Deutsch

Fujitsu Software

BS2000 OS DX
Einführung in die Systembetreuung

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

Einführung in die Systembetreuung 12
1 Einleitung 13
1.1 Zielgruppen und Zielsetzung des Handbuchs 14
1.2 Konzept des Handbuchs 15
1.3 Änderungen gegenüber dem Vorgänger-Handbuch 17
1.4 Darstellungsmittel 18
1.5 Aufgaben der Systemverwaltung 19
1.6 Aufgaben des Operating 20
1.7 Automatisierung 21
2 Systemeinleitung und -beendigung 22
2.1 Überblick über die Abläufe bei der Systemeinleitung 23
2.1.1 Zeitbestimmung bei der Systemeinleitung 31
2.1.2 Format der Meldungen bei der Systemeinleitung 33
2.1.3 Hinweise zur Systemeinleitung 34
2.1.4 Systemeinleitung an SU x86 39
2.2 Arten der Systemeinleitung 40
2.2.1 FAST-Startup 41
2.2.2 AUTOMATIC-Startup 43
2.2.3 DIALOG-Startup 44
2.2.4 Systemeinleitung mit wechselndem Modus 48
2.2.5 Die Auswahl von Startup-Typ und Katalog-Rekonfiguration 49
2.2.6 Beispiel für einen DIALOG-Startup 52
2.3 Systemkorrekturen 57
2.3.1 Funktion und Aufbau einer Rep-Datei 58
2.3.2 Rep-Sätze 61
2.3.3 Rep-Zwischenspeicherung 66
2.3.4 RMS: Montage und Lieferung von Reps 68
2.4 Systembeendigung 69
2.4.1 Planmäßige Beendigung 70
2.4.2 Außerplanmäßige Beendigung 72
2.4.3 Automatischer Restart 73
3 Parameterservice 74
3.1 Auswahl der Parameterdatei 75
3.2 Aufbau und Inhalt einer Parameterdatei 77
3.3 Starten des Abrechnungssystems (ACCOUNT) 80
3.4 Inbetriebnahme der dynamischen Subsystemverwaltung (DSSM) 82
3.5 Steuerung der Systemzeit (GTIME) 83

3.6 Parametersatz IOCONF 88
3.6.1 Anweisung MOD-IO-UNIT 89
3.6.2 Anweisung MOD-IO-PATH 91
3.7 Voreinstellungen für NK-ISAM (ISAM) 92
3.8 Festlegung des TSN-Modus (JMS) 95
3.9 Speicherverwaltung (MEMORY) 96
3.10 Konfiguration und Meldungsunterdrückung an Konsolen (OPR) 100
3.10.1 Anweisung ADD-CMD-ENTRY 102
3.10.2 Anweisung DEFINE-CONSOLE 104
3.10.3 Anweisung SET-CMD-CODE 105
3.10.4 Anweisung SET-CODE 106
3.10.5 Anweisung SET-FILTER 109
3.10.6 Anweisung SET-MSG-SUPPRESSION 110
3.11 Auswahl des Seitenwechselbereichs beim Startup (PAGING) 111
3.12 Snapshot-Initialisierung (SNAP) 113
3.13 Voreinstellung der Systemparameter (SYSOPT-CLASS2) 114
3.14 Änderung von IPL-Optionen (SYSOPT-IPL) 117
4 Speicherverwaltung 120
4.1 Virtueller Adressraum und virtuelle Speicherklassen 121
4.2 Hauptspeicher 123
4.2.1 Seitenverwaltungs-Algorithmus 124
4.2.2 Big Pages für CISCFW-Kompilate (SU x86) 126
4.3 Seitenwechselbereich (Paging-Area) 129
4.3.1 Einrichten, Zuweisen, Freigabe und Löschen von Paging-Dateien 131
4.3.2 Auswahl des Seitenwechselbereichs bei Startup 134
4.4 Maßnahmen zur Verhinderung von Sättigungszuständen 135
4.4.1 Hauptspeicher 137
4.4.2 Systemadressraum 138
4.4.3 Seitenwechselspeicher 139
5 Geräteverwaltung 141
5.1 Bestandteile von Konfigurationen 142
5.1.1 Hardware-Einheiten 144
5.1.2 Logische, rekonfigurierbare Verbindungen 147
5.2 Rekonfiguration 149
5.2.1 Rekonfiguration bei Multiprozessoren 150
5.2.2 Weg- und Zuschalten der Komponenten CPU, CHN, CTL und DVC 151
5.2.3 Wirkung der Rekonfigurationskommandos 153
5.2.4 Besonderheiten für Magnetband- und Plattengeräte 155
5.3 Dynamische I/O-Konfigurationsänderung (SU /390) 157
5.4 Dynamische I/O-Konfigurationsänderung (SU x86) 162

5.5 Rekonfiguration von Extra- und Spare-CPUs 163
5.5.1 Funktionalität im native BS2000-Betrieb 164
5.5.2 Funktionalität im Gastsystem unter VM2000 166
5.6 NDM - Betriebsmittelbelegung und -reservierung 167
5.6.1 Task-Belegung von Datenträgern 169
5.6.2 Systembelegung von Platten 171
5.6.3 Vorgaben für die Privatplattenbelegung 172
5.6.4 Steuerung der Betriebsmittelreservierung 174
5.6.5 Hilfestellung im Umgang mit NDM 176
5.7 Datenträger-Überwachung 183
5.7.1 Geräteauswahlmechanismus für Bandgeräte 184
5.8 Parallel Access Volumes (PAV, SU /390) 186
5.9 Verwaltung privater Datenträger 190
5.9.1 Bandverwaltung mit MAREN 191
5.9.2 Einsatzmöglichkeiten privater Platten 193
5.10 IORM: Steuerung von I/O-Ressourcen 194
5.11 Anzeigen und prüfen der SAN-Konfiguration 195
6 BS2000-Benutzerverwaltung 196
6.1 Struktur eines Benutzerkatalogs 197
6.2 Führen des Benutzerkatalogs 198
6.3 Sicherungskonzept für Benutzerkataloge 200
6.3.1 Sichern des aktuellen Benutzerkatalogs eines Pubsets 201
6.3.2 Wiedereinspielen eines gesicherten Benutzerkatalogs 202
6.3.3 Rekonstruktion der Datei SYSSRPM 203
6.3.3.1 Rekonstruktion durchführen 204
6.3.3.2 Logging-Datei 208
6.4 Benutzerkatalog und SMS-Konzept 211
6.5 Bulletin-Datei (Logon-Informationsdatei) 212
7 POSIX-Benutzerverwaltung 215
8 Dateiverwaltung 218
8.1 Dateikatalog 219
8.1.1 Aufbau eines Dateikataloges 221
8.1.2 Katalogdateien erweitern 225
8.1.3 Performance bei Katalogzugriffen verbessern 226
8.1.4 Zugriffssperren aufheben 227
8.2 ACS: Alias-Katalogsystem 228
8.3 ADM-PFA: Administrator Performant File Access 229
8.4 Versenden von BS2000-Dateien per E-Mail 230
8.5 Unicode in BS2000 232
9 Pubset-Verwaltung 233

9.1 Pubset-Konzept 236
9.1.1 Pubset-Arten 237
9.1.1.1 Home-Pubset 238
9.1.1.2 Benutzer-Pubset (Daten-Pubset) 239
9.1.1.3 Paging-Pubset 240
9.1.1.4 Pubsets mit großen Objekten 241
9.1.2 Pubset-Namen (VSN) 242
9.1.2.1 VSN in PUB-Notation 243
9.1.2.2 VSN in Punkt-Notation 244
9.1.2.3 Doppelpunkt-Notation für Spiegelplatten 245
9.1.2.4 Sonderfälle 246
9.1.2.5 Pubset-Adressierung 247
9.1.3 SF- und SM-Pubsets 248
9.1.3.1 Aufbau eines SF-Pubsets 249
9.1.3.2 Aufbau eines SM-Pubsets 250
9.1.3.3 Pubset-Metadaten 254
9.1.3.4 Benutzerkontingent-Konzept 256
9.2 Pubset-Eigenschaften 259
9.2.1 Physikalische Merkmale 260
9.2.1.1 Plattenformate 261
9.2.1.2 Physikalische Merkmale von Volume-Sets 262
9.2.2 Globale Eigenschaften 263
9.2.3 Benutzerspezifische Eigenschaften 264
9.2.4 Volume-Set- und Volume-spezifische Eigenschaften 265
9.3 Pubsets einrichten 266
9.3.1 SF-Pubsets einrichten 267
9.3.2 SM-Pubsets einrichten und ersetzen 268
9.3.2.1 Einrichten und Erweitern von Volume-Sets und SM-Pubsets 269
9.3.2.2 SF-Pubsets in SM-Pubsets umwandeln 270
9.3.3 Pubsets mit PVSREN erzeugen 272
9.4 Pubsets konfigurieren 273
9.4.1 MRSCAT-Einträge verwalten 274
9.4.2 Globale Pubset-Eigenschaften festlegen 275
9.4.3 Benutzer-spezifische Einstellungen festlegen 276
9.4.3.1 Standard-Katalogkennung (Standard-Pubset) zuordnen 277
9.4.3.2 Grenzwerte und Zugriffsrechte festlegen 278
9.4.4 Eigenschaften eines Volume-Sets festlegen 279
9.4.4.1 Verfügbarkeit definieren 280
9.4.4.2 Performance-Profil definieren 281
9.4.4.3 Nutzungsart festlegen 283
9.4.4.4 Nutzungseinschränkungen 284

9.4.4.5 Sättigungsschwellwerte für Volume-Sets und SF-Pubsets festlegen 285
9.4.4.6 Datei-Services definieren 286
9.4.4.7 Benutzerkontingente festlegen 288
9.5 Pubsets in/außer Betrieb nehmen 289
9.5.1 Pubset in Betrieb nehmen 291
9.5.2 Besonderheiten der Inbetriebnahme 292
9.5.3 Pubset außer Betrieb nehmen 294
9.6 Pubsets administrieren 295
9.6.1 Pubsets rekonfigurieren 296
9.6.1.1 Statische Konfigurationsdaten von SM-Pubsets ändern 297
9.6.1.2 Dynamische Konfigurationsdaten von SF- und SM-Pubsets ändern 298
9.6.1.3 Ersetzen und Entfernen von SF-Pubsets mit gleichem Namen 301
9.6.1.4 Bestehende Pubsets in große Pubsets umwandeln 302
9.6.2 Zugriffsberechtigung und Benutzerkatalog verwalten 303
9.6.3 Speicherplatz-Sättigung überwachen 304
9.6.4 Plattenspeicherplatz reorganisieren 306
9.7 Shared-Pubsets 307
9.7.1 Shared-Pubset-Verbund einrichten 308
9.7.2 Shared-Pubset überwachen und Fehler beheben 310
9.7.2.1 Pubset-Locks manuell beheben 311
9.7.2.2 Verhalten bei Systemausfall 314
9.7.3 Shared Pubsets im XCS-Verbund 316
9.8 Besonderheiten für Standby-Pubsets 317
9.8.1 Standby-Pubsets erstellen 318
9.8.1.1 Einzelnen Standby-Pubset einrichten 319
9.9 SYSEAM-Speicherplatz auf Pubsets verwalten 324
9.10 Speed Catalog Access (SCA) einsetzen 326
9.11 Hardware-Fehler auf Pubsets erfassen und beseitigen 329
10 Net-Storage-Verwaltung 331
10.1 Überblick 332
10.2 Begriffe 333
10.3 Interoperabilität auf Net-Storage 336
10.4 Anbindung von BS2000 an Net-Storage 338
10.5 Zugriff auf Net-Storage 339
10.5.1 Zugriff von BS2000 auf Net-Storage 340
10.5.2 Zugriff offener Systeme auf Net-Storage 342
10.6 Vorbereitungen auf Net-Server und Net-Client 343
10.7 Net-Storage mit BS2000 verbinden 344
10.8 Verwalten von Net-Storage in BS2000 346
10.9 Net-Storage von BS2000 trennen 348

10.10 Randbedingungen 349
10.10.1 BS2000 allgemein 350
10.10.2 Shared-Pubsets 351
10.10.3 Platten-Spiegelung 352
10.10.4 Interoperabilität 354
10.10.5 Net-Storage-Volumes ab BS2000 V21 356
11 Job- und Task-Steuerung 358
11.1 Job-Steuerung 359
11.1.1 Konzept der Job-Klassen 364
11.1.2 Job-Streams, Job- und Klassen-Scheduler 367
11.1.2.1 System-Job-Scheduler 369
11.1.2.2 Standard-Job-Scheduler 370
11.1.2.3 Klassen-Scheduler 378
11.1.3 Die Job-Steuerung während der Systemeinleitung 379
11.1.4 Job-Steuerung im Data Center 380
11.1.5 Unterbrechungsfreie Zeitumstellung 381
11.1.6 JMP: Rekonstruktion von Batch-Jobs 382
11.2 Task-Steuerung 383
11.2.1 Prioritätenkonzept und Warteschlangen 391
11.2.2 Zuteilung von Ressourcen 395
11.2.3 TANGRAM: Verwaltung von affinen Task-Gruppen 400
11.3 Zeitlimitierungen in BS2000 402
11.4 PCS: Performance-Überwachungssystem 405
12 Sicherheit 406
12.1 Zugangsschutz 407
12.2 Zugriffsschutz 408
12.3 Zugriffsschutz in BS2000 413
12.3.1 Kennwörter und Schutzfristen 414
12.3.2 Dateiverschlüsselung 415
12.3.3 Standard-Zugriffskontrolle (ACCESS/USER-ACCESS) 418
12.3.4 Einfache Zugriffskontroll-Liste (BACL) 419
12.4 Privilegien 421
12.4.1 Privilegien der Benutzerkennung TSOS 423
12.4.2 Privilegien für das Operating 425
12.4.3 Beschreibung der Privilegien 426
12.4.4 Privilegienverteilung 434
12.5 Betriebsmittel für Benutzer begrenzen 436
12.6 Erfüllung von Sicherheitsanforderungen durch SECOS 438
13 Datensicherung 440
13.1 Kriterien zur Auswahl von Sicherungskonzepten 441
13.2 Datensicherung mit Snapsets 446

13.2.1 Snapset-Betrieb vorbereiten 449
13.2.2 Sicherungsbetrieb mit Snapsets 454
13.2.3 Rekonstruktion von Dateien und Jobvariablen 459
13.2.4 Rekonstruktion von Pubsets 462
13.3 Rekonstruktion von Dateien und Datenträgern 464
13.4 Archivsysteme 467
14 Accounting 468
14.1 Steuerung des Abrechnungssystems 469
14.2 Abrechnungsdatei 471
14.2.1 Merkmale und Inhalt der Abrechnungsdatei 473
14.2.2 Arten von Abrechnungsdaten 475
14.3 Übersichten über die Abrechnungssätze 478
15 Systemmeldungen 485
15.1 BS2000-Meldungssystem 486
15.2 Meldungsdateien 487
15.2.1 Systemmeldungsdateien 488
15.2.2 Benutzermeldungsdateien 489
15.3 Garantierte Meldungen 490
15.4 Meldungssuche 495
15.5 Struktur einer Meldungseinheit 497
15.6 Meldungsschlüssel 498
15.6.1 Kennbuchstabe für die Sprache 499
15.6.2 Attribute der Meldungseinheit 500
15.6.3 Meldungstext 505
15.6.4 Bedeutungs-/Maßnahmetext 508
15.7 Meldungsausgabe 509
15.7.1 Ausgabeformen der Meldungen 510
15.7.2 Kommando HELP-MSG-INFORMATION 512
15.7.3 Meldungen, die eine Antwort erfordern 513
15.7.4 Ausgabe von Meldungen an Benutzerprogramme 514
15.7.5 Ausgabe von Meldungen in S-Variablen 515
16 Operatorfunktionen 518
16.1 Benutzertask mit OPERATING-Privileg 520
16.1.1 Bereitstellung einer Kennung für das Operating 522
16.1.2 Nutzung des Ereignisstrom-Service für Operating aus Benutzertasks 523
16.1.3 Nutzung des Ereignisstrom-Service für beliebige Benutzertasks 526
16.2 Kommandoeingabe über Konsole 528
16.3 Nachrichten 529
16.3.1 Emergency-Nachrichten 530
16.3.2 Verständigung zwischen System und Operator 531

16.3.3 Nachrichtenformate 535
16.3.4 Steuerung der Nachrichtenzustellung 539
16.3.5 Nachrichtenaustausch der Operator 543
16.4 Verwendung mehrerer Konsolen 546
16.4.1 Hauptkonsole und Nebenkonsolen 548
16.4.2 Ersatz-Konsolen 550
16.4.3 Aufgabengebiete und ihre Aufteilung auf Konsolen 551
16.5 Rückmeldungen 564
16.5.1 Kommandobeendigungsmeldungen 565
16.5.2 Rückmeldungen an physikalischen Konsolen 567
16.5.3 Rückmeldungen an physikalischen und logischen Konsolen 568
16.6 Ablösung der Kommandos STATUS MSG und ASR 570
17 Automatisierung von Operatorfunktionen 572
17.1 Berechtigte Benutzerprogramme mit Operatorfunktionen 573
17.1.1 Anschlüsse mit generierten Berechtigungsnamen 577
17.1.2 Anschlüsse mit dynamischen Berechtigungsnamen 578
17.1.3 Austausch von Nachrichten 582
17.1.4 Nachrichtenformate 586
17.2 Operator-Spezialkommandos in berechtigten Benutzerprogrammen 592
17.2.1 Kommandodefinition 594
17.2.2 Nachrichtenformate 596
17.3 Softwareprodukte OMNIS und PROP-XT 601
17.4 Kommandodateien für den Operator 602
17.4.1 Ausführen und Abbrechen einer Kommandodatei 604
17.4.2 Aufbau von Kommandodateien 607
17.5 Ausübung von Systemverwaltungsfunktionen durch den Operator 610
18 Systemzeit-Verwaltung 613
18.1 Systemzeit 614
18.1.1 TODR als HW-Uhr 615
18.1.2 TODR-Epoche 616
18.1.3 TODR-Korrekturwerte 620
18.1.4 Synchronisation der Systemzeit mit externen Zeitgebern oder im Verbund . 621
18.1.5 Subsystem GET-TIME 623
18.2 SVP-Zeit 624
18.3 Initialisierung der Systemzeit 625
18.4 Synchronisation der Systemzeit 626
18.4.1 Synchronisation mit der SVP-Uhr auf SU /390 627
18.4.2 Synchronisation mit der Zeit des Trägersystems auf SU x86 628
18.4.3 Synchronisation in einem NTP-Verbund 629
18.4.4 Synchronisation in einem XCS-Verbund 631
18.4.5 Generelle Empfehlung für den XCS-Verbund 633

18.4.6 Synchronisationsanzeigen bei /SHOW-SYSTEM-INFORMATION 634
18.5 Unterbrechungsfreie Sommer-/Winterzeitumstellung 635
18.6 Systemstart mit spezieller Systemzeit 637
18.7 TASKDATE: Testen in simulierter Zeit 638
19 Anhang 639
19.1 Zeichensatz bei Ein-/Ausgabe über Konsole 640
19.2 Übersicht zu Testprivilegien 641
19.2.1 Testprivilegien bei AID 642
19.2.2 Testprivilegien bei weiteren Software-Diagnoseprodukten 645
20 Literatur 646

Einführung in die Systembetreuung

1 Einleitung

- Zielgruppen und Zielsetzung des Handbuchs
- Konzept des Handbuchs
- Änderungen gegenüber dem Vorgänger-Handbuch
- Darstellungsmittel
- Aufgaben der Systemverwaltung
- Aufgaben des Operating
- Automatisierung

1.1 Zielgruppen und Zielsetzung des Handbuchs

Das Handbuch „Einführung in die Systembetreuung“ wendet sich an die Systemverwaltung und das Operating des Betriebssystems BS2000.
Es soll helfen, das Betriebssystem zu verwalten, zu steuern und zu überwachen.

1.2 Konzept des Handbuchs

Das Handbuch „Einführung in die Systembetreuung“ informiert über Themen zur Verwaltung und Überwachung von BS2000.

Folgende Themen werden nicht in diesem Handbuch, sondern in eigenen Handbüchern beschrieben:

- „SDF (System Dialog Facility)“,
  siehe die Handbücher „Dialogschnittstelle SDF“ [43] und „SDF-A“ [44].
- „Verwaltung von Subsystemen“,
  siehe Handbuch „Verwaltung von Subsystemen (DSSM/SSCM)“ [18].
- „Produktlieferung und Produktinstallation“, siehe Handbuch „IMON“ [25].
- „Erstellung und Analyse von Diagnoseunterlagen“,
  „Fehler- und Protokolldateien (HEL, SERSLOG, CONSLOG, RESLOG),
  siehe das „Diagnosehandbuch“ [14].

Die in diesem Handbuch genannten Kommandos bzw. Makros werden in den Handbüchern „Kommandos“ [27] bzw. „Makroaufrufe an den Ablaufteil“ [30] und „DVS-Makros“ [20] beschrieben, wenn nicht auf ein anderes Handbuch verwiesen wird.

Architektur der SE Server

Ein FUJITSU Server BS2000 der SE Serie (kurz: SE Server) besteht im Maximalausbau aus folgenden Komponenten:

- Server Units (SU /390 und SU x86)
- Application Units (AU)
- Peripherie (Storage)
- Management Unit (MU) mit SE Manager
- Net Unit, für SU /390 mit HNC

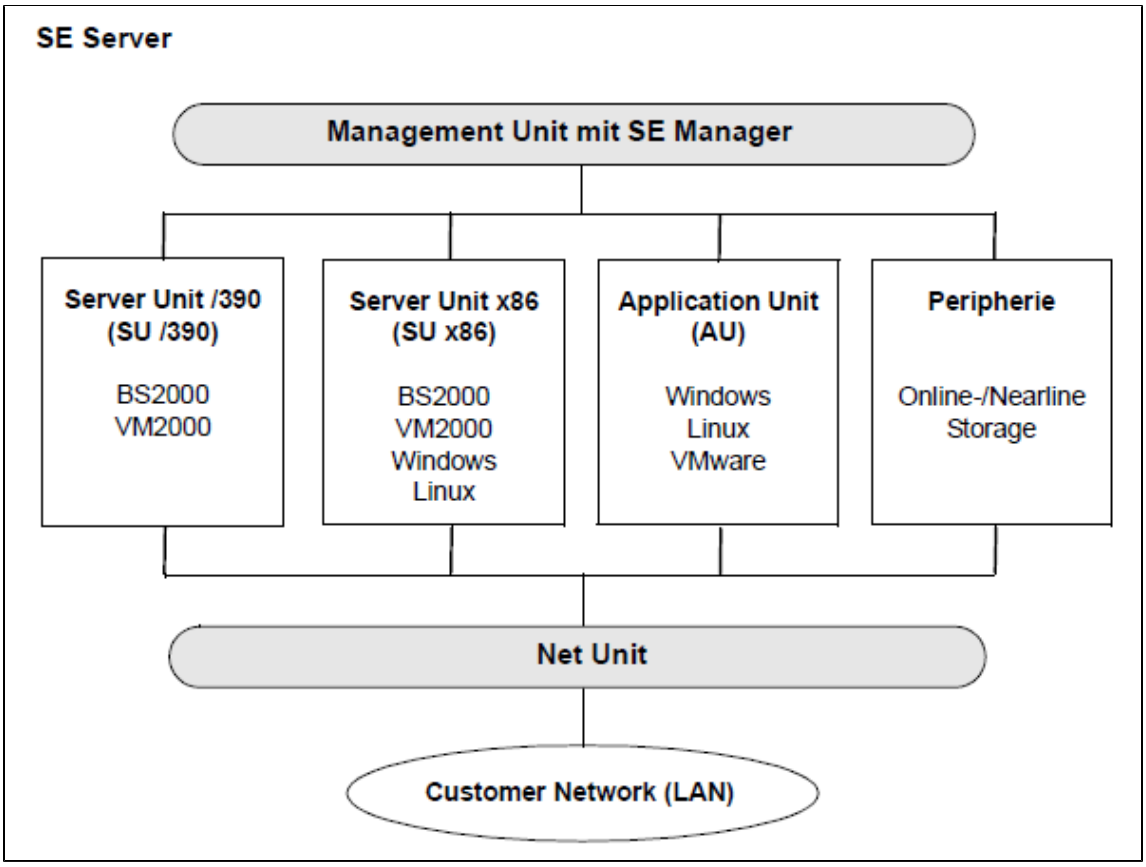


Bild 1: Architektur der SE Server

Mit dem SE Manager können Sie alle Komponenten des SE Servers von der Management Unit aus zentral bedienen und verwalten. Der SE Manager bietet dazu eine komfortable, web-basierte Benutzeroberfläche.

Neben dem Maximalausbau werden Varianten angeboten, die nicht alle Units enthalten.

Die Bedienung des SE Manager ist in der Online-Hilfe zum SE Manager und im Handbuch „Bedienen und Verwalten “ [57] beschrieben.

Die Net Unit, für SU /390 mit High-Speed Network Connect (HNC), bietet höchste Performance und Sicherheit für die interne Kommunikation in einem SE Server und für die Anbindung an Kundennetzwerke (LAN).

Das BS2000-Betriebssystem in den freigegebenen Versionen bedient die Server Unit /390 (/390-Architektur) und die Server Unit x86 (x86-Architektur).

> ¡ Weitere Ausbaustufen der Server Units können terminlich entkoppelt freigegeben werden. Beachten Sie dazu die Produktankündigungen und Freigabeinformationen.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter *http://bs2manuals.ts.fujitsu.com*.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Die Änderungen finden Sie in den entsprechenden Vertriebsschriften und in der Freigabemitteilung unter *http://bs2manuals.ts.fujitsu.com*.

1.4 Darstellungsmittel

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

- **Systemparameter** für die Klasse-2-Systemparameter von BS2000
- **BS2000-Betriebssystem**, wenn eine Unterscheidung von BS2000 nach Version und Ausgestaltung nicht nötig ist.

Bei Standarddateinamen steht `<ver>` für die interne Bezeichnung der BS2000-Version, z.B. 210 für BS2000 V21.0A.

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

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

`Eingabe`	Eingaben in Beispielen werden in halbfetter Schreibmaschinenschrift dargestellt
`Ausgabe`	DSECTS, Übersetzungslisten oder Ausgaben in Beispielen werden in Schreibmaschinenschrift dargestellt
`[ ]`	Optionale Angaben in Syntaxdefinitionen. Die eckigen Klammern sind Metazeichen, die in einer Anweisung nicht angegeben werden.
`{ | | }`	Alternative Angaben in Syntaxdefinitionen. Die zur Auswahl stehenden Operanden sind durch einen senkrechten Strich getrennt.
	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.

1.5 Aufgaben der Systemverwaltung

Die Administration von BS2000 umfasst diejenigen Aufgaben, welche im Hinblick auf eine effiziente, sichere und optimale Nutzung des Systems wahrgenommen werden müssen.

Die Aufgaben der Systemverwaltung sind:

- Zusammenstellen und Einführen der zweckmäßigen Betriebssystemkomponenten
- Verwalten und Aktualisieren des im Data Center eingesetzten Organisationsprogramms und der Programmbibliotheken
- Sicherstellen der Funktionsfähigkeit der System- und systemnahen Software
- Verwalten der Zugangsberechtigung zum System
- Verwalten und Verteilen der Betriebsmittel
- Benutzer- und Dateienverwaltung
- Auswerten von Fehlerstatistiken
- Sichern der System- und Benutzerdaten
- Installieren von Softwareprodukten
- Ergreifen von Maßnahmen zur Rationalisierung und Automatisierung des Betriebes im Data Center
- Einrichten und Zuordnen von Operator-Kennungen und Operator-Rollen (im Zusammenhang mit der Privilegienverwaltung).
- Erstellung von Diagnoseunterlagen bei Betriebssystem-, Benutzerprogramm- und Bedienungsfehlern

Der Systemverwaltung stehen mit privilegierten Benutzerkennungen auf Kommando- und Dienstprogrammebene geeignete Werkzeuge für diese Aufgaben zur Verfügung.

Kommandos

Neben speziellen Systembetreuungskommandos, die für den privilegierten Aufrufer reserviert sind, können sowohl sämtliche Benutzerkommandos als auch eine Untermenge der Operatorkommandos eingesetzt werden.
Die Untermenge an Operatorkommandos erlaubt der Systemverwaltung, an der Bedienschnittstelle steuernden und überwachenden Einfluss auf Domänen der Systembedienung, wie z.B. Lastverteilung und Auftragssteuerung, auszuüben.

Bei den Benutzerkommandos wirkt sich die Privilegierung z.B. der Benutzerkennung TSOS in einem erweiterten Funktionsumfang aus, der der Systemverwaltung angeboten wird. Diese Funktionen realisieren im Wesentlichen eine präzisere, umfassendere Informationsausgabe und die für den „Super-User“ eines Betriebssystems typische Aufhebung von Betriebsmittel-, Prozess- und Kataloggrenzen, die dem nicht-privilegierten Benutzer gesetzt sind.

Dienstprogramme

Der Systemverwaltung stehen sämtliche Dienstprogramme, die auch die Benutzer anwenden dürfen, mit einem zum Teil erweiterten Funktionsumfang zur Verfügung (z.B. DPAGE, HSMS/ARCHIVE, JMU, SPCCNTRL).

Darüber hinaus kann sich die Systemverwaltung spezieller Dienstprogramme bedienen, die nur unter einer privilegierten Benutzerkennung laufen und zur Steuerung und Überwachung des Betriebssystems eingesetzt werden.

1.6 Aufgaben des Operating

Dem Operator obliegen die Steuerung der Inbetriebnahme des Systems, Eingriffe zur Regelung des Systems, die Kommunikation mit dem Benutzer und die Bedienung der peripheren Geräte. Systeminitialisierung (Startup) und Systemspeicherabzug (SLED) können ausschließlich vom Operator bedient werden. Für Funktionen zur Hardware-Diagnose und Rekonfiguration, die dem Operator an der Konsole vorbehalten sind, stehen ihm weitere SVP-Kommandos zur Verfügung.

Für die übrigen Aufgaben verwendet der Operator Kommandos (spezielle Operatorkommandos bzw. Benutzer- oder Systemverwaltungskommandos), deren Nutzung im vorliegenden Handbuch ebenfalls beschrieben ist. Zu den wesentlichen Aufgaben des Operators zählen u.a.:

- Starten und Überwachen des Datenkommunikationssystems
- Rekonfigurieren der Hardwarekomponenten und der Verbindungen zur Peripherie
- Bereitstellen und Zuweisen externer Datenträger
- Überwachen und Steuern der Gerätebelegung
- Steuerung der Betriebsmittelzuweisung
- Betriebsüberwachung
- Kommunikation mit den Benutzern

1.7 Automatisierung

Der Haupteinsatzbereich von BS2000-Systemen ist der performante und störungsfreie Ablauf von Kundenanwendungen.
Bei vielen BS2000-Kunden ist der Betrieb vollständig automatisiert, d.h. die folgenden Phasen laufen vollautomatisch und zuverlässig ab:

- Inbetriebnahme (vom Systemstart bis zur Verfügbarkeit der Anwendungen)
- Betrieb (Systempflege und Ablauf der Anwendungen)
- Systembeendigung

Durch den Einsatz moderner Hardware, wie robotergestützte Magnetbandkassetten-Systeme, kann die manuelle Bedienung von Geräten weitgehend entfallen.
Aus Sicht der Software ist es notwendig, den Ablauf dieser Phasen an die Hardware- und Software-Gegebenheiten des Kunden anzupassen („Customizing“).

Mit der Kommandosprache SDF-P und dem Kommunikationsmittel Jobvariable können die genannten Abläufe auf Kommandoebene programmiert/automatisiert werden.

In den Phasen der Inbetriebnahmne und Systembeendigung, in denen diese Werkzeuge noch nicht oder nicht mehr zur Verfügung stehen, werden die datei- oder datenorientierten Schnittstellen des SVP (z.B. Ein-/Ausschaltsequenzen), der Systemeinleitung (Startup-Parameterservice) und des DSSM (Subsystemkatalog) angeboten.

Für alle weitergehenden Automatisierungsaufgaben der Systembetreuung, wie z.B. die automatische Reaktion auf Ereignisse, bietet das Softwareprodukt PROP-XT passende Benutzerkommandos an und stellt die zugehörigen Daten in S-Variablen bereit.

So steht – durch die ergänzende Funktion von PROP-XT – mit SDF-P und Jobvariablen ein komfortables, einheitliches Werkzeug zur Automatisierung aller Aufgaben der Systembetreuung zur Verfügung.

2 Systemeinleitung und -beendigung

Dieses Kapitel beschreibt die Abläufe bei der Systemeinleitung und -beendigung.

2.1 Überblick über die Abläufe bei der Systemeinleitung

Die Systemeinleitung von BS2000 läuft als „Bootstrapping“ ab, d.h. es werden schrittweise immer mächtigere Funktionseinheiten geladen und gestartet, bis BS2000 betriebsbereit ist.

Der Anstoß zum Ablauf der verschiedenen Routinen erfolgt hardware-abhängig über den Serviceprozessor (SVP) auf SU /390 oder X2000 auf SU x86 bzw. die Restart-Verarbeitung bei automatischem Restart. Mit diesem Urladen (IPL, Initial Program Load) wird die Systemeinleitung gestartet. Dabei wird neben der IPL-Platte auch die Art der Systemeinleitung festgelegt. Durch die Einstellung der Ladeoptionen für BS2000 wird festgelegt, ob der Systemstart komfortabel oder flexibel ablaufen soll.

Für einen komfortablen, weitgehend automatischen Ablauf stehen dem Operator die Modi FAST und AUTOMATIC, für den flexiblen, dialogorientierten Ablauf steht ihm der Modus DIALOG zur Verfügung (siehe Abschnitt „Arten der Systemeinleitung").

Für ein VM2000-Gastsystem wird die Systemeinleitung über den SE Manager (siehe Handbuch „Bedienen und Verwalten“ [57]) oder durch das VM2000-Kommando START-VM (siehe Handbuch „VM2000“ [60]) ausgelöst. Das VM2000-Gastsystem kann auf SU x86 auch über die SVP-Funktionen der KVP-Konsole, die der virtuellen Maschine zugeordnet ist, gestartet werden.

Die wichtigsten Funktionsabläufe während der Systemeinleitung von der Bereitstellung der Hardware bis zum eigentlichen Abschluss des Startups, bis „System Ready“ sind:

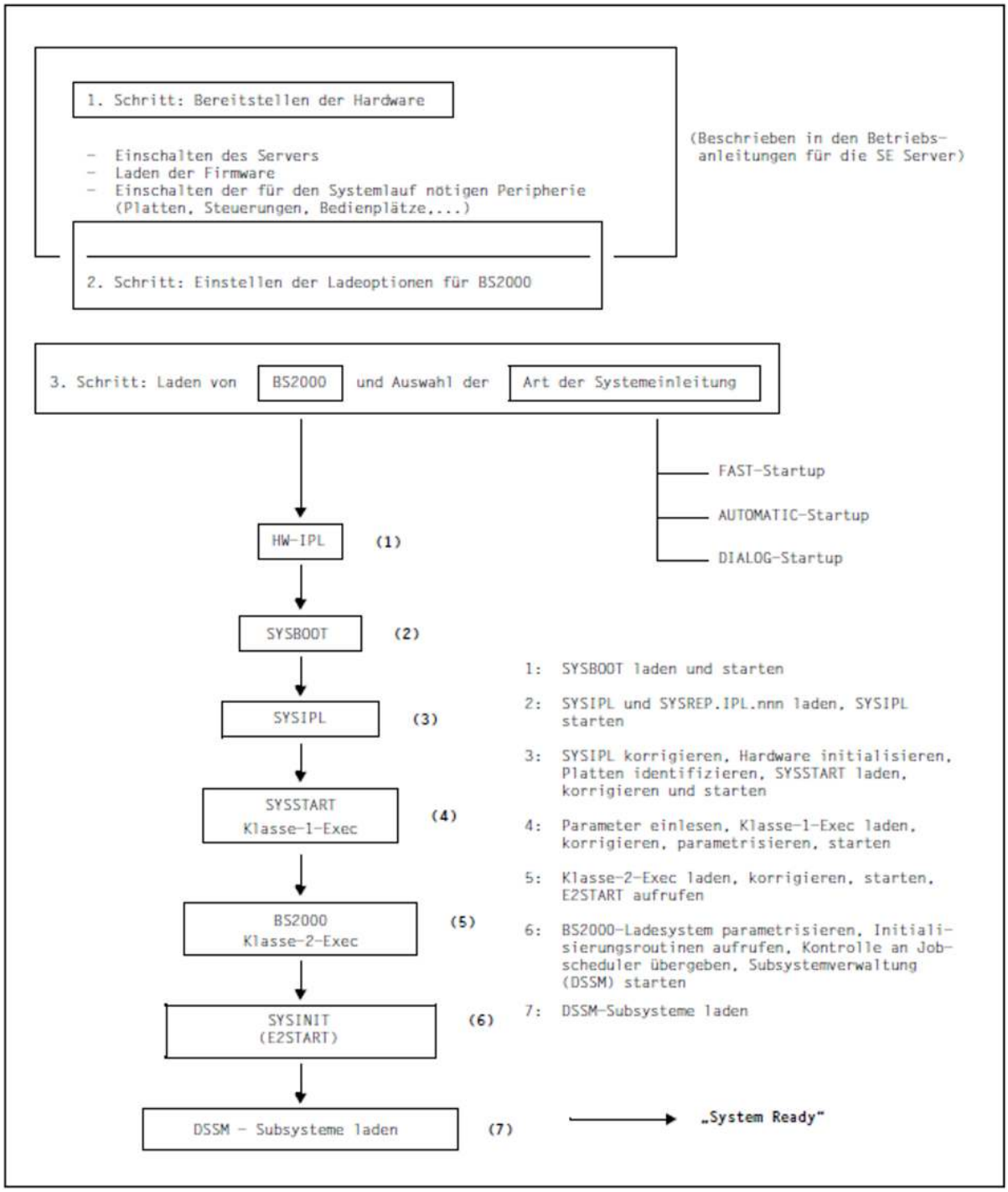


Bild 2: Funktionsabläufe bei der Systemeinleitung von BS2000

Die Systemeinleitung für BS2000 kann beginnen, wenn die dazu notwendigen Hardware-Einheiten (MU, Server Units, periphere Geräte) eingeschaltet und betriebsbereit sind. Der Ablauf dieser Schritte (Einschalten der Stromversorgung, Laden der Firmware usw.) ist ausführlich in den Betriebsanleitungen für die SE Server beschrieben.

Der interne Ablauf der Systemeinleitung für BS2000 beginnt mit dem Laden des „Urladers“ (SYSBOOT). Dies ist die eigentliche Aufgabe der Routine HW-IPL (Vorgang (1) in Bild 2).

SYSBOOT ist das erste Programm der Systemeinleitung, das elementare Prüfungen vornimmt und eine weitere Laderoutine anstößt (Vorgang (2) in Bild 2).

Diese von SYSBOOT geladene und gestartete Routine ist SYSIPL, die die Aufgabe hat, die Optionen im Modus DIALOG abzufragen und die aktuelle Platten- und CPU-Konfiguration zu ermitteln (Vorgang (3) in Bild 2). Die Zeitbasis der Systemzeit wird festgelegt. Die Plattenkonfiguration wird auf Vollständigkeit und Eindeutigkeit geprüft. Außerdem wird von dieser Routine entweder SYSSTART oder SLED geladen und jeweils korrigiert.

Die beiden Programme SYSBOOT und SYSIPL und die IPL-Rep-Datei befinden sich auf festen Plätzen eines bestimmten Plattenspeichers, der IPL-Platte. Die IPL-Platte kann ein gemeinschaftlicher Plattenspeicher – die Platte eines Pubsets – oder eine Privatplatte sein. Sie muss für FAST- und AUTOMATIC-Startup eine der Platten des Home-Pubsets sein.

Nur bei DIALOG-Startup ist ein IPL von Privatplatte möglich. Dabei muss der Operator im späteren Verlauf der Systemeinleitung angeben, welcher Pubset der Home-Pubset für den Systemlauf sein soll.

Beim IPL von gemeinschaftlicher Platte wird der Pubset, zu dem die Platte gehört, automatisch zum Home-Pubset gewählt. Nur bei Dialog-Startup mit der Option ALLDISK kann der Operator den Pubset noch wechseln.

Wenn die IPL-Platte nicht zum späteren Home-Pubset gehört, ist insbesondere darauf zu achten, dass Versionen und Korrekturstände auf beiden gleich sind.

> **i** In einer Plattenkonfiguration können durchaus mehrere IPL-Platten existieren.
> Das Einrichten von IPL-Platten ist Aufgabe der Systembetreuung und wird mit Hilfe des Dienstprogramms SIR vorgenommen, das ausführlich im Handbuch „Dienstprogramme“ [15] beschrieben ist.
> Eine IPL-Platte kann entweder für SU /390 oder für SU x86 verwendet werden.

Beim Laden von SYSBOOT und SYSIPL/SLED stehen die Dateiverwaltungs-Funktionen von BS2000 noch nicht zur Verfügung. Die notwendigen Dateien können daher nur gefunden werden, wenn sie zuvor auf der IPL-Platte „verankert“ worden sind. Dies geschieht mit der Anweisung CREATE-IPL-VOLUME des Dienstprogramms SIR und beinhaltet folgende Einzelschritte:

- Die zum Laden von SYSBOOT und SYSIPL/SLED nötigen Dateien werden auf die IPL-Platte kopiert.
- Die von SYSBOOT und SLED verwendeten Sicherstellungsdateien werden auf der IPL-Platte erzeugt.
- Im SVL der IPL-Platte wird je ein unmittelbarer Verweis auf jede dieser Dateien eingetragen:

Original-Datei	**von SIR erzeugte Datei**
---	SYSDAT.IPL-CONF.DSKnnn
---	SYSPRG.BOOT.DSKnnn.SAVE
SYSPRG.IPL.<ver>	SYSPRG.IPL.DSKnnn
---	SYSPRG.SLED.DSKnnn.SAVE
SYSREP.IPL.<ver>	SYSREP.IPL.DSKnnn
SYSREP.SLED.<ver>	SYSREP.SLED.DSKnnn

„nnn“ steht jeweils für die Nummer der IPL-Platte innerhalb des Pubsets. Handelt es sich bei der IPL-Platte um eine Privatplatte, wird statt des Namensteils DSKnnn die VSN der Privatplatte eingesetzt.

- Sofern noch nicht auf einer anderen Platte des Pubsets vorhanden, wird die Sicherstellungsdatei für Systemkorrekturen SYS.NSI.SAVEREP erzeugt (nicht bei Privatplatten).

Die Original-Dateien werden zur Systemeinleitung nicht mehr gebraucht und können im laufenden System gefahrlos geändert werden, z.B. zur Übernahme eines neuen Korrekturstandes. Alle Änderungen werden für die Systemeinleitung jedoch erst wirksam, nachdem mit SIR neue Kopien erzeugt und verankert worden sind.

Die verankerten Dateien dürfen im laufenden System weder geändert noch gelöscht werden, weil dadurch i.A. die IPL-Fähigkeit der Platte zerstört wird. Sie werden seitens SIR durch BACKUP-CLASS=E und MIGRATE=FORBIDDEN vor Verlagerung und Verdrängung geschützt.

Alle weiteren für die Systemeinleitung benötigten Routinen befinden sich als „normale Dateien“ auf einem Pubset oder einer Privatplatte.
Die Konfigurationstabellen werden dynamisch erzeugt.

SYSSTART (Vorgang (4) in Bild 2) ist ein Programm, das die eigentliche Systemeinleitung für BS2000 vorbereitet und durchführt. In der Vorbereitung werden i.W. die Parameter für BS2000 eingelesen, die Objektcodekorrekturen für Klasse-1-Exec ermittelt, sowie die Versionsstände von SYSSTART und BS2000 auf Konsistenz geprüft. In der Durchführung werden die einzelnen Initialisierungsfunktionen von BS2000 tabellengesteuert aufgerufen. Zu diesen Initialisierungsfunktionen gehören auch die Datenstrukturen der virtuellen Speicherverwaltung und die Initialisierung der Paging-Bereiche, womit SYSSTART auch den Übergang in den virtuellen Adressierungsmodus von BS2000 vorbereitet.

Von SYSSTART wird schließlich das BS2000-Ladesystem (Vorgang (5) in Bild 2), das aus den beiden Teilen „Klasse-1-Exec“ (resident) und „Klasse-2-Exec“ (seitenwechselbar) besteht, aufgerufen.
Auf dieser Stufe der Systemeinleitung werden u.a. bei der Geräteverwaltung Zugriffsrechte für die Geräte der Systemeinleitung beantragt (Platten des Home-Pubsets und der Paging-Pubsets). Nach Durchlauf dieser Systemeinleitungs-Phase liegt das BS2000-Ladesystem **geladen**, **korrigiert** und **parametrisiert** vor.

Geladen heißt, dass sich dieser Teil von BS2000 nach dem Ladevorgang vollständig im Hauptspeicher befinden. Es werden zuerst die hauptspeicherresidenten Teile (Klasse-1-Exec) geladen. Die restlichen Teile sind seitenwechselbar. Diese Routinen werden von SYSINIT über den Hauptspeicher auf den Seitenwechselspeicher kopiert.

Korrigiert heißt, dass Module dieser Teile von SYSSTART während der Systemeinleitung mittels so genannter Rep-Sätze modifiziert werden. Die Rep-Sätze können von max. vier katalogisierten Plattendateien in beliebiger Reihenfolge eingelesen werden. Die Konsole kann – außer für Reps für SYSIPL – zusätzlich als Eingabegerät definiert werden. Bei der Systemeinleitung werden alle verarbeiteten Rep-Sätze von Platte und Konsole in die Sicherstellungsdatei SYS.NSI.SAVEREP geschrieben und später in der Datei $SYSAUDIT.SYS.REPLOG.<datum>.<session-nr>.01 protokolliert.

Parametrisiert heißt, dass eine Menge von Parametersätzen, die Anweisungen für die Initialisierungsroutinen von BS2000 enthalten, eingelesen werden. Die gesamte Parametereingabe besteht aus einer Folge von Abschnitten, die – durch spezifische Schlüsselwörter gekennzeichnet – jeweils bestimmte Funktionseinheiten betreffen und von diesen ausgewertet werden (siehe Kapitel „Parameterservice").

Der Operator kann durch eine Voreinstellung beim Systemstart entscheiden, ob das Laden, Korrigieren und Parametrisieren weitestgehend automatisch und dialogfrei ablaufen soll (in diesem Fall werden jeweils Dateien mit Standardnamen herangezogen), oder ob der Ablauf flexibel – im Dialog mit dem Operator – gesteuert werden soll.

Abschließend wird mit E2START die letzte Phase der Systemeinleitung für BS2000 gestartet (Vorgang (6) in Bild 2). Diese Routine läuft bereits unter BS2000 und bestimmt zunächst den Namen der Kommandodatei (CMDFILE), die nach „System Ready“ automatisch angestartet werden soll.

Zur Erreichung eines funktionstüchtigen BS2000 laufen in dieser Phase Initialisierungsroutinen für die folgenden Funktionseinheiten ab:

- Aktivierung des Task-Schedulers
- Öffnen der Systemdateien (Benutzerkatalog, SYSEAM, usw.)
- Bereitstellung der Dateikatalogverwaltung
- Aktivierung des dynamischen Bindeladers (DBL)
- Aktivierung der Bibliothekszugriffsmethode PLAM
- Start der Funktionen zur Überwachung von Platten- und Bandgeräten
- Aktivierung der SERSLOG-Funktion
- Start der Dynamischen Subsystemverwaltung DSSM [1)]

Nach Freigabe des von der Systemeinleitung belegten Speichers und Start des Job-Schedulers ist „System Ready“ erreicht und die Abarbeitung der in der Kommandodatei CMDFILE hinterlegten Kommandos wird angestoßen. Zwar ist die Verwendung einer Kommandodatei – deren Name frei wählbar ist – nicht zwingend erforderlich, doch wegen der Forderung nach Automatisierung der Abläufe unbedingt zu empfehlen.

Die Verwendung einer Kommandodatei gewährleistet u.a. die automatische Aktivierung derjenigen Systemkomponenten und -einstellungen, die für die Funktionstüchtigkeit des spezifischen Systems relevant sind:

- Inbetriebnahme von optionalen Subsystemen
- Starten des BS2000-Datenkommunikationssystems [2)]
- Laden des SPOOL-Systems [3)]
- Spezifische Lastregulierung
- Aktivierung spezieller Programme über Enter-Dateien

Hinweise

[1] DSSM:

Während der Systemeinleitung erhält die Dynamische Subsystemverwaltung die Steuerung. Diese initialisiert sich mit dem vorgegebenen Subsystemkatalog und aktiviert Subsysteme bzw. leitet die Aktivierung von Subsystemen ein. Der Startzeitpunkt eines Subsystems (vor oder nach „System Ready“) wird von der Systembetreuung bei der Deklaration festgelegt. Auf diese Weise können Subsysteme automatisch aktiviert werden.

Beim Start eines Subsystem ermittelt DSSM über IMON-GPN die Pfadnamen aller Dateien des Subsystems aus dem aktuellen SCI. Ist IMON-GPN nicht verfügbar (wird bereits beim Start von DSSM geladen) oder existiert keine Datei unter dem ermittelten Pfadnamen, verwendet DSSM die im Subsystemkatalog eingetragenen Standard-Namen. Bei Verwendung des Standard-Namens wird die Meldung `ESM0665` ausgegeben.

[2] Datenkommunikationssystem (DCM):

Für den Start des Datenkommunikationssystems noch vor „System Ready“ kann das DCSTART-Kommando auch als BCAM-Parameter in der Startup-Parameterdatei vorgegeben werden (siehe Handbuch „BCAM“ [4], Abschnitt BCAM-BS2000-Parameterdatei).

Ist das nicht der Fall, muss nach jeder Systemeinleitung das Datenkommunikationssystem gesondert in Betrieb genommen werden. Dies geschieht mit dem Kommando DCSTART, das dann zweckmäßigerweise in der CMDFILE hinterlegt wird.

Mit dem Kommando DCSTART wird automatisch die Eröffnung der folgenden internen, privilegierten Anwendungen des Systems eingeleitet:

- $DIALOG (Anwendung für Dialogverarbeitung (TIAM))
- $CONSOLE (Anwendung für logische Konsolen, siehe "Operatorfunktionen")
- $BCAM (Anwendung für den DCM-Informationsdienst)

Erfolgt das erste DCSTART-Kommando später als 10 Min. nach „System Ready“ oder wird bei laufendem BS2000 das DCM beendet (Kommando BCEND) und neu gestartet, dann muss die Anwendung $DIALOG vom Operator mit dem Kommando START-DIALOG-APPLICATION manuell gestartet werden. Eine andere Möglichkeit wäre die Aufnahme von /START-DIALOG-APPLICATION in die SOF (Start Option File) von BCAM. Voraussetzung dafür ist die Einrichtung eines Konsolzugangs für BCAM mit der Berechtigung für START-DIALOG-APPLICATION (siehe Handbuch „BCAM“ [4]).

