=mn(...)
mn der ersten Platte (MAINUNIT-Platte) eines Plattenpaares oder, bei nicht-DRV-Pubsets, die einzige zu benutzende Platte.

SUBUNIT=*NO
Es wird keine zweite Platte angegeben.

SUBUNIT=mn
mn der zweiten Platte (SUBUNIT-Platte) eines Plattenpaares. Sie darf nur bei DRV angegeben werden.

DISK-NUMBER=*RANGE(...)
Mehrere Platten innerhalb eines Bereichs sollen bearbeitet werden. Der Bereich wird durch FROM und TO begrenzt.

FROM=n
Beginn des Bereichs.

TO=m
Ende des Bereichs.

> Für n und m gelten die gleichen Regeln, wie für n bei `DISK-NUMBER`. Es muss gelten: n < m.
>
> Unabhängig von der Notation haben n und m den Wertebereich 0..9, A-Z. Dies bedeutet z.B., dass die Angabe `FROM=1,TO=14` insgesamt 40 Platten (1-9, A-Z, 10-14) umfasst. Soll also in diesem Beispiel ein Pubset aus 14 Platten (1-14) erstellt werden, so sind zwei Anweisungen nötig:
>
> ```
> //CREATE-VOLUME DISK-NUMBER=*RANGE(FROM=1,TO=9,UNIT=(...))
> ```
>
> ```
> //CREATE-VOLUME DISK-NUMBER=*RANGE(FROM=10,TO=14,UNIT=(...))
> ```

OVERWRITE-DISK
Gibt an, ob die VSN vor dem Initialisieren noch überprüft werden soll.

OVERWRITE-DISK=*ANY
Die Initialisierung findet ohne vorherige VSN-Prüfung durch SIR statt. In diesem Fall gibt VOLIN später die Frage NVL0024 aus. D.h. die VSN-Prüfung erfolgt manuell, im Dialog durch den Aufrufer, im Batch durch den Operator.

OVERWRITE-DISK=list-poss(32): vsn
Eine Liste von VSNs ist nur zugelassen, wenn mehrere Platten innerhalb eines Bereichs mit dem Operanden *RANGE bearbeitet werden sollen. Die Anzahl der Listenelemente muss mit der Anzahl der Platten des Bereichs übereinstimmen. Andernfalls wird die Anweisung zurückgewiesen. Wird ein DRV-Pubset/Volume-Set gebildet, muss jede angegebene VSN zweimal im System vorhanden sein.

UNIT
Leitet eine Liste von Plattengeräten ein.

UNIT=*NO
Es sollen keine Platten-Mnemoniks vorgegeben werden.

UNIT=mn(...)
mn der ersten Platte eines Plattenpaares oder, bei nicht-DRV-Pubsets, die einzige zu benutzende Platte.

SUBUNIT=*NO
Es wird keine zweite Platte angegeben.

SUBUNIT=mn
Mnemonik der zweiten Platte eines Plattenpaares. Sie darf nur bei DRV angegeben werden.

FORMAT
Gibt die Art der Bearbeitung an.

FORMAT=*YES
Die Platte wird bei der Initialisierung mit einem bestimmten Bitmuster überschrieben d.h. alle Daten werden gelöscht.

FORMAT=*NO

Die Platte wird nicht mit einem Bitmuster überschrieben, sondern nur initialisiert. Ein Wechsel des Betriebsmodus bezüglich der PAM-Key-Verwendung muss konsistent für das gesamte Pubset/Volume-Set durchgeführt werden.

Hinweise

1. Bei der Initialisierung muss jede Platte des zu bearbeitenden Pubset in einer CREATE-VOLUME-Anweisung angegeben sein, da sie alle mit VSN und Gerätetyp im SVL der Systemplatte des Pubset (Pubres) verankert werden müssen.
2. Bei einer Pubset-Erweiterung (Operand ACTION=*EXTEND in der Anweisung DECLARE-PUBSET) werden dem Pubset die in diesen CREATE-VOLUME-Anweisungen angegebenen Platten hinzugefügt. Wird eine Platte angegeben, die bereits dem Pubset angehört, so wird diese Anweisung ignoriert.
3. Im Gegensatz zu den VOLIN-Anweisungen bzw. zu den Anweisungen INITIALIZE-PRIVATE/PUBLIC-VOLUME ist es hier möglich, mehrere Platten mit einer CREATE-VOLUME-Anweisung bearbeiten zu lassen. Voraussetzung dafür ist jedoch, dass alle in einer Anweisung aufgeführten Platten entweder den gleichen Gerätetyp aufweisen oder über UNIT spezifiziert werden und vom FORMAT-Parameter her gleich zu behandeln sind.
4. Bei der Angabe OVERWRITE-DISK=*ANY wird die montierte Platte nur überschrieben, wenn die Systemverwaltung in einer Meldung ihr Einverständnis gibt. In dieser Meldung ist die VSN der montierten Platte angegeben.
5. Beim Einrichten/Erweitern eines Pubset mit Spiegelplatten werden jeweils zwei Platten initialisiert. Wenn der Operand OVERWRITE-DISK=vsn angegeben wird, müssen in diesem Fall auch zwei Platten mit dieser VSN vorhanden sein.
6. Sind mehrere Platten mit einem bestimmten Bitmuster zu initialisieren, können diese wegen Zeitersparnis mit der Anweisung INITIALIZE-PUBLIC-VOLUME in parallelen SIR-Tasks initialisiert werden, da bei CREATE-VOLUME mit FORMAT=*YES die Initialisierungen mit einem bestimmten Bitmuster sequenziell abgearbeitet werden.

13.6.4.8 DECLARE-PUBSET - Pubset und Verarbeitungsart definieren

Mit dieser Anweisung wird das Pubset, das bearbeitet werden soll, und die Art der SIR-Bearbeitung definiert.

Mit der Definition wird festgelegt:

- der Pubset-Typ
  Legt fest, ob ein SF-Pubset (SINGLE-FEATURE-Pubset) oder ein SM-Pubset (SYSTEM-MANAGED-Pubset) eingerichtet werden soll.
- der Pubset-Identifier
  Bezeichnet das Pubset. Alle Pubsets eines Systems sind im MRS-Katalog verzeichnet. Bei SF-Pubsets ist der Pubset-Identifier syntaktischer Bestandteil der VSN aller zum Pubset gehörenden Platten.
- der System-Identifier
  Er weist das System gegenüber anderen Systemen aus, wenn mit dem SF-Pubset als Home-Pubset gearbeitet wird.
- der Betriebsmodus des SF-Pubset
  Er legt fest, ob PAM-Key auf den Platten des SF-Pubset verwendet wird.
- die Allocation-Unit
  Einheit, mit der das DVS Platzzuweisungen für Dateien auf den Platten des SF-Pubset vornimmt.
- die physikalische Blockgröße auf allen zum SF-Pubset gehörenden Platten.
- ob Platten und Dateien größer 32 GByte berücksichtigt werden sollen.
- der Identifier des Control-Volume-Set eines SM-Pubset, das Verwaltungsinformationen enthält.

DECLARE-**PUB**SET

PUBSET-**TYPE**=***SYS**TEM-**MANAG**ED(...) / ***SING**LE-**FEAT**URE(...)

***SYS**TEM-**MANAG**ED(...)

| **PUB**SET=pubset-id

| ,**ACT**ION= ***INST**ALL(...) / ***EXT**END(...)

| ***INST**ALL(...)

| | **CONTR**OL-**VOL**UME-**SET**=catid

| | ,**LARGE-DISKS-ALLOWED**= ***NO** / ***YES(...)**

| | ***YES(...)**

| | | **LARGE-FILES-ALLOWED**= ***NO** / ***YES**

| ***EXT**END(...)

| | **CONTR**OL-**VOL**UME-**SET**=catid

***SING**LE-**FEAT**URE(...)

| **PUB**SET=pubset-id

| ,**ACT**ION= ***INST**ALL(...) / ***EXT**END / ***IPL**

| ***INST**ALL(...)

| | **FORM**AT= ***K** / ***NK**(...)

| | ***NK**(...)

| | | **PHYS**ICAL-**BLOCK-SIZE**= ***2K** (...) / ***4K**(...)

| | | ***2K**(...)

| | | | **ALLOC**ATION-**UNIT**= **6** / **8** / **64**

| | | ***4K**(...)

| | | | **ALLOC**ATION-**UNIT**= **8** / **64**

| | ,**AVAIL**ABILITY= ***NORM**AL / ***DRV**

| | ,**LARGE-DISKS-ALLOWED**= ***NO** / ***YES(...)**

| | ***YES(...)**

| | | **LARGE-FILES-ALLOWED**= ***NO** / ***YES**

| ,**SYS-ID**= ***STD** / ***CAT-ID** / sys-id

PUBSET-TYPE
Legt den einzurichtenden Pubsettyp fest.

PUBSET-TYPE=*SYSTEM-MANAGED(...)
Es soll ein SM-Pubset eingerichtet werden.

PUBSET=pubset-id
Identifier, der das Pubset kennzeichnet (1 bis 4 alphanumerische Zeichen, darf nicht mit der Zeichenfolge PUB beginnen).

ACTION
Gibt den Bearbeitungsmodus des SIR an.

ACTION=*INSTALL(...)
Das mit der Pubset-Id angegebene Pubset soll eingerichtet werden.

CONTROL-VOLUME-SET=catid
Gibt die Katalogkennung des Volume-Set an, das als Control-Volume-Set des SM-Pubset angelegt werden soll. catid muss eine gültige Katalogkennung sein.

LARGE-DISKS-ALLOWED=*NO / *YES(...)
Gibt an, ob Platten mit einer Speicherkapazität von mehr als 32 GByte in ein Volume-Set des Pubsets aufgenommen werden dürfen (*YES(...)) oder nicht (*NO).

LARGE-FILES-ALLOWED=*NO / *YES
Gibt an, ob Dateien eines Pubsets mit großen Platten größer als 32 GByte werden dürfen (*YES) oder nicht (*NO).

ACTION=*EXTEND(...)
Das mit der Pubset-Id angegebene Pubset soll erweitert werden.

CONTROL-VOLUME-SET=catid
Gibt die Katalogkennung des Control-Volume-Sets an.
Das Control-Volume-Set kann nur eine gültige Katalogkennung als Namen haben.

PUBSET-TYPE=*SINGLE-FEATURE (...)
Es soll ein SF-Pubset eingerichtet werden.

PUBSET=pubset-id
Identifier, der das Pubset kennzeichnet.
Der Pubset-Identifier hat zwei Formate. Bei SF-Pubsets ist er fester Bestandteil im Aufbau der VSN aller zum Pubset gehörenden Platten.

pubset-id1:
Der Identifier ist 1 Zeichen lang (alphanumerisch).
Er steht als 4. Zeichen in der VSN jeder zum SF-Pubset gehörenden Platte. Aufbau der VSN:
PUB**p**xx mit

p = pubset-id1

xx = Nummer der Platte im Pubset. Wertebereich: 00 - 31

pubset-id2:
Der Identifier ist 2 bis 4 Zeichen lang (alphanumerisch).
Er steht am Beginn in der VSN jeder zum SF-Pubset gehörenden Platte. Aufbau der VSN:
pp.xyz oder **ppp**.yz oder **pppp**.z mit

p = Zeichen aus pubset-id2

x,y,z = Nummer der Platte im Pubset

Wertebereich bei Allocation-Unit > 6:

x,y,z: 0 - 9, A - Z.

Wertebereich bei Allocation-Unit = 6

x,y: Die Stellen für x und y sind mit 0 belegt;

z: 0 - 9, A - V.

Da maximal 32 Platten im Pubset möglich sind, sind für z die Werte W bis Z nicht zulässig.

pubset-id2 darf nicht mit der Zeichenfolge PUB beginnen.

ACTION
Gibt den Bearbeitungsmodus des SIR an.

ACTION=*INSTALL(...)
Das mit der Pubset-Id angegebene Pubset soll eingerichtet werden.

FORMAT
Legt das Plattenformat und damit die Betriebsart des Pubset bezüglich der PAM-Key-Verwendung fest.

FORMAT=*K
Der PAM-Key wird bereitgestellt. Das Pubset besteht aus K-Platten.
K-Dateien mit BLKCTRL=PAMKEY sind erlaubt.

FORMAT=*NK(...)
Die Platten des Pubset sind ohne PAM-Key formatiert (NK-Platten).
Nur NK-Dateien mit BLKCTRL=DATA / NO sind möglich.

PHYSICAL-BLOCK-SIZE
Legt die physikalische Blockgröße auf den NK-Platten des Pubset fest.

PHYSICAL-BLOCK-SIZE=*2K(...)
NK2-Platte (Standardwert)
Die minimale Transferunit zwischen Platte und Speicher beträgt 2 Kbyte. D.h., durch einen Schreib /Lese-Auftrag werden immer 2 Kbyte (oder ein Vielfaches) bewegt.

PHYSICAL-BLOCK-SIZE=*4K(...)
NK4-Platte. Die minimale Transferunit beträgt 4 Kbyte.

ALLOCATION-UNIT
Legt die Größe der Einheit fest, mit der das DVS Platz für Dateien auf den Platten des Pubset belegt.

ALLOCATION-UNIT=6
Die Einheit (Größe der kleinsten Datei) ist 6 Kbyte groß. Kann nur für
Pubsets mit NK2-Platten angegeben werden und ist dort Standardwert.

ALLOCATION-UNIT=8 / 64
Die Einheit (Größe der kleinsten Datei) soll 8 oder 64 Kbyte groß sein. Kann für Pubsets mit NK2- oder NK4-Platten angegeben werden. 8 Kbyte ist
Standardwert für NK4-Platten.

AVAILABILITY
Legt die Verfügbarkeit des Pubset fest.

AVAILABILITY=*NORMAL
Es werden keine Spiegelplatten eingerichtet (übliche Verfügbarkeit).

AVAILABILITY=*DRV
Ein Pubset mit DUAL-RECORDING wird eingerichtet.

LARGE-DISKS-ALLOWED=*NO / *YES(...)
Gibt an, ob Platten mit einer Speicherkapazität von mehr als 32 GByte in das Pubset aufgenommen werden dürfen (*YES(...)) oder nicht (*NO).

LARGE-FILES-ALLOWED=*NO / *YES
Gibt an, ob Dateien eines Pubsets mit großen Platten größer als 32 GByte werden dürfen (*YES) oder nicht (*NO).

ACTION=*EXTEND
Das mit der Pubset-Id angegebene Pubset soll erweitert werden.

ACTION=*IPL
IPL-Dateien sollen im SVL verankert werden.

SYS-ID
System-Identifier, der das Pubset bei Verwendung als Home-Pubset des Systems gegenüber anderen Systemen bei der Belegung von gemeinsam benutzbaren Geräten (MBK) kennzeichnet.

SYS-ID muss bei ACTION=*INSTALL angegeben werden. Bei ACTION=*EXTEND / *IPL wird eine Meldung ausgegeben und SYS-ID wird ignoriert.

SYS-ID=*STD
Der Standardwert ist 250.
Ein Pubset mit Standardwert kann nicht urladefähig gemacht werden.

SYS-ID=*CAT-ID
Wenn das Format pubset-id1 des Pubset-Identifiers verwendet wird, muss SYS-ID=*CAT-ID angegeben werden. Eine andere Vorgabe ist in diesem Fall nicht möglich.

SYS-ID=sys-id
numerischer Wert von 65 - 192

Hinweise

1. Wenn mit SIR nur eine Privatplatte urladefähig gemacht werden soll, darf keine Anweisung DECLARE-PUBSET angegeben werden.
2.

Beispiel

```
DECLARE-PUBSET PUBSET-TYPE=*S-F(030,*IPL)

DECLARE-PUBSET PUBSET-TYPE=*S-F(030,*INSTALL(*NK(*4K(64))),*STD)
```

13.6.4.9 DELETE-IPL-FACILITY - Urladefähigkeit und Urladedateien einer Platte löschen

Mit der Anweisung DELETE-IPL-FACILITY werden die Urladefähigkeit und Urladedateien einer Platte gelöscht. Eine Systemeinleitung von dieser Platte ist danach nicht mehr möglich.

```
DELETE-IPL-
FACILITY

 DISK=*PUBLIC(...) / *PRIVATE(...)

     *PUBLIC(...)

         |  DISK=vsn

     *PRIVATE(...)

         |  DISK=vsn

         |  ,DEVICE-TYPE= STDDISK / device-type
```

DISK
Legt die zu bearbeitende Platte fest.

DISK=*PUBLIC(...)
Die Urladefähigkeit einer Public-Platte soll gelöscht werden.

DISK=vsn
VSN der Public-Platte

DISK=*PRIVATE(...)
Die Urladefähigkeit einer Privatplatte soll gelöscht werden.

DISK=vsn
VSN der Privatplatte

DEVICE-TYPE=STDDISK
Standard-Gerätetyp der Privatplatte.

DEVICE-TYPE=device-type
Gerätetyp der Privatplatte.

Hinweise

1. Die Funktion kann auch auf Platten eines SM-Pubset angewandt werden. Zwar kann für Platten eines SM-Pubset keine Urladefähigkeit eingerichtet werden, jedoch können urladefähige Platten durch die Migration von einem SF-Pubset in ein SM-Pubset übernommen worden sein.
   Gehört die Platte zu einem Volume-Set, muss dieses Volume-Set Bestandteil eines SM-Pubset sein.
2. Der Pubset mit der Platte, deren Urladefähigkeit gelöscht werden soll, muss zugreifbar (importiert) sein.
3. Hat der Benutzer die Urladedateien mit einem zusätzlichen Dateischutz versehen, kann SIR diese nicht löschen. Eine entsprechende Meldung wird ausgegeben.
4. Urladedateien auf Privatplatten können nicht gelöscht werden, wenn im Default-Pubset Dateien mit gleichem Namen, aber mit anderer Lage importiert sind. Eine entsprechende Meldung wird ausgegeben.

13.6.4.10 END - SIR beenden

Mit der END-Anweisung wird SIR beendet. Im Prozedurmodus werden die angegebenen Anweisungen jedoch noch abgearbeitet, als ob die Anweisung START-EXECUTION angegeben worden wäre.

END

13.6.4.11 END-VOLUME-SET-DECLARATION - Anweisungsfolge für Volume-Set beenden

Mit der Anweisung END-VOLUME-SET-DECLARATION wird die Anweisungsfolge für ein Volume-Set beendet. Zusammen mit der Anweisung BEGIN-VOLUME-SET-DECLARATION, die die Deklaration einleitet, umschließt sie einen Anweisungsblock, der für jedes einzurichtende Volume-Set wiederholt werden muss.

END-VOLUME**-SET-DECL**ARATION

13.6.4.12 INITIALIZE-PRIVATE-VOLUME - Privatplatten initialisieren

Mit der Anweisung INITIALIZE-PRIVATE-VOLUME wird eine Privatplatte initialisiert. Die Anweisung wird sofort ausgeführt und nicht in die aktuelle Anweisungsfolge übernommen.

INITIALIZE**-PRIV**ATE**-VOL**UME
VOLUME=vsn ,**UNIT**=mn ,**OVERWR**ITE**-DISK**=***ANY** / vsn **,FORM**AT=***UNCHA**NGED / ***K** / ***NK** ,**F1-LAB**EL**-SIZE**=1000 / <integer 1..32766> ,**CAPACITY**=***STD** / <integer 1000..2 147 483 647> ,**FUNC**TION=***CRE**ATE**-LAB**ELS**-ONLY** / ***FORM**AT**-DISK**

VOLUME=vsn
Gibt die VSN der Platte an, die initialisiert werden soll.

UNIT=mn
Zwei- oder vierstellige Mnemotechnische Bezeichnung des Gerätes, auf dem die Platte montiert ist.

OVERWRITE-DISK
Angabe, ob die Platte vor ihrer Verwendung überprüft werden soll.

OVERWRITE-DISK=*ANY
Die Platte wird ohne Prüfung der bestehenden (alten) VSN initialisiert.

OVERWRITE-DISK=vsn
Die Platte wird nur initialisiert, wenn die bestehende VSN mit der durch OVERWRITE-DISK=vsn angegebenen übereinstimmt.

FORMAT
Legt das Plattenformat der zu initialisierenden Platte fest.

FORMAT=*UNCHANGED
Die Platte wird mit dem selben Format, das sie zuvor hatte, initialisiert; ist die Platte noch unformatiert, im Format K.

FORMAT=*K
Die Platte wird mit Format K initialisiert.

FORMAT=*NK
Die Platte wird mit Format NK2, Allocation-Unit 6, initialisiert.

F1-LABEL-SIZE=1000 / <integer 1..32766>
Gibt die Größe des F1-Labels in PAM-Blöcken an. Standardmäßig werden 1000 PAM-Blöcke reserviert. Die mögliche Größe des F1-Labels liegt im Bereich von 1 bis 32.766 PAM-Blöcken. Der angegebene Wert wird auf Allocation-Unit-Größe aufgerundet.

CAPACITY=
Gibt die maximale Plattenkapazität an.
Dieser Operand wird nur für den Plattentyp A5 (D3435) ausgewertet. Für andere Plattentypen werden die Platten stets mit ihrer maximalen Kapazität initialisiert.Der Operand dient vornehmlich zur Plattenmigration.

CAPACITY=*STD
Die Platte wird mit ihrer maximalen Kapazität zur Verfügung gestellt.

CAPACITY=<integer 1000..2 147 483 647>
Die Platte wird mit der angegebenen Kapazität (in PAM-Seiten) zur Verfügung gestellt.Die Angabe darf nicht über der maximalen Kapazität der Platte liegen.

FUNCTION
Gibt an, ob die Platte mit einem bestimmten Bitmuster initialisiert werden soll.

FUNCTION=*CREATE-LABELS-ONLY
Es werden nur die Kennsätze generiert.

FUNCTION=*FORMAT-DISK
Die Platte wird mit einem bestimmten Bitmuster initialisiert.

13.6.4.13 INITIALIZE-PUBLIC-VOLUME - Public-Platten initialisieren

Mit der Anweisung INITIALIZE-PUBLIC-VOLUME wird eine Public-Platte initialisiert. Die Anweisung wird sofort ausgeführt und nicht in die aktuelle Anweisungsfolge übernommen.

```
INITIALIZE-PUBLIC-VOLUME

 VOLUME=vsn

 ,UNIT=mn

 ,OVERWRITE-DISK=*ANY / vsn

 ,FORMAT=*UNCHANGED / *K / *NK(...)

     *NK(...)

         |    PHYSICAL-BLOCK-SIZE=*2K(...) / *4K(...)

         |         *2K(...)

         |              |    ALLOCATION-UNIT=6 / 8 / 64

         |         *4K(...)

         |              |    ALLOCATION-UNIT=8 / 64

 ,CAPACITY=*STD / <integer 1000..2 147 483 647>

 ,FUNCTION=*CREATE-LABELS-ONLY / *FORMAT-DISK
```

VOLUME=vsn
Gibt die sechsstellige VSN der Platte an, die initialisiert werden soll.

UNIT=mn
Zwei- oder vierstellige mnemotechnische Bezeichnung des Gerätes, auf dem die Platte montiert ist.

OVERWRITE-DISK
Angabe, ob die Platte vor ihrer Verwendung überprüft werden soll.

OVERWRITE-DISK=*ANY
Die Platte wird ohne Prüfung der bestehenden (alten) VSN initialisiert.

OVERWRITE-DISK=vsn
Die Platte wird nur initialisiert, wenn die bestehende VSN mit der durch OVERWRITE-DISK=vsn angegebenen übereinstimmt.

FORMAT
Legt das Plattenformat der zu initialisierenden Platte fest.

FORMAT=*UNCHANGED
Die Platte wird mit dem selben Format, das sie zuvor hatte, initialisiert; ist die Platte noch unformatiert, im Format K.

FORMAT=*K
Die Platte wird mit Format K initialisiert.

FORMAT=*NK(...)
Die Platte wird mit Format NK initialisiert.

PHYSICAL-BLOCK-SIZE
Legt die physikalische Blockgröße auf der NK-Platte fest.

PHYSICAL-BLOCK-SIZE=*2K(...)
NK2-Platte (Standardwert)
Die minimale Transferunit zwischen Platte und Speicher beträgt 2 Kbyte.

ALLOCATION-UNIT=6 / 8 / 64
Die Einheit (Größe der kleinsten Datei) soll 6 Kbyte (Standardwert), 8 oder 64 Kbyte groß sein.

PHYSICAL-BLOCK-SIZE=*4K(...)
NK4-Platte. Die minimale Transferunit beträgt 4 Kbyte.

ALLOCATION-UNIT=8 / 64
Die Einheit (Größe der kleinsten Datei) soll 8 oder 64 Kbyte groß sein.

CAPACITY=
Gibt die maximale Plattenkapazität an.
Dieser Operand wird nur für den Plattentyp A5 (D3435) ausgewertet. Für andere Plattentypen werden die Platten stets mit ihrer maximalen Kapazität initialisiert. Der Operand dient vornehmlich zur Plattenmigration.

CAPACITY=*STD
Die Platte wird mit ihrer maximalen Kapazität zur Verfügung gestellt.

CAPACITY=<integer 1000..2 147 483 647>
Die Platte wird mit der angegebenen Kapazität (in PAM-Seiten) zur Verfügung gestellt.Die Angabe darf nicht über der maximalen Kapazität der Platte liegen.

FUNCTION
Gibt an, ob die Platte mit einem bestimmten Bitmuster initialisiert werden soll.

FUNCTION=*CREATE-LABELS-ONLY
Es werden nur die Kennsätze generiert.

FUNCTION=*FORMAT-DISK
Die Platte wird mit einem bestimmten Bitmuster initialisiert.

13.6.4.14 MODIFY-IPL-VOLUME - Modifizieren einer Urladeplatte

Mit dieser Anweisung kann eine Urladeplatte modifiziert werden.

MODIFY**-IPL-VOL**UME

DISK-NUMBER=n / ***PRIV**ATE(...)

***PRIV**ATE(...)

| **DISK**=vsn

| ,**DEV**ICE**-TYPE**= **STDDISK** / device-type

,**IPL-FILE**= ***UNCHA**NGED / ***STD** / name

,**IPL-REP-FILE**= ***UNCHA**NGED / ***STD** / name

,**SLED-REP-FILE**= ***UNCHA**NGED / ***STD** / name

,**HSI-TYPE**= ***BY-OWN-HSI** / ***390** / ***X86** [1)]

[1)] Der Operandenwert *7500 kann aus Kompatibilitätsgründen noch angegeben werden.

DISK-NUMBER
Gibt die Urladeplatte an, die modifiziert werden soll. Abhängig vom Format der Pubset-Id (siehe Anweisung DECLARE-PUBSET) muss die Nummer der Platte im Pubset angegeben werden (siehe Anweisung CREATE-VOLUME).

DISK-NUMBER=n
Ein-, zwei- oder dreistellige Nummer der Platte innerhalb des Pubset (siehe "CREATE-VOLUME - Platten initialisieren").

DISK-NUMBER=*PRIVATE(...)
Gibt eine private Urladeplatte an, die modifiziert werden soll.

DISK=vsn
VSN der privaten Urladeplatte

DEVICE-TYPE=STDDISK
Standard-Gerätetyp der privaten Urladeplatte

DEVICE-TYPE=device-type
Gerätetyp der privaten Urladeplatte

IPL-FILE
Legt die im SVL zu verankernde IPL-Phase fest.

IPL-FILE=*UNCHANGED
Die Phase bleibt unverändert. Sie ist bereits verankert.

IPL-FILE=*STD
Standarddatei, die verankert werden soll.

IPL-FILE=name
Name der Datei, die verankert werden soll (siehe Hinweise).

IPL-REP-FILE
Legt die im SVL zu verankernde IPL-REP-Datei fest.

IPL-REP-FILE=*UNCHANGED
Diese Datei bleibt unverändert. Sie ist bereits verankert.

IPL-REP-FILE=*STD
Standarddatei, die verankert werden soll.

IPL-REP-FILE=name
Name der Datei, die verankert werden soll (siehe Hinweise).

SLED-REP-FILE
Legt die im SVL zu verankernde SLED-REP-Datei fest.

SLED-REP-FILE=*UNCHANGED
Diese Datei bleibt unverändert. Sie ist bereits verankert.

SLED-REP-FILE=*STD
Standarddatei, die verankert werden soll.

SLED-REP-FILE=name
Name der Datei, die verankert werden soll (siehe Hinweise).

HSI-TYPE
Gibt das HSI an, für das die Urladefähigkeit hergestellt wird (siehe Hinweise).

HSI-TYPE=*BY-OWN-HSI
Die Urladefähigkeit wird für das HSI, auf dem SIR läuft, hergestellt.

HSI-TYPE=*390
Die Urladefähigkeit wird für das /390-HSI hergestellt (/390-Server).

HSI-TYPE=*X86
Die Urladefähigkeit wird für das X86-HSI hergestellt (x86-Server).

Hinweise

1. Die Urladefähigkeit einer Platte ist im Abschnitt „SF-Pubsets“ ausführlich beschrieben.
2. Wenn für eine Datei *STD angegeben wird, muss sie entweder über IMON installiert oder in dem zu bearbeitenden Pubset sein.
3. Wenn DISK-NUMBER=*PRIVATE angegeben wird, müssen sich die Urladedateien im Home-Pubset in einem kompatiblen Format befinden.
4. Die mit name angegebenen Dateien müssen nicht im Pubset katalogisiert sein. Sie werden auf die zu bearbeitende Platte kopiert und dort mit der Erweiterung <.DSKxxx> bzw. <.vsn> für Privatplatten neu katalogisiert.
5. Zur Sicherstellung von Systemkorrekturen während der Systemeinleitung wird von SIR die Datei SYS.NSI.SAVEREP angelegt, wenn die Urladefähigkeit mindestens einer Platte des Pubsets hergestellt wird.
6. Der Benutzer kann mehrere, sich nicht widersprechende Anweisungen für eine Platte angeben, indem er die Operanden für die Dateinamen einzeln angibt:

   ```
   MODIFY-IPL-VOLUME DISK-NUMBER=03,IPL-FILE=IPLDATEI
   ```

   ```
   MODIFY-IPL-VOLUME DISK-NUMBER=03,IPL-REP-FILE=IPLREP
   ```

   `IPLREP` und `IPLDATEI` müssen im zu bearbeitenden Pubset sein.
7. Der Benutzer kann mit MODIFY-IPL-VOLUME eine urladefähige Platte erzeugen, indem er mindestens die IPL-Datei angibt. Wenn die IPL-Datei weder verankert noch angegeben ist, wird die Verarbeitung abgebrochen.
8. Wenn die Rep-Dateien für IPL und SLED weder verankert noch angegeben sind, wird eine entsprechende Meldung ausgegeben und die Bearbeitung fortgesetzt. Die Platte ist anschließend urladefähig.
9. Für die Konsistenz der IPL-Datei, die in der Anweisung angegeben ist, wird die Version der enthaltenen Phase überprüft. Für die Konsistenz zwischen REP-Dateien und IPL-Datei ist der Benutzer verantwortlich.
10. Alle verankerten Dateien erhalten die Dateiattribute BACKUP=E, DESTROY=*YES, ACCESS=*READ und MIGRATE=*FORBIDDEN.
11. Dateischutz mit BASIC-ACL für Urladedateien wird von SIR nicht unterstützt. Hat der Benutzer eine dieser Dateien mit BASIC-ACL geschützt, muss er den Schutz aufheben, damit SIR die Urladedateien modifizieren kann.
12. Eine Platte kann immer nur für ein HSI urladefähig sein. D.h. wird mit MODIFY-IPL-VOLUME die Urladefähigkeit für HSI-TYPE=*X86 festgelegt, so geht die ggf. vorher vorhandene Urladefähigkeit für ein /390-HSI verloren.
13. Der Operand HSI-TYPE muss nur dann angegeben werden, wenn die Urladefähigkeit der Platte nicht für das HSI hergestellt werden soll, auf dem SIR aktuell abläuft.
14. Die Anweisung wird abgewiesen für Platten von Pubsets, die nicht als Home-Pubset eingesetzt werden können:
    - Pubsets mit `SYS-ID=*STD`
    - Pubsets mit `PHYSICAL-BLOCK-SIZE=*4K`
    - SM-Pubsets

14 SMPGEN Konvertieren von SF-Pubsets zu SM-Pubsets

Versionsstand:	**SMPGEN V21.0C**
Privilegierung:	**STD-PROCESSING** (für nicht-privilegierte Funktionen) **TSOS** (für privilegierte Funktionen)

SMPGEN V21.0 kann ab BS2000 V21.0 ablaufen und setzt - bis auf die eingeschränkte Prüffunktion - das Privileg TSOS voraus.

SMPGEN bietet folgende Funktionen an:

- Erzeugen neuer SM-Pubsets aus bestehenden SF-Pubsets
- Erweitern existierender SM-Pubsets um ein oder mehrere SF-Pubsets
- Prüfen der Voraussetzungen für eine Konvertierung (Prüffunktion)

Das Dienstprogramm SMPGEN (System Managed Pubset GENeration) konvertiert einen oder mehrere existierende SF-Pubsets in einen neuen oder bereits existierenden SM-Pubset. Dabei bleiben die auf den Pubsets befindlichen Dateien, Dateieinträge und Metadaten erhalten bzw. sie werden in die auf SM-Pubsets gültige Form überführt.

SMPGEN bietet zusätzlich eine Prüffunktion an, mit der die Systembetreuung im Vorfeld überprüfen kann, ob die angegebenen Pubsets die notwendigen Voraussetzungen für eine erfolgreiche Konvertierung erfüllen.
Diese Prüffunktion ist eingeschränkt auch für nicht-privilegierte Anwender aufrufbar, allerdings nur für die jeweils eigene Benutzerkennung. So kann sich der Anwender über die von ihm selbst zu erledigenden Vorbereitungen informieren.

Das Dienstprogramm SMPGEN besteht aus dem nicht-privilegierten Subsystem SMPGEN-U (SMPGEN-Utility) und dem von ihm intern aufgerufenen privilegierten Subsystem SMPGEN-S (SMPGEN-System).

Zum Aufbau von SF- und SM-Pubsets siehe Handbuch „System Managed Storage“ [8]. Einige wichtige und für das Verständnis von SMPGEN nötige Eigenschaften und Verhaltensweisen von SM-Pubsets und Volume-Sets finden Sie im Abschnitt „Eigenschaften von SM-Pubsets und Volume-Sets“.

14.1 Beispielszenario

Bei der Konvertierung von SF- zu SM-Pubsets wird es i.A. nicht genügen, die SMPGEN-Funktionen CREATE- bzw. MODIFY-SYSTEM-MANAGED-PUBSET aufzurufen. Vielmehr hat der Systembetreuer die Aufgabe, durch vorbereitende Aktionen die Konvertierung für die Anwender möglichst problemlos zu gestalten. Im Anschluss an den SMPGEN-Lauf müssen u.U. noch Nacharbeiten erledigt werden, um den neuen SM-Pubset wie gewünscht betreiben zu können.
Insbesondere sind die von SMPGEN erzeugten Pubsets noch nicht HSMS-unterstützt. Um migrierte Dateien bearbeiten zu können, muss erst eine HSMS-Umgebung hergestellt werden. In Abhängigkeit von der Komplexität der bestehenden HSMS-Umgebungen gibt es verschiedene Möglichkeiten der Umstellung der HSMS-Umgebung, die ebenfalls verschiedene Vorbereitungen erfordern; siehe dazu Handbuch „System Managed Storage“ [8].

Im Folgenden wird dargestellt, welche Aktionen bei der Konvertierung zu SM-Pubsets notwendig und sinnvoll sein können:

- Falls mehrere SF-Pubsets zusammengefasst werden sollen, ist es u.U. sinnvoll, zunächst einen Konsistenz-Check auszuführen. Er soll herausfinden, ob eine Konvertierung möglich ist.
- Soll eine Konvertierung erfolgen, so werden die Anwender darüber benachrichtigt, dass in allen BS2000-Prozeduren Verweise auf ehemalige Pubset-Kennungen durch eine Referenz auf die neue Pubset-Kennung zu ersetzen sind.

  Normalerweise sind die Anwender davon nur wenig betroffen, da sie die Dateien über die Standard-Katalogkennung im Benutzerkatalog adressieren.
  Auch Datenhaltungssysteme sind i.A. wenig betroffen, da sie eine Oberfläche besitzen, die durch eine spezifische Adressierungslogik die Lage der Datei vor dem Anwender verbirgt.
  Falls Anwendungen existieren, die mit bestimmten Pubsets arbeiten sollen, müssen sie beim Anlegen der Dateien neben der Pubset-Kennung des SM-Pubsets in Zukunft die betreffende Volume-Set-Kennung oder eine den gewünschten Volume-Set referenzierende Storage-Klasse angeben. Die Angabe einer Volume-Set-Kennung ist jedoch nur zulässig, wenn die entsprechende Benutzerkennung das Recht auf physikalische Allokierung im Pubset hat. Dieses Recht muss der Benutzerkennung im Anschluss an die Konvertierung explizit erteilt werden.

  Falls nur ein SF-Pubset umgewandelt werden soll, ist es sinnvoll, diesen Pubset zunächst mittels PVSREN umzubenennen und die ehemalige Pubset-Kennung als Pubset-Kennung des SM-Pubsets anzugeben. Dadurch ist gewährleistet, dass sich die Katalogkennung in den Dateinamen nicht ändert und die JCL der Anwender nicht angepasst werden muss. Ebenso kann die Katalogkennung als Standard-Katalogkennung im Benutzerkatalog beibehalten werden.

  Analog kann verfahren werden, wenn mehrere SF-Pubsets zusammengefasst werden sollen, aber eines davon eine ausgezeichnete Rolle im System spielt; es kann dann sinnvoll sein, diesen Pubset umzubenennen und seine ehemalige Pubset-Kennung als Pubset-Kennung des SM-Pubsets anzugeben.

  Der Control-Volume-Set sollte möglichst ausfallsicher und deshalb auch nicht allzu groß sein. Auf jeden Fall muss er jedoch genug Platz für die neu anzulegenden Kataloge bieten (siehe Beschreibung im Abschnitt „Erzeugen neuer SM-Pubsets“).
- Falls bereits ein Konsistenz-Check erfolgt ist und Konflikte bei Dateinamen oder Guards ermittelt wurden, fordert der Systembetreuer die Anwender auf, sie zu beseitigen.
- Der Systembetreuer beseitigt die Namenskonflikte bei Systemdateien. Systemdateien, die im SVL verankert sind (IPL, SLEDSAVE usw.), sollten nicht gelöscht, sondern entweder umbenannt oder mit der SIR-Anweisung DELETE-IPL-FACILITY entfernt werden.
  Zur Lösung von Namenskonflikten bei (HSMS-) Migration-Directories sind die Directories umzubenennen und die zugehörigen Archivdefinitionen den neuen Directory-Namen anzupassen.

- Paging-Dateien müssen ggf. mit `/REDUCE-PAGING-AREA` deaktiviert werden.

- Der Systembetreuer prüft, ob zu den SF-Pubsets Snapsets existieren. Wenn dies der Fall ist, löscht er die Snapsets.
- Der Systembetreuer bereitet die Eingaben für den SMPGEN-Lauf und evtl. eine Prozedur zum Importieren des SM-Pubsets vor.
- Falls der SM-Pubset mit HSMS-Unterstützung betrieben werden soll (immer notwendig, falls migrierte Dateien existieren), bereitet der Systembetreuer die Umstellung bzw. Anpassung der HSMS-Umgebung vor.
  (Die Katalogeinträge der migrierten Dateien bleiben bei der Umstellung erhalten, doch ist kein Recall dieser Dateien möglich, solange keine HSMS-Unterstützung etabliert ist. Auch existiert zunächst noch keine Backup-Umgebung. Es ist kein direkter Zugriff auf vorangegangene Dateisicherungen möglich.)
- Eventuell ist ein erneuter Konsistenz-Check im Offline-Modus sinnvoll. Die Anwender werden ggf. zur Beseitigung von Inkonsistenzen aufgefordert.
- Sind alle Inkonsistenzen beseitigt, so kann zu einem mit den Anwendern vereinbarten Zeitpunkt, zu dem die SF-Pubsets nicht in Betrieb sind, mit der Konvertierung zu SM-Pubsets begonnen werden.

  Eventuell sind letzte Vorbereitungen für die Umstellung der HSMS-Umgebung erforderlich, z.B. die Verlagerung der auf die S1-Ebene migrierten Dateien auf die S2-Ebene, damit eine ordnungsgemäße Konvertierung erfolgen kann.

  Zunächst wird eine logische Vollsicherung und eine FDDRL-Sicherung der Pubsets erstellt. Anschließend werden die Pubsets mit der SMPGEN-Anweisung CREATE- bzw. MODIFY-SYSTEM-MANAGED-PUBSET konvertiert.

  Falls zu diesem Zeitpunkt noch oder schon wieder Inkonsistenzen bestehen, werden sie aufgelistet; es erfolgt keine Konvertierung, die Pubsets bleiben als SF-Pubsets importierbar.
- Nach erfolgreicher Konvertierung sind in den Benutzerkatalogen aller Systeme, in denen der Pubset importiert werden soll, die Standard-Katalogkennungen, die den Volume-Set-Kennungen entsprechen, durch die Pubset-Kennung des SM-Pubsets zu ersetzen.
- In den Guards anderer Pubsets, in denen auf ein Programm im neuen SM-Pubset Bezug genommen wird, muss der Programmname modifiziert werden. Der Systembetreuer kann den Anwendern eine entsprechende SDF-P-Prozedur zur Verfügung stellen.
- Der neue SM-Pubset kann jetzt normal importiert werden, wobei ACTUAL-JOIN=*STD gelten muss.
  Ein erweiterter SM-Pubset muss erneut importiert werden, um noch nach dem SMPGEN-Lauf ausstehende Recovery-Maßnahmen durchzuführen. Auch hier ist ACTUAL-JOIN=*STD (Default-Wert) anzugeben.
  SIZE-TOLERANCE=*YES muss, falls nötig, zuvor im MRSCAT-Eintrag des SM-Pubsets eingetragen werden (mit `/MODIFY-PUBSET-CACHE-ATTRIBUTES`) und gilt dann einheitlich für alle Volume-Sets.
- Falls der SM-Pubset Volume-Sets mit USAGE=*WORK enthalten soll, müssen diese mit `/MODIFY-PUBSET-DEFINITION-FILE` dem importierten Pubset hinzugefügt und mit `/MODIFY-PUBSET-PROCESSING` in Betrieb genommen werden.
- Jetzt kann - falls gewünscht - die HSMS-Unterstützung und -Umgebung etabliert bzw. angepasst werden.
- Die Kontingente und pubset-weiten Berechtigungen können den speziellen Bedingungen angepasst werden, z. B. durch Einschränkung der Work-Kontingente. Gruppen können erzeugt bzw. die Gruppenkonstellation vervollständigt werden. Falls gewünscht, können die gruppenspezifischen Kontingente geändert werden.
- Falls gewünscht, kann das Standard-Dateiformat (FILE-FORMAT) geändert werden.
- Falls Guard-Bedingungsfehler gemeldet wurden, müssen die Guard-Eigentümer die betreffenden Guard-Einträge überprüfen und ggf. an die neue Pubset-Kennung anpassen.
- Falls auf einem der konvertierten SF-Pubsets Software mit IMON installiert wurde und sich die Katalogkennung der installierten Dateien geändert hat, muss das Software-Configuration-Inventory (SCI) mit `/MODIFY-IMON-`

SCI geändert werden. Dabei muss für OLD-NAME der Name des SF-Pubsets vor der Konvertierung und bei NEW-NAME der Name des SM-Pubsets angegeben werden.

- Nach erfolgreicher Pubset-Konvertierung benachrichtigt der Systembetreuer die Anwender.

14.2 Voraussetzungen für den Ablauf von SMPGEN

Anforderungen an Betriebssystem und Auftragstask

- Falls in mindestens einem der SF-Pubsets ein Guards-Katalog existiert (d.h. die Datei $TSOS.SYSCAT.GUARDS), muss das Subsystem GUARDS geladen sein.
- Die Ausgabe in eine S-Variable wird nur bei Einsatz des kostenpflichtigen Produkts SDF-P unterstützt. Auch das Subsystem VAS muss geladen sein.

Zusätzlich zu diesen Anforderungen müssen - abhängig von der aufgerufenen Funktion - weitere Bedingungen für die Anweisungen MODIFY- und CREATE-SYSTEM-MANAGED-PUBSET beachtet werden:

- Auf keinem der Pubsets darf die Datei $TSOS.SYSACL.FILE vorhanden sein, da seit BS2000/OSD-BC V5.0 das Subsystem SRPMFACL (FACS) nicht mehr vorhanden ist.
- Für jeden Pubset muss ein MRSCAT-Eintrag existieren.
- Die Pubsets dürfen weder im lokalen noch in einem anderen System importiert oder anderweitig belegt sein. Es darf kein Import oder Export (`/IMPORT-PUBSET` bzw. `/EXPORT-PUBSET`) für sie laufen.
- Im Katalog der Pubsets dürfen keine Dateien auf Net-Storage-Volumes verzeichnet sein. Diese Katalogeinträge müssen vor einem SMPGEN-Lauf gelöscht werden (`/EXPORT-FILE` oder, siehe nächsten Absatz, `/REMOVE-NET-STORAGE-VOLUME`, Operand `FILES-ON-VOLUME=*EXPORT`).
- Den Pubsets müssen alle Net-Storage-Volume entzogen worden sein (`/REMOVE-NET-STORAGE-VOLUME`). Dem bestehenden SM-Pubset dürfen hingegen Net-Storage-Volumes zugeordnet sein.
- Bei Aufruf durch einen nicht-privilegierten Anwender (für den der Wert CHECK-NAME-CONSISTENCY(PUBSET-STATE=*IMPORTED) implizit angenommen wird) müssen die SF-Pubsets im Zustand „local accessible“ sein, d. h. lokal importiert.
  Wird USER-ID=*ALL spezifiziert, so müssen entweder die Pubsets system-exklusiv importiert oder das lokale System der Pubset-Master sein.

Anforderungen an die Ausgangs-Pubsets (SF-Pubsets)

Für die in einen SM-Pubset zu überführenden SF-Pubsets gelten folgende Bedingungen:

- Sie müssen in der aktuell laufenden Version des Betriebssystems mit ACTUAL-JOIN= *STD, d.h. ohne Neuerzeugung des Benutzerkatalogs, importierbar sein. Dies setzt voraus, dass sie bereits mindestens einmal importiert waren.
- Es darf insgesamt nicht mehr als 8190 verschiedene Benutzerkennungen in den SF-Pubsets geben (= Maximalzahl der Benutzer in einem SM-Pubset).
- Maximal 255 SF-Pubsets können in einen SM-Pubset übergeführt werden.
- Die Paging-Dateien auf den SF-Pubsets dürfen nicht aktiv sein (d.h. sie müssen ggf. vorher deaktiviert werden).
- Zu den SF-Pubsets dürfen keine Snapsets existieren (d.h. die Systemverwaltung muss diese ggf. vorher löschen). Leere Snapset-Kataloge werden ggf. von SMPGEN gelöscht.
- Falls der Aufrufer nicht den Operanden S1-MIGRATED-FILES=*ALLOWED spezifiziert, gilt außerdem: Es dürfen keine auf die S1-Ebene migrierte Dateien katalogisiert sein. (d.h. diese müssen ggf. vorher nach S2 migriert werden).

Die ersten drei der genannten Bedingungen gelten ebenso für die Prüffunktion.

Außerdem muss der Aufbau der neuen Systemdateien möglich sein und es dürfen keine Namenskonflikte entstehen. Die genauen Voraussetzungen sind in der Beschreibung der Anweisungen aufgeführt.

Hinweise für Large-Objects-Pubsets

- Ist einer der Ausgangs-Pubsets ein Large-Objects-Pubset, so wird auch der erzeugte SM-Pubset ein Large-Objects-Pubset. Die Anzeigen dafür befinden sich sowohl im MRSCAT des erzeugten SM-Pubsets, als auch im DMS-Eintrag des Control-Volume-Sets.
- Aus Sicherheitsgründen werden die Anzeigen auch in den DMS-Eintrag der anderen Volume-Sets hinterlegt.

Starten von SMPGEN

Die SMPGEN-Funktionalität wird vom nicht-privilegierten Subsystem SMPGEN-U bereitgestellt. SMPGEN-U sollte aus Performancegründen von der Systembetreuung vorab geladen werden mit `/START-SUBSYSTEM SUBSYSTEM-NAME=SMPGEN-U`.

SMPGEN wird mit `/START-SMPGEN` im laufenden Betrieb gestartet. Das Subsystem SMPGEN-S wird intern von SMPGEN-U aufgerufen. Beide Subsysteme sind nicht mittels DSSM austauschbar. Parallele Produktversionen werden nicht unterstützt.

14.3 Prüffunktion (Konsistenz-Check)

Mit der Prüffunktion (auch Konsistenz-Check genannt) werden SF-Pubsets daraufhin geprüft, ob sie in einen neu zu erzeugenden oder bereits bestehenden SM-Pubset konvertiert werden können.

Die Prüfunktion von SMPGEN kann mit den folgenden Anweisungen aufgerufen werden:

```
//MODIFY-SYSTEM-MANAGED-PUBSET
```

```
//CREATE-SYSTEM-MANAGED-PUBSET
```

Abhängig davon, ob der Anwender privilegiert (mit dem Privileg TSOS ausgestattet) oder nicht privilegiert ist, bezieht sich die Überprüfung auf folgende Pubsets:

- nicht-privilegierte Anwender:
  Es werden nur Pubsets geprüft, auf denen die Benutzerkennung des Anwenders eingetragen ist. Der Anwender wird über Namenskonflikte informiert und kann sie ggf. beseitigen.
  Pubsets, auf die der Anwender keinen Zugriff hat, weil seine Benutzerkennung dort nicht eingetragen ist, werden vom Konsistenz-Check ausgeschlossen; SMPGEN gibt eine entsprechende Meldung aus.
  Eine Weiterverarbeitung (Erzeugen oder Erweitern von SM-Pubsets) ist nicht möglich.
- privilegierter Anwender (Benutzerkennung TSOS, Privileg TSOS):
  Der privilegierte Anwender kann die Prüfungen der Pubsets (unter Angabe des Operanden OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY) entweder nur für seine eigene Benutzerkennung oder für alle Benutzerkennungen starten (Unteroperand USER-ID=*OWN/*ALL).

Die Ausführung der Anweisung erfolgt entweder in der laufenden Task oder in einer neu erzeugten Batch-Task. Im letztgenannten Fall kann der Anwender parallel weiter mit SMPGEN arbeiten oder das Programm beenden (Operand EXECUTION-MODE=*SYNCHRONOUS/*ASYNCHRONOUS).

Ebenfalls abhängig von der Privilegierung ist, wann der Konsistenz-Check durchgeführt wird:

- nicht-privilegierte Anwender:
  Für den nicht-privilegierten Anwender werden die Überprüfungen immer bei laufender Pubset-Session durchgeführt. Das Ergebnis enthält dadurch Zufallskomponenten, da ständig Namen neu katalogisiert oder gelöscht werden können. Eine zeitliche Synchronisation der miteinander zu vergleichenden Informationen erfolgt nicht.
- privilegierter Anwender (Benutzerkennung TSOS, Privileg TSOS):
  Der privilegierte Anwender hat die Wahl, die Überprüfungen in der laufenden Pubset-Session oder bei exklusiven Zugriff durch SMPGEN durchzuführen (Operand PUBSET-STATE=*IMPORTED/*NOT-IMPORTED).

  Wird die zweite Variante (exklusiver Zugriff) gewählt, so werden die gemeinschaftlichen Platten durch SMPGEN system-exklusiv belegt. In den MRSCAT-Einträgen der Pubsets wird eine Anzeige gesetzt, dass sie gerade durch SMPGEN bearbeitet werden. Diese Pubsets sind dann während der Überprüfungen nicht importierbar. Ihre MRSCAT-Einträge können während dessen nicht gelöscht werden. Der von `/SHOW-PUBSET-ATTRIBUTES` angezeigte Pubset-Zustand ist „inaccessible“.

Der Konsistenz-Check läuft auch immer (noch einmal) implizit ab, bevor die tatsächliche Konvertierung von SF-Pubsets in SM-Pubsets erfolgt.

14.3.1 Was wird geprüft?

Es werden folgende Überprüfungen vorgenommen:

- Überprüfung der Datei- und Jobvariablen-Namen
- Überprüfung der Guard-Namen
- Weitere Überprüfungen (Systemdateien, migrierte Dateien, usw.)

Überprüfung der Datei- und Jobvariablen-Namen

1. Es darf in den zu konvertierenden Pubsets keine Datei- oder Jobvariablen-Namen geben, die bei Ergänzung mit der künftigen Katalogkennung (i.A. der künftigen SM-Pubset-Kennung) länger als 54 Zeichen werden. Ebenso darf es keine Dateigenerationsgruppen-Namen geben, die bei Ergänzung mit der künftigen Katalogkennung länger als 47 Zeichen werden (bzw. der Name der einzelnen Dateigeneration mit Generationsnummer länger als 54 Zeichen).
2. Guard-Bedingungen, die sich auf Programmnamen beziehen, werden überprüft, ob sie bei der Konvertierung in ein SM-Pubset ungültig werden, aber nicht automatisch angepasst werden können. Dies kann folgende Ursachen haben:
   - Die Katalogkennung des SM-Pubsets ist länger als die des SF-Pubsets, und durch die implizite Anpassung der Pfadnamen würde die maximal zulässige Größe des Bedingungsbereichs überschritten. In diesem Fall bleibt der gesamte
     Bedingungsbereich unverändert und muss vom Eigentümer des Guards angepasst werden.
   - Die Katalogkennung des Programmnamens enthält Wildcard-Symbole. In diesem Fall werden sowohl die Katalogkennungen der zu konvertierenden Pubsets als auch die neue Katalogkennung des SM-Pubsets gegen die Wildcard-
     Katalogkennung geprüft. Erfüllt mindestens eine der alten Katalogkennungen die Wildcard-Bedingung, die neue SM-Pubset-Katalogkennung jedoch nicht, so wird ein Konflikt gemeldet. Der Programmname mit Wildcard-Symbolen bleibt in jedem Fall unverändert.

   Dies ist jedoch insofern eine Ausnahme, als die Pubsets dennoch konvertiert werden können; die Guard-Einträge werden anschließend vom Anwender angepasst.
3. Es darf in den zu konvertierenden SF-Pubsets keine zwei Pfadnamen oder Jobvariablen-Namen geben, die sich nur durch die Katalogkennung unterscheiden.

   Dies betrifft auch Systemdateien, wie z.B. $TSOS.SYSTEM.MRSCAT; der Systembetreuer muss dafür sorgen, dass diese nur auf einem der Pubsets existieren. Ausnahmen bilden solche Systemdateien, die bei der Konvertierung in einen SM-Pubset automatisch gelöscht oder umbenannt werden.

   Dies sind die Dateien:

   $TSOS.SYS.PUBSET.CONFIG
   $TSOS.SYS.PVS.SHARER.CONTROL
   $TSOS.SYSCAT.GUARDS
   $TSOS.SYSEAM
   $TSOS.SYSPBN
   $TSOS.SYSSRPM
   $TSOS.SYSSRPM.BACKUP
   $TSOS.TSOSCAT
   $TSOS.TSOSCAT.#Mnn (mit nn=00..99)
   $TSOS.TSOSCAT.#Pnn (mit nn=00..99)
   $TSOS.TSOSCAT.#Jnn (mit nn=00..99)
   $TSOS.TSOSCAT.$PFI

$TSOS.TSOSCAT.$NLO
$TSOS.TSOSCAT.$NLC
$TSOS.TSOSJOIN

Überprüfung der Guard-Namen

Es dürfen keine zwei Guard-Namen in den Pubsets existieren, die sich nur durch die Katalogkennung unterscheiden.

Weitere Überprüfungen

Folgende Dateinamen dürfen in keinem der Pubsets existieren:

```
$TSOS.SYSCAT.STORCLS
$TSOS.SYSCAT.VSETLST
$SYSHSMS.SYSCAT.HSMS
```

Keiner der Dateinamen darf mit $TSOS.SYSWRK.SMPGEN. beginnen.
Vom Anwender angelegte Dateien $TSOS.SYS.PUBSET.CONFIG oder Dateien, deren Namen mit $TSOS.TSOSCAT. beginnen, dürfen in keinem der Pubsets existieren.

Wird eine bestimmte, vom Anwender spezifizierbare Anzahl von Namenskonflikten überschritten, so wird der Konsistenz-Check abgebrochen. (Unter Namenskonflikt ist hier zu verstehen: Auftreten von zu langen Datei- oder Jobvariablen-Namen oder Auftreten von Dateien, Jobvariablen oder Guards in mehreren Pubsets.)

Falls der Anwender privilegiert und nicht S1-MIGRATED-FILES=*ALLOWED spezifiziert ist, wird auch überprüft, dass keine Dateien auf die S1-Ebene migriert sind.
(Dies ist nicht zulässig; sie müssen ggf. nach S2 migriert werden; der Schlechtfall zählt pro Pubset als 1 Konflikt.)

> Es wird nicht überprüft, ob genug Platz für die neu anzulegenden SM-Pubset-Verwaltungsdateien vorhanden ist.

Hinweise

Standardmäßig wird im Fehlerfall nicht nur die Anzahl der Fehler, sondern auch eine detaillierte Fehlerinformation ausgegeben, der zu entnehmen ist, welche Dateinamen, Jobvariablen und Guards mehrfach existieren und auf welchen Pubsets sie sich befinden.

Außerdem werden alle zu langen Datei- und Jobvariablen-Namen und alle Guard-Bedingungskonflikte (siehe "Überprüfung der Datei- und Jobvariablen-Namen", Punkt 2) ausgegeben. Die Fehlerlisten sind jeweils nach Benutzerkennung bzw. Pubset-/Benutzerkennung sortiert.
Standardmäßig wird die Information in Listenform ausgegeben. Falls die Ausgabe an der Datensichtstation erfolgt, kann jeweils „weitergeblättert“ werden.
Alternativ oder zusätzlich können diese Informationen aber auch in einer S-Variablen abgelegt werden (Operandenwert STRUCTURED-OUTPUT).

SMPGEN setzt eine korrekte SYSFILE-Umgebung voraus. Soll die Funktion in einer neu zu erzeugenden Batch-Task ausgeführt werden, so muss der Standard-Pubset der Ablauf-Benutzerkennung (oder der in ACS definierte SPOOL-Pubset) importiert sein. Scheitern SYSOUT- oder SYSLST-Ausgaben auch bei wiederholtem Versuch, so wird die SMPGEN-Funktion mit Fehler beendet.

14.3.2 Ausgaben

Fehlermeldungen sowie die Meldung über das Starten der Funktion und die abschließende Erfolgsmeldung werden auf dasselbe Ausgabeziel gelenkt wie die Informationen über Inkonsistenzen, d.h. auf das bzw. die durch den operanden TEXT-OUTPUT spezifizierte(n) Ausgabeziel(e).
Im Dialog werden die Meldung über das Starten der Funktion sowie die Erfolgsmeldung zusätzlich auch auf SYSOUT ausgegeben, falls TEXT-OUTPUT=*SYSLST ist.

Werden die Informationen über Inkonsistenzen nicht in Text-Form ausgegeben (TEXT-OUTPUT=*NONE), so werden die Fehlermeldungen, die Meldung über das Starten der Funktion sowie die abschließende Erfolgsmeldung auf SYSOUT ausgegeben.

Folgende Informationen werden ausgegeben:

1. Tabelle der beteiligten Pubsets mit Angabe von Plattenformat, Allokierungseinheit, Verfügbarkeit, maximaler Ein-/Ausgabelänge und Anzeige, ob ein Cache definiert ist. (Nähere Informationen über die Cache-Definition erhalten Sie mit `/SHOW-PUBSET-PARAMETERS`.)

   Zum Format siehe Maske 1 im Abschnitt "Ausgaben in Bildschirmmasken".
   Bei dieser Maske hat der Anwender die Möglichkeit, durch eine entsprechende Eingabe in der Fußzeile den Abbruch der Funktionsausführung zu veranlassen.
2. Protokollierungsmeldungen:
   - Ergebnismeldung (Konsistenz-Check erfolgreich oder nicht).
     Die Ergebnismeldung wird, unabhängig vom TEXT-OUTPUT-Operanden, im Dialog immer auch auf SYSOUT ausgegeben.
   - In Sonderfällen bzw. bei fehlerhaftem Abbruch gehen evtl. weitere Meldungen der Ergebnismeldung voraus.
   - Ist der Anwender privilegiert, so wird die Größe der „alten“ Pubset-Verwaltungsdateien ausgegeben, um den Platzbedarf auf dem künftigen Control-Volume-Set abschätzen zu können; bei Spezifikation der Control-Volume-Set-Kennung zusätzlich der freie Platz auf dem hierfür vorgesehenen Pubset.
3. Im Fehlerfall, bzw. wenn Warnungen auftraten, vor Ausgabe der Ergebnismeldung (Wurde STRUCTURE-OUTPUT angegeben, so werden die Informationen - unabhängig von der gewünschten Informationstiefe - in der S-Variablen hinterlegt.)

   bei **ERROR-INFOMATION=*SUMMARY**:
   - Anzahl der Dateien-, Jobvariablen- und Guard-Namen-Konflikte

     Bei Dateigenerationsgruppen werden nur die Konflikte der Gruppennamen, nicht jedoch die Konflikte bei den einzelnen Dateigenerationen gezählt.
   - Anzahl der zu langen Datei- und Jobvariablen-Namen

     Bei Dateigenerationsgruppen wird nur die Generationsgruppe, nicht jedoch die einzelnen Dateigenerationen gezählt.
   - Anzahl der Guard-Bedingungen, die nicht automatisch an die neue Pubset-Kennung angepasst werden können (Warnungen)

   bei **ERROR-INFORMATION=*DETAIL** erfolgt eine detailliertere Fehlerinformation:
   - Listen der Namenskonflikte

     Bei Dateigenerationsgruppen wird nur die Generationsgruppe gemeldet; die Konflikte bei den einzelnen Dateigenerationen werden nicht gemeldet und nicht gezählt.
     Zum Format der Listen siehe Masken 3-8 im Abschnitt "Ausgaben in Bildschirmmasken".

- Listen der zu langen Datei- und Jobvariablen-Namen

  Bei Dateigenerationsgruppen wird nur die Generationsgruppe (mit Zusatz FGG) gemeldet; die einzelnen Dateigenerationen werden nicht gemeldet und nicht gezählt.
  Zum Format der Listen siehe Masken 9-10 im Abschnitt "Ausgaben in Bildschirmmasken".

- Liste der Guard-Bedingungen, die nicht automatisch an die neue Pubset-Kennung angepasst werden können.
  Zum Format der Liste siehe Maske 11 im Abschnitt "Ausgaben in Bildschirmmasken").

Unabhängig von der gewünschten Informationstiefe werden dem privilegierten Anwender folgende Fehlerinformationen (vor allen weiteren Fehlerinformationen) ausgegeben:

- Gibt es nach S1 migrierte Dateien in einem der Pubsets und ist nicht S1-MIGRATED-FILES=*ALLOWED spezifiziert, so wird dies als Fehler
  gemeldet (jeweils eine Meldung pro fehlerhaftem Pubset).

- Liste der unzulässigen Dateien

  Hiermit sind die in der vorangehenden Anweisungsbeschreibung genannten Dateien gemeint, die in den Pubsets nicht katalogisiert sein dürfen, weil
  SMPGEN Arbeitsdateien oder neue Verwaltungsdateien unter diesem Namen aufbaut.
  Zum Format der Liste siehe Maske 2 im Abschnitt "Ausgaben in Bildschirmmasken".

Die folgende Tabelle erläutert Namen und Inhalte der S-Variablen. Nähere Informationen zu S-Variablen finden Sie im Handbuch „Kommandos“ [1], Band 1.

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Mehrfach vorkommender Dateiname ohne Katalog- und Benutzerkennung	var(*LIST).F.CONFL (*LIST).F	S	" <filename>	INF=*DETAIL
Anzahl der überprüften Pubsets, auf denen der mehrfach vorkommende Dateiname existiert	var(*LIST).F.CONFL (*LIST).NUM-OF-PUBSET	I	" <integer>	INF=*DETAIL
Katalogkennung der betreffenden Pubsets	var(*LIST).F.CONFL (*LIST).PUBSET (*LIST). CAT-ID	S	" <catid>	INF=*DETAIL
Benutzerkennung des mehrfach vorkommenden Dateinamens	var(*LIST).F.CONFL (*LIST).USER-ID	S	" <userid>	INF=*DETAIL
Katalogkennung des zu langen Dateinamens	var(*LIST).F-LEN. ERROR(*LIST).CAT-ID	S	" <catid>	INF=*DETAIL
Zu langer Dateiname ohne Katalog-und Benutzerkennung	var(*LIST).F-LEN. ERROR(*LIST).F	S	" <filename>	INF=*DETAIL
Kennzeichnung, ob es sich um eine Dateigenerationsgruppe handelt (Y) oder nicht (N)	var(*LIST),F-LEN. ERROR(*LIST).FGG-IND	S	Y N	INF=*DETAIL
Anzahl der Zeichen, um die der zu lange Dateiname die erlaubten 54 Zeichen (mit Katalogkennung) überschreitet	var(*LIST).F-LEN. ERROR(*LIST).NUM-OF- CHAR	I	" <integer>	INF=*DETAIL
Benutzerkennung des zu langen Dateinamens	var(*LIST).F-LEN. ERROR(*LIST). USER-ID		" <userid>	INF=*DETAIL
Anzahl der zu langen Dateinamen	var(*LIST).F-LEN. NUM-OF-ERRORS	I	<integer>	INF=*DETAIL/ *SUMMARY
Anzahl der Dateinamens-Konflikte	var(*LIST).F.NUM-OF-CONFL	I	<integer>	INF=*DETAIL/ *SUMMARY

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Anzahl der Fälle, in denen eine Guard-Bedingung nicht automatischan die neue Pubset-Id angepasst wird	var(*LIST).GUARD-COND.NUM-OF-WARN	I	<integer>	INF=*DETAIL/ *SUMMARY
Katalogkennung des Guards bei nicht anpassbarer Guard-Bedingung	var(*LIST).GUARD-COND.WARN (*LIST). CAT-ID	S	" <catid>	INF=*DETAIL
Name des Guards bei nicht anpassbarer Guard-Bedingung ohne Katalog- und Benutzerkennung	var(*LIST).GUARD-COND.WARN (*LIST). GUARD	S	" <guard-name>	INF=*DETAIL
Grund, warum eine Guard-Bedingungnicht automatisch anpassbarist	var(*LIST).GUARD-COND.WARN (*LIST). REASON	S	" <c-string>	INF=*DETAIL
Benutzerkennung des betreffendenGuards	var(*LIST).GUARD-COND.WARN (*LIST). USER-ID	S	" <userid>	INF=*DETAIL
Mehrfach vorkommender Guard-Name ohne Katalog-undBenutzerkennung	var(*LIST).GUARD. CONFL(*LIST). GUARD	S	" <guard-name>	INF=*DETAIL
Anzahl der überprüften Pubsets, auf denen der mehrfach vorkommendeGuard existiert	var(*LIST).GUARD. CONFL(*LIST). NUM-OF-PUBSET	I	" <integer>	INF=*DETAIL
Katalogkennung des betreffenden Pubsets	var(*LIST).GUARD. CONFL(*LIST). PUBSET(*LIST). CAT-ID	S	" <catid>	INF=*DETAIL
Benutzerkennung des betreffendenGuards	var(*LIST).GUARD. CONFL(*LIST). USER-ID	S	" <userid>	INF=*DETAIL
Anzahl der Guard-Namens-Konflikte	var(*LIST).GUARD. NUM-OF-CONFL	I	<integer>	INF=*DETAIL/ *SUMMARY
Pfadname der unzulässigen Datei	var(*LIST). INADMIS.F(*LIST)	S	" <filename>	INF=*DETAIL/ *SUMMARY
		I	<integer>	

Anzahl der unzulässigen Dateinamen (0, wenn die Überprüfung von einem nicht-privilegierten Anwender durchgeführt wurde)	var(*LIST).INADMIS.NUM-OF-ERRORS			INF=*DETAIL/*SUMMARY
Mehrfach vorkommender Jobvariablenname ohne Katalog- und Benutzerkennung	var(*LIST).JV.CONFL(*LIST).JV	S	"<filename>	INF=*DETAIL

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Anzahl der überprüften Pubsets, auf denen die mehrfach vorkommendeJobvariable existiert	var(*LIST).JV.CONFL(*LIST).NUM-OF-PUBSET	I	"<integer>	INF=*DETAIL
Katalogkennung des betreffenden Pubsets	var(*LIST).JV.CONFL(*LIST).PUBSET(*LIST).CAT-ID	S	"<catid>	INF=*DETAIL
Benutzerkennung des mehrfach vorkommenden Jobvariablennamens	var(*LIST).JV.CONFL(*LIST).USER-ID	S	"<userid>	INF=*DETAIL
Katalogkennung des zu langen JV-Namens	var(*LIST).JV-LEN.ERROR(*LIST).CAT-ID	S	"<catid>	INF=*DETAIL
Zu langer Jobvariablenname ohne Katalog- und Benutzerkennung	var(*LIST).JV-LEN.ERROR(*LIST).JV	S	"<filename>	INF=*DETAIL
Anzahl der Zeichen, um die der zu lange Jobvariablenname die erlaubten 54 Zeichen (mit Katalogkennung) überschreitet	var(*LIST).JV-LEN.ERROR(*LIST).NUM-OF-CHAR	I	"<integer>	INF=*DETAIL
Benutzerkennung des zu langen JV-Namens	var(*LIST).JV-LEN.ERROR(*LIST).USER-ID		"<userid>	INF=*DETAIL
Anzahl der zu langen JV-Namen	var(*LIST).JV-LEN.NUM-OF-ERRORS	I	<integer>	INF=*DETAIL/*SUMMARY
Anzahl der JV-Namens-Konflikte	var(*LIST).JV.NUM-OF-CONFL	I	<integer>	INF=*DETAIL/*SUMMARY

14.4 Erzeugen neuer SM-Pubsets

Mit der Anweisung CREATE-SYSTEM-MANAGED-PUBSET kann der privilegierte Anwender bis zu 255 SF-Pubsets in einen SM-Pubset überführen, wobei jeder SF-Pubset zu einem Volume-Set des SM-Pubsets wird. Dabei bleiben die Anwenderdaten sowie die zugehörige Verwaltungs- und Schutzinformation bis auf einige Einschränkungen (siehe "Einschränkungen") erhalten.

Beim Erzeugen neuer SM-Pubsets aus bestehenden SF-Pubsets werden systemintern folgende Schritte ausgeführt:

1. Syntaktische Überprüfung der Anweisung und Vorschau auf das Ergebnis

   Falls die Task im Dialog läuft, wird nach der syntaktischen Überprüfung der Anweisung CREATE-SYSTEM-MANAGED-PUBSET der Aufbau des zu erzeugenden SM-Pubsets tabellarisch in der Bildschirmmaske 12 aufgelistet und nach erfolgtem Importieren der Pubsets erneut in Bildschirmmaske 13 aufgelistet; in beiden Masken hat der Anwender die Möglichkeit, den Abbruch der Funktion zu veranlassen.

2. Impliziter Konsistenz-Check

   Bevor die Platten verändert werden, erfolgt implizit ein Konsistenz-Check der beteiligten Pubsets (siehe auch Abschnitt „Prüffunktion (Konsistenz-Check)“). Werden hierbei Inkonsistenzen festgestellt, wird die Funktion mit Fehlermeldung abgebrochen; detaillierte Fehlerinformationen werden ausgegeben (siehe auch die Ausgaben der Prüffunktion im Abschnitt „Ausgaben“).

   Können Guard-Bedingungen nicht angepasst werden (Punkt 2, "Was wird geprüft?"), so werden diese im Rahmen des Konsistenz-Checks gemeldet. Der Konsistenz-Check gilt als ohne Fehler (jedoch mit Warnungen) beendet.
   Die möglichen Folgen eines Abbruchs im Anschluss an einen erfolgreichen Konsistenz-Check sind dem Abschnitt „Fehlerverhalten“ zu entnehmen.

3. Exklusive Belegung der Platten durch SMPGEN

   Sowohl während des Konsistenz-Checks als auch der nachfolgenden Bearbeitung sind die gemeinschaftlichen Platten durch SMPGEN exklusiv belegt. In den MRSCAT-Einträgen der Pubsets ist eine Anzeige gesetzt, dass sie gerade durch SMPGEN bearbeitet werden.

   Die Pubsets sind während des gesamten Vorgangs nicht importierbar, und ihre MRSCAT-Einträge können weder gelöscht noch modifiziert werden.

   Für den neu zu erzeugenden SM-Pubset wird ebenfalls ein MRSCAT-Eintrag angelegt, der während des Vorgangs nicht gelöscht oder modifiziert werden kann, und in dem der Pubset als im Entstehen gekennzeichnet ist.

4. Umwandlung der Pubsets

   Die Platten-SVLs werden im Laufe des Umwandlungsvorganges verändert, sodass die ehemaligen SF-Pubsets nicht mehr als solche importierbar sind. Ihre Zugehörigkeit zum SM-Pubset und zum Control-Volume-Set wird im SVL fest verankert.

   Als niedrigste Betriebssystem-Version, in der der Pubset unterstützt wird, wird BS2000/OSD-BC V6.0 im SVL hinterlegt.

5. Löschen aller MRSCAT-Einträge der umgewandelten Pubsets

   Im Anschluss an eine erfolgreiche Umwandlung in einen SM-Pubset werden die MRSCAT-Einträge der zu Volume-Sets umgewandelten Pubsets gelöscht. Der MRSCAT-Eintrag des neu erzeugten SM-Pubsets bleibt erhalten.
   Die im Pubset-MRSCAT-Eintrag einzustellenden Werte wie z.B. CACHE-SIZE-TOLERANCE werden auf den Standardwert gesetzt.

Simulations-Modus

Wenn der Operand OPERATIONAL-MODE=*SIMULATION angegeben wird, wird die Funktion nicht real ausgeführt. Es erfolgt lediglich ein Syntax-Check und eine Auflistung der angegebenen Volume-Sets. Nähere Informationen siehe Operandenbeschreibung.

Check-Modus

Wenn der Operand OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY angegeben wird, wird nur die Prüffunktion ausgeführt. Nähere Informationen siehe "Prüffunktion (Konsistenz-Check)".

Hinweise

Die auf den Pubsets befindlichen Dateien bleiben in ihrer Lage unverändert.

Die zugehörigen Katalogeinträge bleiben inhaltlich unverändert erhalten, bis auf die Ausnahme, dass die Datei auf einem Pubset liegt, der zum Volume-Set mit der Eigenschaft AVAILABLITY=*HIGH wird (und es sich nicht um eine temporäre Datei handelt):
In diesem Fall wird im Katalogeintrag die logische Dateieigenschaft AVAILABILITY=*HIGH gesetzt, was bewirkt, dass die Datei nicht durch Migration auf einen Volume-Set mit geringerer Verfügbarkeit verlagert wird.

Die Katalogeinträge von Dateien, die keine Platzzuweisung auf den SF-Pubsets besitzen (z.B. migrierte Dateien, Privatplatten- oder Banddateien) bleiben ebenso wie die Jobvariablen inhaltlich erhalten.

BS2000 bedient ab OSD/BC V11.0 nach Möglichkeit nur mehr das Katalogformat EXTRA LARGE, siehe Handbuch „Systembetreuung“ [5]. Die bisherigen Katalogformate NORMAL und LARGE sind obsolet. Es sind folgende Besonderheiten zu beachten:

1. Kataloge einer Vorgängerversion im Format NORMAL oder LARGE werden automatisch in das Format EXTRA LARGE konvertiert, sobald der Pubset exklusiv oder als Master importiert wird. Dabei werden auch die Spezialkataloge umbenannt. Derart konvertierte Kataloge können auch in einer Vorgängerversion wieder verwendet werden.
2. Der SM-Pubset wird mit Katalogformat EXTRA LARGE erzeugt. Die Konvertierung der Kataloge wird allerdings nicht während des SMPGEN-Laufs vorgenommen, sondern beim ersten Import nach diesem Lauf, siehe den vorherigen Punkt. Deshalb wird empfohlen, den SM-Pubset unmittelbar nach dem SMPGEN-Lauf zu importieren.
3. Bei der Neugenerierung eines SM-Pubsets wird jeweils nur je ein Spezialkatalog TSOSCAT.#M00, TSOSCAT.#J00 und TSOSCAT.#P00 angelegt.
   Reichen diese Kataloge in ihrer Größe nicht zur Aufnahme aller Spezialkatalogeinträge aus, so wird empfohlen, zunächst einen SM-Pubset zu erzeugen, der aus dem zukünftigen Control-Volume-Set besteht, und diesen Pubset dann zu importieren.
   Anschließend können mit `/ADD-CATALOG-FILE` zusätzliche Spezialkataloge anlegt und der SM-Pubset in einem zweiten SMPGEN-Lauf um die restlichen SF-Pubsets erweitert werden.

Wenn GUARDS im Einsatz ist, wird beim Exportieren der Volume-Sets nach erfolgreicher Pubset-Konvertierung die Meldung `PRO6009 ERROR WHEN CLOSING GUARDS CATALOG` für jeden betroffenen Volume-Set ausgegeben; diese Meldung kann ignoriert werden.

Auch die Guards bleiben inhaltlich erhalten. In Guards, die ein in den zu konvertierenden Pubsets katalogisiertes Programm referenzieren, wird die im Pfadnamen enthaltene Katalogkennung durch die neue Pubset-Kennung ersetzt.
(Dies gilt jedoch nur für die Guard-Einträge innerhalb des neuen SM-Pubset; Guards-Einträge in anderen Pubsets

werden nicht angepasst, außerdem kann es nicht automatisch anpassbare Guards geben; siehe Abschnitt „Einschränkungen“).

Die Benutzereinträge werden in das für SM-Pubsets gültige Format überführt, wobei Gruppenstrukturen evtl. verloren gehen (siehe Abschnitt „Einschränkungen“). Existiert eine Benutzerkennung zwar auf einem der Pubsets, jedoch nicht auf demjenigen Pubset, von dem die Logon-Information übernommen werden soll, so wird die Benutzerkennung im SM-Pubset als „gesperrt“ geführt (USER-LOCKED=*YES), da keine Accounting-Information zur Verfügung steht. Für die Benutzung des SM-Pubsets als Daten-Pubset ist dies nicht relevant.
Existiert eine Benutzerkennung auf mehreren der zum SM-Pubset zusammenzuführenden Pubsets, so wird dies so behandelt, als sei es ein identischer Anwender.

SMPGEN setzt eine korrekte SYSFILE-Umgebung voraus. Soll die Funktion in einer neu zu erzeugenden Batch-Task ausgeführt werden, so muss der Standard-Pubset der Ablauf-Benutzerkennung (oder der in ACS definierte Spool-Pubset) importiert sein. Scheitern SYSOUT- oder SYSLST-Ausgaben auch bei wiederholtem Versuch, so wird die SMPGEN-Funktion mit Fehler beendet.

Als DEFAULT FILE FORMAT wird das (physikalische) Format des Control-Volume-Sets eingetragen.

Werden mehrere Pubsets unterschiedlicher Formate zusammengefasst, so muss das von SMPGEN erzeugte DEFAULT FILE FORMAT evtl. geändert werden.

Beispiel

Das Format des Control-Volume-Sets ist K, alle anderen Volume-Sets sind keylos (NK); in diesem Fall werden standardmäßig die Dateien auf dem Control-Volume-Set angelegt und dieses überfüllt.

ACHTUNG!

- CREATE-SYSTEM-MANAGED-PUBSET OPERATIONAL-MODE= *OPERATION sollte nie ohne vorhergehende FDDRL-Sicherung der umzuwandelnden Pubsets aufgerufen werden, da bei einem fehlerhaften Abbruch alle Pubsets zerstört und nicht mehr zugreifbar sein können.
- Eine logische Sicherung mittels HSMS wird ebenfalls empfohlen. Hierbei kann ein neues Backup-Archiv erzeugt werden, das zur Herstellung der HSMS-Umgebung für den neuen SM-Pubset verwendet werden kann. Anleitungen zur Umstellung der HSMS-Umgebung sind im Leitfaden für die SMS-Migration gegeben, siehe Handbuch „System Managed Storage“ [8].
- Spool-Aufträge für Dateien, die in den zu konvertierenden Pubsets katalogisiert sind, sollten nicht mehr ausstehen, da sie durch die Umbenennung der Katalogkennung nicht mehr ausgeführt werden können.
- SECOS-spezifische Benutzerinformationen (z.B. Gruppeninformation, Privilegien) gehen auch bei Angabe von KEEP-USER-ATTRIBUTES=*ALL vollständig verloren, wenn das Betriebssystem während der Pubset-Konvertierung ohne SECOS betrieben wird.

14.4.1 Bestimmung von Kontingenten und anderen Eigenschaften

Die Anzahl der Dateieinträge, Jobvariablen und belegten Seiten in den Pubsets wird addiert, ebenso die CE- und Public-Space-Limits. Die Kontingente werden wie folgt gesetzt:

- FILE-NUMBER-LIMIT = Summe(FILE-NUMBER-LIMIT), summiert über alle SF-Pubsets
- JV-NUMBER-LIMIT = Summe(JV-NUMBER-LIMIT), summiert über alle SF-Pubsets
- TOTAL-SPACE-LIMIT(PERM) = MAXIMUM
- S0-LEVEL-SPACE-LIMIT(PERM) = Summe(PUBLIC-SPACE-LIMIT), summiert über alle SF-Pubsets
- HIGH-PERF-SPACE-LIMIT(PERM) = MAXIMUM
- VERY-HIGH-PERF-SPACE-LIMIT(PERM) = MAXIMUM
- HIGH-AVAIL-SPACE-LIMIT(PERM) = Summe(PUBLIC-SPACE-LIMIT), summiert über alle SF-Pubsets, die zu HIGH-AVAIL-Volume-Sets werden
- TOTAL-SPACE-LIMIT(TEMP) = Summe(TEMP-SPACE-LIMIT), summiert über alle SF-Pubsets
- HIGH-PERF-SPACE-LIMIT(TEMP) = MAXIMUM
- VERY-HIGH-PERF-SPACE-LIMIT(TEMP) = MAXIMUM
- TOTAL-SPACE-LIMIT(WORK) = MAXIMUM
- HIGH-PERF-SPACE-LIMIT(WORK) = MAXIMUM
- VERY-HIGH-PERF-SPACE-LIMIT(WORK) = MAXIMUM

Dabei bedeutet MAXIMUM, dass für das betreffende Unter-Kontingent keine eigene Schranke existiert, sondern es jeweils dynamisch an das umfassendere Ober-Kontingent angepasst wird. So kann z.B. der für VERY-HIGH-PERF-Dateien zur Verfügung stehende Platz höchstens so groß sein wie der gesamte für HIGH-PERF-Dateien zur Verfügung stehende Platz. Für das oberste Kontingent bedeutet MAXIMUM den maximal zuweisbaren Wert (z.B. 2147483647).