Weil im Bedienmodus mit Operator-Logon der Operator nach „System Ready“ erst ein SET-LOGON-PARAMETERS- und ein REQUEST-OPERATOR-ROLE-Kommando eingeben muss, um weitere Kommandos eingeben zu können, empfiehlt es sich, die Voraussetzungen für die ersten beiden Kommandos ebenfalls aus der CMDFILE heraus zu schaffen. Diese Voraussetzungen sind:

- die Operatorkennung muss entsperrt sein
- eine Operatorrolle muss eingerichtet sein
- die Operatorrolle muss der Operatorkennung zugeordnet sein

Da das Entsperren der Operatorkennung unter der Kennung TSOS geschehen muss, das Einrichten einer Operatorrolle und das Zuordnen zu einer Operatorkennung aber nur unter SYSPRIV durchgeführt werden kann, empfiehlt es sich, aus der CMDFILE einen ENTER-Job aufzurufen, der das UNLOCK-USER-Kommando absetzt und eine weitere Prozedur zum Einrichten und Zuordnen der Operatorrollen unter der Kennung SYSPRIV aufruft.

Der gesamte Ablauf könnte etwa folgendermaßen aussehen:

Aufruf aus der CMDFILE:

```
/ENTER-JOB E.OPR-LOGON.TSOS
```

Inhalt der Datei E.OPR-LOGON.TSOS

```
/SET-LOGON-PARAMETERS
/ UNLOCK-USER SYSPRIV
/ SET-JOB-STEP
/ UNLOCK-USER SYSOPR
/ SET-JOB-STEP
/ ENTER-JOB FROM-FILE=$TSOS.E.OPR-LOGON.SYSPRIV,-
/           PROC-ADMISS=*PAR(USER-ID=SYSPRIV,-
/                            ACC=SYSACC,-
/                            PASS=*NONE)
/EXIT-JOB
```

Inhalt der Datei E.OPR-LOGON.SYSPRIV

```
/SET-LOGON-PARAMETERS
/ CREATE-OPERATOR-ROLE OPERATOR-ROLE=SYSADM,ROUT-CODES=*ALL
/ SET-JOB-STEP
/ MODIFY-OPERATOR-ATTR USER-ID=SYSOPR,ADD-OP-ROLE=SYSADM
/ SET-JOB-STEP
/ INFORM-OPERATOR,-
/          MSG='*** OPERATOR-ROLE SYSADM CREATED AND ADDED ***'
/ INFORM-OPERATOR,-
/ MSG='+-----------------------------------------------------+'
/ INFORM-OPERATOR,-
/ MSG='!    THE FIRST OPERATOR COMMANDS AFTER SYSTEM READY    !'
/ INFORM-OPERATOR,-
/ MSG='!    (BEFORE /DCSTART ... )  MUST BE:                  !'
/ INFORM-OPERATOR,-
/ MSG='!    /SET-LOGON-PARAMETERS SYSOPR,SYSACC               !'
/ INFORM-OPERATOR,-
/ MSG='!    /REQUEST-OPERATOR-ROLE OP-ROLE=SYSADM             !'
/ INFORM-OPERATOR,-
/ MSG='+-----------------------------------------------------+'
/EXIT-JOB
```

Erst dann ist BS2000 kommunikationsfähig.

[3)] SPOOL:

Nach jeder Systemeinleitung muss SPOOL gesondert geladen und initialisiert werden. Der SPOOL-Startup wird durch das Kommando START-SUBSYSTEM eingeleitet.
Mit dem Operanden SUBSYSTEM-PARAMETER kann festgelegt werden, ob für SPOOL ein Warm- oder Kaltstart durchgeführt werden soll und ob zusätzlich das Softwareprodukt RSO geladen werden soll. Dieses Kommando sollte entweder gleich nach „System Ready“ gegeben werden oder in der Kommandodatei CMDFILE enthalten sein. Ist SPOOL nicht geladen, können keine SPOOLIN- und SPOOLOUT-Jobs durchgeführt werden. SPOOL-Anforderungen des Operators (z.B. Kommandos START-PRINTER-OUPUT, MODIFY-PRINTER-OUTPUT-STATUS,...) werden entweder abgewiesen, ignoriert oder zurückgestellt.

Zusammenfassend besteht die Systemeinleitung aus folgenden **internen** Schritten:

HW-IPL:	- Laden des 1. Blocks von SYSBOOT
SYSBOOT:	- Laden des 2. Blocks von SYSBOOT - Laden von SYSREP.IPL.<version> - Laden und Starten von SYSIPL
SYSIPL:	- eigene Korrektur - Laden, Korrigieren und Starten von SYSSTART
SYSSTART:	- Einlesen der Parameter - Laden und Korrigieren der hauptspeicherresidenten Teile von BS2000 (Klasse-1-Exec) - Initialisierung des Seitenwechselspeichers
Klasse-1-Exec:	- Initialisierung des residenten Teils von BS2000 - Automatisches Zuschalten von Plattengeräten, die DETACHED generiert wurden und auf denen benötigte Public-Platten montiert sind [1] - Laden des seitenwechselbaren Teils von BS2000 (Klasse-2-Exec) - Korrektur der seitenwechselbaren Teile
Klasse-2-Exec:	- Initialisierung der seitenwechselbaren Teile
SYSINIT (E2START):	- Ermittlung der Kommandodatei und Aufruf von Initalisierungsfunktionen für BS2000-Funktionseinheiten (DSSM, PLAM, usw.) - Freigabe des belegten Speichers und Start des Jobschedulers
„System Ready“ (jetzt erfolgt z.B. der Anstoß der Kommandodatei CMDFILE)	

[1] Während des Startups werden automatisch alle Public-Platten des Home-Pubsets und alle Pubsets, die Paging-Platten enthalten und die beim Parameterservice angegeben wurden, dem System zugeschaltet, auch wenn sie explizit DETACHED generiert worden sind. Die Geräte des Home-Pubsets bleiben während des gesamten Systemlaufs ATTACHED.

Mit dem Kommando SHOW-SYSTEM-INFORMATION können Informationen über die Systemkonfiguration, die eingesetzte VM2000-Version, das Monitorsystem und die Parameter der Zeiteinstellung eingeholt werden.

2.1.1 Zeitbestimmung bei der Systemeinleitung

Siehe auch Kapitel „Systemzeit-Verwaltung“.

Für die Bestimmung von Datum und Uhrzeit während der Systemeinleitung existieren drei Quellen:

- SVP-Uhr: liefert die lokale Zeit
  - Auf SU /390 wird die SVP-Uhr durch die Management Unit synchronisiert.
  - Auf SU x86 wird die SVP-Uhr von X2000 emuliert. Die SVP-Zeit entspricht der Zeit des Trägersystems.
- CPU-Uhr (TODR): ergibt mit einem auf dem Home-Pubset hinterlegten Korrekturwert die lokale Zeit. Diese Uhr läuft auch bei CPU-Stop weiter, aber nicht bei einer Unterbrechung der Stromversorgung.
  - Auf SU /390 ist das TODR eine eigenständige Uhr.
  - Auf SU x86 wird das TODR von X2000 emuliert.
- Operator
  Dem Operator wird die aktuell gültige Zeit an der Konsole gezeigt. Bei Fehlersituationen oder im Modus DIALOG (mit der Option UNLOCK) werden ihm die entsprechenden Meldungen an der Konsole ausgegeben und er wird zur Bestätigung oder Korrektur aufgefordert.

Die lokale Zeit (LT) ist die am Installationsort von BS2000 gültige gesetzliche Zeit. Durch gesetzlich vorgeschriebene Zeitumstellungen (Sommer-/Winterzeit) ist sie nicht monoton fortlaufend, sondern erfordert an den Umstellungszeitpunkten eine positive bzw. negative Korrektur. Diese Korrektur erfolgt automatisch mit Hilfe der GTIME-Parameterdatei, in der der Korrekturwert und die Umstellungszeitpunkte hinterlegt sind (siehe "Steuerung der Systemzeit (GTIME)").

Die Systemzeit basiert auf der lokalen Zeit. BS2000 ist ohne Systemzeit nicht ablauffähig.

Die Systembetreuung definiert mit den Parametern für Zeitzone, Zeitsprung, Sommer- bzw. Winterzeit sowie mit der Festlegung von Umstellungsdaten die Basis zur Berechnung der universellen Weltzeit UTC aus der lokalen Zeit LT und zur Sommer-/Winterzeitumstellung der lokalen Zeit während des Systemlaufs.
Damit können System und Benutzer mit der Funktion GTIME sowohl auf ein lokales als auch auf ein systemübergreifend verständliches Zeitbezugssystem zugreifen.

Automatische Zeitbestimmung bei Startup

Die Zeitbestimmung erfolgt automatisch, wenn folgende Bedingungen erfüllt sind:

- Die IPL-Option UNLOCK ist nicht gesetzt, d.h. die Systemeinleitung erfolgt im Modus FAST oder AUTOMATIC bzw. im Modus DIALOG ohne die IPL-Option UNLOCK.
- Die Zeit der SVP-Uhr ist gültig.
- Die Zeit der SVP-Uhr ist formal richtig (z.B. nicht 31.2.14).
- Die Zeit der SVP-Uhr ist größer als die Zeit des letzten Systemlaufs und der positive Zeitsprung gegenüber dem letzten Systemlauf beträgt nicht mehr als 6 Tage (Ausnahme: im Modus AUTOMATIC kann der positive Zeitsprung beliebig sein).

Erfolgt ein Shutdown unmittelbar vor und der anschließende Startup unmittelbar nach der Umstellung von Sommerzeit auf Winterzeit, beginnt die Systemeinleitung mit einer kleineren Zeit als die des letzten Systemlaufs. Dies ist nur möglich, wenn im Modus DIALOG die Option UNLOCK gesetzt wird (siehe "DIALOG-Startup").

Negative Zeitsprünge dürfen außerhalb dieser Umstellungszeiten nur in Absprache mit der Systembetreuung durchgeführt werden. Die Konsistenz der Datenhaltung kann damit gefährdet werden.

Beeinflussbarkeit der Zeitbestimmung bei DIALOG-Startup

Im Modus DIALOG und bei gesetzter Option UNLOCK wird der Operator auch bei einem gültigen Wert der SVP-Uhr zur Bestätigung aufgefordert. Wird die Zeit (und das Datum) der SVP-Uhr nicht bestätigt, muss der Operator selbst Datum und Uhrzeit angeben. Dabei können auch große positive und negative Zeitsprünge gegenüber dem letzten Systemlauf angegeben werden.
Diese so angegebene Zeit wird, wenn sie nach nochmaliger Protokollierung bestätigt wurde, als gültige Systemzeit für den einzuleitenden Systemlauf verwendet.

2.1.2 Format der Meldungen bei der Systemeinleitung

Jede Meldung der Systemeinleitung ist in einer der folgenden Formen aufgebaut:

```
? P.msgtext
? tsn-makz.hhmmss % msgtext
```

Dabei ist:

?	Meldung mit Antwort-Anforderung (ohne Antwort-Anforderung: %)
p	Meldung wird vom Startup ausgegeben
msgtext	Meldungstext
tsn	Meldung wird von BS2000 ausgegeben; tsn ist die Auftragsnummer einer Systemtask oder ein Auftragsname
-	Trennstrich; entfällt bei Meldungen des residenten Teils der Systemeinleitung
makz	Meldungsauftragskennzeichen (3 Ziffern oder Buchstaben, führende Nullen dürfen entfallen); makz entfällt bei Meldungen des residenten Teils der Systemeinleitung
. / #	Trennzeichen; abhängig vom Systemparameter SECSTART
hhmmss	Tageszeit; entfällt bei Meldungen des residenten Teils der Systemeinleitung

Der Meldungstext beginnt immer mit einer Schlüsselnummer (3-stellige Meldungsklasse und 4-stellige laufende Nummer, für x steht jeweils eine Sedezimalziffer).

Die Meldungen des residenten Teils der Systemeinleitung haben die Schlüsselnummern NSIxxxx (Nucleus System Initialization).

Die Meldungen des seitenwechselbaren Teils der Systemeinleitung haben die Schlüsselnummern EXCxxxx.

Beispiel einer Meldung des residenten Teils der Systemeinleitung:
```
? P.NSI0050  SPECIFY PARAMFILE ...
```
Beispiel einer Meldung nach dem residenten Teil der Systemeinleitung:
```
? TSC-000.112133%  NSI0077  ENTER AUTOMATIC COMMAND FILE
```

Die Antwort selbst beginnt mit der auf das Fragezeichen folgenden Zeichenkette der Meldung, also P. oder tsn (einschließlich eines evtl. vorhandenen Trennstriches, des Meldungsauftragskennzeichens und des darauffolgenden Punktes). Unmittelbar anschließend muss der Antworttext folgen.

Beispiel einer Antwort während des residenten Teils der Systemeinleitung:
```
P.FN=PARAMFILE.OPR
```
Beispiel einer Antwort nach dem residenten Teil der Systemeinleitung:
```
TSC.CMDFILEX
```

Bei der Systemeinleitung werden Meldungen stets in englischer Sprache ausgegeben.

2.1.3 Hinweise zur Systemeinleitung

Unabhängig vom jeweiligen Server-Typ, der vom Operator zu bedienen ist, sind vorab folgende Fragen zu klären, die zum Verständnis des (korrekten) Ablaufs von Bedeutung sind:

- Wie werden die benötigten Dateien und Geräte identifiziert?
- Was gilt es bezüglich der Eindeutigkeit der Plattenkonfiguration zu beachten?
- Wie wird der Home-Pubset bestimmt und auf Vollständigkeit und Konsistenz geprüft?
- Wie verhält sich das System bei belegten Pubsets oder weggeschalteten Geräten?
- Wodurch können Geräte bei der Systemeinleitung nicht verfügbar sein?
- Was gilt es bei Multiprozessoren zu beachten?
- Welche Rolle spielt die Parameterdatei bei der Systemeinleitung?
- Auf welche Weise wird die Ladeprozedur protokolliert?
- Was ist in Bezug auf die Unicode-Fähigkeit zu beachten?

Identifikation der benötigten Dateien und Geräte

Die für HW-IPL bereitzustellenden Datenträger sind wie folgt zu adressieren:

Bei SU /390 wird das IPL-Gerät mit der Device Number bezeichnet. Bei Geräten mit vierstelliger MN ist die Device Number gleich der MN. Für Geräte mit zweistelliger MN existiert eine eindeutige Zuordnung zwischen Device Number und MN, welche, welche im Handbuch „Systeminstallation“ [55] beschrieben ist.

Bei SU x86 wird das IPL-Gerät mit dem mnemotechnischen Gerätenamen (MN) bezeichnet.

Es sollten nur Platten vom Home-Pubset als IPL-Platten verwendet werden. Platten mit einer physikalischen Blockgröße von 4KB werden nicht als IPL-Platte unterstützt.

Alle später benötigten Geräte bei der Ladeprozedur werden mit dem mnemotechnischen Gerätenamen MN angegeben.

Wenn während der Systemeinleitung Dateien verwendet werden, die auf Privatplatten gespeichert sind, so muss neben dem Dateinamen auch die zugehörige VSN angegeben werden. Privatplatten können nur verwendet werden, wenn sie in die Plattenkonfiguration der Systemeinleitung aufgenommen werden.

Eindeutigkeit der Plattenkonfiguration

Nach SYSBOOT ermittelt SYSIPL die verfügbare Plattenkonfiguration (mittels Online-Scan). Zu diesem Zeitpunkt sind die BS2000-Gerätetabellen, d.h. die generierte HW-Konfiguration, noch nicht verfügbar.

Eine mehrfache Ausführung des Online-Scans wird vermieden, indem für alle identifizierten Platten, die nicht in die Gerätetabelle der Systemeinleitung aufgenommen werden, eine gesonderte Tabelle aufgebaut wird. Im Fall einer späteren Nutzung einer dieser Platten, werden sie in die Gerätetabelle der Systemeinleitung aufgenommen, ohne dass ein erneuter Online-Scan durchgeführt wird.

Privatplatten werden in die Plattenkonfiguration der Systemeinleitung nur dann aufgenommen, wenn die IPL-Platte selbst eine Privatplatte ist, wenn für einen Speicherabzug (SLED) Privatplatten benötigt werden oder wenn der Operator dies explizit anfordert (im Systemeinleitungs-Modus DIALOG mit Auswahl der Option ALLDISK).

Die Systemeinleitung verläuft nur dann problemlos, wenn alle erreichbaren Platten mit einer eindeutigen VSN versehen sind. Wenn mehrere Platten mit derselben VSN erreichbar sind, dann wird dies normalerweise erkannt. Ausnahme: Das aktuelle IOCF der SU /390 ist nicht korrekt.

Werden Platten mit identischer VSN entdeckt, wird folgendermaßen vorgegangen: Wenn die Platten zu dem selben Pubset gehören wie die IPL-Platte, wird die Platte mit dem selben Zeitstempel wie die IPL-Platte ausgewählt. Besitzt keine der Platten den selben Zeitstempel wie die IPL-Platte oder gehören die Platten zu einem anderen Pubset als die IPL-Platte, dann wird geprüft, ob für eine der Platten bereits einmal eine Auswahl durch die Beantwortung der Meldung `NSI2208` getroffen wurde. Wenn ja, wird wieder diese Platte ausgewählt. Dies gilt sowohl für Pubset-Platten als auch für Privatplatten.

> **ACHTUNG!**
> Beim oben genannten Ausnahmefall wird nicht erkannt, wenn die reale und generierte Konfiguration nicht übereinstimmen. Bei der Fortsetzung der Systemeinleitung kann es dann zu unvorhersehbaren Ein-/Ausgabe-Fehlern und zur Zerstörung von Datenträgerinhalten kommen!

Bei der Systemeinleitung werden zur Ermittlung der Plattenkonfiguration normalerweise alle angeschlossenen Geräte einer Prüfung unterzogen. Um – gerade bei umfangreicher Peripherie – dieses sehr zeitaufwändige und fehlerträchtige Verfahren (z.B. Ansprechen nicht benötigter und daher nicht betriebsbereiter Geräte) zu umgehen, wird die Möglichkeit geboten, die für die Systemeinleitung tatsächlich benötigte Plattenkonfiguration (d.h. alle Platten des Home- und der Paging-Pubsets und evtl. verwendete Privatplatten) in einer für diesen Zweck auf der IPL-Platte reservierten Datei SYSDAT.IPL-CONF.DSKnnn abzuspeichern (nnn = Nummer der IPL-Platte im Pubset).

Dynamische Partitionierung der IPL-CONF-Datei

Die Datei IPL-CONF (SYSDAT.IPL-CONF.DSKnnn) wird dynamisch partitioniert. Dadurch können verschiedenen Startup-Konfigurationen in dieser Datei gespeichert werden. Jede Konfiguration in der Konfigurationsdatei wird durch die IPL-CPU-ID, die Home-PVS-ID und das IORSF-Datum identifiziert. Wenn während das Startup eine übereinstimmende Konfiguration gefunden wird, wird diese verwendet und es wird kein Online-Scan durchgeführt, es sei denn, dies wird ausdrücklich angefordert. Wenn keine passende Konfiguration vorhanden ist, wird ein Online-Scan initiiert und eine neue Konfiguration in die Datei geschrieben.

Wenn ein Pubset an verschiedenen Servern als Home-Pubset verwendet wird, gehen die Einträge der Startup-Konfiguration in der Datei IPL-CONF nicht verloren, sondern werden jeweils in einer eigenen Partition abgelegt. Beim Wechsel des Servers wird dann die jeweils gültige Startup-Konfiguration verwendet, ohne dass zwingend ein Online-Scan erfolgen muss.

Eine neue Startup-Konfiguration überschreibt eine bestehende Startup-Konfiguration nicht, sondern wird nach der letzten bestehenden als neue eingetragen. Beim Ändern einer bestehenden Startup-Konfiguration wird die ursprüngliche „verworfen“ und die geänderte Startup-Konfiguration wie eine neue nach der letzten bestehenden eingetragen.

Die Datei IPL-CONF wird jedoch nicht vergrößert: Bei Platzmangel werden sukzessiv zuerst die als ungültig erklärten Startup-Konfigurationen eliminiert. Ist dann der freie Platz in der Datei noch immer nicht ausreichend groß, wird die jeweils am längsten ungenutzte Startup-Konfiguration entfernt. Diese Aktion wird ohne Rückfrage beim Operator durchgeführt und so oft wiederholt, bis der Platz für die neue bzw. geänderte Startup-Konfiguration ausreicht.

> In der Datei IPL-CONF können z.B. 12 Startup-Konfigurationen mit jeweils 255 Platten aufgenommen werden. Bei weniger Platten steigt die Anzahl der Konfigurationen: So sind z.B. 30 Konfigurationen mit jeweils 100 Platten oder 57 Konfigurationen mit jeweils 50 Platten möglich.

Das Speichern einer Startup-Konfiguration in der Datei IPL-CONF wird folgendermaßen veranlasst:

- im Modus DIALOG:
  Beantwortet der Operator die Meldung `NSI1110` mit der IPL-CONF-Option SAVE, wird die Startup-Konfiguration als neue Partition gespeichert.
- in allen Startup-Modi:
  Die Startup-Konfiguration wird gespeichert, wenn:
  - die IPL-Platte eine der Platten des Home Pubsets ist
  - und nicht explizit die IPL-CONF-Option IGNORE angegeben wurde
  - und der Online-Scan ausgeführt wurde.

Der Online-Scan wird automatisch immer dann ausgeführt, wenn der Home-Pubset oder einer oder mehrere Paging-Pubsets vergrößert wurde(n) oder ein Paging-Pubset über BS2000-Parameter (in der Datei SYSPAR.BS2.nnn oder über Konsole) neu angegeben wurde.

Die Datei wird dann bei allen folgenden Systemeinleitungen solange automatisch genutzt, wie die Plattenidentifikation VSN der gespeicherten Pubsets, die Geräteidentifikation MN, der Plattentyp und die Menge der zu den Pubsets gehörigen Platten unverändert bleiben.

Um Änderungen an bestehenden, Startup-Konfigurationen in der Datei SYSDAT.IPL-CONF.DSKnnn zu unterstützen, (z.B. wegen der Verwendung eines vergrößerten oder eines anderen Pubsets), werden – im Modus DIALOG mit der Meldung `NSI1110` – dem Operator bei Angabe der Option IPL-CONF folgende Funktionen angeboten:

- Funktion IGNORE:
  Der Inhalt der Datei SYSDAT.IPL-CONF.DSKnnn wird ignoriert. Ein Online-Scan wird ausgeführt.
- Funktion RESET:
  Die aktuell gültige Startup-Konfiguration in der Datei SYSDAT.IPL-CONF.DSKnnn wird für die aktuelle Systemeinleitung noch genutzt und anschließend für ungültig erklärt.
- Funktion SAVE:
  Die für Startup benötigte Plattenkonfiguration wird in der Datei SYSDAT.IPL-CONF.DSKnnn als neue Partition abgespeichert.

Alle Kombinationen außer SAVE und RESET sind zulässig.

Bestimmung des Home-Pubsets und Prüfung auf Vollständigkeit und Konsistenz

Wird bei der Systemeinleitung die Funktion IPL-CONF nicht genutzt, kann – nach der Ermittlung der verfügbaren Plattenkonfigurationen – der Home-Pubset abhängig von der Art der Systemeinleitung wie folgt bestimmt werden:

- im Modus FAST oder AUTOMATIC:
  Es wird der Pubset zum Home-Pubset gewählt, zu welchem die IPL-Platte gehört.
- im Modus DIALOG:
  Falls die IPL-Platte zu einem Pubset gehört, wird der Pubset zum Home-Pubset gewählt, zu welchem die IPL-Platte gehört.
  Falls die IPL-Platte eine Privatplatte ist oder die IPL-Option ALLDISK angegeben wurde und die Plattenkonfiguration aus mehr als einem Pubset besteht, wird die Kennung (PVSID) des Home-Pubsets mit der Meldung `NSI1135` abgefragt.

Platten mit einer physikalischen Blockgröße von 4KB können nicht Platten des Home-Pubsets sein. Auch SM-Pubsets und Shared-Pubsets werden nicht als Home-Pubset unterstützt.

Der Home-Pubset soll hochverfügbar sein, d.h. durch die Hardware wird die Aufzeichnung der Daten für eine logische Platte auf mehreren physikalischen Platten durchgeführt. Durch diese Redundanz tritt beim Ausfall einer physikalischen Platte kein Datenverlust auf.

Alle in der Startup-Konfiguration vorhandenen und nach VSN-Syntax zum Home-Pubset gehörenden Platten werden mit der auf der Pubres vorhandenen Volumeliste verglichen. Im Modus DIALOG oder in Fehlerfällen werden die zum Home-Pubset gehörenden Platten mit dem letzten Zeitstempel und ihrem mnemotechnischen Gerätenamen MN protokolliert (mit der Meldung `NSI1143` oder `NSI1145`).

Folgende Fehlerfälle werden angezeigt:

- Systemkennung (Sysid) des Home-Pubsets nicht definiert (Meldung `NSI1280`)
  Die Systemeinleitung wird mit dem Standardwert fortgesetzt.
- ungültige Sysid des Home-Pubsets (Meldung `NSI1285`)
  Die Systemeinleitung wird mit dem Standardwert fortgesetzt.
- fehlende Platte im Home-Pubset (Meldung `NSI3215`)
  Die Systemeinleitung kann nicht fortgesetzt werden.
- Zeitstempel einer Platte ungleich dem Zeitstempel der Pubres (Meldung `NSI1148`)
  Ob die Systemeinleitung fortgesetzt werden kann, ist abhängig von der Beantwortung der Meldung `NSI1148`.
- DVS-Attribute einer Platte ungleich der DVS-Attribute des Home-Pubsets (`NSI3220`)
  Die jeweiligen DVS-Attribute werden durch die Meldung `NSI3221` angezeigt.
  Die Systemeinleitung wird abgebrochen, da ein derart inkonsistenter Pubset nicht imcatiert werden darf.
- unbekannte Platte im Home-Pubset (Meldung `NSI1145`)
  Nicht zum Home-Pubset gehörende Platten werden mit dem Zusatz `IS NEW` in der Meldung `NSI1145` angezeigt.

START/STOP-Modus bei Multiprozessoren

Bei Multiprozessor-Systemen steuert der START/STOP-Modus die Auswahl der Verarbeitungsprozessoren bei Nutzung der SVP-START/STOP-Funktion. Im BS2000-Betriebsablauf wird die SVP-START/STOP-Funktion nicht benötigt. Trotzdem muss der START/STOP-Modus korrekt eingestellt werden, damit bei Nutzung der SVP-START /STOP-Funktion oder der Adress-Stop-Funktion keine unerwünschten Nebeneffekte eintreten.

Für die Einstellung des START/STOP-Modus ist Folgendes gültig:

- Wenn alle Verarbeitungsprozessoren betrieben werden, muss der START/STOP-Modus auf ALL CPU (alle Verarbeitungsprozessoren) gesetzt werden, damit bei STOP alle Verarbeitungsprozessoren angehalten und bei START wieder alle gestartet werden. Dieser Wert ist grundsätzlich als Voreinstellung zu verwenden.
- Wenn nur ein Verarbeitungsprozessor betrieben wird oder wenn bei der Systemeinleitung oder Dump-Funktion die SVP-START/STOP-Funktion bzw. der Adress-Stop benötigt wird, muss der START/STOP-Modus auf TARGET CPU (laufende CPU) gesetzt werden. Damit wird verhindert, dass mit der SVP-START-Funktion Verarbeitungsprozessoren gestartet werden, welche nicht mehr oder noch nicht von der Software kontrolliert werden.
- Wenn mehr als ein Verarbeitungsprozessor, aber nicht alle Verarbeitungsprozessoren betrieben werden, ist die korrekte Funktion der SVP-START/STOP-Funktion nicht gewährleistet.

Freigabe belegter Pubsets

Falls die zur Systemeinleitung benötigten Home- und/oder Paging-Pubsets beim letzten Systemlauf (mittels Shutdown) nicht ordnungsgemäß exportiert wurden oder falls sie von einem anderen System belegt sind, wird der Operator aufgefordert, diese freizugeben (Meldung `NSI424A`).
In einer dieser Aufforderung vorausgehenden Meldung wird die Art der Belegung dem Operator genau angezeigt.

Die eindeutige Vergabe von Sysids für mehrere parallel betriebene Home-Pubsets ist zwingend erforderlich.

Die Freigabe darf nur dann erfolgen, wenn diese Pubsets zu diesem Zeitpunkt nicht von einem anderen System verwendet werden. Bei Freigabe ohne sorgfältige Prüfung können die Daten der Pubsets durch uneingeschränkten Zugriff zweier Systeme zerstört werden.

Parameterservice

Zur korrekten Durchführung der Systemeinleitung ist die Verwendung einer Startup-Parameterdatei notwendig. Mit dem Parameterservice werden die Software-Komponenten mit Daten versorgt.

Zur genauen Beschreibung des Parameterservice mit Aufbau und Inhalt der Parameterdateien siehe Kapitel „Parameterservice".

Protokollierung des Startup-Dialogs an der IPL-Konsole

Alle Meldungen, auch solche, die in den Modi FAST und AUTOMATIC nicht an der Konsole ausgegeben werden, sowie alle Antworten werden im Hauptspeicher protokolliert und ab Verfügbarkeit der Funktion CONSLOG in die Protokolldatei übernommen: $SYSAUDIT.SYS.CONSLOG.datum.session-nr.lfd-nr.

Zu Aufbau und Inhalt der Protokolldatei CONSLOG siehe das „Diagnosehandbuch“ [14].

Falls die Systemeinleitung vor Verfügbarkeit der Funktion CONSLOG beendet wird, ist der Konsoldialog in einem anschließenden Dump vorhanden.

2.1.4 Systemeinleitung an SU x86

Auf SU x86 (x86-64-Architektur) wird die Konfiguration der Bus- und Fibre Channel-Peripherie über das Trägersystem X2000 festgelegt und beim Startup dynamisch ermittelt. Die SU x86 benötigt keine Hardware-Generierung mit IOGEN.

Dateiname	Nutzung
statt SYSPRG.IPL.<ver>: SKMPRG.IPL.<ver>	Ladeobjekte SYSBOOT, SYSIPL + SYSIPLEX und SLED, Standarddatei
statt SYSPRG.STRT.<ver>: SKMPRG.STRT.<ver>	Ladeobjekt SYSSTART + SYSIPLEX
statt SYSPRG.BS2.<ver>: SKMPRG.BS2.<ver>	Ladeobjekt BS2000-CL1/2-Exec

Tabelle 1: Abweichende Standarddateinamen an SU x86

Automatischer IPL und zeitgesteuerter Startup/Shutdown

Auf SU x86 kann die Systembetreuung (über den SE Manager) in X2000 einen automatischen IPL, einen zeitgesteuerten Startup oder Shutdown konfigurieren. Der Systemparameter SHUTPROC bestimmt bei einer Systembeendigung über X2000, ob der Shutdown sofort oder durch einen Enter-Job ausgeführt wird.
Falls der automatische IPL eingestellt wurde, wird nach Hochfahren des X2000 automatisch ein IPL von der voreingestellten IPL-Platte eingeleitet.

2.2 Arten der Systemeinleitung

Die Systemeinleitung kann komfortabel oder flexibel ablaufen. Für einen komfortablen Ablauf stehen die Modi FAST und AUTOMATIC zur Verfügung, für einen flexiblen Ablauf der Modus DIALOG.

Die Sicherstellungsdatei (Save-Datei) hat den versionsunabhängigen Namen SYS.NSI.SAVEREP. Sie wird – außer bei Privatplatten – beim Einrichten einer IPL-Platte von SIR angelegt.

Zu den Dateinamen für die Parameterdatei siehe Kapitel „Parameterservice".

2.2.1 FAST-Startup

Im Modus FAST läuft die Systemeinleitung dialogfrei ab. Dafür müssen die zu ladenden Programme, Objektcodekorrekturen (Reps) und Parameter in Dateien mit versionsspezifischen Standardnamen gespeichert sein oder die Namen der Dateien in der Parameterdatei hinterlegt werden. Die Parameterdatei selbst muss mit dem versionsspezifischen Dateinamen mit oder ohne Suffix bezeichnet sein.

Auf der IPL-Konsole werden die wichtigsten Ereignisse der Systemeinleitung protokolliert. Das ausführliche Protokoll – wie im Modus DIALOG – wird in die Protokolldatei CONSLOG geschrieben.

Fehlen die Dateien SYSREP.BS2.<ver>, SYSREP.STRT.<ver>, SYSPAR.BS2.<ver>[.<name>], SYSPRG.STRT.<ver> bzw. SYSPRG.BS2.<ver> und ist in der Parameterdatei keine Rep-Datei oder kein BS2000 angegeben, dann können die Korrekturdaten und Parameter wahlweise von Plattendateien mit beliebigen Namen oder über die Konsole im Operatordialog nachgereicht werden (siehe Kapitel „Parameterservice" und Abschnitt „Systemkorrekturen").

Die Wahl des Startup-Typs und die Entscheidung über eine evtl. Rekonstruktion des Benutzerkatalogs ist von den vorher eingestellten Werten der Systemparameter RECONUC und STUPTYPE abhängig.
Zur Wahl des Startup-Typs siehe auch Abschnitt „Die Auswahl von Startup-Typ und Katalog-Rekonfiguration".

Folgende Standarddateinamen sind bei dialogfreier Systemeinleitung gültig und müssen unter der Benutzerkennung TSOS auf dem Home-Pubset katalogisiert sein:

Dateiname	Nutzung
SYSPRG.BOOT.DSKnnn.SAVE	Sicherstellungsbereich [1]
SYSPRG.IPL.DSKnnn	Ladeobjekte SYSBOOT, SYSIPL und SLED, beim Startup verwendet [2]
SYSREP.IPL.DSKnnn	Reps für SYSIPL, beim Startup verwendet [2]
SYSDAT.IPL-CONF.DSKnnn	Sicherstellungsbereich für die Funktion IPL-CONF [1]
SYSPRG.STRT.<ver>	Ladeobjekt SYSSTART
SYSREP.STRT.<ver>	Reps für SYSSTART
SYSREP.BS2.<ver>	Objektcode-Korrekturen (Reps) [3]
SYS.NSI.SAVEREP	Zwischenspeicher für Rep-Protokollierung [4]
SYSPAR.BS2.<ver>[.<name>]	Parameter
SYSPRG.BS2.<ver>	Ladeobjekt BS2000-CL1/2-Exec [3]

[1] Die Datei wird von SIR als leere Sicherungsdatei angelegt und im SVL verankert. „nnn“ im Namensbestandteil „DSKnnn“ entspricht der Nummer der Platte, auf der die Datei von SIR angelegt wurde. Diese Dateinamen werden beim Startup nicht geprüft. Dateien mit einem platten-spezifischen Namen werden bei einer logischen Sicherung nicht berücksichtigt.

[2] Die Datei wurde durch SIR von einer versionsabhängigen Lieferdatei in diese versionsunabhängige Datei kopiert (Suffix „DSKnnn“) und im SVL verankert. „nnn“ im Namensbestandteil „DSKnnn“ entspricht der Nummer der Platte, auf der die Datei von SIR angelegt wurde. Diese Dateinamen werden beim Startup nicht geprüft. Dateien mit einem platten-spezifischen Namen werden bei einer logischen Sicherung nicht berücksichtigt.

[3] Der Dateiname kann mittels Startup-Parameterservice vorgegeben werden.

[4] Die Datei wird von SIR als leere Sicherungsdatei angelegt. Fehlt diese Datei oder ist sie zu klein, hat dies Auswirkung auf den „Sicheren Systemstart“.

Falls eine der Dateien nicht vorhanden ist oder nicht genutzt werden kann, wird der Dateiname über Konsole angefordert (nicht bei verankerten Dateien).

Durch die platten-spezifische Benennung der verankerten, d.h. im SVL hinterlegten Dateien können auch mehrere IPL-Platten in einem Pubset parallel eingerichtet werden.

2.2.2 AUTOMATIC-Startup

Der Modus AUTOMATIC unterstützt den unbedienten Betrieb. In Verbindung mit einem automatischen Hochfahren des Systems nach abnormaler Systembeendigung und eingeschaltetem automatischen Restart wird der AUTOMATIC-Startup angewendet. Seine Eigenschaften und die Protokollierung entsprechen dem Modus FAST.

Der Modus AUTOMATIC darf nur verwendet werden, wenn die SVP-Uhr funktioniert (dieser Punkt entfällt bei Restart) und die Platten des Home-Pubsets und die Paging-Platten nur von diesem System benutzt werden.

Im Modus AUTOMATIC werden in bestimmten Fehlersituationen Standardreaktionen durchgeführt. Vom Operator wird in diesem Fall keine Entscheidung angefordert.
Ein Beispiel für solche Reaktionen ist die automatische Freigabe benötigter Platten, die wegen abnormaler Systembeendigung noch gesperrt sind.

Zur Beschreibung des automatischen Restart siehe Abschnitt „Automatischer Restart".

Die Wahl des Startup-Typs und die Entscheidung über eine evtl. Rekonstruktion des Benutzerkatalogs ist von den vorher eingestellten Werten der Systemparameter RECONUC und STUPTYPE abhängig.
Zur Wahl des Startup-Typs siehe auch Abschnitt „Die Auswahl von Startup-Typ und Katalog-Rekonfiguration".

2.2.3 DIALOG-Startup

Im Modus DIALOG kann der Operator die Systemeinleitung selbst beeinflussen und spezielle Funktionen auswählen. In einem ausführlichen Dialog erhält der Operator Meldungen am Bildschirm, durch die die erlaubten Eingaben angezeigt werden. Mit ihrer Beantwortung wählt der Operator im Rahmen der gegebenen Möglichkeiten die gewünschten Funktionen aus. Diese Funktionen können sein:

- Freie Auswahl der IPL-Platte, der Eingabemedien für die zu ladenden Programme, Objektcodekorrekturen (Reps) und Parameter
- Aktivieren von bestimmten Testfunktionen oder Funktionen zur Behandlung von Ausnahmesituationen

Weil der Operator mit dem DIALOG-Startup die Vorgaben der Systembetreuung für die Systemeinleitung (Parameter, Reps) ändern kann, wird als Sicherheitsfunktion ein spezieller Schutz für den DIALOG-Startup angeboten:
Die Systemeinleitung von BS2000 im Modus DIALOG kann optional auf eine bestimmte Konsole eingeschränkt werden. Diese Einschränkung kann nur im Modus DIALOG eingestellt und wieder aufgehoben werden; sie bezieht sich stets auf die aktuell verwendete Konsole. Als Berechtigungsschlüssel wird der mnemotechnische Gerätename der Konsole im Home-Pubset hinterlegt.

Ablauf des DIALOG-Startup

Der Ablauf des DIALOG-Startups wird ausführlich auf der IPL-Konsole protokolliert.

Im Modus DIALOG erhält der Operator die folgende Meldung, die ihn zur Eingabe von Optionen auffordert:

```
NSI1110 ENTER OPTIONS OR EOT.
        REPLY (UNLOCK,TEST,ALLDISK,DRV-SELECT,CREATE-DRV,IPL-CONF)
```

Es sind beliebige Kombinationen möglich. Die Optionen haben im Einzelnen folgende Bedeutung:

UNLOCK: bewirkt, dass in Konfliktsituationen bei der Bestimmung der Systemzeit fortgesetzt werden bzw. die Bestimmung der Systemzeit durch den Operator erfolgen kann.

TEST: bewirkt, dass dem Operator mit der Meldung NSI1113 die Eingabe von Optionen für bestimmte Testsituationen angeboten wird.

```
NSI1113   ENTER TEST-OPTIONS OR EOT.
REPLY (DUMPTEST, STRTNAME, REPERRIGN, WATCH, ASYNSKPIN)
```

Bedeutung der Test-Optionen

DUMPTEST:
Testoption, dient nicht zur Systemeinleitung.

STRTNAME:
bewirkt, dass vor dem Laden von SYSSTART die Dateinamen für das Ladeobjekt und für die SYSSTART-Rep-Datei erfragt werden.

REPERRIGN:
Testoption, dient nicht zur Systemeinleitung.

WATCH:
Testoption, dient nicht zur Systemeinleitung. Sie unterstützt die Fehlerdiagnose beim Online-Scan. Für ein bestimmtes Gerät wird eine Überwachung aktiviert, die die Systemeinleitung bei Problemen mit diesem Gerät abbricht. Somit können Diagnoseunterlagen (SLED) genau zum Fehlerzeitpunkt erstellt werden.

ASYNSKPIN:
Testoption, dient nicht zur Systemeinleitung.

ALLDISK: In der von Startup genutzten Plattenkonfiguration werden auch Privatplatten erfasst (die Erfassung erfolgt durch einen Online-Scan), so dass Ladeobjekte und Rep- bzw. Parameterdateien auch von Privatplatte gelesen werden können. Wenn die IPL-Platte eine Privatplatte ist, dann wird die Option ALLDISK automatisch gesetzt. Den gewünschten Home-Pubset erfragt Startup über folgende Meldung:

```
NSI1135 ENTER ACTUAL HOME-PVS IDENTIFIER. REPLY (PVSI)
```

Es wird implizit die IPL-CONF-Option IGNORE gesetzt. Die Verwendung dieser Option ist nur möglich, wenn max. 1290 Platten online sind.
Im Fehlerfall wird folgende Meldung ausgegeben:

```
NSI2335 OVERFLOW OF STARTUP DEV.TABLE; SWITCH UNNEEDED DISKS OFF
REPLY (R(RETRY); S(STOP))
```

Je nach Antwort wird der Online-Scan wiederholt oder die Systemeinleitung abgebrochen.

DRV-SELECT: bewirkt, dass die Auswahl der für Startup verwendeten Platten eines DRV-Plattenpaares vom Operator getroffen werden soll.
Dies ist sinnvoll, wenn nach einem Plattenfehler auf einer DRV-Platte während des Startups beim neuen Startup auf die andere Platte eines DRV-Plattenpaares ausgewichen werden soll.
Es wird implizit die IPL-CONF-Option IGNORE gesetzt.

CREATE-DRV: bewirkt, dass durch implizites Starten des Subsystems DRV nach der Systemeinleitung aus einem SRV-Home-Pubset (Single Recording by Volume) ein DRV-Pubset im Zustand DRV-MONO gemacht wird.

`IPL-CONF:` mit dieser Option wird die Behandlung der server-spezifischen Partition der auf der IPL-Platte immer eingerichteten Datei SYSDAT.IPL-CONF.DSKnnn gesteuert (zur Bedeutung dieses Parameters siehe auch "Hinweise zur Systemeinleitung"). Wenn diese Option nicht angegeben wird, wird wie in allen anderen Startup-Modi (FAST, AUTOMATIC) versucht, die server-spezifische Partition der Datei auszuwerten.

Bei erfolgreicher Abarbeitung der server-spezifischen Partition wird der Online-Scan unterdrückt.

Bei Angabe dieser Option wird der Operator mit der Meldung `NSI1116` aufgefordert, die Bearbeitung der server-spezifischen Partition in der Datei SYSDAT.IPL-CONF.vvv festzulegen:

```
NSI1116  ENTER IPL-CONF OPTION OR EOT.
REPLY (IGNORE; RESET; SAVE. EOT=IGNORE)
```

Bedeutung der Angaben

IGNORE:
Die server-spezifische Partition in der Datei SYSDAT.IPL-CONF.DSKnnn wird ignoriert. Ein Online-Scan wird ausgeführt. Diese Option wird implizit gesetzt, wenn die Option ALLDISK angegeben wurde.

RESET:
Die für diesen Server aktuell gültige Partition in der Datei SYSDAT.IPL-CONF.DSKnnn wird für die aktuelle Systemeinleitung noch genutzt und anschließend für ungültig erklärt.

SAVE:
Die für Startup benötigte Plattenkonfiguration wird in der Datei SYSDAT.IPL-CONF.DSKnnn als neue Partition abgespeichert.

Alle Kombinationen außer SAVE und RESET sind zulässig.

Nach Verarbeitung der eingegebenen Optionen ermittelt die Startup-Routine die verfügbare Plattenkonfiguration. Zu diesem Zweck wird entweder eine auf der IPL-Platte vorhandene und gültige Partition in der Datei SYSDAT.IPL-CONF.DSKnnn ausgewertet (falls nicht durch die Option IGNORE explizit ausgeschlossen) oder der Online-Scan ausgeführt. Falls dabei Platten dieselbe VSN haben, muss die Plattenkonfiguration auf Platten mit eindeutiger VSN eingeschränkt werden (siehe dazu Abschnitt „Eindeutigkeit der Plattenkonfiguration" (Hinweise zur Systemeinleitung)).

Falls an dem System mehrere Pubsets verfügbar sind, wird der Operator aufgefordert, den Home-Pubset zu bestimmen (`NSI1135`). Außer Home-Pubset und CPUs werden dann noch Datum und Uhrzeit protokolliert. Gegebenenfalls wird der Operator zur Bestätigung oder Korrektur aufgefordert.

Zum Laden von SYSSTART wird stets der Standard-Dateiname herangezogen.
Der Operator erhält nur dann die folgende Aufforderung, den Dateinamen anzugeben, wenn der Standard-Dateiname im Katalog nicht gefunden wird, oder bei der Eingabe der Test-Optionen explizit STRTNAME angegeben wurde:

```
NSI1190 ENTER STARTUP-FILENAME.
        REPLY (FILENAME(,VOL=VSN); EOT (USE STANDARD FILE))
```

Für die SYSSTART-Rep-Datei wird der Standard-Dateiname verwendet.
Ist diese Datei nicht vorhanden oder wurde die Option STRTNAME gewählt, wird der Name der Datei bzw. das Eingabemedium über die Meldung `NSI0050` erfragt (siehe "Rep-Sätze").

Danach wird der Dateiname und das Eingabemedium für die Parameter erfragt (siehe auch Kapitel „Parameterservice"). Anschließend erscheint wieder die Meldung:

```
NSI1190 ENTER BS2000-FILENAME.
        REPLY (FILENAME(,VOL=VSN); EOT (USE STANDARD FILE))
```

Der Operator kann den Dateinamen des zu ladenden BS2000-Systems eingeben und die Archivnummer, falls die Datei sich auf Privatplatte befindet. Wird kein Dateiname angegeben, wird die Parameterdatei ausgewertet. Ist auch dort kein Dateiname eingetragen, wird BS2000 aus der Standarddatei SYSPRG.BS2.<ver> geladen.

Im Anschluss erfolgt die Rep-Verarbeitung (siehe "Rep-Sätze"), dann werden die ausgewählten Seitenwechselplatten protokolliert.

Mit folgender Meldung wird der Operator zur Angabe des Namens einer Kommandodatei aufgefordert, deren Kommandos nach „System Ready“ automatisch ausgeführt werden:

```
NSI0077 ENTER AUTOMATIC COMMAND FILE NAME.
        REPLY (FILE NAME; N(DO NOT USE); EOT(USE STANDARD))
```

Mit folgender Meldung wird der Operator aufgefordert, die Art des Systemstarts (siehe "Die Auswahl von Startup-Typ und Katalog-Rekonfiguration") anzugeben:
(siehe auch die Beschreibung des Systemparameters STUPTYPE)

```
NSI6005 SYSTEM PARAMETER STUPTYPE = (&00). SHALL VALUE BE CHANGED?
        REPLY ( U(NCHANGED), W(ARM), C(OLD), S(ELECTIVE), Z(IP),
        T(SN FILE RESET ONLY), J(OIN AND TSN FILE RESET), EOT=UNCHANGED)
```

Durch Beantwortung folgender Meldung kann durch den Operator bestimmt werden, ob und wie eine Benutzerkatalog-Rekonstruktion durchgeführt werden soll (siehe dazu die Beschreibung des Systemparameters RECONUC).

```
NSI6010 SYSTEM PARAMETER RECONUC = (&00). SHALL VALUE BE CHANGED?
        REPLY ( U(NCHANGED), N(O), B(ACKUP), T(SOSCAT), A(LL), R(ESET),
        EOT=UNCHANGED )
```

Der weitere Ablauf der Systemeinleitung ist durch DSSM gesteuert (siehe auch Hinweis auf "Überblick über die Abläufe bei der Systemeinleitung"). Mit der Meldung „System Ready“ ist die Systemeinleitung beendet.

2.2.4 Systemeinleitung mit wechselndem Modus

Um auch in Sonderfällen – Erprobung neuer Software, Parameter oder Korrekturen – die Systemeinleitung möglichst dialogfrei ablaufen zu lassen, kann in der Parameterdatei neben den Abweichungen vom Standardfall auch die Option verankert werden, nach Abarbeitung der Vorgaben in einem anderen Modus fortzufahren. Der Operator wählt also im SVP-Menü den Modus DIALOG und gibt in dessen Verlauf die passende Parameterdatei an. Mit dem Parameterdatei-Eintrag NEW-IPL-MODE=FAST unter dem Schlüsselwort SYSOPT-IPL kann die Systembetreuung dann vorgeben, dass nach Auswertung der Parameterdatei die Systemeinleitung im Modus FAST, also unbedient, fortgesetzt wird.

Zum Parametersatz SYSOPT-IPL siehe auch Abschnitt „Änderung von IPL-Optionen (SYSOPT-IPL)".

2.2.5 Die Auswahl von Startup-Typ und Katalog-Rekonfiguration

Die Entscheidung, welcher Systemstart gewählt werden soll, trifft der Operator beim DIALOG-Startup durch Beantworten der Meldung

```
NSI6005 SYSTEM PARAMETER STUPTYPE = (&00). SHALL VALUE BE CHANGED?
        REPLY ( U(NCHANGED), W(ARM), C(OLD), S(ELECTIVE), Z(IP),
        T(SN FILE RESET ONLY), J(OIN AND TSN FILE RESET), EOT=UNCHANGED)
```

Bei den Modi FAST und AUTOMATIC kann diese Entscheidung nur durch Belegung des Systemparameters STUPTYPE getroffen werden.

Beim DIALOG-Startup sind folgende Antworten möglich:

- U=Unchanged

  Der Systemstart erfolgt wie von der Systembetreuung durch den Systemparameter STUPTYPE vereinbart.
- W=Warmstart

  Der Warmstart ist die Standardvorbelegung des Systemparameters STUPTYPE.

  Alle Jobs, die noch in den JMS-Auftragswarteschlangen sind (Verzeichnis mit Benutzerkennungen und Auftragsfolgenummern für Type 1 und Type 2) und nicht während des vorangegangenen Systemlaufs bearbeitet wurden, werden jetzt durch die Auftragssteuerung wieder zur Verarbeitung bereitgestellt.
- C=Kaltstart

  Bei diesem Typ des Systemstarts bleiben Jobs in der Warteschlange unberücksichtigt. Sie werden ohne Bearbeitung gelöscht. Wenn z.B. ein Plattenfehler in den Auftragswarteschlangendateien einen Systemabbruch verursacht, muss der nächste Systemlauf immer mit einem Kaltstart eingeleitet werden. Denn nach einem Plattenfehler muss die Warteschlange vollständig neu aufgebaut werden. Der Kaltstart sollte nach Möglichkeit vermieden werden, da dadurch Termin- und Repeat-Jobs verloren gehen. Für Termin- und Repeat-Jobs können vor dem Kaltstart mit dem Dienstprogramm JMP die erforderlichen ENTER-JOB-Kommandos rekonstruiert werden (siehe Abschnitt „JMP:Rekonstruktion von Batch-Jobs").
- T=First-Start mit Rücksetzen der TSN-Datei

  Dieser Typ des Systemstarts muss bei der erstmaligen Benutzung eines Pubsets verwendet werden.

  Das Zurücksetzen der TSN-Datei ($TSOS.TSOSJOIN.TSNN) bewirkt, dass die JMS-Auftragswarteschlange und die TSN-Datei zurückgesetzt werden.
  D.h. existierende Jobs werden gelöscht, die Session-Nummer wird auf 1 zurückgesetzt und die nächste Auftragsnummer (tsn), die das System vergibt, ist – je nach TSN-Modus – 0001 oder 0AAA.

- J=First-Start mit Rücksetzen des Benutzerkatalogs

  Es werden der Benutzerkatalog, die JMS-Auftragswarteschlange und die TSN-Datei zurückgesetzt.

  Das Zurücksetzen des Benutzerkatalogs bewirkt, dass ein vorhandener alter Benutzerkatalog gelöscht und ein neuer erzeugt wird. Dieser neue Benutzerkatalog enthält nur Einträge für die System-Benutzerkennungen (SERVICE, SYSAUDIT, SYSDUMP, SYSGEN, SYSHSMS, SYSMAREN, SYSNAC, SYSPRIV, SYSSNAP, SYSSNS, SYSSOPT, SYSSPOOL, SYSUSER, SYSROOT, SYSOPR, SYSWSA und TSOS).

  > **ACHTUNG!**
  > Alle Dateien außer denen auf der Kennung TSOS werden gelöscht!
  > Außer zu den Systemdateien ist kein Zugriff (über den Katalog) zu sämtlichen Benutzerdateien mehr möglich. Auch der von diesen gelöschten Benutzerdateien zuvor belegte Speicherplatz wird freigegeben.

  Die Benutzerkennungen des Systems (mit Ausnahme von TSOS und SERVICE) erhalten die Abrechnungsnummer SYSACC und sind gesperrt. Durch das Kommando UNLOCK-USER können diese Kennungen von der Systembetreuung zugänglich gemacht werden. Sie sollten dann jedoch mit Zugangsschutzattributen versehen werden.

  Das Zurücksetzen der TSN-Datei ($TSOS.TSOSJOIN.TSNN) bewirkt, dass die JMS-Auftragswarteschlange und die TSN-Datei zurückgesetzt werden.
  D.h. existierende Jobs werden gelöscht, die Session-Nummer wird auf 1 zurückgesetzt und die nächste Auftragsnummer (tsn), die das System vergibt, ist – je nach TSN-Modus – 0001 oder 0AAA.

- S=Selective-Start

  Bei diesem Typ des Systemstarts kann der Operator bestimmte Warteschlangen angeben, die erhalten bleiben sollen, während die übrigen vom System wie bei einem Kaltstart behandelt werden.

  Nach der Antwort S auf die Meldung NSI6005 und dem Beantworten der Meldung NSI6010 erscheint die Meldung

```
? JOBP... JMS0510 SPECIFY QUEUE(S) OF BATCH JOBS TO BE SAVED WITH
          SELECTIVE START. REPLY: (1: AWAITING PROCESSING; 2: BEING
          PROCESSED)
```

  Durch Beantworten der Meldung JMS0510 können die JMS-Jobs (in den Warteschlangen Q1 und Q2) angegeben werden, die erhalten bleiben sollen.

  Das Format der Antwort ist: JOBP.n

  wobei
  n=1 die Auftragswarteschlange für zu verarbeitende Batch-Jobs und
  n=2 die Batch-Jobs, die in Bearbeitung waren, bezeichnet.

  Bei Angabe von JOBP. werden keine Warteschlangen ausgewählt.

  Beispiel

```
JMS0510 SPECIFY QUEUE(S) OF BATCH JOBS TO BE SAVED WITH SELECTIVE
        START. REPLY: (1: AWAITING PROCESSING; 2: BEING PROCESSED)
JOBP.1
```

  Danach werden die alten Warteschlangendateien (JOBPOOL-Datei für die JMS-Jobs in den Warteschlangen Q1 und Q2) daraufhin geprüft, ob sie Jobs enthalten, die erhalten bleiben sollen. Ist dies der Fall, so werden die Einträge der Auftragsbearbeitung in die neue Warteschlangendateien übernommen und die zugehörigen Zähler entsprechend gesetzt. Wenn die Dateien gelesen worden sind, werden sie geschlossen und aus dem Katalog gestrichen. Die neuen Dateien werden im neuen Systemlauf abgearbeitet.

- Z=Zip-Start

  Dieser Typ des Systemstarts wird angewendet, wenn für einen normalen Start nicht mehr genug freier Plattenspeicherplatz zur Verfügung steht. Bei der Antwort Z auf die Meldung NSI6005 wird kein Versuch unternommen, Plattenspeicherraum für Warteschlangendateien zuzuweisen. Diese Betriebsart ist nur vorgesehen, um nach dem Einleiten des Systems Dateien auszulagern oder zu löschen, um wieder freien Speicherplatz auf den Platten zu erhalten. In diesem Systemlauf sollten keine anderen Aufgaben durchgeführt werden.

Durch Beantwortung folgender Meldung kann im DIALOG-Startup in analoger Form durch den Operator bestimmt werden, ob und wie eine Rekonstruktion des Benutzerkatalogs durchgeführt werden soll:
(siehe dazu die Beschreibung des Systemparameters RECONUC)

```
NSI6010 SYSTEM PARAMETER RECONUC = (&00). SHALL VALUE BE CHANGED?
        REPLY ( U(NCHANGED), N(O), B(ACKUP), T(SOSCAT), A(LL), R(ESET),
        EOT=UNCHANGED )
```

2.2.6 Beispiel für einen DIALOG-Startup

Das folgende Beispiel zeigt den Ablauf eines DIALOG-Startups (in Auszügen) an einer SU /390. Die Darstellung entspricht der Protokollierung in der CONSLOG-Datei (teilweise gekürzt); für die Ausgabe auf die Konsole können sich Abweichungen bezüglich des Meldungsstroms ergeben.

```
TCLOG    .125439  ***2021-01-25*** 000001 **** UTC+01:00
************************************************************
(C0) %   P-000.125439 NSI00E3 IPL-REPS READ: 1; EXECUTED: 1
(C0) %   P-000.125439 NSI1100 IPL DEVICE = MH20.0; IPL PATH = BE18 (MN=BE18)
(C0) %   P-000.125439 NSI1106 *** BS2000 DIALOGUE STARTUP ***
(C0) ?   P-000.125439 NSI1110 ENTER OPTIONS OR EOT.
                      REPLY (UNLOCK,TEST,ALLDISK,DRV-SELECT,CREATE-DRV,IPL-CONF)
   P R(C0)-000.125439.
-------------------------------------------------------------------------------  1.
(C0) %   P-000.125439 NSI3135 IPL DISK-SETUP READ FROM IPL-CONF PREPARED 2021-01-12 13:29:55
(C0) %   P-000.125439 NSI1143 002 PUBVOLS OF HOME-PVS MH20 WITH SYSID 117 ONLINE
-------------------------------------------------------------------------------  2.
(C0) %   P-000.125439 NSI1145 MH20.0 2021-01-25 12:53:37 ON BE18
(C0) %   P-000.125439 NSI1145 MH20.1 2021-01-25 12:53:37 ON BE19
(C0) %   P-000.125439 NSI1153 STATE OF PROCESSORS ONLINE:
(C0) %   P-000.125439 NSI1155 CPU 00 ONLINE, ATTACHED (IPL CPU)
(C0) %   P-000.125439 NSI1158 CPU 01 ONLINE, DETACHED
(C0) %   P-000.125439 NSI1163 LOCAL DATE = 2021-01-25, TIME = 12:53:52 FROM SVP
-------------------------------------------------------------------------------  3.
(C0) %   P-000.125439 NSI1180 LOAD ADDRESS OF SYSPRG.STRT.210 = 00777000
(C0) %   P-000.125439 NSI00E6 SYSREP.STRT.210 IS USED AS REP FILENAME
-------------------------------------------------------------------------------  4.
(C0) ?   P-000.125439 NSI0050 SPECIFY PARAM FILE OR DEVICE.
                      REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
   P R(C0)-000.125439.
-------------------------------------------------------------------------------  5.
(C0) %   P-000.125439 NSI00E6 SYSPAR.BS2.210.VM07S700 IS USED AS PARAM FILENAME
(C0) %   P-000.125439 NSI006B FILE SYSPAR.BS2.210.VM07S700 NOT FOUND
(C0) %   P-000.125439 NSI00BC LAST PARAM FILE/DEVICE IGNORED
(C0) %   P-000.125439 NSI00E6 SYSPAR.BS2.210 IS USED AS PARAM FILENAME
(C0) ?   P-000.125439 NSI0050 SPECIFY PARAM FILE OR DEVICE.
                      REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
   P R(C0)-000.125439.END
-------------------------------------------------------------------------------  6.
(C0) ?   P-000.125439 NSI1190 ENTER BS2000-FILENAME.
                      REPLY (FILENAME(,VOL=VSN); EOT (USE STANDARD FILE))
   P R(C0)-000.125439.
-------------------------------------------------------------------------------  7.
(C0) ?   P-000.125439 NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
                      REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
   P R(C0)-000.125439.
-------------------------------------------------------------------------------  8.
(C0) %   P-000.125439 NSI00E6 SYSREP.BS2.210 IS USED AS REP FILENAME
(C0) ?   P-000.125439 NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
                      REPLY (EOT (END); FN=FILENAME(,VOL=VSN);CONS;END)
   P R(C0)-000.125439.END
-------------------------------------------------------------------------------  9.
(C0) %   P-000.125439 NSI0028 AVAILABLE MEMORY = 4.0 GB. CPU = 60029001 26000000
------------------------------------------------------------------------------ 10.
(C0) %   P-000.125439 NSI5103 BS2000 : NAME = G10BXS, VERSION = 21.0A00,
                      GEN-TIME = 2020-10-06 17:34:20
(C0) %   P-000.125439 NSI5102 COPYRIGHT (C) 2021 FUJITSU TECHNOLOGY SOLUTIONS GMBH. ALL
```

```
RIGHTS RESERVED
(C0) %   P-000.125439 EMM2960 EFFECTIVE SYSSIZE VALUE IS 240 MB
(C0) %   P-000.125439 NSI5150 CPU'S ONLINE: 2  CPU'S INSTALLED: 2
(C0) %   P-000.125439 ETMGT10 GTIME-VALUES FOR SESSION: ZONE=+01:00, DIFF=1:00
(C0) %   P-000.125439 ETMGT01 TODR-EPOCH FOR SESSION: EPOCH=00
(C0) %   P-000.125439 NSI3130 IPL-CONF SAVED ONTO DISK MN=BE18 WITHOUT ERRORS
(C0) %   P-000.125439 NSI5104 IOCF: NAME = SU700001EM-2 29001 26 , 10.10.2020
                      GEN-TIME = 2021-01-10 11:04:17
(C0) %   E-000.125439 EMM2301 PAGING AREA ON VOLUME MH20.0 SIZE = 02560.00 MB.
(C0) %   E-000.125439 EMM2301 PAGING AREA ON VOLUME MH20.1 SIZE = 02560.00 MB.
(C0) %   E-000.125439 EMM2850 THE SIZE OF THE PAGING-AREA IS 0005120.00 MB;
                      THE RESERVED SIZE OF THE PAGING-AREA IS 0000011.48 MB.
(C0) %   P-000.125439 NSI6102 BS2000 EXECUTIVE: CLASS2 ACTIVE
------------------------------------------------------------------------------- 11.
(C0) %   P-000.125439 NSI4110 /INPUT DISK F=SYSPAR.BS2.210
------------------------------------------------------------------------------- 12.
(C0) %   P-000.125439 NSI4110 /BEGIN GTIME
(C0) %   P-000.125439 NSI4110 ZONE=+00:00
(C0) %   P-000.125439 NSI4110 DIFF=1:00
(C0) %   P-000.125439 NSI4110 SEASON=S
(C0) %   P-000.125439 NSI4110 EPOCH=00
(C0) %   P-000.125439 NSI4110 CHDATE=1900-01-01/00:00
(C0) %   P-000.125439 NSI4110 CHDATE=1980-03-16/02:00
(C0) %   P-000.125439 NSI4110 CHDATE=1980-10-26/03:00
...
(C0) %   P-000.125439 NSI4110 CHDATE=2024-03-31/01:00
(C0) %   P-000.125439 NSI4110 CHDATE=2024-10-27/02:00
(C0) %   P-000.125439 NSI4110 NEXTZONE
(C0) %   P-000.125439 NSI4110 ZONE=+01:00
(C0) %   P-000.125439 NSI4110 DIFF=1:00
(C0) %   P-000.125439 NSI4110 SEASON=S
(C0) %   P-000.125439 NSI4110 CHDATE=1900-01-01/00:00
(C0) %   P-000.125439 NSI4110 CHDATE=1980-04-06/02:00
(C0) %   P-000.125439 NSI4110 CHDATE=1980-09-28/03:00
...
(C0) %   P-000.125439 NSI4110 CHDATE=2021-03-28/02:00
(C0) %   P-000.125439 NSI4110 CHDATE=2021-10-31/03:00
(C0) %   P-000.125439 NSI4110 NEXTZONE
(C0) %   P-000.125439 NSI4110 /EOF
(C0) %   P-000.125439 NSI4110 /BEGIN OPR
(C0) %   P-000.125439 NSI4110 DEFINE-CONSOLE CONSOLE=C0
(C0) %   P-000.125439 NSI4110 SET-CODE CODE=*ALL,CONSOLE=*IPL
(C0) %   P-000.125439 NSI4110 SET-CODE CODE=*ALL,CONSOLE=(CON1,CON2,CON3,CON4,CON5,CON6,CON7)
(C0) %   P-000.125439 NSI4110 SET-CODE CODE=T,CONSOLE=MARE
(C0) %   P-000.125439 NSI4110 SET-CODE CODE=E,CONSOLE=MARE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE E,ADD-FILE-LINK
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE $,MODIFY-SPACE-SATURATION-LEVELS
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE $,MODIFY-USER-PUBSET-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE $,PRINT-DOCUMENT
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SEVER
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CATJV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,FSTATUS
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-SPOOL-CHARACTER-SETS
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CATM
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-SPOOL-DEVICES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,COPY
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,IMPORT
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-SPOOL-FORMS
```

```
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DCLJV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,JOIN
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-SPOOL-PARAMETERS
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ERAJV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-USER-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ERAM
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,PASSWORD
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,STAJV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ERASE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,PRINT
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ADD-MASTER-CATALOG-ENTRY
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,IMPORT-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ADD-PASSWORD
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,LOCK-USER
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,ADD-USER
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-FILE-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,COPY-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-FILE-GENERATION-SUPPORT
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CREATE-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-FILE-GROUP-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CREATE-FILE-GENERATION
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-JV-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CREATE-FILE-GROUP
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-MASTER-CATALOG-ENTRY
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CREATE-JV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,MODIFY-USER-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,CREATE-TAPE-SET
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,PRINT-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,REMOVE-MASTER-CATALOG-ENTRY
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-FILE-GENERATION
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-FILE-GROUP
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,REMOVE-USER
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-JV
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-FILE-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-TAPE-SET
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-JV-ATTRIBUTES
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,DELETE-SYSTEM-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SET-FILE-LINK
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,EXPORT-FILE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SET-JV-LINK
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,REMOVE-JV-LINK
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-JV-LINK
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,EXTEND-TAPE-SET
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,UNLOCK-USER
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,WRITE-SPOOL-TAPE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,REMOVE-PASSWORD
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,TYPE
(C0) %   P-000.125439 NSI4110 SET-CMD-CODE @,SHOW-LIVE-MIGRATION-HISTORY
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=R,C-N=CMD1,A-N=CON1
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=A,C-N=CMD2,A-N=CON2
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=U,C-N=CMD3,A-N=CON3
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=A,C-N=CMD4,A-N=CON4
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=U,C-N=CMD5,A-N=CON5
(C0) %   P-000.125439 NSI4110 ADD-CMD-ENTRY A-C=R,C-N=CMD6,A-N=CON6
(C0) %   P-000.125439 NSI4110 /EOF
(C0) %   E-000.125439 EMM2310 THE SIZE OF MAIN MEMORY IS 00004096 MB;
                      THE MINIMAL SIZE OF MAIN MEMORY IS 00004096 MB
(ZA) % UCO-000.125441  %  NBR0706 TEST OUTPUT FOR CONSOLE 'ZA'
```

```
...
(C0) % UCO-000.125441  %  NBR0706 TEST OUTPUT FOR CONSOLE 'C0'
  <* % UCO-000.125441  %  EXC0903 CONSOLE 'ZA' INOPERABLE
(C0) % UCO-000.125441  %  EXC0652 CONSOLE 'C0' TAKES OVER ALL FUNCTIONS AND AUTHORIZATIONS
OF CONSOLE 'ZA'
...
  <* ? TSC-000.125442 NSI0077 ENTER AUTOMATIC COMMAND FILE NAME.
                      REPLY (FILE NAME; N(DO NOT USE); EOT(USE STANDARD))
   TSC R(C0)-000.125457.
------------------------------------------------------------------------------------- 13.
  <* % UCO-000.125457  %  NBR0792 QUESTION '000' FROM ' TSC' ANSWERED BY '(C0)'. REPLY:
  <* ?   Q-000.125457  %  NSI6005 SYSTEM PARAMETER STUPTYPE = W. SHALL VALUE BE CHANGED?
                         REPLY ( U(NCHANGED),W(ARM),
                         C(OLD), S(ELECTIVE), Z(IP), T(TSN FILE RESET ONLY),
                         J(JOIN AND TSN FILE RESET),EOT=UNCHANGED)
   Q R(C0)-000.125500.C
------------------------------------------------------------------------------------- 14.
  <* % UCO-000.125500  %  NBR0792 QUESTION '000' FROM '   Q' ANSWERED BY '(C0)'. REPLY: C
  <* ?   Q-000.125500  %  NSI6010 SYSTEM PARAMETER RECONUC = N. SHALL VALUE BE CHANGED?
                         REPLY ( U(NCHANGED), N(O), B(ACKUP), T(SOSCAT), A(LL),
                         R(ESET), EOT=UNCHANGED
   Q R(C0)-000.125504. N
------------------------------------------------------------------------------------- 15.
  <* % UCO-000.125504  %  NBR0792 QUESTION '000' FROM '   Q' ANSWERED BY '(C0)'. REPLY:
  <* %   Q-000.125504  %  DMS035B IMCAT TASK 'XABW' FOR PUBSET WITH PUBSET ID 'MH20' CREATED
AND STARTED
...
  <R %DSSM-000.125504  %  ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'CALENDAR/V21.0'
COMPLETELY PROCESSED
...
  <* %   M-000.125505  %  NMH1102 MESSAGE OUTPUT FILE ':MH20:$TSOS.SYSMES.BS2CP.210',
                         ACCESS=DLAM + ISAM, ACTION=STARTUP
  <* %   M-000.125505  %  NMH1102 MESSAGE OUTPUT FILE ':MH20:$TSOS.SYSMES.EKP.01',
                         ACCESS=ISAM, ACTION=STARTUP
  <* %   M-000.125505  %  NMH1112 MESSAGE PROCESSING READY
...
  <* %SERS-000.125515  %  NER1500 SERSLOG LOGGING FILE ':MH20:$TSOS.SYS.SERSLOG.2021-01-
25.025.01' OPENED
  <H % TSC-000.125515  %  HEL0001 HW ERROR LOGGING FILE ':MH20:$TSOS.SYS.HEL.2020-11-
28.154947'
                         OPENED WITH 'SPACE=  402'
  <R % TSC-000.125515  %  EXC0040 LOGGING FILE ':MH20:$SYSAUDIT.SYS.REPLOG.2021-01-
25.025.01' OPENED
...
  <@ %HT1C-000.125516  %  CMD0695 SYSTEM SYNTAX FILE ':MH20:$TSOS.SYSSDF.SDF.210' ACTIVATED
  <@ %HT1C-000.125516  %  CMD0695 SYSTEM SYNTAX FILE ':MH20:$TSOS.SYSSDF.ACS.210' ACTIVATED
...,
  <A % TSC-000.125522  %  NJA0001 D.E.R INITIALIZED SUCCESSFULLY
  <* % MSG-000.125522  %  ETMRK18 CPU 01 ATTACHED
  <* %ACCT-000.125522  %  NAM0001 NEW ACCOUNTING FILE ':MH20:$TSOS.SYS.ACCOUNT.2021-01-
25.025.01'
                         OPENED WITH 'SPACE=(48,48)'
  <* % DRV-000.125522  %  DRV1010 FILE ':MH20:$TSOS.SYSPAR.DRV.032' IS USED AS DRV
CONFIGURATION FILE.
  <* % TSC-000.125522  %  NSI0000  ***  S Y S T E M   R E A D Y  ***
------------------------------------------------------------------------------------- 16.
  <J % TSC-000.125522  %  JMS0066 JOB 'JSTREAM' ACCEPTED ON 21-01-25 AT 12:55, TSN = 59Y2
  <R % TSC-000.125522  %  JMS0300 JOB STREAM '$SYSJS' 'ATTACHED'
  <J %59Y2-000.125522  %  JMS0154 'TSOS' LOGGED ON FOR 'NTL'.
```

```
                          JOB NAME 'JSTREAM'. CALLER 'TSN  TSC'. TID 00020050
  <R %59Y2-000.125522  %  JMS0300 JOB STREAM 'JSTREAM' 'ATTACHED'
...
  <* %SEST-000.125522  %  NBR3000 SYSTEM EVENT STREAM FILE ':MH20:$SYSAUDIT.SYSLOG.ESS.
SYSTEM' OPENED
OPRR /(C0)-000.125522 RUN CMDFILE
------------------------------------------------------------------------------------ 17.
  <* % UCO-000.125522  %  NBR0972 OPERATOR TASK WITH TSN 'XABY' CREATED FOR PROCESSING OF
/RUN COMMAND SEQUENCES
(C0) +RUNT-000.125522  %  NBR1000 COMMAND '/RUN' RECEIVED. RUN ID '0001' ASSIGNED
(C0) +RUNT-000.125522  %  NBR1001 RUN ID '0001'. READING OF /RUN COMMAND FILE ':MH20:$TSOS.
CMDFILE' STARTED
OPRR /(C0)-000.125522 DCSTART DCSOF=SOF.ABGSE211.SE
OPRR /(C0)-000.125522 START-SUBS CRTEBASY
OPRR /(C0)-000.125522 START-SUBS SPOOL
...
```

1. Der DIALOG-Startup wird ohne zusätzliche Option (Antwort: `EOT`) weitergeführt.
2. Der Home-Pubset ist online.
3. Datum und Uhrzeit laut SVP werden angezeigt.
4. Die SYSSTART-Rep-Datei wird verarbeitet.
5. Es soll die Standard-Startup-Parameterdatei verarbeitet werden (Antwort: `EOT`).
6. Es soll keine weitere Parameterdatei verarbeitet werden (Antwort: `END`).
7. Der Dateiname für BS2000 soll eingegeben werden. Mit der Antwort `EOT` wird BS2000 aus der Standarddatei geladen.
8. Mit der Antwort `EOT` wird die Standard-Rep-Datei verarbeitet.
9. Es soll keine weitere Rep-Datei verarbeitet werden (Antwort: `END`).
10. Es werden verschiedene Systeminformationen ausgegeben: Name und Version des Betriebssystems, Zeitpunkt der Generierung, verfügbarer Speicherplatz, Zeitzoneninformationen, TODR-Epoche, Konfigurationsdaten, Größe der Seitenwechseldateien.
11. Der seitenwechselbare Teil des BS2000-Ablaufteils wurde erfolgreich geladen und korrigiert (REP-Verarbeitung). Er ist somit ablaufbereit.
12. Die unter (5) ausgewählten Parametersätze werden in der CONSLOG protokolliert. Die Verarbeitung findet – je nach Parameter-Schlüsselwort – zu unterschiedlichen Zeitpunkten statt.
13. Die Kommandodatei mit dem durch den Systemparameter CMDFILE festgelegten Namen soll nach „System Ready“ automatisch verarbeitet werden (Antwort: `EOT`). Wird mit `N` geantwortet (`DO NOT USE`), kann eine CMDFILE später mit dem Kommando RUN CMDFILE nachträglich gestartet werden.
14. Es soll ein Kaltstart durchgeführt werden (Antwort: `C`).
15. Die Katalog-Rekonfiguration soll wie im Parameterservice definiert erfolgen (Antwort: `N`). Im Beispiel also mit RECONUC=N.
16. „System Ready“, das BS2000-System ist betriebsbereit.
17. Automatisches Starten der Kommandodatei `CMDFILE` (siehe (13)).

2.3 Systemkorrekturen

- Funktion und Aufbau einer Rep-Datei
- Rep-Sätze
- Rep-Zwischenspeicherung
- RMS: Montage und Lieferung von Reps

2.3.1 Funktion und Aufbau einer Rep-Datei

In der Systemeinleitung können zur Korrektur der Ladeobjekte SYSIPL, SYSSTART und SLED, sowie zur Korrektur des Organisationsprogramms Rep-Dateien, bestehend aus Rep-Sätzen, verwendet werden.

Rep-Sätze erlauben eine byteweise Korrektur der oben angeführten Ladeobjekte. Auch Liefereinheiten, die nicht zum Exec gebunden sind, können über Rep-Sätze korrigiert werden. Die entsprechende Rep-Datei ist mit dem Namen SYSREP.<liefereinheit>.<ver> katalogisiert. Ganze Module können nicht über Rep-Sätze ausgetauscht werden, sondern müssen mit LMS in die dazugehörige Bibliothek (OML) eingebracht werden.

Die Rep-Verarbeitung findet in der Regel für das zuletzt geladene Objekt statt. Das Laden des Betriebssystems BS2000 läuft in zwei Schritten ab; die beiden Teile werden einzeln korrigiert.
Klasse-1-Rep-Sätze werden unmittelbar nach dem Laden des residenten Teils des Organisationsprogramms verarbeitet. Sie sollen nur für diejenigen Klasse-1-Module (residenter Teil) des Organisationsprogramms erstellt werden, welche zum Laden und Initialisieren der Klasse-2-Module (nichtresidenter, seitenwechselbarer Teil) des Organisationsprogramms benötigt werden. Die Relativierung der Korrekturdaten in Klasse-1-Rep-Sätzen ist auf Klasse-1-Module und Module der Systemeinleitung beschränkt; Entries können nur zum Modulanfang relativiert werden.
Klasse-2-Rep-Sätze werden unmittelbar nach dem Laden des nichtresidenten Teils des Organisationsprogramms verarbeitet. Mit ihnen können alle Module des Organisationsprogramms korrigiert werden. Die Relativierung von Korrekturdaten ist hier für alle Module und Entries möglich.

Rep-Verarbeitung bei der Systemeinleitung

Die Rep-Verarbeitung wird bei FAST- und AUTOMATIC-Startup mit Ausnahme von Fehlerfällen automatisch durchgeführt, d.h. ohne Dialog mit dem Operator. Beim DIALOG-Startup kann die Rep-Verarbeitung, außer für SYSIPL und SLED, vom Operator beeinflusst werden.

Beim FAST- und AUTOMATIC-Startup werden die zu verarbeitenden Reps in den Dateien $TSOS.SYSREP.BS2.<ver> und $TSOS.SYSREP.STRT.<ver> auf dem Home-Pubset erwartet, sofern über die Startup-Parameterdatei keine anderen Rep-Dateien für das Organisationsprogramm angegeben werden (Parameter REPFILEx).
Ein Dialog mit dem Operator wird nur in Fehlerfällen aufgenommen.

Die Standardnamen der Rep-Dateien lauten:

- für BS2000: SYSREP.BS2.<ver>
- für SYSIPL: SYSREP.IPL.<ver> (kopiert in Datei SYSREP.IPL.DSKnnn)
- für SYSSTART: SYSREP.STRT.<ver>
- für SLED: SYSREP.SLED.<ver> (kopiert in Datei SYSREP.SLED.DSKnnn)

(Diese Dateien (SYSREP.IPL.<ver> und SYSREP.SLED.<ver>) werden durch SIR kopiert und dann im SVL verankert. Sie können nicht über Dateinamen angesprochen werden (siehe unten).)

Beim DIALOG-Startup können Rep-Dateien auf gemeinschaftlicher Platte oder Privatplatte vorliegen oder über Konsole eingegeben werden.

Die Reihenfolge der Verarbeitung wird vom Operator festgelegt, indem er auf die Meldung `NSI0050` jeweils ein Eingabegerät angibt. Die Reps werden sofort verarbeitet. Danach erscheint wieder die Meldung `NSI0050`. Dieser Vorgang wiederholt sich solange, bis der Operator als Antwort `P.END` (bzw. `P.`, wenn bereits eine Repdatei mit Standardnamen verarbeitet wurde) eingibt (siehe folgende Seite).

Eine Plattendatei kann viermal und die Konsole zweimal als Eingabe angegeben werden. Es wird überprüft, ob die angegebene Datei bereits einmal verarbeitet wurde.
Diese Einschränkungen gelten nur für die BS2000-Rep-Dateien; für alle anderen Objekte gibt es diese Einschränkungen nicht.

Der Dialog an der Konsole für Klasse-1-Rep-Sätze und Klasse-2-Rep-Sätze wird separat eröffnet. Die eingegebenen Daten werden genauso behandelt, als wenn sie über eine Rep-Datei eingegeben werden.
Sie werden also auch in die Sicherstellungsdatei SYS.NSI.SAVEREP und später in die Datei REPLOG ($SYSAUDIT.SYS.REPLOG.<datum>.<session-nr>.01) geschrieben.

> *Rep-Verarbeitung für SYSIPL und SLED*
>
> Reps für SYSIPL und SLED befinden sich jeweils in einer Rep-Datei, die von SIR im SVL verankert wird. Der Operator hat während der Systemeinleitung bzw. der Erstellung von Diagnoseinformationen keine Möglichkeit, eine andere Rep-Datei auszuwählen.
> Aus demselben Grund hat auch die Art der Systemeinleitung keinen Einfluss auf diesen Teil der Rep-Verarbeitung.
> Bei einer Änderung dieser Rep-Dateien muss erneut die Verankerung im SVL mit SIR (Funktion CREATE-IPL oder MODIFY-IPL) erfolgen.

Aufbau einer Rep-Datei

Eine Rep-Datei für die Systemeinleitung hat folgenden Aufbau:

BS2000_LOADER [_kommentar]	1. Satz (obligatorisch)
Klasse-1-Rep-Sätze	Rep-Sätze für ausgewählte Module des residenten Teils des Organisationsprogramms und der Systemeinleitung (wahlweise)
_END[_kommentar]	Ende-Anweisung für Klasse-1-Rep-Sätze (obligatorisch)
Klasse-2-Rep-Sätze	Rep-Sätze für das gesamte Organisationsprogramm (wahlweise)
_END[_kommentar]	Ende-Anweisung für Klasse-2-Rep-Sätze
/[_kommentar]	Dateiende-Kennung
*%kommentar oder *%%kommentar	Kommentarsatz wird an der Konsole ausgegeben (gilt nicht für Kommentare in SYSREP.IPL.vvv)

Dieser Aufbau gilt für SYSIPL, SLED, SYSSTART und BS2000 gleichermaßen. Die Unterscheidung zwischen Klasse-1-Rep-Verarbeitung und Klasse-2-Rep-Verarbeitung ist nur für BS2000 relevant.

Als Endekriterium für Klasse-2-Repsätze und die Rep-Datei muss entweder eine zweite END-Anweisung oder eine Dateiendekennung existieren.

Auf Platte können die Rep-Sätze 1-256 Byte lang sein, wobei die Zeichen nach dem 80. Byte nicht verarbeitet werden.
Die Rep-Datei ist eine SAM-Datei mit variablen Sätzen und Standardblockung BUFFER-LENGTH=STD(SIZE=1) oder (SIZE=2). Der Dateiname kann beliebig gewählt werden. Änderungen in der Rep-Datei auf Platte sollten nur mit RMS (siehe Abschnitt „RMS: Montage und Lieferung von Reps") durchgeführt werden.

Die Rep-Dateien werden in der Reihenfolge gelesen und verarbeitet, die von der Systembetreuung festgelegt wird.

Die Konsole kann zweimal als Rep-Eingabegerät angegeben werden. Wurden evtl. fehlerhafte Platten-Reps eingelesen, so können diese zum Schluss noch einmal über Konsole korrigiert werden.
Kommentarsätze (mit * in Spalte 1) können nach dem Satz „BS2000 LOADER“ an beliebiger Stelle der Rep-Datei eingefügt werden. Diese werden vom Startup nicht ausgewertet. Kommentarsätze, die in der 2. Spalte das %-Zeichen enthalten, werden über die Konsole protokolliert (gilt nicht für Kommentare in SYSREP.IPL.vvv).

Rep-Sätze für Module, die nicht in das Organisationsprogramm eingebunden sind, deren Namen dem Organisationsprogramm jedoch bekannt sind, werden ohne Fehlermeldung übergangen. Damit ist es möglich, Rep-Sätze für alle Module einer BS2000-Version in eine Rep-Datei zu integrieren.

Rep-Sätze, die einen ungültigen Modulnamen enthalten, werden als fehlerhaft protokolliert. Enthält der Rep-Satz jedoch in Spalte 69 ein „S“ oder „U“, dann wird die Fehlermeldung unterdrückt. Damit können Rep-Sätze für Module in die Rep-Datei aufgenommen werden, die dem Organisationsprogramm (noch) unbekannt sind (z.B. Selectable Units, "Funktion und Aufbau einer Rep-Datei").

2.3.2 Rep-Sätze

Die Adresse der zu korrigierenden Daten ist stets relativ zum Modulanfang anzugeben. Jeder Rep-Satz wird vor der Verarbeitung auf richtiges Format untersucht. Prüfdaten, Kontrollzahl und Modulversion werden nur geprüft, wenn sie angegeben sind, d.h. zu Testzwecken können sie auch weggelassen werden. Fehlerhafte Sätze werden zusammen mit einer Fehlermeldung an der Konsole protokolliert, eine Korrektur wird für solche Sätze nicht ausgeführt.

Korrekturdaten, die von der Adresse eines anderen Moduls abhängen, können in der Form „Basis + Distanz“ angegeben werden. Diese Art Reps werden „relative Reps“ genannt. Die Distanz ist relativ zum Modulanfang/Entry /ISL-Entry anzugeben und die Basis in Gestalt des Modul-/Entry-/ISL-Entry-Namens. Damit sind Rep-Sätze invariant gegenüber Änderungen in allen übrigen Modulen des Objektes.

In der folgenden Tabelle steht:

- **a** für ein alphanumerisches Zeichen (0-9, A-Z)
- **x** für ein Sedezimalzeichen (0-9, A-F)

Spalte	Inhalt	Bedeutung
1	_	Leerzeichen
2 - 4	REP	
5	_	Leerzeichen
6 -10	xxxxx	Rep-Adresse, relativ zum Modulanfang
11	_	Leerzeichen
12 -14	3-stellige Zahl	Laufende Nummer der Objektkorrekturversion
15	_	Leerzeichen
16	X, I, O, S, P, T	X: Standard- oder relativer Rep I: relativer Rep (ISL-Entry) O, S, P, T: relativer Rep für x86-64-Code
17 - max. 50	' xxx...x' oder ' xx...x' + Name	max.32 Korrekturdaten, in Anführungszeichen eingeschlossen, oder max.22 Korrekturdaten, in Anführungszeichen eingeschlossen, gefolgt von einem Pluszeichen und dem Namen eines Moduls/Entries/ISL-Entries. Die Adresse wird von der Systemeinleitung zu den letzten 8 Stellen der Korrekturdaten addiert. Der Name muss (wie in Spalte 73-80) 8-stellig angegeben werden. x86-64-Reps haben ein gesondertes Rep-Format (siehe "Rep-Sätze")
51	_	Leerzeichen
52 - 55	xx__ oder xxxx	2 oder 4 Prüfdaten; es sind das erste oder die ersten beiden Byte anzugeben, die von den Korrekturdaten überschrieben werden sollen.

56	_	Leerzeichen
57	x	Kontrollzahl (parity digit) für Rep-Adresse, Korrekturdaten und Prüfdaten
58 - 65	aaaaaaaa	Nummer der Problemmeldung
66 - 68	aaa	Modulversion
69	a	Rep-Kennzeichen
70	1 oder 2	1 für Klasse-1-Rep-Sätze, 2 für Klasse-2-Rep-Sätze
71	a oder Leerzeichen	Laderversion (A-Z)
72	a	Kennzeichen für die Auswahl von Rep-Sätzen für unterschiedliche Code-Varianten. 'K' kennzeichnet Korrekturen von x86-64-Code. Ein Leerzeichen kennzeichnet Korrekturen für /390-Code.
73 - 80	aaaaaaaa	Modulname, 8-stellig; die Anfangsadresse dieses Moduls wird zu der Rep-Adresse addiert.

Tabelle 2: Format der Rep-Sätze

Hinweise zum Format der Rep-Sätze

- Klasse-1-Rep-Sätze und Klasse-2-Rep-Sätze besitzen dasselbe Format und erlauben dieselben Korrekturfunktionen. Sie unterscheiden sich lediglich durch die Klassenangabe in Spalte 70 und ihre Verwendung sowie die Möglichkeit zur Angabe von Entries/ISL-Entries bei „relativen Reps“.
- Die Adresse der zu korrigierenden Daten wird stets durch Addition der Moduladresse (Name in Spalte 73ff) und der Rep-Adresse gebildet.
- Um bei relativen Reps Unterscheidungsmöglichkeiten zu haben, werden zusätzlich zum Format mit dem Indikator `X` die Indikatoren `I, O, S, P,` und `T` angeboten. Die Rep-Daten werden im Anschluss an den Indikator in einschließenden Hochkommas angegeben.
  - Indikator `X` bezeichnet reale Module/Entries. Die Indikatoren `P` und `T` kennzeichnen spezielle relative Reps für x86-64-Code. Dazu gibt es ein spezielles Format:

    Format: X'distanz'+basisadresse
    Hier wird für die Basisadresse die Adresse des realen Entrys vermerkt.

  - Indikator `I` zeigt an, dass der angegebene Entry-Name den ISL-Entry bezeichnet.

    Beispiel: I'distanz'+<name-des-isl-entrys>

  - Die Indikatoren `O` und `S` kennzeichnen spezielle relative Reps für x86-64-Code.
    Dazu gibt es ein spezielles Format:

    Beispiel: `O'<code>','<distanz>'+<entry/module-name>`

    Aus <code> und berechneter Adresse wird die Korrekturinformation aufgebaut. <code> und <distanz> sind jeweils genau 8 Zeichen lang.

- Da die Korrekturdaten in hexadezimaler Schreibweise angegeben werden, muss deren Anzahl stets gerade sein.
- Bei zu relativierenden Korrekturdaten müssen mind. 8 Korrekturdaten vorhanden sein.

- Falls die zu korrigierenden Daten eine Adresse darstellen, ist der alte Wert generierungsabhängig, d.h. es sollten keine Prüfdaten angegeben werden.
- Die Kontrollzahl (Spalte 57) dient zur Sicherung des Rep-Satz-Inhalts. Sie ergibt sich aus der Summe aller Stellen von Rep-Adresse, Korrekturdaten und Prüfdaten
- Zur Summe ist die Anzahl der Korrekturdaten zu addieren. Das Ergebnis mod(16) ergibt für die Kontrollzahl einen Wert zwischen 0 und F.
- mod(16) bedeutet: Die Summe ist durch 16 zu dividieren. Der verbleibende Rest ist die gesuchte Kontrollzahl. Die Kontrollzahl wird mit dem Dienstprogramm RMS (Rep Montage System) automatisch erzeugt, wenn sie nicht vorhanden ist.
- Der Inhalt der Spalten 58-65 dient organisatorischen Zwecken und wird von der Systemeinleitung nicht ausgewertet.
- Wahlweise können Prüfdaten, Kontrollzahl und Modulversion weggelassen werden. Bei Startup entfallen dann die entsprechenden Prüfungen.
- Folgende Rep-Kennzeichen in Spalte 69 sind definiert:

  `'D'` = Diagnose-/Fang-Rep
  `'O'` = Optionaler Rep
  `'Q'` = Selectable Unit, Diagnose-/Fang-Rep
  `'S'` = Selectable Unit
  `'T'` = Trace (einschalten)
  `'U'` = Selectable Unit, optional
  `'V'` = Vorläufiger Rep
  `' '` = Normaler Rep
- Das Kennzeichen in der Spalte 72 steuert die Auswahl der Rep-Sätze für unterschiedliche Code-Varianten (/390- oder x86-64-Code). Bei Kennzeichnung mit 'K' wird die Korrektur nur für x86-64-Code berücksichtigt. Enthält die Spalte 72 ein Leerzeichen, gilt die Korrektur für /390-Code. Alle anderen Inhalte werden ignoriert. Entsprechend dem Ablauf auf den Server Units werden die Reps für das jeweils andere HSI ignoriert.
- Anstelle eines Modulnamens kann bei Klasse-2-Rep-Sätzen in den Spalten 18-49 auch ein ENTRY-Name stehen.
- Anstelle eines Modulnamens kann in den Spalten 73-80 auch ein ENTRY- oder CSECT-Name stehen. In diesen Fällen darf dann keine Modulversion angegeben sein (nur für Klasse-2-Rep-Sätze).

Eingabe von Rep-Sätzen über Konsole

Während der Ladeprozedur erscheint folgende Meldung:

```
?P.NSI0050 SPECIFY BS2000 REP FILE OR DEVICE. REPLY (EOT (USE STANDARD
           FILE); FN=FILENAME(VOL=VSN); CONS; END)
```

Antwortet der Operator mit `P.CONS` beginnt der Dialog für die Klasse-1-Rep-Sätze:

Meldung:

```
?P.NSI0070 ENTER CLASS 1 MODULE NAME. REPLY (NAME; EOT; (NO MORE CLASS 1
           CONSOLE REPS))
```

Antwort: `P.<modulname>`
(Name des Moduls, das geändert werden soll)

Meldung:

```
?P.NSI0071 ENTER RELATIVE ADDRESS IN MODULE. REPLY (5 CHAR)
```

Antwort: `P.<korrekturadresse>`
(Modul-relative Adresse eines Reps; 5 sedezimale Ziffern)

Meldung:

```
?P.NSI0073 ENTER CORRECTION DATA. REPLY (MAX 32 CHARACTERS)
```

Antwort: `P.<änderungsdaten>`
(Daten zur Korrektur eines Modulobjektcodes, 2-32 sedezimale Ziffern)

Die Eingabe von relativen Reps ist erweitert:
Neben der Eingabe von distanz+basisadresse ist jetzt auch `X'distanz'+basisadresse` und `I'distanz'+<isl-entry-adresse>` sowie die Eingabe der entsprechenden relativen Repformate für x86-64-Code möglich.
`basisadresse` ist die Adresse eines realen Entrys oder Moduls.

Meldung:

```
?P.NSI0074 ENTER CHECK INFORMATION. REPLY (CCCC,P,MMM (OLD DATA, PARITY,
           VERSION); EOT(SKIP CHECK))
```

mögliche Antworten:

`CCCC`	1 oder 2 Prüfbytes
`P`	Kontrollzahl oder leer
`MMM`	Modulversionsnummer oder leer

Für alle eingegebenen Daten findet eine Prüfung statt, bei fehlenden Daten wird die entsprechende Prüfung nicht durchgeführt.

oder Antwort: `P.` (und ENTER-Taste drücken)
Bei dieser Antwort findet keine Prüfung statt.

Danach kommt wieder die Meldung `NSI0070` und fordert die Eingabe des nächsten Rep-Satzes an. Der gleiche Dialog wiederholt sich solange, bis keine Klasse-1-Rep-Sätze mehr eingegeben werden sollen. In dem Fall muss die Meldung `NSI0070` mit `P.` und dem Drücken der ENTER-Taste beantwortet werden.

Nach weiteren informativen Meldungen beginnt der Dialog für Klasse-2-Rep-Sätze. Er wird eingeleitet mit der Meldung

```
?P.NSI0075 CONSOLE ASSIGNED AS REP-LOADER FOR CLASS2 MODULES.
           REPLY (EOT(CONSOLE IS USED); N(NO)).
```

Die Antwort darauf ist `P.` oder `P.N` (keine Klasse-2-Rep-Sätze) und ENTER-Taste drücken.

2.3.3 Rep-Zwischenspeicherung

Alle Rep-Sätze werden in der Datei SYS.NSI.SAVEREP zwischengespeichert und dann in der REPLOG-Datei protokolliert. Falls keine aktuelle REPLOG-Datei (siehe "Rep-Zwischenspeicherung") vorhanden ist, wird die SAVEREP-Datei von SLED sichergestellt und kann wahlweise mit DAMP aufbereitet werden.

SAVEREP (Systemeinleitung für BS2000)

Jeder Rep-Satz, der das System tatsächlich modifiziert hat, wird mit einer Ziffer in der Spalte 72 gekennzeichnet, die anderen Rep-Sätze erhalten einen Buchstaben.
Dabei bedeutet:

1 oder A: Rep von Platte

4 oder D : Rep von Konsole

Der Lader-Code wird in den Satz BS2000 LOADER geschrieben. Der Lader-Code wird als Funktion aller Rep-Sätze (ohne SYSIPL-Reps) erzeugt und dient zur Identifizierung der Rep-Datei. Er ergibt sich aus der Summe der Rep-Adressen und der Kontrollzahlen aller existierenden Rep-Sätze.

Für jedes Rep-Medium werden bei der Sicherstellung je zwei START- und END-Sätze eingetragen, die Datum, Zeit, Komponente und den vollständigen Dateinamen zur eindeutigen Identifizierung enthalten.
Reps für ein anderes als das aktuelle HSI werden nicht in der SAVEREP protokolliert (und damit auch nicht in der REPLOG) und auch bei der Meldung zur Verarbeitung der IPL-Reps nicht mitgezählt.

Diese Sätze werden in folgender Form eingetragen:

```
START <objekt> yyyy-mm-tt,hh:mm:ss
START <datei oder eingabegerät>
:
: <Rep- und Kommentarsätze>
:
END <objekt> yyyy-mm-tt,hh:mm:ss
END <datei oder eingabegerät>
```

Für <objekt> können folgende Werte stehen:

SYSIPL: Reps für SYSIPL
SYSSTART: Reps für SYSSTART
BS2000-CL1: Reps für Klasse-1-Exec
BS2000-CL2: Reps für Klasse-2-Exec (ohne Datum und Uhrzeit)
VM2000: Reps für VM2000-Hypervisor (SU /390)

<datei> enthält den vollständigen Dateinamen der Rep-Datei aus den Teilen Katalogkennung, Benutzerkennung und Dateiname.

Für <eingabegerät> können folgende Werte stehen:

IPL-Pfad: für SYSIPL-Reps

*CONSOLE: für Reps, die von der Konsole eingegeben werden

Nach dem letzten END-Satz für BS2000-Klasse-2-Reps wird ein zusätzlicher Satz (END BS2000-Satz) hinterlegt, der Informationen über die Anzahl der verarbeiteten und ersetzten Reps enthält:

```
END BS2000: LOADER VERSION Z, mmmmm PROCESSED REPS, nnnnn REPLACED REPS
```

Die Datei SYS.NSI.SAVEREP wird im weiteren Verlauf der Systemeinleitung in die Logging-Datei $SYSAUDIT.SYS.REPLOG.<datum>.<session-nr>.01 kopiert. Dort werden die Korrekturdaten für BS2000 und alle nachgeladenen Systemteile (Subsysteme) protokolliert.

REPLOG

Die Korrekturdaten für SYSIPL, SYSSTART, BS2000 und alle nachgeladenen Subsysteme sowie alle mit ROSI im laufenden System durchgeführten Korrekturen werden in der Loggingdatei $SYSAUDIT.SYS.REPLOG.<datum>.<session-nr.>.01 protokolliert.

> ℹ Zusätzlich wird, neben den tatsächlich durchgeführten Korrekturen von BS2000 und den nachgeladenen Systemteilen/Subsystemen, die alte Information, welche durch die Korrektur ersetzt wird (Rep before image), im Speicher protokolliert. Dadurch können Korrekturen zuverlässig und konsistent erfasst und aus dem laufenden System wieder entfernt werden.

Mit dem Systemparameter SECSTART kann die Systembetreuung festlegen, ob der Korrekturstand über mehrere Systemläufe hinweg protokolliert werden soll.
Bei SECSTART=N werden alle evtl. existierenden REPLOG-Dateien gelöscht.
Bei SECSTART=Y bleiben die REPLOG-Dateien erhalten; die Korrekturen können über mehrere Systemläufe hinweg verfolgt werden.

Mit dem Kommando SET-REPLOG-READ-MARK kann die Systembetreuung die REPLOG-Datei des aktuellen Systemlaufs kurzzeitig schließen. Alle bis zu diesem Zeitpunkt protokollierten Daten können so kopiert und analysiert werden.

Kommando	Bedeutung
SET-REPLOG-READ-MARK	Lesenden Zugriff auf die REPLOG-Datei ermöglichen

2.3.4 RMS: Montage und Lieferung von Reps

Das Dienstprogramm RMS (Rep Montage System) ist ein dialogorientiertes Programm für die Montage und Lieferung von Rep-Mengen.

Das Programm verwaltet in einer zentralen Datei komprimiert alle Korrekturen und Beschreibungen, deren Herkunft und Produktzugehörigkeit sowie Umfang und Zeitpunkt aller Aktionen. Es stehen somit alle für die Lieferung und den Einsatz der Korrekturen relevanten Informationen zur Verfügung und jede Aktion ist für die Fehlerdiagnose beliebig reproduzierbar.

RMS ist im Handbuch „Dienstprogramme“ [15] beschrieben.

2.4 Systembeendigung

Es gibt zweierlei Arten der Systembeendigung:

- Planmäßige bzw. normale Beendigungen sind alle Beendigungen, die durch das SHUTDOWN-Kommando eingeleitet werden.
- Außerplanmäßige bzw. abnormale Beendigungen können als Folge von Software- oder Hardwarefehlern auftreten.

Kommando	**Bedeutung**
BCEND	DCM im Server beenden
INFORM-ALL-JOBS	Nachricht an alle Benutzertasks senden
SET-RESTART-OPTIONS	automatischen Restart steuern
SHOW-RESTART-OPTIONS	Informationen über automatischen Restart anfordern
SHUTDOWN	Systemlauf beenden
UNLOCK-DISK	Systembelegungsprotokoll bereinigen

Tabelle 3: Kommandoübersicht zur Systembeendigung

2.4.1 Planmäßige Beendigung

Bevor der Operator das System endgültig beendigt, sollte er den Dialogteilnehmern mit dem Kommando SHUTDOWN MODE=QUIET eine Warnung zukommen lassen, um ihnen die Gelegenheit zu geben, ihre Tasks ordnungsgemäß abzuschließen.

Dabei wird zum einen ein INFORM-ALL-JOBS-Kommando simuliert zur Benachrichtigung des Teilnehmerbetriebes, zum anderen wird ein BCEND W=Y,TERM=N abgesetzt, der den Teilhaberanwendungen eine Warnung über den bevorstehenden BCEND zukommen lässt. Ab diesem Zeitpunkt werden keine weiteren Benutzer zum System zugelassen und keine neuen Jobs (z.B. mit ENTER-JOB) gestartet.

Die endgültige Systembeendigung leitet der Operator mit dem Kommando SHUTDOWN MODE=END ein. Alle noch laufenden Tasks werden abgebrochen, und das Kommando BCEND W=N wird simuliert. SPOOL- und Job-Einträge werden jedoch sichergestellt und, je nach Art des Systemstarts, im nächsten Systemlauf evtl. wieder abgearbeitet. Alle Privatplattenbelegungen werden abgebaut.

Es gibt die Möglichkeit, mit dem Kommando SHUTDOWN MODE=END(RESTART=*YES) einen automatischen Restart nach Systembeendigung einzuleiten.
Mit SHUTDOWN MODE=END(RESTART=*YES(DELAY=...)) kann dieser Restart zeitverzögert eingeleitet werden. Beide Erweiterungen sind bei der Fernbedienung des Systems von Relevanz.