Die Werte für PUBLIC-SPACE-EXCESS, DMS-TUNING-RESOURCES und PHYSICAL-ALLOCATION werden so gesetzt, dass die am wenigsten restriktive Bedingung von allen betroffenen SF-Pubsets übernommen wird. Reihenfolge mit aufsteigender Restriktion:

- PUBLIC-SPACE-EXCESS: ALLOWED, TEMPORARILY-ALLOWED, NO
- DMS-TUNING-RESOURCES: EXCLUSIVE-USE, CONCURRENT-USE, NONE
- PHYSICAL-ALLOCATION: ALLOWED, NOT-ALLOWED

Bei der Ermittlung der SPACE-USED-Werte werden alle Dateien eines Benutzers entsprechend ihrer Lage auf den Hintergrundebenen, ihren Eigenschaften temporär/permanent, ihren Performance- und Verfügbarkeitsattributen in den entsprechenden Belegungszählern berücksichtigt. Die auf einem HIGH-AVAILABILITY-Volume-Set liegenden Dateien werden, da sie durch SMPGEN das Dateiattribut HIGH-AVAIL bekommen, in das HIGH-AVAIL-Kontingent eingerechnet.
Falls Gruppenstrukturen von einem der Pubsets übernommen werden, gilt für diese:

- PUBLIC-SPACE-LIMIT: MAXIMUM
- TEMP-SPACE-LIMIT: MAXIMUM
- WORK-SPACE-LIMIT: MAXIMUM
- FILE-NUMBER-LIMIT: MAXIMUM
- JV-NUMBER-LIMIT: MAXIMUM

Die Werte PUBLIC-SPACE-EXCESS und DMS-TUNING-RESOURCES und PHYSICAL-ALLOCATION werden von dem als ATTRIBUTE-ORIGIN gekennzeichneten Pubset übernommen.

Als DEFAULT FILE FORMAT wird das (physikalische) Format des Control-Volume-Sets eingetragen.

Die Allokierungs-Standards (für Primary und Secondary Allocation) werden von dem SF-Pubset übernommen, der zum Control-Volume-Set wird.
Die Sättigungs-Schwellwerte bleiben in jedem Volume-Set so, wie sie für den umgewandelten SF-Pubset eingestellt waren.

Auf dem Control-Volume-Set werden folgende zur Verwaltung des SM-Pubsets benötigten Dateien angelegt:

- $TSOS.SYS.PUBSET.CONFIG: Konfigurationsbeschreibung
- $TSOS.TSOSCAT.#M00: Katalog der migrierten und nospace-Dateien
- $TSOS.TSOSCAT.#P00: Katalog der Privatplatten- und Banddateien
- $TSOS.TSOSCAT.#J00: Katalog der Jobvariablen
- $TSOS.TSOSCAT.$PFI: Index aller Katalogeinträge
- $TSOS.TSOSCAT.$NLO: Liste aller Dateinamen
- $TSOS.TSOSCAT.$NLC: Kopie der Dateinamensliste

Katalogeinträge, die in SM-Pubsets in Spezialkatalogen geführt werden (das sind alle außer die Dateien, die Speicherplatz im Pubset belegen), werden in diese neu zu erstellenden Spezialkataloge ausgelagert.

Die Dateien $TSOS.TSOSCAT werden in Volume-Set-spezifische Namen umbenannt.

Die Dateien $TSOS.SYSCAT.GUARDS und $TSOS.SYSSRPM werden auf allen Pubsets, die nicht zum Control-Volume-Set werden, gelöscht.
Die Guard- und Benutzereinträge werden in die entsprechende Datei auf dem Control-Volume-Set übertragen.

Falls EAM auf dem SM-Pubset betrieben werden soll, muss die EAM-Datei neu eingerichtet werden. Folgende Dateien werden gelöscht:

- $TSOS.SYSEAM
- $TSOS.SYSPBN
- $TSOS.TSOSJOIN
- $TSOS.SYSSRPM.BACKUP

Die Cache-Konfiguration wird aus den MRSCAT-Einträgen der zu konvertierenden Pubsets in die Pubset-Konfigurationsdatei des SM-Pubsets übernommen. Ausnahme: die pubsetbezogene Cache-SIZE-TOLERANCE, die - wenn erwünscht - für den neuen Pubset neu eingestellt werden muss.

Der im SVL des zum Control-Volume-Set konvertierten Pubsets verankerte Shareability-Wert bestimmt die Mehrbenutzbarkeit des neu erzeugten SM-Pubsets.

Die Systemkennungen (SYSIDs) in den SVLs bleiben i.A. erhalten; nur beim Control-Volume-Set wird sie im Falle einer 1 Byte langen SM-Pubset-Kennung durch diese ersetzt.

14.4.2 Einschränkungen

Die Anwenderdaten sowie die zugehörige Verwaltungs- und Schutzinformation bleiben bis auf folgende Einschränkungen erhalten:

- Falls einer oder mehrere der in Volume-Sets überführten SF-Pubsets als Home-Pubset einsetzbar war(en), so geht diese Möglichkeit verloren; ein SM-Pubset kann nicht als Home-Pubset verwendet werden.
- Nur die Guard-Einträge in den angegebenen, zu konvertierenden Pubsets werden inhaltlich angepasst und bleiben gültig. (Soweit sie anpassbar sind; ist dies nicht der Fall, werden die im Abschnitt „Ausgaben“ beschriebenen Warnungen ausgegeben; die betreffenden Guard-Einträge muss der Guard-Eigentümer selbst anpassen.)

  Guard-Einträge in anderen Pubsets können durch die Konvertierung ungültig werden. Das ist dann der Fall, wenn die Zugriffsbedingung den Namen eines Programms enthält, das in den SF-Pubsets katalogisiert war. Da sich der Pfadname der Programme durch die Konvertierung geändert hat, wird das zugriffsberechtigte Programm nicht mehr gefunden. Der Guard-Eigentümer muss daher diese Guards selbst modifizieren.
- Die SYSEAM-Dateien gehen verloren.
- Die nur Home-Pubset-relevanten Privilegien und Berechtigungen gehen - in Abhängigkeit vom Parameter KEEP-USER-ATTRIBUTES - entweder ganz verloren oder werden nur von genau einem der SF-Pubsets übernommen.
- Die Logon-Kennwörter gehen - in Abhängigkeit vom Parameter KEEP-USER-ATTRIBUTES - entweder ganz verloren oder werden nur von genau einem der SF-Pubsets übernommen.
  (Im letztgenannten Fall gehen die Logon-Kennwörter derjenigen Benutzer verloren, die keinen Benutzereintrag auf dem bezeichneten SF-Pubset haben. Da das System auf SM-Pubsets hinterlegte Kennwörter jedoch nicht prüft, ist dies nicht von Belang.)
- Die Gruppenstrukturen gehen - in Abhängigkeit vom Parameter KEEP-USER-ATTRIBUTES - entweder ganz verloren oder werden nur von genau einem der SF-Pubsets übernommen.
  Die Benutzerkennungen, für die keine Gruppenstruktur übernommen wird, werden der Gruppe *UNIVERSAL zugeordnet.
  (Dies sind im einen Fall alle Benutzerkennungen, im andern Fall alle diejenigen Benutzerkennungen, die nicht auf dem bezeichneten SF-Pubset eingetragen oder die dort nicht einer Gruppe zugeordnet sind.)
- Die Datei SYSSRPM.BACKUP geht verloren. Das bedeutet, dass die Pubset-Konvertierung nicht gleichzeitig mit dem Umstieg auf einen anderen Benutzerkatalog erfolgen kann.
- Spool-Aufträge gehen verloren, wenn sie sich auf Dateien mit der Katalogkennung eines konvertierten SF-Pubsets beziehen, da die Katalogkennung in der Spool-Auftragsschlange nicht angepasst wird. Von einer vergangenen Session evtl. noch erhaltene Druckaufträge müssen also neu angestartet und hierzu gehörige temporäre Dateien müssen anschließend explizit vom Systembetreuer gelöscht werden.
- Die Datei HSMS-EXCEPT-FILE verliert ihre Wirkung. Um den Schutz gegen unerwünschte Dateiverdrängung wiederherzustellen, kann der Systembetreuer auf dem SM-Pubset eine Pubset-spezifische EXCEPT-FILE anlegen und diese bei Herstellung der HSMS-Umgebung dem Pubset zuordnen, oder er kann den betroffenen Dateien die Dateieigenschaft MIGRATE=*FORBIDDEN geben.
- Pubset-Konfigurationsdateien, die von einem abgebrochenen Rekonfigurationsauftrag oder abgebrochener Cache-Betriebsänderung stammen, werden implizit gelöscht, wobei eine Warnung ausgegeben wird.
  Es wird empfohlen, die Pubset-Konvertierung nicht bei unvollständig ausgeführter Rekonfiguration oder Cache-Betriebsänderung durchzuführen, sondern zuvor den Pubset mit `/RESUME-PUBSET-RECONFIGURATION` zu bereinigen.

14.4.3 Ausgaben

Fehler- und Protokollierungsmeldungen sowie die abschließende Erfolgsmeldung werden auf dasselbe Ausgabeziel gelenkt wie die Informationen über Inkonsistenzen, d.h. auf das bzw. die durch TEXT-OUTPUT spezifizierte(n) Ausgabeziel(e).
Im Dialog werden die Erfolgsmeldung und die Protokollierungsmeldungen zusätzlich auch auf SYSOUT ausgegeben, falls TEXT-OUTPUT=*SYSLST ist.

Werden die Informationen über Inkonsistenzen nicht in Text-Form ausgegeben (TEXT-OUTPUT=*NONE), so werden die Fehlermeldungen, die Meldung über das Starten der Funktion sowie die abschließende Erfolgsmeldung auf SYSOUT ausgegeben.

1. Im Dialog:Tabelle der Volume-Sets mit den spezifizierten logischen Eigenschaften. Diese Tabelle wird unabhängig vom TEXT-OUTPUT-Operanden auf SYSOUT ausgegeben.

   Zum Format siehe Maske 12 im Abschnitt "Ausgaben in Bildschirmmasken".
   Bei dieser Maske hat der Anwender die Möglichkeit, durch eine entsprechende Eingabe in der Fußzeile den Abbruch der Funktionsausführung zu veranlassen.
2. Tabelle der Volume-Sets mit Angabe von Plattenformat und spezifizierten logischen Eigenschaften.

   Zum Format siehe Maske 13 im Abschnitt "Ausgaben in Bildschirmmasken".
   Bei dieser Maske hat der Anwender die Möglichkeit, durch eine entsprechende Eingabe in der Fußzeile den Abbruch der Funktionsausführung zu veranlassen.
3. Protokollierungsmeldungen:
   - Meldung, dass die Funktion bzw. die funktionsausführende Task gestartet wurde
   - Fortschrittsprotokollierung: Meldung, dass die GUARDS-/SRPM-/CMS-Konvertierung gestartet wurde
   - Ergebnis des Konsistenz-Checks
     Die Ergebnismeldung wird unabhängig vom TEXT-OUTPUT-Operanden im Dialog immer auch auf SYSOUT ausgegeben.
   - In Sonderfällen, bzw. bei fehlerhaftem Abbruch, gehen evtl. weitere Meldungen der Ergebnismeldung voraus.
4. Im Fehlerfall:
   - Auflistung der unzulässigen Dateien, Namenskonflikte, zu langen Namen und Guard-Bedingungsfehler wie bei CHECK-NAME-CONSISTENCY (siehe Abschnitt „Ausgaben“).

     Zum Format siehe Masken 2 bis 11 im Abschnitt "Ausgaben in Bildschirmmasken".
   - Gibt es nach S1 migrierte Dateien in einem der Pubsets und ist nicht S1-MIGRATED-FILES=*ALLOWED spezifiziert, so wird dies als Fehler gemeldet. (Jeweils eine Meldung pro fehlerhaftem Pubset).

Ausgaben im Simulations-Modus

1. Ergebnismeldung der Syntaxprüfung
2. Tabelle der Volume-Sets mit den spezifizierten logischen Eigenschaften.

   Zum Format siehe Maske 12 im Abschnitt "Ausgaben in Bildschirmmasken".

14.5 Erweitern eines bestehenden SM-Pubsets

Mit der Anweisung MODIFY-SYSTEM-MANAGED-PUBSET kann der privilegierte Anwender ein bereits bestehendes SM-Pubset um SF-Pubsets erweitern, wobei jeder SF-Pubset zu einem Volume-Set des SM-Pubsets wird. Bei der Ausführung der Anweisung wird kontrolliert, dass max. 255 Volume-Sets im erweiterten SM-Pubset enthalten sind.

Die Anwenderdaten sowie die zugehörige Verwaltungs- und Schutzinformation der aufzunehmenden SF-Pubsets und des bestehenden SM-Pubsets bleiben bis auf einige Einschränkungen (siehe „Abschnitt „Einschränkungen“) erhalten.

Beim Erweitern von SM-Pubsets werden systemintern folgende Schritte ausgeführt:

1. Syntaktische Überprüfung der Anweisung und Vorschau auf das Ergebnis

   Falls die Task im Dialog läuft, wird nach der syntaktischen Überprüfung der Anweisung MODIFY-SYSTEM-MANAGED-PUBSET der Aufbau des zu erweiternden SM-Pubsets tabellarisch in der Bildschirmmaske 12 aufgelistet und nach erfolgtem Importieren der Pubsets erneut in Bildschirmmaske 13 aufgelistet; in beiden Masken hat der Anwender die Möglichkeit, den Abbruch der Funktion zu veranlassen.

   Bei MODIFY-SYSTEM-MANAGED-PUBSET werden in den Bildschirmmasken 12 und 13 nur die neu hinzugekommenen Volume-Sets aufgelistet. Die bereits bestehenden Volume-Sets des SM-Pubsets werden in der Liste nicht angezeigt.

2. Impliziter Konsistenz-Check

   Bevor die Platten verändert werden, erfolgt implizit ein Konsistenz-Check der beteiligten Pubsets (siehe auch Abschnitt „Prüffunktion (Konsistenz-Check)“). Werden hierbei Inkonsistenzen festgestellt, wird die Funktion mit Fehlermeldung abgebrochen; detaillierte Fehlerinformationen werden ausgegeben (siehe auch die Ausgaben der Prüffunktion im Abschnitt „Ausgaben“).

   Können Guard-Bedingungen nicht angepasst werden (Punkt 2 im Abschnitt "Was wird geprüft"), so werden diese im Rahmen des Konsistenz-Checks gemeldet. Der Konsistenz-Check gilt als ohne Fehler (jedoch mit Warnungen) beendet.
   Die möglichen Folgen eines Abbruchs im Anschluss an einen erfolgreichen Konsistenz-Check sind dem Abschnitt „Fehlerverhalten“ zu entnehmen.

3. Exklusive Belegung der Platten durch SMPGEN

   Sowohl während des Konsistenz-Checks als auch der nachfolgenden Bearbeitung sind die gemeinschaftlichen Platten durch SMPGEN exklusiv belegt. In den MRSCAT-Einträgen der Pubsets ist eine Anzeige gesetzt, dass sie gerade durch SMPGEN bearbeitet werden.

   Die Pubsets sind während des gesamten Vorgangs nicht importierbar, und ihre MRSCAT-Einträge können weder gelöscht noch modifiziert werden. Der MRSCAT-Eintrag des zu erweiternden SM-Pubsets wird als „in Generierung“ gekennzeichnet.

4. Umwandlung der Pubsets

   Die Platten-SVLs der SF-Pubsets werden im Laufe des Umwandlungsvorganges verändert, sodass die ehemaligen SF-Pubsets nicht mehr als solche importierbar sind. Ihre Zugehörigkeit zum SM-Pubset und zum Control-Volume-Set wird im SVL fest verankert.

   Als niedrigste Betriebssystem-Version, in der der Pubset unterstützt wird, wird BS2000/OSD-BC V6.0 im SVL hinterlegt.

5. Löschen aller MRSCAT-Einträge der umgewandelten Pubsets

   Im Anschluss an eine erfolgreiche Erweiterung des SM-Pubsets werden die MRSCAT-Einträge der zu Volume-Sets umgewandelten SF-Pubsets gelöscht.

6. Nacharbeiten

   Nach Ausführung der SMPGEN-Anweisung zur Erweiterung eines SM-Pubsets, ist die Erweiterung des SM-Pubsets jedoch noch nicht vollständig erfolgt. Auch ist unmittelbar nach Ausführung der SMPGEN-Anweisung zur Erweiterung eines SM-Pubsets keine erneute Erweiterung des SM-Pubsets möglich. Die Erweiterung eines SM-Pubsets ist erst nach dem erfolgreichen Import des SM-Pubsets vollständig abgeschlossen und danach ist die nächste Erweiterung des SM-Pubsets mit SMPGEN erlaubt.

   Beim nachfolgenden Import - der vom Anwender angestoßen werden muss - werden die CMS-Kataloge des SM-Pubsets um überflüssige Jobvariableneinträge bereinigt.

Simulations-Modus

Wenn der Operand OPERATIONAL-MODE=*SIMULATION angegeben wird, wird die Funktion nicht real ausgeführt. Es erfolgt lediglich ein Syntax-Check und eine Auflistung der angegebenen Volume-Sets. Nähere Informationen siehe Operandenbeschreibung.

Check-Modus

Wenn der Operand OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY angegeben wird, wird nur die Prüffunktion ausgeführt. Nähere Informationen siehe "Prüffunktion (Konsistenz-Check)".

Hinweise

Wenn GUARDS im Einsatz ist, wird beim Exportieren der Volume-Sets nach erfolgreicher Pubset-Konvertierung die Meldung `PRO6009  ERROR WHEN CLOSING GUARDS CATALOG` für jeden betroffenen Volume-Set ausgegeben; diese Meldung kann ignoriert werden.

BS2000 bedient ab OSD/BC V11.0 nach Möglichkeit nur mehr das Katalogformat EXTRA LARGE, siehe Handbuch „Systembetreuung“ [5]. Die bisherigen Katalogformate NORMAL und LARGE sind obsolet. Werden SM-Pubsets unter Beteiligung von SF-Pubsets erweitert, sind folgende Besonderheiten zu beachten:

1. Kataloge einer Vorgängerversion im Format NORMAL oder LARGE werden automatisch in das Format EXTRA LARGE konvertiert, sobald der Pubset exklusiv oder als Master importiert wird. Dabei werden auch die Spezialkataloge umbenannt. Derart konvertierte Kataloge können auch in einer Vorgängerversion wieder verwendet werden.
2. Während der Erweiterung eines SM-Pubsets stehen nur die Spezialkataloge zur Verfügung, die bereits existieren. Diese werden allerdings bei Bedarf während des SMPGEN-Laufs bis auf ihre maximal mögliche Größe erweitert (32008 4K-Blöcke).

SMPGEN setzt eine korrekte SYSFILE-Umgebung voraus. Soll die Funktion in einer neu zu erzeugenden Batch-Task ausgeführt werden, so muss der Standard-Pubset der Ablauf-Benutzerkennung (oder der in ACS definierte SPOOL-Pubset) importiert sein. Scheitern SYSOUT- oder SYSLST-Ausgaben auch bei wiederholtem Versuch, so wird die SMPGEN-Funktion mit Fehler beendet.

Werden mehrere Pubsets unterschiedlicher Formate zusammengefasst, so muss das von SMPGEN erzeugte DEFAULT FILE FORMAT evtl. geändert werden.

Beispiel

Das Format des Control-Volume-Sets ist K, alle anderen Volume-Sets sind keylos (NK); in diesem Fall werden standardmäßig die Dateien auf dem Control-Volume-Set angelegt und dieses überfüllt.

ACHTUNG!

- MODIFY-SYSTEM-MANAGED-PUBSET OPERATIONAL-MODE=*OPERATION sollte nie ohne vorhergehende FDDRL-Sicherung der umzuwandelnden SF-Pubsets aufgerufen werden, da bei einem fehlerhaften Abbruch alle SF-Pubsets zerstört und nicht mehr zugreifbar sein können.
- Eine logische Sicherung mittels HSMS wird ebenfalls empfohlen. Hierbei kann ein neues Backup-Archiv erzeugt werden, das zur Herstellung der HSMS-Umgebung für den erweiterten SM-Pubset verwendet werden kann. Anleitungen zur Umstellung der HSMS-Umgebung sind im Leitfaden für die SMS-Migration gegeben, siehe Handbuch „System Managed Storage“ [8].
- Spool-Aufträge für Dateien, die in den zu konvertierenden SF-Pubsets katalogisiert sind, sollten nicht mehr ausstehen, da sie durch die Umbenennung der Katalogkennung nicht mehr ausgeführt werden können.

14.5.1 Einschränkungen

Die Anwenderdaten sowie die zugehörige Verwaltungs- und Schutzinformation bleiben bis auf folgende Einschränkungen erhalten:

- Die SYSEAM-Dateien auf den SF-Pubsets gehen verloren.
- Die nur Home-Pubset-relevanten Privilegien und Berechtigungen auf den SF-Pubsets gehen ganz verloren.
- Die Logon-Passworte auf den SF-Pubsets gehen ganz verloren.
- Die Gruppenstrukturen auf den neu aufzunehmenden SF-Pubsets gehen ganz verloren.
- Die Datei SYSSRPM.BACKUP auf den neu aufzunehmenden SF-Pubsets geht verloren. Das bedeutet, dass die Pubset-Konvertierung nicht gleichzeitig mit dem Umstieg auf einen anderen Benutzerkatalog erfolgen kann.
- Pubset-Konfigurationsdateien auf den neu aufzunehmenden SF-Pubsets, die von einem abgebrochenen Rekonfigurationsauftrag oder abgebrochener Cache-Betriebsänderung stammen, werden implizit gelöscht, wobei eine Warnung ausgegeben wird.
  Es wird empfohlen,die Pubset-Konvertierung nicht bei unvollständig ausgeführter Rekonfiguration oder Cache-Betriebsänderung durchzuführen, sondern zuvor den Pubset mit `/RESUME-PUBSET-RECONFIGURATION` zu bereinigen.

Für die Lage der auf den SF-Pubsets befindlichen Dateien, die Katalogeinträge von Dateien und Jobvariablen auf den SF-Pubsets sowie die Guards auf den SF-Pubsets gelten die gleichen Aussagen wie bei der Konvertierung von SF-Pubsets in neue SM-Pubsets.

- Die Benutzereinträge auf den SF-Pubsets werden in das für SM-Pubsets gültige Format übergeführt, wobei Gruppenstrukturen verloren gehen.

  Existiert eine Benutzerkennung zwar auf einem der SF-Pubsets, jedoch nicht auf demjenigen Pubset, von dem die Logon-Information übernommen werden soll, so wird die Benutzerkennung im SM-Pubset als „gesperrt“ geführt (USER-LOCKED=*YES), da keine Accounting-Information zur Verfügung steht.

  Für die Benutzung des SM-Pubsets als Daten-Pubset ist dies nicht relevant. Existiert eine Benutzerkennung auf mehreren der in den SM-Pubset aufzunehmenden SF-Pubsets oder auf dem SM-Pubset, so wird dies so behandelt, als sei es ein identischer Anwender. Die Anzahlen der Dateieinträge, JVs und belegten Seiten in den aufzunehmenden SF-Pubsets und dem SM-Pubset werden addiert, ebenso die CE- und Public-Space-Limits.

Auch für die Aktualisierung bzw. Bildung der benutzerspezifischen Kontingente, SPACE-USED-Werte und Einstellungen gelten die gleichen Aussagen wie bei der Konvertierung von SF-Pubsets in neue SM-Pubsets, wobei Summierungen über alle aufzunehmenden SF-Pubsets und den SM-Pubset erfolgen.

- Die Werte für PUBLIC-SPACE-EXCESS, DMS-TUNING-RESOURCES und PHYSICAL-ALLOCATION werden so gesetzt, dass die am wenigsten restriktive Bedingung von allen betroffenen SF-Pubsets übernommen wird. Reihenfolge mit aufsteigender Restriktion:
  - PUBLIC-SPACE-EXCESS: ALLOWED, TEMPORARILY-ALLOWED, NO
  - DMS-TUNING-RESOURCES: EXCLUSIVE-USE, CONCURRENT-USE, NONE
  - PHYSICAL-ALLOCATION: ALLOWED, NOT-ALLOWED
- Das DEFAULT FILE FORMAT und die Allocation Defaults (Standard für Primary und Secondary Allocation) des SM-Pubsets bleiben unverändert. Die Sättigungs-Schwellwerte bleiben in jedem Volume-Set so, wie sie für den umgewandelten SF-Pubset eingestellt waren.

Auf dem Control-Volume-Set werden folgende zur Verwaltung des SM-Pubsets benötigten Dateien aktualisiert:

- $TSOS.SYS.PUBSET.CONFIG: Konfigurationsbeschreibung
- $TSOS.TSOSCAT.#Mnn (nn=00..99): Katalog(e) der migrierten und nospace-Dateien
- $TSOS.TSOSCAT.#Pnn (nn=00..99): Katalog(e) der Privatplatten- und Banddateien
- $TSOS.TSOSCAT.#Jnn (nn=00..99): Katalog(e) der Jobvariablen
- $TSOS.TSOSCAT.$PFI: Index aller Katalogeinträge
- $TSOS.TSOSCAT.$NLO: Liste aller Dateinamen
- $TSOS.TSOSCAT.$NLC: Kopie der Dateinamensliste

Katalogeinträge aus den neu aufzunehmenden SF-Pubsets, die auf dem SM-Pubsets in Spezialkatalogen geführt werden (das sind alle außer die Dateien, die Speicherplatz im Pubset belegen), werden in diese bestehenden Spezialkataloge ausgelagert.

Die Dateien $TSOS.TSOSCAT werden in Volume-Set-spezifische Namen umbenannt.

Folgende Dateien werden auf den neu aufzunehmenden SF-Pubsets gelöscht:

- $TSOS.SYSCAT.GUARDS
- $TSOS.SYSEAM
- $TSOS.SYSPBN
- $TSOS.SYSSRPM
- $TSOS.SYSSRPM.BACKUP
- $TSOS.TSOSJOIN

Die Guard- und Benutzereinträge aus den Dateien $TSOS.SYSCAT.GUARDS und $TSOS.SYSSRPM werden in die entsprechende Datei auf dem Control-Volume-Set übertragen.

Die Cache-Konfiguration wird aus den MRSCAT-Einträgen der zu konvertierenden SF-Pubsets in die Pubset-Konfigurationsdatei des SM-Pubsets übernommen. Ausnahme: die system- bzw. pubset-bezogene Cache-SIZE-TOLERANCE, die - wenn erwünscht - für den neuen Pubset neu eingestellt werden muss.

Der im SVL des Control-Volume-Sets verankerte Shareability-Wert bleibt unverändert.

Die Systemkennungen (SYSIDs) in den SVLs bleiben i.A. erhalten.

14.5.2 Ausgaben

Fehler- und Protokollierungsmeldungen sowie die abschließende Erfolgsmeldung werden auf dasselbe Ausgabeziel gelenkt wie die Informationen über Inkonsistenzen, d.h. auf das bzw. die durch TEXT-OUTPUT spezifizierte(n) Ausgabeziel(e).
Im Dialog werden die Erfolgsmeldung und die Protokollierungsmeldungen zusätzlich auch auf SYSOUT ausgegeben, falls TEXT-OUTPUT=*SYSLST ist.

Werden die Informationen über Inkonsistenzen nicht in Text-Form ausgegeben (TEXT-OUTPUT=*NONE), so werden die Fehlermeldungen, die Meldung über das Starten der Funktion sowie die abschließende Erfolgsmeldung auf SYSOUT ausgegeben.

1. Im Dialog: Tabelle der zu Volume-Sets umzuwandelnden SF-Pubsets aus dem Operanden ADD-VOLUME-SET mit den spezifizierten logischen Eigenschaften. Diese Tabelle wird unabhängig vom TEXT-OUTPUT-Operanden auf SYSOUT ausgegeben.

   Zum Format siehe Maske 12 im Abschnitt "Ausgaben in Bildschirmmasken".
   Bei dieser Maske hat der Anwender die Möglichkeit, durch eine entsprechende Eingabe in der Fußzeile den Abbruch der Funktionsausführung zu veranlassen.
2. Tabelle der zu Volume-Sets umzuwandelnden SF-Pubsets aus dem Operanden ADD-VOLUME-SET mit Angabe von Plattenformat und spezifizierten logischen Eigenschaften.

   Zum Format siehe Maske 13 im Abschnitt "Ausgaben in Bildschirmmasken".
   Bei dieser Maske hat der Anwender die Möglichkeit, durch eine entsprechende Eingabe in der Fußzeile den Abbruch der Funktionsausführung zu veranlassen.
3. Protokollierungsmeldungen:
   - Meldung, dass die Funktion bzw. die funktionsausführende Task gestartet wurde
   - Fortschrittsprotokollierung: Meldung, dass die GUARDS-/SRPM-/CMS-Konvertierung gestartet wurde
   - Ergebnis des Konsistenz-Checks
     Die Ergebnismeldung wird unabhängig vom TEXT-OUTPUT-Operanden im Dialog immer auch auf SYSOUT ausgegeben.
   - In Sonderfällen bzw. bei fehlerhaftem Abbruch gehen evtl. weitere Meldungen der Ergebnismeldung voraus.

1. Im Fehlerfall:
   - Auflistung der unzulässigen Dateien, Namenskonflikte, zu langen Namen und Guard-Bedingungsfehler wie bei CHECK-NAME-CONSISTENCY
     (siehe Abschnitt „Ausgaben“).

     Zum Format siehe Masken 2 bis 11 im Abschnitt "Ausgaben in Bildschirmmasken".
   - Gibt es nach S1 migrierte Dateien in einem der Pubsets und ist nicht S1-MIGRATED-FILES=*ALLOWED spezifiziert, so wird dies als Fehler gemeldet. (Jeweils eine Meldung pro fehlerhaftem Pubset).

Ausgaben im Simulations-Modus

1. Ergebnismeldung der Syntaxprüfung
2. Tabelle der zu Volume-Sets umzuwandelnden SF-Pubsets aus dem Operanden ADD-VOLUME-SET mit den spezifizierten logischen Eigenschaften.
   Zum Format siehe Maske 12 im Abschnitt "Ausgaben in Bildschirmmasken".

14.6 Starten und Beenden von SMPGEN

Das Programm wird gestartet mit /START-SMPGEN

START-SMPGEN	Alias: **SMPGEN**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Mit der Anweisung END wird SMPGEN beendet.

Format

END

Die Anweisung besitzt keine Operanden.

14.7 Anweisungen

- Übersicht über die Anweisungen von SMPGEN
- Beschreibung der Anweisungen
  - CREATE-SYSTEM-MANAGED-PUBSET - Konvertieren eines SM-Pubsets aus SF-Pubsets
  - MODIFY-SYSTEM-MANAGED-PUBSET - Erweitern eines bestehenden SM-Pubsets

14.7.1 Übersicht über die Anweisungen von SMPGEN

Anweisung	Bedeutung
CREATE-SYSTEM-MANAGED-PUBSET	Einen oder mehrere SF-Pubsets zu einem SM-Pubset konvertieren. • Im Simulationsmodus: Überprüfen der Anweisung und Auflisten des zu erzeugenden Pubsets • Im Prüfungsmodus: Prüfen, ob die angegebenen SF-Pubsets konvertierbar sind und sich zu einem SM-Pubset zusammenfassen lassen
MODIFY-SYSTEM-MANAGED-PUBSET	Einen oder mehrere SF-Pubsets konvertieren und mit einem bereits bestehenden SM-Pubset zusammenfassen. • Im Simulationsmodus: Überprüfen der Anweisung und Auflisten des zu erzeugenden Pubsets • Im Prüfungsmodus: Prüfen, ob die angegebenen SF-Pubsets konvertierbar sind und sich mit einem bestehenden SM-Pubset zusammenfassen lassen
CHECK-NAME-CONSISTENCY [1)]	Prüfen, ob die angegebenen SF-Pubsets zu einem SM-Pubset konvertierbar sind.

[1)] Die Anweisung CHECK-NAME-CONSISTENCY wird nur noch aus Kompatibilitätsgründen unterstützt und ist im geführten Dialog nicht mehr sichtbar. Es stehen die erweiterten Funktionen der Anweisungen CREATE- und MODIFY-SYSTEM-MANAGED-PUBSET zur Verfügung.

Außerdem unterstützt SMPGEN die Ausführung der SDF-Standardanweisungen (siehe Handbuch „Dialogschnittstelle SDF“ [20]).

14.7.2 Beschreibung der Anweisungen

- CREATE-SYSTEM-MANAGED-PUBSET - Konvertieren eines SM-Pubsets aus SF-Pubsets
- MODIFY-SYSTEM-MANAGED-PUBSET - Erweitern eines bestehenden SM-Pubsets

14.7.2.1 CREATE-SYSTEM-MANAGED-PUBSET - Konvertieren eines SM-Pubsets aus SF-Pubsets

- für privilegierte Anwender

  Die Anweisung CREATE-SYSTEM-MANAGED-PUBSET erstellt entweder einen neuen SM-Pubset oder prüft, ob sich angegebene SF-Pubsets konfliktlos in einen neuen SM-Pubset überführen lassen.
  Beim Erstellen eines neuen SM-Pubsets werden max. 255 SF-Pubsets zu einem neuen SM-Pubset konvertiert, wobei jeder SF-Pubset zu einem Volume-Set des SM-Pubsets wird.
  Die Prüffunktion führt einen Konsistenz-Check durch, der über systemweite Namenskonflikte bei den betroffenen Pubsets informiert.

- für nicht-privilegierte Anwender

  Für nichtprivilegierte Anwender stellt die Anweisung CREATE-SYSTEM-MANAGED-PUBSET nur die Prüffunktion zur Verfügung. Dabei entscheidet ein Konsistenz-Check, ob sich die angegebenen SF-Pubsets in einen neuen SM-Pubset überführen lassen. Die Überprüfungen beziehen sich ausschließlich auf die Benutzerkennung des Aufrufers. Der Aufrufer kann sich somit darüber informieren, welche von ihm verursachten Namenskonflikte es gibt und wie er sie beseitigen kann.

Ausgaben

Die Ausgaben dieser Anweisung werden im Abschnitt „Ausgaben (Erzeugen neuer SM-Pubsets)“ bzw. im Abschnitt „Ausgaben (Prüffunktion)“ beschrieben, die Masken im Abschnitt „Ausgaben in Bildschirmmasken“.

Arbeitsweise

Die generelle Arbeitsweise dieser Anweisung, Einschränkungen sowie die Kontingentverteilung werden im Abschnitt „Erzeugen neuer SM-Pubsets“ beschrieben.

ACHTUNG!

- CREATE-SYSTEM-MANAGED-PUBSET OPERATIONAL-MODE=*OPERATION sollte nie ohne vorhergehende FDDRL-Sicherung der umzuwandelnden Pubsets aufgerufen werden, da bei einem fehlerhaften Abbruch alle Pubsets zerstört und nicht mehr zugreifbar sein können.
- Eine logische Sicherung mittels HSMS wird ebenfalls empfohlen. Hierbei kann ein neues Backup-Archiv erzeugt werden, das zur Herstellung der HSMS-Umgebung für den neuen SM-Pubset verwendet werden kann. Anleitungen zur Umstellung der HSMS-Umgebung sind im Leitfaden für die SMS-Migration gegeben, siehe Handbuch „System Managed Storage“ [8].
- Spool-Aufträge für Dateien, die in den zu konvertierenden Pubsets katalogisiert sind, sollten nicht mehr ausstehen, da sie durch die Umbenennung der Katalogkennung nicht mehr ausgeführt werden können.

Simulations- und Check-Modus

Wenn der Operand OPERATIONAL-MODE=*SIMULATION angegeben wird, wird die hier beschriebene Funktion nicht real ausgeführt. Es erfolgt lediglich ein Syntax-Check und eine Auflistung der angegebenen Volume-Sets.

Wenn der Operand OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY angegeben wird, wird nur die Prüffunktion ausgeführt.

Nähere Informationen siehe Operandenbeschreibung.

Format für privilegierte Anwender

CREATE-SYSTEM-MANAGED-PUBSET

```
PUBSET=<catid>
,CONTROL-VOLUME-SET=<catid>(...)
    <catid>(..)
        |   AVAILABILITY=*STD/*HIGH
        |   ,PERFORMANCE-ATTR=*STD/*PARAMETERS(...)
        |       *PARAMETERS(...)
        |           |   PERFORMANCE=list-poss(3):*STD/*HIGH/ *VERY-HIGH
        |           |   ,WRITE-CONSISTENCY=*BY-CLOSE/*IMMEDIATE
,ADD-VOLUME-SET=*NONE/list-poss(254):<catid>(...)
    <catid>(...)
        |   AVAILABILITY=*STD/*HIGH
        |   ,USAGE = *STD / *HSMS-CONTROLLED
        |   ,PERFORMANCE-ATTR=*STD/*PARAMETERS(...)
        |       *PARAMETERS(...)
        |           |   PERFORMANCE=list-poss(3):*STD/*HIGH/ *VERY-HIGH
        |           |   ,WRITE-CONSISTENCY=*BY-CLOSE/*IMMEDIATE
,KEEP-USER-ATTRIBUTES=*ALL(...)/*BASIC
    *ALL(...)
        |   ATTRIBUTE-ORIGIN=*CONTROL-VOLUME-SET/<catid>
,S1-MIGRATED-FILES=*NOT-ALLOWED/*ALLOWED
,MAX-ERRORS=32767/<integer0..32767>
,MONJV=*NONE/ <filename 1..54 without-gen-vers>
,ERROR-INFORMATION=*DETAIL/*SUMMARY
,EXECUTION-MODE=*SYNCHRONOUS(...)/*ASYNCHRONOUS(...)
    *SYNCHRONOUS(...)
        |   TEXT-OUTPUT=*STD/*SYSOUT/*SYSLST/*NONE
        |   ,STRUCTURE-OUTPUT=*SYSINF/*NONE/<composed-name1..100>
```

*ASYNCHRONOUS(...)

| TEXT-OUTPUT=*SYSLST/<filename 1..54 without-gen-vers>

```
,OPERATIONAL-MODE=*OPERATION / *SIMULATION / *CHECK-NAME-CONSISTENCY(...)

    *CHECK-NAME-CONSISTENCY(...)

        |    PUBSET-STATE=*IMPORTED/*NOT-IMPORTED

        |    ,USER-ID=*OWN/*ALL
```

Operanden

PUBSET = <catid>
Kennung des zu erzeugenden oder auf Erzeugbarkeit zu überprüfenden SM-Pubsets.

Die Kennung muss sich von den Kennungen der zu konvertierenden SF-Pubsets unterscheiden.

Falls nicht nur eine Prüfung durchgeführt werden soll (Operand OPERATIONAL-MODE= *CHECK-NAME-CONSISTENCY), sondern auch eine Konvertierung, darf noch kein gleich lautender MRSCAT-Eintrag existieren. Die angegebene Kennung geht ein in die Überprüfung der Dateinamenslänge und der automatischen Modifizierbarkeit von GUARD-Bedingungen.

CONTROL-VOLUME-SET = <catid>(...)
Kennung des SF-Pubsets, der zum Control-Volume-Set des neuen SM-Pubsets werden soll.

Sie muss sich von der Kennung des zu erzeugenden SM-Pubsets unterscheiden.
Wie für alle zu konvertierenden SF-Pubsets, muss auch für diesen Pubset bereits ein MRSCAT-Eintrag mit dem richtigen Gerätetyp existieren.

Falls nicht nur eine Prüfung durchgeführt werden soll (Operand OPERATIONAL-MODE= *CHECK-NAME-CONSISTENCY), sondern auch eine Konvertierung, darf der Pubset jedoch weder im lokalen noch in einem anderen System importiert sein.
Der Control-Volume-Set muss die im Abschnitt „Voraussetzungen für den Ablauf von SMPGEN“ beschriebenen Anforderungen erfüllen.

Bei der reinen Prüffunktion (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY) wird die Angabe der folgenden Volume-Set-Eigenschaften ignoriert.

AVAILABILITY =
Bestimmt den Verfügbarkeitsgrad von Dateien, bezogen auf die Ausfallsicherheit des Volume-Sets.
Die Verfügbarkeit ist eine logische Volume-Set-Eigenschaft, die bei der Speicherplatzwahl als Auswahlkriterium zur Ablage von Dateien dient. Eine Änderung der Volume-Set-Eigenschaft muss der Systembetreuer explizit veranlassen.

AVAILABILITY = *STD
Voreinstellung: Der Volume-Set bietet keine erhöhte Ausfallsicherheit.

AVAILABILITY = *HIGH
Der Volume-Set bietet hohe Ausfallsicherheit.
Es liegt in der Verantwortung des Systembetreuers, zu überprüfen, dass die Ausfallsicherheit auch physikalisch gewährleistet ist.

PERFORMANCE-ATTR =
Bestimmt die Performance-Eigenschaften des Volume-Sets, bezogen auf Dateizugriffe. Es handelt sich hierbei

um logische Attribute, die bei der Speicherplatzwahl als Auswahlkriterium dienen. Es erfolgt keine Validierung bzgl. der vorhandenen Hardware (z.B. Cache).

PERFORMANCE-ATTR = *STD
Voreinstellung: Für den Volume-Set sollen keine erhöhten Performance-Attribute bzgl. Dateizugriffen definiert werden.

PERFORMANCE-ATTR = *PARAMETERS(...)
Die für den Volume-Set geltenden Performance-Eigenschaften werden über folgende Unterstruktur definiert:

PERFORMANCE =
Beschreibt das I/O-Performance-Profil des Volume-Sets. Die Angabe einer Liste von bis zu drei Werten ist möglich.

PERFORMANCE = *STD
Voreinstellung: Der Volume-Set bietet keine erhöhte I/O-Performance.

PERFORMANCE = *HIGH
Der Volume-Set bietet erhöhte I/O-Performance.

PERFORMANCE = *VERY-HIGH
Der Volume-Set bietet sehr hohe I/O-Performance.

WRITE-CONSISTENCY =
Bestimmt, ob die zur Erhöhung der I/O-Performance eingesetzte Hardware auch im Falle eines Systemabsturzes Datenkonsistenz garantiert.

WRITE-CONSISTENCY = *BY-CLOSE
Voreinstellung: Schreib-I/Os mit erhöhter Performance sind nur unter Verzicht auf sofortige absturzsichere Datenablage möglich; erst beim Schließen der Datei gelangen die Daten auf ein Medium, das Datenerhaltung auch im Falle eines Systemabsturzes garantiert.

WRITE-CONSISTENCY = *IMMEDIATE
Auch bei Schreib-I/Os mit erhöhter Performance gelangen die Daten unmittelbar auf ein Medium, das Datenerhaltung auch im Falle eines Systemabsturzes garantiert.

ADD-VOLUME-SET =
Spezifiziert, ob außer dem als Control-Volume-Set angegebenen Pubset weitere SF-Pubsets in den SM-Pubset aufgenommen werden sollen.

ADD-VOLUME-SET = *NONE
Voreinstellung: Es sollen keine weiteren SF-Pubsets aufgenommen werden.