Mit dem Kommando SHUTDOWN MODE=END(RESTART=*YES(IPL-DEVICE=...)) kann ein automatischer Restart mit Wechsel der IPL-Platte durchgeführt werden.
Mit dem Wechsel der IPL-Platte ist auch der Restart eines anderen Systems (im Modus AUTOMATIC) möglich.

Gastsystem-Beendigung im VM2000-Betrieb

Im VM2000-Betrieb kann ein Gastsystem auch über den SE Manager mit der Aktion „BS2000 Shutdown“ oder über das VM2000-Kommando SHUTDOWN-VM beendet werden.

Auch das Monitorsystem kann über den SE Manager mit der Aktion „BS2000 Shutdown für Monitor-VM“ oder über das VM2000-Kommando SHUTDOWN-VM beendet werden.

Für ausführliche Informationen siehe Handbuch „VM2000“ [60].

Systembeendigung über den SE Manager (Hauptmenü Systeme)

Der Shutdown von BS2000 kann auch über den SE Manager im Hauptmenü *Systeme* (Server Unit ausgewählt) in der Registerkarte *Bedienung* mit der Aktion „BS2000 Shutdown“ eingeleitet werden. Dieser Shutdown wird immer ohne Zeitbegrenzung eingeleitet. Die Shutdown-Anforderung wird in einer dynamisch kreierten System-Task ausgeführt und an der Konsole mit der Meldung `NRT1201` protokolliert.

Systembeendigung über den SE Manager (Hauptmenü Hardware)

Der Shutdown von BS2000 kann auch über den SE Manager im Hauptmenü *Hardware* (*Server* ausgewählt) in der Registerkarte *Units* mit dem Aktion-Icon „Ausschalten“ eingeleitet werden. Dieser Shutdown wird auf SU /390 ohne Zeitbegrenzung eingeleitet. Einem BS2000-System auf SU x86 bleibt die ggf. im SE Manager eingestellte Restlaufzeit für den ordnungsgemäßen Shutdown.

Die Shutdown-Anforderung wird in einer dynamisch kreierten System-Task ausgeführt. Sie wird bei eingestellter Restlaufzeit an der Konsole protokolliert mit `NRT1200 SHUTDOWN WITH RUNTIME LIMIT REQUESTED BY 'X2000'. REMAINING RUNTIME: '<HH>:<MM>'`. Wenn keine Restlaufzeit eingestellt ist, dann wird, wie oben, die Meldung `NRT1201` ausgegeben.

Die Restlaufzeit wird in der Sonder-Jobvariablen $SYSJV.REMAINING-BS2000-RUNTIME abgelegt und mit der ablaufenden Zeit heruntergezählt. Die Jobvariable besteht aus fünf Zeichen mit den möglichen Werten:

- UNDEF keine Shutdown-Anforderung durch X2000
- UMLIM keine Zeitbegrenzung
- <hh>:<mm>
  verbleibende Zeit bis zum abrupten Anhalten von BS2000 durch X2000

Über den Systemparameter SHUTPROC kann zusätzlich gesteuert werden, ob BS2000 nach einer Anforderung durch den SE Manager sofort heruntergefahren oder ob zur Systembeendigung ein Enter-Job gestartet werden soll.

Das sofortige Herunterfahren erfolgt analog dem Kommando
SHUTDOWN MODE=*END(RESTART=*NO).

Der Enter-Job wird ggf. in der Datei $TSOS.SYSENT.SHUTDOWN erwartet. Er wird unter der Benutzerkennung, die beim Kommando SET-LOGON-PARAMETERS des Enter-Jobs angegeben ist, gestartet. Die letzte Aktion der kundenspezifischen Vorbereitungen muss das Kommando SHUTDOWN MODE=*NO/*END(RESTART=*NO) sein. Zur Ausführung des Kommandos SHUTDOWN wird das Privileg OPERATING benötigt.

Eine Muster-Kommandodatei für den Enter-Job wird als Release-Item SYSENT.SHUTDOWN.TEMPLATE der Release-Unit BS2CP geliefert und mit IMON installiert. Die Systembetreuung muss diese Datei an die Systemumgebung und an die kundenspezifischen Beendigungsschritte anpassen und unter dem Namen $TSOS.SYSENT.SHUTDOWN bereitstellen.

Für weitere Informationen siehe Handbuch „Bedienen und Verwalten“ [57].

2.4.2 Außerplanmäßige Beendigung

Zur abnormaler Beendigung des Systemlaufs kann es z.B. aus folgenden Gründen kommen:

- Hardware-Fehler an peripheren Geräten oder am Server
- Software-Fehler durch Sättigungs- oder Deadlocksituationen
- Zustände, in denen das System selbst erkennt, dass eine sinnvolle Weiterarbeit nicht möglich ist

In diesen Fällen kann das System nicht ordnungsgemäß beendet werden, d.h. alle Jobs, die zu diesem Zeitpunkt tätig waren, müssen i.A. nach Wiederherstellung des Systems wiederholt werden.

In den meisten Fällen ist es notwendig, den Systemzustand zum Fehlerzeitpunkt mit dem Dienstprogramm SLED sicherzustellen, um eine spätere Diagnose zu ermöglichen.

Die Ausgabedateien für den SLED (siehe „Diagnosehandbuch“ [14]) dürfen auch außerhalb des Home-Pubsets liegen, jedoch nur auf Platten bzw. Pubsets, die als IPL-Platte bzw. Home-Pubset geeignet wären, also z.B. nicht auf SM-Pubsets.

Es ist möglich, dass es sich um eine Störung handelt, die nicht durch ein unmittelbares Neuladen des Systems zu beheben ist, z.B. wenn der Katalog oder andere wichtige Platteninhalte zerstört sind. In diesem Fall müssen die Platten wiederhergestellt werden (letzter Sicherungsstand, Kapitel „Datensicherung").

Belegungen von Privatplatten können nicht mehr zurückgegeben werden, d.h. das System bleibt im SVL der Platte eingetragen und muss ggf. mit dem Kommando UNLOCK-DISK ausgetragen werden.

Belegungen von Pubsets können ebenfalls nicht mehr aufgehoben werden. Bei der nächsten Systemeinleitung weist die Meldung `NSI424A` auf diese Situation hin. Mit der Antwort `ACCEPT` werden solche Pubsets wieder freigegeben.
Während der BS2000-Session kann der Operator mit dem Kommando UNLOCK-DISK noch existierende Dateilocks für einzelne Platten oder ganze Pubsets zurücksetzen.

2.4.3 Automatischer Restart

Die Systemfunktion „Automatischer Restart“ unterstützt den operatorlosen Betrieb. Sie ermöglicht, dass nach einem Systemabbruch das Betriebssystem automatisch neu geladen wird. Ein evtl. notwendiger Systemspeicherauszug mit SLED oder SNAP wird ohne Operatoreingriffe erstellt.

Der automatische Restart kann durch den Operator an- oder abgeschaltet werden (siehe Kommandos SET-RESTART-OPTIONS MODE=*ON/*OFF, DUMP=*SLED/*SNAP und SHOW-RESTART-OPTIONS). Standardmäßig ist er abgeschaltet.

Voraussetzungen für den automatischen Restart mit SLED

Die Voraussetzungen für einen FAST-Startup müssen erfüllt sein und es dürfen keine fehlerhaften Reps und Parameter vorliegen.

Bei einem Systemspeicherauszug mit SLED muss eine der beiden Dateien zur Verfügung stehen:

- $TSOS.SLEDFILE (bei SLED ohne Parameterdatei)
- $TSOS.SYSPAR.SLED.<ver> (bei SLED mit Parameterdatei)

Ist für den Fall des SLED ohne Parameterdatei in der Datei $TSOS.SLEDFILE bereits ein Speicherauszug enthalten (nicht logisch leer), kann kein neuer Speicherauszug erstellt werden und das System wird entsprechend den Vorgaben aus dem Kommando SET-RESTART-OPTIONS sofort neu geladen.

Die Dump-Datei $TSOS.SLEDFILE kann auch eine „große“ Datei mit mehr als 32 GB, auch auf dem Home-Pubset sein.

Voraussetzungen für den automatischen Restart mit SNAP

Die SNAP-Funktion muss eingeschaltet sein. Dies erfolgt über den Parameter SNAP-ACTIVE-SWITCH (Parametersatz SNAP) im Startup-Parameterservice (siehe "Snapshot-Initialisierung (SNAP)") oder im laufenden System durch das Kommando ACTIVATE-SNAPSHOT.

Wenn die Systemdatei $TSOS.SNAPFILE nicht vorhanden ist, dann wird sie beim Startup (Parameter SNAP-ACTIVE-SWITCH=ON) in Standardgröße bzw. bei Ausführung des Kommandos ACTIVATE-SNAPSHOT in der angegebenen Größe angelegt.

Informationen über den Status von SNAP liefert das Kommando SHOW-SNAPSHOT-STATUS.

3 Parameterservice

Über eine oder mehrere Startup-Parameterdateien werden verschiedene Software-Komponenten während des Startups mit Daten versorgt.

Es werden folgende Parameter eingelesen und an die betroffenen Routinen weitergeleitet. In Klammern sind jeweils die „Schlüsselwörter“ innerhalb der Parameter-Anweisung zur Identifizierung der Software-Komponenten aufgeführt.

- Abrechnungssystem (ACCOUNT, "Starten des Abrechnungssystems (ACCOUNT)")
- Inbetriebnahme der dynamischen Subsystemverwaltung (DSSM, "Inbetriebnahme der dynamischen Subsystemverwaltung (DSSM)")
- Bestimmung der Systemzeit und der universellen Weltzeit (GTIME, "Steuerung der Systemzeit (GTIME)")
  Die Angabe der GTIME-Parameter ist notwendig. Ohne vorgegebene GTIME-Parameter wird bei jeder Systemeinleitung eine Frage an den Operator ausgegeben, die dieser beantworten muss.
- Änderung des Konfigurationszustandes (IOCONF, "Parametersatz IOCONF")
- Voreinstellungen für NK-ISAM (ISAM, "Voreinstellungen für NK-ISAM (ISAM)")
- Festlegung des TSN-Modus (JMS, "Festlegung des TSN-Modus (JMS)")
- Verwaltung des Speichers (MEMORY, "Speicherverwaltung (MEMORY)")
- Festlegung der Konsolkonfiguration, Änderung der Konsoleigenschaften, Zuordnung von Berechtigungsschlüsseln und Filterstufen und Unterdrückung bestimmter Meldungen an Konsolen (OPR, "Konfiguration und Meldungsunterdrückung an Konsolen (OPR)")
- Zuweisung des Seitenwechselbereichs bei Startup (PAGING, "Auswahl des Seitenwechselbereichs beim Startup (PAGING)")
- Snapshot-Initialisierung (SNAP, "Snapshot-Initialisierung (SNAP)")
- Voreinstellung von Systemparametern (SYSOPT-CLASS2, "Voreinstellung der Systemparameter (SYSOPT-CLASS2)")
- Änderung von Voreinstellungen der Systemeinleitung (SYSOPT-IPL, "Änderung von IPL-Optionen (SYSOPT-IPL)")
- Festlegung der Parameter für das Datenkommunikationssystem (BCAM)
- Angabe des Sicherheitsbeauftragten bei Einsatz von SECOS (SRPM)
- Konfigurationsdaten für VM2000 (VM2000)

Die Parametersätze werden im Folgenden in alphabetischer Reihenfolge beschrieben.Ausnahmen: BCAM und VM2000 werden in den jeweiligen Produkthandbüchern erläutert und SRPM im SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46].

3.1 Auswahl der Parameterdatei

Automatische Auswahl der Parameterdatei

Wenn ein System auf wechselnden Server Units betrieben werden soll, kann je nach Server Unit eine andere Parameterdatei verwendet werden.

Eine automatische Wahl dieser Parameterdatei erfolgt durch BS2000 im **FAST**- und **AUTOMATIC**-Startup sowie im **DIALOG**-Startup bei Auswahl der Standard-Paramterdatei (Eingabe von `P.`) in folgender Reihenfolge:

- im Native-Betrieb und im VM2000-Monitorsystem:
  1. Die Datei `$TSOS.SYSPAR.BS2.<ver>.<name>` (für SU /390).
     `<name>` ist dabei der Wert des Operanden `NAME` aus der GEN-Anweisung von IOGEN, der in die ersten acht Zeichen des IOCF-Kommentarfeldes des aktiven IOCF eingetragen ist.
  2. Die Datei `$TSOS.SYSPAR.BS2.<ver>.<system-name>` (für SU x86).
     `<system-name>` ist dabei der Systemname, der in der BS2000-Konfiguration vergeben wurde (Auto-IPL-Option); er kann im IPL-Menü (Dialogfenster des SE Managers oder im SVP-Menü der zugeordneten KVP-Konsole) nochmals verändert werden.
- in einem VM2000-Gastsystem:
  1. Die Datei `$TSOS.SYSPAR.BS2.<ver>.<vm-name>`
     (für SU /390 und für SU x86, wenn das Gastsystem mit `/START-VM` gestartet wird).
     `<vm-name>` ist dabei der VM-Name des Gastsystems.
  2. Die Datei `$TSOS.SYSPAR.BS2.<ver>.<system-name>`
     (für SU x86, wenn das Gastsystem nicht mit `/START-VM` gestartet wird).
     `<system-name>` ist dabei der Systemname, der entweder für die persistente VM vergeben wurde (Auto-IPL-Option) oder im IPL-Menü (Dialogfenster des SE Managers oder im SVP-Menü der zugeordneten KVP-Konsole) angegeben oder verändert wurde.

  > **i** Wenn das Gastsystem auf SU x86 mit `/START-VM` gestartet wird, dann wird der Systemname dem VM-Namen angeglichen.

- wird keine spezifische Parameterdatei gefunden, wird nach der Datei mit dem Standardnamen `$TSOS.SYSPAR.BS2.<ver>` gesucht.
- wird keine der o.g. Parameterdateien gefunden, so wird ein FAST- oder AUTOMATIC-Startup unterbrochen und es wird für die Dauer des Parameter-Einlesens auf DIALOG-Startup umgeschaltet.

Dialoggesteuerte Auswahl der Parameterdatei

Beim **DIALOG-**Startup stehen dem Operator die Eingabequellen Platte und Konsole für die Zuweisung zur Verfügung. Die Zuweisung erfolgt mit Beantwortung der Meldung `NSI0050` (siehe "Rep-Sätze", Eingabe von Rep-Sätzen über Konsole):

- Parametersätze können aus Dateien eingelesen oder direkt an der Konsole eingegeben werden.
- Hat die Systembetreuung die Parameterdatei auf einer Platte hinterlegt, dann erfolgt die Zuweisung über den entsprechenden Dateinamen.
- Durch die Antwort `P.CONS` auf die Meldung `NSI0050` wird auf die Eingabe von Konsole umgestellt. Folgende Meldung fordert den Operator zur Eingabe des Schlüsselwortes für die nachfolgenden Parametersätze auf:

  ```
  NSI4120 ENTER PARAM KEYWORD: (<KEYWORD>; ? (DISPLAY LIST OF POSSIBLE
          KEYWORDS); EOT (END))
  ```

  Der Operator hat die Möglichkeit, sich durch Angabe von „?“ alle möglichen Schlüsselwörter anzeigen zu lassen. Das Schlüsselwort muss direkt eingegeben werden. Die Ausgabe `BEGIN` wird intern für die Protokollierung in der CONSLOG-Datei generiert.

  Die Eingabe der einzelnen Parametersätze wird durch folgende Meldung angefordert:

  ```
  NSI4125 ENTER PARAM RECORD FOR KEYWORD (&00); (<PARAM>; /- (IGNORE LAST
          RECORD); EOT (END))
  ```

  Diese Meldung wird solange wiederholt, bis `EOT` eingegeben wird. In diesem Fall wird die Anweisung EOF intern generiert. Es folgt wiederum die Meldung `NSI4120`, die zur Eingabe des nächsten Schlüsselworts auffordert.

  Mit der Angabe von `EOT` wird die Eingabe von der Konsole beendet, ein „/END-PARAMS“-Satz wird ebenfalls intern generiert.

Das Einlesen der Datensätze erfolgt unmittelbar vor dem Laden des Klasse-1-Execs.
Die Parameter- und Steuersätze werden komprimiert in einem Puffer im Startup-Lademodul gespeichert.

Alle Sätze, die über den Parameterservice verarbeitet wurden, werden in Form von Meldungen in der Protokolldatei CONSLOG erfasst.

Wird das Einlesen einer Parameterdatei mit Fehler abgebrochen, dann erhält der Operator an der Konsole nach einer Fehlermeldung die Meldung `NSI008F`. Der Operator kann angeben, ob die bereits eingelesenen Sätze erhalten bleiben oder ignoriert werden sollen. Danach erscheint die Meldung `NSI0050`. Ein Eingabegerät kann jeweils angegeben werden.

Dieser Vorgang wiederholt sich solange, bis der Operator den Parameterservice mit der Eingabe `END` beendet. Stehen in der Parameterdatei fehlerhafte Anweisungen für den Parameterservice, so werden diese Anweisungen an der Konsole gemeldet und können sofort vom Operator korrigiert oder übergangen werden.

3.2 Aufbau und Inhalt einer Parameterdatei

Eine Parameterdatei setzt sich zusammen aus Parametersätzen, Kommentarsätzen und Steuersätzen.

Parametersätze sind Datensätze, die nur von der entsprechenden Software-Komponente interpretiert werden. Die Einleseroutine übernimmt die Parametersätze ungeprüft. Kommentarsätze beginnen immer mit einem *(Stern), werden vom Parameterservice ignoriert und deshalb auch nicht protokolliert.
Steuersätze sind die Sätze, die Anweisungen für den Parameterservice enthalten.
Sie beginnen immer mit einem Schrägstrich.

Anweisung **BS2000 PARAMS**

Diese Anweisung ist immer die erste in der Parameterdatei (entfällt bei der Eingabe von der Konsole).

Anweisung **BEGIN <schlüsselwort>**

Das angegebene Schlüsselwort bestimmt die Zugehörigkeit der folgenden Parametersätze zur jeweiligen Software-Komponente. Mögliche Schlüsselwörter sind:

- ACCOUNT
- DSSM
- GTIME
- IOCONF
- ISAM
- JMS
- MEMORY
- OPR
- PAGING
- SNAP
- SYSOPT-CLASS2
- SYSOPT-IPL

Das Schlüsselwort SRPM wird vom Softwareprodukt SECOS zur Festlegung der Benutzerkennung des Sicherheitsbeauftragten verarbeitet. Der Parametersatz wird im SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46] beschrieben. Die Schlüsselwörter BCAM und VM2000 werden in den jeweiligen Produkthandbüchern erläutert.

Anweisung **EOF**

Diese Anweisung schließt den Parameterabschnitt für die mit <schlüsselwort> angesprochene Software-Komponente ab.

Anweisung **ADD**

Eine Parameterdatei kann auch die Namen weiterer Parameterdateien enthalten, die verwendet werden sollen. Die mit den ADD-Anweisungen angegebenen Dateien werden eingelesen, sobald die Parameterdatei, die die ADD-Anweisungen enthält, vollständig eingelesen ist.

Es dürfen maximal 16 ADD-Anweisungen angegeben werden.
Nach einer ADD-Anweisung sind keine Anweisungen außer der ADD-Anweisung selber und der END-PARAMS-Anweisung erlaubt. In einer über ADD definierten Parameterdatei können keine weiteren ADD-Anweisungen angegeben werden.

Anweisung **END-PARAMS**

Diese Anweisung ist die letzte in der Parameterdatei.

Die zu einer Software-Komponente gehörenden Parametersätze müssen nicht zwingend alle hintereinander angegeben sein. Es dürfen mehrere BEGIN-Anweisungen mit demselben Schlüsselwort und darauffolgenden Parametersätzen und EOF-Anweisungen in der Parameterdatei stehen.

Als Muster der Systemparameterdatei wird die Datei SYSPAR.BS2.<ver>.TEMPLATE ausgeliefert. Sie enthält

- die Anweisungen BEGIN <schlüsselwort> und EOF
- Mustervorgaben für Parameter, wenn ohne Vorgaben keine Standardwerte gültig werden (betrifft nur den Parametersatz für GTIME)

  Nur für GTIME sind explizite Angaben notwendig. In die Musterdatei wurden die zum Zeitpunkt der Versionsfreigabe bekannten gesetzlichen Zeitangaben für drei Zeitzonen aufgenommen (Zeitzonen 0, 1 und 2, einschl. der Informationen zur Sommer-/Winterzeit-Umstellung).
  Bei Betrieb von BS2000 in einer dieser Zeitzonen können die Angaben für die jeweils anderen Zeitzonen gelöscht werden (bei Betrieb in der Zeitzone 3 müssen die Kommentarzeichen entfernt werden).
  Bei Betrieb in einer anderen als den angegebenen Zeitzonen müssen die Angaben in gleicher Form nach den gültigen Umstellzeiten vom Kunden selber erstellt werden.

Die Standardwerte für den Parametersatz OPR haben zur Folge, dass alle Konsolmeldungen, die eine explizite Routingvorgabe brauchen, nicht auf der Hauptkonsole ausgegeben werden.
Die Musterdatei muss zum Betrieb als Systemparameterdatei zu SYSPAR.BS2.<ver> umbenannt werden. Sie kann um kundenspezifische Parametervorgaben erweitert werden.

Beispiel für eine Parameterdatei

```
/BS2000 PARAMS

*Accouning system
/BEGIN ACCOUNT
/START-ACC NAME=$RZ.
/EOF

*Paging
/BEGIN PAGING
PAGING VOLUME=(PUBA04,PUBA05)
/EOF

*Memory management
/BEGIN MEMORY
SHRSIZE 6,UNIT=1MB
ASAMCTRL 4
/EOF

*System time and season
/BEGIN GTIME
ZONE=+01:00
DIFF=1:00
SEASON=S
CHDATE=1900-01-01/00:00
CHDATE=2011-03-27/02:00
CHDATE=2011-10-30/03:00
CHDATE=2012-03-25/02:00
CHDATE=2012-10-28/03:00
/EOF

*TSN mode
/BEGIN JMS
TSN-MODE=A
/EOF

*DSSM
/ADD PAR.FILE.DSSM

/END-PARAMS
```

Jede auf Platte befindliche Parameterdatei muss eine SAM-Datei mit variablen Sätzen in Standard-Blockung sein. Die Satzlänge muss ohne Satzlängenfeld 1-72 Byte sein, mit Satzlängenfeld demnach 5-76 Byte.
Für BLKCTRL=PAMKEY gilt, dass in den ersten 2 Byte eines Blockes die sedezimale Blocklänge, in den ersten 2 Byte eines Satzes die sedezimale Satzlänge (d.h. Satzformat V) erwartet wird.
Im NONKEY-Modus wird die Blocklänge aus dem simulierten Schlüssel innerhalb des Datenfeldes geholt (Länge 4 Byte).

Parameter-Eingabedateien können z.B. mit EDT erstellt, fortgeschrieben und gemischt werden.

3.3 Starten des Abrechnungssystems (ACCOUNT)

Das Abrechnungssystem kann im Rahmen der Systemeinleitung oder später per Kommando START-ACCOUNTING aktiviert werden, indem die dazu notwendigen Anweisungen über den Parameterservice mitgeteilt werden. Die Systembetreuung muss für diesen Zweck in der Parameterdatei mit dem standardmäßigen Namen SYSPAR.BS2.<ver>[.<name>] die entsprechenden Angaben bereitstellen.

Wird das Abrechnungssystem über den Parameterservice aktiviert und liegt die Abrechnungsdatei auf einem anderen als dem Home-Pubset, kann das zu folgendem Problem führen: Die Datei kann nicht eröffnet werden, weil das Importieren dieses Pubsets noch nicht abgeschlossen ist. Deshalb muss in diesem Fall die Abrechnungsdatei generell auf dem Home-Pubset liegen.

Schlüsselwort für den BEGIN-Satz ist **ACCOUNT**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Der Parametersatz beginnt wahlweise mit der Anweisung START-ACCOUNTING bzw. STOP-ACCOUNTING. Die Anweisungen und die dazugehörigen Operanden können bis zur Eindeutigkeit abgekürzt werden.

Anweisung **START-ACCOUNTING**

Die Anweisung /START-ACCOUNTING aktiviert das Abrechnungsverfahren, wobei grundsätzlich die selben Operanden vereinbart werden können wie beim Kommando START-ACCOUNTING.

Format des Parametersatzes zum Starten des Abrechnungssystems

Anweisung	Operanden
/START- ACC[OUNTING]	[NAME = *STD / file] [,SPACE = STD / (primary,secondary)] [,BLKSIZE = STD / (*STD,n)] [,VOLUME = *STD / vsn] [,ALT[ERNATIVE-FILES] = *NONE / file / (file,...)] [,SET[-RECORD-TYPE] = *STD / *ALL / record / (record,...)] [,ADD[-RECORD-TYPE] = *NONE / record / (record,...)] [,REMOVE[-RECORD-TYPE] = *NONE / record / (record,...)] [,ACCOUNTING-PERIOD = *STD / period] [,JOB-CLASS = *NONE / *ALL / job-class / (job-class,...)]

Die Anweisung START-ACCOUNTING kann mit dem Fortsetzungszeichen Bindestrich auf mehrere Zeilen aufgeteilt werden.

Die Fortsetzungszeilen müssen syntaktisch abgeschlossen sein, da sie vom System einzeln überprüft werden. Deshalb dürfen bei der Anweisung START-ACCOUNTING die Operanden mehrfach angegeben werden.

Da es vorkommen kann, dass nicht alle Werte einer Liste in eine Zeile passen, wirken die Operanden ADD-RECORD-TYPE, REMOVE-RECORD-TYPE und ALTERNATE-FILES ergänzend. Die Liste kann dann in der nächsten Zeile mit gleichen Operanden fortgesetzt werden (der Operand SET-RECORD-TYPE wird mit dem Operanden ADD-RECORD-TYPE fortgesetzt).

Anweisung **STOP-ACCOUNTING**

Die Anweisung /STOP-ACCOUNTING vereinbart, dass im Systemlauf keine Abrechnungssätze gesammelt werden sollen.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN ACCOUNT
/START-ACCOUNTING NAME=$RZ.,-                                    1.
SPACE=(99,99),-                                                  2.
SET=*ALL,JOB-CLASS=*ALL                                          3.
/EOF
:
/END-PARAMS
```

1. Die Abrechnungsdatei wird unter der Benutzerkennung RZ mit ihrem Standardnamen SYS.ACCOUNT.<date>.xxx.nn katalogisiert, wobei <date> in der Form yy.mm.dd oder yyyy-mm-dd auftreten kann, abhängig von der Einstellung des Systemparameters FMTYFNLG.
2. Die Speicherplatzzuweisung beträgt primär und sekundär 99 PAM-Blöcke.
3. Sämtliche Abrechnungssätze werden geschrieben, und die Daten aller Jobklassen sollen periodisch in der Abrechnungsdatei erfasst werden.

Zum Abrechnungssystem von BS2000 siehe auch das Kapitel „Accounting".

3.4 Inbetriebnahme der dynamischen Subsystemverwaltung (DSSM)

Die dynamische Subsystemverwaltung wird während der BS2000-Systemeinleitung gestartet.
Über den Parameterservice werden alle zur Initialisierung von DSSM notwendigen Informationen mitgegeben. Dazu zählen der Name des statischen Subsystemkataloges und die DSSM-Versionsnummer. Des Weiteren kann bereits – wenn unbedingt erforderlich – die Protokollierung von DSSM-spezifischen Daten für die Fehlerdiagnose eingeschaltet werden.

Schlüsselwort für die Inbetriebnahme der Subsystemverwaltung ist **DSSM**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Format des Parametersatzes zur Inbetriebnahme der dynamischen Subsystemverwaltung

Format	Bedeutung
SSMCAT = name	Name des statischen Subsystemkatalogs
VERSION = version	Versionsnummer von DSSM
LOGGING = ON / OFF	Steuert die DSSM-spezifische Protokollierung zur Fehlerdiagnose

Subsysteme, die nicht während der Systemeinleitung automatisch aufgebaut werden, können im BS2000-Systemlauf mit dem Kommando START-SUBSYSTEM gestartet werden.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN DSSM
SSMCAT=<name>                                                    1.
VERSION=<version>                                                2.
LOGGING=OFF                                                      3.
/EOF
:
/END-PARAMS
```

1. Jede DSSM-Version kann Subsystemkataloge verwenden, die mit bestimmten SSCM-Versionen generiert wurden (aktuell DSSM V4.3 und SSCM V2.3).
   Die Steuer- und Parametersätze müssen in der Parameterdatei nur vorhanden sein, wenn die Systembetreuung von folgenden Standardwerten abweichen will: SSMCAT=$TSOS.SYS.SSD.CAT.X, VERSION=043 und LOGGING=OFF.
2. Die Versionsnummer bezieht sich auf alle DSSM-spezifischen Dateinamen (z.B. SYSLNK.DSSM.043, SYSREP.DSSM.043) und wird dreistellig angegeben (z.B. 043).
3. Die Anweisung LOGGING=OFF (Standardwert) bewirkt, dass die Protokollierung ausgeschaltet ist; (bei LOGGING=ON würde bereits während der Inbetriebnahme von DSSM ein Protokoll der Diagnosedaten erzeugt).

3.5 Steuerung der Systemzeit (GTIME)

In der Parameterdatei werden die Initialisierungsdaten für das Subsystem GET-TIME hinterlegt, das dem Benutzer über die Systemfunktion GTIME Informationen zur standardisierten Weltzeit und der Zeitverschiebung liefert. Auch das Betriebssystem benötigt diese Information.

Zur Initialisierung und Verwaltung der Systemzeit siehe das Kapitel „Systemzeit-Verwaltung".

Die Systembetreuung legt über verschiedene Parameter die Relation zwischen der Systemzeit (lokale Zeit) und der universellen Weltzeit UTC fest. Damit kann das System und der Benutzer des Subsystems GET-TIME sowohl auf ein lokales als auch auf ein systemübergreifend verständliches Zeitbezugssystem (UTC) zugreifen.
Ohne diese Daten (aus der Parameterdatei oder im Dialog über Konsole) kann die Systemeinleitung **nicht** durchgeführt werden.

Die SVP-Uhr muss, außer bei automatischem Restart oder bei Gastsystembetrieb, zur Systemeinleitung die korrekte lokale Zeit (Systemzeit) enthalten.

Schlüsselwort für die Festlegung der Relation zwischen der Systemzeit und der universellen Weltzeit in der Parameterdatei ist **GTIME**.
Die maximale Anzahl der dafür zugelassenen Parametersätze beträgt 256.

Format des Parametersatzes zur Steuerung der Systemzeit

Format	Bedeutung
[NEXTZONE]	Es beginnt ein neuer GTIME-Parameterblock
ZONE = +hh:mm / -hh:mm	Zeitzone
DIFF = h:mm	Größe des Zeitsprungs (Umstellungsdifferenz)
SEASON = S / W	Angabe Sommer-/Winterzeit vor dem ersten Umstellungszeitpunkt
EPOCH = 00 / nn	Angabe der Epoche für das TODR (2 Sedezimalzeichen). EPOCH=00 bezeichnet die Standard-Epoche von 1900 bis 2042.
CHDATE = yyyy-mm-dd/hh:mm : CHDATE = yyyy-mm-dd/hh:mm	Umstellungszeitpunkt 1 : Umstellungszeitpunkt n (max. 125)
SINGLE	Keine zukünftigen Umstellungszeitpunkte (Single Season)

> **i** Umstellungszeitpunkte können auch im laufenden BS2000-Betrieb mit den Kommandos ADD-/MODIFY-/REMOVE-/SHOW-CHANGE-DATE verwaltet werden.
> Änderungen von Umstellungszeitpunkten durch die Kommandos müssen aber manuell in die Parameterdatei eingetragen werden, wenn sie im nächsten Systemlauf gültig sein sollen.

NEXTZONE
Trennt die GTIME-Parameter verschiedener Zeitzonen voneinander ab. Dadurch können die Daten für mehrere Zeitzonen in der GTIME-Parameterdatei enthalten sein.
Die Angabe dieses Operanden kann entfallen, wenn in der Parameterdatei nur die Daten für eine Zeitzone enthalten sind.

ZONE = -/+hh:mm
Zeitzone in Stunden und Minuten.
Dieser Wert beschreibt die lokale, gesetzliche Zeitzone im Vergleich zur „Greenwich Mean Time“, der UTC (Universal Time Coordinate).
Wertebereich: -12:00 <= hh:mm <= +11:59
Beispielsweise beträgt der Vorsprung der mitteleuropäischen Zeit gegenüber der UTC eine Stunde; als Wert ist folglich +01:00 anzugeben.
Der Wert ZONE **muss** in der Parameterdatei angegeben werden.

DIFF = h:mm
Größe des Zeitsprungs bei einer Umstellung zwischen Sommer- und Winterzeit.
Wertebereich: 0:00 <= h:mm <= 9:59
Der Wert DIFF **muss** in der Parameterdatei angegeben werden.
Ist DIFF nicht 0:00, müssen die Operanden SEASON und – mindestens ein – CHDATE angegeben werden.

SEASON = S / W
Gibt an, ob vor dem Zeitpunkt der **ersten** Umstellung **S**ommer- oder **W**interzeit gültig war. (Als „Winterzeit“ wird hier die eigentliche Normalzeit bezeichnet: Die von der Normalzeit abweichende „Daylight-Saving-Time“ wird „Sommerzeit“ genannt.)
Dieser Wert muss für die interne Zeitberechnung angegeben werden, wenn mit dem Operanden CHDATE eine Umstellung der Zeit vereinbart wird. Von diesem Ausgangswert muss die Systemfunktion GTIME, auch nach mehreren Umstellungen, auf die jeweils gültige Zeit schließen.

Dieser Wert wird von der Systemfunktion CTIME nicht ausgewertet. Da andere Systemfunktionen (z.B. JMS, DVS) intern CTIME benutzen, sind die beim Operanden CHDATE aufgeführten Hinweise zu CTIME und SEASON zu berücksichtigen.

EPOCH = 00 / xx
Gibt die Epoche für das TODR an (2 Sedezimalzeichen).
EPOCH=00 bezeichnet die Standard-Epoche von 1.1.1900 bis 17.9.2042 .

> **!** Voraussetzung für den Einsatz einer geänderten TODR-Epoche ist, dass Zeitstempel nicht mehr benutzt werden, welche zuvor mit einer anderen Epocheneinstellung ermittelt und von einer Applikation abgespeichert wurden und die außerhalb des mit der neuen TODR-Epoche darstellbaren Bereichs liegen. Solche gespeicherten Zeitstempel würden mit der geänderten TODR-Epoche falsch interpretiert werden. Dies ist durch die System- oder Anwendungsbetreuung vor der Änderung der TODR-Epoche unbedingt zu überprüfen.
>
> "Alte" Zeitstempel können nach Umwandlung in das TODX-Format mit "neuen" Zeitstempeln verglichen werden, siehe Abschnitt "Umwandlung von TODR-Zeitstempeln in TODX-Zeitstempel" in "TODR-Epoche ". Zeitstempel im TODX-Format können stets verglichen werden, denn TODX-Werte sind im Zeitraum von 1900 bis 4317 monoton steigend. Solch eine Umwandlung setzt allerdings voraus, dass bekannt ist, mit welcher Epocheneinstellung die "alten" Zeitstempel erzeugt worden sind.

CHDATE = yyyy-mm-dd/hh:mm

Vereinbarung von Umstellungszeitpunkten (1..125). Das erste Datum muss mit 1900 beginnen, die nachfolgenden müssen lückenlos und zeitlich aufsteigend sein (siehe Beispiel auf der nächsten Seite).
Format und Wertebereich des Datums:

yyyy: Jahr (1900 <= yyyy < 2042)

mm: Monat (1 <= mm <= 12)

dd: Tag des Monats (1 <= dd <= 31)

hh: Stunde (0 <= hh <= 23)

mm: Minute (0 <= mm <= 59)

Die Datumsangaben dienen dem System beim Startup zur Bestimmung, ob momentan Sommer- oder Winterzeit vorherrscht, und somit zur Festlegung der Differenz zwischen lokaler Zeit und der Weltzeit UTC. Damit kann die Parameterdatei über mehrere Umstellungszeitpunkte hinweg verwendet werden.

Weiterhin werden die Umstellungszeitpunkte für die CTIME-Funktion benötigt, die u.a. Zeitangaben von lokaler Zeit in UTC umwandelt. CTIME geht bei der Interpretation von lokalen Zeitstempeln immer davon aus, dass von 1900-01-01/00:00 bis zum ersten CHDATE Winterzeit herrscht, auch wenn z.B. das erste CHDATE 1994-09-25/03:00 lautet und SEA-SON=S angegeben wurde. Es werden dann aus Benutzersicht Zeitstempel vor dem ersten CHDATE fälschlicherweise als Winterzeitstempel interpretiert.
Um diese Zweifelsfälle zu vermeiden, ist es sinnvoll, die Liste bzgl. der zurückliegenden Umstellungszeitpunkte möglichst vollständig zu halten. Optimal für die einwandfreie Funktion der CTIME-Schnittstelle ist es, als erstes Datum 1900-01-01/00:00 mit SEASON=S einzutragen. Das zweite CHDATE muss dann eine Winter- nach Sommerzeitumstellung festlegen. Dann kann für alle zurückliegenden Zeitangaben entschieden werden, ob sie als Sommer- oder Winterzeit interpretiert werden müssen.

Die Differenz zwischen zwei Umstellungszeitpunkten muss im Bereich von 4 bis 8 Monaten liegen (Ausnahme: die Differenz zwischen dem CHDATE 1900-01-01/00:00 und dem zweiten CHDATE darf beliebig groß sein).

SINGLE

Gibt an, dass es in der Zukunft keine Umstellung zwischen Sommer- und Winterzeit mehr geben wird. Dieses Schlüsselwort muss unmittelbar hinter dem letzten und abschließenden Umstellungszeitpunkt (CHDATE) für die Region angegeben werden. Diese Funktion wurde eingeführt, um das BS2000 auf die mögliche Einführung einer festen, unveränderlichen Zeitzone in bestimmten Regionen vorzubereiten. Fehlt dieses Schlüsselwort, führt das Warnmeldungen (ETMGT18) an der Konsole, falls zukünftige Umstellungszeitpunkte fehlen.

Unrichtige GTIME-Parameter verfälschen die Zeitrelation von Systemzeit zur universellen Weltzeit UTC und haben ebenso fatale Auswirkungen wie eine falsch eingestellte SVP-Uhr!
Die Zeitumstellung erfolgt unterbrechungsfrei, d.h. das System wird kontinuierlich über eine Zeitumstellung hinweg betrieben, weil das TODR nicht mehr die genaue lokale Zeit enthalten muss. Die lokale Zeit wird aus dem Inhalt des TODR und einem Korrekturwert ermittelt (siehe Kapitel „Systemzeit-Verwaltung").

Die zentral ausgelieferte Muster-Parameterdatei enthält GTIME-Parametern für mehrere Zonen. Um die Parameter für die richtige Zeitzone auszuwählen, sollte an SU /390 die Differenz zwischen UTC zur vorliegenden Zeitzone am SVP eingestellt werden. Auf SU x86 ist die Zeitzone im Trägersystem eingestellt. Die Zeitzone wird auf allen Servern aus der SVP-Zeit (STORE REAL CLOCK) ermittelt (siehe Kapitel „Systemzeit-Verwaltung").

Wenn die Zeitzone nicht aus der SVP-Zeit ermittelt werden kann (Fehlerfall), dann wird der Eintrag im SVL der Startup-Platte zur Auswahl der richtigen Parameter herangezogen. Wenn auch dort kein gültiger Eintrag vorhanden ist und mehr als eine Zeitzone in den Parametersätzen enthalten ist, dann wird der Operator mit den Meldungen `ETMGT30` und `ETMGT31` nach einer Zonenangabe gefragt. Alternativ kann er die Systemeinleitung beenden.
Wenn die Zeitzone weder durch die SVP-Zeit noch durch das SVL der Startup-Platte bestimmt werden kann, die Parameterdatei aber genau eine Zeitzone enthält, dann wird diese als einzustellende Zeitzone übernommen.

Wenn eine Zeitzone durch die SVP-Zeit oder das SVL bestimmt ist, es aber keinen passenden Parametersatz dazu gibt, dann wird der Operator mit den Meldungen `ETMGT35` und `ETMGT36` nach der einzustellenden Zeitzone gefragt. Alternativ kann er die Systemeinleitung beenden.

In allen Fällen, in denen die Systemeinleitung fortgesetzt wird, wird die erkannte bzw. akzeptierte Zeitzone im SVL der Startup-Platte hinterlegt.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN GTIME
ZONE=+01:00                                                    1.
DIFF=1:00                                                      2.
SEASON=S                                                       3.
EPOCH=00                                                       4.
CHDATE=1900-01-01/00:00                                        5.
CHDATE=1980-04-06/02:00
CHDATE=1980-09-28/03:00
CHDATE=1981-03-29/02:00
:
CHDATE=2013-03-21/02:00                                        6.
CHDATE=2013-10-27/03:00
CHDATE=2014-03-30/02:00
CHDATE=2014-10-26/03:00
CHDATE=2015-03-29/02:00
CHDATE=2015-10-25/03:00
CHDATE=2016-03-27/02:00
CHDATE=2016-10-30/03:00
:
/EOF
:
/END-PARAMS
```

1. Es ist die mitteleuropäische Zeitzone angegeben.
2. Der Differenzwert von einer Stunde bezeichnet die Höhe des Sprungs, der bei einer Umstellung zwischen Sommer- und Winterzeit erfolgt.
3. Vor dem Pseudo-CHDATE muss Sommerzeit eingestellt sein.
4. Es gilt die Standard-Epoche von 01.01.1900 bis 17.09.2042 für das TODR.
5. Pseudo-CHDATE: Er bewirkt, dass bis zum ersten wirklichen CHDATE Winterzeit gilt. Das entspricht der CTIME-Philosophie, die davon ausgeht, dass von 1900 an bis zum ersten eingetragenen CHDATE Winterzeit herrscht.
6. Zukünftige Umschalttage werden eingetragen.

3.6 Parametersatz IOCONF

Die Ein-/Ausgabe-Tabellen von BS2000 werden bei Startup aufgebaut.
Die Kanalperipherie (SU /390) wird aus dem aktiven IOCF ermittelt.
Die Bus- und Fibre Channel-Peripherie (SU x86) wird über X2000 bestimmt.

Reihenfolge der Ermittlung des Konfigurationszustandes bei Startup:

1. Aus den Voreinstellungen für die Busperipherie (SU x86):

   ATTACHED: Konsolen, Netzwerk-/LAN-Geräte, Geräte der Familie '50' und alle Plattengeräte; Um Platten standardmäßig DETACHED zu generieren, kann ein optionaler Rep eingesetzt werden.

   DETACHED: alle übrigen Geräte.

2. Die Konfigurationszustände der Geräte auf SU /390 werden aus den IOGEN-Daten im IOCF übernommen. Die Voreinstellungen für die Kanäle, Steuerungen und alle Verbindungen sind:

   ATTACHED: Kanäle und Mehrfachsteuerungen (Controller)

   INCLUDED: alle Verbindungen zwischen den Hardware-Einheiten

3. Aus dem Startup-Parameterservice (IOCONF) über die Anweisungen:

   MOD-IO-UNIT Konfigurationszustand für eine Hardware-Einheit festlegen

   MOD-IO-PATH Konfigurationszustand für eine Verbindung festlegen

4. Durch automatische Rekonfiguration im Startup:

   ATTACHED: alle Platten des Home-Pubsets, Paging-Platten, IPL-Konsole

   DETACHED: alle Bandgeräte

   Die impliziten Zustände aller Hardware-Einheiten und Verbindungen werden ermittelt.

Das Schlüsselwort für die Änderung von Konfigurationszuständen ist **IOCONF**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 256.

Fehlerhafte Parametersätze werden ignoriert. Nach Beendigung des Startup wird die Meldung `NDI0550` ausgegeben, die die Anzahl der empfangenen Parametersätze und die Anzahl der fehlerhaften Sätze enthält. Bis zu 10 fehlerhafte Sätze werden mit der Meldung `NDI0551` ausgegeben, zusammen mit der Information, welche Art von Fehler vorliegt.

Die über den Parameterservice eingestellten Werte gelten für die laufende Session.

Anweisung	**Bedeutung**
MOD-IO-UNIT	Konfigurationszustand einer Hardware-Einheit ändern
MOD-IO-PATH	Konfigurationszustand einer Verbindung zwischen Hardware-Einheiten ändern

Tabelle 4: Übersicht der Anweisungen zum Parametersatz IOCONF

3.6.1 Anweisung MOD-IO-UNIT

Format zur Änderung des Konfigurationszustandes einer Hardware-Einheit

Anweisung	Operanden	Bedeutung
MOD-IO-UNIT	*class(unit)	Klasse und mnemotechnischer Gerätename der Hardware-Einheit
	,STATE = ATT / DET / INV	Neuer Konfigurationszustand der Hardware-Einheit
MOD-IO-UNIT	*DEV(*RA[NGE](unit, range))	Bereich von Hardware-Einheiten
	,STATE = ATT / DET / INV	Neuer Konfigurationszustand der Hardware-Einheiten des Bereichs

> Mit Ausnahme der explizit angegebenen Abkürzungsmöglichkeiten müssen alle anderen Angaben vollständig ausgeschrieben werden. Auch das Weglassen von Operandennamen (Stellungsschreibweise) ist nicht zulässig.

***class(unit)**
Klasse und mnemotechnischer Gerätename der Hardware-Einheit, für die der Konfigurationszustand angegeben wird.

Für die Klasse *class können folgende Werte angegeben werden:

- `*CHA` bzw. `*CHN` (Kanal, Channel)
- `*CON` bzw. `*CTL` (Mehrfachsteuerung, Controller)
- `*DEV` bzw. `*DVC` (Endgerät, Device)

unit (2 oder 4 Zeichen) ist der mnemotechnische Gerätename einer Hardware-Einheit der angegebenen Klasse.

***DEV(*RA[NGE](unit,range))**
Mnemotechnischer Gerätename einer bestimmten Hardware-Einheit und Anzahl der darauf folgenden Geräte, für die der Konfigurationszustand angegeben wird.

unit (2 oder 4 Zeichen) ist der mnemotechnische Gerätename einer Hardware-Einheit.

range ist ein dezimaler Wert zwischen 2 und 999, der die Anzahl der Geräte angibt, für die – angefangen bei unit – der Konfigurationszustand geändert werden soll.

Die range-Angabe ist nur für Endgeräte (*class = *DEV/*DVC) erlaubt.

Bei 4-Byte-Mnemoniks wird ab unit hexadezimal weitergezählt, bei 2-Byte-Mnemoniks wird ab unit alphabetisch und anschließend mit den Ziffern 0 .. 9 weitergezählt.

range = 1 wird ignoriert, range = 0 mit Syntaxfehler abgewiesen.

Voraussetzung ist die Existenz der ersten Mnemonik (angegeben bei unit) im System.

Alle folgenden Mnemoniks werden ignoriert, wenn sie nicht im System vorhanden sind.

Beispiele

*RANGE(A100,32)	A100, A101, ..., A11F
*RA(A1FF,5)	A1FF, A200, A201, A202, A203
*RA(FA,40)	FA, FB, ..., FZ, F0, F1, F2, ..., F9, GA, GB, GC, GD

STATE =
Bestimmt den neuen Konfigurationszustand der Hardware-Einheit.

STATE = ATT / DET / INV
Die Hardware-Einheit ist im Zustand ATTACHED (zugeschaltet), DETACHED (weggeschaltet) oder INVALID (ungültig).

3.6.2 Anweisung MOD-IO-PATH

Format zur Änderung des Konfigurationzustandes von Verbindungen

Anweisung	Operanden	Bedeutung
MOD-IO-PATH	FROM = *class(unit)	Ausgangspunkt der Verbindung
	,TO = *class(unit)	Endpunkt der Verbindung
	,STATE = INC / REM / INV	Neuer Konfigurationszustand der Verbindung

> Mit Ausnahme der explizit angegebenen Abkürzungsmöglichkeiten müssen alle anderen Angaben vollständig ausgeschrieben werden. Auch das Weglassen von Operandennamen (Stellungsschreibweise) ist nicht zulässig.

FROM = *class(unit)
Klasse und mnemotechnischer Gerätename des Ausgangspunkts der Verbindung, für die der Konfigurationszustand angegeben wird.
Für die Klasse *class können folgende Werte angegeben werden:

- `*CHA` bzw. `*CHN` (Kanal, Channel)
- `*CON` bzw. `*CTL` (Mehrfachsteuerung, Controller)
- `*DEV` bzw. `*DVC` (Endgerät, Device)

unit (2 oder 4 Zeichen) ist der mnemotechnische Gerätename einer Hardware-Einheit der angegebenen Klasse.

TO = *class(unit)
Klasse und mnemotechnischer Gerätename des Endpunkts der Verbindung, für die der Konfigurationszustand angegeben wird.
*class(unit) siehe FROM.

STATE =
Bestimmt den neuen Konfigurationszustand der Verbindung.

STATE = INC / REM / INV
Die Verbindung ist im Zustand INCLUDED (zugeschaltet), REMOVED (weggeschaltet) oder INVALID (ungültig).

3.7 Voreinstellungen für NK-ISAM (ISAM)

Die indexsequenzielle Zugriffsmethode NK-ISAM des DVS unterstützt die Verwendung von PAM-Key-losen Platten. Darüber hinaus bietet NK-ISAM Möglichkeiten zur Performance-Optimierung durch die Einsparung von Platten-Ein-/Ausgaben. Dazu müssen die Pufferbereiche von NK-ISAM, die ISAM-Pools, geeignet dimensioniert werden.

Diese **ISAM-Pools** werden im privilegierten Adressraum oder in Data-Spaces angelegt und dienen als Zwischenspeicher für die Verarbeitung einer oder mehrerer NK-ISAM-Dateien. ISAM-Pools können sowohl vom Benutzer explizit über spezielle Makros bzw. Kommandos angelegt und verwaltet werden als auch implizit vom System. Sowohl der Benutzer als auch das System können tasklokale oder taskübergreifende ISAM-Pools anlegen. Taskübergreifende ISAM-Pools werden automatisch dateispezifisch in einem Data Space angelegt.

Tasklokale ISAM-Pools können nur von der aufrufenden Task genutzt werden, taskübergreifende ISAM-Pools definieren einen Pufferbereich, der über Angabe des Pool-Namens auch anderen Jobs zugänglich ist.

Wird eine ISAM-Datei eröffnet, ohne dass sie vom Benutzer im Kommando oder Makro einem bestimmten ISAM-Pool zugeordnet wurde, verwendet das DVS für die Verarbeitung einen Standard-ISAM-POOL des Systems:
Bei einem OPEN mit SHARUPD=NO wird ein tasklokaler Standard-ISAM-POOL zugeordnet, bei einem OPEN mit SHARUPD=YES ein taskübergreifender Standard-ISAM-POOL.

Die Größe eines Standard-ISAM-Pools legt die Systembetreuung in der Parameterdatei fest. Dort wird ebenso die Größe eines benutzerdefinierten ISAM-Pools festgelegt, wenn beim Anlegen des ISAM-Pools keine Angaben zu dessen Größe gemacht wurden.

Schlüsselwort für den BEGIN-Satz ist **ISAM**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Die Parameter für NK-ISAM können mehrfach angegeben werden. Gültig ist jeweils der zuletzt angegebene Wert.

Format des Parametersatzes zur Voreinstellungen für NK-ISAM

LCLDFPS = zahl	
GLBDFPS = zahl	*wird nicht mehr ausgewertet*
GLBPS = zahl	
LCLPS = zahl	
DFPPROC = *STD / *ADV[ANCED]	*wird nicht mehr ausgewertet*
GLBDFPN = zahl	*wird nicht mehr ausgewertet*
MAXDSBN = zahl	

LCLDFPS = zahl
gibt die Größe der tasklokalen Standard-ISAM-Pools in PAM-Seiten an.
Mögliche Werte: 96 <= zahl <= 8192
Standardwert: 512

GLBPS = zahl
legt die minimale Größe eines ISAM-Pools in einem Data-Space in PAM-Seiten fest. „zahl“ wird, wenn nötig, auf das nächste Vielfache von 512 gerundet und, wenn nötig, wieder auf den Maximalwert 32766 verringert.
Mögliche Werte: 512 <= zahl <= 32766
Standardwert: 512

LCLPS = zahl
legt den Standardwert für die Pool-Größe beim Anlegen von tasklokalen ISAM-Pools per Kommando oder Makro in PAM-Seiten fest.
Mögliche Werte: 32 <= zahl <= 8192
Standardwert: 512

MAXDSBN = zahl
definiert die Anzahl der maximal für taskübergreifende ISAM-Pools bereitzustellenden Data-Spaces.
Mogliche Werte: 1 <= zahl <= 127
Standardwert: 2

> Das Kommando SHOW-ISAM-CACHING informiert die Systembetreuung über die aktuell als ISAM-Caches verwendeten Data-Spaces, die zur Aufnahme von taskübergreifenden ISAM-Pools angelegt und verwaltet werden. Die Anzahl der verwendeten Data-Spaces kann für den laufenden Betrieb mit dem Kommando MODIFY-ISAM-CACHING geändert werden.

Hinweis zur Einstellung von MAXDSBN und GLBPS

Mit den Voreinstellungen MAXDSBN=2 und GLBPS=512 können max. ca. 4000 gleichzeitig im Modus SHARUPD=YES geöffnete NK-ISAM-Dateien in separaten ISAM-Pools gepuffert werden. Die darüber hinaus im Modus SHARUPD=YES geöffneten NK-ISAM-Dateien werden in den bereits existierenden ISAM-Pools gepuffert.

Wenn sichergestellt werden soll, dass (fast) jede im Modus SHARUPD=YES geöffnete NK-ISAM-Datei separat gepuffert wird, muss zunächst ermittelt werden, wieviele NK-ISAM-Dateien maximal gleichzeitig mit SHARUPD=YES geöffnet werden. Wenn die Anzahl gleichzeitig geöffneter Dateien (=#FILES) den Grenzwert von 4000 überschreitet, muss die Anzahl der benötigten Data Spaces angepasst werden (mit Rundung auf das nächstgrößere Vielfache von 2 GB):

```
MAXDSBN = (#FILES x GLBPS) / 1.000.000
```

Anmerkung: Da die Poolgröße in Einheiten zu 2 KB angegeben wird, muss die Größe eines Data Space in 1.000.000 x 2 KB (= 2 GB) ausgedrückt werden.

Der Seitenwechselbereich ist ggf. entsprechend zu erweitern, wobei der Bedarf für Data Space für taskübergreifende ISAM-Pools mit `(MAXDSBN + 1) x 2 GB` anzusetzen ist.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN ISAM
LCLDFPS=100
GLBPS=512
LCLPS=512
MAXDSBN=4
/EOF
:
/END-PARAMS
```

3.8 Festlegung des TSN-Modus (JMS)

Mit der JMS-Anweisung in der Parameterdatei kann die Systembetreuung festlegen, ob im folgenden Systemlauf ausschließlich numerische Zeichen oder auch alphanumerische Zeichen für die Stellen 2 bis 4 einer Benutzer-TSN vom System vergeben werden.
Soll der Modus in zwei aufeinander folgenden Systemläufen von alphanumerisch nach numerisch gewechselt werden, kann die Systemeinleitung nur mit einem „Kaltstart“ durchgeführt werden. Der Operator erhält in diesem Fall einen entsprechenden Hinweis auf der Konsole.