ADD-VOLUME-SET = <catid>(...)
Kennungen der weiteren SF-Pubsets, die als Volume-Set in den SM-Pubset aufgenommen werden sollen. Die Pubset-Kennung wird zur Volume-Set-Kennung.

Alle Volume-Set-Kennungen müssen sich untereinander, von der Kennung des Control-Volume-Sets und von der gewählten SM-Pubset-Kennung unterscheiden.
Falls nicht nur eine Prüfung durchgeführt werden soll (Operand OPERATIONAL-MODE= *CHECK-NAME-CONSISTENCY), sondern auch eine Konvertierung, dürfen die SF-Pubsets jedoch weder im lokalen noch in einem anderen System importiert sein; es müssen dafür jedoch bereits MRSCAT-Einträge mit dem richtigen Gerätetyp existieren.Die Volume-Sets müssen die im Abschnitt „Voraussetzungen für den Ablauf von SMPGEN“ beschriebenen Anforderungen erfüllen.

Die als Unterstruktur angebbaren Volume-Set-Eigenschaften sind bis auf USAGE unter dem Operanden CONTROL-VOLUME-SET beschrieben.
Bei der reinen Prüffunktion (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY) wird die Angabe der folgenden Volume-Set-Eigenschaften ignoriert.

USAGE =
Bestimmt die Art der Nutzung des Volume-Sets.

USAGE = *STD
Der Volume-Set soll als Ablageort für die Standarddateien genutzt werden. Arbeitsdateien (Dateiattribut WORK) dürfen dort nicht abgelegt werden.

USAGE = *HSMS-CONTROLLED
Der Volume-Set dient dem Subsystem HSMS zur Realisierung der Speicherebene S1 und des Backup-Volume-Sets eines SM-Pubsets. Benutzer dürfen dort keine Dateien ablegen.

KEEP-USER-ATTRIBUTES =
Spezifiziert, wie mit den Logon-Kennwörtern und der Benutzergruppenstruktur sowie den nur Home-Pubset-relevanten Benutzerinformationen (Privilegien, Berechtigungen, Operator-Roles) verfahren werden soll.

Bei der reinen Prüffunktion (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY) wird die Angabe dieses Operanden ignoriert.

KEEP-USER-ATTRIBUTES = *ALL(...)
Voreinstellung: Die Informationen sollen von einem der zusammenzuführenden SF-Pubsets übernommen werden. Die Informationen auf allen anderen Ausgangs-Pubsets bleiben unberücksichtigt.

ATTRIBUTE-ORIGIN = *CONTROL-VOLUME-SET
Die Informationen sollen von dem Pubset übernommen werden, der Control-Volume-Set werden soll.

ATTRIBUTE-ORIGIN = <catid>
Spezifiziert die Kennung des Pubsets, von dem die Informationen übernommen werden sollen. Er muss einer der zu konvertierenden Pubsets sein.

KEEP-USER-ATTRIBUTES = *BASIC
Nur die hauptsächlich Daten-Pubset-relevanten Benutzerinformationen sollen übernommen werden. Die Informationen gehen bei der Pubset-Konvertierung verloren (auch dann, wenn nur ein Pubset konvertiert wird.)

S1-MIGRATED-FILES =
Vereinbart, ob es als Konflikt anzusehen ist, wenn in einem der zu überprüfenden Pubsets Dateien katalogisiert sind, die gegenwärtig auf die S1-Ebene migriert sind.

S1-MIGRATED-FILES = *NOT-ALLOWED
Voreinstellung: Der oben genannte Fall ist als Konflikt anzusehen.

S1-MIGRATED-FILES = *ALLOWED
Der oben genannte Fall ist nicht als Konflikt anzusehen.
Die S1-migrierten Dateien sind im Anschluss an die Konvertierung jedoch nicht mehr zugänglich, sondern müssen aus einer Backup-Datei zurückgeholt werden. Ihre Katalogeinträge sind noch erhalten, sodass sie nach Erzeugung der HSMS-Umgebung durch REPAIR-CATALOG-BY-RESTORE gezielt auf die S1-Ebene des SM-Pubsets gebracht werden können.

MAX-ERRORS = 32767 / <integer 0..32767>
Bei Auftreten der angegebenen Anzahl von Fehlern wird der Konsistenz-Check abgebrochen. Die vor dem Abbruch aufgetretenen Konflikte werden ausgegeben, falls ERROR-INFORMATION = *DETAIL angegeben wurde. 32767 ist Voreinstellung.

MONJV =
Vereinbart eine überwachende Jobvariable.

MONJV = *NONE
Voreinstellung: Es gibt keine überwachende Jobvariable.

MONJV = <filename 1..54 without-gen-vers>
Gibt den Namen der überwachenden Jobvariablen an. Sie muss bereits in einem zugreifbaren Pubset katalogisiert sein und darf nicht durch ein Kennwort geschützt sein.

Die Zustandsanzeige der Jobvariablen wird während der Konvertierung (OPERATIONAL-MODE=*OPERATION) auf folgende Weise gesetzt:

$I zu Beginn der Bearbeitung

$R nach erfolgreicher Durchführung der Bearbeitung

$A bei fehlerhaftem Abbruch der Bearbeitung

Die Zustandsanzeige der Jobvariablen wird während des Konsistenz-Checks (OPERATIONAL-MODE=*CHECK-NAME- CONSISTENCY) auf folgende Weise gesetzt:

$I zu Beginn der Konsistenz-Checks

$R am Ende des Konsistenz-Checks, wenn Konsistenz besteht

$E am Ende des Konsistenz-Checks, wenn keine Konsistenz besteht

$A bei nicht vollständig durchgeführtem Konsistenz-Check

Die Angabe des Operanden ist nur sinnvoll bei Einsatz des Software-Produkts JV.

Kann die Jobvariable nicht versorgt werden (z.B. weil ein Kennwort gesetzt ist), wird der Konsistenz-Check trotzdem durchgeführt.
Wenn die Jobvariable auch zu Beginn der Funktionsausführung einen definierten Inhalt haben soll, muss der Anwender sie selbst initialisieren. Wird die Funktionsausführung gleich zu Anfang abgebrochen, z.B. weil ein semantischer Fehler festgestellt wird, so versorgt SMPGEN die Jobvariable nicht.

Liegt die Jobvariable auf einem der zusammenzuführenden Pubsets und ist nicht OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY(PUBSET-STATE= *IMPORTED) spezifiziert worden, so wird der SMPGEN-Auftrag zurückgewiesen.

ERROR-INFORMATION =
Steuert den Umfang der Fehlerinformationen, die in Text- und/oder Variablenform ausgegeben werden.

ERROR-INFORMATION = *DETAIL
Voreinstellung: Eine detaillierte Fehlerinformation soll ausgegeben werden. Dabei wird nicht nur die Anzahl der Namenskonflikte ausgegeben, sondern es werden auch alle mehrfach vorkommenden oder zu langen Namen einzeln aufgelistet.

Im Dialog an der Datensichtstation werden Bildschirmmasken ausgegeben, die die Möglichkeit bieten, die Fortsetzungsbildschirme oder die Detail-Bildschirme zu unterdrücken oder auf SYSLST umzuschalten (siehe Abschnitt „Ausgaben in Bildschirmmasken“).
Im Batch-Betrieb und auf SYSLST erfolgt die Ausgabe im selben Format, nur ohne Reaktionsmöglichkeit.

ERROR-INFORMATION = *SUMMARY
Es sollen keine detaillierten Fehlerinformationen ausgegeben werden, d.h. keine Listen von Namenskonflikten, sondern nur die Anzahl der Konflikte in den verschiedenen Namensklassen.

EXECUTION-MODE =
Gibt an, ob die Funktion in der aktuellen Task oder in einer neu zu erzeugenden Batch-Task ausgeführt werden soll.

EXECUTION-MODE = *SYNCHRONOUS(...)
Voreinstellung: Die Funktion soll in der laufenden Task ausgeführt werden.

TEXT-OUTPUT =
Steuert das Ausgabeziel der in Textformat erfolgenden Ausgaben, d.h. der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *STD
Voreinstellung: Die Ausgabe soll entsprechend der standardmäßigen Voreinstellung erfolgen. Im Dialogbetrieb erfolgt die Ausgabe sowohl auf SYSOUT als auch auf SYSLST. Im Batch-Betrieb erfolgt die Ausgabe auf SYSLST.

TEXT-OUTPUT = *SYSOUT
Die Ausgabe soll auf SYSOUT erfolgen.
Diese Angabe ist nur im Dialogbetrieb erlaubt. SYSOUT darf nicht auf eine Datei umgewiesen worden sein.
Ist SYSOUT-Ausgabe gewünscht, so werden Bildschirmmasken ausgegeben, die mit der DUE-Taste zu quittieren sind. Beim Auftreten vieler Fehler ist es möglich, das detaillierte Fehlerprotokoll nachträglich auf SYSLST zu legen oder zu unterdrücken.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll auf SYSLST erfolgen.

TEXT-OUTPUT = *NONE
Es wird keine Ausgabe in Listen- oder Meldungsform gewünscht. Diese Angabe ist nur sinnvoll, wenn stattdessen der Operand STRUCTURE-OUTPUT angegeben wird. Fehlermeldungen und die Meldungen über Start und Erfolg der Funktion sind nicht unterdrückbar; sie werden auch bei der Angabe TEXT-OUTPUT=*NONE auf SYSOUT ausgegeben.

STRUCTURE-OUTPUT =
Steuert die Ausgabe von Fehlerinformationen in S-Variablen.

STRUCTURE-OUTPUT = *SYSINF
Voreinstellung: Die Fehlerinformationen sollen in die S-Variable ausgegeben werden, auf die der Anwender den Variablenstrom SYSINF zugewiesen hat.
Ist dieser Variablenstrom nicht aktiv, so erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = *NONE
Es erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = <composed-name 1..100>
Name der S-Variablen, in der die Fehlerinformationen abgelegt werden.
Die Variable muss vom Anwender als dynamisch definierte Struktur deklariert sein (`/DECLARE-VARIABLE ..., MULTIPLE-ELEMENTS=*LIST,TYPE=*STRUCTURE`).

EXECUTION-MODE = *ASYNCHRONOUS(...)
Die Funktion soll in einer neu zu erzeugenden Batch-Task ausgeführt werden. In diesem Fall erfolgt keine Informationsausgabe in eine S-Variable.

TEXT-OUTPUT =
steuert die Ausgabe der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll in eine Spoolout-Datei erfolgen, die bei Jobende ausgedruckt und gelöscht wird. Der Name der Spoolout-Datei entspricht der Standardeinstellung der SYSLST-Datei für ENTER-Tasks.

TEXT-OUTPUT = <filename 1..54 without-gen-vers>
Die Ausgabe soll in die angegebene Datei erfolgen. Falls die Datei bereits existiert, muss der Anwender das Schreibzugriffsrecht darauf besitzen. Die Datei darf nicht durch ein Kennwort geschützt sein. Sie muss lokal zugreifbar und darf keine temporäre Datei sein. Bei Jobende wird die Datei nicht ausgedruckt und auch nicht gelöscht.

OPERATIONAL-MODE =
Bestimmt, ob die Funktion „Erzeugen eines SM-Pubsets“ real ausgeführt oder nur simuliert oder ob nur die Prüffunktion (Konsistenz-Check) ausgeführt werden soll.

OPERATIONAL-MODE = *OPERATION
Voreinstellung: Die Funktion soll real ausgeführt werden.

OPERATIONAL-MODE = *SIMULATION
Die Funktion soll nur simuliert werden.
Die Syntax der angegebenen Operanden wird geprüft.
Die Operanden MAX-ERRORS, MONJV, ERROR-INFORMATION und EXECUTION-MODE (mit Unteroperanden) haben jedoch nicht die in der Operandenbeschreibung aufgeführte Bedeutung; Ausgabemedium, ausführende Task usw. sind im Simulationsmodus nicht einstellbar.
Die Ausführung erfolgt immer in der laufenden Task: falls dies eine Dialog-Task ist, erfolgt die Ausgabe auf SYSOUT und SYSLST, falls es eine Batch-Task ist, nur auf SYSLST.

Bei erfolgreicher Syntaxprüfung wird eine Tabelle der Volume-Sets mit den spezifizierten logischen Eigenschaften ausgegeben.

Die Ausgangs-Pubsets müssen für diese Funktion nicht zugänglich sein; es erfolgen keine Plattenoperationen. Demgemäß werden auch keine Überprüfungen vorgenommen, für die die Platten verfügbar sein müssen, also kein Namens-Konsistenz-Check. Die MRSCAT-Einträge werden ebenfalls nicht überprüft.

OPERATIONAL-MODE = *CHECK-NAME-CONSISTENCY(...)
Es wird die Prüffunktion (Konsistenz-Check) von SMPGEN ausgeführt.
Dabei werden die SF-Pubsets, die in den Operanden CONTROL-VOLUME-SET und ADD-VOLUME-SET spezifiziert wurden, daraufhin überprüft, ob sie in einen neu zu schaffenden SM-Pubset übergeführt bzw. konvertiert werden können.

PUBSET-STATE =
Gibt an, ob der Konsistenz-Check bei laufender Pubset-Session, also möglicherweise parallel zu anderen Anwender-Zugriffen auf die Pubsets, erfolgen soll.

PUBSET-STATE = *IMPORTED
Voreinstelling: Der Konsistenz-Check soll bei laufender Pubset-Session erfolgen. Die Pubsets sind bereits lokal importiert. Der Konsistenz-Check liefert nur eine ungefähre Aussage, die durch zwischenzeitliche Änderungen bereits überholt sein kann.

PUBSET-STATE = *NOT-IMPORTED
SMPGEN soll die Pubsets für den Konsistenz-Check exklusiv belegen. Die Pubsets sind weder lokal noch von einem anderen System importiert. Während des Konsistenz-Checks werden sie gegen andere Zugriffe gesperrt, daher ist das Ergebnis zum Zeitpunkt der Ausgabe korrekt.

USER-ID =
Spezifiziert, für welche Benutzerkennung(en) die Konsistenzprüfungen der Pubsets erfolgen sollen.

USER-ID = *OWN
Voreinstellung: Die Konsistenzprüfungen sollen sich nur auf Dateinamen, JVs und Guards unter der Aufrufer-Benutzerkennung erstrecken.
Die in der Anweisungsbeschreibung genannten Systemverwaltungs-spezifischen allgemeinen Prüfungen (unerlaubte Dateien, S1-Migration) erfolgen auch bei dieser Angabe.

USER-ID = *ALL
Die Konsistenzprüfungen sollen sich auf alle Benutzerkennungen erstrecken. Diese Angabe ist unzulässig, falls einer oder mehrere der Pubsets im SHARED-SLAVE-Modus importiert ist.

Format für nicht-privilegierte Anwender

```
CREATE-SYSTEM-MANAGED-PUBSET

 PUBSET=<catid>

 ,CONTROL-VOLUME-SET=<catid>

 ,ADD-VOLUME-SET=*NONE/list-poss(254):<catid>

 ,MAX-ERRORS=32767/<integer0..32767>

 ,MONJV=*NONE/ <filename 1..54 without-gen-vers>

 ,ERROR-INFORMATION=*DETAIL/*SUMMARY

 ,EXECUTION-MODE=*SYNCHRONOUS(...)/*ASYNCHRONOUS(...)

     *SYNCHRONOUS(...)

         |   TEXT-OUTPUT=*STD/*SYSOUT/*SYSLST/*NONE

         |   ,STRUCTURE-OUTPUT=*SYSINF/*NONE/<composed-name1..100>

     *ASYNCHRONOUS(...)

         |   TEXT-OUTPUT=*SYSLST/<filename 1..54 without-gen-vers>
```

Operanden

PUBSET = <catid>
Kennung eines SM-Pubsets, der auf Erzeugbarkeit überprüft werden soll.
Die Kennung muss sich von den Kennungen der zusammenzuführenden SF-Pubsets unterscheiden.
Es darf ein gleichlautender MRSCAT-Eintrag existieren: Auf einen entsprechenden Pubset wird - etwa zum Zwecke von Überprüfungen - nicht zugegriffen.
Die angegebene Kennung geht ein in die Überprüfung der Dateinamenslänge und der automatischen Modifizierbarkeit von GUARD-Bedingungen.

CONTROL-VOLUME-SET = <catid>
Kennung des SF-Pubsets, der zum Control-Volume-Set des neuen SM-Pubsets werden soll. Die Kennung wird zur Volume-Set-Kennung des Control-Volume-Sets. Sie muss sich von der Kennung des zu erzeugenden SM-Pubsets unterscheiden. Der SF-Pubset muss im lokalen System importiert sein.

ADD-VOLUME-SET =
Spezifiziert, ob außer dem als Control-Volume-Set angegebenen Pubset weitere SF-Pubsets für eine Aufnahme in den SM-Pubset überprüft werden sollen.

ADD-VOLUME-SET = *NONE
Voreinstellung: Es sollen keine weiteren SF-Pubsets überprüft werden.

ADD-VOLUME-SET = <catid>
Kennungen der weiteren SF-Pubsets, die überprüft werden sollen.
Die Pubset-Kennung wird zur Volume-Set-Kennung.

Alle Volume-Set-Kennungen müssen sich untereinander, von der Kennung des Control-Volume-Sets und von der gewählten SM-Pubset-Kennung unterscheiden.
Die SF-Pubsets müssen im lokalen System importiert sein.

Die als Unterstruktur angebbaren Volume-Set-Eigenschaften sind bis auf USAGE unter dem Operanden CONTROL-VOLUME-SET beschrieben.

MAX-ERRORS = 32767 / <integer 0..32767>
Bei Auftreten der angegebenen Anzahl von Fehlern wird der Konsistenz-Check abgebrochen. Die vor dem Abbruch aufgetretenen Konflikte werden ausgegeben, falls ERROR-INFORMATION = *DETAIL angegeben wurde. 32767 ist Voreinstellung.

MONJV =
Vereinbart eine überwachende Jobvariable.

MONJV = *NONE
Voreinstellung: Es gibt keine überwachende Jobvariable.

MONJV = <filename 1..54 without-gen-vers>
Gibt den Namen der überwachenden Jobvariablen an. Sie muss bereits in einem zugreifbaren Pubset katalogisiert sein und darf nicht durch ein Kennwort geschützt sein.

Die Zustandsanzeige der Jobvariablen wird während des Konsistenz-Checks auf folgende Weise gesetzt:

$I	zu Beginn der Konsistenz-Checks
$R	am Ende des Konsistenz-Checks, wenn Konsistenz besteht
$E	am Ende des Konsistenz-Checks, wenn keine Konsistenz besteht
$A	bei nicht vollständig durchgeführtem Konsistenz-Check

Die Angabe des Operanden ist nur sinnvoll bei Einsatz des Software-Produkts JV.

> Kann die Jobvariable nicht versorgt werden (z.B. weil ein Kennwort gesetzt ist), wird der Konsistenz-Check trotzdem durchgeführt.
> Wenn die Jobvariable auch zu Beginn der Funktionsausführung einen definierten Inhalt haben soll, muss der Anwender sie selbst initialisieren. Wird die Funktionsausführung gleich zu Anfang abgebrochen, z.B. weil ein semantischer Fehler festgestellt wird, so versorgt SMPGEN die Jobvariable nicht.

ERROR-INFORMATION =
Steuert den Umfang der Fehlerinformationen, die in Text- und/oder Variablenform ausgegeben werden.

ERROR-INFORMATION = *DETAIL
Voreinstellung: Eine detaillierte Fehlerinformation soll ausgegeben werden. Dabei wird nicht nur die Anzahl der Namenskonflikte ausgegeben, sondern es werden auch alle mehrfach vorkommenden oder zu langen Namen einzeln aufgelistet.
Im Dialog an der Datensichtstation werden Bildschirmmasken ausgegeben, die die Möglichkeit bieten, die Fortsetzungsbildschirme oder die Detail-Bildschirme zu unterdrücken oder auf SYSLST umzuschalten (siehe Abschnitt „Ausgaben in Bildschirmmasken“).
Im Batch-Betrieb und auf SYSLST erfolgt die Ausgabe im selben Format, nur ohne Reaktionsmöglichkeit.

ERROR-INFORMATION = *SUMMARY
Es sollen keine detaillierten Fehlerinformationen ausgegeben werden, d.h. keine Listen von Namenskonflikten, sondern nur die Anzahl der Konflikte in den verschiedenen Namensklassen.

EXECUTION-MODE =
Gibt an, ob die Funktion in der aktuellen Task oder in einer neu zu erzeugenden Batch-Task ausgeführt werden soll.

EXECUTION-MODE = *SYNCHRONOUS(...)
Voreinstellung: Die Funktion soll in der laufenden Task ausgeführt werden.

TEXT-OUTPUT =
Steuert das Ausgabeziel der in Textformat erfolgenden Ausgaben, d.h. der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *STD
Voreinstellung: Die Ausgabe soll entsprechend der standardmäßigen Voreinstellung erfolgen. Im Dialogbetrieb erfolgt die Ausgabe sowohl auf SYSOUT als auch auf SYSLST. Im Batch-Betrieb erfolgt die Ausgabe auf SYSLST.

TEXT-OUTPUT = *SYSOUT
Die Ausgabe soll auf SYSOUT erfolgen.
Diese Angabe ist nur im Dialogbetrieb erlaubt. SYSOUT darf nicht auf eine Datei umgewiesen worden sein. Ist SYSOUT-Ausgabe gewünscht, so werden Bildschirmmasken ausgegeben, die mit der DUE-Taste zu quittieren sind. Beim Auftreten vieler Fehler ist es möglich, das detaillierte Fehlerprotokoll nachträglich auf SYSLST zu legen oder zu unterdrücken.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll auf SYSLST erfolgen.

TEXT-OUTPUT = *NONE
Es wird keine Ausgabe in Listen- oder Meldungsform gewünscht. Diese Angabe ist nur sinnvoll, wenn stattdessen der Operand STRUCTURE-OUTPUT angegeben wird.

Fehlermeldungen und die Meldungen über Start und Erfolg der Funktion sind nicht unterdrückbar; sie werden auch bei der Angabe TEXT-OUTPUT=*NONE auf SYSOUT ausgegeben.

STRUCTURE-OUTPUT =
Steuert die Ausgabe von Fehlerinformationen in S-Variablen.

STRUCTURE-OUTPUT = *SYSINF
Voreinstellung: Die Fehlerinformationen sollen in die S-Variable ausgegeben werden, auf die der Anwender den Variablenstrom SYSINF zugewiesen hat.
Ist dieser Variablenstrom nicht aktiv, so erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = *NONE
Es erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = <composed-name 1..100>
Name der S-Variablen, in der die Fehlerinformationen abgelegt werden.
Die Variable muss vom Anwender als dynamisch definierte Struktur deklariert sein (`/DECLARE-VARIABLE ..., MULTIPLE-ELEMENTS=*LIST,TYPE=*STRUCTURE`).

EXECUTION-MODE = *ASYNCHRONOUS(...)
Die Funktion soll in einer neu zu erzeugenden Batch-Task ausgeführt werden. In diesem Fall wird die Informationsausgabe in eine S-Variable nicht unterstützt.

TEXT-OUTPUT =
steuert die Ausgabe der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll in eine Spoolout-Datei erfolgen, die bei Jobende ausgedruckt und gelöscht wird. Der Name der Spoolout-Datei entspricht der Standardeinstellung der SYSLST-Datei für ENTER-Tasks.

TEXT-OUTPUT = <filename 1..54 without-gen-vers>
Die Ausgabe soll in die angegebene Datei erfolgen. Falls die Datei bereits existiert, muss der Anwender das Schreibzugriffsrecht darauf besitzen. Die Datei darf nicht durch ein Kennwort geschützt sein. Sie muss lokal zugreifbar und darf keine temporäre Datei sein. Bei Jobende wird die Datei nicht ausgedruckt und auch nicht gelöscht.

14.7.2.2 MODIFY-SYSTEM-MANAGED-PUBSET - Erweitern eines bestehenden SM-Pubsets

- für privilegierte Anwender

  Die Anweisung MODIFY-SYSTEM-MANAGED-PUBSET erweitert entweder einen bestehenden SM-Pubset oder prüft, ob sich angegebene SF-Pubsets konfliktlos in einen bestehenden SM-Pubset überführen lassen.

  Die Erweiterung eines SM-Pubsets ist nach fehlerfreier Ausführung der MODIFY-SYSTEM-MANAGED-PUBSET-Anweisung noch nicht vollständig abgeschlossen, sondern erst nach einem auf den SMPGEN-Lauf folgenden normalen Import des SM-Pubsets. Daraus ergibt sich auch, dass eine erneute Erweiterung nur nach einem zwischenzeitlich erfolgten Import des SM-Pubsets möglich ist.
- für nicht-privilegierte Anwender

  Für nichtprivilegierte Anwender stellt die Anweisung MODIFY-SYSTEM-MANAGED-PUBSET nur die Prüffunktion zur Verfügung. Dabei entscheidet ein Konsistenz-Check, ob sich die angegebenen SF-Pubsets in einen bereits bestehenden SM-Pubset überführen lassen. Die Überprüfungen beziehen sich ausschließlich auf die Benutzerkennung des Aufrufers. Der Aufrufer kann sich somit darüber informieren, welche von ihm verursachten Namenskonflikte es gibt und wie er sie beseitigen kann.

Ausgaben

Die Ausgaben dieser Anweisung werden im Abschnitt „Ausgaben (Erzeugen neuer SM-Pubsets)“ bzw. im Abschnitt „Ausgaben (Prüffunktion)“ beschrieben, die Masken im Abschnitt „Ausgaben in Bildschirmmasken“.

Arbeitsweise

Die generelle Arbeitsweise dieser Anweisung wird im Abschnitt „Erweitern einesbestehenden SM-Pubsets“ beschrieben.

> **ACHTUNG**!
>
> - MODIFY-SYSTEM-MANAGED-PUBSET OPERATIONAL-MODE= *OPERATION sollte nie ohne vorhergehende FDDRL-Sicherung der umzuwandelnden SF-Pubsets aufgerufen werden, da bei einem fehlerhaften Abbruch alle SF-Pubsets zerstört und nicht mehr zugreifbar sein können.
> - Eine logische Sicherung mittels HSMS wird ebenfalls empfohlen. Hierbei kann ein neues Backup-Archiv erzeugt werden, das zur Herstellung der HSMS-Umgebung für den erweiterten SM-Pubset verwendet werden kann. Anleitungen zur Umstellung der HSMS-Umgebung sind im Leitfaden für die SMS-Migration gegeben, siehe Handbuch „System Managed Storage“ [8].
> - Spool-Aufträge für Dateien, die in den zu konvertierenden SF-Pubsets katalogisiert sind, sollten nicht mehr ausstehen, da sie durch die Umbenennung der Katalogkennung nicht mehr ausgeführt werden können.

Simulations- und Check-Modus

Wenn der Operand OPERATIONAL-MODE=*SIMULATION angegeben wird, wird die hier beschriebene Funktion nicht real ausgeführt. Es erfolgt lediglich ein Syntax-Check und eine Auflistung der angegebenen Volume-Sets.

Wenn der Operand OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY angegeben wird, wird nur die Prüffunktion ausgeführt.

Nähere Informationen siehe Operandenbeschreibung.

Format für privilegierte Anwender

```
MODIFY-SYSTEM-MANAGED-PUBSET

 PUBSET=<catid>

 ,ADD-VOLUME-SET=list-poss(254):<catid>(...)

      <catid>(...)

           |    AVAILABILITY=*STD/*HIGH

           |    ,USAGE = *STD / *HSMS-CONTROLLED

           |    ,PERFORMANCE-ATTR=*STD/*PARAMETERS(...)

           |         *PARAMETERS(...)

           |              |    PERFORMANCE=list-poss(3):*STD/*HIGH/ *VERY-HIGH

           |              |    ,WRITE-CONSISTENCY=*BY-CLOSE/*IMMEDIATE

 ,S1-MIGRATED-FILES=*NOT-ALLOWED/*ALLOWED

 ,MAX-ERRORS=32767/<integer0..32767>

 ,MONJV=*NONE/ <filename 1..54 without-gen-vers>

 ,ERROR-INFORMATION=*DETAIL/*SUMMARY

 ,EXECUTION-MODE=*SYNCHRONOUS(...)/*ASYNCHRONOUS(...)

      *SYNCHRONOUS(...)

           |    TEXT-OUTPUT=*STD/*SYSOUT/*SYSLST/*NONE

           |    ,STRUCTURE-OUTPUT=*SYSINF/*NONE/<composed-name1..100>

      *ASYNCHRONOUS(...)

           |    TEXT-OUTPUT=*SYSLST/<filename 1..54 without-gen-vers>

 ,OPERATIONAL-MODE=*OPERATION/*SIMULATION / *CHECK-NAME-CONSISTENCY(...)

      *CHECK-NAME-CONSISTENCY(...)

           |    PUBSET-STATE=*IMPORTED/*NOT-IMPORTED

           |    ,USER-ID=*OWN/*ALL
```

Operanden

PUBSET = <catid>
Kennung des zu erweiternden oder auf Erweiterbarkeit zu prüfenden, schon bestehenden SM-Pubsets. Die Kennung muss sich von den Kennungen der zusammenzuführenden SF-Pubsets unterscheiden.

Die angegebene Kennung geht ein in die Überprüfung der Dateinamenslänge und der automatischen Modifizierbarkeit von GUARD-Bedingungen.

ADD-VOLUME-SET =
Spezifiziert, welche weiteren SF-Pubsets in den SM-Pubset aufgenommen werden sollen.

ADD-VOLUME-SET = <catid>(...)
Kennungen der weiteren SF-Pubsets, die als Volume-Set in den SM-Pubset aufgenommen werden sollen. Die Pubset-Kennung wird zur Volume-Set-Kennung. Alle Volume-Set-Kennungen müssen sich untereinander und von der gewählten SM-Pubset-Kennung unterscheiden.

Die Volume-Sets müssen die im Abschnitt „Voraussetzungen für den Ablauf von SMPGEN“ beschriebenen Anforderungen erfüllen.
Falls nicht nur eine Prüfung durchgeführt werden soll (Operand OPERATIONAL-MODE= *CHECK-NAME-CONSISTENCY), sondern auch eine Konvertierung, dürfen die SF-Pubsets jedoch weder im lokalen noch in einem anderen System importiert sein; es müssen dafür jedoch bereits MRSCAT-Einträge mit dem richtigen Gerätetyp existieren.

Bei der reinen Prüffunktion (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY) wird die Angabe der folgenden Volume-Set-Eigenschaften ignoriert.

AVAILABILITY =
Bestimmt den Verfügbarkeitsgrad von Dateien, bezogen auf die Ausfallsicherheit des Volume-Sets.
Die Verfügbarkeit ist eine logische Volume-Set-Eigenschaft, die bei der Speicherplatzwahl als Auswahlkriterium zur Ablage von Dateien dient. Eine Änderung der Volume-Set-Eigenschaft muss der Systembetreuer explizit veranlassen.

AVAILABILITY = *STD
Voreinstellung: Der Volume-Set bietet keine erhöhte Ausfallsicherheit.

AVAILABILITY = *HIGH
Der Volume-Set bietet hohe Ausfallsicherheit.
Es liegt in der Verantwortung des Systembetreuers, zu überprüfen, dass die Ausfallsicherheit auch physikalisch gewährleistet ist.

USAGE =
Bestimmt die Art der Nutzung des Volume-Sets.

USAGE = *STD
Der Volume-Set soll als Ablageort für die Standarddateien genutzt werden. Arbeitsdateien (Dateiattribut WORK) dürfen dort nicht abgelegt werden.

USAGE = *HSMS-CONTROLLED
Der Volume-Set dient dem Subsystem HSMS zur Realisierung der Speicherebene S1 und des Backup-Volume-Sets eines SM-Pubsets. Benutzer dürfen dort keine Dateien ablegen.

PERFORMANCE-ATTR =
Bestimmt die Performance-Eigenschaften des Volume-Sets, bezogen auf Dateizugriffe. Es handelt sich hierbei um logische Attribute, die bei der Speicherplatzwahl als Auswahlkriterium dienen. Es erfolgt keine Validierung bzgl. der vorhandenen Hardware (z.B. Cache, SSD).

PERFORMANCE-ATTR = *STD
Voreinstellung: Für den Volume-Set sollen keine erhöhten Performance-Attribute bzgl. Dateizugriffen definiert werden.

PERFORMANCE-ATTR = *PARAMETERS(...)
Die für den Volume-Set geltenden Performance-Eigenschaften werden über folgende Unterstruktur definiert:

PERFORMANCE =
Beschreibt das I/O-Performance-Profil des Volume-Sets. Die Angabe einer Liste von bis zu drei Werten ist möglich.

PERFORMANCE = *STD
Voreinstellung: Der Volume-Set bietet keine erhöhte I/O-Performance.

PERFORMANCE = *HIGH
Der Volume-Set bietet erhöhte I/O-Performance.

PERFORMANCE = *VERY-HIGH
Der Volume-Set bietet sehr hohe I/O-Performance.

WRITE-CONSISTENCY =
Bestimmt, ob die zur Erhöhung der I/O-Performance eingesetzte Hardware auch im Falle eines Systemabsturzes Datenkonsistenz garantiert.

WRITE-CONSISTENCY = *BY-CLOSE
Voreinstellung: Schreib-I/Os mit erhöhter Performance sind nur unter Verzicht auf sofortige absturzsichere Datenablage möglich; erst beim Schließen der Datei gelangen die Daten auf ein Medium, das Datenerhaltung auch im Falle eines Systemabsturzes garantiert.

WRITE-CONSISTENCY = *IMMEDIATE
Auch bei Schreib-I/Os mit erhöhter Performance gelangen die Daten unmittelbar auf ein Medium, das Datenerhaltung auch im Falle eines Systemabsturzes garantiert.

S1-MIGRATED-FILES =
Vereinbart, ob es als Konflikt anzusehen ist, wenn in einem der zu überprüfenden Pubsets Dateien katalogisiert sind, die gegenwärtig auf die S1-Ebene migriert sind.

S1-MIGRATED-FILES = *NOT-ALLOWED
Voreinstellung: Der oben genannte Fall ist als Konflikt anzusehen.

S1-MIGRATED-FILES = *ALLOWED
Der oben genannte Fall ist nicht als Konflikt anzusehen.
Die S1-migrierten Dateien sind im Anschluss an die Konvertierung jedoch nicht mehr zugänglich, sondern müssen aus einer Backup-Datei zurückgeholt werden. Ihre Katalogeinträge sind noch erhalten, sodass sie nach Erzeugung der HSMS-Umgebung durch REPAIR-CATALOG-BY-RESTORE gezielt auf die S1-Ebene des SM-Pubsets gebracht werden können.

MAX-ERRORS = 32767 / <integer 0..32767>
Bei Auftreten der angegebenen Anzahl von Fehlern wird der Konsistenz-Check abgebrochen. Die vor dem Abbruch aufgetretenen Konflikte werden ausgegeben, falls ERROR-INFORMATION = *DETAIL angegeben wurde. 32767 ist Voreinstellung.

MONJV =
Vereinbart eine überwachende Jobvariable.

MONJV = *NONE
Voreinstellung: Es gibt keine überwachende Jobvariable.

MONJV = <filename 1..54 without-gen-vers>
Gibt den Namen der überwachenden Jobvariablen an. Sie muss bereits in einem zugreifbaren Pubset katalogisiert sein und darf nicht durch ein Kennwort geschützt sein.

Die Zustandsanzeige der Jobvariablen wird während der Konvertierung (OPERATIONAL-MODE=*OPERATION) auf folgende Weise gesetzt:

$I	zu Beginn der Bearbeitung
$R	nach erfolgreicher Durchführung der Bearbeitung
$A	bei fehlerhaftem Abbruch der Bearbeitung

Die Zustandsanzeige der Jobvariablen wird während des Konsistenz-Checks (OPERATIONAL-MODE=*CHECK-NAME- CONSISTENCY) auf folgende Weise gesetzt:

$I	zu Beginn der Konsistenz-Checks
$R	am Ende des Konsistenz-Checks, wenn Konsistenz besteht
$E	am Ende des Konsistenz-Checks, wenn keine Konsistenz besteht
$A	bei nicht vollständig durchgeführtem Konsistenz-Check

Die Angabe des Operanden ist nur sinnvoll bei Einsatz des Software-Produkts JV.

> **i** Kann die Jobvariable nicht versorgt werden (z.B. weil ein Kennwort gesetzt ist), wird der Konsistenz-Check trotzdem durchgeführt. Wenn die Jobvariable auch zu Beginn der Funktionsausführung einen definierten Inhalt haben soll, muss der Anwender sie selbst initialisieren. Wird die Funktionsausführung gleich zu Anfang abgebrochen, z.B. weil ein semantischer Fehler festgestellt wird, so versorgt SMPGEN die Jobvariable nicht.Liegt die Jobvariable auf einem der zusammenzuführenden Pubsets und ist nicht OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY(PUBSET-STATE= *IMPORTED) spezifiziert worden, so wird der SMPGEN-Auftrag zurückgewiesen.

ERROR-INFORMATION =
Steuert den Umfang der Fehlerinformationen, die in Text- und/oder Variablenform ausgegeben werden.

ERROR-INFORMATION = *DETAIL
Voreinstellung: Eine detaillierte Fehlerinformation soll ausgegeben werden. Dabei wird nicht nur die Anzahl der Namenskonflikte ausgegeben, sondern es werden auch alle mehrfach vorkommenden oder zu langen Namen einzeln aufgelistet.
Im Dialog an der Datensichtstation werden Bildschirmmasken ausgegeben, die die Möglichkeit bieten, die Fortsetzungsbildschirme oder die Detail-Bildschirme zu unterdrücken oder auf SYSLST umzuschalten (siehe Abschnitt „Ausgaben in Bildschirmmasken“).
Im Batch-Betrieb und auf SYSLST erfolgt die Ausgabe im selben Format, nur ohne Reaktionsmöglichkeit.

ERROR-INFORMATION = *SUMMARY
Es sollen keine detaillierten Fehlerinformationen ausgegeben werden, d.h. keine Listen von Namenskonflikten, sondern nur die Anzahl der Konflikte in den verschiedenen Namensklassen.

EXECUTION-MODE =
Gibt an, ob die Funktion in der aktuellen Task oder in einer neu zu erzeugenden Batch-Task ausgeführt werden soll.

EXECUTION-MODE = *SYNCHRONOUS(...)
Voreinstellung: Die Funktion soll in der laufenden Task ausgeführt werden.

TEXT-OUTPUT =
Steuert das Ausgabeziel der in Textformat erfolgenden Ausgaben, d.h. der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *STD
Voreinstellung: Die Ausgabe soll entsprechend der standardmäßigen Voreinstellung erfolgen. Im Dialogbetrieb erfolgt die Ausgabe sowohl auf SYSOUT als auch auf SYSLST. Im Batch-Betrieb erfolgt die Ausgabe auf SYSLST.

TEXT-OUTPUT = *SYSOUT
Die Ausgabe soll auf SYSOUT erfolgen.
Diese Angabe ist nur im Dialogbetrieb erlaubt. SYSOUT darf nicht auf eine Datei umgewiesen worden sein. Ist SYSOUT-Ausgabe gewünscht, so werden Bildschirmmasken ausgegeben, die mit der DUE-Taste zu quittieren sind. Beim Auftreten vieler Fehler ist es möglich, das detaillierte Fehlerprotokoll nachträglich auf SYSLST zu legen oder zu unterdrücken.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll auf SYSLST erfolgen.

TEXT-OUTPUT = *NONE
Es wird keine Ausgabe in Listen- oder Meldungsform gewünscht. Diese Angabe ist nur sinnvoll, wenn stattdessen der Operand STRUCTURE-OUTPUT angegeben wird. Fehlermeldungen und die Meldungen über Start und Erfolg der Funktion sind nicht unterdrückbar; sie werden auch bei der Angabe TEXT-OUTPUT=*NONE auf SYSOUT ausgegeben.

STRUCTURE-OUTPUT =
Steuert die Ausgabe von Fehlerinformationen in S-Variablen.

STRUCTURE-OUTPUT = *SYSINF
Voreinstellung: Die Fehlerinformationen sollen in die S-Variable ausgegeben werden, auf die der Anwender den Variablenstrom SYSINF zugewiesen hat.
Ist dieser Variablenstrom nicht aktiv, so erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = *NONE
Es erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = <composed-name 1..100>
Name der S-Variablen, in der die Fehlerinformationen abgelegt werden.
Die Variable muss vom Anwender als dynamisch definierte Struktur deklariert sein (`/DECLARE-VARIABLE ..., MULTIPLE-ELEMENTS=*LIST, TYPE=*STRUCTURE`).

EXECUTION-MODE = *ASYNCHRONOUS(...)
Die Funktion soll in einer neu zu erzeugenden Batch-Task ausgeführt werden. In diesem Fall wird die Informationsausgabe in eine S-Variable nicht unterstützt.

TEXT-OUTPUT =
steuert die Ausgabe der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll in eine Spoolout-Datei erfolgen, die bei Jobende ausgedruckt und gelöscht wird. Der Name der Spoolout-Datei entspricht der Standardeinstellung der SYSLST-Datei für ENTER-Tasks.

TEXT-OUTPUT = <filename 1..54 without-gen-vers>
Die Ausgabe soll in die angegebene Datei erfolgen. Falls die Datei bereits existiert, muss der Anwender das Schreibzugriffsrecht darauf besitzen. Die Datei darf nicht durch ein Kennwort geschützt sein. Sie muss lokal zugreifbar und darf keine temporäre Datei sein. Bei Jobende wird die Datei nicht ausgedruckt und auch nicht gelöscht.

OPERATIONAL-MODE =
Bestimmt, ob die Funktion „Erweitern eines SM-Pubsets“ real ausgeführt oder nur simuliert oder ob nur die Prüffunktion (Konsistenz-Check) ausgeführt werden soll.

OPERATIONAL-MODE = *OPERATION
Voreinstellung: Die Funktion soll real ausgeführt werden.

OPERATIONAL-MODE = *SIMULATION
Die Funktion soll nur simuliert werden.
Die Syntax der angegebenen Operanden wird geprüft.
Die Operanden MAX-ERRORS, MONJV, ERROR-INFORMATION und EXECUTION-MODE (mit Unteroperanden) haben jedoch nicht die in der Operandenbeschreibung aufgeführte Bedeutung; Ausgabemedium, ausführende Task usw. sind im Simulationsmodus nicht einstellbar.
Die Ausführung erfolgt immer in der laufenden Task: falls dies eine Dialog-Task ist, erfolgt die Ausgabe auf SYSOUT und SYSLST, falls es eine Batch-Task ist, nur auf SYSLST.

Bei erfolgreicher Syntaxprüfung wird eine Tabelle der Volume-Sets mit den spezifizierten logischen Eigenschaften ausgegeben.

Die Ausgangs-Pubsets müssen für diese Funktion nicht zugänglich sein; es erfolgen keine Plattenoperationen. Demgemäß werden auch keine Überprüfungen vorgenommen, für die die Platten verfügbar sein müssen, also kein Namens-Konsistenz-Check. Die MRSCAT-Einträge werden ebenfalls nicht überprüft.

OPERATIONAL-MODE = *CHECK-NAME-CONSISTENCY(...)
Es wird die Prüffunktion (Konsistenz-Check) von SMPGEN ausgeführt.
Dabei werden die SF-Pubsets, die in den Operanden CONTROL-VOLUME-SET und ADD-VOLUME-SET spezifiziert wurden, daraufhin überprüft, ob sie mit dem beim Operanden PUBSET angegebene SM-Pubset zusammengeführt bzw. konvertiert werden können.

PUBSET-STATE =
Gibt an, ob der Konsistenz-Check bei laufender Pubset-Session, also möglicherweise parallel zu anderen Anwender-Zugriffen auf die Pubsets, erfolgen soll.

PUBSET-STATE = *IMPORTED
Voreinstelling: Der Konsistenz-Check soll bei laufender Pubset-Session erfolgen. Die Pubsets sind bereits lokal importiert. Der Konsistenz-Check liefert nur eine ungefähre Aussage, die durch zwischenzeitliche Änderungen bereits überholt sein kann.

PUBSET-STATE = *NOT-IMPORTED
SMPGEN soll die Pubsets für den Konsistenz-Check exklusiv belegen. Die Pubsets sind weder lokal noch von einem anderen System importiert. Während des Konsistenz-Checks werden sie gegen andere Zugriffe gesperrt, daher ist das Ergebnis zum Zeitpunkt der Ausgabe korrekt.

USER-ID =
Spezifiziert, für welche Benutzerkennung(en) die Konsistenzprüfungen der Pubsets erfolgen sollen.

USER-ID = *OWN
Voreinstellung: Die Konsistenzprüfungen sollen sich nur auf Dateinamen, JVs und Guards unter der Aufrufer-Benutzerkennung erstrecken.
Die in der Anweisungsbeschreibung genannten allgemeinen Prüfungen für privilegierte Anwender (unerlaubte Dateien, S1-Migration) erfolgen auch bei dieser Angabe.

USER-ID = *ALL
Die Konsistenzprüfungen sollen sich auf alle Benutzerkennungen erstrecken. Diese Angabe ist unzulässig, falls einer oder mehrere der Pubsets im SHARED-SLAVE-Modus importiert ist.

Format für nicht-privilegierte Anwender

```
MODIFY-SYSTEM-MANAGED-PUBSET

 PUBSET=<catid>

 ,ADD-VOLUME-SET=list-poss(254):<catid>

 ,MAX-ERRORS=32767/<integer0..32767>

 ,MONJV=*NONE/ <filename 1..54 without-gen-vers>

 ,ERROR-INFORMATION=*DETAIL/*SUMMARY

 ,EXECUTION-MODE=*SYNCHRONOUS(...)/*ASYNCHRONOUS(...)

     *SYNCHRONOUS(...)

         |    TEXT-OUTPUT=*STD/*SYSOUT/*SYSLST/*NONE

         |    ,STRUCTURE-OUTPUT=*SYSINF/*NONE/<composed-name1..100>

     *ASYNCHRONOUS(...)

         |    TEXT-OUTPUT=*SYSLST/<filename 1..54 without-gen-vers>
```

Operanden

PUBSET = <catid>
Kennung des auf Erweiterbarkeit zu prüfenden, schon bestehenden SM-Pubsets. Die Kennung muss sich von den Kennungen der zusammenzuführenden SF-Pubsets unterscheiden.

Die angegebene Kennung geht ein in die Überprüfung der Dateinamenslänge und der automatischen Modifizierbarkeit von GUARD-Bedingungen.

ADD-VOLUME-SET =
Spezifiziert, welche weiteren SF-Pubsets in den SM-Pubset aufgenommen werden sollen.

ADD-VOLUME-SET = <catid>
Kennungen der weiteren SF-Pubsets, die als Volume-Set in den SM-Pubset aufgenommen werden sollen. Die Pubset-Kennung wird zur Volume-Set-Kennung. Alle Volume-Set-Kennungen müssen sich untereinander und von der gewählten SM-Pubset-Kennung unterscheiden.

Die Volume-Sets müssen die im Abschnitt „Voraussetzungen für den Ablauf von SMPGEN“ beschriebenen Anforderungen erfüllen.
Falls nicht nur eine Prüfung durchgeführt werden soll (Operand OPERATIONAL-MODE= *CHECK-NAME-CONSISTENCY), sondern auch eine Konvertierung, dürfen die SF-Pubsets jedoch weder im lokalen noch in einem anderen System importiert sein; es müssen dafür jedoch bereits MRSCAT-Einträge mit dem richtigen Gerätetyp existieren.

MAX-ERRORS = 32767 / <integer 0..32767>
Bei Auftreten der angegebenen Anzahl von Fehlern wird der Konsistenz-Check abgebrochen. Die vor dem Abbruch aufgetretenen Konflikte werden ausgegeben, falls ERROR-INFORMATION = *DETAIL angegeben wurde. 32767 ist Voreinstellung.

MONJV =
Vereinbart eine überwachende Jobvariable.

MONJV = *NONE
Voreinstellung: Es gibt keine überwachende Jobvariable.

MONJV = <filename 1..54 without-gen-vers>
Gibt den Namen der überwachenden Jobvariablen an. Sie muss bereits in einem zugreifbaren Pubset katalogisiert sein und darf nicht durch ein Kennwort geschützt sein.

Die Zustandsanzeige der Jobvariablen wird während des Konsistenz-Checks auf folgende Weise gesetzt:

$I	zu Beginn der Konsistenz-Checks
$R	am Ende des Konsistenz-Checks, wenn Konsistenz besteht
$E	am Ende des Konsistenz-Checks, wenn keine Konsistenz besteht
$A	bei nicht vollständig durchgeführtem Konsistenz-Check

Die Angabe des Operanden ist nur sinnvoll bei Einsatz des Software-Produkts JV.

> Kann die Jobvariable nicht versorgt werden (z.B. weil ein Kennwort gesetzt ist), wird der Konsistenz-Check trotzdem durchgeführt. Wenn die Jobvariable auch zu Beginn der Funktionsausführung einen definierten Inhalt haben soll, muss der Anwender sie selbst initialisieren. Wird die Funktionsausführung gleich zu Anfang abgebrochen, z.B. weil ein semantischer Fehler festgestellt wird, so versorgt SMPGEN die Jobvariable nicht.

ERROR-INFORMATION =
Steuert den Umfang der Fehlerinformationen, die in Text- und/oder Variablenform ausgegeben werden.

ERROR-INFORMATION = *DETAIL
Voreinstellung: Eine detaillierte Fehlerinformation soll ausgegeben werden. Dabei wird nicht nur die Anzahl der Namenskonflikte ausgegeben, sondern es werden auch alle mehrfach vorkommenden oder zu langen Namen einzeln aufgelistet.
Im Dialog an der Datensichtstation werden Bildschirmmasken ausgegeben, die die Möglichkeit bieten, die Fortsetzungsbildschirme oder die Detail-Bildschirme zu unterdrücken oder auf SYSLST umzuschalten (siehe Abschnitt „Ausgaben in Bildschirmmasken“).
Im Batch-Betrieb und auf SYSLST erfolgt die Ausgabe im selben Format, nur ohne Reaktionsmöglichkeit.

ERROR-INFORMATION = *SUMMARY
Es sollen keine detaillierten Fehlerinformationen ausgegeben werden, d.h. keine Listen von Namenskonflikten, sondern nur die Anzahl der Konflikte in den verschiedenen Namensklassen.

EXECUTION-MODE =
Gibt an, ob die Funktion in der aktuellen Task oder in einer neu zu erzeugenden Batch-Task ausgeführt werden soll.

EXECUTION-MODE = *SYNCHRONOUS(...)
Voreinstellung: Die Funktion soll in der laufenden Task ausgeführt werden.

TEXT-OUTPUT =
Steuert das Ausgabeziel der in Textformat erfolgenden Ausgaben, d.h. der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *STD
Voreinstellung: Die Ausgabe soll entsprechend der standardmäßigen Voreinstellung erfolgen. Im Dialogbetrieb erfolgt die Ausgabe sowohl auf SYSOUT als auch auf SYSLST. Im Batch-Betrieb erfolgt die Ausgabe auf SYSLST.

TEXT-OUTPUT = *SYSOUT
Die Ausgabe soll auf SYSOUT erfolgen.
Diese Angabe ist nur im Dialogbetrieb erlaubt. SYSOUT darf nicht auf eine Datei umgewiesen worden sein. Ist SYSOUT-Ausgabe gewünscht, so werden Bildschirmmasken ausgegeben, die mit der DUE-Taste zu quittieren sind. Beim Auftreten vieler Fehler ist es möglich, das detaillierte Fehlerprotokoll nachträglich auf SYSLST zu legen oder zu unterdrücken.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll auf SYSLST erfolgen.

TEXT-OUTPUT = *NONE
Es wird keine Ausgabe in Listen- oder Meldungsform gewünscht. Diese Angabe ist nur sinnvoll, wenn stattdessen der Operand STRUCTURE-OUTPUT angegeben wird. Fehlermeldungen und die Meldungen über Start und Erfolg der Funktion sind nicht unterdrückbar; sie werden auch bei der Angabe TEXT-OUTPUT=*NONE auf SYSOUT ausgegeben.

STRUCTURE-OUTPUT =
Steuert die Ausgabe von Fehlerinformationen in S-Variablen.

STRUCTURE-OUTPUT = *SYSINF
Voreinstellung: Die Fehlerinformationen sollen in die S-Variable ausgegeben werden, auf die der Anwender den Variablenstrom SYSINF zugewiesen hat.
Ist dieser Variablenstrom nicht aktiv, so erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = *NONE
Es erfolgt keine Ausgabe in eine S-Variable.

STRUCTURE-OUTPUT = <composed-name 1..100>
Name der S-Variablen, in der die Fehlerinformationen abgelegt werden.
Die Variable muss vom Anwender als dynamisch definierte Struktur deklariert sein (`/DECLARE-VARIABLE ..., MULTIPLE-ELEMENTS=*LIST, TYPE=*STRUCTURE`).

EXECUTION-MODE = *ASYNCHRONOUS(...)
Die Funktion soll in einer neu zu erzeugenden Batch-Task ausgeführt werden. In diesem Fall wird die Informationsausgabe in eine S-Variable nicht unterstützt.

TEXT-OUTPUT =
steuert die Ausgabe der Listen, Protokollierungs- und Fehlermeldungen.

TEXT-OUTPUT = *SYSLST
Die Ausgabe soll in eine Spoolout-Datei erfolgen, die bei Jobende ausgedruckt und gelöscht wird. Der Name der Spoolout-Datei entspricht der Standardeinstellung der SYSLST-Datei für ENTER-Tasks.

TEXT-OUTPUT = <filename 1..54 without-gen-vers>
Die Ausgabe soll in die angegebene Datei erfolgen. Falls die Datei bereits existiert, muss der Anwender das

Schreibzugriffsrecht darauf besitzen. Die Datei darf nicht durch ein Kennwort geschützt sein. Sie muss lokal zugreifbar und darf keine temporäre Datei sein. Bei Jobende wird die Datei nicht ausgedruckt und auch nicht gelöscht.

14.8 Fehlerverhalten

SMPGEN legt während der Abarbeitung der Anweisungen verschiedene Arbeitsdateien (WORK-Dateien) an, die normalerweise nur während des SMPGEN-Laufs existieren.

Bei Systemabbruch oder Recovery-Fehlern können solche Dateien natürlich über den SMPGEN-Lauf hinaus existieren. Sie sind an den Präfixen SYSWRK.SMPGEN bzw. S.SMPGEN zu erkennen. Die Dateien SYSWRK.SMPGEN... und müssen vor einem neuen SMPGEN-Lauf durch den Systembetreuer gelöscht werden.

Besonderheiten beim Erzeugen oder Erweitern von SM-Pubsets

SMPGEN versucht, so früh wie möglich Gründe, die die Bildung eines SM-Pubsets unmöglich machen, durch einen Konsistenz-Check auszuschließen. Auch nach erfolgreichem Check werden zunächst keine Pubset-Strukturen geändert, sondern nur neue Verwaltungsdateien aufgebaut.
Bei einer Erweiterung werden die alten Verwaltungsdaten des SM-Pubsets durch Kopieren gesichert. Während der Konvertierung bleiben die abgelösten Verwaltungsdateien ebenfalls weitgehend erhalten, sodass SMPGEN im Falle eines Scheiterns der Konvertierung oder auch eines Taskabbruchs die alte Umgebung wiederherstellen kann. Die beteiligten SF-Pubsets sind dann wieder in ihrem alten Zustand.
Lediglich die SYSPBN- und die EAM-Dateien können gelöscht sein.

Nur in der letzten Phase (nachdem bereits alle Prüfungen und Plattenspeicher-Anforderungen erfolgreich waren) ist eine Rückkehr zum alten Zustand der SF-Pubsets nicht mehr möglich. Bei einem nichtbehebbaren Fehler in dieser Phase, sind - wie bei einem Systemabbruch in der mittleren Phase der Konvertierung - die SF-Pubsets nicht mehr importierbar; sie müssen aus einer vorhergehenden FDDRL-Sicherung rekonstruiert werden.
Während der kritischen Umwandlungsphase sind die Pubsets bzw. Volume-Sets als nichtimportierbar gekennzeichnet.

Ist im Anschluss an einen Systemabbruch ein Import möglich, entweder als SF- oder als SM-Pubset, so kann davon ausgegangen werden, dass alle relevanten Informationen erhalten sind und ein konsistenter Plattenzustand erreicht ist.

Ist im Anschluss an einen Systemabbruch während einer SM-Pubset-Erweiterung ein Import als SM-Pubset möglich, so kann davon ausgegangen werden, dass ein konsistenter Plattenzustand erreicht ist. Jedoch hat der SM-Pubset abhängig vom Zeitpunkt des Systemabbruchs entweder seinen Ausgangszustand oder den korrekt um SF-Pubsets erweiterten Zustand.

Falls - bei gescheitertem Konvertierungsversuch - die Pubsets zu SF-Pubsets rückkonvertiert wurden und dabei die Wiederherstellung der alten GUARDS-Umgebung nicht gelang, so werden die Pubsets dennoch als importierbar gekennzeichnet. Nach dem Import als SF-Pubsets sind dann mit Guards geschützte Objekte zunächst nicht ansprechbar. Die alte GUARDS-Umgebung kann durch den Systembetreuer jedoch selbst wiederhergestellt werden, indem er mit `/CHANGE-GUARD-FILE` die von SMPGEN umbenannten Guards-Kataloge zuweist. Er findet sie unter dem Namen `SYSWRK.SMPGEN.SYSCAT.GUARDS.<pubsetid>`.

Die neuen bzw. erweiterten CMS-Verwaltungsdateien werden bei Rückkonvertierung zu SF-Pubsets gelöscht bzw. auf den Ausgangszustand gebracht.
Können sie bei der Rückkonvertierung aus irgendeinem Grund nicht gelöscht werden, so ist dies kein Hindernis für einen erneuten SMPGEN-Lauf:
Soll wieder derselbe Volume-Set zum Control-Volume-Set gemacht werden, so werden ggf. die darauf noch existierenden neuen CMS-Verwaltungsdateien wieder verwendet. Soll ein anderer Pubset Control-Volume-Set werden, so löscht das CMS implizit die auf dem „einfachen“ Volume-Set befindlichen, vom vorherigen SMPGEN-Lauf übrig gebliebenen CMS-Verwaltungsdateien.

Beim Exportieren der Pubsets kann - bei Einsatz von GUARDS - folgende Meldung für jedes betroffene Pubset ausgegeben werden:

```
PRO6009   ERROR WHEN CLOSING GUARDS CATALOG
```

Diese Meldung kann ignoriert werden.

Besonderheiten bei SMPGEN-Anweisungen, bei denen alle betroffenen Pubsets durch SMPGEN selbst importiert werden

(z.B. CREATE-SYSTEM-MANAGED-PUBSET mit OPERATIONAL-MODE=*OPERATION oder mit OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY(PUBSET-STATE= *NOT-IMPORTED))

Bei fehlerhaftem Abbruch, auch abnormaler Programm-/Taskterminierung, werden die MRSCAT-Einträge bereinigt. Nur wenn auch die Recovery-Routine abbricht, bleiben die MRSCAT-Einträge im Zustand „inaccessible/in SMPGEN-Bearbeitung/nicht löschbar“. Die MRSCAT-Einträge sind dann nicht löschbar, sodass die Pubsets in diesem Systemlauf nicht mehr importiert werden können und kein zweites Starten des SMPGEN-Laufs möglich ist.

STEP-Terminierung

Wird eine syntaktisch oder semantisch falsche SMPGEN-Anweisung eingegeben oder kann die Anweisung nicht vollständig bearbeitet werden, dann wird zur nächsten STEP-Anweisung verzweigt, falls der Fehler innerhalb des Programmlaufs und nicht in der asynchron erzeugten Task auftritt.

Tritt ein Fehler auf, der die Ablauffähigkeit des Programms überhaupt infrage stellt (z.B. keine Informationsausgabe möglich, Dateiende auf SYSDTA), so beendet sich das Programm mit Step-Terminierung, d.h. innerhalb einer Prozedur werden alle Anweisungen und Kommandos bis zum nächsten `/SET-JOB-STEP` oder `/EXIT-JOB` übersprungen.

14.9 Ausgaben in Bildschirmmasken

Die Bildschirmmasken bestehen aus jeweils 80 Spalten und 24 Zeilen und haben folgenden allgemeinen Aufbau:

- Kopfteil mit Überschrift
- Datenteil zur Ausgabe von Objekt-Attributen (Ausgabefelder)
  Die Ausgabefelder werden von SMPGEN variabel versorgt.
  Sie können nicht vom Anwender überschrieben werden.
  Ausnahmen bilden lediglich die Felder „MORE“ in den Masken 3, 5 und 7, bei denen eine Eingabemöglichkeit besteht.
- Fußzeile mit Eingabemöglichkeit (z.B. Wechseln des Ausgabemediums) Bei Ausgabe auf SYSLST oder bei Batch-Tasks entfällt die Fußzeile. Es wird dann immer die vollständige Information ausgegeben (entsprechend der Eingabe MORE=Y bei Ausgabe an der Datensichtstation).

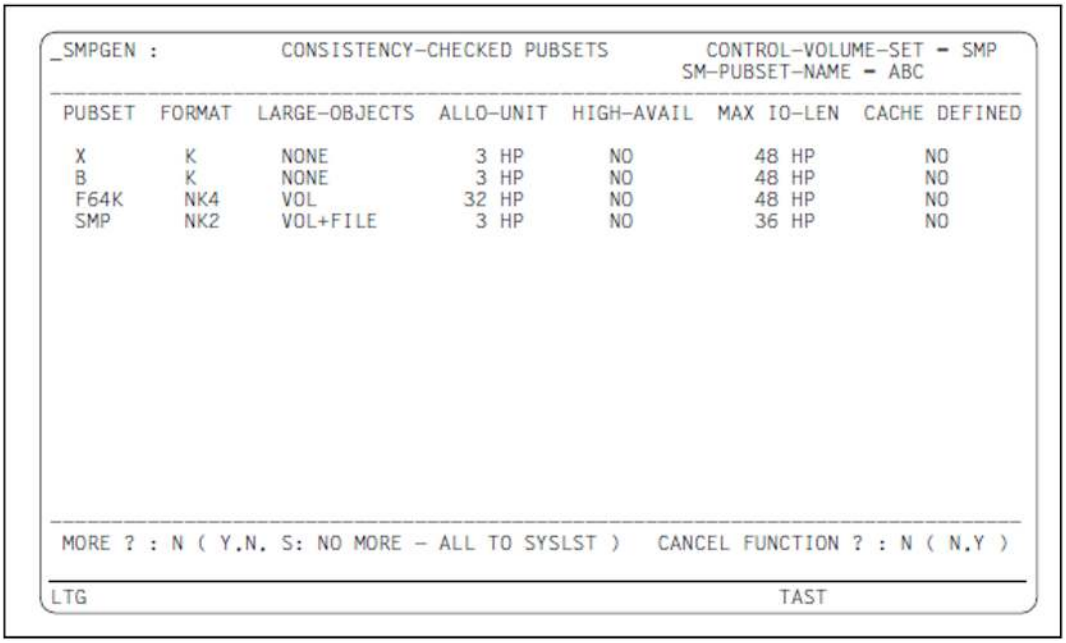
Das Zeichen in Zeile 1, Spalte 1 (Unterstützung der lokalen Hardcopy-Funktion) ist überschreibbar, um mit der Schreibmarke darauf positionieren zu können.
An der Datensichtstation wird zur Fortsetzung des Programmlaufs jeweils die DUE-Taste gedrückt. Die Tasten K1, K2 und K3 werden ignoriert.

Folgende Masken werden angeboten:

1. CONSISTENCY-CHECKED PUBSETS
2. INADMISSIBLE FILES
3. FILE-NAME CONFLICTS
4. MULTIPLE FILE-NAME CONFLICTS
5. JV-NAME CONFLICTS
6. MULTIPLE JV-NAME CONFLICTS
7. GUARD-NAME CONFLICTS
8. MULTIPLE GUARD-NAME CONFLICTS
9. FILE-NAME LENGTH ERRORS
10. JV-NAME LENGTH ERRORS
11. GUARD CONDITION MODIFICATION FAILURES
12. VOLUME-SET TABLE 1
13. VOLUME-SET TABLE

MASKE 1: CONSISTENCY-CHECKED PUBSETS

Diese Maske wird bei den Anweisungen CREATE- und MODIFY-SYSTEM-MANAGED-PUBSET bei Ausführung der reinen Prüffunktion bzw. bei Ausführung der Anweisung CHECK-NAME-CONSISTENCY erzeugt.



KOPF

CONTROL-VOLUME-SET

Die für den gleichnamigen Operanden eingegebene Pubset-Kennung;
*NONE wird durch *NON dargestellt

SM-PUBSET-NAME

Über eine Operanden eingegebene SM-Pubset-Kennung

DATEN

PUBSET

Pubset-Kennung des überprüften Pubsets

FORMAT

Ermitteltes Plattenformat: K / NK2 / NK4

LARGE-OBJECTS

Zeigt an, ob Volumes oder Dateien größer als 32 GB erlaubt sind:

VOL+FILE:	Volumes und Dateien größer als 32 GB sind erlaubt
VOL:	Volumes größer als 32 GB sind erlaubt
NONE:	Weder Volumes noch Dateien größer als 32 GB sind erlaubt

ALLOCATION-UNIT

Allokierungseinheit in Halbseiten (je 2KB)

HIGH-AVAIL

Erhöhte Verfügbarkeit: YES (RAID etc.) / NO

MAXIMAL IO-LENGTH

Maximale Transfer-Länge bei Platten-Ein-/Ausgaben

CACHE DEFINED

Ist ein Cache-Medium definiert worden?: YES / NO

FUSSZEILE

MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr in der aktuellen Anweisung angegebene Pubsets überprüft wurden, als auf diesem und den bereits gezeigten Bildschirmen dargestellt sind.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Pubsets an der Datensichtstation ausgegeben.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Pubsets an der Datensichtstation, aber die gesamte Pubset-Information (unabhängig vom OUTPUT-Parameter) auf SYSLST ausgegeben.

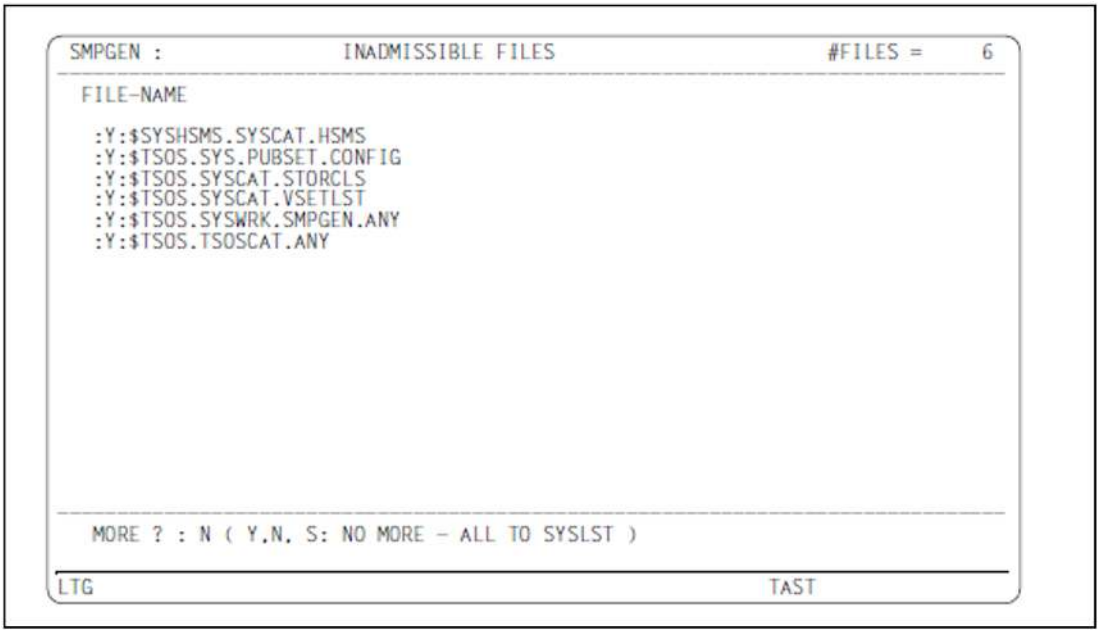
CANCEL FUNCTION

Dieses Feld ist mit „N“ vorbelegt. Durch Überschreiben mit einem anderen Inhalt wird die Funktion abgebrochen.

MASKE 2: INADMISSIBLE FILES

Diese Maske wird für privilegierte Anwender von SMPGEN erzeugt, wenn unzulässige Dateien in den zu überprüfenden Pubsets existieren.

Unzulässige Dateien sind solche, die in der Beschreibung der Prüffunktion genannt sind und die im SM-Pubset nicht verwendet werden können.
Die unzulässigen Dateien können in den Pubsets nicht katalogisiert werden, weil SMPGEN Arbeitsdateien oder neue Verwaltungsdateien unter demselben Namen aufbaut bzw. sie als Paging-Dateien im SM-Pubset verwendet.



KOPF

#FILES

Anzahl der unzulässigen Dateien

DATEN

FILE-NAME

Pfadname

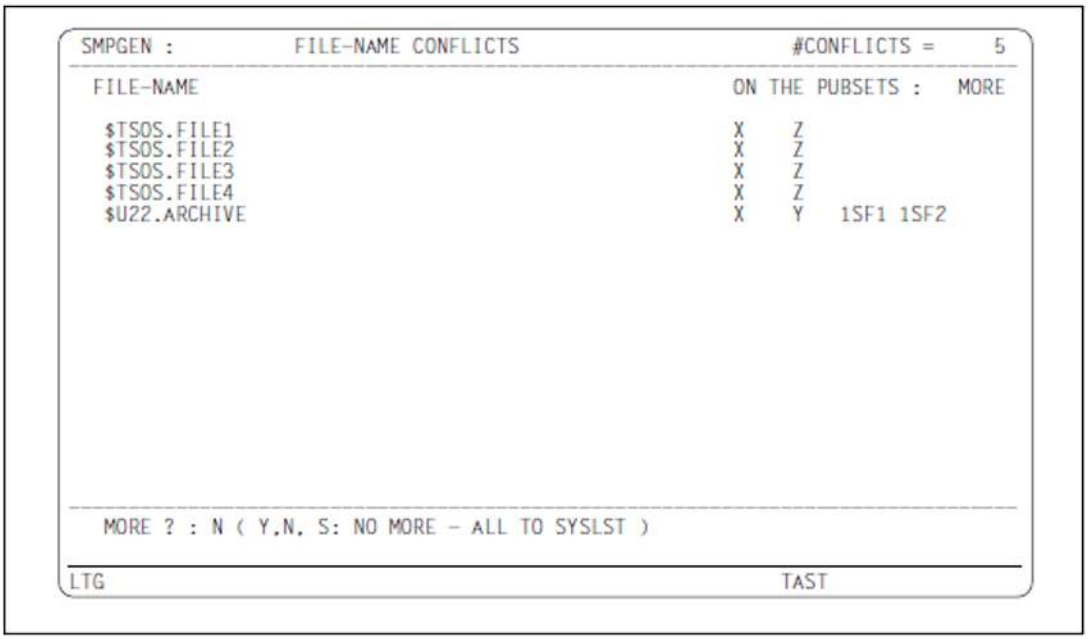
FUSSZEILE

MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr unzulässige Dateien existieren, als auf diesem und den bereits gezeigten Bildschirmen dargestellt wurden.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Pubsets an der Datensichtstation ausgegeben.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Dateinamen an der Datensichtstation ausgegeben, aber alle unzulässigen Dateien unabhängig vom TEXT-OUTPUT-Operanden auf SYSLST aufgelistet. Die anderen Ausgaben werden normal fortgesetzt.

MASKE 3: FILE-NAME CONFLICTS

Diese Maske wird erzeugt, wenn Dateinamenskonflikte auftreten. Kommen Dateinamen auf mehr als vier Pubsets vor, so wird im Anschluss an diesen Bildschirm für jeden dieser Dateinamen eine vollständige Pubset-Tabelle ausgegeben (Maske 4), falls der Anwender im überschreibbaren Feld am rechten Rand nichts anderes bestimmt. Erst im Anschluss an die vollständigen Pubset-Tabelle wird die Ausgabe der Dateinamenskonflikte mit dem nächsten Maske-3-Bildschirm fortgesetzt.



KOPF

#CONFLICTS

Anzahl der Dateien, die auf mehreren der überprüften Pubsets katalogisiert sind

DATEN

FILE-NAME

Dateiname ohne Katalogkennung

ON THE PUBSETS

Katalogkennung der betreffenden Pubsets

MORE

Ist mit „Y“ vorbelegt, falls der Name auf mehr als vier Pubsets existiert, sonst mit Leerzeichen.
Durch Überschreiben mit „N“ oder Leerzeichen wird Maske 4 nicht mehr an der Datensichtstation ausgegeben.

Durch Überschreiben mit „S“ wird Maske 4 nicht mehr an der Datensichtstation, aber unabhängig vom TEXT-OUTPUT-Operanden auf SYSLST ausgegeben.

FUSSZEILE

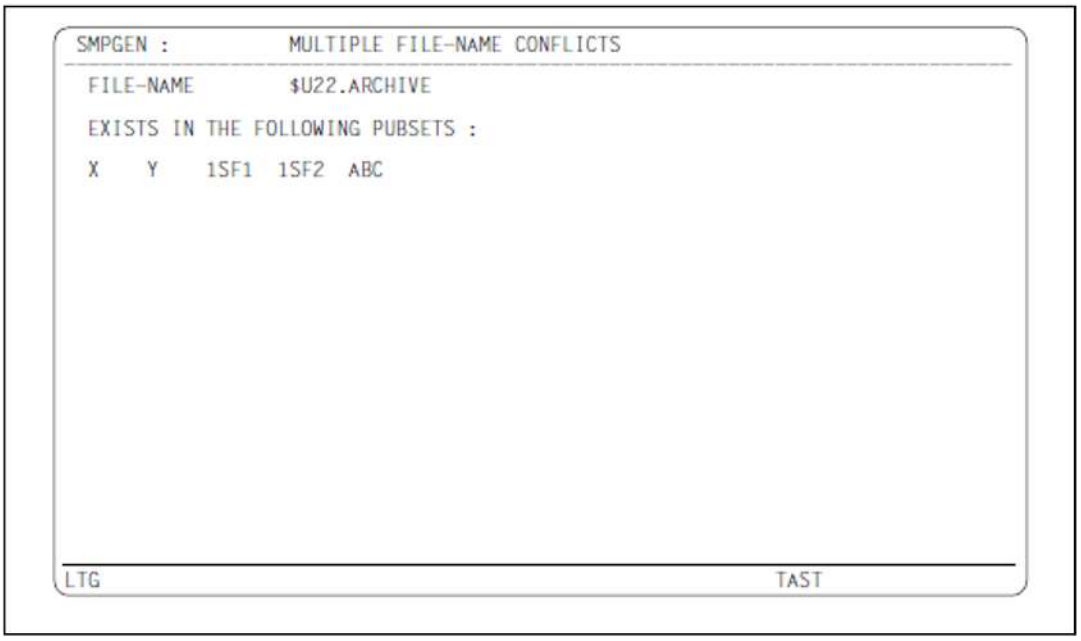
MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr Dateinamenskonflikte bestehen, als auf diesem und den bereits gezeigten Bildschirmen dargestellt.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Dateinamen mehr ausgegeben, weder an der Datensichtstation noch auf SYSLST.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Dateinamen mehr an der Datensichtstation, aber alle Dateinamenskonflikte unabhängig vom TEXT-OUTPUT-Operanden auf SYSLST ausgegeben. Die anderen Ausgaben werden normal fortgesetzt.

MASKE 4: MULTIPLE FILE-NAME CONFLICTS

Diese Maske wird erzeugt, wenn Dateinamen auf mehr als vier Pubsets vorkommen. Sie wird im Anschluss an Maske 3 für jeden dort mit MORE=Y markierten Dateinamen ausgegeben, wenn das Feld nicht mit „N“ überschrieben wurde.

Es werden auch die bereits in Maske 3 angezeigten Pubsets ausgegeben. Eingaben sind nicht möglich.



DATEN

FILE-NAME

Dateiname ohne Katalogkennung

EXISTS IN THE FOLLOWING PUBSETS

Pubset-Kennungen der betreffenden Pubsets

MASKE 5: JV NAME CONFLICTS

Diese Maske ist in Aufbau, Bedeutung und Reaktionsmöglichkeiten analog zu Maske 3, nur dass hier die Konflikte bei Jobvariablen-Namen betrachtet werden.

MASKE 6: MULTIPLE JV NAME CONFLICTS

Diese Maske ist in Aufbau, Bedeutung und Reaktionsmöglichkeiten analog zu Maske 4, nur dass hier die Konflikte bei Jobvariablen-Namen betrachtet werden.

MASKE 7: GUARD-NAME CONFLICTS

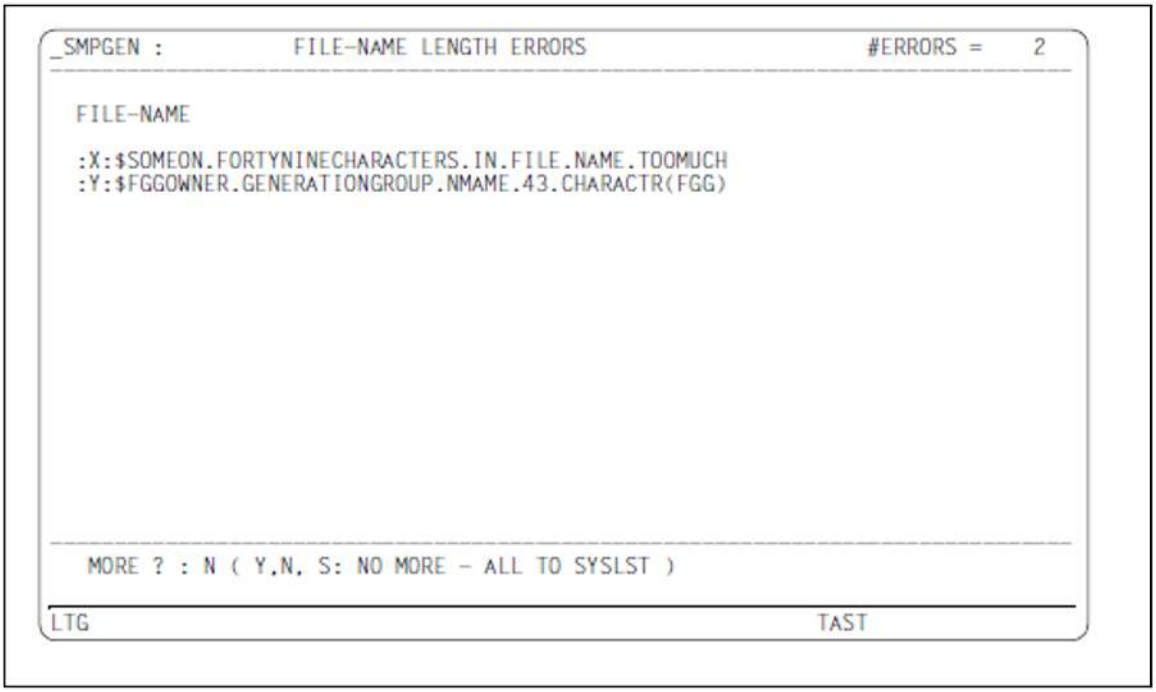
Diese Maske ist in Aufbau, Bedeutung und Reaktionsmöglichkeiten analog zu Maske 3, nur dass hier die Konflikte bei Guard-Namen betrachtet werden. Eine Verbindungslinie aus Punkten überbrückt den Zwischenraum zwischen Guard-Namen und betroffenen Pubsets.

MASKE 8: MULTIPLE GUARD-NAME CONFLICTS

Diese Maske ist in Aufbau, Bedeutung und Reaktionsmöglichkeiten analog zu Maske 4, nur dass hier die Konflikte bei Guard-Namen betrachtet werden.

MASKE 9: FILE-NAME LENGTH ERRORS

Diese Maske wird erzeugt, wenn Dateinamen auftreten, die bei Ersetzen der alten Pubset-Kennung durch die neue Pubset-Kennung die erlaubte Pfadlänge von 54 Zeichen überschreiten bzw. bei Datei-Generationsgruppennamen, die mehr als 47 Zeichen lang sind.
Die einzelnen Datei-Generationen werden nicht extra angezeigt.



KOPF

#ERRORS

Anzahl der Dateinamen, die die erlaubte Länge überschreiten.
Pfadnamen, die sich nur durch die Katalogkennung unterscheiden, werden wie unterschiedliche Dateinamen behandelt.

DATEN

FILE-NAME

Pfadname (mit Katalogkennung); bei Datei-Generationsgruppennamen gefolgt von der Zeichenkette „FGG)“.

FUSSZEILE

MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr Dateinamen betroffen sind, als auf diesem und den bereits gezeigten Bildschirmen dargestellt.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Dateinamen an der Datensichtstation

ausgegeben.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Dateinamen an der Datensichtstation, aber alle zu langen Dateinamen unabhängig vom TEXT-OUTPUT-Operanden auf SYSLST ausgegeben. Die anderen Ausgaben werden normal fortgesetzt.

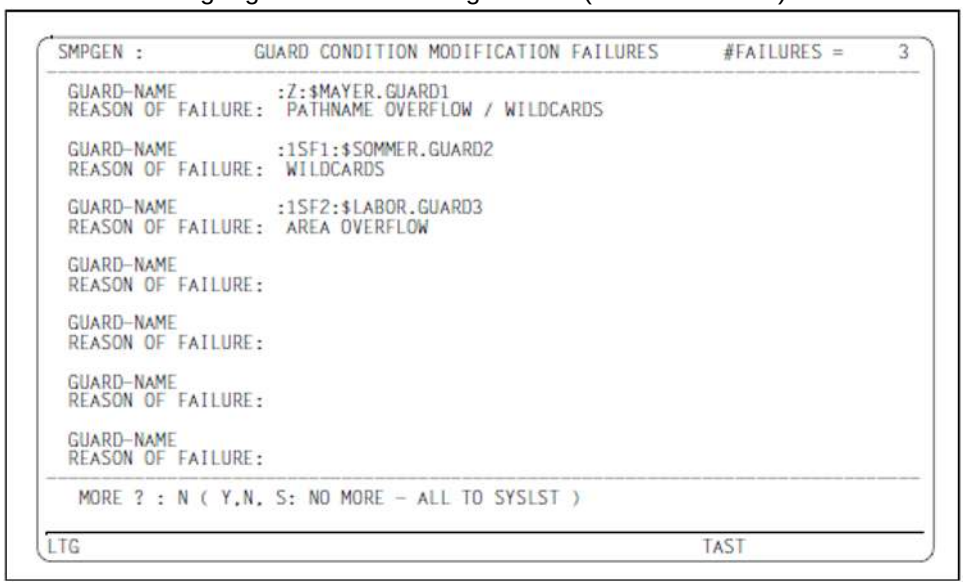
MASKE 10: JV-NAME LENGTH ERRORS

Wie Maske 9, nur dass hier die Jobvariablen-Namen aufgelistet werden.

MASKE 11: GUARD CONDITION MODIFICATION FAILURES

Diese Maske wird erzeugt, wenn Guard-Bedingungs-Fehler auftreten, d.h. wenn Programmnamen in Guard-Bedingungen nicht an die neue Pubset-Kennung angepasst werden konnten. Mögliche Ursachen sind:

- der Programmname enthält Wildcards
- der Programmname würde länger als 54 Zeichen sein
- der Guard-Bedingungsteil würde zu lang werden (= Area overflow)



KOPF

#FAILURES

Anzahl der Guard-Bedingungs-Fehler

DATEN

GUARD-NAME

Pfadname des betreffenden Guards

REASON OF FAILURE

Ursache bzw. Liste der Ursachen: WILDCARDS oder PATHNAME OVERFLOW oder AREA OVERFLOW oder eine beliebige Kombination der drei Ursachen.