Schlüsselwort für die Festlegung des TSN-Modus ist **JMS**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Format des Parametersatzes zur Festlegung des TSN-Modus

TSN-MODE = A[LPHANUMERIC] / N[UMERIC]

TSN-MODE = ALPHANUMERIC
Es wird vereinbart, dass das System für die zweite bis vierte Stelle einer TSN auch alphanumerische Zeichen verwenden darf.
Wird in der Parameterdatei kein oder ein fehlerhafter Wert für die TSN-Vergabe gesetzt, dann wird dieser Modus für den Systemlauf wirksam.

TSN-MODE = NUMERIC
Es wird vereinbart, dass das System nur numerische Zeichen für die TSN einer Benutzertask vergeben darf. Folgt dieser Modus einem Systemlauf, in dem alphanumerische Zeichen vergeben wurden, kann die Systemeinleitung nur mit „Kaltstart“ durchgeführt werden.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN JMS
TSN-MODE=A                    1.
/EOF
:
/END-PARAMS
```

1. Für den folgenden Systemlauf darf das System für die TSN einer Benutzer-Task für die Stellen 2 bis 4 auch alphanumerische Zeichen verwenden.

3.9 Speicherverwaltung (MEMORY)

Die Konfiguration der von der Systemfunktion Memory Management verwalteten Betriebsmittel Virtuelle Adressräume, Hauptspeicher und Erweiterungsspeicher kann bei der Systemeinleitung über entsprechende Einträge in der Parameterdatei definiert werden.

> ¡ Die Parameter ES-UNIT, MEM-RECONF, HSDABSIZE und PAGING-GSSIZE werden nicht mehr ausgewertet. Ihre Angabe wird ignoriert.
> Der Parameter SYSSIZE ist nicht mehr erforderlich. Siehe hierzu "Hinweise zum Parameter SYSSIZE (Größe des Systemadressraumes auf SU /390)".

Schlüsselwort für den BEGIN-Satz des Parameterservice ist **MEMORY**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Format des Parametersatzes zur Speicherverwaltung

SHRSIZE n	
ASAMCTRL n	
ALAMCTRL n	
MEM-TEST n	*wird an SU x86 ignoriert*
SHXSIZE n	*wird an SU /390 ignoriert*
BIG-PAGE-QUOTA n	*wird an SU /390 ignoriert*
BIG-PAGE-SHRSIZE n	*wird an SU /390 ignoriert*
PAGING-SATURATION-WARNING-LIMIT n [,UNIT=4KB / MB]	
PAGING-SATURATION-LIMITS-QUOTA n	

SHRSIZE n
Größe des für Klasse-4-Speicher unterhalb der 16-MB-Grenze reservierten virtuellen Adressbereiches. Dieser Adressbereich wird für Shared-Code, DSSM-Subsysteme und TU-Logicals benötigt.

n ist die Größe des reservierten Bereichs in MB.
Die mit SHRSIZE reservierten Bereiche dürfen 8 MB nicht überschreiten.
Mögliche Werte: 1 <= n <= 8
Standardwert: n = 2

ASAMCTRL n
Dieser Parameter dient zur Einstellung der Test and Trace Facility (TTF), die in ASAM eingebaut ist. Er legt die Größe des Klasse-3-Arbeitsspeichers, der für die Trace-Ausgabe zugewiesen wird, in 4KB-Blöcken fest. Die TTF dient nur zur Fehlerdiagnose und sollte nur im Bedarfsfall eingeschaltet werden.

n ist die Größe des TTF-Bereichs in 4KB-Blöcken.
Mögliche Werte: 0 <= n <= 15
Standardwert: 0

ALAMCTRL n
Dieser Parameter dient zur Einstellung der Test and Trace Facility (TTF), die in ALAM eingebaut ist. Er legt die Größe des Klasse-3-Arbeitsspeichers, der für die Trace-Ausgabe zugewiesen wird, in 4KB-Blöcken fest. Die TTF dient nur zur Fehlerdiagnose und sollte nur im Bedarfsfall eingeschaltet werden.

n ist die Größe des TTF-Bereichs in 4KB-Blöcken.
Mögliche Werte: 0 <= n <= 15
Standardwert: 0

MEM-TEST n
Dieser Parameter gibt an, ob bzw. wann ein Hauptspeichertest durchgeführt werden soll.

n bezeichnet die Option für den Hauptspeichertest.
Mögliche Werte für n:

0 Beim Systemstart und während des Systemlaufs erfolgt kein Speichertest.

1 Der Speichertest soll beim Systemstart durchgeführt werden. Bei großen Arbeitsspeichern kann dies den Systemstart verzögern.

2 Vor jeder Neuvergabe einer Hauptspeicherseite an eine virtuelle Seite soll die Hauptspeicherseite getestet werden. Diese Option kann sich negativ auf die Systemperformance auswirken.

Standardwert: n = 0

Der Parameter wird an SU x86 ignoriert.

SHXSIZE n
Größe des Klasse-4-Speichers, der im Bereich oberhalb 16 MB und unterhalb 2GB angelegt wird. Dieser Speicherbereich wird von BLS und DSSM zum Laden von Shared Programmen und nichtprivilegierten Subsystemen, die im x86-64-Code vorliegen, genutzt. Der Bereich ist in allen Adressräumen eingebettet und kann im nichtprivilegierten Modus gelesen, aber nicht beschrieben werden.

n ist die Größe des Bereichs in MB und muss ein Vielfaches von 16 sein.
Mögliche Werte für n: 16, 32, 48, ..., 128
Standardwert: n = 64

Werden andere Werte eingegeben, wird die Systemeinleitung abgebrochen.

Der Parameter wird an SU /390 ignoriert.

BIG-PAGE-QUOTA n
Mit diesem Parameter wird die Zielgröße des Anteils am Hauptspeicher, der für „Big Pages“ reserviert sein soll, festgelegt. Eine Big Page hat eine Größe von 4 MB und wird für CISCFW-Kompilate genutzt.

n ist die Größe des für Big Pages vorgesehenen Hauptspeicher in %.
Mögliche Werte: 1 <= n <= 99
Standardwert: n = 40

Der Parameter wird an SU /390 ignoriert.

> **i** Es ist nicht immer möglich, den angegebenen Prozentsatz in Big Pages anzulegen. Dies hängt von verschiedenen Randbedingungen ab (siehe Abschnitt „Big Pages für CISCFW-Kompilate (SU x86)").

BIG-PAGE-SHRSIZE n
Mit diesem Parameter wird die Größe der „Shared Big Pages“ festgelegt, die im Shared-Memory (Klasse 3) angelegt und für CISCFW-Kompilate von Shared-Programmen genutzt werden.

n ist die Größe der Shared Big Pages in MB und muss ein Vielfaches von 16 sein.
Mögliche Werte: 0 <= n
Standardwert: n = 64

Der Parameter wird an SU /390 ignoriert.

PAGING-SATURATION-WARNING-LIMIT n [,UNIT=4KB / MB]
Mit diesem Parameter wird ein Schwellwert für die Anzahl der nicht reservierten Seiten im Seitenwechselspeicher festgelegt. Wird dieser Schwellwert unterschritten, dann werden zur Warnung folgende Meldungen ausgegeben:

```
EXC0873 NEW PAGING MEMORY SATURATION LEVEL = 1
EXC0873 NEW PAGING MEMORY SATURATION LEVEL = 0
```

Diese Meldungen warnen dann vor einem Sättigungszustand Grad 2 (Schwellwert 256), der sonst ohne diese Warnungen erreicht werden würde.

Informationen zu den Sättigungszuständen finden Sie im Abschnitt „Seitenwechselspeicher".

n ist die Anzahl der nicht reservierten Seiten im Seitenwechselspeicher.
Mögliche Werte: 256 <= n <= 4.194.304 falls UNIT=4KB
Mögliche Werte: 1 <= n <= 16.384 falls UNIT=MB
Standardwert: n = 256, UNIT = 4KB

PAGING-SATURATION-LIMITS-QUOTA n
Mit diesem Parameter kann ein Prozentwert größer oder gleich 100 angegeben werden. Die Schwellwerte für die Sättigungszustände Grad 2 und 3 im Seitenwechselspeicher (bei Speichersättigung) sowie für die Sättigungszustände Grad 1 und 0 (bei Rückgang der Sättigung) werden mit diesem Prozentwert multipliziert. Dadurch lässt sich erreichen, dass die Sättigungszustände Grad 2 und 3 früher gemeldet werden als bei den Standard-Einstellungen. Die Sättigungszustände Grad 1 und 0 werden dann entsprechend später gemeldet. Alle Schwellwerte beziehen sich auf die Anzahl der nicht reservierten Seiten im Seitenwechselspeicher.

Informationen zu den Sättigungszuständen finden Sie im Abschnitt „Seitenwechselspeicher".

n ist der Prozentwert für die Multiplikation der Schwellwerte.
Mögliche Werte: 100 <= n <= 1.638.500 (in Prozent)
Standardwert: n = 100

Hinweise zum Parameter SYSSIZE (Größe des Systemadressraumes auf SU /390)

Der Parameter ist nicht mehr erforderlich, da der für den Systemadressraum zur Verfügung stehende Bereich sich aus der Größe des Benutzeradressraums und der sich daraus ableitenden Größe des Gesamtadressraums errechnet.

Ist trotzdem ein Wert für SYSSIZE im MEMORY-Parametersatz angegeben, so wird er auf Konsistenz mit dem errechneten Wert für SYSSIZE überprüft.

Mögliche Werte: 240, 256, ..., 512 MB (Vielfaches von 16).
Wurden andere Werte eingegeben, wird die Systemeinleitung abgebrochen.

- Ist ein Wert für SYSSIZE angegeben und kleiner als der errechnete Wert, wird der angegebene Wert ignoriert und mit dem errechneten Wert gearbeitet.

- Ist ein Wert für SYSSIZE angegeben und der sich ergebende Wert für den Gesamtadressraum ist größer als 2 GB, wird die Systemeinleitung mit einer entsprechenden Meldung abgebrochen.
- Ist ein Wert für SYSSIZE angegeben und hat eine Verdopplung des Gesamtadressraums zur Folge, wird die Systemeinleitung mit dem erhöhten Wert für den Gesamtadressraum fortgesetzt. Es wird eine entsprechende Meldung ausgegeben, die auf die Inkonsistenz zwischen Generierung und Parameterdatei hinweist.

3.10 Konfiguration und Meldungsunterdrückung an Konsolen (OPR)

Die Funktion OPR-Tabellen-Parametrisierung gibt der Systembetreuung die Möglichkeit, die Konsolkonfiguration, die durch die Tabellen des Bediensystems logisch repräsentiert wird, erst während der Systemeinleitung zu ändern.

Die Änderungen betreffen folgende Punkte:

- Aufnahme von Operatorkommandos in die Operatorkommando-Tabelle
- Änderung der Konsoleigenschaften
- Änderung der Berechtigungsschlüssel für Operatorkommandos
- Zuordnung von Berechtigungsschlüsseln zu Konsolen
- Zuordnung von Filterstufen zu Konsolen (gilt nicht für berechtigte Anwendungen und im Fall von NBCONOPI=Y nur bis „System Ready“ und nach Shutdown)
- Unterdrückung bestimmter Meldungen an Konsolen (gilt nicht für berechtigte Anwendungen und im Fall von NBCONOPI=Y nur bis „System Ready“ und nach Shutdown)

Zum Zeitpunkt der Eingabe eines Operatorkommandos muss dieses, um ausgeführt werden zu können, dem Subsystem OPR bekannt sein. Zu diesem Zweck wird eine Kommandotabelle im Speicher aufgebaut, in der für jedes Kommando ein Eintrag enthalten ist. Dieser Eintrag besteht aus:

- Name des Kommandos
- Aliasnamen (Abkürzungen) des Kommandos
- Notwendige Berechtigung zur Kommandoausführung
- Identifikation des Kommando-Servers
- Kennwort-Indikator

Die Systembetreuung hat durch die dynamische Anpassung bzw. Erweiterung im Parameterservice (Anweisungen ADD-CMD-ENTRY, SET-CMD-CODE) die Möglichkeit, neue Kommandos in die Operatorkommando-Tabelle einzubringen bzw. Berechtigungsschlüssel für bestehende Kommandos zu ändern.

Semantische Fehler innerhalb eines Parametersatzes, die von der Fehlerroutine des Startup nicht erkannt und im Fehlerdialog behoben werden könnten, werden an der Konsole gemeldet.

Schlüsselwort für die Änderung oben genannter Punkte in der Parameterdatei ist **OPR**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 384, dabei dürfen die einzelnen Parametersätze jeweils in einer maximalen Anzahl vertreten sein (siehe folgende Tabelle).

Anweisung	**Bedeutung**	**max. Anzahl**
ADD-CMD-ENTRY	Operatorkommando in die Kommando-Tabelle aufnehmen	100
DEFINE-CONSOLE	Konsolen definieren oder ändern, Ersatz-Konsolen zuweisen	24
SET-CMD-CODE	Berechtigungsschlüssel eines Operatorkommandos ändern	100
SET-CODE	Berechtigungsschlüssel vergeben	165
SET-FILTER	Filterstufen festlegen	40
SET-MSG-SUPPRESSION	Ausgabe von Meldungen unterdrücken	50

Tabelle 5: Übersicht über die Anweisungen zum Parametersatz OPR

Ist das Operator-Logon eingeschaltet (NBCONOPI=Y), wird die Bedienung an physikalischen Konsolen und an $CONSOLE-Anwendungen mit dynamischen Berechtigungsnamen in einem (im Sinne von Datensicherheit und Datenschutz) sicheren Modus durchgeführt.

Bei NBCONOPI=Y ist, ebenso wie bei NBCONOPI=N, in den Anweisungen SET-FILTER und SET-MSG-SUPPRESSION die Zuweisung an physikalische Konsolen durch die Angabe CONSOLE=*IPL möglich. Zuweisungen über mnemotechnische Gerätenamen – auch über die MN der IPL-Konsole – werden dagegen ignoriert.

Wurden Angaben in den Anweisungen SET-CODE, SET-FILTER und SET-MSG-SUPPRESSION ignoriert, erscheint folgende Meldung:

```
NBR1202 '(&00)' PARAM SETS FOR CONSOLES ONLY ALLOWED FOR *IPL
```

Der nachfolgende Ausschnitt aus der Parameterdatei zeigt die für die Änderung der Konsolkonfiguration relevanten Anweisungen bei NBCONOPI=N :

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN OPR
DEFINE-CONSOLE CONSOLE=C1,REPLACEMENT=C2                              1.
SET-CODE CODE=A,CONSOLE=(C3,C4)                                       2.
SET-FILTER 3,CODE=*ALL,CONSOLE=*IPL                                   3.
SET-FILTER *ALL,CODE=*ALL,CONSOLE=C5                                  4.
SET-MSG-SUPPRESSION (MSG0001,MSG0002),*ALL                            5.
/EOF
:
/END-PARAMS
```

1. Der Konsole C1 wird die Ersatz-Konsole C2 zugewiesen.
   War C1 schon durch die Hardware-Generierung vergeben, werden die Eigenschaften von C1 damit geändert. Wurde C1 noch nicht in den Konsoltabellen geführt, wird ein Platzhalter-Eintrag für C1 mit den angegebenen Eigenschaften belegt.
   Der REPLACEMENT-Operand ist bei NBCONOPI=N von Bedeutung.
   Bei eingeschaltetem Operator-Logon wird er nicht ausgewertet.
2. Den Konsolen C3 und C4 wird der Berechtigungsschlüssel A zugewiesen.
3. Für alle der IPL-Konsole zugewiesenen Berechtigungsschlüssel wird die Filterstufe 3 gesetzt. Dies bewirkt, dass alle über Berechtigungsschlüssel verteilten Meldungen mit einem Meldungsgewicht 40-59 an der IPL-Konsole nicht ausgegeben werden.
4. Für alle der Konsole C5 zugewiesenen Berechtigungsschlüssel werden die Filterstufen 1, 2, 3, 4 und 5 gesetzt. Dies bewirkt, dass alle über Berechtigungsschlüssel verteilten Meldungen unabhängig vom Meldungsgewicht (Fragen ausgenommen) an der Konsole C5 nicht ausgegeben werden. Diese Einstellung entspricht derjenigen, die mit dem Kommando ASR NOINF an der Konsole C5 vorgenommen werden könnte.
   Die explizite Angabe eines mnemotechnischen Gerätenamens ist nur bei NBCONOPI=N möglich.
5. Die Meldungen mit den Meldungsnummern `MSG0001` und `MSG0002` werden an allen in der Konsoltabelle geführten Konsolen unterdrückt.
   Der Meldungsunterdrückungsmechanismus gilt bei eingeschaltetem Operator-Logon nur bis „System Ready“.

3.10.1 Anweisung ADD-CMD-ENTRY

Format der Anweisung zum Aufnehmen von Operator-Kommandos in die Kommando-Tabelle

Anweisung	Operanden	Bedeutung
ADD-CMD-ENTRY	AUTHORIZATION-CODE = bs	Berechtigungsschloss für das Kommando
	,CMD-NAME = kmd	Langform des Kommandonamens
	,SAME-NAME = *NONE / kmd1..3	Vergabe von Aliasnamen
	,APPLICATION-NAME = name	Vereinbarung der zuständigen $CONSOLE-Anwendung
	,PASSWORD-POSSIBLE = NO / YES	Vereinbarung der Kennwortprotokollierung

ℹ Die Funktionalität von ADD-CMD-ENTRY kann von berechtigten Benutzerprogrammen auch mit dem Kommando CONNECT-CMD-SERVER abgedeckt werden.

AUTHORIZATION-CODE = bs
Berechtigungsschloss des Kommandos (ein Zeichen aus der Menge A-Z, 0-9, *, #, @, $). Mit diesem Berechtigungsschloss wird festgelegt, über welchen passenden Schlüssel (in Form des gleichlautenden Zeichens, auch Routing-Code genannt) ein Kommandogeber verfügen muss, um das Kommando nutzen zu können. Für Sonderfunktionen der Berechtigungsschlüssel @, $ und * siehe Tabelle 45, "Aufgabengebiete und ihre Aufteilung auf Konsolen".

CMD-NAME = kmd
Kommandoname (Original- bzw. Langform), für den ein Eintrag in der Operatorkommando-Tabelle erzeugt werden soll. Der Name kann bis zu 30 Zeichen lang sein und muss den Konventionen für Kommandonamen entsprechen (in SDF-Syntax: <structured-name 1..30>).

SAME-NAME =
Legt fest, ob und welche Aliasnamen (d.h. zulässige Abkürzungen) für das Kommando definiert werden sollen.

SAME-NAME = *NONE
Voreinstellung: Es sollen keine Aliasnamen vereinbart werden.

SAME-NAME = kmd1..3
Menge der Aliasnamen (max. 3), die mit dem Kommandonamen gleichzusetzen sind.

APPLICATION-NAME = name
Name der kommandoausführenden $CONSOLE-Anwendung. Sofern noch nicht geschehen, wird die logische Konsole mit diesem Berechtigungsnamen automatisch intern generiert. Der Anwendungsname muss aus vier alphanumerischen Zeichen aus der Menge A-Z, 0-9, @, $, # bestehen, wobei das erste Zeichen keine Ziffer und nicht „#“ sein darf. Das erste Zeichen sollte ferner nicht „@“ sein, da dieses Zeichen für dynamische Berechtigungsnamen verwendet wird.

PASSWORD-POSSIBLE =
Vereinbart, ob das zu definierende Kommando möglicherweise ein Kennwort beinhaltet (z.B. Angabe eines Dateikennwortes in einem Operanden).
Ist ein solches Kennwort Bestandteil der Kommando-Zeichenfolge, kann mittels dieses Operanden verhindert werden, dass es in der System-Loggingdatei CONSLOG protokolliert wird, falls ein entsprechender SDF-Eintrag existiert.

PASSWORD-POSSIBLE = NO
Voreinstellung: Die gesamte Kommando-Zeichenfolge enthält kein Kennwort.
Vor dem Eintrag in die CONSLOG-Datei soll die Zeichenfolge also nicht nach einem Kennwort durchsucht werden.

PASSWORD-POSSIBLE = YES
Die gesamte Kommando-Zeichenfolge kann Kennwörter enthalten. Um die Protokollierung dieses Kennwortes in der CONSLOG-Datei zu verhindern, wird die Systemtask CLOG zuständig für die CONSLOG-Datei) „befugt“, SDF aufzurufen und das Kennwort vor Aufnahme in die CONSLOG-Datei zu eliminieren.

> In den ADD-CMD-ENTRY- und SET-CODE-Anweisungen können zusammen maximal 384 verschiedene logische Konsolen genannt werden. Bei Angabe von mehr als 384 logischen Konsolen werden nur die ersten 384 akzeptiert.
> Die Länge einer Parameteranweisung ist generell auf 72 Zeichen beschränkt.
> Es empfiehlt sich, wie folgende Beispiele zeigen, die Schlüsselwörter in einer Anweisung entweder stark abzukürzen oder ganz wegzulassen.
> Der SAME-NAME muss dabei immer kürzer als der CMD-NAME sein.

Beispiele

```
---------1---------2---------3---------4---------5---------6---------7--
ADD-CMD-ENTRY A-C=E,CMD-NAME=SPEC-CMD,SAME-NAME=SPEC,APPL-NAM=CON7,P-P=Y
A-C-E F,SPECIAL-COMMAND-FILTER-KDOXYZ,SPECIAL-COMMAND-FILT-KXYZ,CON7,Y
```

3.10.2 Anweisung DEFINE-CONSOLE

Format der Anweisung zur Definition oder Änderung von Konsolen und Zuweisung von Ersatz-Konsolen

Anweisung	Operanden	Bedeutung
DEFINE-CONSOLE	CONSOLE = name	Mnemotechnischer Gerätename der Konsole
	,REPLACEMENT = STD /name	Mnemotechnischer Gerätename einer Ersatz-Konsole
	,TELESERVICE = NO / YES	Teleservice-Eigenschaft

CONSOLE = name
Mnemotechnischer Gerätename (2 Zeichen) der Konsole, für die die Zuweisungen gelten sollen. Der Name bezieht sich auf eine bereits definierte Konsole.

REPLACEMENT =
Mnemotechnischer Gerätename einer Konsole, die als Ersatz-Konsole verwendet werden soll. Die Angabe des Operanden wird bei NBCONOPI=Y ignoriert. Es wird folgende Meldung ausgegeben:

```
NBR1203 OPERAND 'REPLACEMENT' IN 'DEFINE-CONSOLE' PARAMETER SET IGNORED
```

REPLACEMENT = STD
Voreinstellung: Die Zuweisung der Ersatz-Konsole erfolgt durch die UCON-Task.

REPLACEMENT = name
Mnemotechnischer Gerätename einer durch die Hardware-Generierung oder durch einen Parametersatz definierten Konsole, die als Ersatz-Konsole verwendet werden soll. Gibt es die angegebene Konsole nicht, erfolgt die Zuweisung der Ersatz-Konsole durch die UCON-Task.

TELESERVICE =
Vereinbarung der Teleservice-Eigenschaft.

TELESERVICE = NO
Voreinstellung: Die Konsole soll keine Teleservice-Eigenschaft besitzen.

TELESERVICE = YES
Die Konsole soll Teleservice-Eigenschaft besitzen und kann nicht mehr Ersatz-Konsole einer anderen Konsole und nicht mehr Hauptkonsole werden.

3.10.3 Anweisung SET-CMD-CODE

Format der Anweisung zum Ändern des Berechtigungsschlosses eines Operatorkommandos

Anweisung	Operanden	Bedeutung
SET-CMD-CODE	AUTHORIZATION-CODE = bs	Berechtigungsschloss für das Kommando
	,CMD-NAME = kmd	Langform des Kommandonamens

AUTHORIZATION-CODE = bs
Neues Berechtigungsschloss des Kommandos (ein Zeichen aus der Menge A-Z, 0-9, *, #, @ oder $).
Mit diesem Berechtigungsschloss wird festgelegt, über welchen passenden Schlüssel
(in Form des gleichlautenden Zeichens, auch Routing-Code genannt) ein Kommandogeber verfügen muss, um das Kommando nutzen zu können. Für Sonderfunktionen der Berechtigungsschlüssel @ und $ siehe Tabelle 45, "Aufgabengebiete und ihre Aufteilung auf Konsolen".

CMD-NAME = kmd
Kommandoname (Original- bzw. Langform), für den der Eintrag in der Operatorkommando-Tabelle geändert werden soll.
Der Name kann bis zu 30 Zeichen lang sein und muss den Konventionen für Kommandonamen entsprechen (in SDF-Syntax: <structured-name 1..30>).
Das zugeordnete Berechtigungsschloss (AUTHORIZATION-CODE) wirkt auch auf die dem Kommando zugeordneten Aliasnamen. Dies gilt auch für die mit der Anweisung ADD-CMD-ENTRY angemeldeten sog. Operator-Spezialkommandos.

3.10.4 Anweisung SET-CODE

Format der Anweisung zur Vergabe von Berechtigungsschlüsseln

Anweisung	Operanden	Bedeutung
SET-CODE	CODE = bs / *ALL	(alle) Berechtigungsschlüssel
	,CONSOLE = *IPL / name / (name,...)	Mnemotechnischer Name der Konsole

CODE = bs / *ALL
Berechtigungsschlüssel (A..Z, 0..9, *,#,@,$), der einer physikalischen Konsole oder einer logischen Konsole mit generiertem Berechtigungsnamen zugeordnet wird.
Es können auch alle Berechtigungsschlüssel (*ALL) zugeordnet werden.
Die Schlüssel 0, *, $ und @ haben eine Sonderbedeutung (siehe "Berechtigungsschlüssel mit Sonderbedeutungen", "Aufgabengebiete und ihre Aufteilung auf Konsolen").

CONSOLE = name / (name,...,name) / *IPL
Mnemotechnischer Name von logischen Konsolen mit generierten Berechtigungsnamen (4 alphanumerische Zeichen) oder von physikalischen Konsolen (2 alphanumerische Zeichen), für die die Zuweisungen gelten sollen.
Die Konsole, von der das System geladen wird (IPL-Konsole), kann auch mit dem Wert *IPL anstatt ihres Namens angesprochen werden.
Bei Angabe eines 4-stelligen Namens wird, sofern noch nicht geschehen, automatisch eine logische Konsole mit diesem Berechtigungsnamen generiert.
Das erste Zeichen des Berechtigungsnamens darf weder eine Ziffer noch das Zeichen „#“ sein, und es sollte auch nicht „@“ sein. Es empfiehlt sich jedoch, in den vier alphanumerischen Zeichen immer auch Sonderzeichen (#, @ oder $) anzugeben, um eine eindeutige Unterscheidung gegenüber Auftragsnummern (TSN) zu erreichen.

Eine logische Konsole wird mit der Zuweisung eines Berechtigungsschlüssels berechtigt, Operator-Kommandos einzugeben sowie Meldungen zu empfangen, die diesem Schlüssel zugeordnet sind.

Für physikalische Konsolen hängt die Wirkung der Zuweisung vom Systemparameter NBCONOPI ab.

- NBCONOPI=N

  Auch physikalische Konsolen werden wird mit der Zuweisung eines Berechtigungsschlüssels berechtigt, Operator-Kommandos einzugeben sowie Meldungen zu empfangen, die diesem Schlüssel zugeordnet sind. Es gibt keinen Unterschied zu logischen Konsolen mit generiertem Berechtigungsnamen.

- NBCONOPI=Y

  Für physikalische Konsolen werden explizite und implizite Berechtigungsschlüssel (Routing-Codes) unterschieden.

  Nur ein mit dem Kommando REQUEST-OPERATOR-ROLE zugewiesener Berechtigungsschlüssel berechtigt sowohl zur Kommandoeingabe als auch zum Meldungsempfang („expliziter Routing-Code“).

  Mit SET-CODE zugewiesene Berechtigungsschlüssel sind „implizite Routing-Codes“, die lediglich zum Empfang (und ggf. Beantworten) von Meldungen berechtigen, nicht aber zur Eingabe von Kommandos. Sie enthalten auch den Routing-Code * und können ausschließlich der IPL-Konsole (Wert *IPL) zugewiesen werden. Eine Zuweisung an Konsolnamen wird abgewiesen (Meldung `NBR1202`). Während des Systemlaufs werden sie bei Bedarf automatisch von der IPL-Konsole an die jeweils aktive Hauptkonsole übertragen.

  Die Wirkungsdauer von impliziten Routing-Codes hängt vom Systemparameter NBIMPRCA ab (siehe auch Abschnitt „Zuordnung von Aufgabengebieten zu Konsolen“, "Aufgabengebiete und ihre Aufteilung auf Konsolen ").

> **i** In den ADD-CMD-ENTRY- und SET-CODE-Anweisungen können zusammen maximal 384 verschiedene logische Konsolen genannt werden. Bei Angabe von mehr als 384 logischen Konsolen werden nur die ersten 384 akzeptiert.

3.10.5 Anweisung SET-FILTER

Format der Anweisung zur Festlegung von Filterstufen

Anweisung	Operanden	Bedeutung
SET-FILTER	FILTER = *ALL / zahl / (zahl,...)	Filterstufe
	,CODE = *ALL / bs / (bs,...)	Berechtigungsschlüssel
	,CONSOLE = *IPL / name / (name,...)	Mnemotechnischer Name der Konsole

Diese Anweisung wird bei NBCONOPI=Y nur für die IPL-Konsole zugelassen.
Die eingestellten Filterstufen wirken dann nur bis „System Ready“ und nach Shutdown.

FILTER = *ALL
Alle Filterstufen werden in den angegebenen Berechtigungsschlüsseln gesetzt. Damit werden die über die im CODE-Operanden angegebenen Berechtigungsschlüssel verteilten Meldungen unabhängig vom Meldungsgewicht an der IPL-Konsole unterdrückt.

FILTER = (zahl,...,zahl)
Vereinbarung einer Filterstufe (Ziffer, Wertebereich 1...5), die zur Unterdrückung der über Berechtigungsschlüssel und Gewicht versandten Meldungen herangezogen wird.

CODE = *ALL
Der Filter soll für alle Berechtigungsschlüssel gesetzt werden.

CODE = (bs,...,bs)
Berechtigungsschlüssel (A..Z, 0..9, oder *,#,@,$), für die der Filter gesetzt wird.
Das Setzen von Filterstufen für den Berechtigungsschlüssel @ hat auf die Meldungsausgabe an Konsolen keinen Einfluss.

CONSOLE = *IPL
Die angegebenen Filter sollen für die Konsole gesetzt werden, von der das System geladen wird.
Bei NBCONOPI=Y wird dieser Wert aus Kompatibilitätsgründen nicht abgewiesen.

CONSOLE = (name,,...name)
Mnemotechnischer Gerätename (2 Zeichen) von Konsolen, an denen der Filter gesetzt werden soll.
Bei NBCONOPI=Y wird dieser Wert ignoriert.

3.10.6 Anweisung SET-MSG-SUPPRESSION

Format des Parametersatzes zur Unterdrückung von Meldungsausgaben

Anweisung	Operanden	Bedeutung
SET-MSG-SUPPRESSION	MSG-ID = msg-id / (msg-id, ...)	Meldungsnummer
	,CONSOLE = *ALL / *IPL / name / (name, ...)	Mnemotechnischer Gerätename der Konsole
	STARTUP-TYPE = ANY / DIALOG / NON-DIALOG	Abhängigkeit vom Startup-Typ

Diese Anweisung wird bei NBCONOPI=Y nur für die IPL-Konsole zugelassen.
Die Meldungsunterdrückung wirkt dann nur bis „System Ready“.

MSG-ID = (msg-id, ...)
Angabe von 7-stelligen Meldungsnummern, deren Ausgabe auf der angegebenen Konsole unterdrückt werden soll.

> Maximal 50 Meldungsnummern können unterdrückt werden. Wenn weitere Meldungsnummern unterdrückt werden sollen, dann ist dies erst nach SYSTEM READY mit dem gleichnamigen Kommando möglich (z.B. in der CMDFILE).

CONSOLE = *ALL
Die vereinbarten Meldungen sollen an allen Konsolen unterdrückt werden.
Bei NBCONOPI=Y wird dieser Wert ignoriert.

CONSOLE = *IPL
Die vereinbarten Meldungen sollen an der Konsole unterdrückt werden, von der das System geladen wird.
Bei NBCONOPI=Y wird dieser Wert aus Kompatibilitätsgründen nicht abgewiesen.

CONSOLE = name / (name, ...)
Mnemotechnischer Gerätename (2 Zeichen) von Konsolen, an denen die Meldung unterdrückt werden soll.

STARTUP-TYPE = ANY
Die Meldungen sollen unabhängig vom Startup-Typ unterdrückt werden.

STARTUP-TYPE = DIALOG
Die Meldungen sollen nur bei DIALOG-Startup unterdrückt werden.

STARTUP-TYPE = NON-DIALOG
Die Meldungen sollen nur bei Nicht-DIALOG-Startup unterdrückt werden, also im FAST-oder AUTOMATIC-Startup.

3.11 Auswahl des Seitenwechselbereichs beim Startup (PAGING)

Die Systembetreuung kann über den Parameterservice die Datenträger bestimmen, die fürs Paging benutzt werden sollen. Dabei ist zu beachten, dass auf diesen Datenträgern Seitenwechseldateien (Paging-Dateien) mit dem Namen SYS.PAGING.<vsn> angelegt sind und dass im Startup nicht die max. mögliche Anzahl Paging-Dateien ausgenutzt werden kann (s.u.).

Der von Startup initialisierte Seitenwechselbereich muss eine Mindestgröße von 50 MB haben, sonst wird die Systemeinleitung mit der Meldung NSI5225 abgebrochen.
Ist der Seitenwechselbereich größer als 50 MB, aber kleiner als 200 MB, wird als Warnung die Meldung NSI5115 an der Konsole ausgegeben; die Systemeinleitung wird fortgesetzt.

Durch geeignete Auswahl der zum Seitenwechsel zu verwendenden Paging-Dateien kann die Seitenwechselaktivität bereits zum Startup-Zeitpunkt auf weniger belastete Datenträger gelenkt werden.
Es müssen nicht alle für Paging vorgesehene Datenträger zum Startup-Zeitpunkt verfügbar und verwendbar sein, da auch während des Systemlaufs zusätzliche Paging-Dateien in Betrieb genommen werden können.

Hat die Systembetreuung keine Parameter für die Auswahl von Paging-Dateien angegeben, dann werden automatisch alle auf dem Home-Pubset eingerichteten Paging-Dateien SYS.PAGING.<vsn> für den Seitenwechsel verwendet.

Mit dem Parameterservice können Datenträger mit Paging-Dateien innerhalb und außerhalb des Home-Pubsets angegeben werden.

Einschränkungen

- Zum Startup-Zeitpunkt muss mindestens eine Paging-Platte spezifiziert werden oder (ohne Parameterangaben) mindestens eine Paging-Datei auf dem Home-Pubset eingerichtet sein.
- Es dürfen maximal 128 Paging-Platten zum Startup-Zeitpunkt spezifiziert werden.
- Jede Paging-Datei SYS.PAGING.<vsn> kann aus mehreren Extents bestehen, die aber alle auf derselben Platte liegen müssen. Zusätzliche Extents verringern jedoch die Maximalzahl verwendbarer Paging-Dateien: Hat jede Datei z.B. 2 Extents, fasst die Startup-Tabelle nur noch 102 statt 128 Paging-Dateien; sind es 64 Dateien, dürfen diese max. 320 Extents haben.
- Die im Startup angegebenen Paging-Dateien dürfen sich auf max. 16 Pubsets verteilen.
- Kann keine Paging-Datei für eine spezifizierte Platte ermittelt werden, wird diese Platte ignoriert (Meldung NSI5110).
- Es können keine Privatplatten für Seitenwechsel verwendet werden.
- Paging-Pubsets müssen vollständig und exklusiv zur Verfügung stehen, dürfen also z.B. keine „shared Pubsets“ sein. Sie werden für den einzuleitenden Systemlauf vollständig und exklusiv von der Startup-Paging-Initialisierung belegt.
  Bei einem SM-Pubset wird nur der Volume-Set, der die Paging-Platten enthält, auf Fremdbelegungen überprüft.
- Die in der IPL-CONF abgespeicherte Gerätekonfiguration der Systemeinleitung wird bei neu angegebenen Paging-Platten automatisch um diese erweitert.

Schlüsselwort für den BEGIN-Satz des Parameterservice ist **PAGING**.

Die Maximalzahl dafür zugelassener Parametersätze beträgt 16. Überzählige Angaben werden ignoriert (Meldung NSI0049); die Systemeinleitung wird fortgesetzt. Wird während der Systemeinleitung in der PAGING-Anweisung ein Syntaxfehler diagnostiziert, wird sie – ohne Angabe eines Grundes – komplett abgewiesen.
An der Konsole kann die Anweisung entweder mit korrekten Werten wiederholt oder ignoriert werden.

Format des Parametersatzes zur Auswahl des Seitenwechselbereichs (Paging-Area) zum Startup-Zeitpunkt

Anweisung	Operand
PAGING	VOLUME = vsn / (vsn,...,vsn)

VOLUME =
Angabe der Datenträger, die für den Seitenwechsel verwendet werden sollen. Die Datenträgerkennung VSN ist immer 6 Zeichen lang.

Folgende formale Regeln sind zu beachten:

- Mehrere identische Angaben werden nur einmal berücksichtigt.
- Es können max. 128 VSN-Angaben verarbeitet werden (16 Parametersätze mit je max. 8 VSNs). Überzählige Angaben werden ignoriert (Meldung `NSI0049`); die Systemeinleitung wird fortgesetzt.
- Werden alle angegebenen Paging-Anweisungen wegen Syntaxfehlern ignoriert, dann werden automatisch alle auf dem Home-Pubset eingerichteten Paging-Dateien `SYS.PAGING.<vsn>` für den Seitenwechsel verwendet.
- Für Seitenwechsel können in einem BS2000-Systemlauf grundsätzlich verschiedene Plattentypen kombiniert und gleichzeitig genutzt werden (im Hinblick auf Performance siehe auch „Performance Handbuch“ [37]).

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN PAGING
PAGING VOLUME=(PUBA01,PUBA02,PUBB01,PUBB02,PUBB03)                    1.
PAGING VOLUME=ABC.01                                                  2.
/EOF
:
/END-PARAMS
```

1. Die Paging-Dateien (`SYS.PAGING.<vsn>`) auf den angegebenen Datenträgern der Pubsets mit den Katalogkennungen A und B werden Bestandteil des Seitenwechselbereichs (der Paging-Area) bei Startup.
2. Auch die Paging-Datei auf dem angegebenen Datenträger des Pubsets mit der Katalogkennung ABC soll bei Startup zum Seitenwechselbereich zählen.

3.12 Snapshot-Initialisierung (SNAP)

SNAP sichert unter der Steuerung des eigenständigen, BS2000-unabhängigen SNAP-EXECs BS2000-spezifische Diagnoseinformationen (siehe die Beschreibung der SNAP-Funktion im „Diagnosehandbuch“ [14]). Dabei werden BS2000 und dessen Anwendungen für max. 24 Sekunden (siehe Systemparameter SNAPTIME) angehalten.

Die Aktivierung der SNAP-Funktion erfolgt automatisch beim Startup von BS2000, wenn der Parameter SNAP-ACTIVE-SWITCH des SNAP-Parametersatzes nicht auf „OFF“ steht.

Schlüsselwort für den Snapshot-Parametersatz ist **SNAP**.
Es gibt für SNAP nur einen Parameter.

Wird ein ungültiger Wert im Parametersatz angegeben, wird der SNAP implizit ausgeschaltet (OFF) und eine Fehlermeldung beim Startup von BS2000 ausgegeben.
Fehlt der Parameter SNAP-ACTIVE-SWITCH ganz, tritt die Voreinstellung (ON) in Kraft.

Format des Parametersatzes zur Snapshot-Initialisierung

SNAP-ACTIVE-SWITCH = ON / ON*blank* *) / OFF

*) *blank* = 1 Leerzeichen

SNAP-ACTIVE-SWITCH = ON / ON*blank*
Voreinstellung: Die SNAP-Funktion soll für die laufende Session eingeschaltet werden.

Wenn die Systemdatei $TSOS.SNAPFILE nicht vorhanden ist, dann wird sie beim Startup automatisch (in Standardgröße) angelegt.

Wenn die SNAPFILE-Datei noch von einem SNAP-Dump der vorherigen Session beschrieben ist (Previous-SNAP), dann wird dieser SNAP-Dump in der neuen Session umkopiert, d.h. mit Indexstruktur versehen und in einer neuen Datei unter der Kennung SYSSNAP bereitgestellt.

SNAP-ACTIVE-SWITCH = OFF
SNAP wird bei der Systemeinleitung nicht initialisiert. Er steht für diese Session zunächst nicht zur Verfügung. SNAP-Aufrufe werden mit einem entsprechenden Returncode beendet. SNAP kann später dynamisch durch das Kommando ACTIVATE-SNAPSHOT aktiviert werden. Ein Previous-SNAP bleibt in der neuen Session unverändert erhalten.

Die Parameterwerte ON und OFF können auch von Hochkommas eingeschlossen werden ('ON' / 'ON ' / 'OFF').

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN SNAP
SNAP-ACTIVE-SWITCH=ON
/EOF
:
/END-PARAMS
```

3.13 Voreinstellung der Systemparameter (SYSOPT-CLASS2)

Es können nach dem Laden und der Rep-Verarbeitung des Klasse-2-Exec die Klasse-2-Systemparameter (kurz: Systemparameter) über den Startup-Parameterservice vorgegeben werden. Damit kann die Systembetreuung während der Systemeinleitung flexibel auf veränderte Vorgaben und Zielsetzungen reagieren.

Folgende Einschränkungen müssen berücksichtigt werden:

- Das Löschen oder Hinzufügen von Systemparametern ist nicht zulässig.
- Das Ändern bestimmter Attribute der Optionen (z.B. Länge der erlaubten Werte) ist nicht möglich.

Schlüsselwort für die Änderung von Systemparametern ist **SYSOPT-CLASS2**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 128.

Format des Parametersatzes zur Vorgabe von Systemparametern

<sysopt> = <wert>

sysopt
Bezeichnet den max. 8-stelligen Namen eines Systemparameters, dessen Wert verändert werden soll.

wert
Bezeichnet den Wert aus einem für den angegebenen Systemparameter gültigen Bereich. Wird einer Systemoption ein ungültiger Wert zugewiesen, dann wird die SYSOPT-Anweisung zurückgewiesen. Wird für eine Systemoption mehrfach ein gültiger Wert verwendet, dann wird nur der zuletzt angegebene gültige Wert berücksichtigt.
Gültig ist eine Anweisung dann, wenn der Typ und der zugewiesene Wert bzw. die Länge zur zugehörigen Systemoption passen.

Eine Kurzbeschreibung aller Systemparameter mit den zulässigen Werten finden Sie beim Kommando SHOW-SYSTEM-PARAMETERS im Handbuch „Kommandos“ [27].

Folgende Aufstellung zeigt für jeden **Typ** die spezifischen Interpretationsregeln:

- Systemparameter vom Typ C (Zeichenketten)

  Der zugewiesene Wert wird als Zeichenkette interpretiert. Soll die Zeichenkette Leerzeichen enthalten, muss der Wert mit vorangestelltem C in Hochkommas eingeschlossen sein, kann dann jedoch selbst keine Hochkommas enthalten. Die Zeichenkette wird linksbündig eingetragen und ggf. mit Leerzeichen aufgefüllt. Ist die angegebene Zeichenkette länger als die maximale Länge des Systemparameters, wird die SYSOPT-Anweisung zurückgewiesen.
- Systemparameter vom Typ A (Arithmetische Werte)

  Der zugewiesene Wert wird als positiver dezimaler Wert interpretiert. Ist für den angegebenen Systemparameter ein Minimum und/oder Maximum definiert, wird der Wert daraufhin überprüft. Sonst wird als Minimum 0 angenommen und das Maximum aus der generierten Länge berechnet (siehe Längentabelle).
- Systemparameter vom Typ X (Hexadezimale Werte)

  Der zugewiesene Wert wird als Hexadezimalwert interpretiert, darf also nur aus Hexadezimalziffern (mit vorangestelltem X und in Hochkommas eingeschlossen) bestehen. Die Überprüfung auf Minimum und Maximum erfolgt wie bei Typ A aus der generierten Länge.