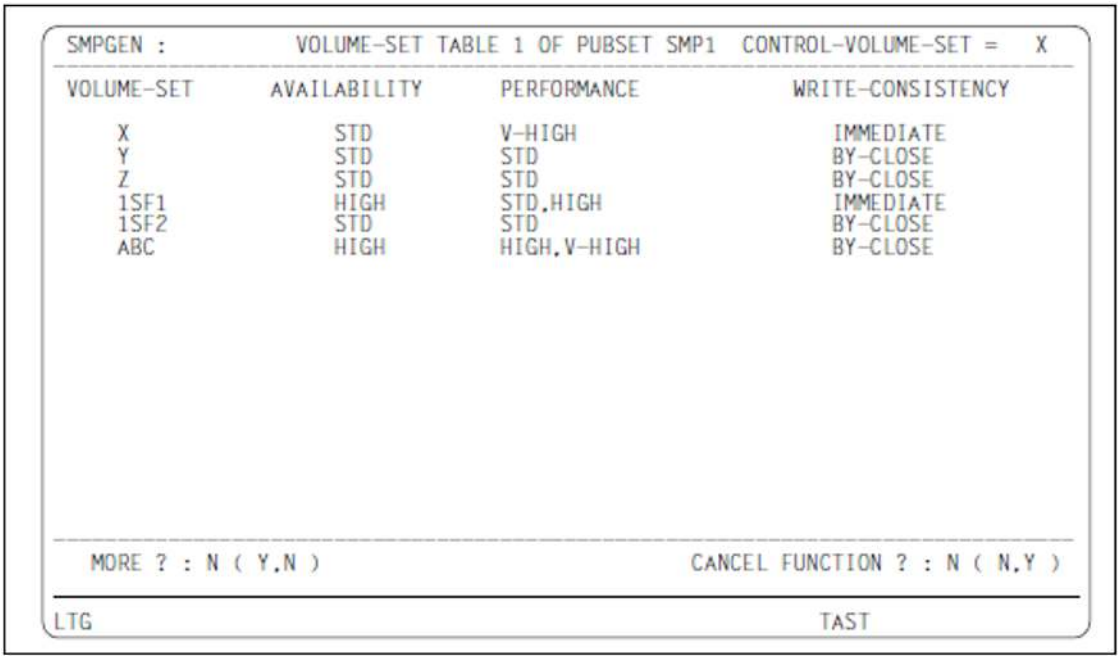
FUSSZEILE

MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr Guards-Bedingungs-Fehler aufgetreten sind, als auf diesem und den bereits gezeigten Bildschirmen dargestellt.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Warnungen dieser Art an der Datensichtstation ausgegeben.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Warnungen dieser Art an der Datensichtstation ausgegeben, aber alle Warnungen dieser Art - unabhängig vom TEXT-Output-Operanden - auf SYSLST aufgelistet. Die anderen Ausgaben werden normal fortgesetzt.

MASKE 12: VOLUME-SET TABLE 1

Diese Maske wird bei der Anweisung CREATE-/MODIFY-SYSTEM-MANAGED-PUBSET erzeugt, falls der Aufruf aus einer Dialog-Task kommt. Diese Tabelle wird unabhängig vom OUTPUT-Operanden auf SYSOUT ausgegeben, damit die Eingaben vor Ausführung der Anweisung noch einmal kontrolliert werden können. Im Fußfeld der Maske kann ein Abbruch der Funktion veranlasst werden.



KOPF

PUBSET

Katalogkennung des zu erzeugenden oder zu erweiternden SM-Pubset

CONTROL-VOLUME-SET

Die für den gleichnamigen Operanden eingegebene Pubset-Kennung beim Erzeugen eines SM-Pubsets bzw. die Volume-Set-Kennung des Control-Volume-Sets beim Erweitern eines SM-Pubsets.
*NONE wird durch *NON dargestellt

DATEN

VOLUME-SET

Kennung des neu einzurichtenden Volume-Sets (ehemalige SF-Pubset-Kennung)

AVAILABILITY

Operandenwert des gleichnamigen Unteroperanden von ADD-VOLUME-SET

PERFORMANCE

Operandenwert des gleichnamigen Unteroperanden von ADD-VOLUME-SET.Format: STD / HIGH / STD, HIGH / V-HIGH / HIGH, V-HIGH / STD, HIGH, V-HIGH

WRITE-CONSISTENCY

Operandenwert des Unteroperanden WRITE-CONSISTENCY des ADD-VOLUME-SET-Operanden

FUSSZEILE

MORE

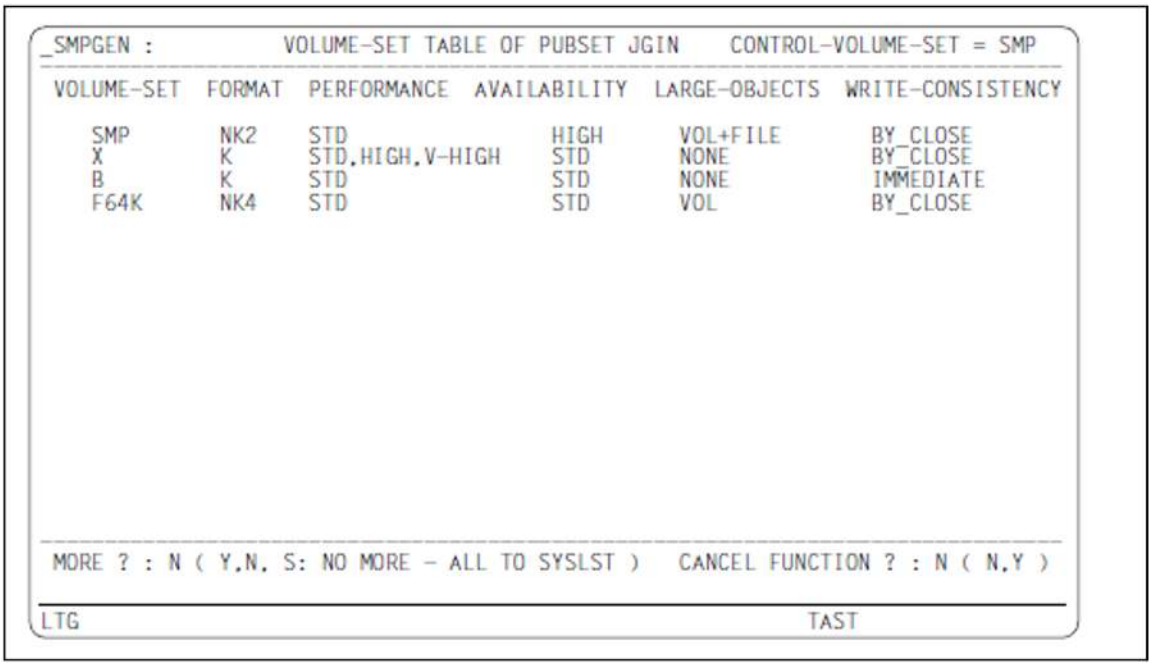
Dieses Feld ist mit „Y“ vorbelegt, falls mehr Volume-Sets zum zu erzeugenden bzw. zu erweiternden SM-Pubset gehören bzw. dazukommen, als auf diesem und den bereits gezeigten Bildschirmen dargestellt. Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Volume-Sets an der Datensichtstation ausgegeben. Die weiteren Ausgaben werden normal fortgesetzt.

CANCEL FUNCTION

Dieses Feld ist mit „N“ vorbelegt. Durch Überschreiben mit einem anderen Inhalt wird die Funktion abgebrochen.

MASKE 13: VOLUME-SET TABLE

Diese Maske wird bei den Anweisungen CREATE-/MODIFY-SYSTEM-MANAGED-PUBSET - wenn nicht nur die reine Prüffunktion ausgeführt wird - erzeugt.



KOPF

CONTROL-VOLUME-SET

Die für den gleichnamigen Operanden eingegebene Pubset-Kennung beim Erzeugen eines SM-Pubsets bzw. die Volume-Set-Kennung des Control-Volume-Sets beim Erweitern eines SM-Pubsets.
*NONE wird durch *NON dargestellt

PUBSET

Katalogkennung des zu erzeugenden oder zu erweiternden SM-Pubset

DATEN

VOLUME-SET

Kennung des neu einzurichtenden Volume-Sets (= ehemalige SF-Pubset-Kennung)

FORMAT

Ermitteltes Plattenformat: K / NK2 / NK4

AVAILABILITY

Operandenwert des gleichnamigen Unteroperanden von ADD-VOLUME-SET

LARGE-OBJECTS

Zeigt an, ob Volumes oder Dateien größer als 32 GB erlaubt sind:

VOL+FILE:	Volumes und Dateien größer als 32 GB sind erlaubt
VOL:	Volumes größer als 32 GB sind erlaubt
NONE:	Weder Volumes noch Dateien größer als 32 GB sind erlaubt

PERFORMANCE

Operandenwert des gleichnamigen Unteroperanden von ADD-VOLUME-SET.Format: STD / HIGH / STD, HIGH / V-HIGH / HIGH, V-HIGH / STD, HIGH, V-HIGH

WRITE-CONSISTENCY

Operandenwert des Unteroperanden WRITE-CONSISTENCY des ADD-VOLUME-SET-Operanden

FUSSZEILE

MORE

Dieses Feld ist mit „Y“ vorbelegt, falls mehr Volume-Sets zum zu erzeugenden bzw. zu erweiternden SM-Pubset gehören, als auf diesem und den bereits gezeigten Bildschirmen dargestellt.
Durch Überschreiben dieses Feldes mit „N“ werden keine weiteren Volume-Sets an der Datensichtstation ausgegeben.
Durch Überschreiben dieses Feldes mit „S“ werden keine weiteren Volume-Sets an der Datensichtstation, aber die vollständige Volume-Set-Liste unabhängig vom Operanden OUTPUT auf SYSLST ausgegeben. Die weiteren Ausgaben werden normal fortgesetzt.

CANCEL FUNCTION

Dieses Feld ist mit „N“ vorbelegt. Durch Überschreiben mit einem anderen Inhalt wird die Funktion abgebrochen.

14.10 Meldungen von SMPGEN

Die Meldungen des Dienstprogramms SMPGEN haben die Meldungsklasse SGU oder SPG.
SMPGEN gibt keine garantierten Meldungen aus.

Siehe auch den Abschnitt „Meldungen und ihre Bedeutung“ (Darstellungsmittel).

14.11 Eigenschaften von SM-Pubsets und Volume-Sets

Im Folgenden sind einige wichtige und für das Verständnis von SMPGEN nötige Eigenschaften und Verhaltensweisen von SM-Pubsets und Volume-Sets aufgeführt. Einzelheiten zu SM-Pubsets sind den Handbüchern „System Managed Storage“ [8] und „Systembetreuung“ [5] zu entnehmen.

Pubset-Eigenschaften

SM- und SF-Pubsets haben folgende gemeinsame Eigenschaften:

- Sie werden i.A. als Ganzes in und außer Betrieb genommen.
- Sie bilden einen Namensraum für Dateien, Jobvariablen und Guards.
- Der Ablageort einer Datei innerhalb des Pubsets wird vom System bestimmt (außer bei „physikalischer“ Allokierung, für die der Anwender besonders berechtigt sein muss).
- Sie können als „große Pubsets“ angelegt bzw. in solche konvertiert werden. Große Pubsets unterstützen Dateien und Volumes mit einer Größe von mehr als 32 GB.
- Es gibt benutzerbezogene, pubset-weit gültige Speicher-, Datei- und Jobvariablen-Kontingente.
- Es gibt benutzerbezogene, pubset-weit gültige Berechtigungen (Recht auf besonders performante Dateibearbeitung, PUBLIC-SPACE-EXCESS, Recht auf physikalische Allokierung).

SM-Pubsets sind jedoch im Gegensatz zu SF-Pubsets nicht strukturlos bis auf die Unterscheidung von Pubres und anderen Platten. Innerhalb eines SM-Pubsets gibt es i.A. mehrere Mengen zusammengehöriger Platten mit ähnlichen VSNs, die Volume-Sets genannt werden.
Ein SM-Pubset kann aus max. 255 Volume-Sets bestehen.

Falls einer dieser Volume-Sets ausfällt, kann er aus dem SM-Pubset herauskonfiguriert und der SM-Pubset ohne ihn weiterbetrieben werden. Die Dateien, die in diesem Volume-Set lagen, können gezielt aus bestehenden Sicherungen wiedergeholt werden.
Ausnahme: Fällt der Control-Volume-Set aus, kann der SM-Pubset nicht weiterbetrieben werden.

Volume-Set-Eigenschaften

Einige Eigenschaften, die für ein SF-Pubset pubset-weit gelten, gelten in einem SM-Pubset nur für die einzelnen Volume-Sets. Solche Eigenschaften sind u.A.:

- der Volume-Set-Identifier bildet, vollkommen analog zur SF-Pubset-Kennung, einen Teil der VSN der zugehörigen Platten
- die Allokierungseinheit ist auf allen Platten einheitlich
- das Plattenformat (K, NK2, NK4) stimmt überein
- die Cache-Konfiguration ist Volume-Set-bezogen
- alle Extents einer Datei liegen innerhalb eines Volume-Sets

Auf Grund der Aufteilung in mehrere Volume-Sets kann ein SM-Pubset - und damit ein Datei-Namensraum - verschiedene Services unterstützen und damit in größerem Umfang als bisher geeignete Ablagemöglichkeiten für unterschiedliche Dateien bieten.

Da die Volume-Sets für die Ablage einer bestimmten Datei unterschiedlich gut geeignet sind, sind ihnen logische Volume-Set-Eigenschaften zugeordnet:

Ausfallsicherheit (AVAILABILITY), Performance (PERFORMANCE), Datenkonsistenz (WRITE-CONSISTENCY). Diese werden vom System herangezogen, um in Verbindung mit den logischen Dateieigenschaften den geeignetesten Ablageort für eine Datei zu bestimmen.

Ein Volume-Set des SM-Pubsets, der Control-Volume-Set, wird zur Ablage der wichtigsten Meta-Informationen benutzt (z.B. Volume-Set-Konfiguration, Benutzereinträge, Dateikatalog-Verweise, Schutzprofile). Er kann daher auch im Fehlerfall nicht aus dem SM-Pubset entfernt werden.

Nach ihrem Gebrauch unterscheiden sich die ein SM-Pubset bildenden Volume-Sets in USAGE=*STD, USAGE=*WORK und USAGE=*HSMS-CONTROLLED-Volume-Sets. USAGE=*STD-Volume-Sets dienen der normalen Dateiablage, USAGE=*WORK-Volume-Sets der Ablage von Work-Dateien, und USAGE=*HSMS-CONTROLLED-Volume-Sets werden in HSMS-unterstützten SM-Pubsets von HSMS als Hintergrundebene verwendet.

SMPGEN unterstützt USAGE=*STD und USAGE=*HSMS-CONTROLLED Volume-Sets. Volume-Sets mit USAGE=*WORK müssen durch Rekonfigurationskommandos hinzugenommen werden.

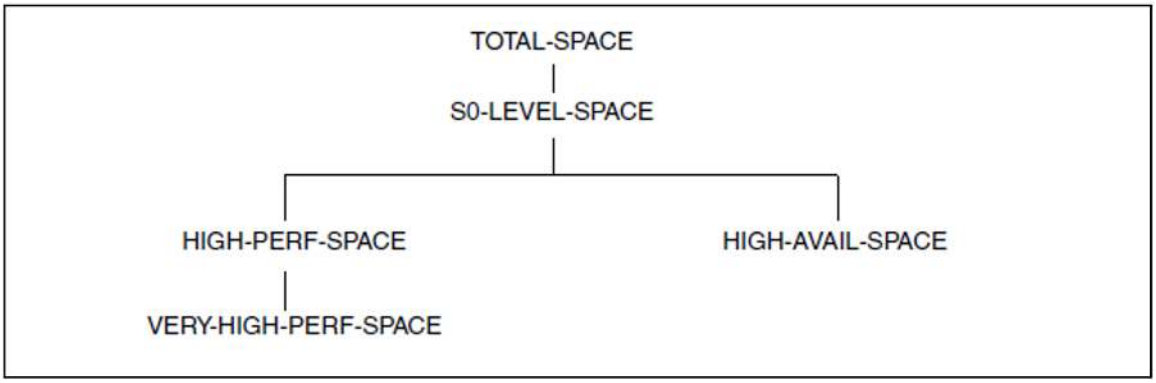
Speicherplatz-Kontingente

Die nachstehend beschriebenen Kontingente beziehen sich jeweils auf eine Benutzerkennung und einen SM-Pubset; es gibt keine Aufschlüsselung auf einzelne Volume-Sets.
Auf einem SM-Pubset ist die Speicherplatz-Kontingentierung etwas komplexer als auf einem SF-Pubset. Es existieren nebeneinander drei Kontingentstrukturen:

- Kontingente für permanente Dateien(das sind alle Dateien, die keine Work-Dateien und nicht temporär sind)
- Kontingente für temporäre Dateien
- Kontingente für Work-Dateien (Arbeitsdateien)

„Kontingentstruktur“ besagt, dass es für diese drei Arten von Dateien jeweils ein Speicherplatz-Kontingent gibt, das den gesamten von diesen Dateien belegbaren Speicherplatz bestimmt. Zusätzlich existieren sog. Unterkontingente, mit denen die Belegung besonders wertvoller Ressourcen weiter eingeschränkt werden kann.

Für permanente Dateien sieht die Kontingentstruktur folgendermaßen aus:

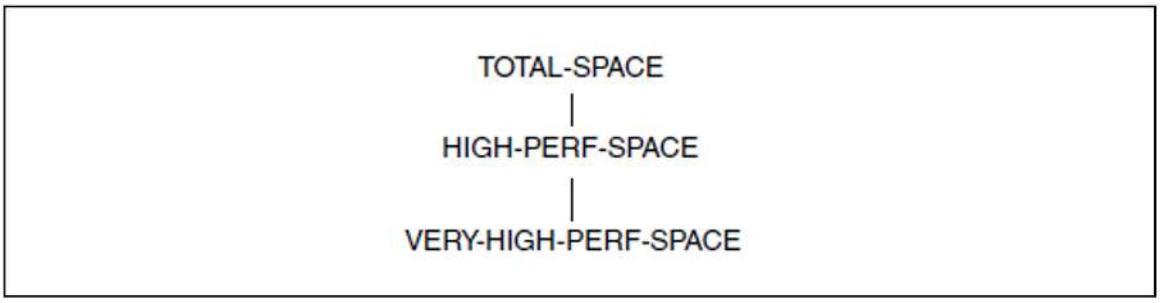


Dabei haben die Kontingent-Namen folgende Bedeutung:

- TOTAL-SPACE(PERM)
  ist der Platz für alle auf dem Pubset liegenden permanenten Dateien, auch diejenigen auf Hintergrundebenen. Liegt die Datei auf einer Hintergrundebene, so wird für sie jedoch nicht der real belegte Speicherplatz berechnet, sondern der Platz, den sie bei einem Recall auf der S0-Ebene belegen würde.

- S0-LEVEL-SPACE(PERM)
  ist der Platz, den permanente Dateien auf der S0-Ebene des Pubsets belegen. Dies entspricht dem Kontingent PUBLIC-SPACE-LIMIT auf SF-Pubsets.
- HIGH-PERF-SPACE(PERM)
  ist der Platz, den permanente Dateien mit erhöhtem Performance-Attribut (PERFORMANCE=*HIGH oder *VERY-HIGH) auf der S0-Ebene des Pubsets belegen.
- VERY-HIGH-PERF-SPACE(PERM)
  ist der Platz, den permanente Dateien mit dem Attribut PERFORMANCE=*VERY-HIGH auf der S0-Ebene des Pubsets belegen.
- HIGH-AVAILABILITY-SPACE(PERM)
  ist der Platz, den permanente Dateien mit erhöhtem Verfügbarkeits-Attribut (AVAILABILITY=*HIGH) auf der S0-Ebene des Pubsets belegen.

Für temporäre Dateien und Work-Dateien sieht die Kontingentstruktur folgendermaßen aus:



Die Bedeutung der Kontingente stimmt mit derjenigen bei permanenten Dateien überein.

Dass es sich um Kontingente für temporäre Dateien oder Work-Dateien handelt, wird durch den Suffix <kontingentname>(TEMP) bzw. <kontingentname>(WORK) angezeigt.
Die Kontingentierung von S0-LEVEL-SPACE und HIGH-AVAIL-SPACE erübrigt sich, da nur permanente Dateien migriert bzw. mit dem Attribut AVAILABILITY=*HIGH versehen werden können.
Das Kontingent TOTAL-SPACE(TEMP) entspricht dem Kontingent TEMP-SPACE-LIMIT auf SF-Pubsets.

Die Speicherplatz-Belegungszähler in den Benutzereinträgen entsprechen der Kontingentstruktur.

Auf Gruppenebene ist die Kontingentstruktur einfacher:
Es existieren lediglich die bereits von SF-Pubsets bekannten Werte TEMP-SPACE-LIMIT und PUBLIC-SPACE-LIMIT sowie die neue Schranke WORK-SPACE-LIMIT.
(Die oben genannten Limits bilden Schranken für die Möglichkeit des Gruppenverwalters, einem Mitglied seiner Gruppe entsprechende Kontingente zuzuteilen; sie sind nicht als Kontingente für die Menge aller Gruppenmitglieder zu verstehen).

15 SPCCNTRL Prüfen und Verwalten von Platzzuweisungen auf Platten

Versionsstand: SPCCNTRL V21.0C

Das Programm SPCCNTRL (SPaCe CoNTRoL) dient zur Kontrolle von Platzzuweisungen auf Platten.

Das Programm steht sowohl der Systemverwaltung als auch dem nichtprivilegierten Benutzer zur Verfügung. Einige Funktionen sind allerdings nur der Systemverwaltung vorbehalten (siehe bei den einzelnen Anweisungen).

In einigen Fällen können statt der SPCCNTRL-Programmfunktionen alternative Funktionen genutzt werden:

- Über File-Locks informiert `/SHOW-FILE-LOCK`.
- Zur Diagnose können bestimmte Katalogeinträge mit `/LIST-CATALOG-ENTRY` in eine Datei ausgegeben werden.
- Über Katalogdateien und ihren Füllungsgrad informiert `/SHOW-PUBSET-CATALOG-ALLOCATION.`
- Dateikataloge werden im laufenden Betrieb bei Bedarf automatisch erweitert.

15.1 Bedienung

Falls die Meldungsdatei SYSMES.SPCCNTRL.<ver> noch nicht zugewiesen wurde, kann die Meldungsdatei auch nachträglich mit der SPCCNTRL-Anweisung MODIFY zugeordnet werden.

Das Programm wird gestartet mit `/START-SPCCNTRL.`

START-SPCCNTRL	Alias: **SPCCNTRL**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

Alternativ ist der Aufruf wie folgt möglich:

```
/START-EXECUTABLE-PROGRAM FROM-FILE=$SPCCNTRL
```

Nach Programmaufruf können die Steueranweisungen des Programms eingegeben werden.

15.2 Anweisungen

- Übersicht über die Anweisungen von SPCCNTRL
- Beschreibung der Anweisungen
  - BKPT - Das Programm SPCCNTRL unterbrechen
  - CHECK - Durchführen von Belegungsprüfungen
  - DISPLAY - Ausgeben über SYSOUT
  - END - SPCCNTRL beenden
  - HELP - Beschreibung über SPCCNTRL-Anweisungen ausgeben
  - LIST - Ausgeben über SYSLST
  - MODIFY - Standardmäßig implementierte Voreinstellungen verändern
  - PURGE - Entfernen von totem Platz und Löschen von Katalogeinträgen
  - TRACE - Identifizieren und Ausgeben von Blöcken und Einträgen im Systemkatalog und VTOC-Bereich einer Privatplatte

15.2.1 Übersicht über die Anweisungen von SPCCNTRL

Anweisung	Bedeutung
BKPT	Programm SPCCNTRL unterbrechen.
CHECK	Durchführen von Belegungsprüfungen einer gemeinschaftlichen Platte und Prüfungen auf fehlerhafte Einträge im Systemkatalog oder im VTOC-Bereich einer Privatplatte.
DISPLAY	Ausgaben sollen über SYSOUT erfolgen, d.h. im Dialog erscheinen sie an der Datensichtstation.
END	Programm SPCCNTRL beenden.
HELP	Beschreibung einer bestimmten Anweisung oder Überblick über alle SPCCNTRL-Anweisungen ausgeben lassen.
LIST	Ausgaben sollen über SYSLST erfolgen.
MODIFY	Bei SPCCNTRL standardmäßig implementierte Voreinstellungen verändern.
PURGE	Entfernen von totem Platz auf gemeinschaftlichen Platten und Löschen von Katalogeinträgen aus dem Systemkatalog und dem VTOC-Bereich einer privaten Platte.
TRACE	Identifizieren und Ausgeben von Blöcken und Einträgen im Systemkatalog und VTOC-Bereich einer Privatplatte oder Ausgeben eines beliebigen Blockes einer Platte.

15.2.2 Beschreibung der Anweisungen

In den nachfolgend aufgeführten Funktionen kann filename (falls nichts anderes angegeben ist) nur vollqualifiziert angegeben werden. In einigen Fällen ist auch die Angabe einer coded File-Id in der Form X'hhhhhhhh' möglich. Bei fehlender Angabe „type“ wird angenommen, dass die spezifizierte Platte online verfügbar ist.

SPCCNTRL interpretiert jede Eingabe, die keine SPCCNTRL-Anweisung ist, als BS2000-Kommando. Es können so im Programmodus auch Systemkommandos eingegeben und verarbeitet werden.

15.2.2.1 BKPT - Das Programm SPCCNTRL unterbrechen

Mit der Anweisung BKPT wird das Programm im Dialog unterbrochen, im Batch-Betrieb kommt es zum Wartezustand.

Format

BKPT

Diese Anweisung besitzt keine Operanden.

- An Stelle der Anweisung kann in Spalte 1 ein Schrägstrich (/) eingegeben werden.
- Wird nach dem Schrägstrich noch ein Systemkommando oder eine SPCCNTRL-Anweisung eingegeben, erfolgt zuerst die Ausführung des Kommandos oder der Anweisung und dann erst die Programmunterbrechung.
- Die Anweisung BKPT und der Schrägstrich sind nur im Dialog sinnvoll. Im Batch-Betrieb leiten sie einen Wartezustand ein, der erst endet, wenn dem Batch-Auftrag mit /INFORM-PROGRAM eine Nachricht geschickt wird. Das Eintreffen der Nachricht entspricht dem Systemkommando EOF, mit dem das Einlesen von Daten aus SYSDTA beendet wird. Dadurch beendet sich auch SPCCNTRL. Eine Rückkehr in den Programm-Modus ist nicht möglich.

15.2.2.2 CHECK - Durchführen von Belegungsprüfungen

Mit der CHECK-Anweisung können folgende Prüfungen durchgeführt werden:

- Belegungsprüfungen einer gemeinschaftlichen Platte
- Prüfungen auf fehlerhafte Einträge im Systemkatalog
- Prüfungen des Inhalts vom VTOC-Bereich (F1-Kennsatz) einer Privatplatte auf fehlerhafte Einträge

Für jeden gefundenen Fehler wird eine Meldung ausgegeben, die die fehlerhafte Seite, die Benutzerkennung und den Dateinamen enthält. Danach wird mit der Überprüfung fortgefahren.

Die Anweisung ist der Systemverwaltung vorbehalten.

Es werden sowohl importierte, als auch nicht importierte Kataloge unterstützt.

Format

CHECK / C

ALLOCATION,vsn

CATALOG

[, { CATID = { cat-id [, { VSID = vsid / SEL= { ALL / FILE / PRIV / MIG / JV } }] /

vsn[,type] /

VSID = vsid }]

[,ENTRY = filename / (filename,JV)]

VTOC,vsn[,type] [,ENTRY = filename]

Operanden

A[LLOCATION]
Ein gemeinschaftlicher Datenträger soll auf allgemeine Belegungsverteilung geprüft werden, d.h. es soll kontrolliert werden, ob tote Bereiche und/oder Doppelbelegungen vorliegen.

Es wird eine Übersicht über die Belegung der Platte ausgegeben. Die Prüfung wird abgebrochen, wenn Doppelbelegungen entdeckt werden.
Diese können dann mit LIST ALLOCATION lokalisiert werden.
Die Angabe einer Katalogkennung ist nicht erforderlich.

vsn
Bezeichnet die gemeinschaftliche Platte, die überprüft werden soll.

C[ATALOG]
Es soll der Systemkatalog auf fehlerhafte Einträge überprüft werden. Es werden Fehlercodes vergeben, die mit HELP abgefragt werden können.

CATID = catid
bezeichnet

1. den TSOSCAT eines importierten SF-Pubsets.
2. den Gesamtkatalog eines SM-Pubsets.
   Wird kein zusätzlicher Parameter angegeben, erfolgt die Ausgabe der Information für sämtliche Teilkataloge des SM-Pubsets.

   VSID = vsid
   Die Angabe einer VSID beschränkt die Ausgabe auf den Dateikatalog des selektierten Volume-Sets (nur bei SM-Pubsets zulässig).

   SEL = ALL / FILE / PRIV / MIG / JV
   Zusätzliche Auswahl (nur bei SM-Pubsets).

   SEL = ALL
   bearbeitet alle Teilkataloge des ausgewählten SM-Pubsets (analog zu kein zusätzlicher Parameter). Es werden bei den Spezialkatalogen alle Katalogdateien bearbeitet.

   SEL = FILE
   selektiert sämtliche Dateikataloge des SM-Pubsets.

   SEL = PRIV
   bewirkt die Bearbeitung des Spezialkataloges für Privatplatten- und Banddateieinträge. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#P00) bearbeitet.

   SEL = MIG
   wählt den Spezialkatalog für migrierte Dateien bzw. Dateien, für die noch kein Platz zugewiesen wurde, aus. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#M00) bearbeitet.

   SEL = JV
   ermöglicht die Ausgabe von Informationen über den Spezialkatalog für Jobvariable. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#J00) bearbeitet.

vsn
bezeichnet eine Pubres, die entweder den Dateikatalog eines Volume-Sets, oder den TSOSCAT eines SF-Pubsets enthält.

type
bezeichnet den Gerätetyp der durch <vsn> bezeichneten Platte.

VSID = vsid
bezeichnet den Dateikatalog eines Volume-Sets.
Dieser Parameter ist nur bei der Bearbeitung eines SM-Pubsets zulässig.

ENTRY = filename
Bezeichnet einen vollqualifizierten Dateinamen, der die Prüfung auf den entsprechenden Katalogeintrag beschränkt.

ENTRY = (filename,JV)
Die Prüfung soll sich auf die mit filename bezeichnete Jobvariable beschränken. Die Angabe filename kann auch als coded File-Id gemacht werden.

V[TOC]
Der Inhalt des F1-Kennsatzes einer privaten Platte soll auf fehlerhafte Einträge überprüft werden. Die Funktion entspricht der Funktion CHECK CATALOG.

vsn[,type]
Bezeichnet die private Platte, deren VTOC-Bereich geprüft werden soll. Ist die Platte nicht verfügbar (online), muss auch der Plattentyp „type“ angegeben werden.

ENTRY = filename
Beschränkt die Prüfung auf einen bestimmten Dateieintrag mit dem Dateinamen „„filename“. Die Angabe einer coded File-Id ist möglich.

Beispiele

1. Ausgabe von Informationen zur gemeinschaftlichen Platte mit folgender Anweisung:

```
*check allocation,1SBZ.0
 ASSIGNMENT SUMMARY FOR VOLUME 1SBZ.0
 -------------------------------------
 NUMBER OF FILE ENTRIES:             902
 NUMBER OF EXTENTS:                 1369

 PAM PAGES ASSIGNED TO FILES:     216486
 PAM PAGES NOT ASSIGNED:          115185
 PAM PAGES FREE:                       0
*
```

2. Ausgabe von Informationen zur Katalogkennung:

```
*CHECK CATALOG,CATID=1SBZ
%  SPC0060 NO ERROR IN THE CATALOG FILE 1SBZ:$TSOS.TSOSCAT
*
```

oder

```
*CHECK CATALOG,CATID=A
ERROR SUMMARY FOR CATALOG :A:$TSOSCAT

ERROR    |    BLOCK  |  BYTE |   USER   | FILENAME
-------------------------------------------------
SPC1506  |     1635  |   584 | TEXACO   | DATOUT
SPC1503  |     1800  |   403 | ULTIMO   | PROC03
```

`/HELP-MSG-INFORMATION SPC1506` informiert über die Fehlerursache:

```
% SPC1506 VSN IN VOLUME TABLE ENTRY NOT PRINTABLE
```

15.2.2.3 DISPLAY - Ausgeben über SYSOUT

Mit der DISPLAY-Anweisung werden die in den Operanden angeforderten Informationen über SYSOUT ausgegeben. Im Dialog werden sie an der Datensichtstation gezeigt.

Format

DISPLAY / D
SPACE , { **P**UBLIC [,CATID = { cat-id / vsn[,type }] / **PR**IVATE[,vsn[,type]] / **E**AM [, { TSN = { tsn / (ALL) } / USER = userid / CATID = cat-id }] / **C**ATALOG [, { CATID = cat-id [, { VSID = vsid / SEL = { ALL / FILE / PRIV / MIG / JV } }] / vsn[,typ] / VSID = vsid }] }
CATALOG ,ENTRY = filename / (filename,JV) [,CATID = catid] [,HEX = **N**O / **Y**ES]
VTOC ,VSN[,type],ENTRY = filename [,HEX = **N**O / **Y**ES]

Operanden

DISPLAY SPACE
Dieser Operand ist der Benutzerkennung TSOS vorbehalten.

PUBLIC
Es soll der Umfang von freiem und belegtem Speicherplatz auf gemeinschaftlichen, importierten Datenträgern des Home-Pubsets ausgegeben werden. SM-Pubsets werden nicht unterstützt. Informationen über die Speicherplatzbelegung von SM-Pubsets liefert `/SHOW-PUBSET-SPACE-ALLOCATION`, siehe Handbuch „Kommandos“ [1].

Ausgegeben werden neben VSN und Gerätetyp auch die Anzahl der freien Seiten, die Anzahl der belegten Seiten und die Anzahl der Dateien der angegebenen Platte. Soll die Information über mehr als eine Platte geliefert werden, gibt SPCCNTRL nach der Auflistung der Werte der einzelnen Platten auch eine Summenzeile aus. Falls weder CATID noch VSN angegeben wurde, wird die Funktion für alle Platten des Home-Pubsets durchgeführt.

vsn
Bezeichnet eine spezielle gemeinschaftliche Platte, über deren Umfang der Platzbelegung Informationen gewünscht werden.

type
Bezeichnet den Plattentyp

CATID = catid
catid bezeichnet die Katalog-ID eines importierten SF-Pubsets. Es werden Informationen über die Platzbelegung von Platten, die in diesem Katalog eingetragen sind, ausgegeben. Ohne Angabe dieses Operanden ist immer der Home-Katalog gemeint. Im Fall eines inhomogenen Verbundes werden Informationen lediglich für den Master ausgegeben.

PRIVATE
Es soll der Umfang von freiem und belegtem Platz auf allen privaten Platten ausgegeben werden (inklusive Labels)

vsn
Bezeichnet eine spezielle private Platte, über die Informationen gewünscht werden.

type
Bezeichnet den Plattentyp (z.B. D3435). Diese Angabe ist erforderlich, wenn die private Platte nicht online ist.

EAM
Es sollen Informationen über die Systemdatei SYSEAM ausgegeben werden. Ohne Angabe einer tsn oder von ALL erhält man einen Überblick über die Gesamtbelegung von SYSEAM. Ist eine tsn angegeben, beschränken sich die Informationen über die Belegung auf die betreffende tsn.

TSN = tsn / (ALL)
Die Informationen über die Systemdatei SYSEAM sollen sich nur auf die bezeichnete tsn oder auf alle tsns (ALL) beziehen.

USER = userid
Die Informationen über die Systemdatei SYSEAM sollen sich auf die Kennung userid beziehen.

CATID = catid
Ist die Katalogkennung des gewünschten Katalogs. Standardwert ist bei fehlender Angabe des Operanden der Home-Katalog.

CATALOG
Diese Funktion gibt Informationen über die Gesamtbelegung einer oder mehrerer (bei SM-Pubsets) Katalogdatei(en) aus.

Es werden die Anzahl der Benutzerkennungen, die Größe der Katalogdatei, die Anzahl der belegten Seiten (laut Zähler und laut BITMAP) sowie ein Fehlerindikator ausgegeben. Bei den Angaben handelt es sich um interne Verwaltungsdaten des Katalogverwaltungssystems (CMS).

Es werden sowohl importierte als auch nicht-importierte Kataloge unterstützt.

> Für Dateien, die sich auf LARGE-OBJECTS-Pubsets befinden, ist diese Funktion der Systemverwaltung vorbehalten. Alle sonstigen Benutzer werden mit der Meldung `DMS0576` abgewiesen.

CATID = catid
bezeichnet

1. den TSOSCAT eines importierten SF-Pubsets.

2. den Gesamtkatalog eines SM-Pubsets.
   Wird kein zusätzlicher Parameter angegeben, erfolgt die Ausgabe der Information für sämtliche Teilkataloge des SM-Pubsets.

   VSID = vsid
   Die Angabe einer VSID beschränkt die Ausgabe auf den Dateikatalog des selektierten Volume-Sets (nur bei SM-Pubsets zulässig).

   SEL = ALL / FILE / PRIV / MIG / JV
   Zusätzliche Auswahl (nur bei SM-Pubsets). Es werden bei den Spezialkatalogen alle Katalogdateien bearbeitet.

   SEL = ALL
   bearbeitet alle Teilkataloge des ausgewählten SM-Pubsets (analog zu kein zusätzlicher Parameter).

   SEL = FILE
   selektiert sämtliche Dateikataloge des SM-Pubsets.

   SEL = PRIV
   bewirkt die Bearbeitung des Spezialkataloges für Privatplatten- und Banddateieinträge. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#P00) bearbeitet.

   SEL = MIG
   wählt den Spezialkatalog für migrierte Dateien bzw. Dateien, für die noch kein Platz zugewiesen wurde, aus. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#M00) bearbeitet.

   SEL = JV
   ermöglicht die Ausgabe von Informationen über den Spezialkatalog für Jobvariable. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#J00) bearbeitet.

vsn
bezeichnet eine Platte, die entweder den Dateikatalog eines Volume-Sets oder den TSOSCAT eines SF-Pubsets enthält.

type
bezeichnet den Gerätetyp der durch <vsn> bezeichneten Platte.

VSID = vsid
bezeichnet den Dateikatalog eines Volume-Sets.
Dieser Parameter ist nur bei der Bearbeitung eines SM-Pubsets zulässig.

DISPLAY CATALOG
Dieser Operand steht jedem Benutzer zur Verfügung.

> ¡ Wird durch den Unteroperanden ENTRY eine Datei > 32 GB oder eine Datei auf einem Volume > 32 GB angegeben und besitzt der Aufrufer nicht die Kennung TSOS, so werden in der Ausgabe ungültige Werte geliefert, die als Null erkennbar sind (für die Datei bei Total Space Allocated und für die Extents bei Low, High, Pages Allocated).

Es werden Informationen aus dem Katalogeintrag einer Datei oder einer Jobvariablen nach SYSOUT ausgegeben, die `/SHOW-FILE-ATTRIBUTES` nicht liefert.

Es werden u.a. die Größe der Datei oder Jobvariablen, die Größe des Katalogeintrages, die CFID oder JVID, Primary Block Number, eventuelle Locks und die Anzahl der Extents ausgegeben.

Die Primary Block Number wird bei der Eingabe der Anweisung TRACE CATALOG, block (Primary Block Number), byte benötigt. Außerdem wird der Name des entsprechenden Kataloges/Teilkataloges angezeigt.

Der nicht-privilegierte Anwender kann nur Dateien bzw. Jobvariablen der eigenen Benutzerkennung ansprechen. DISPLAY CATALOG unterstützt nur Dateien importierter Kataloge.

ENTRY = filename
Für die mit filename bezeichnete Datei wird die Ausgabe des Katalogeintrags gewünscht. filename kann auch als coded File-Id in der Form X'hhhhhhhh' angegeben werden.

ENTRY = (filename,JV)
Für die mit filename bezeichnete Jobvariable wird die Ausgabe des Katalogs gewünscht. filename kann auch als JV-ID oder als coded File-Id in der Form X'hhhhhhhh' angegeben werden.

CATID = catid
Ist die Katalogkennung des gewünschten Katalogs (SF-Pubset oder SM-Pubset). Standardwert ist die Defaultkennung der Benutzerkennung des Aufrufers.

HEX = NO / YES
gibt an, ob der gesamte Katalogeintrag sedezimal ausgegeben werden soll. NO ist Standardwert.

DISPLAY VTOC
Dieser Operand steht jedem Benutzer zur Verfügung.

Es wird die Ausgabe der im F1-Kennsatz enthaltenen Informationen eines privaten Datenträgers gewünscht. Es werden die gleichen Informationen geliefert wie bei Angabe von DISPLAY CATALOG, als Quelle dient jedoch der VTOC der Privatplatte.

vsn[,type]
Bezeichnet den Datenträger, aus dessen VTOC-Bereich Informationen gewünscht werden. Ist die Platte nicht online, muss auch der Plattentyp „type“ angegeben werden.

ENTRY = filename
Dateieintrag im F1-Kennsatz, über den Informationen ausgegeben werden sollen. filename kann auch als coded File-Id in der Form X'hhhhhhhh' angegeben werden.

HEX = NO / YES
gibt an, ob der gesamte Eintrag sedezimal ausgegeben werden soll. NO ist Standardwert.

Beispiele

1. Ausgabe von DISPLAY SPACE, PUBLIC

```
DISPLAY S,PUBLIC
VOLUME I DEVICE I PAGES FREE I PAGES USED I FILES
-------I--------I------------I------------I--------
2SBZ.0 I  D3435 I    157758  I    450948  I  1660
2SBZ.1 I  D3435 I    152823  I    455886  I  1655
2SBZ.2 I  D3435 I      3408  I    328266  I  1656
-------------------------------------------------
TOTAL           I    313989  I   1235100  I  4971
```

2. Ausgabe von DISPLAY SPACE, EAM

```
D S,EAM
SUMMARY FOR :K:$TSOS.SYSEAM:

SYSEAM SPACE ASSIGNED:              348 PAGES
SYSEAM SPACE ALLOCATED:           38712 PAGES
SYSEAM CATALOG ENTRY SIZE:          468 BYTES
```

3. Ausgabe von DISPLAY SPACE, EAM, TSN=(ALL)

```
D S,EAM,TSN=(ALL)
CATID I TSN  I USER     I TYPE I FILES I PAGES USED I ALLOCATED
------I------I----------I------I-------I------------I----------
 K    I 5015 I TSOS     I   3  I    2  I         6  I      6
 K    I 5060 I A2A3A4A5 I   3  I    5  I         6  I      6
 K    I 5068 I RZDATEX  I   3  I    1  I        32  I     33
 K    I 5158 I TERM     I   3  I    1  I         0  I      0
 K    I 5167 I DATA     I   3  I   40  I         1  I      3
 K    I 5208 I BASIS    I   3  I    3  I         3  I      6
 K    I 5232 I ARCHIV   I   3  I    3  I         0  I      0
 K    I 5233 I CONS     I   3  I    2  I        62  I     63
 K    I 5239 I DELTA    I   3  I    6  I        23  I     24
 K    I 5328 I F1F2F3F4 I   3  I    1  I         5  I      6
 K    I 5346 I BERGER   I   3  I   17  I         4  I      6
 K    I 5352 I FIFO     I   3  I    5  I        45  I     45
```

4. Ausgabe von DISPLAY SPACE, CATALOG

```
 D S,C
 TSOSCAT/HOME
-------------------------------------------------------------------------
 USERID' S IN JOINFILE              69
 EFFECTIVE FILE SIZE              2049
 PAGES USED (4K)    (COUNTER)      279
 PAGES USED (4K)    (BIT TABLE)    279
 PREASSIGNED (ERROR IF NOT 0)        0
```

5. Ausgabe von DISPLAY CATALOG, ENTRY=TEMP.12

```
D C,ENTRY=$TSOS.TEMP.12
PUBLIC DISK FILE:   :2BV:$TSOS.TEMP.12
------------------------------------------------------------------------
TOTAL SPACE ALLOCATED:               3 PAGES
CODED FILE IDENTIFICATION:     8D32C3B4
*   FLT LOCK ENTRY:                  00000000 00000000 00000001 00000000
*                                    00000000 00000000 00000000 00000000
*                                    00000000 00000000 00000000 00000000
*                                    00000000 00000000 00000000 00000000
*                                    00000000 00000000 00000000 00000000
*                                    00000000 00000000 00000000 00000000
PRIMARY BLOCK NUMBER (BLK=4KB):      16
CURRENT BLOCK NUMBER:                16 CATALOG ENTRY AT BYTE:     3665
CATALOG ENTRY SIZE:                 121 BYTES
# OF EXTENT ENTRIES:                  1
EXTENT I  VOLUME  I       LOW       I      HIGH      I   PAGES ALLOCATED
-------I----------I-----------------I----------------I------------------
   1   I  2BV.01  I        6679     I        6681    I           3
```

* Diese Ausgaben sind an das TSOS-Privileg gebunden.

Bedeutung der Ausgabefelder bei FLT LOCK ENTRY:

jeweils von links nach rechts zeilenweise, Wertangabe in hexadezimaler Schreibweise

Anzahl der Öffner mit OPEN=INPUT,SHARUPD=WEAK
Anzahl der Öffner mit OPEN=INPUT,SHARUPD=NO
Anzahl der Öffner mit OPEN=INPUT,SHARUPD=YES
Anzahl der Öffner mit DIV, MODE=*INPUT,SHARUPD=YES

Anzahl der Öffner mit DIV, MODE=*INPUT,SHARUPD=WEAK
Anzahl der Input-Reservierungslocks für File-Transfer
Anzahl der Input-Reservierungs-Locks für SPOOL (print locks)
Anzahl der Öffner mit OPEN=OUTPUT,SHARUPD=YES

Anzahl der Öffner mit DIV, MODE=*OUTPUT,SHARUPD=YES
Anzahl der Öffner mit DIV, MODE=*OUTPUT,SHARUPD=WEAK
Anzahl der Öffner mit OPEN=OUTPUT,SHARUPD=NO
Anzahl der Write exklusiven Reservierungen

Anzahl der Öffner mit OPEN=OUTPUT,SHARUPD=WEAK
Anzahl der Secure Locks
Anzahl der Output-Reservierungs-Locks für File-Transfer
Anzahl der HSMS Secure Locks

Anzahl der Concurrent-Copy-Locks
Anzahl der Concurrent-Copy-Locks für den Aufbau der Concurrent-Copy-Session
Anzahl der exklusiven Pfad-Locks für Dateien auf Privatplatten
Anzahl der 0-Locks

reserviert für interne Verwendung
reserviert für interne Verwendung
Anzahl der Host-Environment-Locks
Anzahl der XCS-Environment-Locks

Die Anzeige der File-Locks kann auch mit `/SHOW-FILE-LOCKS` erreicht werden.

6. Ausgabe von DISPLAY VTOC, RZWORK, ENTRY=SPC.TEST.PR

```
D V,RZWORK,ENTRY=SPC.TEST.PR
PRIVATE DISK FILE: :HOME:$TSOS.SPC.TEST.PR
TOTAL SPACE ALLOCATED:              3 PAGES
CODED FILE IDENTIFICATION:    92032342
FLT LOCK ENTRY:                   NONE
PAM LOCK ENTRY:                   NONE
VTOC LOCK ENTRY:                  NONE
VTOC SHARER ENTRIES:              NONE
PRIMARY BLOCK NUMBER:               2
CURRENT BLOCK NUMBER:            2426        CATALOG ENTRY AT BYTE:
VTOC/F1 ENTRY ON VOLUME:        RZWORK       ON-LINE
VTOC/F1 BLOCK POINTER:              10
VTOC/F1 ENTRY IN BLOCK:             10       CATALOG ENTRY AT BYTE:   1874
CATALOG ENTRY SIZE:                125 BYTES
# OF EXTENT ENTRIES:                  1
EXTENT I  VOLUME  I       LOW       I      HIGH      I   PAGES ALLOCATED
-------I----------I-----------------I----------------I------------------
   1   I  RZWORK  I        2878     I        2880    I           3
```

15.2.2.4 END - SPCCNTRL beenden

Mit der Anweisung END wird SPCCNTRL beendet.

Format

END

Diese Anweisung besitzt keine Operanden.

> Wird bei der Eingabe von E noch ein Operand eingegeben, nimmt SPCCNTRL an, es handle sich um `/ENTER-JOB` und führt dieses Kommando aus.

15.2.2.5 HELP - Beschreibung über SPCCNTRL-Anweisungen ausgeben

Mit der Anweisung HELP kann eine Beschreibung über eine bestimmte Anweisung oder ein Überlick über alle SPCCNTRL-Anweisungen ausgegeben werden.

Format

HELP
[stmt[,mode]]

Operanden

stmt
Bezeichnet den Namen einer SPCCNTRL-Anweisung. Es soll eine Beschreibung zu dieser Anweisung ausgegeben werden. Für stmt kann angegeben werden:

```
HELP
BKPT
END
MODIFY
DISPLAY
LIST
CHECK
PURGE
TRACE
```

Bei fehlender Angabe stmt wird eine Übersicht über alle Anweisungen ausgegeben.

mode
Bezeichnet eins der folgenden Schlüsselwörter der mit stmt bezeichneten Anweisung:

```
A[LLOCATION]
S[PACE]
C[ATALOG]
V[TOC]
```

Es wird eine Beschreibung der angegebenen Funktion ausgegeben. Fehlt der Operand, so wird eine Übersicht über die verfügbaren Funktionsmodi der Anweisung ausgegeben.

Wurde die Anweisung voll mit HELP ausgeschrieben und statt einer SPCCNTRL-Anweisung im stmt-Operanden ein unzulässiger Operand angegeben, so wird `/HELP-MSG-INFORMATION` ausgeführt.

15.2.2.6 LIST - Ausgeben über SYSLST

Mit der LIST-Anweisung können Informationen über SYSLST ausgegeben werden. Das sind Informationen, die zu umfangreich für die Ausgabe über den Bildschirm der Datenstation sind.

Die Anweisung ist der Benutzerkennung TSOS vorbehalten.

Format

LIST / L
ALLOCATION,vsn[,type] [USER = userid] **C**ATALOG [, { CATID = cat-id [,{ VSID = vsid / SEL = { ALL / FILE / PRIV / MIG / JV } }] / vsn[,type] / VSID = vsid }] [,USER = userid] **V**TOC,vsn[,type] [,ENTRY = filename]

Operanden

ALLOCATION
Es soll die allgemeine Belegungsverteilung auf einem Datenträger ausgegeben werden.

Die Liste weist die Belegung sortiert nach Extents aus. Der Beginn jedes Extents ist außerdem in Form von cchhr angegeben.

Es können nur Informationen über einen Datenträger angefordert werden, der durch seine Archivnummer VSN gekennzeichnet wird. Dabei braucht nicht zwischen privaten und gemeinschaftlichen Datenträgern (bei der Angabe der Operanden) unterschieden zu werden.

In der Ausgabeliste können folgende Kommentare enthalten sein:

RESERVED	Der Bereich ist zwar belegt, aber ohne Dateinamen. Ein Dateiname konnte eingetragen werden.
FREE SPACE	Der Bereich auf Public-Platten ist frei.
OVERLAY!	Eine Doppelbelegung wurde vorgefunden.
UL-SVL-F5-F1	Platz durch SVL, F5 und F1 belegt (Privatplatten).
NO VTOC REF.	Der Bereich auf Platten ist frei.
F1 OTH. DISC	Der Bereich ist durch F1 einer anderen Privatplatte beschrieben.
NOT ASSIGNED	Bereich ist nicht zugewiesen.
NO REFERENCE	Eventuell ausstehende Allocator-Anforderungen oder vom CMS wurde ein Fehler angezeigt.
SPACE PROBLEM	Fehlerhafter Bereich.

vsn[,type]
Bezeichnet die Platte, deren Belegung aufgelistet werden soll. Wenn eine private Platte nicht online ist, muss außerdem noch der Plattentyp „type“ angegeben werden.
Bei gemeinschaftlichen Platten wird auch freier und toter Platz aufgelistet.

Toter Platz ist ein Bereich, der laut Angabe im F5-Kennsatz zwar belegt ist, auf den aber keine Datei aus dem Systemkatalog verweist.

USER = userid
Mit der Angabe einer Benutzerkennung userid wird die Ausgabe eingeschränkt. In diesem Fall werden auch bei gemeinschaftlichen Datenträgern keine Informationen über freien oder toten Platz ausgegeben.

CATALOG
Diese Funktion gibt eine Liste der belegten Blöcke einer Katalogdatei aus.

Es werden sowohl importierte, als auch nicht importierte Kataloge unterstützt.

Es werden pro belegtem Katalogblock die USERID, der Blockzähler, die Primary Block Number (hexadezimal und dezimal), die Anzahl der Entries in diesem Block und die Anzahl der unbenutzten Bytes des Blocks ausgegeben. Nach ordnungsgemäßer Durchführung der Anweisung wird die Meldung 'LIST PROCESSING COMPLETED' ausgegeben

CATID = catid
bezeichnet

1. den TSOSCAT eines importierten SF-Pubsets.
2. den Gesamtkatalog eines SM-Pubsets.
   Wird kein zusätzlicher Parameter angegeben, erfolgt die Ausgabe der Information für sämtliche Teilkataloge des SM-Pubsets.

   VSID = vsid
   Die Angabe einer VSID beschränkt die Ausgabe auf den Dateikatalog des selektierten Volume-Sets (nur bei SM-Pubsets zulässig).

   SEL = ALL / FILE / PRIV / MIG / JV
   Zusätzliche Auswahl (nur bei SM-Pubsets).

   SEL = ALL
   bearbeitet alle Teilkataloge des ausgewählten SM-Pubsets (analog zu kein zusätzlicher Parameter). Es werden bei den Spezialkatalogen alle Katalogdateien bearbeitet.

   SEL = FILE
   selektiert sämtliche Dateikataloge des SM-Pubsets.

   SEL = PRIV
   bewirkt die Bearbeitung des Spezialkataloges für Privatplatten- und Banddateieinträge. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#P00) bearbeitet.

   SEL = MIG
   wählt den Spezialkatalog für migrierte Dateien bzw. Dateien, für die noch kein Platz zugewiesen wurde, aus. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#M00) bearbeitet.

   SEL = JV
   ermöglicht die Ausgabe von Informationen über den Spezialkatalog für Jobvariable. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#J00) bearbeitet.

vsn
bezeichnet eine Pubres, die entweder den Dateikatalog eines Volume-Sets, oder den TSOSCAT eines SF-Pubsets enthält.

type
bezeichnet den Gerätetyp der durch <vsn> bezeichneten Platte.

VSID = vsid
bezeichnet den Dateikatalog eines Volume-Sets.
Dieser Parameter ist nur bei der Bearbeitung eines SM-Pubsets zulässig.

USER = userid
Mit der Angabe der Benutzerkennung userid wird die Ausgabe der Informationen auf diese beschränkt.

VTOC
Damit wird die Ausgabe von Informationen aus dem F1-Kennsatz einer privaten Platte angefordert.

vsn[,type]
Ist die Archivnummer der Platte, aus deren VTOC-Bereich Informationen gewünscht werden. Wenn der Datenträger nicht online ist, muss zusätzlich auch der Plattentyp „type“ angegeben werden.

ENTRY = filename
Damit wird die Ausgabe auf den Dateieintrag „filename“ beschränkt. Ein
vollqualifizierter Dateiname ist hier nicht verboten, aber sinnvoll ist nur die Angabe eines teilqualifizierten Dateinamens.

Beispiele

1. Ausgabe von LIST ALLOCATION,PAGM.0

```
S P C C N T R L    VERSION <version>            DATE: <date> TIME: <time>                 PAGE 1
COMMAND IN EXECUTION:   LIST ALLOCATION,PAGM.0
                                  EXTENTS ON VOLUME PAGM.0
  CCHHR (LOW) ||     LOW    |    HIGH    ||   ASSIGNMENT   | PAM PAGES ||   USERID   |  FILENAME
--------------||------------|------------||-----------------|-----------||------------|----------------
 ----------   ||          1  |        54  ||  RESERVED      |       54  ||            |
 ----------   ||         55  |      4152  ||                 |     4098  ||  TSOS      |  TSOSCAT
 ----------   ||       4153  |      4176  ||                 |       24  ||  TSOS      |  SYSSRPM
 ----------   ||       4177  |      4182  ||                 |        6  ||  TSOS      |  SYSSRPM.BACKUP
 ----------   ||       4183  |      4224  ||                 |       42  ||            |
 ----------   ||       4225  |      4272  ||                 |       48  ||  TSOS      |  SYSCAT.GUARDS
 ----------   ||       4273  |      4305  ||                 |       33  ||  TSOS      |  SYSDAT.APUB...
 ----------   ||       4306  |      4320  ||                 |       15  ||            |
 ----------   ||       4321  |      4368  ||                 |       48  ||  TSOS      | SYSCAT.GUARD...
 ----------   ||       4369  |      4416  ||                 |       48  ||            |
 ----------   ||       4417  |   1028418  ||                 |  1024002  ||  TSOS      | SYS.PAGING.P...
 ----------   ||    1028419  |   1966080  ||                 |   937662  ||            |
...
```

Hinweise zur Spalte ASSIGNMENT

RESERVED	Der Bereich ist zwar belegt, aber ohne Dateinamen. Ein Dateiname konnte nicht eingetragen werden (Spalte 7).
FREE SPACE	Der Bereich auf Public-Platten ist frei.
OVERLAY!	Eine Doppelbelegung wurde vorgefunden.
UL-SVL-F5-F1	Platz durch SVL, F5 und F1 belegt (Privatplatten).
NO VTOC REF.	Der Bereich auf Platten ist frei.

F1 OTH. DISC	Der Bereich ist durch F1 einer anderen Privatplatte beschrieben.
NOT ASSIGNED	Bereich ist nicht zugewiesen.
NO REFERENCE	Eventuell ausstehende Allocator-Anforderungen oder vom CMSwurde ein Fehler angezeigt.
SPACE PROBLEM	Fehlerhafter Bereich.

2. Ausgabe von LIST CATALOG

```
S P C C N T R L    VERSION <version>            DATE: <date> TIME: <time>                 PAGE     1
COMMAND IN EXECUTION:   LIST CATALOG,CATID=2SBZ,USER=TSOS
                                        SUMMARY OF CONTENTS IN :2SBZ:$TSOS.TSOSCAT
                                                FOR USERID TSOS
      ||                    FILE ENTRIES                    ||                JOB VARIABLE ENTRIES
USERID||BLK COUNT|LBN(DEC)|LBN(HEX)|# ENTRIES|UNUSED||BLK COUNT|LBN(DEC)|LBN(HEX)|# ENTRIES|UNUSED
------||---------|--------|--------|---------|------||---------|--------|--------|---------|-------
TSOS  ||       1 |      2 |  0002  |       1 | 3762 ||       1 |    281 |  0119  |      16 |    96
      ||       2 |      4 |  0004  |      12 |  188 ||       2 |    311 |  0137  |      14 |   181
      ||       3 |     17 |  0011  |      12 |  236 ||       3 |    321 |  0141  |      17 |    24
      ||       4 |     18 |  0012  |      12 |  212 ||       4 |    329 |  0149  |      16 |   112
      ||       5 |     19 |  0013  |      12 |  170 ||       5 |    503 |  01F7  |      14 |   171
      ||       6 |     20 |  0014  |      12 |  205 ||       6 |    506 |  01FA  |      15 |    75
      ||       7 |     21 |  0015  |      12 |  110 ||       7 |    510 |  01FE  |      13 |   183
      ||       8 |     22 |  0016  |      12 |  225 ||       8 |    517 |  0205  |      14 |    47
      ||       9 |     23 |  0017  |      12 |  237 ||       9 |    518 |  0206  |      14 |    47
      ||      10 |     24 |  0018  |      12 |  196 ||      10 |    528 |  0210  |      16 |     0
      ||      11 |     25 |  0019  |      12 |  268 ||      11 |    562 |  0232  |      12 |    80
      ||      12 |     26 |  001A  |      12 |  259 ||      12 |    629 |  0275  |      14 |    59
      ||      13 |     27 |  001B  |      12 |  244 ||      13 |    591 |  024F  |      17 |   165
      ||      14 |     28 |  001C  |      12 |  144 ||      14 |    592 |  0250  |      15 |    98
      ||      15 |     29 |  001D  |      12 |  194 ||      15 |    577 |  0241  |      13 |    14
      ||      16 |     30 |  001E  |      12 |  218 ||      16 |    646 |  0286  |      13 |    13
      ||      17 |     31 |  001F  |      12 |  256 ||      17 |    584 |  0248  |      14 |    27
      ||      18 |     32 |  0020  |      12 |  256 ||      18 |    645 |  0285  |       1 |  3722
      ||      19 |     33 |  0021  |      12 |  205 ||      19 |    648 |  0288  |       7 |  2172
      ||      20 |     34 |  0022  |      12 |  183 ||      20 |    654 |  028E  |       1 |  3730
      ||      21 |     35 |  0023  |      12 |  196 ||         |        |        |         |
      ||      22 |     36 |  0024  |      12 |  141 ||         |        |        |         |
      ||      23 |     37 |  0025  |      12 |  237 ||         |        |        |         |
      ||      24 |     38 |  0026  |      12 |  231 ||         |        |        |         |
      ||      25 |     39 |  0027  |      12 |  177 ||         |        |        |         |
      ||      26 |     40 |  0028  |      12 |  207 ||         |        |        |         |
      ||      27 |     41 |  0029  |      12 |  175 ||         |        |        |         |
      ||      28 |     42 |  002A  |      12 |  139 ||         |        |        |         |
      ||      29 |     43 |  002B  |      12 |   63 ||         |        |        |         |
      ||      30 |     44 |  002C  |      12 |  115 ||         |        |        |         |
      ||      31 |     45 |  002D  |      12 |  147 ||         |        |        |         |
      ||      32 |     46 |  002E  |      12 |  131 ||         |        |        |         |
      ||      33 |     47 |  002F  |      12 |  154 ||         |        |        |         |
      ||      34 |     48 |  0030  |      12 |  149 ||         |        |        |         |
      ||      35 |     49 |  0031  |      12 |  164 ||         |        |        |         |
      ||      36 |     50 |  0032  |      12 |  169 ||         |        |        |         |
      ||      37 |     51 |  0033  |      12 |  128 ||         |        |        |         |
      ||      38 |     52 |  0034  |      12 |  176 ||         |        |        |         |
      ||      39 |     53 |  0035  |      12 |  153 ||         |        |        |         |
      ||      40 |     54 |  0036  |      12 |  142 ||         |        |        |         |
      ||      41 |     55 |  0037  |      12 |  153 ||         |        |        |         |
      ||      42 |     56 |  0038  |      12 |  167 ||         |        |        |         |
      ||      43 |     57 |  0039  |      12 |  167 ||         |        |        |         |
      ||      44 |     58 |  003A  |      12 |  167 ||         |        |        |         |
```

15.2.2.7 MODIFY - Standardmäßig implementierte Voreinstellungen verändern

Mit der MODIFY-Anweisung können im SPCCNTRL standardmäßig implementierte Voreinstellungen, wenn sie anders gewünscht werden, geändert werden. Die Änderung gilt aber nur für den aktuellen Programmlauf.

Die Anweisung kann auch für eine nachträgliche Zuordnung der Meldungsdatei von SPCCNTRL oder irgendeiner anderen Meldungsdatei verwendet werden. Das ist nur unter der Benutzerkennung TSOS möglich.
Die Meldungsdatei bleibt im System bis zum Ende des Systemlaufs, falls nicht mit `/MODIFY-MSG-FILE-ASSIGNMENT` eine Änderung veranlasst wurde.

Format

MODIFY
{ LINES = nn / MSG = { **SPC**CNTRL / (FILE,filename) } }

Operanden

LINES = nn
Bezeichnet die gewünschte Zeilenanzahl nn pro Seite bei der Listenausgabe (14 <= nn <= 127). Fehlt der Operand, wird als Standardwert 60 angenommen.

MSG = SPCCNTRL
Nur TSOS.
Die für SPCCNTRL gültige Meldungsdatei soll zugeordnet werden. Als Dateiname wird SYSMES.SPCCNTRL.version erwartet.
Als „version“ gilt die Versionsnummer des jeweils verwendeten SPCCNTRL.

MSG = (FILE,filename)
Es sollen Meldungen aus der Datei, die hier unter filename angegeben ist, nachgeladen werden.

15.2.2.8 PURGE - Entfernen von totem Platz und Löschen von Katalogeinträgen

Mit der PURGE-Anweisung kann toter Platz auf gemeinschaftlichen Datenträgern entfernt und Katalogeinträge können aus dem Systemkatalog gelöscht werden. Weiterhin ist das Entfernen von unbenutztem Platz in den einzelnen Dateien möglich.

Format

```
PURGE

SPACE
    , { PUBLIC [,{ CATID = cat-id / VSID = vsid / vsn } ]  /
      PRIVATE[,vsn[,type]]   }
    [,USER = userid]
ALLOCATION,vsn
CATALOG
    [,CATID = catid]
    ,ENTRY = filename / (filename,JV)
```

Operanden

SPACE
Freigeben unbenutzten Platzes in den einzelnen Benutzerdateien.
Die zugewiesenen Puffergrößen werden berücksichtigt, d.h. es wird nur soviel Platz freigegeben, dass die betreffende Datei anschließend ohne Fehler bearbeitet werden kann. Diese Funktion steht jedem Benutzer zur Verfügung. In diesem Fall werden jedoch nur Dateien der eigenen Kennung berücksichtigt.
Geschützte Dateien bleiben unberührt. Ist der ungenutzte Bereich einer Datei gleich dem reservierten Bereich dieser Datei, bleibt der Datei 1 Allokierungseinheit erhalten.

PUBLIC
Es soll unbenutzter Platz auf gemeinschaftlichen Datenträgern freigegeben werden. Die Funktion wird nur auf importierten Pubsets durchgeführt.

CATID = catid
Bezeichnet die Katalogkennung eines SF-Pubsets oder eines SM-Pubsets, aus dem Einträge gelöscht werden sollen.
Ist weder vsn noch CATID= angegeben, wird die Standard-Katalogkennung der bei USER= angegebenen Benutzerkennung angenommen.

VSID = vsid
stellt die Kennung eines Volume-Sets dar.
Dieser Operand ist der Benutzerkennung TSOS vorbehalten.

vsn
Ist die Archivnummer eines bestimmten gemeinschaftlichen Datenträgers, auf dem unbenutzter Platz freigegeben werden soll.

PRIVATE
Es soll unbenutzter Platz auf privaten Datenträgern freigegeben werden. Ein nichtprivilegierter Benutzer kann nur diejenigen seiner Dateien behandeln, die auch Einträge im Systemkatalog haben.

vsn[,type]
Bezeichnet die Platte, auf der der Platz freigegeben werden soll. Die Angabe der Archivnummer (vsn) und des Typs „type“ ist unter nichtprivilegierter
Benutzerkennung unbedingt erforderlich, unter TSOS genügt für verfügbare Platten (online) die vsn.

USER = userid
Bezeichnet die Benutzerkennung, deren Dateien behandelt werden sollen. Fehlt die Angabe und SPCCNTRL läuft unter der Kennung TSOS, dann werden bei PURGE SPACE,PUBLIC die Dateien aller Benutzer mit Ausnahme von TSOS behandelt.
Bei PURGE SPACE,PUBLIC ohne weitere Parameter unter einer Kennung ungleich TSOS werden sämtliche Dateien des Default-Pubsets der Aufruferkennung auf freizugebenden Platz überprüft.
Bei PURGE SPACE,PRIVATE werden bei fehlender USER-Angabe die Dateien aller Benutzer (auch TSOS) behandelt, falls SPCCNTRL unter TSOS läuft.

ALLOCATION
Dieser Operand ist der Benutzerkennung TSOS vorbehalten.
Die Belegungsverteilung auf einem gemeinschaftlichen Datenträger soll bereinigt, d.h. es soll toter Platz entfernt werden. Das geschieht aber nicht im gleichen SPCCNTRL-Lauf, sondern nach dessen Beendigung und Angabe von `/IMPORT-PUBSET`.
Toter Platz kann durch Systemfehler entstanden sein oder explizit durch die Anweisung PURGE CATALOG, ENTRY= bewirkt werden.

vsn
Archivnummer eines gemeinschaftlichen Datenträgers, der zu bereinigen ist.

CATALOG
Dieser Operand ist der Benutzerkennung TSOS vorbehalten.
Es soll ein Eintrag aus dem Systemkatalog entfernt werden. Dies kann sinnvoll sein für fehlerhafte Einträge, die nicht mehr angesprochen werden können.

CATID = catid
Bezeichnet den Katalog, aus dem ein Eintrag entfernt werden soll. Fehlt der Operand, ist der Home Katalog Standardwert.

ENTRY = filename
Bezeichnet den Namen der Datei, deren Katalogeintrag entfernt werden soll.

ENTRY = (filename,JV)
Bezeichnet den Namen der Jobvariablen, deren Katalogeintrag entfernt werden soll.

Beispiel

Ausgabe von PURGE SPACE, PUBLIC

```
PURGE S,PUBLIC
% SPC0025 PURGE PROCESSING COMPLETED
NUMBER OF FILES SELECTED FOR PURGE:     1062   --------------------------- 1.
NUMBER OF ERRORS DURING DEALLOCATION:    121   --------------------------- 2.
AMOUNTED OF PAM PAGES DEALLOCATED:     20787   --------------------------- 3.
```

1. Anzahl der Dateien, bei denen freizugebender Speicherplatz festgestellt wurde.
2. Anzahl der Dateien, bei denen die Freigabe nicht durchgeführt werden konnte.
3. Gesamtanzahl der freigegebenen PAM-Seiten.

15.2.2.9 TRACE - Identifizieren und Ausgeben von Blöcken und Einträgen im Systemkatalog und VTOC-Bereich einer Privatplatte

Mit der TRACE-Anweisung können Blöcke und Einträge im Systemkatalog und VTOC-Bereich einer Privatplatte identifiziert und nach SYSOUT oder SYSLST ausgegeben werden. Darüber hinaus kann auch jeder beliebige Block einer gemeinschaftlichen oder privaten Platte gezeigt werden.
Um Schreibarbeit zu sparen, gelten die Einstellungen der Schlüsselwortoperanden solange, bis sie durch eine nachfolgende Eingabe geändert werden.

Die Anweisung ist der Benutzerkennung TSOS vorbehalten.

Format

TRACE

CATALOG, block[,byte]

[, { CATID = cat-id [,{ VSID = vsid / SEL = { PRIV / MIG / JV} }] /

vsn[,type] /

VSID = vsid }]

[,USER = userid]

[,JV = **N**O / **Y**ES]

[,OUTPUT = (BLOCK[,{ SYSOUT / SYSLST }]) / (OWNER[,SYSOUT]) / SYSOUT / SYSLST]

VTOC,{ block / (block,PHP) } [,byte],vsn[,type]

[,OUTPUT = (BLOCK[,{ SYSOUT / SYSLST }]) / SYSOUT / SYSLST]

SPACE,block[,byte],vsn[,type]

[,OUTPUT = (BLOCK[,{ SYSOUT / SYSLST }]) / SYSOUT / SYSLST]

Operanden

CATALOG
Es sollen Informationen aus dem Systemkatalog geliefert werden. Jeder Block, auch wenn er keinem Benutzer zugewiesen ist, einschließlich der Kontrollblöcke mit den Bit-Tabellen kann gezeigt werden.

Es werden sowohl importierte, als auch nicht importierte Kataloge unterstützt.

block
Bezeichnet die Nummer des auszugebenden Blocks. Die Angabe kann auch sedezimal in der Form X'hhhh' gemacht werden. Ist eine Benutzerkennung USER=userid angegeben, dann gilt die Angabe relativ zur Katalog-Kette des Benutzers, ansonsten wird „block“ als absolute Blocknummer interpretiert.

Zusammen mit OUTPUT=(BLOCK,SYSLST) kann „block“ weggelassen werden.
In diesem Fall wird der gesamte Katalog ausgegeben. Bei Katalogen mit ungerader Seitenzahl wird die letzte (Halb-)Seite nicht ausgegeben. Diese wird nicht genutzt und ist daher leer.

byte
Bezeichnet die Byte-Nummer innerhalb des Blocks, ab der ausgegeben werden soll. Auch hier kann die Angabe sedezimal in der Form X'hhhh' gemacht werden.

CATID = catid
bezeichnet

1. den TSOSCAT eines importierten SF-Pubsets.
2. den Gesamtkatalog eines SM-Pubsets. In diesem Fall ist eine weitere Spezifizierung durch VSID oder SEL zwingend.

VSID = vsid
Die Angabe einer VSID beschränkt die Ausgabe auf den Dateikatalog des selektierten Volume-Sets (nur bei SM-Pubsets zulässig, siehe auch nachfolgenden Hinweis).

SEL = PRIV / MIG / JV
bezeichnet einen Spezialkatalog (nur bei SM-Pubsets).

SEL = PRIV
bewirkt die Bearbeitung des Spezialkataloges für Privatplatten- und Banddateieinträge. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#P00) bearbeitet.

SEL = MIG
wählt den Spezialkatalog für migrierte Dateien bzw. Dateien, für die noch kein Platz zugewiesen wurde, aus. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#M00) bearbeitet.

SEL = JV
ermöglicht die Ausgabe von Informationen über den Spezialkatalog für Jobvariable. Es wird nur die erste Datei des Spezialkatalogs (TSOSCAT.#J00) bearbeitet.

> **i** Bei SF-Pubsets sind die Operanden VSID und SEL verboten. Bei SM-Pubsets ist die Angabe eines der Operanden VSID oder SEL zwingend!

vsn
bezeichnet eine Pubres, die entweder den Dateikatalog eines Volume-Sets, oder den TSOSCAT eines SF-Pubsets enthält.

type
bezeichnet den Gerätetyp der durch <vsn> bezeichneten Platte.

VSID = vsid
bezeichnet den Dateikatalog eines Volume-Sets. Dieser Parameter ist nur bei der Bearbeitung eines SM-Pubsets zulässig.

USER = userid
Bezeichnet die Kennung des Benutzers, aus dessen Katalogeintrag Blöcke ausgegeben werden sollen. Bei Angabe einer Userid ist für SM-Pubsets lediglich die Bearbeitung eines Teilkataloges möglich. Dieser Teilkatalog muss durch die Angabe einer <vsid> oder des Parameters SEL definiert werden.

JV = NO / YES
Dieser Parameter ist nur bei Bearbeitung eines SF-Pubset-Kataloges zulässig. Im Fall eines SM-Pubsets wird er ignoriert.

JV = NO
Jobvariablen sollen ignoriert werden (Standard). Es wird ein Block aus der Dateinamenkette des Benutzers ausgegeben.

JV = YES
Es sollen Blöcke aus einer Jobvariablenkette gezeigt werden. Diese Angabe ist nur sinnvoll zusammen mit der Angabe USER=userid, ansonsten wird sie ignoriert.

OUTPUT = (BLOCK,SYSOUT)
Die Ausgabe der Informationen soll nach SYSOUT erfolgen (Standard).

OUTPUT = (BLOCK,SYSLST)
Die Ausgabe soll nach SYSLST erfolgen.

OUTPUT = (OWNER,SYSOUT)
gibt nicht den Inhalt des gewünschten Blocks, sondern den Eigentümer (Benutzerkennung) nach SYSOUT aus.

OUTPUT = SYOUT / SYSLST
Die Angabe ist gleichbedeutend mit (BLOCK,SYSOUT) bzw. (BLOCK,SYSLST).

VTOC
Es sollen Informationen aus dem VTOC-Bereich einer privaten Platte geliefert werden.

block
Bezeichnet die Nummer des auszugebenden Blocks. Die Angabe kann auch sedezimal in der Form X'hhhh' gemacht werden.
Zusammen mit OUTPUT=(BLOCK,SYSLST) kann „block“ weggelassen werden.
In diesem Fall wird der gesamte VTOC-Bereich ausgegeben.

PHP
Bedeutet, dass die Blocknummer physikalisch und nicht relativ zum Beginn des VTOC-Bereichs gemeint ist.

byte
Bezeichnet die Byte-Nummer innerhalb des Blocks, ab der ausgegeben werden soll. Die Angabe kann auch sedezimal in der Form X'hhhh' gemacht werden.

vsn[,type]
Ist die Archivnummer der privaten Platte, deren VTOC ausgegeben werden soll. Ist die Platte nicht verfügbar (online), muss auch der Plattentyp „type“ angegeben werden.

OUTPUT = (BLOCK,SYSOUT)
Die Ausgabe der Informationen soll nach SYSOUT erfolgen (Standard).

OUTPUT = (BLOCK,SYSLST)
Die Ausgabe soll nach SYSLST erfolgen.

OUTPUT = SYOUT / SYSLST
Die Angabe ist gleichbedeutend mit (BLOCK,SYSOUT) bzw. (BLOCK,SYSLST).

SPACE
Es soll eine beliebige Information aus der spezifizierten Platte gezeigt werden.

block
Bezeichnet die Nummer des auszugebenden Blocks. Die Angabe kann auch sedezimal in der Form X'hhhh' gemacht werden. Zusammen mit OUTPUT=SYSLST kann block weggelassen werden. In diesem Fall wird das gesamte SVL zusammen mit dem F5-Kennsatz aufgelistet.

byte
Bezeichnet die Byte-Nummer innerhalb des Blocks, ab der ausgegeben werden soll. Die Angabe kann auch sedezimal (X'hhhh') gemacht werden.

vsn[,type]
Ist die Archivnummer der gemeinschaftlichen oder privaten Platte, von der Informationen gezeigt werden sollen. Ist die Platte nicht verfügbar (online), muss auch der Plattentyp „type“ angegeben werden.

OUTPUT = (BLOCK,SYSOUT)
Die Ausgabe der Informationen soll nach SYSOUT erfolgen (Standard).

OUTPUT = (BLOCK,SYSLST)
Die Ausgabe soll nach SYSLST erfolgen.

OUTPUT = SYOUT / SYSLST
Die Angabe ist gleichbedeutend mit (BLOCK,SYSOUT) bzw. (BLOCK,SYSLST).

Beispiel

```
(IN)     TRACE CATALOG,1,10,CATID=2SBZ
(OUT)    :2SBZ:TSOSCAT       LBN:     1  (0001)    PHP:       28 ON  VOLUME 2SBZ.0
(NL)     ------------------------------------------------------------------------------
(NL)
(NL)      0009   (000):     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      ~~~~~~~~~~~~~~~~
(NL)      0019   (010):     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      ~~~~~~~~~~~~~~~~
(NL)      0029   (020):     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      ~~~~~~~~~~~~~~~~
(NL)      0039   (030):     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      ~~~~~~~~~~~~~~~~
(NL)      0049   (040):     FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      ~~~~~~~~~~~~~~~~
(NL)      0059   (050):     FFFFFFFF 00000000 00000000 00000000      ~~~~............
(NL)      0069   (060):     00000000 00000000 00000000 00000000      ................
(NL)      0079   (070):     00000000 00000000 00000000 00000000      ................
(NL)      0089   (080):     00000000 00000000 00000000 00000000      ................
(NL)      0099   (090):     00000000 00000000 00000000 00000000      ................
(NL)      00A9   (0A0):     00000000 00000000 00000000 00000000      ................
(NL)      00B9   (0B0):     00000000 00000000 00000000 00000000      ................
(NL)      00C9   (0C0):     00000000 00000000 00000000 00000000      ................
(NL)      00D9   (0D0):     00000000 00000000 00000000 00000000      ................
(NL)      00E9   (0E0):     00000000 00000000 00000000 00000000      ................
(NL)      00F9   (0F0):     00000000 00000000 00000000 00000000      ................
(NL)      0109   (100):     00000000 00000000 00000000 00000000      ................
(NL)      0119   (110):     00000000 00000000 00000000 00000000      ................
(NL)      0129   (120):     00000000 00000000 00000000 00000000      ................
(OUT)    %  SPC0020 SCREEN OVERFLOW, CONTINUE (Y/N)?
```

16 TPCOMP2 Vergleichen der Daten zweier Magnetbänder

Versionsstand: TPCOMP2 V21.0C

TPCOMP2 ist ein Dienstprogramm, das die Daten zweier Magnetbänder vergleicht. Aufgelistet werden die Teile, die nicht übereinstimmen.

Der Vergleich der Daten wird in Dekaden durchgeführt. Dadurch ist es möglich, 5 Gruppen voneinander abweichender Daten in einer Zeile auf einem Drucker mit 132 Schreibstellen oder 6 Gruppen auf einem Drucker mit 160 Schreibstellen je Zeile auszudrucken.

Ein- und Ausgabe

Die Eingabe für das Bandvergleichsprogramm besteht aus zwei Magnetbändern, die im selben Format beschrieben wurden.
Für die Standardfunktionen dieses Programms müssen keine Parameter eingegeben werden. Sollen zusätzliche Funktionen ausgeführt werden, so sind die Parameter über SYSDTA einzugeben.

Die Ausgabe ist die Druckerliste aller Abweichungen, die zwischen den miteinander verglichenen Magnetbändern bestehen.

16.1 Funktionen

Neben den folgenden Standardfunktionen können durch Angabe von Parametern zusätzliche Funktionen des Programms ausgewählt werden (siehe „Wahlweise Funktionen“).

Standardfunktionen

TPCOMP2 hat vorgegebene Parameter, die folgende Standardfunktionen vorsehen:

1. Rückspulen beider Bänder zur Anfangsmarke zu Beginn und Ende des Vergleichslaufes.
2. Ausdrucken voneinander abweichender Daten im Sedezimal- und EBCDIC-Format auf einem Drucker mit 132 Schreibstellen je Zeile.
3. Beenden des Vergleichs, wenn eine doppelte Abschnittsmarke erkannt wird.(Wenn eine doppelte Abschnittsmarke auf einem Band festgestellt wird und das andere Band nicht ebenfalls auf eine doppelte Abschnittsmarke positioniert ist, werden die restlichen Daten des 2. Bandes solange ausgedruckt, bis die Abschnittsmarken erkannt werden.)

Wahlweise Funktionen

Durch die Angabe von Parametern, die über SYSDTA eingegeben werden, können die folgenden Funktionen ausgewählt werden:

1. Entsprechend der angegebenen Anzahl von Abschnittsmarken oder Blöcken können die Bänder in Vorwärts- oder Rückwärtsrichtung positioniert werden.
2. Der Vergleich kann auf Grund einer gegebenen Anzahl von Abschnittsmarken oder Blöcken beendet werden.
3. Es besteht die Möglichkeit, einen Drucker mit 160 Schreibstellen pro Zeile anzugeben und in sedezimaler, in EBCDIC-Darstellung oder in einer Kombination von beiden auszugeben.
4. Es können ungeblockte Sätze mit variabler Länge oder geblockte Sätze mit fester Länge verarbeitet werden.

16.2 Programmlauf starten

Das Dienstprogramm TPCOMP2 wird gestartet mit

```
/START-EXECUTABLE-PROGRAM $.SYSPRG.TPCOMP2.210
```

Ferner stehen Ihnen die Operanden CPU-LIMIT, TEST-OPTIONS, MONJV, RESIDENT-PAGES und VIRTUAL-PAGES von `/START-EXECUTABLE-PROGRAM` zur Verfügung, um z.B. den Programmlauf überwachen zu können (siehe Handbuch „Kommandos“ [1]).

16.3 Anweisungen

- Übersicht über die Anweisungen von TPCOMP2
- Beschreibung der Anweisungen
  - COM - Bereiche von Eingabebänder vergleichen
  - END - TPCOMP2 beenden
  - LIM - Bandvergleich nach bestimmter Anzahl von Blöcken beenden
  - POS - Magnetbänder positionieren
  - RCD - Verarbeitung von Eingabesätzen variabler Länge festlegen
  - STP - TPCOMP2-Lauf bei abweichenden Magnetbändern anhalten

16.3.1 Übersicht über die Anweisungen von TPCOMP2

Operation	Operanden	Bedeutung
COM	stcnnnnn	Vergleichen bestimmter Bereiche der Bänder und Festlegen der Druckzeilenlänge und des Druckmodus
END		Beenden der Eingabe
LIM	nnnnn	Beenden des Vergleichs nach einer bestimmten Anzahl von Blöcken
POS1 POS2	pppnnnnn pppnnnnn	Positionieren der Magnetbänder
RCD	t nnnnn	Angeben des Satzformats und der Satzlänge (bei Sätzen fester Länge)
STP		Stoppen des Programmlaufs, wenn sich die Daten zweier Bänder unterscheiden

16.3.2 Beschreibung der Anweisungen

- COM - Bereiche von Eingabebänder vergleichen
- END - TPCOMP2 beenden
- LIM - Bandvergleich nach bestimmter Anzahl von Blöcken beenden
- POS - Magnetbänder positionieren
- RCD - Verarbeitung von Eingabesätzen variabler Länge festlegen
- STP - TPCOMP2-Lauf bei abweichenden Magnetbändern anhalten

16.3.2.1 COM - Bereiche von Eingabebänder vergleichen

Die COM-Anweisung wird benötigt, wenn nur bestimmte Bereiche der Eingabebänder verglichen werden sollen. Sie kann auch dazu verwendet werden, die Druckzeilenlänge festzulegen und den Druckmodus auszuwählen.

Format

COM
stcnnnnn

Operanden

s
Anzahl der Schreibstellen:

s = 0 Drucker mit 132 Schreibstellen (Standard).

s = 1 Drucker mit 160 Schreibstellen.

t
Druckmodus:

t = C Druckmodus ist Sedezimal- und EBCDIC-Modus (Standard).

t = H Druckmodus ist Sedezimalmodus.

t = G Druckmodus ist EBCDIC-Modus.

c
Angabe, wann beendet werden soll:

c = T Beenden nach dem Lesen von nnnnn Abschnittsmarken auf beiden Bändern.

c = B Beenden nach dem Lesen von nnnnn Blöcken auf beiden Bändern.

nnnnn
Dezimale Anzahl der Abschnittsmarken oder Blöcke (mit führenden Nullen).

Beispiele

```
COM OCT00006
COM 1HB00672
```

> Nachdem die durch eine Vergleichsanweisung angegebene Operation ausgeführt wurde, nimmt das Programm eine Positionierungsanweisung an (falls vorhanden). An dieser Stelle kann das Programm durch Angabe einer END-Anweisung beendet werden, oder es können, durch Eingabe weiterer Vergleichs- und Positionierungsanweisungen, weitere Vergleiche der Eingabebänder ausgeführt werden. Wenn zwei Positionierungsanweisungen unmittelbar aufeinander folgen, wird die Vergleichsfunktion mit den Standardwerten ausgeführt.

16.3.2.2 END - TPCOMP2 beenden

Die END-Anweisung kennzeichnet das Ende der Eingabe.

Format

END

Diese Anweisung besitzt keine Operanden.

16.3.2.3 LIM - Bandvergleich nach bestimmter Anzahl von Blöcken beenden

Mit dieser Anweisung wird der Bandvergleich nach einer bestimmten Anzahl von Blöcken auf beiden Bändern beendet.

Format

LIM
nnnnn

Operanden

nnnnn

Anzahl der Blöcke auf beiden Bändern, die verglichen werden sollen. Standardmäßig werden alle Blöcke verglichen.

16.3.2.4 POS - Magnetbänder positionieren

Die POS-Anweisung dient zum Positionieren der Magnetbänder.

Format

POS1 / POS2
pppnnnnn

Operanden

Die Anweisung POS2 mit ihren Operanden wird nur benötigt, wenn beide Magnetbänder positioniert werden sollen.

ppp
Positionierung:
ppp=FTM Lesen vorwärts nnnnn Abschnittsmarken.
ppp=RTM Lesen rückwärts nnnnn Abschnittsmarken.
ppp=FBK Lesen vorwärts nnnnn Blöcke.
ppp=RBK Lesen rückwärts nnnnn Blöcke.
ppp=BOT Rückspulen zur Anfangsmarke.

nnnnn
Dezimale Anzahl von Abschnittsmarken oder Blöcken (mit führenden Nullen).

Beispiele

```
POS1 FTM00002
POS2 FBK00333
POS1 FBK00017,POS2 FBK00017
```

16.3.2.5 RCD - Verarbeitung von Eingabesätzen variabler Länge festlegen

Die RCD-Anweisung wird verwendet, um anzuzeigen, dass die Eingabesätze variable Länge haben oder die geblockten Sätze mit fester Länge als logische Sätze verarbeitet werden sollen. Diese Anweisung ist erforderlich, wenn Sätze variabler Länge oder geblockte Sätze fester Länge miteinander verglichen werden sollen.

Format

RCD
t nnnnn *)

*) Die Operanden t und nnnnn werden mit einem Leerzeichen von einander getrennt.

Operanden

t
Angabe, ob Sätze fester Länge (F) oder variabler Länge (V).

nnnnn
Längenangabe bei Sätzen fester Länge (mit führenden Nullen).
Dieser Eintrag ist nicht erforderlich bei Sätzen variabler Länge.)

16.3.2.6 STP - TPCOMP2-Lauf bei abweichenden Magnetbändern anhalten

Die STP-Anweisung wird verwendet, um den Ablauf von TPCOMP2 zu stoppen, wenn beim Vergleich zweier Magnetbänder die Daten voneinander abweichen. Sie muss vor jeder COM-Anweisung oder vor zwei aufeinander folgenden POS-Anweisungen wiederholt werden.

Format

STP

Diese Anweisung besitzt keine Operanden.

16.4 Anwendung von /ADD-FILE-LINK

Bevor TPCOMP2 aufgerufen wird, muss der Benutzer für jede zu vergleichende Banddatei einen Dateikettungsnamen (Link-Namen) vereinbaren (`/ADD-FILE-LINK`):

Erstes Band: LINK-NAME=COM001
Zweites Band: LINK-NAME=COM002

Für nichtkatalogisierte Banddateien muss zuerst mit `/IMPORT-FILE` ein Katalogeintrag erstellt werden.

Folgende Parameter der nötigen Kommandos können angegeben werden:

Katalogisierte Banddatei

```
/ADD-FILE-LINK
   LINK-NAME=COM00X
   ,FILE-NAME=dateiname
   [,SUPPORT=*TAPE(VOLUME-LIST=*CATALOG(VOL-SEQ-NUM=n))]
```

Nichtkatalogisierte Banddatei

```
/IMPORT-FILE
   SUPPORT=*TAPE
   (VOLUME=archivnummer
   ,DEVICE-TYPE=XXXX
   ,FILE-NAME=dateiname
   [,PREMOUNT-LIST=n])
/ADD-FILE-LINK
   LINK-NAME=COM00X
   ,FILE-NAME=dateiname
```

Zusätzlich kann bei `/ADD-FILE-LINK` folgender Operand angegeben werden:

`[,SUPPORT=*TAPE(LABEL-PROCESSNG=*PAR(LABEL-TYPE=...))]`

Beim Vergleich von zwei Mastertapes ist LABEL-TPYE=*NON-STD anzugeben. In diesem Fall ist die Art der Etikettierung durch die Programmparameter anzugeben.

Beispiele

1. Eine katalogisierte und eine nichtkatalogisierte Banddatei sollen standardmäßig miteinander verglichen werden.

```
/ADD-FILE-LINK LINK-NAME=COM001,FILE-NAME=DATEI1
/IMPORT-FILE SUPPORT=*TAPE(VOLUME=BAND1,DEVICE-TYPE=<device-type>, -
/            FILE-NAME=DATEI2)
/ADD-FILE-LINK LINK-NAME=COM002,FILE-NAME=DATEI2
/START-EXECUTABLE-PROGRAM FROM-FILE=$TPCOMP2
END
```

2. Es sollen die Daten, die zwischen der vierten und fünften Abschnittsmarke auf einem Magnetband stehen, mit den Daten verglichen werden, die zwischen der ersten und zweiten Abschnittsmarke auf einem anderen Magnetband stehen.
   Alle Abweichungen sollen im Zeichenmodus auf einem Drucker mit 160 Druckstellen ausgedruckt werden.
   Die zwei zu vergleichenden Banddateien sind bereits katalogisiert.

```
/ADD-FILE-LINK LINK-NAME=COM001,FILE-NAME=DATEI1
/ADD-FILE-LINK LINK-NAME=COM002,FILE-NAME=DATEI2
/START-EXECUTABLE-PROGRAM FROM-FILE=$.SYSPRG.TPCOMP2.210
POS1 FTM00004,POS2 FTM00001
COM  1GT00001
END
```

16.5 Meldungen von TPCOMP2

Im Falle einer außergewöhnlichen Situation wird eine der folgenden Meldungen nach SYSOUT ausgegeben:

```
2801A PARAMETER ERROR
```

Bedeutung
Unzulässiges Format.

Maßnahme
Beendigung.

```
2802 ERROR READING PARAMETER CARD
```

Bedeutung
Fehler beim Einlesen einer Karte.

Maßnahme
Beendigung.

```
2803 RECORD TRUNCATED FILE zzz BLK yyyyy IPT x
```

Bedeutung
Die Länge des gelesenen Blocks ist größer als die angegebene Blocklänge.

Maßnahme
Beendigung.

```
2804 WR xxx ERROR. RETURNCODE: yyyy
```

Bedeutung
Fehler (yyy) bei der Ausgabe nach SYSLST (xx=LST) oder SYSOUT (xxx=OUT)

Maßnahme
Beendigung.

```
2809A BOT OR DOUBLE TAPE MARK WHILE POSITIONING IPTx
```

Bedeutung
Während der Positionierung des Eingabebandes (x=1/2) wurde eine doppelte Abschnittsmarke oder die Endemarke des Bandes erkannt, bevor die angegebene Blockzahl erreicht ist. Dabei steht x=1 für das erste und x=2 für das zweite Band.

Maßnahme
Fortsetzung. Das Band wird zurückgespult und neu gestartet, wenn es bei der Anfangspositionierung passiert. Wenn es bei der Endepositionierung passiert, wird der abschließende Ausdruck erzeugt.

```
2810 Unmatched TM on IPTx
```

Bedeutung
Auf dem Eingabeband (x=1/2) wurde keine passende Abschnittsmarke gefunden.

2810A DBL TM ON IPTx

Bedeutung
Doppelte Abschnittsmarke erreicht, bevor die angegebene Anzahl Blöcke verglichen ist. Dabei steht x=1 für das erste und x=2 für das zweite Band.

Maßnahme
Fortsetzung. Das andere Band wird ausgedruckt.

2821A SHORT RECORD

Bedeutung
Die Länge des eingelesenen Satzes ist kürzer als die angegebene feste Satzlänge.

Maßnahme
Fortsetzung.

2830 yyyyy BLOCK IPTx READ ERROR

Bedeutung
Nicht behebbarer Lesefehler im Block yyyyy des Eingabebandes. Dabei steht x=1 für das erste Band und x=2 für das zweite Band.

Maßnahme
Der Block yyyyy wird mit dem entsprechenden Block des Bandes verglichen. Die Verarbeitung wird fortgesetzt.

2800 VERSION xx OF TAPE COMPARE READY

Bedeutung
Programm ist geladen. xx=Versionsnummer.

Maßnahme
Die Verarbeitung wird fortgesetzt.

2835 OPEN ERROR xy CODE=xxxx COMPARE TERMINATED

Bedeutung
OPEN-Fehler für Eingabeband xy. xxxx=DVS-Fehlercode.

Maßnahme
Beendigung.

2836 CLOSE ERROR xy CODE=xxxx COMPARE TERMINATED

Bedeutung
CLOSE-Fehler für Eingabeband xy. xxxx=DVS-Fehlercode.

Maßnahme
Beendigung.

2840 BTAM ERROR CODE=xxxx

Bedeutung
I/O-Fehler. xxxx=DVS-Fehlercode.

Maßnahme
Beendigung.

17 VOLIN Initialisieren von Plattenspeichern

Das Dienstprogramm VOLIN (VOLume INitializer) bereitet eine Magnetplatte (kurz: Platte) so vor, dass sie als leerer Datenträger in BS2000 benutzt werden kann. Dabei spielt es keine Rolle, ob der Datenträger neu ist oder vorher schon in BS2000 oder einem anderen Betriebssystem Verwendung fand. Wichtig ist allein, dass der Plattentyp in BS2000 bedient wird (siehe Handbuch „Systeminstallation“ [7]).

Zusätzlich zur Kennsatzgenerierung können Sie mit VOLIN auch die Nutzungsart einer Platte festlegen. Von der Nutzungsart leitet das Datenverwaltungssystem ab, ob die Zugriffsfunktionen verwendet werden dürfen, die von dem Vorhandensein des PAM-Schlüssels ausgehen. Bei der Nutzungsart ohne PAM-Schlüssel (NK) sind nur die NK-Zugriffsmethoden erlaubt.

Der Volumetyp wird durch folgende Eigenschaften definiert:

- DVS-PAMKEY-Eigenschaft, d. h. Nutzungsart (PK):
  mit Schlüssel (K) / ohne Schlüssel (NK)
- kleinste Allokierungseinheit (AU), d.h. Größe der kleinsten Datei, genauer:
  kleinste Speicherplatzeinheit: 6 KB / 8 KB / 64 KB
- kleinste Transfer Unit (TU), d.h. kleinste Übertragungseinheit zwischen Platte und Hauptspeicher: 2 KB

Durch Kombination dieser Eigenschaften, die mit VOLIN festgelegt werden, stehen folgende Volumetypen zur Verfügung:

Nutzungsart (PK)	Allokierungseinheit (AU)	Transfer Unit (TU)	Kurzbezeichnung	für Plattentyp
K	6K	2K	K-Platte	Privat, Public
NK	6K	2K	NK2-Platte	Privat, Public
NK	8K	2K	NK2(8K)-Platte	Public
NK	64K	2K	NK2(64K)-Platte	Public

17.1 Ablauf von VOLIN

Das Subsystem VOLIN wird nicht mit `/START-SUBSYSTEM` geladen, sondern wird durch den Aufruf des Dienstprogrammes VOLIN gestartet:

`/START-VOLIN`

START-VOLIN	Alias: **VOLIN**
VERSION = *STD / <product-version> **,MONJV** = ***NONE** / <filename 1..54 without-gen-vers> **,CPU-LIM**IT = ***JOB-RE**ST / <integer 1..32767 *seconds*>	

VOLIN wird auch durch Aufruf des Dienstprogrammes SIR automatisch nachgeladen:

`/START-SIR (/SIR)`

Mit der Anweisung END wird VOLIN beendet.

Mit Beendigung des Dienstprogrammes wird auch das Subsystem entladen.

17.2 Erstellung der BS2000-Kennsätze (Kennsatzgenerierung)

VOLIN schreibt auf einen Datenträger neue Kennsätze entweder nach Abschluss des Überschreibens (Anweisung FMT=YES oder FMT=n) oder sofort (Anweisung FMT=NO).

Bei FMT=YES wird die Platte einmal, bei FMT=n n-mal mit einem bestimmten Bitmuster überschrieben.

VOLIN schließt das Initialisieren der Platte mit der Übertragung des neuen Standard-Datenträger-Kennsatzes ab. Sein Vorhandensein ist eine Garantie, dass auch die restlichen Kennsätze auf dem Datenträger erneuert worden sind.

Die Verwendung der Platte nach dem VOLIN-Lauf richtet sich danach, ob durch das Initialisieren ein privater Datenträger, ein gemeinschaftlicher Datenträger oder die Pubres eines Pubsets eingerichtet wurde.

1. Ein privater Datenträger kann unmittelbar nach dem VOLIN-Lauf durch das Datenverwaltungssystem von BS2000 bedient werden.
2. Ein gemeinschaftlicher Datenträger kann bei einem späteren Laden eines BS2000-Systems (STARTUP) oder bei einem späteren IMPORT-PUBSET montiert werden, um eine bestehende gemeinschaftliche Datenträgergruppe (Pubset) zu erweitern.

   Folgende Pubset-globalen Eigenschaften müssen berücksichtigt werden:

   - Seine Archivnummer (VSN=Volume Serial Number) muss zum Pubset passen.
   - Er muss die gleichen Volume-Eigenschaften bzgl. PK (PAMKEY-Eigenschaft), AU (Allocation Unit) und TU (Transfer Unit) aufweisen wie die Pubres. Danach sind folgende Pubset-Typen möglich:
     - K-Pubset (implizit AU=6 KB)
     - NK2-Pubset (implizit AU=6 KB)
     - NK2-Pubset mit AU=8 KB
     - NK2-Pubset mit AU=64 KB
3. Ein von VOLIN anhand der Kennsätze zur Verwendung als Pubres generierter Datenträger ist nicht funktionsfähig und hat keine direkte Verwendung in BS2000. Ein funktionsfähiges Pubset kann direkt nur mit SIR oder mit der Erstinstallation erstellt werden.
4. Eine mit VOLIN initialisierte Platte, gleichgültig, ob privat oder gemeinschaftlich, kann als Zieldatenträger für jede physikalische Datensicherung dienen (Sicherungsprogramm FDDRL, Funktionen Copy oder Reload). Es ist jedoch zu beachten, dass der Plattentyp, die Kapazität und die Blockgröße dieses Datenträgers (2 KB) mit der des Quelldatenträgers übereinstimmt.

Erläuterungen zu den verschiedenen Kennsätzen:

1. Urlader-Block

   Der Urlader-Block ist der erste Block (2 KB) auf dem Datenträger. VOLIN setzt hier einen Pseudo-Urlader ein. Dieser löst bei einem Ladeversuch von dem Datenträger die Meldung `ILLEGAL LOAD FROM BS2000 DATA VOLUME vsn` und den Übergang des Systems in den Wartezustand aus.

2. Standard-Datenträger-Kennsatz (SVL)

   Der Standard-Datenträger-Kennsatz ist der zweite Block (2 KB) auf dem Datenträger. Er enthält Kennzeichen, die die Platte gegenüber dem System als einen bestimmten Datenträger ausweisen, Verweise auf die weiteren Kennsätze und Informationen über die Formatierung.

3. Format-5-Kennsatz (F5)

   Der Beginn dieses Kennsatzes ist im Standard-Datenträger-Kennsatz (SVL) verankert. Die Zahl der insgesamt von ihm belegten Blöcke (2 KB) ist plattentypspezifisch. Der Format-5-Kennsatz dient der Verwaltung des Speicherplatzes auf dem Datenträger. Die kleinste Belegungseinheit des Datenverwaltungssystems von BS2000 ist die Allocation-Unit (1 AU=6 / 8 / 64 KB, siehe Beschreibung "VOLIN Initialisieren von Plattenspeichern", Aufteilung fortlaufend, beginnend mit dem ersten 2 KB-Block). Wichtigster Bestandteil des Format-5-Kennsatzes ist eine Tabelle mit den freien und belegten Units des Datenträgers. Nach einem VOLIN-Lauf sind alle Units vor der F1-Startadresse sowie die Units des F1-Kennsatzes (siehe 4.) als belegt gekennzeichnet.

4. Format-1-Kennsatz (F1)

   Dieser Kennsatz beginnt an einer Allocation-Unit-Grenze hinter dem F5-Kennsatz. Seine Startadresse ist einheitlich für alle Platten eines Typs.

   Der Format-1-Kennsatz existiert nur auf privaten Datenträgern. Er nimmt den Katalog der auf dem Datenträger beginnenden Dateien auf. Der F1-Startadresse entspricht auf gemeinschaftlichen Datenträgern der Beginn des Speicherplatzes, der den Dateien zur Verfügung steht. Der F1-Kennsatz bildet zusammen mit dem F5-Kennsatz den so genannten VTOC-Bereich auf der Platte (VTOC=volume table of contents).

   Die Länge des F1-Kennsatzes kann die Systemverwaltung mit dem VOLIN-Lauf festlegen (über die F1SIZE-Anweisung, siehe "Beschreibung der Anweisungen"). VOLIN initialisiert den F1-Kennsatz mit leeren Katalogblöcken.

17.3 Programmablauf

VOLIN ist sowohl unter BS2000 als auch im Rahmen der Erstinstallation ablauffähig. Zusätzlich ist seine Funktionalität in das Dienstprogramm SIR integriert.

Der gesamte VOLIN-Anweisungsvorrat steht unter SIR als SDF-Oberfläche zur Verfügung, mit Ausnahme einiger veralteter Anweisungen. Dem Aufrufer wird somit entweder bei direktem Aufruf von VOLIN die alte ISP-Oberfläche angeboten oder bei Aufruf von SIR und Eingabe der entsprechenden VOLIN-Anweisungen alternativ eine SDF-Oberfläche.

In diesem Abschnitt wird der Ablauf von VOLIN unter BS2000 beschrieben. Besonderheiten, die sich beim Ablauf unter der Erstinstallation bzw. durch die Integration in SIR ergeben, sind im Kapitel „SIR Pubsets einrichten“ beschrieben. Dort wird auch die in SIR integrierte SDF-Syntax des VOLIN beschrieben. Das vorliegende Handbuch enthält jedoch eine Gegenüberstellung von ISP- und entsprechenden SDF-Anweisungen bzw. SDF-Operanden (siehe "Gegenüberstellung von ISP- und SDF-Anweisungen"). Wird im Textteil auf VOLIN-Anweisungen Bezug genommen, so sind immer ISP-Anweisungen gemeint.

VOLIN benutzt privilegierte Funktionen und kann nur unter der Benutzerkennung TSOS ablaufen. Die Ausführung unter einer anderen Benutzerkennung wird mit Meldung `NVL0001` und der sofortigen Beendigung quittiert.

VOLIN kann in einem Dialog- oder einem Batch-Auftrag aufgerufen werden.
Im Dialog wickelt VOLIN Fragen und Antworten, die die Platteninitialisierung betreffen, über das Datensichtgerät ab, sonst über denjenigen Bedienungsplatz, dem Aufgaben der Systemverwaltung zugeordnet sind.

In einem VOLIN-Lauf können beliebig viele Platten initialisiert werden. Das Initialisieren wird nacheinander ausgeführt. Jedes Initialisieren wird durch einen Satz von Anweisungen gesteuert, der von SYSDTA eingelesen wird. Nach einem Fehler wird das Initialisieren des augenblicklich bearbeiteten Datenträgers abgebrochen. Im Dialog kann anschließend ein neuer Satz von Anweisungen eingegeben werden; in einem Batch-Auftrag wird auch der VOLIN-Lauf abgebrochen (TERM MODE=ABNORMAL).

> ¡ Mehrere parallele VOLIN-Läufe können den laufenden BS2000-Betrieb beeinträchtigen. Sie sollten daher vermieden werden. Zumindest sollten die zu überschreibenden Platten nicht über Kanäle oder Steuerungen angesprochen werden, über die auch andere intensiv genutzte Platten betrieben werden.

Zuweisung der Datenträger

VOLIN fordert die zu initialisierenden Platten über die Geräteverwaltung des Systems an und belegt sie exklusiv im eigenen System.

VOLIN benötigt zur Zuweisung einer Platte mindestens die Angabe des Gerätetyps oder des mnemotechnischen Gerätenamens (MN) (Anweisungen DEVICE und UNIT). Soll eine bestimmte Platte initialisiert werden, muss VOLIN neben dem Gerätetyp oder dem MN auch die alte Archivnummer vorgegeben werden (mit der CHECK-Anweisung, siehe "Beschreibung der Anweisungen"). Kann der so spezifizierte Datenträger belegt werden, dann weist ihn die Geräteverwaltung des Systems direkt zu. Ist die Belegung nicht möglich, bricht VOLIN das Initialisieren ab.

Wird VOLIN nur der Gerätetyp vorgegeben, kann eine beliebige Platte dieses Typs zugewiesen werden („scratch volume“). VOLIN informiert die Systemverwaltung mithilfe des SVL über den montierten Datenträger und fragt, ob dieser überschrieben werden darf. Wird die Frage mit ja beantwortet (d.h. die Systemverwaltung stimmt dem Überschreiben zu), wird die montierte Platte initialisiert.
Wird die Frage mit nein beantwortet, können folgende Fälle eintreten:

- Handelt es sich um ein mit der Anweisung UNIT spezifiziertes Festplattengerät, wird die Initialisierung abgebrochen.

- Wurde kein Plattengerät mit der Anweisung UNIT spezifiziert, wird ein anderes Plattengerät angefordert.
- Verweigert der Operator die Bereitstellung einer weiteren Platte in der Antwort auf die Meldung der Geräteverwaltung, wird das Initialisieren abgebrochen.

Statt des Gerätetyps kann auch nur der MN vorgegeben werden. VOLIN ermittelt dann aus dem MN den Gerätetyp. Die weitere Vorgehensweise ist dieselbe wie bei gleichzeitiger Angabe von Gerätetyp und MN.

Bei der Bedienung des VOLIN aus SIR wird die Geräteauswahl nur noch per MN und nicht mehr per Gerätetyp unterstützt.

Es muss sichergestellt werden, dass das Initialisieren mit der Arbeit anderer Systeme koordiniert ist, die ebenfalls Zugriff zu dem mit dem Datenträger belegten Gerät haben.

Besonderheiten für Snap- und Clone-Units

Das Initialisieren einer Snap- oder Clone-Unit wird von VOLIN abgewiesen. Wenn eine Original-Unit mit dem Dienstprogramm VOLIN initialisiert wird, beendet VOLIN nach einer Operatorfrage ggf. alle Snap- oder Clone-Sessions mit dieser Unit und ggf. der SRDF-Target-Unit.
Dies entspricht `/STOP-SNAP-SESSION UNIT=...,SNAP-UNIT=*ALL` bzw.
`/STOP-CLONE-SESSION UNIT=...,CLONE-UNIT=*ALL.`

17.4 Bedienung

Kommandoebene

Das Programm wird unter TSOS gestartet mit `/START-VOLIN` bzw. `/VOLIN`.
Aus Kompatibilitätsgründen wird noch das Kommando `/START-EXECUTABLE-PROGRAM FROM-FILE=$VOLIN` unterstützt.
In diesen Fällen stehen die Anweisungen in ISP-Oberfläche zur Verfügung.

Alternativ dazu kann SIR unter TSOS mit `/START-SIR` bzw. `/SIR` gestartet werden. Anschließend kann VOLIN mit SDF-Oberfläche bedient werden.

Bedienplatzebene

VOLIN sendet Meldungen an den Bedienungsplatz (Berechtigungsschlüssel A) und nach SYSOUT. Wenn VOLIN in einem Batch-Auftrag abläuft, werden vom Bedienungsplatz auch Antworten auf Meldungen eingelesen, siehe Abschnitt „Programmablauf“.

VOLIN-Anweisungen

Jedes Initialisieren einer Platte wird durch einen Satz von Anweisungen gesteuert, den VOLIN von SYSDTA liest. Nach dem Einlesen eines Anweisungssatzes, prüft VOLIN dessen Konsistenz und Vollständigkeit. Wenn kein Fehler vorliegt, wird das Initialisieren durchgeführt und anschließend ein neuer Anweisungssatz angefordert.

Die Anweisungen werden VOLIN in Eingabezeilen übergeben. Jede Eingabezeile kann mehrere, durch Kommata getrennte Anweisungen enthalten, die Länge einer Eingabezeile darf aber 251 Zeichen nicht überschreiten. Entsprechend den Regeln der BS2000-Kommandosprache können Leerzeichen und Kommentare eingefügt werden.

Die erste Zeile eines Anweisungssatzes wird von VOLIN mit folgender Meldung angefordert:

```
NVL0002 ENTER VSN AND DEVICE TYPE AND/OR UNIT, OR TERMINATE VOLIN WITH 'END'
```

Alle weiteren Zeilen werden mit folgender Meldung angefordert:

```
NVL0003 ENTER ADDITIONAL INITIALIZATION PARAMETERS OR TERMINATE INPUT WITH 'EOT=YES'
```

Ein Anweisungssatz wird abgeschlossen mit einer Zeile, die allein oder neben anderen Anweisungen EOT=YES enthält. Eine Leerzeile (ETX) hat dieselbe Wirkung wie EOT=YES. Jede Zeile kann jede Kombination unterschiedlicher Anweisungen enthalten. Schon spezifizierte Anweisungen können in neuen Eingabezeilen wiederholt und auf diese Weise abgeändert werden.

VOLIN wird beendet, wenn die Meldung `NVL0002` mit END beantwortet wird.
Wird dieselbe Antwort auf Meldung `NVL0003` gegeben, dann generiert VOLIN zuerst eine EOT=YES-Bedingung und arbeitet den so abgeschlossenen Anweisungssatz ab. Anschließend wird kein neuer Anweisungssatz angefordert, sondern der Lauf beendet.

Dateiende (EOF) auf SYSDTA hat dieselbe Wirkung wie END.

17.5 Anweisungen

Die Anweisungen von SIR sind im Abschnitt „Anweisungen“ beschrieben.

17.5.1 Übersicht über die Anweisungen von VOLIN

Format	Bedeutung
[VSN=vsn]	Archivnummer nach dem Initialisieren.
[DEVICE=device-type]	Gerätetyp der Platte (muss angegeben werden, wenn keine UNIT spezifiziert wurde).
[C[HECK]= SCRATCH / vsn]	Archivnummer vor dem Initialisieren oder SCRATCH für eine beliebige Platte.
[UNIT=mn]	mnemotechnischer Gerätename; anzugeben, wenn das Initialisieren auf einem bestimmten Gerät erfolgen soll.
[F[MT]= YES / NO / n]	Initialisierung mit (Anzahl = n) oder ohne Überschreiben.
[UNAME=name]	Eigentümername, der im SVL hinterlegt werden soll (wird vom System nicht ausgewertet).
[{ F1SIZE / VTOC } =n]	Anzahl der PAM-Blöcke für den Format-1-Kennsatz.
[FORMAT= { UNCHANGED / K / NK([PHYSICAL-BLOCK-SIZE=] 2K(ALLOCATION-UNIT = 6 / 8 / 64)) }]	Festlegung des Volumetyps.
[EOT=Y[ES]]	Abschluss des Anweisungssatzes mit der aktuellen Eingabezeile.
[END / **H**ALT]	Beendigung des VOLIN-Laufes nach Verarbeitung des aktuellen Anweisungssatzes.

17.5.2 Beschreibung der Anweisungen

VSN = vsn
Archivnummer (muss angegeben werden).

Der maximal 6-stellige alphanumerische Wert wird in das Archivnummernfeld des Standard-Datenträger-Kennsatzes (SVL) geschrieben. Soll der Datenträger zur späteren Verwendung als gemeinschaftlicher Datenträger initialisiert werden, dann ist abhängig von der Länge der Pubset-Kennung Folgendes zu beachten:

- einstellige Kennung des Pubsets
  Die Archivnummer beginnt mit den Zeichen PUB. Das Vierte ist die Kennung des Pubsets. Die letzten zwei Zeichen müssen Ziffern sein, z.B. PUBA01. Nullen an diesen Stellen kennzeichnen einen Datenträger als Systemresidenz, z.B. PUBA00.
- zwei- bis vierstellige Kennung des Pubsets
  Die ersten Zeichen sind die Kennung des Pubsets. Das Ende der Kennung wird durch einen Punkt an der darauf folgenden Stelle gekennzeichnet. Für die restlichen Stellen gilt folgende Regelung:
  - NK2(8K,64K)-Platten:
    Die Stellen können Ziffern (0..9) und Buchstaben (A..Z) enthalten.
    Beispiele
    Archivnummer mit zweistelliger Pubsetkennung: RZ.90Z
    Archivnummer mit dreistelliger Pubsetkennung: KH2.W0
    Archivnummer mit vierstelliger Pubsetkennung: AFLN.8
  - NK2(6K)-Platten:
    Die letzte Stelle muss eine Ziffer oder ein Buchstabe aus der Menge A...V sein. Alle verbleibenden Stellen müssen Nullen enthalten. *Beispiele*
    Archivnummer mit zweistelliger Pubsetkennung: AB.005
    Archivnummer mit dreistelliger Pubsetkennung: ABC.0F
    Archivnummer mit vierstelliger Pubsetkennung: ABCD.0

DEVICE = device-type
Gerätetyp (muss angegeben werden, wenn keine UNIT spezifiziert wurde). Typ der Platte, der initialisiert/repariert werden soll.
Die gültigen Werte für diese Anweisung: siehe Handbuch „Systeminstallation“ [7].

Die DEVICE-Angabe kann entfallen, wenn das Gerät über die UNIT-Anweisung spezifiziert wurde.

CHECK = SCRATCH
Angabe (Standardwert), wenn kein bestimmter Datenträger initialisiert werden soll. VOLIN informiert über den zugewiesenen Datenträger und fragt, ob er überschrieben werden darf.

CHECK = vsn
Archivnummer (maximal 6 Stellen).
Alte Archivnummer der zu initialisierenden Platte. Nur ein Datenträger, der diese Archivnummer trägt, wird initialisiert. Der Datenträger wird nach der Zuweisung ohne Rückfrage beim Benutzer überschrieben.

UNIT = mn
Mnemotechnischer Name.

mn ist der 2- bzw. 4-Zeichen lange Name, der einem bestimmten Gerät bei der Hardware-Generierung zugeordnet worden ist. Diese Anweisung muss dann eingegeben werden, wenn der Anwender ein bestimmtes Gerät zum Initialisieren vorgeben will. Wenn die Anweisung UNIT=mn nicht angegeben ist, wählt das System bzw. der Operator durch Beantworten der MOUNT-Meldung ein verfügbares Gerät des richtigen Gerätetyps aus.

Die DEVICE-Angabe kann entfallen, wenn das Gerät über die UNIT-Anweisung spezifiziert wurde.

Die Initialisierung wird nicht durchgeführt, wenn sich die Angaben von DEVICE und UNIT widersprechen.

FMT = YES
Die Platte soll bei der Initialisierung einmal mit einem bestimmten Bitmuster überschrieben werden (Standard). YES ist gleichbedeutend mit FMT=1 (siehe beim Operanden FMT=n).

FMT = NO
Die Initialisierung der Platte soll ohne Überschreiben durchgeführt werden.

FMT = n
Die Platte soll bei der Initialisierung n-mal mit einem bestimmten Bitmuster überschrieben werden. Für n kann eine Dezimalzahl von 0 bis 10 angegeben werden.
0 ist gleichbedeutend mit FMT=NO, kein Überschreiben. Bei jedem Überschreiben wird ein bestimmtes Prüfmuster aufgezeichnet. Bei einem neuen Durchgang wird jeweils zum zyklisch nächsten Prüfmuster gewechselt.

> ¡ FMT wird für Thin Devices ignoriert. Thin Devices können nur mit SHC-OSD erkannt werden.

UNAME = name
Max. 10 alphanumerische Zeichen langer Eigentümername, der in das Namensfeld des SVL geschrieben wird. Das erste Zeichen muss ein Buchstabe oder eines der Zeichen $, # oder @ sein. Wenn nichts angegeben ist, werden Leerzeichen verwendet.

F1SIZE = n
Dezimalzahl (1 - 32766).
Diese Anweisung ist nur für private Datenträger von Bedeutung. Die Dezimalzahl ist die Anzahl von PAM-Blöcken, die für den Format-1-Kennsatz reserviert werden soll. Der angegebene Wert wird, wenn notwendig, aufgerundet, um eine ganzzahlige Anzahl von Units zu ergeben. In einem Block können maximal etwa 16 Dateien katalogisiert werden. Wenn nichts angegeben ist, werden 1000 PAM-Blöcke auf einem privaten Datenträger für den Format-1-Kennsatz zugewiesen. (Aus Kompatibilitätsgründen kann statt F1SIZE auch das Schlüsselwort VTOC verwendet werden.)

FORMAT = UNCHANGED
Der Volumetyp des Datenträgers soll beibehalten werden. Dieser Wert setzt ein gültiges SVL des Datenträgers voraus. Andernfalls wird standardmäßig der Wert K angenommen.

FORMAT = K
Der Datenträger soll mit PAM-Schlüssel benutzbar sein.

FORMAT = NK(PHYSICAL-BLOCK-SIZE = 2K(ALLOCATION-UNIT = 6/8/64))
Der Datenträger soll ohne PAM-Schlüssel benutzbar sein. Es sind nur die NK-Zugriffsmethoden des DVS zulässig.

Die Blockgröße soll 2 KB betragen, die kleinste Allokierungseinheit, d.h. die kleinste vom System zur Verfügung gestellte Speichereinheit 6, 8 oder 64 KB.
Voreinstellung für ALLOCATION UNIT: 6 KB.

PHYSICAL-BLOCK-SIZE = 2K
Länge der minimalen TRANSFER UNIT (TU), d.h. Größe des kleinstmöglichen Datenblocks in KB, der zwischen Platte und Hauptspeicher transferiert werden kann.

ALLOCATION-UNIT = 6 / 8 / 64
Minimale Allokierungseinheit in KB, d.h. kleinstmögliche Speicherplatzeinheit, die beim Anlegen einer Datei vom System zur Verfügung gestellt wird.

Einschränkungen
Folgende Angaben sind nur für gemeinschaftliche Platten (Public Volumes) zulässig:
FORMAT=NK(PHYSICAL-BLOCK-SIZE=2K(ALLOCATION-UNIT=8))
FORMAT=NK(PHYSICAL-BLOCK-SIZE=2K(ALLOCATION-UNIT=64))

EOT = YES
Der Anweisungssatz soll mit der laufenden Eingabezeile abgeschlossen und das Initialisieren durchgeführt werden. (Alle Anweisungen in der laufenden Eingabezeile werden mitberücksichtigt.)

END
VOLIN soll beendet werden, nachdem der laufende Anweisungssatz abgearbeitet worden ist. (Aus Kompatibilitätsgründen ist statt END auch H oder HALT möglich. Neben END bzw. HALT werden keine weiteren Anweisungen in der Eingabezeile akzeptiert.)

17.5.3 Gegenüberstellung von ISP- und SDF-Anweisungen

Alle relevanten VOLIN-Anweisungen stehen als SDF-Oberfläche zur Verfügung. Voraussetzung ist der Aufruf des Programms SIR, das das Subsystem VOLIN implizit startet. Die genaue Beschreibung der SDF-Anweisungen von SIR finden Sie im Abschnitt „Anweisungen“.

Nachfolgend wird zur leichteren Orientierung eine kurze Gegenüberstellung der ISP- und SDF-Anweisungen bzw. -Operanden gegeben. Dabei ist Folgendes zu beachten:

- Die SDF-Anweisungen INITIALIZE-PRIVATE-VOLUME und INITIALIZE-PUBLIC-VOLUME decken die Funkionen Kennsatzgenerierung ab, jeweils getrennt für Privat- und Public-Platten.
- Die weiteren ISP-Anweisungen werden jeweils als Operanden in den dafür relevanten SDF-Anweisungen angeboten.
- Einige ISP-Anweisungen, die überflüssig oder nur noch für veraltete Hardware relevant sind, wurden nicht mehr in die SDF-Oberfläche implementiert.

Die Ziffern 1 und 2 in der rechten Spalte der folgenden Tabelle stellen folgende Anweisungen dar:

1 = `INITIALIZE-PRIVATE-VOLUME`

2 = `INITIALIZE-PUBLIC-VOLUME`

ISP-Anweisung	SDF-Operand / -Anweisung	in Anweisung 1	in Anweisung 2
VSN	VOLUME	x	x
DEVICE	nicht unterstützt (Auswahl per UNIT)		
CHECK	OVERWRITE-DISK	x	x
UNIT	UNIT	x	x
FMT	FUNCTION = *FORMAT-DISK	x	x
	FUNCTION = *CREATE-LABELS-ONLY	x	x
UNAME	nicht unterstützt		
F1SIZE	F1-LABEL-SIZE	x	
FORMAT	FORMAT	x	x
EOT	implizit mit Anweisungs-Ende		
END	Anweisung END des SIR		

18 Literatur

Die Handbücher finden Sie im Internet unter *http://bs2manuals.ts.fujitsu.com*. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie in auch gedruckter Form bestellen.

[1] **BS2000 OS DX**
Kommandos
Benutzerhandbuch

[2] **BS2000 OS DX**
Diagnosehandbuch
Benutzerhandbuch

[3] **BS2000 OS DX**
DVS-Makros
Benutzerhandbuch

[4] **BS2000 OS DX**
Einführung in das DVS
Benutzerhandbuch

[5] **BS2000 OS DX**
Einführung in die Systembetreuung

Benutzerhandbuch

[6] **BS2000 OS DX**
Makroaufrufe an den Ablaufteil
Benutzerhandbuch

[7] **BS2000 OS DX**
Systeminstallation
Benutzerhandbuch

[8] **BS2000 OS DX**
System Managed Storage
Benutzerhandbuch

[9] **BS2000 OS DX**
Performance-Handbuch
Benutzerhandbuch

[10] **Fujitsu Server BS2000 SE Serie**
Bedienen und Verwalten
Benutzerhandbuch

[11] **openNet Server**
BCAM
Benutzerhandbuch

[12] **BINDER**
Binder in BS2000
Benutzerhandbuch

[13] **BLSSERV**
Bindelader-Starter in BS2000
Benutzerhandbuch

[14] **EDT** (BS2000)
Anweisungen
Benutzerhandbuch

[15] **IMON**
Installationsmonitor
Benutzerhandbuch

[16] **JV**
Jobvariablen
Benutzerhandbuch

[17] **LMS**
SDF-Format
Benutzerhandbuch

[18] **MAREN**
Bandverwaltung in BS2000
Benutzerhandbuch

[19] **POSIX**
Grundlagen für Anwender und Systemverwalter

Benutzerhandbuch

[20] **SDF**
Dialogschnittstelle SDF
Benutzerhandbuch

[21] **SECOS**
Security Control System - Beweissicherung
Benutzerhandbuch

[22] **SHC-OSD**
Storage Manangement für BS2000
Benutzerhandbuch

[23] **VM2000**
Virtuelles Maschinensystem
Benutzerhandbuch