Für Typ A kann die generierte Länge 1, 2 oder 4 Byte betragen, für Typ X 1, 2, 4 oder 8 Byte. Ist kein Maximum definiert, werden folgende Maximalwerte akzepiert:

Längentabelle

Typ	Länge =1 Byte	Länge = 2 Byte	Länge = 4 Byte	Länge = 8 Byte
A	255	65535	2147483647	-
X	FF	FFFF	FFFFFFFF	FFFFFFFFFFFFFFFF

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN SYSOPT-CLASS2
BMTNUM=32                                                    1.
ENCRYPT=Y                                                    2.
EAMSPVS=X'01'                                                3.
TEMPFILE=C'#'                                                4.
/EOF
:
/END-PARAMS
```

1. Die Anzahl der Ein-/Ausgabe-Puffer für die Katalogverwaltung wird auf 32 festgelegt.
   Typ des Systemparameters: A.
2. Es wird vereinbart, dass das System im folgenden Systemlauf mit verschlüsselten Kennwörtern arbeiten soll.
   Typ des Systemparameters: C.
3. Diese Anweisung steuert die Behandlung der SYSEAM-Datei auf einem Shared-Pubset.
   Typ des Systemparameters: X.
4. Als Sonderzeichen, das als Präfix bei temporären Dateien und Jobvariablen vor den Datei- bzw. JV-Namen gesetzt werden muss, wird für den nächsten Systemlauf „#“ vereinbart.
   Typ des Systemparameters: C.

Dynamisch änderbare Systemparameter

Folgende Systemparameter können unter der Benutzerkennung TSOS mit dem Kommando MODIFY-SYSTEM-PARAMETERS im laufenden Betrieb geändert werden:

BLSCOPYN	FILECRYP	NUMBACK	SSMMILOG
BLSCOPYR	FST32GB	PWACTIVE	SSMOUT
BLSLDPXS	ISBLKCTL	PWENTERD	SSMPNOQ
BLSOPENX	ISBLKVAL	PWERRORS	SSMSDEVB
CONSDDE7	JTABSMEM	PWPENTI	SSMSDEVD
DIATTL	JTMAXMEM	RDTESTPR	SYSGJASL
DUMPCL5P	JTSHMEM	SHUTARCH	SYSGJCPU
DUMPCTRL	JTSTDMEM	SHUTCTL	SYSGJPRI
DUMPSD#	MIGHOST	SHUTPROC	TCHOFLO
DUMPSEPA	NBLOGT0	SNAPTIME	TCHREAD
DUMPSREF	NBMSGCSD	SSMAPRI	TCHTACK
EACTETYP	NETCODE	SSMASEC	VMGIORAL
	NRTKILL	SSMCOPT	WRTESTPR

Nicht-privilegierte Systemparameter

Folgende Systemparameter und ihre aktuellen Werte werden auch für nicht-privilegierte Anwender im Kommando SHOW-SYSTEM-PARAMETERS angezeigt:

ASRSW1	DUMPCL5P	NBACODE	NUMBACK
ASRSW2	DUMPSEPA	NBESSIZE	RDTESTPR
AUDALLOW	ENCRYPT	NBLOGENF	SECSTART
BLKCTRL	FILECRYP	NBMESLG	SECSTENF
BLSCOPYN	FMTYFNLG	NBMSGCSD	SHUTARCH
BLSCOPYR	FREFCRYP	NBOPTINT	SHUTPROC
BLSLDPXS	FST32GB	NBRCILU	SSMLGOF1
CMDFILE	HOSTCODE	NBRCSCK	SSMLGOF2
DEFLUID	ISBLKCTL	NBRCSCKN	SSMOUT
DIATTL	MIGHOST	NBRUNSP	SVC79
DMCMAXP	MSGDEST	NBRUNWT	TEMPFILE
		NETCODE	WRTESTPR

3.14 Änderung von IPL-Optionen (SYSOPT-IPL)

Mit dem Parametersatz SYSOPT-IPL kann die Systembetreuung das starre Prinzip „Flexibilität oder Komfort“ bei der Systemeinleitung, das in der Wahl des Modus (DIALOG oder FAST) entschieden wird, aufheben.

Durch den Modus DIALOG können Systembetreuung und Operator neue Software oder Korrekturen flexibel und im differenzierten Dialog mit dem Betriebssystem einsetzen. Allerdings verlängert sich dadurch die Ablaufzeit und durch die Beantwortung von Fragen muss auch eine erhöhte Gefahr von Fehleingaben in Kauf genommen werden. Im komfortablen, dialogfreien Modus FAST hingegen kann wegen der Beschränkung auf Standarddateinamen nicht genügend flexibel auf veränderte Anforderungen (Einsatz neuer Software oder Korrekturen) reagiert werden.

Die Systembetreuung kann mit den Optionen des Parametersatzes SYSOPT-IPL einen Satz bestimmter Vorgaben für die Systemeinleitung auf die Verarbeitung der Parameterdatei auslagern und somit einen Mittelweg zwischen den Modi FAST und DIALOG einschlagen.
Die Anweisungen vom Typ SYSOPT-IPL werden unmittelbar vor dem Laden des BS2000-Klasse-1-Execs eingelesen und verarbeitet (zum genauen Ablauf der Systemeinleitung siehe Abschnitt „FAST-Startup" und Abschnitt „DIALOG-Startup").

Schlüsselwort für die Änderung von IPL-Optionen ist **SYSOPT-IPL**.
Die Maximalzahl der dafür zugelassenen Parametersätze beträgt 16.

Format des Parametersatzes zur Änderung von IPL-Optionen

Format	Bedeutung
BS2000 = datei	Dateiname für BS2000
IOCF-CHECK = NO / zeit	Vorgaben zur Prüfung der IOCF-Identifikation
REPFILE1 = datei : REPFILE4 = datei	Dateiname für Klasse-1/2-Reps
LINKAGE-AUDIT-SCOPE = NO / INTERRUPT-HANDLING / SYSTEM-LEVEL	Vorgaben zur Aktivierung der Funktion LINKAGE-AUDIT
NEW-IPL-MODE = UNCHANGED / FAST	Angaben zur Änderung des Systemeinleitungsmodus

BS2000 = datei
Mit dieser Anweisung wird der Dateiname für das zu ladende BS2000-Klasse-1-Exec und Klasse-2-Exec festgelegt. Ist diese Anweisung in der Parameterdatei belegt, dann wird diese Datei zum Laden der residenten (Klasse-1) und seitenwechselbaren (Klasse-2) Teile des Organisationsprogramms verwendet.

Kann die angegebene Datei nicht verarbeitet werden, so wird dies mit folgender Meldung protokolliert und ein ersatzweise zu verwendender Dateiname im Dialog erfragt:

```
NSI1192 INVALID FILENAME FOR BS2000 IN PARAMETER-SECTION SYSOPT-IPL
```

IOCF-CHECK =
Mit dieser Anweisung wird festgelegt, ob eine Überprüfung der Identifikation (= Zeitpunkt der Generierung) des aktuellen IOCF stattfinden soll oder nicht. Die Systembetreuung kann mit der Anweisung sicherstellen, dass das beim Starten des Servers (IMPL) vom Operator ausgewählte IOCF (I/O-Konfiguration für die Hardware) genau das zu einem bestimmten Zeitpunkt erstellte ist.

IOCF-CHECK = NO
Voreinstellung: Die Identifikation des aktuellen IOCF wird nicht geprüft.

IOCF-CHECK = zeit
Es soll geprüft werden, ob der im I/O-CONFIGURATION-REPORT des IOGEN protokollierte Generierungszeitpunkt des aktuellen IOCF mit der vorgegeben Zeitangabe übereinstimmt.

Der Wert <zeit> ist in der Form `yyyy-mm-dd hh:mm:ss` einzugeben. Wenn die geforderte Prüfung negativ ausfällt, wird dies mit folgender Meldung angezeigt:

```
NSI5206 IOCF-CHECK FAILED. GEN-TIME REQ.=yyyy-mm-dd hh:mm:ss
        IS=yyyy-mm-dd hh:mm:ss
```

REPFILE[1..4] = datei
Mit den Anweisungen REPFILE1, REPFILE2, REPFILE3 und REPFILE4 werden die Dateinamen der maximal vier möglichen Dateien für die zu verarbeitenden Reps (Objektcodekorrekturen) von BS2000 festgelegt. Ist diese Anweisung in der Parameterdatei belegt, werden die angegebenen Dateien in allen Modi in der Reihenfolge REPFILE1 bis REPFILE4 für die Systemeinleitung zum Laden der Korrekturen verwendet.
Können die angegebene Datei oder die Dateien nicht verarbeitet werden, so wird dies mit folgender Meldung protokolliert und ein ersatzweise zu verwendender Dateiname im Dialog erfragt:

```
NSI1192 INVALID FILENAME FOR REPFILEn IN PARAMETER-SECTION SYSOPT-IPL
```

LINKAGE-AUDIT-SCOPE =
Mit dieser Anweisung kann gesteuert werden, ob die AUDIT-Funktionseinheit Linkage-AUDIT die Aufzeichnung von Sprungzieladressen nach Aufruf der Befehle BASR, BALR und BASSM sofort nach Laden des Klasse-1-Execs für einen zu bestimmenden Gültigkeitsbereich eingeschaltet werden soll.
Linkage-AUDIT kann auch zur Diagnoseunterstützung von Problemen genutzt werden, die in einer frühen Phase der Systemeinleitung auftreten können (prozessorlokaler Linkage-AUDIT).
Ist der Systemparameter AUDALLOW=NO eingestellt, werden die AUDIT-Funktionen bei Abarbeitung der Systemparameter für die Dauer des Systemlaufs ausgeschaltet.
Bei Bedarf kann die Linkage-AUDIT-Funktion nach „System Ready“ auch mit dem Kommando STOP-LINKAGE-AUDIT wieder ausgeschaltet werden.

LINKAGE-AUDIT-SCOPE = NO
Voreinstellung: Die Protokollierung wird nicht eingeschaltet.

LINKAGE-AUDIT-SCOPE = INTERRUPT-HANDLING
Die Aufzeichnung der Sprungzieladressen wird nur für den SIH-Teil von BS2000 (Funktionszustand SIH) eingeschaltet.

LINKAGE-AUDIT-SCOPE = SYSTEM-LEVEL
Die Aufzeichnung der Sprungzieladressen wird für den gesamten privilegiert ablaufenden Teil von BS2000 (Funktionszustände TPR und SIH) eingeschaltet.

NEW-IPL-MODE =
Mit dieser Anweisung kann festgelegt werden, ob der vom Operator ausgewählte Modus der Systemeinleitung nach Einlesen der Parameterdatei beibehalten oder auf den Modus FAST geändert werden soll.

NEW-IPL-MODE = UNCHANGED
Voreinstellung: Der zu Beginn der Systemeinleitung eingestellte Modus bleibt unverändert.

NEW-IPL-MODE = FAST
Falls die Systemeinleitung im Modus DIALOG begonnen wurde, wird sie nach Einlesen der Parameterdatei im dialogfreien Modus FAST fortgesetzt.

Ausschnitt aus der Parameterdatei

```
/BS2000 PARAMS
:
/BEGIN SYSOPT-IPL
BS2000=$TSOS.SYSPRG.BS2.<ver>.TEST1                                    1.
REPFILE1=$TSOS.SYSREP.BS2.<ver>.REP1                                   2.
REPFILE2=$TSOS.SYSREP.BS2.<ver>.REP2
NEW-IPL-MODE=FAST                                                      3.
/EOF
:
/END-PARAMS
```

1. Der Dateiname für das während der Systemeinleitung zu ladende BS2000-Klasse-1-Exec und Klasse-2-Exec lautet $TSOS.SYSPRG.BS2.<ver>.TEST1.
2. Es wird vereinbart, dass das System für den folgenden Systemlauf die Objektcodekorrekturen aus den Dateien $TSOS.SYSREP.BS2.<ver>.REP1 bzw. $TSOS.SYSREP.BS2.<ver>.REP2 einlesen soll.
3. Falls die Systemeinleitung im Modus DIALOG begonnen wurde, soll sie nach Einlesen der Parameterdatei im dialogfreien Modus FAST fortgesetzt werden.

4 Speicherverwaltung

Das Kapitel beginnt mit einer Beschreibung des virtuellen Adressraums und der virtuellen Speicherklassen. Anschließend wird die Verwaltung von Big Pages beschrieben. Danach werden die verschiedenen Speichermedien vorgestellt und Maßnahmen zur Verhinderung von Sättigungszuständen aufgezeigt.

Die Speicherverwaltung von BS2000 umfasst folgende Aufgaben:

- Verwaltung des virtuellen Adressraumes des Systems und der Benutzer
- Steuern von Anforderungen zur Reservierung, Belegung und Freigabe von Seiten im
  - virtuellen Adressraum (Virtual Address Space)
  - Hauptspeicher (Main Memory)
  - Seitenwechselspeicher (Paging Area)

  sowie Aufbau und Aktualisierung von Tabellen zur Verwaltung dieser Speicher
- Steuerung des Seitenwechsels
- Zeitliche Überwachung der Belegung von Seiten im Hauptspeicher
- Erkennen von Sondersituationen und Einleiten entsprechender Maßnahmen zur
  - Sättigungsbehandlung
  - Fehlerbehandlung

Die systeminternen Mechanismen und Funktionen bieten der Systembetreuung von BS2000 eine Reihe von Schnittstellen und Einflussmöglichkeiten, die im Folgenden erläutert werden.

4.1 Virtueller Adressraum und virtuelle Speicherklassen

Der virtuelle Adressraum für System und Benutzer, der durch Hauptspeicher und Seitenwechselspeicher realisiert ist, wird in folgende 6 Klassen unterteilt:

- Klasse 1: residente Systemmodule (z.B Unterbrechungsanalyse, Pagingroutine, Task-Management, physikalische Ein-/Ausgabe); residente Tabellen
- Klasse 2: seitenwechselbare Systemmodule (z.B Makro- und Kommandoverarbeitung)
- Klasse 3: residente Tabellen, die dynamisch angelegt und wieder abgebaut werden (z.B. TCBs, PCBs, Adressumsetztabellen)
- Klasse 4: seitenwechselbare Tabellen, die dynamisch angelegt und wieder abgebaut werden (z.B. Job-Management-Tabellen, DVS-Tabellen,Tabellen für Datenfernverarbeitung); nachladbare Systemmodule und sog. Shared Code, der über DSSM geladen und entladen werden kann
- Klasse 5: seitenwechselbare Tabellen, die nur von der jeweiligen Task benötigt werden (z.B TFT, E/A-Bereiche, Tabellen zur Verwaltung von Prozeduren, Kennwort-Tabelle)
- Klasse 6: Benutzerprogramm, Common-Memory-Pools

Die Klassen 1 bis 4 sind privilegierter Systemadressraum, der für den Benutzer nicht adressierbar ist. Eine Ausnahme ist der Shared-Code im Klasse-4-Speicher, der im Allgemeinen auch von Benutzern gelesen und ausgeführt werden kann.

Die Klasse 5 ist überwiegend privilegierter Benutzeradressraum, der als Kommunikationsbereich zwischen Benutzer- und Systemadressraum fungiert und mit Ausnahme der Ein-/Ausgabe-Bereiche für den Benutzer nicht adressierbar ist.

Die virtuelle Adressraumaufteilung kann an die installationsspezifischen Bedürfnisse angepasst werden, indem das Standard-EXEC auf der Server Unit /390 entsprechend modifiziert wird (siehe Kapitel „Anpassen des BS2000-Organisationsprogramms" im Handbuch "BS2000 OS DX Systeminstallation"[55]). Außerdem werden weitere hierzu benötigte Daten für die Speicherverwaltung (Memory Management) über den Parameterservice eingelesen (siehe Kapitel „Parameterservice").

Kommando	**Bedeutung**
ADD-USER	Eintrag im Benutzerkatalog erstellen
MODIFY-USER-ATTRIBUTES	Festlegung der max. Größe des Klasse-6-Speichers einer Benutzerkennung (ADDRESS-SPACE-LIMIT)
SHOW-USER-ATTRIBUTES	Ausgabe der max. erlaubten Größe des Klasse-6-Speichers einer Benutzerkennung (Feld ADDRESS-SPACE-LIMIT)
SHOW-MEMORY-POOL-STATUS	Ausgabe der aktuell im System angelegten Memory Pools und der jeweils angeschlossenen Tasks
Makro	**Bedeutung**
ALESRV	Programm an einen Datenraum anschließen oder die Verbindung auflösen
ALINF	Informationen über die Zugriffslisten ausgeben, mit denen Datenräume und deren Verbindungen verwaltet werden

CSTAT	Status von Speicherseiten eines Programms ändern
CSTMP	Lese-/Schreibstatus auf einen Memory Pool ändern
DISMP	Teilnahme an einem Memory Pool beenden
DSPSRV	virtuellen Adressraum für Datenadressierung erzeugen, erweitern, verkleinern oder löschen, Informationen über einen Datenraum ausgeben oder diesen freigeben
ENAMP	Memory Pool einrichten oder die Teilnahme an einem bestehenden Memory Pool ermöglichen
MINF	Informationen über die Speicherbelegung und Größe des Klasse-6-Speichers oder eines Memory Pools ausgeben
RELM	zusammenhängenden Speicherbereich des aufrufenden Programms freigeben
RELMP	(zusammenhängenden) Speicherplatz eines Memory Pools freigeben
REQM	(zusätzlichen) Speicherplatz für das aufrufende Programm anfordern
REQMP	(zusammenhängenden) Speicherplatz für einen Memory Pool anfordern
SHOWMP	Ausgabe der aktuell im System angelegten Memory Pools und der jeweils angeschlossenen Tasks

Tabelle 6: Schnittstellenübersicht zur Verwaltung des virtuellen Adressraums

4.2 Hauptspeicher

Der Hauptspeicher (Arbeitsspeicher) umfasst residente Seiten und Seiten, die in Working-Sets und im Free-Pool verwaltet werden, siehe Abschnitt „Seitenverwaltungs-Algorithmus".
Working-Sets bestehen aus virtuellen Seiten, die aktuell zugreifbar sind, aber prinzipiell auf den Seitenwechselspeicher ausgelagert werden können.
Zum Free-Pool gehören Seiten, deren Inhalt entweder gar nicht mehr benötigt wird oder (zumindest vorübergehend) nur im Seitenwechselspeicher aufbewahrt werden soll.
Bei Bedarf wird der Free-Pool aus den Working-Sets aufgefüllt.

Im BS2000-Native-Betrieb, für das VM2000-Monitorsystem und für jedes VM2000-Gastsystem wird Hauptspeicher > 2 GB unterstützt.

Kommando	Bedeutung
ADD-USER	Anzahl der residenten Hauptspeicherseiten für Benutzerprogramme einer Kennung festlegen (RESIDENT-PAGES)
MODIFY-MEMORY-PARAMETERS	Prozentsatz des Hauptspeichers festelegen, der für Big-Page-Speicher verwendet wird, sowie ob und wann Big-Page-Speicher abgebaut oder erweitert werden darf
MODIFY-SYSTEM-BIAS	Anzahl der residenten Hauptspeicherseiten für Benutzerprogramme festlegen
MODIFY-USER-ATTRIBUTES	Anzahl der residenten Hauptspeicherseiten für Benutzerprogramme einer Kennung ändern (RESIDENT-PAGES)
SHOW-MEMORY-CONFIGURATION	Aktuelle Konfiguration des Hauptpeichers inkl. Angaben zum Big-Page-Speicher anzeigen
SHOW-SYSTEM-STATUS	Anzahl der residenten Hauptspeicherseiten für Benutzerprogramme anzeigen (INFORMATION=SYSTEM-PARAM)
SHOW-USER-ATTRIBUTES	Anzahl der residenten Hauptspeicherseiten für Benutzerprogramme einer Kennung anzeigen (RESIDENT-PAGES)

Tabelle 7: Kommandoübersicht zur Hauptspeicherverwaltung

4.2.1 Seitenverwaltungs-Algorithmus

Der Seitenverwaltungs-Algorithmus (System-Working-Set-Verfahren, SYS-WS) verwaltet die Hauptspeicher-Seiten global, wobei die Einteilung entsprechend ihrem „Zugriffsalter“ in fünf Gruppen erfolgt:

Gruppe 1	Seiten, die gerade den Frame zugewiesen bekommen haben	Aktive Gruppen: **System-Working-Set** Seiten in diesen Gruppen sind ohne Hardware-Unterbrechung ansprechbar
Gruppe 2	Seiten, bei denen eine Referenz vorlag, deren CPU-Zeit noch nicht abgelaufen war oder die eine zweite Chance verdient haben	
Gruppe 3	Seiten, die zur Überprüfung des Referenz-Indikators anstehen	
Gruppe 4	Seiten, die bei der periodischen Working-Set-Überprüfung als Kandidaten zur Verdrängung aus dem System-Working-Set ermittelt worden sind	
Gruppe 5	Read Write Queue: Seiten, die noch auf die Paging-Area gesichert werden müssen Read Only Queue: Seiten, die ein Abbild auf der Paging-Area haben Empty Queue: Freie Frames	Inaktive Gruppe: **Free-Pool**

Die Seiten jeder Gruppe - genauer gesagt die Hardware-Frames auf denen die Seiten liegen - sind miteinander verkettet.

Prinzipieller Ablauf

Grundsätzlich wird versucht, möglichst viele Seiten aktiv zu halten. Der Free-Pool ist daher relativ klein. Eine gewisse Minimalgröße zur Befriedigung schnell aufeinander folgender Seitenanforderungen ist jedoch notwendig. Bei Erreichen der Minimalgröße wird daher eine Free-Pool-Regulierung mit dem Ziel durchgeführt, immer genügend freie Frames in der Empty Queue vorrätig zu haben.

Das System-Working-Set-Verfahren wählt nach dem LRU(Least Recently Used)-Verfahren die Seiten aus, die zugreifbar im Hauptspeicher bleiben, und die Seiten, die aus dem System-Working-Set in den Free-Pool verdrängt werden:

- Modifizierte Seiten kommen dabei an das Ende der Read Write Queue, da der Seiteninhalt noch auf die Paging-Area gesichert werden muss. Die Paging-Routine sorgt durch Zurückschreiben von Seiten aus der Read Write Queue und anschließendem Umhängen in die Read Only Queue dafür, dass die Read Only Queue im Normalfall nie leer wird.
- Nicht-modifizierte Seiten kommen an das Ende der Read Only Queue, da bereits ein Abbild der Seite auf der Paging-Area existiert.

Bei der Freigabe einer Hauptspeicherseite kommt der zugehörige Frame in die Empty Queue.

Wird eine Hauptspeicherseite angefordert, dann entnimmt das System einen Frame aus der Gruppe 5, füllt ihn bei Bedarf mit dem aktuellen Seiteninhalt aus der Paging-Area oder - bei neu angeforderten Seiten - mit Nullen und bringt ihn dann in die Gruppe 1. Danach ist die Seite zugreifbar und der Referenz-Indikator ist gesetzt.

Außer bei einem Reclaim, bei dem der Eigentümer auf eine bereits deaktivierte Seite im Free-Pool neu zugreift, bedient das System Seitenanforderungen immer aus der Empty Queue, die die unbenutzten Frames enthält. Wird die Empty Queue dadurch zu klein, wird sie aufgefüllt mit Seiten aus der Read Only Queue. Diese enthält Seiten, die überschrieben werden dürfen, weil ihr Inhalt bereits auf der Paging-Area gesichert ist. Beim Übergang in die Empty Queue "verlieren" die Seiten ihren Eigentümer und danach ist kein Reclaim mehr möglich.

Die Empty Queue und die Read Only Queue bilden den Vorrat, aus dem das System Speicheranforderungen unmittelbar befriedigen kann. Wird dieser Vorrat zu klein, schreibt das System Seiten aus der Read Write Queue, die Seiten mit neuem/geändertem Inhalt enthält, auf die Paging-Area. Mit Ende des Schreibvorgangs können die Seiten im Speicher überschrieben werden und gelangen in die Read Only Queue.

Zur Ermittlung, ob eine Seite referenziert wurde, wird von der Hardware (auf x86-Servern von X2000 nachgebildet) beim Zugriff auf die Seite der zugehörige Referenz-Indikator gesetzt.

Beim System-Working-Set-Verfahren gilt:

- Referenzierte Seiten bleiben immer im System-Working-Set.
- Task-lokale Seiten bleiben auch bei Nicht-Referenz im System-Working-Set, wenn die Task seit dem letzten Zugriff eine bestimmte CPU-Zeit noch nicht verbraucht hat.
- Globale Seiten bleiben bei Nicht-Referenz noch einmal im System-Working-Set, d.h. sie bekommen eine zweite Chance und werden erst bei der nachfolgenden Prüfung verdrängt, wenn bis dahin keine Referenz vorliegt.
- Fixierte Seiten bleiben immer im System-Working-Set.

Das System-Working-Set-Verfahren wird sowohl On-Demand bei Engpässen im Free-Pool als auch periodisch in bestimmten Zeitabständen durchgeführt bzw. wenn eine Logische Maschine in den Zustand 'IDLE' geht.

Der einfache Algorithmus des SYS-WS-Verfahrens führt zu einer Aufwandsreduzierung im Paging-Management, aber auch zu einer „gröberen“ Bestimmung des Working-Set-Bedarfs der einzelnen Tasks. Die Aufwandsreduzierung macht sich besonders bei großem Hauptspeicher bemerkbar.

Wird eine Task deaktiviert, so behält sie auch im inaktiven Zustand ihren Working-Set. Wird die Task erneut aktiviert, sind keine Page-Reclaims zur Wiederbeschaffung ihres Working-Sets notwendig.

Das SYS-WS-Verfahren kennt zwar den UPG-Wert (Used-Pages) jeder Task, hat jedoch keine Möglichkeit festzustellen, in welcher Phase sich die Task befindet (auch inaktive Tasks haben einen UPG-Wert != 0).

Die Bestimmung des PPC-Wertes (Planned Page Count) erfolgt daher näherungsweise durch einen Vergleich des UPG-Wertes mit einem angenommenen, vom Kategorietyp abhängigen, mittleren Working-Set-Wert.

Da möglichst viele Seiten aktiv gehalten werden, liegt der Summen-UPG-Wert praktisch immer in der Größenordnung des verfügbaren seitenwechselbaren Hauptspeichers und hat nur begrenzte Aussagekraft. Der Working-Set-Bedarf wird ausschließlich durch den PPC-Wert ausgedrückt. Der Wert ist eine grobe Schätzung. Bei den heute üblichen großen Speicherausbauten ergeben sich Vorteile dadurch, dass die Aufwände für die ohnehin nicht benötigte genaue Abschätzung des Speicherbedarfs unterbleiben.

4.2.2 Big Pages für CISCFW-Kompilate (SU x86)

In der Regel wird der Hauptspeicher in 4-K-Einheiten verwaltet. Diese Einheiten werden Frames genannt und den virtuellen Seiten zugeordnet.
Unter einer (realen) „**Big Page**“ versteht man eine Zusammenfassung von hintereinanderliegenden Frames in einer Größenordnung von mehreren MB, wobei der erste Frame auf das entsprechende MB-Vielfache ausgerichtet ist.

BS2000 unterstützt Big Pages auf SU x86 in einer Größe von 4 MB, das entspricht 1024 Frames. Sie werden für CISCFW-Kompilate genutzt, weil diese auf Big Pages schneller ablaufen.

Benutzte Big Pages können von der CISCFW auf Verlangen zurückgeben werden (z.B. bei der Reduktion des Hauptspeichers eines VM2000-Gastsystems).

Zur Festlegung der gewünschten Anzahl von Big Pages stehen Parameter in der Startup-Parameterdatei und das Kommando MODIFY-MEMORY-PARAMETERS zur Verfügung. Das Kommando SHOW-MEMORY-CONFIGURATION gibt neben Angaben über die Größe des Hauptpeichers auch Angaben zum Big-Page-Speicher aus (Beispiel siehe ''Ausgaben der Hauptspeicher- und Big-Page-Verwaltung'').

Big Pages bei der Systemeinleitung

Die Startup-Parameterdatei kann auch Parameter zur Anlage von Big Pages enthalten.

Einstellungen in der Parameterdatei

Im Parametersatz MEMORY wird Folgendes festgelegt:

- mit dem Parameter BIG-PAGE-SHRSIZE die Größe (in MB) des Big-Page-Speichers, der im Shared-Memory (Klasse-3-Speicher) angelegt und für CISCFW-Kompilate von Shared-Programmen genutzt werden soll
- mit dem Parameter BIG-PAGE-QUOTA, welcher Anteil (in %) vom Hauptspeicher für Big Pages reserviert werden soll (gewünschte Zielgröße)

Güte und Gültigkeit der beim Startup festgelegten Werte

Wieviel realer Big-Page-Speicher (realer Hauptspeicher der Big Pages) im System angelegt werden soll, wird beim Systemstart im MEMORY-Parameterservice (Parameter BIG-PAGE-QUOTA) festgelegt. Der dort angegebene Prozentsatz ist allerdings nur eine Sollgröße. Die Istgröße fällt u.U. geringer aus, weil die Anlage des Big-Page-Speichers an die maximal mögliche Anzahl der logischen Maschinen (CPUs) des Systems gekoppelt ist. Je geringer die Speichergröße, je mehr CPUs, desto höher ist die Wahrscheinlichkeit, dass die Istgröße des Big-Page-Speichers die Sollgröße nicht erreicht.

Die eingestellten Werte für die Arbeit mit Big Pages sind für die aktuelle Session solange gültig, bis sie entweder explizit mit dem Kommando MODIFY-MEMORY-PARAMETERS oder implizit durch bestimmte Systemeigenschaften und -zustände verändert werden. Dazu gehören die Vergrößerung oder Reduzierung des Hauptspeichers oder eine drohende Hauptspeichersättigung.

Anpassung der Big Pages im laufenden Betrieb

Big Pages können gleichmäßig über den Hauptspeicher verteilt werden, da sie auch oberhalb des Hauptspeicherminimums liegen dürfen. Hieraus ergibt sich automatisch, dass die Größe des Big-Page-Speichers nach einer Hauptspeicher-Rekonfiguration weiter der BIG-PAGE-QUOTA entspricht (sofern nicht zuvor schon eine ungleichmäßige Verteilung der Big Pages entstanden war, z.B. durch explizite Veränderung der eingestellten Werte).

Sonst ist eine Veränderung der Ist-Größe des Big-Page-Speichers während des Systemlaufs nur beschränkt möglich. Das Kommando MODIFY-MEMORY-PARAMETERS bietet dazu mehrere Möglichkeiten, siehe "Hauptspeicher", Tabelle 7. Die Möglichkeiten sind detailliert im Handbuch „Kommandos“ [27] beschrieben.

Ausgaben der Hauptspeicher- und Big-Page-Verwaltung

Die Größe des Speicherbereiches für die Big Pages wird über die Meldung `EMM2309` an der Konsole ausgegeben. Diese Meldung erscheint auch, wenn sich an dieser Größe im laufenden Betrieb etwas ändert, z.B.:

```
EMM2309 THE SIZE OF THE BIG PAGE MEMORY IS 168 MB.
```

Mit dem Kommando SHOW-MEMORY-CONFIGURATION kann jederzeit eine Statusabfrage zu Zuteilung und Auslastung des gesamten Hauptspeichers und der Big-Page-Bereiche erfolgen.

Beispiel

```
/SHOW-MEMORY-CONFIGURATION
 REAL MEMORY MANAGEMENT REPORT                                          1.
   MEM-SIZE  MIN-MEM-SIZE  FREE-CORE-SIZE  FREE-MEM-SIZE  PAGE-MEM-SIZE  CSL
    1024 MB        256 MB        32.25 MB       32.25 MB      800.25 MB  0
 BIG PAGE MANAGEMENT REPORT                                             2.
   QUOTA   PLANNED  #LM    ACTUAL     FREE     SHR   FREE-SHR       REDUCE
      40    400 MB    4    168 MB   104 MB   64 MB      16 MB   CORE/QUOTA
```

1. Im Beispiel hat das System einen Hauptspeicher von MEM-SIZE = 1024 MB, wobei seine Mindestgröße MIN-MEM-SIZE = 256 MB beträgt.

   (In einem VM2000-Gastsystem kann MIN-MEM-SIZE kleiner als MEM-SIZE sein. Im Native-Betrieb sind die Werte derzeit gleich.)

   FREE-CORE-SIZE ist der Anteil am freien Hauptspeicher, der noch für residente Klasse-3-Speicher-Anforderungen des Systems zur Verfügung steht. Diese Größe ist auch maßgebend für den Hauptspeichersättigungsgrad.

   FREE-MEM-SIZE ist der Anteil am freien Hauptspeicher, der für die restlichen residenten Speicheranforderungen wie residente Memory-Pools oder Data Spaces zur Verfügung steht. (Bei Hauptspeicher <= 2 GB sind FREE-CORE-SIZE und FREE-MEM-SIZE identisch.)

   PAGE-MEM-SIZE stellt die Größe des Hauptspeichers dar, der für nicht-residente, seitenwechselbare Seiten zur Verfügung steht.

   (In Systemen, bei denen MEM-SIZE und MIN-MEM-SIZE identisch sind, sind auch FREE-MEM-SIZE und PAGE-MEM-SIZE identisch.)

   Die letzte Spalte CSL gibt den aktuellen Hauptspeichersättigungsgrad an, dabei sind die Werte 0 (keine Sättigung), 1, 2 oder 3 (höchste Alarmstufe) möglich (siehe auch "Hauptspeicher").

2. QUOTA gibt den Prozentsatz des Hauptspeichers an, der als Big-Page-Speicher verwendet werden soll (und der mit dem Parameter BIG-PAGE-QUOTA in der Parameterdatei bzw. mit einem vorangegangenen Kommando MODIFY-MEMORY-PARAMETERS eingestellt wurde).

   Aus der Hauptspeichergröße (MEM-SIZE) und dem Prozentsatz (QUOTA) berechnet sich die Sollgröße, die bei PLANNED angezeigt wird.

#LM ist die Anzahl der max. möglichen logischen Maschinen (CPUs) des Systems, also nicht die zum Zeitpunkt des Kommandoaufrufes angeschlossenen (attached) LMs. Neben der Mindestgröße des Arbeitsspeichers spielt die Anzahl der LMs eine wesentliche Rolle für die aktuelle Größe des Big-Page-Speichers: je höher die Anzahl der LMs, desto kleiner kann die Größe des Big-Page-Speichers ausfallen.

Die Spalte ACTUAL gibt die aktuelle Größe des Big-Page-Speichers aus, d.h. seine Istgröße.
Diese kann von der geplanten Sollgröße (PLANNED) abweichen, wenn z.B. Big-Page-Speicher wegen drohender Hauptspeichersättigung abgebaut wurde.

FREE gibt die Größe des freien (aktuell nicht benutzten) Big-Page-Speichers aus.

Die Größe des vorgesehenen shared Big-Page-Speichers für CISCFW-Kompilate im Klasse-3-Speicher gibt die Spalte SHR aus und FREE-SHR die Größe des freien shared Big-Page-Speichers.

Eine wichtige Angabe findet sich in der letzten Spalte REDUCE: Hier steht, wann Big-Page-Speicher abgebaut werden darf. Im Beispielfall bei drohender Hauptspeichersättigung (CORE) und falls sich durch Änderung der Hauptspeichergröße oder von BIG-PAGE-QUOTA eine neue Sollgröße des Big-Page-Speichers ergibt, die kleiner als die Istgröße ist (QUOTA).

Allgemeine Hinweise

- Für die Hauptspeicherverwaltung gelten generell die folgenden Bedingungen:
  - MEM-SIZE >= MIN-MEM-SIZE > FREE-MEM-SIZE >= FREE-CORE-SIZE
  - MEM-SIZE > PAGE-MEM-SIZE >= FREE-MEM-SIZE >= FREE-CORE-SIZE
- CISCFW-Speicher oder DAB-Puffer werden in dem mit FREE-MEM-SIZE bezeichneten Speicher angelegt. Trotzdem kann eine Hauptspeichersättigung durch Abbau von DAB-Puffern oder Terminierung von Programmen beseitigt werden, wenn FREE-MEM-SIZE minus FREE-CORE-SIZE kleiner als die Größe der DAB-Puffer und /oder des CISCFW-Speichers ist.
  Ist dies nicht der Fall, sind Maßnahmen zur Reduktion des Klasse-3-Speichers bzw. zum Abbau von Adressräumen (Tasks, Data Spaces) erforderlich.

Hinweise für VM2000

- Nur in VM2000-Gastsystemen gibt es ein sog. Hauptspeicherminimum.
  Dieses Minimum, das von der Gesamtspeichergröße nach unten abweicht, ermöglicht es, eine Speicherreduktion des Systems durchzuführen.
- In einem VM2000-Gastsystem kann man der gewünschten Sollgröße des Big-Page-Speichers näher kommen, wenn man den Gesamtspeicher des Gastsystems vergrößert.
- Ist für ein laufendes VM2000-Gastsystem eine Speicherreduktion vorgesehen, dann ist für diese VM eine MIN-MEM-SIZE abweichend von der MEMORY-SIZE festzulegen. Um eine solche Speicherreduktion zu ermöglichen, befriedigt die Speicherverwaltung von BS2000 alle Anforderungen von residentem Speicher, die unterhalb dieser MIN-MEM-SIZE liegen. Eine ausreichende Dimensionierung dieses Parameters ist daher notwendig. Siehe dazu auch den Abschnitt „Performance-Aspekte bei VM2000-Betrieb (Hauptspeicher)“ im „Performance-Handbuch“ [37].

4.3 Seitenwechselbereich (Paging-Area)

Paging-Dateien und Seitenwechselbereich

Für die Bereitstellung von virtuellem Adressraum wird Seitenwechselspeicher benötigt, der mit Seitenwechseldateien (Paging-Dateien) realisiert wird. Paging-Dateien tragen immer den Namen `SYS.PAGING.<vsn>`.
Um zum Seitenwechsel herangezogen zu werden, muss eine eingerichtete Paging-Datei explizit dem Seitenwechselspeicher zugewiesen werden.

Die Gesamtheit der dem Seitenwechselspeicher zugewiesenen Paging-Dateien im System heißt Seitenwechselbereich (Paging-Area).

Der Seitenwechselbereich kann während des Systemlaufs dynamisch erweitert oder reduziert werden, indem Paging-Dateien zum Seitenwechselbereich zugewiesen oder aus ihm freigegeben werden. Während der Systemeinleitung ist eine bestimmte Größe und Beschaffenheit des Seitenwechselbereichs Voraussetzung (siehe "Auswahl des Seitenwechselbereichs bei Startup").

Durch die Größe des Seitenwechselbereichs wird der verfügbare virtuelle Adressraum eines Systemlaufs bestimmt, da für jede virtuelle Seite ein Block (zwei 2K-PAM-Seiten) im Seitenwechselbereich vorhanden sein muss. Die max. Gesamtgröße des Seitenwechselbereichs beträgt 4 TB.

Mit Hilfe des Parameterservices können während der Systemeinleitung aus der Menge der eingerichteten und verfügbaren Paging-Dateien diejenigen Dateien ausgewählt werden, die im anschließenden Systemlauf verwendet werden sollen (siehe Abschnitt „Auswahl desSeitenwechselbereichs beim Startup (PAGING)").
Deshalb ist zwischen dem Einrichten von Paging-Dateien (inklusive der Reservierung von Speicherplatz) und der Auswahl von Paging-Dateien für den Systemlauf zu unterscheiden.

Randbedingungen für eine Paging-Datei:

- Sie darf keine Datei >= 32 GB sein
- Sie darf auf keiner Privatplatte eingerichtet werden
- Sie darf auf keinem Shared-Pubset eingerichtet werden

Paging-Platte und Paging-Pubset

Eine Platte wird als Paging-Platte bezeichnet, wenn dort eine Paging-Datei nicht nur eingerichtet, sondern auch dem Seitenwechselspeicher zugewiesen worden ist.

Generell sind alle Plattentypen erlaubt. Die Platten des Seitenwechselbereichs müssen nicht vom selben Typ sein.

Ein Pubset wird als Paging-Pubset bezeichnet, wenn mindestens eine Platte des Pubsets als Paging-Platte genutzt wird.

Enthält ein Pubset nur Paging-Dateien, ist er ein reiner Paging-Pubset.
Liegt kein reiner Paging-Pubset vor und besteht der Pubset aus mehreren Platten, sollte keine Paging-Datei auf der Pubres des Pubsets angelegt werden.

Es wird empfohlen, den Seitenwechselbereich überwiegend mit Ein-Platten-Pubsets zu realisieren, da dadurch die Rekonfiguration des Seitenwechselbereichs im laufenden Betrieb bei Aufnahme einer Paging-Datei vereinfacht wird.

Kommando	Bedeutung
CREATE-PAGING-FILE	Paging-Datei erzeugen
DELETE-PAGING-FILE	Paging-Datei zerstören (überschreiben mit binär Null)
EXTEND-PAGING-AREA	Seitenwechselbereich dynamisch erweitern
MODIFY-PAGING-AREA-ATTRIBUTES	Attribute von Paging-Dateien auf Platte ändern
REDUCE-PAGING-AREA	Seitenwechselbereich reduzieren
SHOW-PAGING-CONFIGURATION	Informationen über genutzte und ungenutzte Paging-Dateien ausgeben

Tabelle 8: Kommandoübersicht Seitenwechselbereich

Das Kommando SHOW-PAGING-CONFIGURATION gibt folgende Informationen aus:

- Welche Paging-Dateien werden genutzt?
- Welche Paging-Dateien (auf importierten Pubsets) sind ungenutzt?
- Enthält ein Volume eine genutzte oder eine ungenutzte Paging-Datei?
- Welche Gesamtgröße haben die gefundenen genutzen/ungenutzten Paging-Dateien?

Beispiele für die Ausgabe von Paging-Dateien

```
/SHOW-PAGING-CONFIGURATION VOLUME=*UNUSED,INFORMATION=*ALL
% LIST OF THE FOUND AND UNUSED PAGING-FILES :
%
%  VOLUME  CAT-ID        SIZE                   RESTRICTION
%  1OSH.0   1OSH       256.00 MB                   NONE
%  1OSH.2   1OSH       256.00 MB                   NONE
%  ------------------------------------------------------------------------
%  SUMMARY             512.00 MB

/SHOW-PAGING-CONFIGURATION VOLUME=*USED,INFORMATION=*ALL
% LIST OF THE USED PAGING-FILES :
%
%  VOLUME  CAT-ID        SIZE       FREESIZE     UTILIZATION  REDUCTION  R
%  1OP1.1 * 1OP1      2500.00 MB     1510.99 MB    *LOW       *NOT-REQ   N
%  1OP1.0 * 1OP1      2500.00 MB     2200.00 MB    *LOW       *NOT-REQ   N
%  1OPP.0 * 1OPP      2500.00 MB      395.90 MB    *LOW       *NOT-REQ   N
%  1OSH.1   1OSH       256.00 MB      200.00 MB    *LOW       *NOT-REQ   N
%  ------------------------------------------------------------------------
%  SUMMARY            7756.01 MB
%
%  EMM2317 THE SIZE OF VOLUME(S) - MARKED WITH * - IS INCREASED BY 25%,
            BECAUSE THE PAM-KEYS ARE USED AS PAGING-AREA
```

4.3.1 Einrichten, Zuweisen, Freigabe und Löschen von Paging-Dateien

Einrichten von Paging-Dateien

Paging-Dateien werden unter der Benutzerkennung der Systemverwaltung mit dem Namen `SYS.PAGING.<vsn>` katalogisiert. Eine Paging-Datei darf auf dem Datenträger mit dem angegebenen Datenträgerkennzeichen vsn mehrere Extents haben. In einem Pubset kann je Datenträger genau eine Paging-Datei `SYS.PAGING.<vsn>` existieren.

Eine Paging-Datei hat eine Mindestgröße von 1 MB. Aus Performancegründen wird empfohlen, den Seitenwechselbereich auf mehrere Paging-Dateien (also auf mehrere Datenträger) zu verteilen und die Größe der Paging-Dateien nahezu gleichzuhalten.

> **i** Je höher die Paging-Rate ist, desto wichtiger ist die Verteilung des Seitenwechselbereichs auf mehrere Paging-Dateien. Die Paging-Ein-/Ausgaben können beschleunigt werden, indem die verwendeten Platten an unterschiedliche Kanäle angeschlossen werden. Die Paging-Rate kann mit openSM2 (siehe Handbuch „openSM2“ [49]) gemessen werden.

Paging-Dateien können im laufenden Betrieb mit dem Kommando CREATE-PAGING-FILE eingerichtet werden. Durch dieses Kommando wird eine Paging-Datei `SYS.PAGING.<vsn>` in der angegebenen Größe erstellt. Über die VSN kann dabei jede Platte eines importierten Pubsets ausgewählt werden.

Zuweisen von Paging-Dateien zum Seitenwechselbereich

Eine existierende Paging-Datei kann über das Kommando EXTEND-PAGING-AREA in Betrieb genommen, d.h. dem Seitenwechselbereich zugeordnet werden. Damit sie auch in den folgenden Systemläufen von Beginn an verwendet wird, muss sie in die System-Parameterdatei eingetragen werden (siehe Parametersatz PAGING im Abschnitt „Auswahl des Seitenwechselbereichs beim Startup (PAGING)").

Das Einrichten von Paging-Dateien kann auch mit der SIR-Anweisung CREATE-PAGING-FILE erfolgen; SIR ist im Handbuch „Dienstprogramme“ [15] beschrieben.

Es ist sinnvoll, bereits bei der Systeminstallation Paging-Dateien auf dem Home-Pubset einzurichten, da somit automatisch die Anforderungen des Betriebssystems sichergestellt sind.

Die benötigte Größe des Seitenwechselbereichs hängt von der Anzahl und der virtuellen Programmgröße der Programme ab, die gleichzeitig ablaufen. Hinzu kommt der jeweils eingestellte Systemadressraum.

Für eine erste Abschätzung wird folgende Formel empfohlen:

```
benötigte Größe = 2 * (Anzahl Benutzertask * virtuelle Programmgröße +
Systemadressraum)
```

Nach Aufnahme des Betriebs sollte eine Überprüfung (z.B mit dem Kommando SHOW-PAGING-CONFIGURATION oder mit openSM2; siehe Handbuch „openSM2“ [49]) erfolgen, um die tatsächlich benötigte Größe des Seitenwechselbereichs festzustellen (siehe auch „Performance-Handbuch“ [37]).

Freigabe von Paging-Dateien aus dem Seitenwechselbereich

Mit dem Kommando REDUCE-PAGING-AREA kann im laufenden Betrieb eine Paging-Datei aus dem Seitenwechselbereich entnommen werden.

Die Freigabe ist aber nur dann möglich, wenn der vorhandene Seitenwechselbereich noch nicht ausgeschöpft ist und das System durch diese Reduzierung nicht in einen Sättigungszustand gerät.
Das Kommando REDUCE-PAGING-AREA läuft asynchron in einer Server-Task ab.

Zu Möglichkeiten zur beschleunigten Freigabe siehe "Möglichkeiten zur Beschleunigung der Freigabe von Paging-Dateien".

Löschen von Paging-Dateien

Zum Löschen einer Paging-Datei steht das Kommando DELETE-PAGING-FILE zur Verfügung. Der zugehörige Pubset muss dabei importiert sein.

Die Ausführung des Kommandos DELETE-PAGING-FILE dauert relativ lange, weil der Dateiinhalt gelöscht wird (Überschreiben mit binären Nullen).
Das Kommando läuft deshalb asynchron in einer Server-Task ab.

Eine Besonderheit besteht beim Löschen einer Paging-Datei des Home-Pubsets:
Das Löschen mit DELETE-PAGING-FILE wird abgelehnt, wenn nicht genügend andere Paging-Dateien im Home-Pubset erhalten bleiben.

Dynamische Rekonfiguration des Seitenwechselbereichs

Während des Systemlaufs kann der Seitenwechselbereich jederzeit mit dem Kommando EXTEND-PAGING-AREA dynamisch erweitert werden. Voraussetzung ist das Importieren des Pubsets mit der aufzunehmenden Paging-Datei und die Beachtung der max. Größe des Seitenwechselbereichs von 4 TB.

Mit dem Kommando REDUCE-PAGING-AREA kann der Seitenwechselbereich dynamisch verkleinert werden. Dabei sind einige Vorausetzungen zu beachten, z.B. ob die verbleibende Größe des Seitenwechselbereichs für die anstehenden Aufgaben ausreicht.

> **ACHTUNG!**
> Es wird empfohlen, mit der Rekonfiguration des Seitenwechselbereichs im laufenden Betrieb sehr sorgsam umzugehen und sie auf ein Mindestmaß zu beschränken. Häufiges Rekonfigurieren kann die Gesamt-Performance des Systems immer mehr verschlechtern. Dies gilt besonders bei Systemen mit hoher Paging-Last.

Folgende Arbeitsweise dient zur Erhaltung einer guten Gesamt-Performance:

- Beschränkung auf Paging-Dateien, die voraussichtlich nie oder selten aus dem Seitenwechselbereich genommen werden sollen.
- Die Verwendung zusammenhängender Paging-Dateien wird empfohlen.
- Es ist günstiger, viele kleine Platten zu verwenden als wenige große.
- Für diese Platten wird die exklusive Verwendung für Paging-Dateien empfohlen (reine Paging-Pubsets).

Möglichkeiten zur Beschleunigung der Freigabe von Paging-Dateien

Ist die freizugebende Paging-Datei fast vollständig ausgelastet, so ist durch die Verlagerung der virtuellen Seiten auf andere Paging-Dateien mit einer längeren Kommandoausführungsdauer, einer hohen I/O-Auslastung und mit einer hohen CPU-Belastung zu rechnen. Um diese Belastungen zu verringern bzw. die Reduktion zu erleichtern, sind folgende Möglichkeiten gegeben:

1. Die Reduktion wird dem Betriebssystem einige Zeit vorher angekündigt.

   Die betroffene Paging-Datei wird mit dem Kommando MODIFY-PAGING-AREA-ATTRIBUTES (Operand UTILIZATION=*LOW) ab diesem Zeitpunkt – so weit wie möglich – nicht mehr zur Aufnahme virtueller Seiten herangezogen.
   Durch Zugriff auf Seiten innerhalb der Paging-Datei oder Seitenfreigabe wird die Paging-Datei leerer und bei der eigentlichen Reduktion sind weniger Ein-/Ausgaben zum Auslagern dieser Seiten nötig.

   Ein Erfolg dieser Maßnahme ist aber nicht garantiert: Entweder kann auf das Einlagern von Seiten in diese Paging-Datei nicht verzichtet werden (wegen zu hoher Paging-Rate) oder auf die Seiten innerhalb der Paging-Datei wird nicht zugegriffen.

   Ob diese Maßnahme den gewünschten Erfolg bringt, kann über das Kommando SHOW-PAGING-CONFIGURATION ermittelt werden: Ein Ansteigen der ausgegebenen freien Größe der betroffenen Paging-Datei ist ein Maß für den Erfolg und sagt aus, wann die Reduzierung zu starten ist. Dies kann nach einigen Minuten oder auch erst Stunden der Fall sein.
2. Dem Betriebssystem wird bereits beim Zuweisen der Paging-Datei zum Seitenwechselbereich mit dem Kommando EXTEND-PAGING-AREA ...,UTILIZATION=*LOW mitgeteilt, dass diese Paging-Datei zur Aufnahme der virtuellen Seiten weniger herangezogen werden soll.
   Es gelten die gleichen Aussagen zum Erfolg wie im vorhergehenden Punkt.
3. Beim Kommando EXTEND-PAGING-AREA gibt es bei Angabe des Operanden LATER-REDUCTION=*REQUESTED die Möglichkeit, die CPU-Belastung bei der Reduktion zu verringern.
   Für die Verwaltung dieser Paging-Dateien wird jedoch mehr residenter Arbeitsspeicher verbraucht.

Bei allen genannten Maßnahmen gilt die Empfehlung, sie auf maximal zwei Paging-Dateien anzuwenden.

4.3.2 Auswahl des Seitenwechselbereichs bei Startup

Die Systembetreuung kann zum Startup-Zeitpunkt über den Parameterservice die Datenträger bestimmen, die für den Seitenwechsel benutzt werden sollen. Dabei ist zu beachten, dass auf diesen Datenträgern Paging-Dateien `SYS.PAGING.<vsn>` angelegt sind und dass die maximale Anzahl von 128 Datenträgern nicht überschritten wird.

Der von Startup initialisierte Seitenwechselbereich sollte eine Mindestgröße von 200 MB haben. Ist der Seitenwechselbereich kleiner als 200 MB, wird als Warnung die Meldung `NSI5115` an der Konsole ausgegeben; die Systemeinleitung wird fortgesetzt.

Hat die Systembetreuung keine Parameter für die Auswahl von Paging-Dateien angegeben, dann werden automatisch alle auf dem Home-Pubset eingerichteten Paging-Dateien `SYS.PAGING.<vsn>` für den Seitenwechsel verwendet.
Mit dem Parameterservice können Datenträger für Seitenwechsel innerhalb und außerhalb des Home-Pubsets angegeben werden.
Die dazu notwendigen Anweisungen muss die Systembetreuung in der Parameterdatei hinterlegen (siehe PAGING-Parametersatz, Kapitel „Parameterservice“).

4.4 Maßnahmen zur Verhinderung von Sättigungszuständen

In folgenden Bereichen kann die Ausführung einer Speicherplatzanforderung zu Sättigungszuständen führen: Hauptspeicher, Systemadressraum und Seitenwechselspeicher.

Speicherengpässe können auch bei folgenden Gelegenheiten auftreten, die an anderer Stelle beschrieben werden:

- bei zu hoher Hauptspeicher-Auslastung oder zu hoher Verdrängungsrate
  (siehe "Zuteilung von Ressourcen")
- bei der Anforderung von Speicherplatz auf gemeinschaftlichen Datenträgern
  (siehe "Speicherplatz-Sättigung überwachen")

Die Meldungen, die abhängig von Sättigungsursache und -zustand ausgegeben werden, sind in den folgenden Abschnitten aufgeführt. Das Gleiche gilt auch für die Maßnahmen, die sowohl vom Operator als auch von der Systemverwaltung durchgeführt werden können. In diesem Abschnitt sollen die langfristigen Maßnahmen zur Verhinderung von Sättigungszuständen im Vordergrund stehen.

Hauptspeicher

- Die Hauptspeichergröße sollte entsprechend den Empfehlungen des Performance-Handbuchs gewählt werden. Ein zu kleiner Hauptspeicher wirkt sich meist sehr negativ auf die Performance aus, weil die Paging-Rate und die Verdrängungsrate zu hoch werden.
- Bei VM2000-Gastsystemen müssen die Empfehlungen des Performance-Handbuchs für den Wert von MINIMAL-MEMORY-SIZE eingehalten werden.
- Mit dem Kommando MODIFY-SYSTEM-BIAS kann Benutzerprogrammen erlaubt werden, eine größere Anzahl von Benutzerseiten resident zu machen.
  Es wird empfohlen, diese Erlaubnis auf die Benutzerkennungen zu begrenzen, in denen solche Programme laufen müssen (Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES).

Systemadressraum

- Der Bedarf an Systemadressraum richtet sich stark nach der System-Konfiguration und der Geräte-Peripherie, nach den eingesetzten Subsystemen und nach der Last.
- Es wird dringend empfohlen, den Systemadressraum großzügig zu bemessen (die Größe ist nur bei SU /390 einstellbar, siehe "Systemadressraum").
- Ein zu geringer Vorhalt kann unerwünschte Betriebsunterbrechungen erzwingen (z.B. wegen Subsystem-Versionsumstieg, Anstieg der Last, Ausbau der Systemkonfiguration oder wegen der Nachwirkungen eines Sättigungszustands).
- Empfohlen wird auch die Nutzung der vorhandenen Möglichkeiten zur Lastbegrenzung. So kann z.B. die Anzahl der Benutzer in den Jobklassen begrenzt werden (Kommando MODIFY-JOB-CLASS). Auch die Programmgröße kann benutzerspezifisch beschränkt werden (über die Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES).

Seitenwechselspeicher

- Die Größe des Seitenwechselspeichers sollte entsprechend den Empfehlungen des Performance-Handbuchs eingestellt werden.
- Ein ausreichend großer Seitenwechselspeicher muss bereits bei der Systemeinleitung in Betrieb genommen werden (Parameterservice).

- Falls der Bedarf an Seitenwechselspeicher während des Systemlaufs stark anwächst, sollte die Erweiterung möglichst frühzeitig erfolgen (über das Kommando EXTEND-PAGING-AREA).
- Die Möglichkeiten zur Lastbegrenzung sollten genutzt werden (siehe oben bei „Systemadressraum“)

Um einen Leistungsabfall zu verhindern, treten bei Sättigungszuständen Algorithmen in Kraft, die der Sättigung entgegenwirken.

Vom Organisationsprogramm werden dabei die in den folgenden Tabellen enthaltenen Meldungen auf der Konsole ausgegeben.

4.4.1 Hauptspeicher

Sättigungszustand

Meldung: `EXC0870 CORE SATURATION LEVEL=i`

`i` ist eine Ziffer von 0 bis 3 und bezeichnet den Grad der Sättigung.

Grad	Auswirkungen im System	Maßnahme des Operators
0	Normaler Systemablauf	Keine
1	Neue Batch-Jobs werden nicht gestartet.	Keine
2	Neue Batch-Jobs werden nicht gestartet.	Keine
3	Neue Batch-Jobs werden nicht gestartet. Die Anforderung von residenten Speicherseiten wird nur noch bei besonders wichtigen System-Jobs befriedigt.	Der Operator kann Jobs abbrechen.

Rückgang der Sättigung

Grad	Auswirkungen im System	Maßnahme des Operators
2 ... 0	Rückgang der Sättigung Nach Erreichen von Level=0 können wieder Batch-Jobs gestartet werden.	Keine

Maßnahmen zur Verhinderung der Sättigung

Das Auftreten eines Sättigungszustandes zeigt an, dass das System überlastet ist.
Wenn der Hauptspeicherbedarf sich nicht auf Dauer verringern lässt, benötigt das System einen größeren Hauptspeicher. Handelt es sich um ein VM2000-Gastsystem, so ist eine Erhöhung der MINIMAL-MEMORY-SIZE durch den VM2000-Administrator erforderlich.
Eine größere NEW-MEMORY-SIZE bei gleichbleibender MINIMAL-MEMORY-SIZE ist als Gegenmaßnahme ungeeignet.

Bei SU x86 ist hier der BS2000-Speicher gemeint. Welcher Anteil des gesamten Speichers das ist, wird über X2000 eingestellt.

Beispiele zur Verringerung des Hauptspeicherbedarfs eines Systems

- DAB-Cache-Bereiche im Hauptspeicher verkleinern
- Anzahl der residenten Katalog-Puffer einzelner Pubsets gezielt verringern oder die Puffer seitenwechselbar machen
- Nicht belegte Pubsets exportieren
- Entladen von Anwendungen, die Benutzerseiten resident gesetzt haben
- Entladen großer Anwendungen bzw. Programme
- Deutliche Verringerung der Taskanzahl im System

4.4.2 Systemadressraum

Sättigungszustand

Meldung: `EXC0874 ADDRESS SPACE SATURATION=i`

`i` ist eine Ziffer von 0 bis 2 und bezeichnet den Grad der Sättigung.

Grad	Auswirkungen im System	Maßnahme des Operators
0	Normaler Systemablauf	Keine
1	Es werden keine neuen Batch-Jobs gestartet.	Systemverwaltung verständigen
2	Es werden keine neuen Batch-Jobs gestartet.	

Maßnahmen zur Verhinderung der Sättigung

Das Auftreten eines Sättigungszustandes zeigt an, dass das System überlastet ist. Wenn das System immer wieder in einen Sättigungszustand kommt, ist es nicht ausreichend konfiguriert.
Dadurch kann es notwendig werden, den Benutzeradressraum zu verkleinern (siehe auch Handbuch „Systeminstallation“ [55], Anpassen des BS2000-Organisationsprogramms, Prozedur SYSPRC.BS2000-EXEC.<ver>).

An SU x86 bestehen diese Konfigurationsmöglichkeiten nicht.

Beispiele zur Verringerung des Systemadressraumbedarfs eines Systems

- Anzahl der Katalog-Puffer einzelner Pubsets gezielt verringern.
- Nicht belegte Pubsets exportieren.
- Entladen großer Anwendungen bzw. Programme.
- Deutliche Verringerung der Taskanzahl im System.
- Vor dem Entladen großer Subsysteme wird ausdrücklich gewarnt. Hierdurch kann zwar der Sättigungszustand meist kurzfristig aufgehoben werden, häufig lässt sich aber das entladene Subsystem auch später nicht mehr laden, sodass für den Rest des Systemlaufs darauf verzichtet werden muss.
- Häufige Sättigung im Systemadressraum trotz richtiger Konfigurierung kann auf einen Systemfehler hinweisen. Bitte konsultieren Sie den Service (Systemdiagnose).

4.4.3 Seitenwechselspeicher

Sättigungszustand

Meldung: `EXC0873 NEW PAGING MEMORY SATURATION LEVEL=i`
`i` ist eine Ziffer von 0 bis 3 und bezeichnet den Grad der Sättigung.

Grad	Auswirkungen im System	Maßnahme des Operators
0	Normaler Systemablauf	Keine
1	Warnung von einem möglichen Sättigungszustand (bei Angabe des Parameters PAGING-SATURATION-WARNING-LIMIT, siehe "Speicherverwaltung (MEMORY)")	Siehe „Maßnahmen zur Verhinderung der Sättigung"
2	Batch-Jobs, die Klasse-6-Speicher anfordern, werden ausgelagert.[1)] Neue Jobs werden nicht gestartet.	Der Operator muss versuchen, Jobs abzubrechen oder mit EXTEND-PAGING-AREA den Seitenwechselbereich dynamisch zu erweitern.
3	Dialog- und Batch-Jobs, die Klasse-6-Speicher anfordern, werden ausgelagert. Folgende Meldung erscheint auch auf SYSOUT der Dialog-Jobs. `EXC0844 TASK TEMPORARILY HELD DUE TO PAGING DEVICE SATURATION. TASK WILL BE CONTINUED AUTOMATICALLY` Die TCB-Adressen der Jobs werden in einer Warteschlange verwaltet. Sie werden dort wie folgt eingereiht: • Dialog-Jobs vor Batch-Jobs • Jobs mit hoher Priorität vor Jobs mit niederer Priorität • Jobs gleicher Priorität abhängig von der Zahl ausgelagerter Seiten (Jobs mit weniger ausgelagerten Seiten kommen zuerst) Neue Jobs werden nicht gestartet.	Der Operator muss versuchen, Jobs abzubrechen oder mit EXTEND-PAGING-AREA den Seitenwechselbereich dynamisch zu erweitern. Es kann zum Systemstillstand bzw. Systemabbruch kommen.

[1)] Der Klasse-6-Speicher des Jobs wird in eine temporäre Plattendatei (EAM) gerettet und dann freigegeben. Ein solcher Job muss folgende Bedingungen erfüllen:

- Die Klasse-6-Speicheranforderung bezieht sich nicht auf gemeinsamen Speicherplatz für mehrere Jobs (Memory Pool).
- Der Job ist an keine Serialisierungskennung angeschlossen.

- Der Job hat keine Datei-Seiten gesperrt.

Rückgang der Sättigung

Grad	Auswirkungen im System	Maßnahme des Operators
1	Level=1 zeigt lediglich an, dass der erste Job in der Warteschlange wieder eingelagert wurde, es aber noch weitere ausgelagerte Jobs gibt.	Keine
0	Alle ausgelagerten Jobs werden wieder bedient. Es können wieder Jobs gestartet werden.	Keine

Maßnahmen zur Verhinderung der Sättigung

Das Auftreten eines Sättigungszustandes zeigt an, dass das System überlastet ist.

Wenn das System immer wieder in einen Sättigungszustand kommt, ist der Seitenwechselspeicher zu klein oder die Maßnahmen zur Lastbegrenzung waren unzureichend.

- Vergrößern Sie den Seitenwechselspeicher.
- Sorgen Sie dafür, dass nachfolgende Systemläufe bei der Systemeinleitung einen ausreichend großen Seitenwechselspeicher in Betrieb nehmen (Parameterservice).
- Überprüfen Sie Ihre Maßnahmen zur Lastbegrenzung.

Beispiele zur Verringerung des Seitenwechselspeicherbedarfs eines Systems (falls Erweiterung nicht gewünscht wird)

- Entladen großer Anwendungen bzw. Programme
- Deutliche Verringerung der Taskanzahl im System

Einstellungen für Schwellwerte und Warnungen

Die Schwellwerte für die Sättigungszustände und Warnungen vor dem Erreichen von Sättigungsstufen können Sie im Startup-Parameterservice, Parametersatz MEMORY konfigurieren, siehe Abschnitt „Speicherverwaltung (MEMORY)".

5 Geräteverwaltung

Die Geräteverwaltung von BS2000 (Nucleus Device Management, NDM) verwaltet die periphere Konfiguration einer Server Unit. Die Kanalperipherie der SU /390 wird bei der Hardware-Generierung mit IOGEN (siehe Handbuch „Systeminstallation“ [55]) vorgegeben. Die Bus- und Fibre Channel-Peripherie der SU x86 wird über X2000 bekannt gemacht.

Die Geräteverwaltung umfasst folgende Funktionen:

- Sie steuert die Verfügbarkeit aller Hardware-Einheiten und deren Verbindungen für das System.
- Sie stellt Belegungs- und Freigabemechanismen für Geräte und Datenträger zur Verfügung.
- Sie wickelt Montieraufträge für Datenträger ab und schützt diese Datenträger vor Fehleingriffen.
- Sie ermöglicht die Reservierung von Geräten, Datenträgern und Dateien unter Berücksichtigung von Auftragsprioritäten.
- Sie informiert über Belegungs- und Verfügbarkeitszustände der Konfiguration.

In diesem Kapitel werden die Konfiguration, Rekonfiguration und die dynamische I/O-Konfigurationsänderung für SU /390 ausführlich behandelt. Für SU x86 werden jeweils die Besonderheiten herausgestellt.

Danach werden die Komponente NDM (Nucleus Device Management) zur Betriebsmittelbelegung und -reservierung sowie die Überwachung, Auswahl und Bedienung von gemeinschaftlichen Datenträgern beschrieben.

Es folgen detaillierte Informationen zur Fehlerbehandlung ausgewählter Gerätesteuerungen und zur Verwaltung privater Datenträger. Danach folgen Hinweise zu dem Dienstprogramm IORM .

5.1 Bestandteile von Konfigurationen

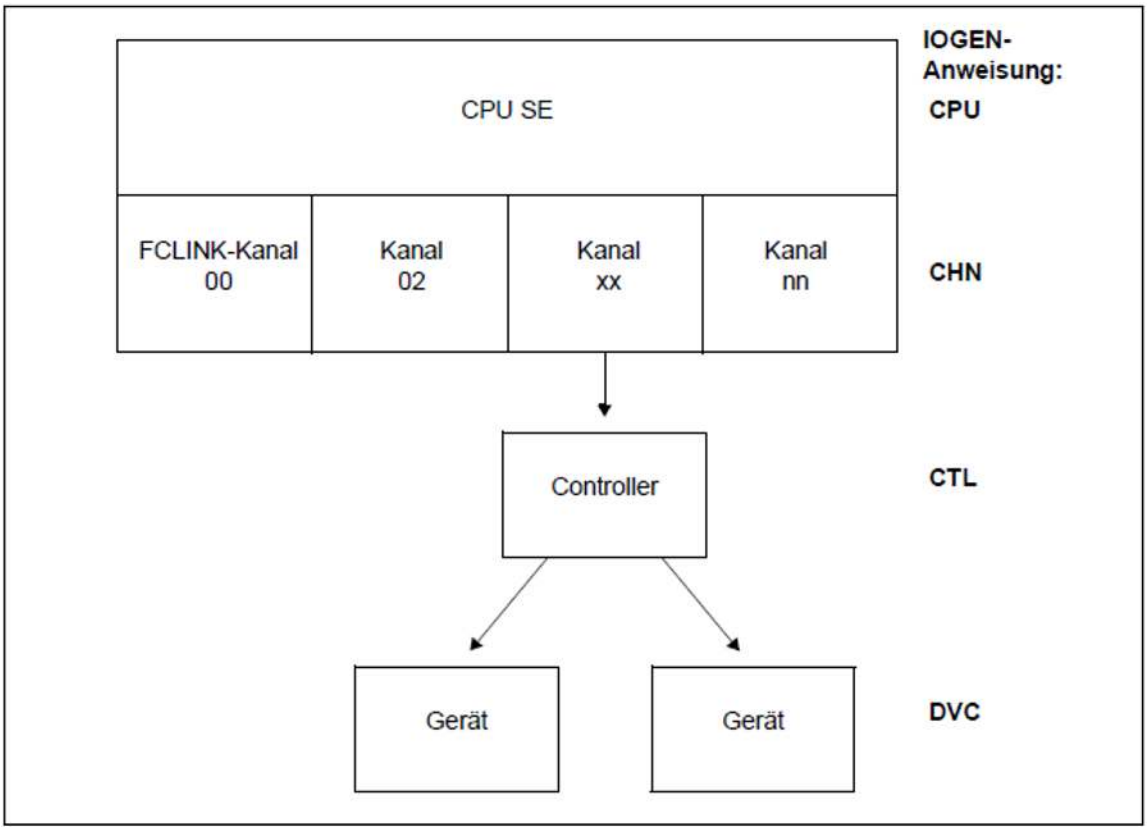


Bild 3: Ausschnitt einer Konfiguration (SU /390)

Schnittstellen zur Verwaltung der Konfiguration

Kommando	Bedeutung
ADD-IO-UNIT	neue Ein-/Ausgabe-Einheiten dynamisch hinzufügen [1]
ATTACH-DEVICE	Zuschalten einer Hardware-Einheit, bei Kanälen und Controllern lokal oder VM2000-global
DETACH-DEVICE	Wegschalten einer Hardware-Einheit, bei Kanälen und Controllern lokal oder VM2000-global
INCLUDE-DEVICE-CONNECTION	Zuschalten einer logischen Verbindung lokal oder VM2000-global
MODIFY-IO-OPTIONS	FastDPAV Einstellungen für Geräte ändern, Timeout-Einstellungen ändern [2]
MODIFY-IO-UNIT	Pfadbeschreibung einer Steuerung ändern, [1] bevorzugten Pfad eines Gerätes ändern, [1] bevorzugtes PAV-Gerät definieren, Timeout-Einstellungen ändern [2]

REMOVE-DEVICE-CONNECTION	Wegschalten einer logischen Verbindung lokal oder VM2000-global
REMOVE-IO-UNIT	Ein-/Ausgabe-Einheiten dynamisch entfernen [1]
SECURE-RESOURCE-ALLOCATION	Anfordern von Betriebsmitteln
SHOW-DEVICE-CONFIGURATION	Abfragen der Ein-/Ausgabe-Konfiguration
START-CONFIGURATION-UPDATE	dynamische I/O-Konfigurationsänderung einleiten
STOP-CONFIGURATION-UPDATE	dynamische I/O-Konfigurationsänderung beenden, aktuelles IORSF sichern
UNLOCK-DEVICE	Rücksetzen der Hardware-Reservierung
Makro	**Bedeutung**
NKDINF	Informieren über den Belegungs- und Verfügbarkeitszustand der (peripheren) Konfiguration

Tabelle 9: Schnittstellenübersicht zur Verwaltung der Konfiguration

[1] Diese Funktionen sind nur verfügbar, wenn eine dynamische I/O-Konfigurationsänderung an SU /390 eingeleitet ist.

[2] Diese Funktion ist auch außerhalb einer dynamischen Rekonfiguration unabhängig von der Architektur der Server Unit verfügbar.

5.1.1 Hardware-Einheiten

Arten von Hardware-Einheiten

In einer Konfiguration gibt es folgende Arten von Hardware-Einheiten:

- die CPUs (Central Processor Unit)
- die Kanäle (Channel, CHN), die mit der IOGEN-Anweisung CHN definiert werden
- die Mehrgerätesteuerungen (Controller, CTL), die mit der IOGEN-Anweisung CTL festgelegt werden (Band- und Plattensteuerungen)
- die Geräte (Device, DVC), definiert mit der IOGEN-Anweisung DVC

> Auf SU x86 sind in BS2000 nur Kanäle (CHN) und Geräte (DVC), jedoch keine Controller (CTL) bekannt und sichtbar.

Man unterscheidet bezüglich der logischen Verbindungen zwischen inneren und äußeren Einheiten, die der Ein-/Ausgabe dienen.

Die Kanäle (CHN) sind die innersten, die Geräte (DVC) die äußersten Hardware-Einheiten. Die verschiedenen Hardware-Einheiten stehen, von innen nach außen, in der Reihenfolge CHN – CTL – DVC

Von zwei logisch direkt verbundenen Hardware-Einheiten, z.B. CTL – DVC, ist CTL die innere, DVC die äußere Einheit.

Zustände von Hardware-Einheiten

Diese Hardware-Einheiten können folgende Zustände haben:

ATTACHED

Die Hardware-Einheit ist zum System zugeschaltet und kann für Ein-/Ausgaben benutzt werden. Dieser Zustand ist eingestellt:

- nach der Systemeinleitung, wenn die entsprechende Einheit bei der Hardware-Generierung mit IOGEN mit dem Attribut „A“ generiert wurde (z.B. DVC A0,CB,**A**,(...))
- nach der Systemeinleitung, wenn die entsprechende Einheit im Startup-Parameterservice mit der IOCONF-Anweisung MOD-IO-UNIT ..., STATE=ATT geändert wurde
- nach erfolgreichem Aufruf des Kommandos ATTACH-DEVICE (ATT)
- nach erfolgreichem Aufruf des Kommandos ADD-IO-UNIT (mit STATE=*ATTACHED)

ATTACH-PENDING

Für Platten- oder Bandgeräte erfolgt eine ATTACH-Bearbeitung. Die Hardware-Einheit kann noch nicht für Ein-/Ausgaben verwendet werden. Wird der Rekonfigurationsauftrag ausgeführt, geht die Hardware-Einheit in den Zustand ATTACHED über.
Den Zustand DETACHED erhält die Hardware-Einheit, wenn der Rekonfigurationsauftrag nicht ausgeführt werden kann (vergleiche Abschnitt „Besonderheiten für Bandgeräte“, "Besonderheiten für Magnetband- und Plattengeräte").

Dieser Zustand ist während der Durchführung von Ein-/Ausgaben zur Ermittlung von Steuerungs- und Gerätedaten sowie zur Reservierung von MBK-Geräten eingestellt.

DETACHED

Die Hardware-Einheit ist vom System weggeschaltet und kann nicht für Ein-/Ausgaben benutzt werden. Der Zustand DETACHED wird weiter unterteilt in:

- DETACHED EXPLICITLY. Dieser Zustand ist eingestellt:
  - nach der Systemeinleitung, wenn die entsprechende Einheit bei der Hardware-Generierung mit IOGEN mit dem Attribut „D“ generiert wurde (z.B. DVC_A0,CB,**D**,(...))
  - nach der Systemeinleitung, wenn die entsprechende Einheit im Startup-Parameterservice mit der IOCONF-Anweisung MOD-IO-UNIT ..., STATE=DET geändert wurde
  - nach einem erfolgreichen Aufruf des Kommandos DETACH-DEVICE (DET)
  - nach einer automatischen Rekonfiguration durch das Betriebssystem
  - nach der Systemeinleitung, wenn eine Hardware-Einheit als „offline“ erkannt wird
- DETACHED IMPLICITLY
  Alle inneren Verbindungen der entsprechenden Einheit befinden sich im Zustand REMOVED.

DETACH-PENDING

Die Einheit ist noch von einer Task oder vom System belegt und steht nach Belegungsende nicht mehr zur Verfügung. Weitere Belegungsversuche werden nicht mehr zugelassen. Der Zustand DETACH-PENDING tritt i.A. nur während einer angegebenen Wartezeit (Operand FORCE=*NO) im DETACH-DEVICE-Kommando auf. Er kann auch dann auftreten, wenn ein DETACH-DEVICE-Kommando mit Operand FORCE=*YES für ein Magnetbandkassetten-Gerät abgesetzt wurde (vergleiche Abschnitt „Besonderheiten für Bandgeräte“, "Besonderheiten für Magnetband- und Plattengeräte").
Kann der Rekonfigurationsauftrag während der Wartezeit durchgeführt werden, geht die Einheit in den Zustand DETACHED über, sonst wird (wieder) der Zustand ATTACHED eingestellt. Der Zustand DETACH-PENDING wird unterteilt in:

- DETACH-PENDING EXPLICITLY
  Dieser Zustand wird eingestellt, wenn die Einheit direkt weggeschaltet wird.
- DETACH-PENDING IMPLICITLY
  Dieser Zustand wird eingestellt, wenn keine innere Verbindungen der Einheit im Zustand INCLUDED und zumindest eine im Zustand REMOVE-PENDING ist.

INVALID

Das Gerät kann nicht benutzt und auch nicht rekonfiguriert (zugeschaltet) werden.
Beispiel: Das Gerät wurde mit REMOVE-IO-UNIT entfernt.

ENABLED

Zustand für FastDPAV-Alias-Geräte: Das FastDPAV-Alias-Gerät kann für Ein-/Ausgaben benutzt werden.
Der Zustand ist eingestellt:

- wenn im Native-Betrieb das FastDPAV-Alias-Gerät nicht explizit weggeschaltet wurde (Kommando DETACH-DEVICE)
- wenn im VM-Betrieb das FastDPAV-Alias-Gerät der virtuellen Maschine zugewiesen und danach nicht explizit weggeschaltet wurde (Kommando DETACH-DEVICE)

DISABLED

Zustand für FastDPAV-Alias-Geräte: Das FastDPAV-Alias-Gerät kann nicht für Ein-/Ausgaben benutzt werden. Der Zustand ist eingestellt:

- wenn im Native-Betrieb das FastDPAV-Alias-Gerät explizit weggeschaltet wurde (Kommando DETACH-DEVICE)
- wenn im VM-Betrieb das FastDPAV-Alias-Gerät nicht der virtuellen Maschine zugewiesen wurde oder danach explizit weggeschaltet wurde (Kommando DETACH-DEVICE)

5.1.2 Logische, rekonfigurierbare Verbindungen

Arten von logischen, rekonfigurierbaren Verbindungen

In einer Konfiguration bestehen folgende logische (rekonfigurierbare) Verbindungen:

- zwischen den Kanälen und den direkt daran generierten Geräten
- zwischen den Kanälen und den daran generierten Steuerungen
- zwischen den Steuerungen und den daran generierten Geräten

Man unterscheidet innere und äußere logische Verbindungen:

- Eine innere Verbindung geht immer in Richtung zu einer inneren Einheit (z.B. von DVC zu CTL oder von CTL zu CHN).
- Eine äußere Verbindung geht in Richtung zu einer äußeren Einheit (CHN zu CTL oder CTL zu DVC).

Zustände von logischen, rekonfigurierbaren Verbindungen

INCLUDED

Die logische Verbindung ist zum System zugeschaltet und kann für Ein-/Ausgaben benutzt werden. Dieser Zustand ist eingestellt:

- nach der Systemeinleitung, wenn nicht eine der Bedingungen für den Zustand REMOVED vorliegt (siehe unten)
- nach erfolgreichem Aufruf des Kommandos INCLUDE-DEVICE-CONNECTION (INC)

REMOVED

Die logische Verbindung ist vom System weggeschaltet und kann nicht für Ein-/Ausgaben benutzt werden. Der Zustand REMOVED wird weiter unterteilt in:

- REMOVED EXPLICITLY
  Dieser Zustand wird eingestellt:
  - nach der Systemeinleitung, wenn die entsprechende Einheit im Startup-Parameterservice mit der IOCONF-Anweisung MOD-IO-UNIT *CON(..., STATE=REM) geändert wurde
  - nach einer automatischen Rekonfiguration durch das Betriebssystem
  - nach erfolgreichem Aufruf des Kommandos REMOVE-DEVICE-CONNECTION
- REMOVED IMPLICITLY
  Dieser Zustand wird eingestellt, wenn die innere Hardware-Einheit im Zustand DETACHED ist.

REMOVE-PENDING

Die logische Verbindung ist noch belegt und steht danach dem System nicht mehr zur Verfügung. Dieser Zustand tritt nur während einer Wartezeit in den Kommandos REMOVE-DEVICE-CONNECTION bzw. DETACH-DEVICE auf. Kann der Rekonfigurationsauftrag durchgeführt werden, geht die Verbindung in den Zustand REMOVED über, sonst wird (wieder) der Zustand INCLUDED eingestellt. REMOVE-PENDING wird unterteilt in:

- REMOVE-PENDING EXPLICITLY
  Dieser Zustand wird eingestellt, wenn die Verbindung direkt (mit Wartezeit: Operand FORCE=NO im Kommando REMOVE-DEVICE-CONNECTION) weggeschaltet wird.
- REMOVE-PENDING IMPLICITLY
  Dieser Zustand wird eingestellt, wenn die innere Einheit dieser Verbindung im Zustand DETACH-PENDING ist.

5.2 Rekonfiguration

Unter Rekonfiguration versteht man das Zu- oder Wegschalten von Komponenten der Konfiguration einer Server Unit. Eine Konfiguration besteht aus Hardware-Einheiten und deren logischen Verbindungen.

Dazu stehen dem Operator folgende sog. Rekonfigurationskommandos zur Verfügung:

Kommando	**Bedeutung**
ATTACH-DEVICE	Zuschalten einer Hardware-Einheit, bei Kanälen und Controllern lokal oder VM2000-global
DETACH-DEVICE	Wegschalten einer Hardware-Einheit, bei Kanälen und Controllern lokal oder VM2000-global
INCLUDE-DEVICE-CONNECTION	Zuschalten einer logischen Verbindung lokal oder VM2000-global
REMOVE-DEVICE-CONNECTION	Wegschalten einer logischen Verbindung lokal oder VM2000-global

Tabelle 10: Rekonfigurationskommandos

Weitere für die Rekonfiguration relevante Kommandos und Makros sind der Tabelle 12 (NDM - Betriebsmittelbelegung und -reservierung) zu entnehmen.

5.2.1 Rekonfiguration bei Multiprozessoren

Das System reagiert automatisch auf Hardware-Ausfälle. Wenn eine Komponente (CPU, Channel) ausfällt, wird sie vom Betriebssystem weggeschaltet. Der Betrieb wird auf den verbleibenden Komponenten weitergeführt.

Der Operator kann Komponenten mittels Rekonfigurationskommandos zu- bzw. wegschalten. Wenn eine Komponente zu Wartungszwecken oder aus anderen Gründen deaktiviert werden soll, kann sie der Operator mit dem DETACH-DEVICE-Kommando wegschalten. Der Restbetrieb wird dann auf den verbleibenden Komponenten weitergeführt. Soll eine weggeschaltete Komponente wieder verfügbar gemacht werden, kann sie der Operator mit dem Kommando ATTACH-DEVICE zuschalten.

Die Rekonfiguration schlägt fehl, wenn einer der folgenden Umstände vorliegt:

- nach dem Wegschalten einer CPU ist nicht noch mindestens eine CPU arbeitsfähig
- bei Ausfall einer CPU sind Daten des Betriebssystems inkonsistent und die Hardware konnte keinen gültigen Hardware-Kontext mehr zur Verfügung stellen
- gemeinschaftliche Datenträger oder die letzte Konsole sind durch den Ausfall betroffen

Bei bestimmten Geräten muss nach einer Rekonfiguration ein Umschalten bzw. Einschalten durchgeführt werden, damit sie wieder vom System angesprochen werden können.
Mit Umschalten ist in diesem Fall nicht nur das physikalische Umschalten des Gerätes auf einen Alternativpfad gemeint, obwohl dies eine Voraussetzung dafür ist. Umschalten beinhaltet alle Aktionen die erforderlich sind, damit das System die Geräte wieder ansprechen kann. Um- bzw. Einschalten liegt in der Verantwortung des Operators.

> **i** Tritt bei Server Units mit mehr als einer CPU in einer der zugeschalteten CPUs ein Fehler auf, so wird in vielen Fällen die Server Unit automatisch rekonfiguriert. Nur in den Fällen, bei denen ein gültiger Programm-Kontext der ausgefallenen CPU nicht hergestellt werden kann und der Lock-Mechanismus des Betriebssystems aktiv ist, kann keine Rekonfiguration durchgeführt werden.

5.2.2 Weg- und Zuschalten der Komponenten CPU, CHN, CTL und DVC

Wegschalten von Komponenten durch das Betriebssystem

Bei Ausfall werden die Komponenten automatisch weggeschaltet. Der Operator erhält dann folgende Meldung an der Hauptkonsole:

NKR0048	'CHANNEL=mn' DETACHED BY SYSTEM **********	(für CHN)
ETMRK48	CPUxy DETACHED BY SYSTEM **********	(für CPU)

mn Mnemotechnischer Gerätename der Komponente

Die Meldung ETMRK48 muss vom Operator quittiert werden. Dies soll verhindern, dass ein Wegschalten übersehen wird.

Die Meldung wird an der Hauptkonsole ausgegeben, die am intakten Ein-/Ausgabe-Prozessor angeschlossen ist.

Das (Wieder-)Zuschalten erfolgt mittels ATTACH-DEVICE durch den Operator, ist aber nicht in jedem Fall möglich.

Wegschalten von Komponenten durch den Operator

Der Operator kann Komponenten mit dem Kommando DETACH-DEVICE (siehe Handbuch „Kommandos“ [27]) wegschalten.

Das ordnungsgemäße Wegschalten bestätigt das Betriebssystem für jede Komponente mit folgender Meldung an der Konsole:

NKR0045	'CHANNEL=mn' DETACHED **********	(für CHN)
ETMRK19	CPUxy DETACHED **********	(für CPU)

Das Kommando DETACH-DEVICE wird mit einer entsprechenden Meldung zurückgewiesen, wenn es momentan nicht ausgeführt werden kann.
Bei Kanälen und Gerätesteuerungen unter VM2000 auf SU /390 kann im Monitorsystem das Wegschalten nur im lokalen System oder in allen BS2000-Gastsystemen (VM2000-global) erfolgen. Das Kommando muss dann im Monitorsystem eingegeben werden.

Das (Wieder-)Zuschalten erfolgt mittels ATTACH-DEVICE.

Zuschalten von Komponenten durch den Operator

Der Operator kann Komponenten in folgenden Fällen mittels ATTACH-DEVICE zuschalten:

- wenn eine Komponente nach Ausfall oder Wartung wieder verfügbar ist
- wenn die Systemeinleitung ohne eine Komponente durchgeführt wurde

Bei Kanälen und Gerätesteuerungen unter VM2000 kann im Monitorsystem das Zuschalten nur im lokalen System oder in allen BS2000-Gastsystemen (VM2000-global) erfolgen.

Das Zuschalten bestätigt das Betriebssystem für jede Komponente durch folgende Meldung an der Konsole:

NKR0040	'<unit-class>=mn' ATTACHED	(für CHN, CTL, DVC)

ETMRK18	CPUxy ATTACHED	(für CPU)

Zu- und Wegschalten von Extra-CPUs

Extra-CPUs kann es an jeder Server Unit geben.

Der Operator kann eine oder mehrere Extra-CPUs im laufenden Betrieb zuschalten (Kommando ATTACH-DEVICE UNIT=*EXTRA-CPU(CPU-ID=...)).

Das Wegschalten der Extra-CPUs erfolgt mit dem Kommando DETACH-DEVICE.

Für ausführliche Informationen zum Zu- und Wegschalten von Extra-CPUs siehe "Zu- und Wegschalten von Extra-CPUs" im "Funktionalität im native BS2000-Betrieb".

5.2.3 Wirkung der Rekonfigurationskommandos

Ein Rekonfigurationskommando bleibt ohne Wirkung, wenn ein Zustand eingestellt werden soll, der bereits eingestellt ist oder der nicht eingestellt werden kann. Es werden entsprechende Meldungen ausgegeben.

Das Zu- und Wegschalten von Kanälen, Steuerungen und Verbindungen unter VM2000 kann im Monitorsystem nur für das lokale System oder für alle BS2000-Gastsysteme (VM2000-global) erfolgen.
Dafür wurde der Operand SCOPE=OWN-SYSTEM-ONLY/VM2000-GLOBAL eingeführt.

ATTACH-DEVICE

Wird ein gültiges Kommando ATTACH-DEVICE eingegeben, so hat das folgende Auswirkungen:

Waren die angegebenen Hardware-Einheiten im Zustand DETACHED EXPLICITLY, so werden sie ATTACHED. Die Hardware-Einheiten können wieder benutzt werden.

Waren die zu diesen Hardware-Einheiten gehörenden äußeren Verbindungen REMOVED IMPLICITLY, so werden sie INCLUDED. Die logischen Verbindungen können wieder benutzt werden.

Die mit diesen Einheiten verbundenen äußeren Hardware-Einheiten, die sich im Zustand DETACHED IMPLICITLY befanden, werden ATTACHED, vorausgesetzt, mindestens eine logische Verbindung zu der äußeren Einheit ist im Zustand INCLUDED.

DETACH-DEVICE

Ein gültiges Kommando DETACH-DEVICE wirkt sich folgendermaßen aus:

Jede der angegebenen Hardware-Einheiten geht in den Zustand DETACHED EXPLICITLY über. Sie können nicht benutzt werden.

Alle zu diesen Hardware-Einheiten gehörenden äußeren Verbindungen erhalten den Zustand REMOVED IMPLICITLY. Sie können nicht benützt werden.

Alle mit diesen Einheiten verbundenen äußeren Hardware-Einheiten gehen in den Zustand DETACHED IMPLICITLY, wenn die Verbindung zu der direkt weggeschalteten Einheit die letzte oder die einzige logische Verbindung war. Diese äußeren Hardware-Einheiten können nicht benützt werden.

Das Wegschalten mit dem Parameter SCOPE=*VM2000-GLOBAL ist nur dann erfolgreich, wenn in allen BS2000-Gastsystemen das Wegschalten erfolgreich durchgeführt werden konnte.

Mit dem Operanden FORCE=*UNCONDITIONAL-OFFLINE unter VM2000 wird im Monitorsystem das unbedingte Wegschalten von Kanälen ermöglicht (SU /390).

INCLUDE-DEVICE-CONNECTION

Die Eingabe eines gültigen Kommandos INCLUDE-DEVICE-CONNECTION hat folgende Wirkung:

Waren die angegebenen logischen Verbindungen im Zustand REMOVED EXPLICITLY, so werden sie INCLUDED. Die Verbindungen können wieder benützt werden.

Waren die zu den Verbindungen gehörenden äußeren Einheiten DETACHED IMPLICITLY, so erhalten sie den Zustand ATTACHED. Diese Hardware-Einheiten können wieder benützt werden.

Die von diesen Hardware-Einheiten ausgehenden äußeren Verbindungen werden INCLUDED, wenn sie vorher REMOVED IMPLICITLY waren. Diese äußeren Verbindungen können wieder benützt werden.

REMOVE-DEVICE-CONNECTION

Eine gültige Eingabe des Kommandos REMOVE-DEVICE-CONNECTION wirkt sich so aus:

Die angegebenen logischen Verbindungen gehen in den Zustand REMOVED EXPLICITLY. Sie können nicht mehr benützt werden.

Die an diesen Verbindungen angeschlossenen äußeren Hardware-Einheiten erhalten den Zustand DETACHED IMPLICITLY, falls die weggeschaltete Verbindung die letzte oder einzige Verbindung nach innen war. Diese Hardware-Einheiten können nicht mehr benützt werden.

Die von diesen äußeren Hardware-Einheiten im Zustand DETACHED IMPLICITLY ausgehenden äußeren Verbindungen erhalten den Zustand REMOVED IMPLICITLY. Sie können nicht mehr benützt werden.

Anschlüsse für KVP-Konsolen

Die beiden Ein-/Ausgabe-Anschlüsse (Gerätetyp SKP2) bilden für das Betriebssystem eine logische Einheit. Bei ATTACH/DETACH-Aufträgen sorgt es dafür, dass beide Anschlüsse den gleichen Zustand einnehmen. Voraussetzung ist, dass auch die beiden Pfade von der Steuerung zu den Anschlüssen im gleichen Zustand sind. Werden mit INCLUDE-/REMOVE-DEVICE-CONNECTION die Pfade in unterschiedliche Zustände gebracht, ist eine ordnungsgemäße Behandlung der Anschlüsse durch das System nicht mehr gewährleistet.

> **i**
>
> Das Einschalten von Hardware-Einheiten und Verbindungen liegt in der Verantwortung des Operators. Das Ausschalten von Hardware-Einheiten und Verbindungen erfolgt entweder vom Operator oder (bei schwerwiegenden Fehlern) vom System.
>
> Der Operator ist verantwortlich für die Koordination des Zu- und Wegschaltens von Hardware-Einheiten, die an mehreren Systemen angeschlossen sind.
>
> Geräte im Zustand DETACH-PENDING können von der Datenträgerüberwachung bei Remount-Aufträgen verwendet werden, falls sonst keine freien Geräte verfügbar sind.

FastDPAV-Alias-Geräte

Nach erfolgreicher Ausführung des Kommandos ATTACH-DEVICE für ein FastDPAV-Alias-Gerät ist der Zustand ENABLED eingestellt. Das bedeutet, dass das FastDPAV-Alias-Gerät für Ein-/Ausgaben benutzt werden kann.

Nach erfolgreicher Ausführung des Kommandos DETACH-DEVICE für ein FastDPAV-Alias-Gerät ist der Zustand DISABLED eingestellt. Das bedeutet, dass das FastDPAV-Alias-Gerät für Ein-/Ausgaben nicht benutzt werden kann.

5.2.4 Besonderheiten für Magnetband- und Plattengeräte

Bei der System-Initialisierung werden Magnetbandgeräte vom System zunächst explizit DETACHED.

Für ATTACHED-generierte Plattengeräte und Plattengeräte, die zum Home-Pubset gehören, werden über Ein-/Ausgabe-Befehle Geräte- und Steuerungseigenschaften ermittelt. Ein Gerät wird vom System explizit DETACHED, wenn diese Eigenschaften wegen Ein-/Ausgabe-Fehlern nicht ermittelt werden können.

Beim ATTACH eines Magnetband- oder Plattengerätes erfolgt ebenso zunächst die Ermittlung von Geräte- und Steuerungseigenschaften. Ist eine erfolgreiche Ermittlung nicht möglich, wird der ATTACH abgewiesen.

Ausnahme

> Wurde die Ermittlung der Eigenschaften für Magnetbandgeräte wegen `NO PATH AVAILABLE` (CC=3) abgewiesen, wird der ATTACH zugelassen.
> Damit ist eine Überkonfigurierung zum Zweck einer Überbelegung möglich.

Besonderheiten für Bandgeräte

Bandgeräte (MBK-Geräte und emulierte Bandgeräte) werden nur dann ATTACHED, wenn sie hardwaremäßig für das eigene System reserviert werden können.
Nach erfolgreichem Zuschalten sind Bandgeräte einem System exklusiv zugeordnet.

ATTACH-DEVICE

Beim ATTACH eines Bandgerätes wird der Versuch unternommen, das Gerät zu reservieren. Es wird die Meldung `NKR0042 DEVICE=mn ATTACH ACCEPTED` ausgegeben und das Gerät geht in den Zustand ATTACH-PENDING über.

Nach erfolgreicher Reservierung erfolgt die Meldung `NKR0110 DEVICE=mn DEVICE ATTACHED AND ASSIGNED` und das Bandgerät geht in den Zustand ATTACHED über.

Ist das Bandgerät von einem fremden System aus reserviert, so erfolgen die Meldungen `NKR0111 DEVICE=mn DEVICE ASSIGNED ELSEWHERE` und `NKR0044 DEVICE=mn ATTACHMENT REJECTED` und das Bandgerät geht in den Zustand DETACHED-EXPLICITLY über.

Kann das Bandgerät aus anderen Gründen nicht zugewiesen werden, erfolgt die Meldung `NKR0112 DEVICE=mn COULD NOT BE ASSIGNED`.

Ein Dummy-Bandgerät wird auch dann ATTACHED, wenn die Zuweisung wegen Nichtverfügbarkeit von Gerät oder Gerätesteuerung abgewiesen wird. In diesem Fall werden statt der Meldung `NKR0110` folgende Meldungen ausgegeben:

```
NKR0112 DEVICE=mn COULD NOT BE ASSIGNED
NKR0040 DEVICE=mn ATTACHED
```

DETACH-DEVICE

Bei DETACH (explizit und implizit) wird versucht, die Hardware-Reservierung zurückzusetzen:

- Während dieser Zeit sind alle vom Rekonfigurationsauftrag betroffenen Geräte/Pfade im Zustand DETACH-/REMOVE-PENDING.
- Nach erfolgreicher Ausführung erfolgt die Meldung `NKR0114 DEVICE=mn DEVICE UNASSIGNED` oder, falls die Hardware-Reservierung nicht freigegeben werden konnte, die Meldung `NKR0115 DEVICE=mn COULD NOT BE UNASSIGNED`.
- Ein ATTACH-DEVICE-Kommando während der Wartezeit bewirkt den Versuch einer erneuten Zuweisung.

Auswirkungen eines DETACH-DEVICE/REMOVE-DEVICE-CONNECTION-Kommandos mit dem Operanden FORCE=*YES auf die Steuerung:

1. Die Steuerung quittiert Schreibaufträge, wenn die zu übertragenden Daten in einem Puffer der Steuerung abgelegt sind, und überträgt dann diese Daten asynchron auf den Datenträger.
   Es wird maximal 2 Minuten auf die erfolgreiche Übertragung auf das Bandgerät bzw. auf das Wiedereinlesen der nicht übertragbaren Daten gewartet. Es wird eine der folgenden Meldungen ausgegeben:
   `NKR0020 DETACH DEVICE ACCEPTED`
   `NKR0021 REMOVE DEVICE CONNECTION ACCEPTED`

   Während dieser Wartezeit befindet sich das Bandgerät im Zustand DETACH-PENDING.
2. Spätestens nach Ablauf dieser Wartezeit wird der Versuch unternommen, die Hardware-Reservierung freizugeben.
3. Ein ATTACH-DEVICE-Kommando während der Wartezeit bewirkt ein Aufheben des DETACH-Auftrags.

Bei Systembeendigung wird der Versuch unternommen, alle (noch) zum System bestehenden Hardware-Reservierungen freizugeben. Ein Scheitern der Freigabe wird durch folgende Meldung angezeigt:
`NKR0115 DEVICE=mn COULD NOT BE UNASSIGNED`

Wird ein Zuschaltversuch mit der Meldung `NKR0111 DEVICE=mn DEVICE ASSIGNED ELSEWHERE` abgewiesen, so ist das Bandgerät von einem fremden System reserviert. Mit dem Kommando UNLOCK-DEVICE kann dem fremden System die Reservierung entzogen werden. Nach einem Systemabsturz kann mit diesem Kommando das Bandgerät freigeschaltet werden.

5.3 Dynamische I/O-Konfigurationsänderung (SU /390)

Rekonfiguration im laufenden Betrieb ist das Zu- und Wegschalten bereits definierter Komponenten der Konfiguration einer Server Unit (siehe Abschnitt „Rekonfiguration").

Darüber hinaus besteht die Möglichkeit, Ein-/Ausgabe-Einheiten während des laufenden Betriebs der Konfiguration der Server Unit hinzuzufügen bzw. daraus zu entfernen.

Die Ein-/Ausgabe-Konfiguration kann im laufenden Betrieb geändert werden, d.h. die bestehende Konfiguration kann ohne neuen IMPL erweitert oder reduziert werden. Die Änderungen werden direkt auf dem aktiven IORSF der Server Unit ausgeführt.

Diese Möglichkeit der Konfigurationsänderung wird im Folgenden als „dynamische I/O-Konfigurationsänderung“ bezeichnet. Die dynamische I/O-Konfigurationsänderung wird für folgende Geräteklassen unterstützt:

- Platten- und Bandgeräte
- Netzwerk-/LAN-Geräte

Vorbereitung der dynamischen I/O-Rekonfiguration

Damit im laufenden Betrieb neue Geräte und Steuerungen der Konfiguration hinzugefügt werden können, erzeugt das System bei der Systemeinleitung in der Gerätetabelle zusätzliche freie Tabelleneinträge für Geräte und Steuerungen. Die Tabelleneinträge für Kanäle und Verbindungen zwischen Ein-/Ausgabe-Einheiten werden dynamisch angelegt und müssen von der Systembetreuung bei der Generierung des IORSF nicht mit berücksichtigt werden.

Die Anzahl der freien Tabelleneinträge für Geräte und Steuerungen richtet sich nach der Größe des Reservebereichs im IORSF.
Dieser reicht bei einem mit IOGEN erstellten IORSF für 512 zusätzliche Einträge (Kanäle, Steuerungen, Geräte und deren Verbindungen), siehe Handbuch „Systeminstallation“ [55]. Ein bereits modifiziertes IORSF kann weniger oder mehr freie Einträge besitzen, entweder durch Hinzufügen oder Wegnahme von Einheiten im aktiven IORSF oder durch Zurückschreiben des aktiven IORSF mit dem Kommando STOP-CONFIGURATION-UPDATE ..., DEVICE-RESERVE=.

Bei der Systemeinleitung werden für Netzwerk-/LAN-Geräte 64 freie Einträge in der Gerätetabelle vorgesehen, für Platten- und Bandgeräte jeweils die aufgrund der freien Einträge im IORSF maximal mögliche Anzahl (höchstens aber je 2048). Die Gesamtzahl der dynamisch hinzufügbaren Einheiten kann dabei nicht die im IORSF gesetzten Grenzen überschreiten.

Maßnahmen zur Vermeidung von Engpässen

Jedes Gerät oder jede Steuerung, die der Konfiguration neu hinzugefügt wird, belegt einen freien Tabelleneintrag. Das Hinzufügen wird abgewiesen, wenn kein freier Tabelleneintrag mehr existiert. In diesem Fall ist ein Hinzufügen erst wieder möglich, wenn eine entsprechende Ein-/Ausgabe-Einheit entfernt wird.

Ist bereits bei der Generierung des IORSF abzusehen, dass diese freien Tabelleneinträge für die in der nächsten Session vorgesehene dynamische I/O-Konfigurationsänderung nicht ausreichen werden, sollte eine genügend große Anzahl zukünftiger Ein-/Ausgabe-Einheiten im IORSF mitkonfiguriert werden. Diese zusätzlichen Ein-/Ausgabe-Einheiten können dann zum Zeitpunkt der Inbetriebnahme mit den Kommandos der dynamischen Rekonfiguration entsprechend der realen Konfiguration umdefiniert werden.

Dynamische I/O-Konfigurationsänderung vornehmen

Folgende Kommandos stehen zur dynamischen I/O-Konfigurationsänderung zur Verfügung. Die Kommandos sind im Handbuch „Kommandos“ [27] ausführlich beschrieben:

Kommando	Bedeutung
ADD-IO-UNIT	neue Ein-/Ausgabe-Einheiten definieren
MODIFY-IO-UNIT	Pfadbeschreibung einer Steuerung ändern, bevorzugten Pfad eines Gerätes ändern, Timeout-Einstellungen ändern, bevorzugtes Gerät für PAV festlegen
REMOVE-IO-UNIT	Ein-/Ausgabe-Einheiten entfernen
SHOW-DEVICE-CONFIGURATION	Abfragen der Ein-/Ausgabe-Konfiguration
START-CONFIGURATION-UPDATE	dynamische I/O-Konfigurationsänderung einleiten
STOP-CONFIGURATION-UPDATE	dynamische I/O-Konfigurationsänderung beenden, aktuelles IORSF sichern

Tabelle 11: Komandos der dynamischen I/O-Konfigurationsänderung

Bevor Änderungen an der Konfiguration vorgenommen werden, muss die dynamischeI/O-Konfigurationsänderung mit dem Kommando START-CONFIGURATION-UPDATE eingeleitet werden. Die Kommandos zur Änderung der Konfiguration werden erst nach erfolgreicher Einleitung angenommen.

Ausnahme

> Sollen nur die Timeout-Einstellungen von Geräten verändert werden, kann das Kommando MODIFY-IO-UNIT (mit UNIT=*DEVICE(...) und TIMEOUT=...) auch außerhalb einer dynamischen Rekonfiguration eingegeben werden. Dasselbe gilt für die Festlegung eines bevorzugten Gerätes für PAV (Operand PAV-PREFERRED-DEVICE).
> **Diese Änderungen sind unabhängig von der Architektur der Server Unit möglich!**

Ist die dynamische I/O-Konfigurationsänderung erfolgreich eingeleitet, kann die Systembetreuung im laufenden System mit den Kommandos ADD-IO-UNIT, REMOVE-IO-UNIT und MODIFY-IO-UNIT die Ein-/Ausgabe-Konfiguration dynamisch ändern. Die Änderungen werden im aktiven IORSF (Input/Output Resource File) vorgenommen, d.h. sie werden sofort wirksam.

Allgemeiner Ablauf einer dynamischen I/O-Konfigurationsänderung

Jede über Kommando angeforderte Konfigurationsänderung wird mehrstufig unter Steuerung der Systemkomponente IORECONFabgearbeitet:

1. Prüfphase:
   IORECONF prüft die Durchführbarkeit der angeforderten Konfigurationsänderung und weist nicht durchführbare Anforderungen zurück. So wird z.B. ein ADD-IO-UNIT-Kommando für ein Gerät aus einem der folgenden Gründe abgewiesen:
   - Die Steuerung oder der Kanal, an dem das Gerät angeschlossen werden soll, ist nicht definiert.

- Das Gerät ist bereits definiert.
- Es ist kein freier Tabelleneintrag vorhanden.

Danach werden alle weiteren Systemkomponenten, die von der Konfigurationsänderung betroffen sind, aufgerufen (z.B. die Komponenten der Geräte- und Datenträgerverwaltung). Die Anforderung kann dann durch eine der gerufenen Komponenten abgelehnt werden.

2. Modifikation des IORSF:
   Nach abgeschlossener Prüfung wird die Konfigurationsänderung in das aktive IORSF eingetragen. Für die Dauer der IORSF-Änderung sind alle betroffenen Geräte für Ein-/Ausgaben gesperrt.
3. Modifikation der Tabellen:
   Nach erfolgreicher IORSF-Änderung ändert IORECONF die Ein-/Ausgabe-Tabellen und informiert die betroffenen Systemkomponenten über die Konfigurationsänderung.

Kann die Anforderung einer Konfigurationsänderung nicht vollständig ausgeführt werden, versucht IORECONF den Ausgangszustand wiederherzustellen.
Sowohl Durchführung als auch Abweisung von Anforderungen zur Konfigurationsänderung werden durch Konsol-Meldungen (`NKR....`) protokolliert.

I/O-Konfiguration dynamisch erweitern

Mit dem Kommando ADD-IO-UNIT kann die Systembetreuung der Konfiguration neue Ein-/Ausgabe-Einheiten (Geräte, Steuerungen, Kanäle) hinzufügen. Geräte können als PAV-Alias-Geräte definiert werden (siehe Abschnitt „Parallel Access Volumes (PAV, SU/390)").

Neue Ein-/Ausgabe-Einheiten müssen in der Reihenfolge Kanal - Steuerung - Gerät angegeben werden. Bis zu 256 Geräte lassen sich mit einem ADD-IO-UNIT-Kommando an einer Steuerung hinzufügen.

Einschränkungen und Besonderheiten

Eine neue Kanalbaugruppe kann mit folgenden Schritten hinzugefügt werden:

1. Kommando ADD-IO-UNIT *CHN(...),STATE=DET für jeden Kanal der Baugruppe
2. Einbau der Kanalgruppe und Aktivierung über CM-Frame
3. Kommando ATTACH-DEVICE *CHN(...) für jeden Kanal der Baugruppe

Kanäle auf bereits bei IMPL vorhandenen Baugruppen können zwar mit ADD-IO-UNIT dynamisch generiert werden, sie lassen sich danach aber nicht „online“ schalten.
Deshalb sollten auch alle bereits vorhandenen Kanäle vorher generiert werden.

Für den Betrieb eines neuen Bandgerätes sind nach der Neudefinition mit ADD-IO-UNIT und bei Einsatz eines MBK-Archivsystems noch folgende Maßnahmen erforderlich:

1. Mit dem Kommando ADD-DEVICE-DEPOT muss eine Lagerortzuordnung getroffen werden.
2. Falls MAREN im Einsatz ist und das neue Gerät der Freibandzuweisung unterliegen soll, muss das Programm MARENUCP beendet und neu gestartet werden.
3. Ist das neue Gerät Teil eines von ROBAR gesteuerten MBK-Archivsystems und noch nicht in der ROBAR-Konfigurationsdatei definiert, muss ROBAR-SV beendet und die ROBAR-SV-Konfigurationsdatei erweitert werden. Danach kann ROBAR-SV neu gestartet werden.

Bei Konfiguationen ohne MBK-Archivsystems muss das Bandgerät nur noch ATTACHED werden (Kommando ATTACH-DEVICE), wenn es nicht bereits implizit mit dem Kommando ADD-IO-UNIT ...,STATE=*ATTACHED zugeschaltet wurde.

Ein neues Netzwerk-/LAN-Gerät muss vor oder nach dem Hinzufügen mit BCIN-Kommando im DCM bekannt gemacht werden.

Das Subsystem IOTRACE berücksichtigt neu hinzugefügte Ein-/Ausgabe-Einheiten nur, wenn es nach der Konfigurationsänderung neu gestartet wurde.

I/O-Konfiguration dynamisch verkleinern

Mit dem Kommando REMOVE-IO-UNIT kann die Systembetreuung Ein-/Ausgabe-Einheiten (Geräte, Steuerungen, Kanäle), die sich im Zustand DETACHED befinden, aus der aktuellen Konfiguration entfernen. Wird eine „innere“ Einheit entfernt (Kanal bzw. Steuerung), so werden vom System implizit die zugehörigen „äußeren“ Einheiten (Steuerung bzw. Geräte) entfernt, sofern diese keiner weiteren „inneren“ Einheit angehören. Zu entfernende Ein-/Ausgabe-Einheiten müssen in der Reihenfolge Gerät - Steuerung - Kanal angegeben werden.

I/O-Konfigurationseigenschaften dynamisch modifizieren

Mit dem Kommando MODIFY-IO-UNIT kann die Systembetreuung die Konfigurationseigenschaften von Steuerungen oder Geräten dynamisch verändern. Bei einer Steuerung können die Ein-/Ausgabe-Pfade und die Verfügbarkeit der Verbindung zwischen Kanal und Steuerung (INCLUDED oder REMOVED) neu festgelegt werden.

Somit kann eine Steuerung, die mindestens zwei Kanalanschlüsse besitzt, unterbrechungsfrei umgehängt werden. Bei einem Gerät kann der bevorzugte Ein-/Ausgabe-Pfad geändert werden. Mit dem Operanden STATE=*PAV-PREFERRED-DEVICE(...) kann festgelegt werden, dass das PAV-Gerät unter VM2000 bei Ein-/Ausgaben zu bevorzugen ist (zu PAV siehe Abschnitt „Parallel Access Volumes (PAV, SU /390)").

Außerdem kann mit dem Kommando die Timeout-Einstellung von Geräten verändert werden (MODIFY-IO-UNIT UNIT=*DEVICE(...),TIMEOUT=<wert_in_sekunden>).
Werden nur die Timeout-Einstellungen von Geräten verändert oder ein bevorzugtes Gerät für PAV festgelegt, kann das Kommando auch außerhalb einer dynamischen Rekonfiguration (d.h. auch unabhängig von der Architektur der Server Unit) eingegeben werden.
Die Veränderung der Timeout-Werte sollte nur kurzfristig für bestimmte Aktionen, z.B. den Online-Update einer Plattenspeichersystem-Steuerung, erfolgen. Danach sollten wieder die Standardwerte mit TIMEOUT=*DEFAULT eingestellt werden.

Dynamische I/O-Konfigurationsänderung beenden

Die dynamische I/O-Konfigurationsänderung wird mit dem Kommando STOP-CONFIGURATION-UPDATE abgeschlossen. Damit die aktuelle Ein-/Ausgabe-Konfiguration auch für eine Folge-Session verfügbar ist, kann das aktuelle IORSF über den Operanden IORSF-UPDATE gesichert werden.
Kommt es vor dem Zurückschreiben des aktiven IORSF zu einem Systemabsturz, gehen die bis dahin vorgenommenen Konfigurationsänderungen beim erneuten IPL nicht verloren, da die BS2000-Ein-/Ausgabe-Tabellen bei der Systemeinleitung aus dem aktiven IORSF abgeleitet werden.

Nach erfolgreichem Abschluss der dynamischen Rekonfiguration werden die Rekonfigurationskommandos ADD-IO-UNIT, REMOVE-IO-UNIT und MODIFY-IO-UNIT (Ausnahme Timeout-Einstellung) nicht mehr angenommen.

Dynamische I/O-Konfigurationsänderung unter VM2000

Unter VM2000 können die Kommandos zur dynamischen I/O-Rekonfiguration nur im Monitor-System eingegeben werden. Die erforderlichen Änderungen der Ein-/Ausgabe-Tabellen werden im Hypervisor, im Monitorsystem und automatisch in allen aktiven Gastsystemen durchgeführt. Bei Eingabe des START-CONFIGURATION-UPDATE-Kommandos wird in der Meldung `NKR0178` angezeigt, wie viele Gastsysteme aktiv sind und die dynamische I/O-Konfigurationsänderung unterstützen.

5.4 Dynamische I/O-Konfigurationsänderung (SU x86)

An SU x86 wird Busperipherie nicht mit IOGEN, sondern über X2000 konfiguriert. Damit entfällt auch die Möglichkeit der Überdefinition der Ein-/Ausgabe-Konfiguration, die es erlaubt, vordefinierte Geräte zu einem späteren Zeitpunkt dynamisch in Betrieb zu nehmen.

Platten, Band- und Netzwerk-/LAN-Geräte können im laufenden Betrieb der Konfiguration hinzugefügt werden. BS2000 erzeugt für diesen Zweck bei der Systemeinleitung zusätzliche freie Gerätetabelleneinträge:

- für Netzwerk-/LAN-Geräte 64
- für Platten- und Bandgeräte jeweils ca. 10% der Anzahl bereits in der Gerätetabelle definierten Geräte (mindestens 64, höchstens 2048)

5.5 Rekonfiguration von Extra- und Spare-CPUs

Die Server Units bieten gesteigerte Hochverfügbarkeit für CPUs sowie die Möglichkeit, über die nominelle Server-Leistung hinaus zeitlich begrenzt weitere, bereits vorinstallierte CPUs zuzuschalten.

Bei Ausfall einer BS2000-CPU wird automatisch eine vorhandene Ersatz-CPU zugeschaltet. Damit bleibt die nominelle Leistung der Server Unit garantiert.
Auf der SU/390 steht dafür eine Spare CPU zur Verfügung.
Auf der SU x86 wird beim Systemstart nach einem CPU-Ausfall eine bisher nicht von BS2000 genutzte CPU verwendet.

Bei Bedarf an zeitlich begrenzter zusätzlicher Leistung der Server Unit können über den nominellen Multiprozessorgrad (MP-Grad) der Server Unit hinaus neben der Spare-CPU weitere CPUs (Extra-CPUs) installiert sein, deren Nutzungsdauer vertraglich geregelt ist. Das Zu- und Wegschalten erfolgt durch die Systembetreuung mittels erweiterter CPU-Rekonfigurationskommandos. Die Nutzungsintervalle werden in BS2000 in einer eigenen Protokolldatei sowie als Meldungen im Remote Service aufgezeichnet.

Zur Nutzung von Spare-/Extra-CPUs in VM2000 siehe Handbuch „VM2000“ [60].

5.5.1 Funktionalität im native BS2000-Betrieb

Installation

Im Rahmen der Neuinstallation bzw. der Umkonfiguration einer Server Unit hinterlegt die vom Service eingelesene Modell-Lock-Diskette die Modell- und HW-Charakteristika in der Server Unit. Das sind: die Modell-ID, die global eindeutigen IDs der installierten CPUs, der nominelle MP-Grad der Server Unit sowie die Anzahl der verfügbaren Spare-CPUs (nur bei SU /390) sowie der vertraglich vereinbarten Extra-CPUs.

IPL

Beim Startup werden gemäß dem nominellen MP-Grad CPUs in Betrieb genommen (ATTACHED, ONLINE, NORMAL).

Extra-CPUs bleiben in BS2000 weggeschaltet (DETACHED, OFFLINE, EXTRA).

Spare-CPUs bleiben ebenfalls weggeschaltet (DETACHED, OFFLINE, SPARE), falls es sich um ein Modell mit nominell zwei oder mehr CPUs handelt. Bei Modellen mit einer nominellen CPU wird die Spare-CPU in den Zustand DETACHED, ONLINE, SPARE gebracht.

Die Festlegung einer CPU für den Normalbetrieb erfolgt jeweils bei IPL durch die Firmware und kontrolliert durch die BS2000-Software; der Zustand der CPUs wird mit dem Kommando SHOW-DEVICE-CONFIGURATION CLASS=*CPU angezeigt.

CPU-Ausfall

Bei Ausfall einer CPU (MFA=Malfunction Alert, MCKI=Machine Check Interrupt) auf einer SU /390 schaltet BS2000 die defekte CPU (DETACHED, OFFLINE, ERROR) weg und gleichzeitig eine betriebsbereite Spare-CPU zu.

Dabei werden folgenden Meldungen, die auch als Teleservice-Call an das zuständige Service-Center gemeldet werden, ausgegeben:

```
ETMRK48 CPU (&01) DETACHED BY SYSTEM
ETMRK20 CPU (&01) ATTACHED BY SYSTEM
```

Die weggeschaltete CPU ist nach einem MFA für folgende „IPL/Power on Reset“ bzw. IMPL gesperrt. BS2000 zeigt sie ab dem nächsten IPL bzw. IMPL nicht mehr an. Erst nach Reparatur bzw. Austausch durch den Service (der ganze Server muss hierzu offline sein) kann dieser die CPU per SVP-Frame wieder verfügbar erklären.

Beim CPU-Ausfall aus anderen Gründen (z.B. CPU Loop) findet das Wegschalten der CPU herkömmlich ohne Auswirkung für das nächste Hochfahren statt (DETACHED, ONLINE, NORMAL). Das ersatzweise Zuschalten einer Spare-CPU findet nicht statt.

Zu- und Wegschalten von Extra-CPUs

Gemäß der vertraglich vereinbarten Nutzungsdauer können Extra-CPUs für eine begrenzte Zeit zugeschaltet werden. Im BS2000-Betrieb erfolgt dies mit erweiterten Operanden der Kommandos ATTACH-DEVICE bzw. DETACH-DEVICE:

```
/ATTACH-DEVICE UNIT=*EXTRA-CPU(CPU-IDENTIFIER=<x-string 2..2>/*ALL/*ANY)
```

*ALL Alle weggeschalteten Extra-CPUs sind zuzuschalten.

*ANY Eine beliebige der weggeschalteten Extra CPUs ist zuzuschalten.

Zusätzliche Meldungen in der Bearbeitung des Kommandos (asynchron, keine Komman-do-Returncodes):

```
ETMRK58 EXTRA-CPU (&00) ATTACHED
ETMRK5A ALL EXTRA-CPU'S ALREADY ATTACHED
ETMRK5E OBJECT EXTRA-CPU NOT EXISTING ON THIS HARDWARE
```

Das Wegschalten der Extra-CPUs erfolgt analog:

```
/DETACH-DEVICE UNIT=*EXTRA-CPU(CPU-IDENTIFIER=<x-string 2..2>/*ALL/*ANY)
```

*ALL Alle zugeschalteten Extra-CPUs sind wegzuschalten.

*ANY Eine beliebige der zugeschalteten Extra CPUs ist wegzuschalten.

Zusätzliche Meldungen in der Bearbeitung des Kommandos (asynchron, keine Komman-do-Returncodes):

```
ETMRK59 EXTRA-CPU (&00) DETACHED
ETMRK5B ALL EXTRA-CPU'S ALREADY DETACHED
ETMRK5E OBJECT EXTRA-CPU NOT EXISTING ON THIS HARDWARE
```

Das Zu- und Wegschalten wird in der speziellen Protokolldatei SYS.RESLOG.<server-id> festgehalten (siehe „Diagnosehandbuch“ [14]).

Auch bei Ausfall einer Extra-CPU wird eine betriebsbereite Spare-CPU automatisch zugeschaltet (nur auf SU /390).

Die Nutzung von Extra-Leistung endet mit Ende der BS2000-Session bzw. bei deren abnormaler Beendigung. Mit Beginn einer neuen Session sind Extra-CPUs wieder explizit hinzuzunehmen.

Erweiterung der Ausgabe in Informationskommandos

Beispiel: Informationen über CPUs (SU /390 mit 2 Extra- und 1 Spare-CPU

```
/SHOW-DEVICE-CONFIGURATION UNIT=*SELECT(CLASS=*CENTRAL-PROC)

MNEM  UN-CLASS  UN-TYPE  CONF-STATE  POOL/SIDE

00    CPU       SU700-30  ATTACH /ON  NORMAL
01    CPU       SU700-30  ATTACH /ON  NORMAL
02    CPU       SU700-30  ATTACH /ON  NORMAL
03    CPU       SU700-30  ATTACH /ON  EXTRA
04    CPU       SU700-30  DET(EX)/OFF EXTRA
05    CPU       SU700-30  DET(EX)/OFF SPARE
```

5.5.2 Funktionalität im Gastsystem unter VM2000

BS2000-Gastsystemen sind die Extra-CPUs verborgen. Ihre Nutzung ist ausschließlich über VM2000-Kommandos möglich, siehe Handbuch „VM2000“ [60].

In VM2000 werden allen Gastsystemen neben den normalen virtuellen CPUs (deren Anzahl beim Einrichten der VM festgelegt wird) so viele virtuelle Spare-CPUs zugewiesen, wie die Server Unit reale Spare-CPUs konfiguriert hat. Die bei Ausfall einer realen CPU momentan darauf ablaufende und damit ebenfalls ausfallende virtuelle CPU wird vom betroffenen Gastsystem durch eine virtuelle Spare-CPU ersetzt.

Dies ist von besonderer Bedeutung für die Verfügbarkeit von Gastsystemen, denen nur eine CPU zugewiesen wurde, da solche Systeme ohne Spare-CPU beim Auftreten von CPU-Fehlern während ihres Ablaufs sofort beendet sind. Mit einer (virtuellen) Spare-CPU gibt es im Fehlerfall eine realistische Chance zum unterbrechungsfreien Weiterlauf des Gastsystems nach interner CPU-Rekonfiguration.

5.6 NDM - Betriebsmittelbelegung und -reservierung

Das NDM (Nucleus Device Management) ist die eigentliche Geräteverwaltung von BS2000. Grundfunktion der Geräteverwaltung ist die Gerätebelegung. Da i.A. nicht immer jede Geräteanforderung erfüllbar ist, muss NDM auch eine Gerätereservierungsfunktion anbieten sowie Warteschlangen verwalten und abarbeiten.

NDM erfasst Anzahl und Zustand der Geräte beim Systemstart sowie Zustandsänderungen während des laufenden Betriebs.

Stellt ein Benutzerprogramm einen Antrag auf ein bestimmtes Gerät, ist dieses verfügbar und wird ihm von NDM zugewiesen, so muss NDM auch dafür sorgen, dass das Gerät als belegt gekennzeichnet wird und kein anderes Benutzerprogramm Zugang zu dem Gerät hat. Dieses Gerät gilt als belegt, bis das zugriffsberechtigte Benutzerprogramm seine Arbeit mit dem Gerät beendet und dies NDM mitgeteilt hat.

Kommando	**Bedeutung**
ADD-DEVICE-DEPOT	Zuordnung Bandgerät zu Lagerort vereinbaren
CHANGE-DISK-MOUNT	Privatplatte für Zugriff sperren
CHANGE-TAPE-MOUNT	Montierzustand von Bändern ändern
CHECK-DISK-MOUNT	Montierzustand einer Platte prüfen
CHECK-TAPE-MOUNT	Montierzustand von Bandgeräten und MBK prüfen
MODIFY-MOUNT-PARAMETERS	Vorgaben für das Montieren und Demontieren von Bändern und Platten ändern
MODIFY-RESOURCE-COLLECTION	Collector-Auswahl steuern
REMOVE-DEVICE-DEPOT	Zuordnung Bandgerät zu Lagerort aufheben
SECURE-RESOURCE-ALLOCATION	Fordert Betriebsmittel an
SET-DISK-DEFAULTS	Standardwerte für DISK-Parameter festlegen
SET-DISK-PARAMETER	Vorgaben für die Überwachung von Platten setzen
SHOW-DEVICE-CONFIGURATION	Abfragen der Ein-/Ausgabe-Konfiguration
SHOW-DEVICE-DEPOT	Zuordnung Bandgerät zu Lagerort abfragen
SHOW-DEVICE-STATUS	Belegungs- und Überwachungsinformationen für Geräte anfordern
SHOW-DISK-STATUS	Belegung und DISK-Parameter abfragen
SHOW-MOUNT-PARAMETER	Monitor-Vorgaben von Platten und Bändern abfragen
SHOW-RESOURCE-ALLOCATION	Task-Belegung und offene Operator-Aktionen abfragen
SHOW-RESOURCE-REQUESTS	Zustand von Gerätewarteschlange und Collector-Task abfragen

SHOW-TAPE-STATUS	Bandbelegung und -überwachung abfragen
START-RESOURCE-COLLECTION	Collector-Auswahl starten
STOP-RESOURCE-COLLECTION	Collector-Auswahl beenden
UNLOCK-DISK	Systembelegungs-Protokoll bereinigen
Makro	**Bedeutung**
NKDINF	Information über den Belegungs- und Verfügbarkeitszustand der (peripheren) Konfiguration
NKGTYPE	Information über Namen, Gerätetypcode, Geräteeigenschaften usw. eines Geräte- oder Volumtyps oder über Namen und Gerätetypcodes der Gerätetypen, die zu einer Gerätefamilie oder -klasse gehören

Tabelle 12: Schnittstellenübersicht zur Betriebsmittelbelegung und -reservierung

Die NDM-Informationsdienste liefern dem Operator bestimmte, dem angegebenen Kommando und dem gewünschten Umfang der Information entsprechende Ausgabefelder.

Eine Übersicht über die SHOW-Kommandos und die Erklärung der ausgegebenen Informationen finden Sie im Anhang des Handbuchs „Kommandos“ [27].

5.6.1 Task-Belegung von Datenträgern

Die Geräteverwaltung steuert und überwacht die Belegung der vom Benutzer angeforderten Betriebsmittel (Geräte, Datenträger, Dateien), abhängig von der Benutzungsart der Datenträger.

Für Bänder und private Platten gibt es die Benutzungsarten:

- „DMS“
  für alle DVS-Zugriffe (über Kommandos SECURE-RESOURCE-ALLOCATION, CREATE-FILE, ADD-FILE-LINK, COPY-FILE...)
- „SPECIAL“
  für Sonderanwendungen (z.B. Dienstprogramme INIT, VOLIN, FDDRL ...)

Für Bänder gibt es darüber hinaus noch die Benutzungsart:

- „WORK“
  für DVS-Arbeitsbänder, siehe Kommandos SECURE-RESOURCE-ALLOCATION, ADD-FILE-LINK im Handbuch „Kommandos“ [27]

Task-exklusive Belegung

Ein Datenträger kann nicht gleichzeitig von mehreren Tasks belegt werden.
Alle Belegungsanforderungen anderer Tasks für denselben Datenträger innerhalb desselben Systems sowie auch von anderen Systemen werden abgewiesen.

Ausnahme: Exklusive Belegungsanforderungen (SECURE-RESOURCE-Kommando) von Batch-Tasks warten, bis der Datenträger freigegeben wird.

Task-shareable Belegung

Ein Datenträger kann gleichzeitig von mehreren Tasks belegt werden. Die einzigen Datenträger, die task-shareable belegt werden können, sind Platten im USE-MODE=DMS (gemeinschaftliche und Privatplatten).

Belegungsarten für private Datenträger und sonstige Betriebsmittel:

- Belegung von Bändern

  Bänder werden immer task-exklusiv belegt, eine task-shareable Belegung ist nicht möglich. Das System unterscheidet, ob das Band nur belegt oder tatsächlich genutzt werden soll. Wird es nur belegt, d.h. andere Tasks können nicht auf das Band zugreifen, es finden jedoch keine I/O-Operationen statt, so wird der Operator mit einer Premount-Meldung aufgefordert, eine Gerätezuordnung zu treffen, falls das Band nicht als montiert erkannt wird. Soll das Band dagegen für I/O-Operationen genutzt werden, wird der Operator mit einer Mount-Aufforderung auf diese Absicht hingewiesen.

- Belegung von Privatplatten

  Privatplatten werden für DVS-Betrieb (Benutzungsart DMS) standardmäßig task-shareable belegt. Task-exklusive Belegung ist nur über eine entsprechende Vorab-Reservierung (SECURE) möglich.

  Platten für Benutzungsart SPECIAL werden task-exklusiv belegt.

  Der Operator kann mit dem DISK-Parameter USER-ALLOCATION festlegen, welche Task-Belegung für eine Platte (für Benutzungsart DMS) zugelassen ist.

  Mit USER=*EXCL kann eine Platte nur task-exklusiv (mit SECURE-RESOURCE-ALLOCATION DISK= ..., ALLOC=*EXCL) belegt werden. Die auf diese Weise belegte Platte ist dann nur für diese Task verfügbar. Jeder Versuch, diese Platte task-shareable zu belegen, wird abgewiesen.

  Mit USER=*SHARE kann die Platte für alle DVS- und SECURE-Anforderungen ausschließlich task-shareable belegt werden. Jede task-exklusive Belegungsanforderung der Platte wird abgewiesen.

  Mit USER=*ALL ist die Platte für alle task-shareablen und -exklusiven DVS- und Secure-Anforderungen belegbar. Dies gilt jedoch nur für die Erstbelegung der Platte. Wird die Platte task-exklusiv belegt, werden alle weiteren Belegungsanforderungen von anderen Tasks für diese Platte abgewiesen.

  Mit USER=*NO kann die Platte von keinem Benutzer belegt werden. Diese Einstellung erfolgt vom System automatisch bei: `/CHANGE-DISK-MOUNT UNIT=mn/*VOLUME(vsn), ACTION=*CANCEL` oder bei Beantwortung einer Remount-Meldung mit `tsn.N`.

- Belegung von sonstigen Betriebsmitteln

  Alle Belegungsanforderungen für Geräte (Band-, Plattengeräte, etc) werden ohne Eingriff durch den Operator task-exklusiv durchgeführt.

5.6.2 Systembelegung von Platten

Jede Belegung einer Platte wird in ihrem Standard-Volume-Label (SVL) hinterlegt. Dazu beinhaltet das SVL der Platte ein Systembelegungs-Protokoll, in das bei gemeinschaftlichen Platten bis zu 16 System-Katalogkennungen, die die Platte belegen, eingetragen werden können. Für Platten gibt es folgende Belegungen:

System-exklusive Belegung

Eine Platte kann nur von einem System belegt werden. Die System-Id des Home-Pubsets dieses Systems wird im Systembelegungs-Protokoll des SVL so eingetragen, dass dies eine gleichzeitige Belegung der Platte durch andere Systeme verhindert.

System-shareable Belegung

Eine Platte eines Shared-Pubsets kann von maximal 16 Systemen gleichzeitig belegt werden. Die System-Id jedes Systems, das die Platte belegt, wird so im Systembelegungs-Protokoll eingetragen, dass bis zu 15 weitere Systeme gleichzeitig die Platte belegen können.

> ¡ Bei Einsatz des Softwareprodukts MSCF sind die Hinweise in Abschnitt „Shared-Pubsets“ zu beachten.

5.6.3 Vorgaben für die Privatplattenbelegung

ASSIGN-TIME

Mit dem DISK-Parameter ASSIGN-TIME kann der Operator Vorgaben einstellen für den Zeitpunkt der Belegung bzw. Freigabe einer Privatplatte sowie eines Plattengerätes mit entsprechender Aktualisierung des Systembelegungs-Protokolls auf der Platte.

- Mit ASS=*USER beginnt die Belegung der Platte und des Gerätes, auf dem sie montiert ist, mit der Belegung der Platte durch den Benutzer. Gleichzeitig wird die System-Id des Systems ins Systembelegungs-Protokoll eingetragen. Die Freigabe der Platte und des Gerätes erfolgt mit der Freigabe der Platte durch den Benutzer, das Systembelegungs-Protokoll wird bereinigt.
- Mit ASS=*OPERATOR erfolgt die Belegung der Platte sowie des Gerätes, auf dem sie montiert ist, unabhängig von der Belegung durch den Benutzer. Ist die Platte bereits online, so wird sie und das Gerät sofort belegt, sonst nach erfolgtem Einschalt-Interrupt, spätestens jedoch bei einer Belegungsanforderung durch einen Benutzer. Die Platte und das Gerät bleiben solange belegt, bis der Operator mit der Eingabe ASS=*USER die Belegung zurückgibt und alle Benutzer die Platte freigegeben haben.

  Platten, auf die zumindest über einen längeren Zeitraum von einem System zugegriffen werden soll, sollten über ASS=*OPERATOR belegt werden. Damit werden unnötige Belegungs- und Freigabeprotokolle vermieden.
  Bei der Belegung mit ASS=*OPERATOR ist zu beachten, dass vor dem Wegschalten mit dem Kommando DETACH-DEVICE ...,FORCE=*NO die Einstellung ASS=*USER vorzunehmen ist, da das DETACH-DEVICE-Kommando sonst abgewiesen wird. Nur so ist eine korrekte Bereinigung des Belegungsprotokolls sichergestellt.

Belegungsbeginn

- bei ASS=*USER mit der ersten Benutzeranforderung
- bei ASS=*OPERATOR
  - mit dem Einschalt-Interrupt bei Montierung (ATTACH-DEVICE)
  - sofort, wenn Platte bereits online
  - im Rahmen der ersten Belegungsanforderung durch den Benutzer

Belegungsfreigabe

- bei ASS=*USER mit der letzten Benutzerfreigabe
- bei ASS=*OPERATOR
  - bei bzw. nach SET-DISK-PARAMETER ASSIGN-TIME=*USER
  - bei bzw. nach SET-DISK-PARAMETER ASSIGN-TIME=*STD (und Standardwert ASSIGN-TIME=*USER eingestellt)
  - wenn keine Task die Platte mehr belegt

OPERATOR-CONTROL

Mit diesem DISK-Parameter legt der Operator fest, ob er Erst-Belegungen von Platten durch Tasks kontrollieren will. Schaltet der Operator die Kontrolle für eine Platte ein, so wird bei jeder ersten Belegung der Platte durch eine Task folgende Meldung ausgegeben:

```
NKA0004 ALLOCATION OF DISK '(&00)' IN USAGE MODE '(&01)' BY USER TASK
        PERMITTED? REPLY (Y=ALLOCATION ACCEPTED; N=ALLOCATION REJECTED)
```

Der Operator wird dadurch aufgefordert, zu entscheiden, ob die angegebene Platte der anfordernden Task in der Belegungsart zugewiesen werden soll. Nur wenn diese Meldung positiv (`TSN.Y`) beantwortet wird, wird die Platte von der Task belegt, sonst wird dieser und alle folgenden Belegungsversuche dieses Jobs ohne weitere Operatoraktion abgewiesen.

Ist eine Platte, für die der DISK-Parameter OPERATOR-CONTROL gesetzt wird, zu diesem Zeitpunkt bereits von einem Job belegt, so wird für diese Platte erst dann die Meldung `NKA0004` ausgegeben, wenn sie vom belegenden Job freigegeben wird und wieder neu belegt werden soll.

5.6.4 Steuerung der Betriebsmittelreservierung

Wird von einer Task ein Reservierungsauftrag gegeben (mit dem Kommando SECURE-RESOURCE-ALLOCATION), versucht das System, alle angeforderten Betriebsmittel (Geräte, Datenträger oder Dateien) für diese Task zu reservieren.
Sind die angeforderten Betriebsmittel verfügbar und können für die Task reserviert werden, so ist der Reservierungsauftrag abgeschlossen und die Bearbeitung weiterer Aufträge kann erfolgen.

Kann der Reservierungsauftrag nicht oder nur teilweise ausgeführt werden, wird der Job in eine Warteschlange, die Secure-Queue, eingetragen. Wartende Aufträge halten keine Betriebsmittel (Ausnahme ist ggf. die Collector-Task).

Die Collector-Task

Der Operator hat die Möglichkeit, jeweils einen Job in der Secure-Queue privilegiert zu behandeln. Diese Privilegierung besteht darin, dass diese Task am Anfang der Queue steht und die angeforderten Betriebsmittel sammeln kann. Diese Task wird als Collector-Task bezeichnet. Zur Steuerung der Collector-Auswahl stehen dem Operator die Kommandos START-RESOURCE-COLLECTION, MODIFY-RESOURCE-COLLECTION und STOP-RESOURCE-COLLECTION zur Verfügung.

Mit diesen Kommandos kann der Operator folgende Funktionen durchführen:

- Starten der COLLECTOR-Auswahl

  Mit dem Kommando START-RESOURCE-COLLECTION schaltet der Operator die Auswahl einer Collector-Task ein. Das System berechnet für jede Task in der Secure-Queue ein Gewicht nach der Formel:

  `W = T + N * U`

  `W` Weight; Gewicht

  `T` Time; Wartezeit, welche die Task bereits in der Secure-Queue verweilt

  `U` Urgency; dieser Wert berechnet sich aus der Task-Priorität, wobei die Task mit der höchsten Priorität die niedrigste Urgency hat und umgekehrt

  `N` Faktor, der vom Operator im Operanden TIME-WEIGHT des Kommandos START-RESOURCE-COLLECTION eingegeben wird (Standardwert ist 10).
  Durch gezielte Wahl des Faktors N kann der Operator die Berechnung der Gewichte beeinflussen:
  Wird N klein gewählt, bekommt die Wartezeit, die eine Task bereits in der Secure-Queue verweilt, große Bedeutung.
  Für N=0 ist die Wartezeit gleich dem Gewicht, die „Urgency“ hat keinen Einfluss auf die Berechnung.
  Wird N jedoch groß gewählt, steigt die Bedeutung der Task-Priorität für die Berechnung des Gewichtes.
  Für N=600 berechnet sich das Gewicht fast nur aus der „Urgency“; die Wartezeit fällt bei der Berechnung kaum ins Gewicht.

  Nachdem das System für alle Tasks in der Secure-Queue diese Berechnung durchgeführt hat, wird diejenige Task Collector-Task, die das höchste Gewicht hat. Die Collector-Task wird an den Anfang der Secure-Queue gestellt und kann ihre angeforderten Betriebsmittel sammeln. Wenn die Collector-Task alle von ihr angeforderten Betriebsmittel sammeln konnte, wird sie aus der Secure-Queue herausgenommen und das System berechnet für die in der Queue verbliebenen Tasks erneut die Gewichte, worauf eine neue Task zur Collector-Task wird.

- Tasks von der Auswahl ausschließen

  Der Operator kann mit dem Kommando MODIFY-RESOURCE-COLLECTION ACTION=*REMOVE-COLLECTOR eine Task von der Auswahl ausschließen. Wenn das System eine neue Collector-Task bestimmt (siehe Punkt 1), dann wird diese Task nicht berücksichtigt, d.h. sie kann nicht zur Collector-Task werden. Ist die angegebene Task zum Zeitpunkt des Ausschlusses Collector-Task, so verliert sie ihre Collector-Eigenschaft und damit alle bis dahin reservierten Betriebsmittel.

- Tasks zur Auswahl zulassen

  Der Operator kann eine vorher ausgeschlossene Task (siehe Punkt 2) wieder zur Auswahl zulassen (mit dem Operanden ACTION=*ADD-COLLECTOR); diese Task wird bei der nächsten Collector-Auswahl durch das System wieder berücksichtigt.

- Eine Task zur Collector-Task erklären

  Der Operator kann mit dem Kommando MODIFY-RESOURCE-COLLECTION ACTION=*SET-COLLECTOR eine Task zur Collector-Task machen. Ist eine andere Task Collector-Task, so verliert diese die Collector-Eigenschaft und damit die bereits gesammelten Betriebsmittel; die im Kommando angegebene Task wird zur Collector-Task. Diese Funktion ist auch ohne vorhergehendes START-RESOURCE-COLLECTION-Kommando möglich.

- Beenden der Collector-Auswahl

  Mit dem Kommando STOP-RESOURCE-COLLECTION beendet der Operator die mit dem Kommando START-RESOURCE-COLLECTION (siehe Punkt 1) gestartete Collector-Auswahl wieder. Ist eine Task Collector-Task, so verliert diese ihre Collector-Eigenschaft nicht, sie kann weiter die von ihr angeforderten Betriebsmittel sammeln. Entzieht der Operator dieser Task die Collector-Eigenschaft (siehe Punkt 2) oder kann die Collector-Task alle angeforderten Betriebsmittel sammeln, so wird vom System keine neue Collector-Task ausgewählt.

5.6.5 Hilfestellung im Umgang mit NDM

NDM bietet dem Operator ein breites Spektrum an Steuerungsmöglichkeiten für Datenträger. Da diese Parametereinstellungen unabhängig vom Online-Zustand (also auch vom Demontieren des entsprechenden Datenträgers) gelten, kann es beim Nichtbeachten der momentanen Einstellung der Parameter zu einem unerwarteten Systemverhalten kommen (z.B. kann eine Platte unter Umständen nicht belegt werden, obwohl sie schon montiert ist; soll die Belegung der Platte wieder zugelassen werden, muss der Operator den entsprechenden Parameterwert ändern). Alle Werte sind über Informationsfunktionen abfragbar.

Weitere unerwartete Reaktionen können eintreten, wenn der Zustand der Hardware nicht mit dem der Software übereinstimmt. (Wird z.B. ein Band auf ein Gerät im Zustand ATTACHED montiert, die Bandsteuerung ist jedoch nicht an die Server Unit geschaltet, kann das Band nicht belegt werden.)

Im Folgenden werden die wichtigsten Punkte des Systemverhaltens beschrieben und es wird gezeigt, wie sich die entsprechende Situation in der SHOW-Ausgabe darstellt. Außerdem werden Hinweise gegeben, wie solche Situationen bewältigt werden können.

Folgende Situationen werden beschrieben:

- Mount-Meldung trotz montiertem Datenträger
- SVL-Belegung durch andere Systeme
- Permanenter Hardware-Fehler für belegte Geräte
- Privatplatte permanent von einem anderen System gesperrt
- Prüfung auf freie Platten
- SECURE-Deadlock-Situationen
- Änderung der USER-ALLOCATION
- Information über Reservierungen
- Vorschläge zur performance-günstigen Einstellung von NDM-Parametern

Mount-Meldung trotz montiertem Datenträger

Es kann vorkommen, dass eine Mount-Anforderung an der Konsole ausgegeben wird, obwohl der Operator den angeforderten Datenträger bereits montiert hat. Dies kann folgende Ursachen haben:

1. Mehrere Datenträger gleicher VSN sind online (SHOW-DISK-STATUS, SHOW-TAPE-STATUS). Der Operator muss entscheiden, welcher Datenträger belegt werden soll.
2. Das Gerät, auf dem der Datenträger montiert wurde, ist von einer anderen Task belegt (SHOW-DEVICE-STATUS). Der Datenträger muss umgehängt werden.
3. Konfigurationszustand des Gerätes (es ist explizit oder implizit detached) verhindert die Belegung:
   - Steuerung oder Gerät ist detached (SHOW-DEVICE-STATUS),
   - Pfad ist (teilweise) removed (SHOW-DEVICE-CONFIGURATION).

   Datenträger umhängen oder Gerät/Pfad zuschalten (ATT/INC).
4. Der Hardwarezustand des Gerätes verhindert die Belegung:
   - Gerät ist noch nicht eingeschaltet (Platte),
   - Band ist noch nicht auf Bandanfangsmarke,
   - Steuerung ist nicht zugeschaltet.

5. Einschalt-Interrupt für montierten Datenträger wurde von der Hardware nicht geliefert. Lesen der VSN kann durch CHECK-DISK-/CHECK-TAPE-MOUNT erzwungen werden.
6. Der montierte Datenträger besitzt eine andere VSN als die vom Benutzer spezifizierte (SHOW-DEVICE-STATUS, SHOW-DISK-STATUS, SHOW-TAPE-STATUS).
7. Weitere Ursachen bei Plattenanforderungen:
   - Der vom Benutzer angegebene Gerätetyp stimmt nicht mit dem Typ der montierten Privatplatte überein (SHOW-DISK-STATUS VOL=vsn,INF=*PAR). Mount-Meldung abweisen bzw. ggf. zweite Platte gleicher VSN mit anderem Gerätetyp zuweisen.
   - VOLIN: UNIT-Vorgabe verlangt ein anderes Gerät als das, worauf die Platte montiert wurde. Für eine Fortsetzung muss der Datenträger umgehängt werden oder die Mount-Meldung abgewiesen und der Benutzer zur Freigabe des Gerätes veranlasst werden.
   - Die anfordernde Task hat sich das Gerät, auf dem die Platte montiert wurde, mittels /SECURE-RESOURCE-ALLOCATION UNIT=mn belegt und will die Platte für DVS-Betrieb belegen (USE=*DMS). Da Platten mit UNIT-Anforderungen nur für Benutzungsart SPECIAL reserviert werden, muss die Mount-Meldung abgewiesen und der Benutzer zur Freigabe des Gerätes veranlasst werden.
   - Eine Platte kann auf Grund ihrer SVL-Belegung durch andere Systeme nicht belegt werden: Beschreibung der möglichen Reaktionen siehe Abschnitt „SVL-Belegungdurch andere Systeme").
   -
8. Weitere Ursachen bei Bandanforderungen:
   - MODIFY-MOUNT-PAR ALLOC=*NO ist eingestellt (SHOW-MOUNT-PAR):
     Die Mount-Meldung muss beantwortet werden.
   - Der vom Benutzer angegebene Volumetyp wird von dem Gerät, auf dem der Datenträger montiert ist, nicht unterstützt (SHOW-RESOURCE-ALLOCATION, SHOW-DEVICE-CONFIGURATION). Band muss umgehängt werden.
   - Der Benutzer hat ein anderes Gerät explizit reserviert (SECURE-RESOURCE-ALLOCATION UNIT=mn) als das, worauf das Band montiert wurde.
     Die SECURE-UNIT-Anforderung des Benutzers ist zwingend, d.h. das Band muss auf das angegebene Gerät montiert werden.

SVL-Belegung durch andere Systeme

Obwohl eine Platte bereits als online protokolliert wurde, kann eine Belegungsanforderung durch einen Benutzer zu einer Mount-Meldung für dieselbe Platte führen. Dies ist dann der Fall, wenn eine Inkonsistenz zwischen der gültigen SYSTEM-ALLOCATION der Platte und ihrer tatsächlichen SVL-Belegung durch andere Systeme erkannt wird. Durch die Mount-Meldung, der ein Hinweis auf die vorliegende Diskrepanz vorausgeht, soll der Operator zu einer Entscheidung bzw. Reaktion aufgefordert werden.

Im Folgenden werden die möglichen Reaktionen und Eingriffe des Operators erläutert. Ausgangspunkt für die Überlegungen ist die Ausgabe von SHOW-DISK-STATUS VOL=vsn,INF=*ALL

1. Alle im SVL hinterlegten Systeme (SYSTEMS) sind nicht mehr aktiv:
   Die eingetragenen Systeme können mit UNLOCK-DISK VOL=vsn,SYS-ID=(...) ausgetragen werden. Die Mount-Meldung wird danach automatisch beantwortet, die Belegung wird dadurch akzeptiert.
2. Eingetragene Systeme arbeiten (teilweise) noch mit der Platte und die Platte soll vom eigenen System system-exklusiv belegt werden:
   Die Mount-Meldung muss abgewiesen werden.

3. Eingetragene Systeme arbeiten (teilweise) noch mit der Platte und die Platte soll vom eigenen System system-shareable belegt werden:

   - SVL-ALLOC=EXCL:
     Die Mount-Meldung muss abgewiesen werden, da die Platte nicht gleichzeitig von einem weiteren System belegt werden kann.
   - SVL-ALLOC=SHARE und alle Systeme aktiv:
     Im SVL sind 16 Fremdsysteme (bei gemeinschaftlichen Platten) hinterlegt:
     Verfahrensweise in diesem Fall wie in Punkt 3a)
   - SVL-ALLOC=SHARE und ein Teil der eingetragenen Systeme ist nicht mehr aktiv:
     Die inaktiven Systeme können – wie in Punkt 1 beschrieben – ausgetragen werden.Weitere Verfahrensweise wie in Punkt 3b).

Permanenter Hardware-Fehler für belegte Geräte

Für den Fall, dass ein Band- oder Plattengerät, das durch einen montierten und von einem oder mehreren Benutzern verwendeten Datenträger belegt ist, laufend Hardware-Fehler (INOP,...) meldet, bieten sich folgende Reaktionsmöglichkeiten an:

1. Datenträger ist eine Festplatte:

   Kann der Hardware-Fehler nicht behoben werden, muss für eine Privatplatte die Remount-Meldung abgewiesen werden (impliziter Datenträger-Cancel); handelt es sich um eine gemeinschaftliche Platte, kommt der BS2000-Betrieb mit diesem Pubset zum Erliegen.

2. Datenträger ist ein Band:

   Der betroffene Datenträger soll auf einem anderen Gerät montiert werden (Kommando CHANGE-TAPE-MOUNT VOLUME=vsn,ACTION=*MOVE). Falls noch ein freies Gerät verfügbar ist, wird dem Operator mit der Remount-Meldung ein Ersatzgerät vorgeschlagen. Wird das Kommando abgewiesen, kann dies folgende Ursachen haben:

   - Der Belegung des Datenträgers ging ein SECURE-RESOURCE-ALLOCATION UNIT=mn voraus, dadurch bleibt der Datenträger während seiner Benutzung durch die belegende Task zwingend dem Gerät zugeordnet. Kann der Hardware-Fehler nicht behoben werden, muss die Remount-Meldung mit `TSN.N` beantwortet werden. Der Datenträger wird gecancelt.
   - Es ist kein Ersatzgerät mehr verfügbar.

   In jedem Fall muss das Gerät mit DETACH-DEVICE UNIT=mn,FORCE=*YES weggeschaltet werden.

   Das Wegschalten des Gerätes kann folgende Wirkungen erzielen:

   - Der Datenträger wird implizit gecancelt, sofern er ein Nicht-Standard-Label besitzt oder der Belegung ein SECURE UNIT=*mn vorausging.
   - Der Datenträger wird in den ACTION-Status „NO DEVICE“ überführt, d.h. sobald wieder ein freies Gerät existiert, wird dies dem privaten Datenträger zugeordnet.

   Beim Wegschalten von Geräten mit wechselbaren Datenträgern wird in allen Phasen der Nutzung ein Warten auf das Freiwerden eines geeigneten Ersatzgerätes unterstützt, also sowohl im Zustand IN-USE als auch während der Bereitstellung (Zustand MOUNT oder PREMOUNT) sowie für reine Gerätetypbelegungen, z.B. nach SECURE-RESOURCE-ALLOCATION DEVICE=(TYPE=TAPE-C4, NUMBER=1).

   Dabei kann der ACTION-Status „NO DEVICE“ für Datenträger in Benutzung oder während der Bereitstellung unmittelbar bei DETACH UNIT=mn,FORCE=*YES eintreten, wenn kein geeignetes Ersatzgerät verfügbar ist. Es kann ein „freies“ Gerät in dem Sinn verfügbar sein, dass diesem zwar momentan kein Datenträger zugeordnet ist, für das aber eine sog. Typbelegung existiert. Diese Belegung wird verdrängt.

Die betroffene Task wartet – während der Phase der Bereitstellung des mit dieser Typbelegung verknüpften Datenträgers – auf das Freiwerden eines geeigneten Ersatzgerätes.
Datenträger im Zustand IN-USE verdrängen Datenträger im Zustand PREMOUNT. Das Verdrängen eines Datenträgers im Zustand MOUNT wird nur bei manuell bedienten Geräten automatisch durchgeführt. Bei von ROBAR unterstützten Geräten wird ein einmal eingeleiteter Montiervorgang nicht mehr abgebrochen, d.h. die Remount-Recover-Routine wird mit NO DEVICE terminiert und der Mount-Vorgang wird abgeschlossen (Eigenschaften von ROBAR).

Bei Freiwerden eines Gerätes durch ATTACH-DEVICE oder durch Belegungsende eines anderen Datenträgers erfolgt die Auswahl des zuzuordnenden Datenträgers in Abhängigkeit von der jeweiligen Phase (und zwar in folgender Reihenfolge: IN-USE vor MOUNT vor PREMOUNT), wenn mehr als ein Datenträger im Zustand NO DEVICE ist, der auf diesem Gerät vom Typ und der Lagerortzuordnung her verarbeitet werden könnte.

Privatplatte permanent von einem anderen System gesperrt

Eine Platte kann nicht belegt werden, falls ein anderes System als VTOC-Lockhalter im SVL der Platte hinterlegt ist (siehe Feld „VTOC-SYS“ bei SHOW-DISK INF=*SYS).
Der Operator wird in folgenden Fällen über diesen Zustand informiert:

- Belegungsaufforderung durch den Benutzer führt zu Meldung `NDV0002`
  (Platte permanent durch ein anderes System gesperrt).
- Impliziter UNLOCK für die eigene System-Id bei Einhängen der Platte bzw. expliziter UNLOCK (UNLOCK-DISK VOL=vsn,SYS-ID=sys-id) führt ebenfalls zu Meldung `NDV0002`.
- SET-DISK-PARAMETER VOL=vsn,ASS=*OPER wird mit einem Hinweis auf den VTOC-Lockhalter abgewiesen.

Es muss sichergestellt werden, dass das als VTOC-Lockhalter eingetragene System nicht mehr mit der Platte arbeitet und der Lock auf Grund eines Fehlverhaltens (z.B. Systemabsturz) oder durch das Canceln der Platte zurückgeblieben ist. Nur für diesen Fall darf der Lockhalter ausgetragen werden; ansonsten muss auf die Freigabe des Locks durch das belegende System gewartet werden, da es sonst zu Nebeneffekten kommen kann.

Für die Punkte a) und b) kann ein Austragen des Lockhalters durch Beantwortung der Meldung `NDV0002` mit `tsn.F` erreicht werden.

Für Punkt c) muss das Austragen des belegenden Systems explizit mit UNLOCK-DISK VOL=vsn,SYS-ID=sys-id eingeleitet werden. Diese Anforderung führt ebenfalls wieder zur Meldung `NDV0002`, die wie zuvor beschrieben mit `tsn.F` beantwortet werden muss.

Prüfung auf freie Platten

Ob eine Platte frei ist, wird z.B. dann geprüft werden, wenn sie neu initialisiert werden soll. Eine Platte wird nicht von Benutzern belegt, wenn das Kommando SHOW-DISK folgende Informationen liefert:

- PHASE=ONLINE
- PHASE=IN-USE und VOL-A=FREE bei Platten mit der Benutzungsart USE=DMS;
  In diesem Fall ist die Platte nur noch durch ihre ASSIGN-TIME belegt und kann durch SET-DISK-PARAMETER VOL=vsn,ASS=*USER[,USER=*NO] freigegeben werden.

SECURE-Deadlock-Situationen

Mit den Meldungen `NKS0022` und `NKS0054` wird der Operator auf Deadlock-Situationen hingewiesen, die bei der SECURE-Verarbeitung der aufgeführten Tasks festgestellt wurden.
Diese zwei Meldungen weisen auf folgende Deadlock-Situationen hin:

1. NKS0022 – Deadlock wegen Betriebsmittel, die die Tasks über die SECURE-Bearbeitung hinaus behalten dürfen:

   Eine Behebung der Deadlock-Situation ist nur über das Abbrechen einer oder mehrerer Tasks möglich.

   Beispiel

TASK 1	TASK 2
/CREATE-FILE FILE-NAME=DAT1,- SUPPORT=PRIVATE-DISK(VOLUME=- VOL1,DEVICE-TYPE=DEV1) /ASSIGN-SYSLST TO-FILE=DAT1 : /SEC-RES DISK=(VOL=VOL2,- TYPE=DEV2,- ALLOC=EX),- WAIT=.....	/CREATE=FILE FILE-NAME=DAT2,- SUPPORT=PRIVATE-DISK(VOLUME=- VOL2,DEVICE-TYPE=DEV2) /ASSIGN-SYSLST TO-FILE=DAT2 : /SEC-RES DISK=(VOL=VOL1,- TYPE=DEV1,- ALLOC=EX),- WAIT=.....

   Sowohl Task 1 als auch Task 2 belegen für sich durch Öffnen einer Privatplattendatei (ASSIGN-SYSLST) jeweils eine Privatplatte task-shareable. Die SECURE-Anforderung der beiden Tasks zur task-exklusiven Belegung der von der jeweils anderen Task bereits task-shareable belegten Platte führt dazu, dass beide Tasks auf eine Plattenfreigabe warten.

   Lösung:
   Diese klassische Deadlock-Situation kann nur durch das Beenden einer der beiden Tasks aufgehoben werden.

2. NKS0054 – Deadlock wegen Collecting

   Unter der Voraussetzung, dass während der laufenden Session eine Auswahl einer Collector-Task stattfindet, kann folgende Deadlock-Situation eintreten:

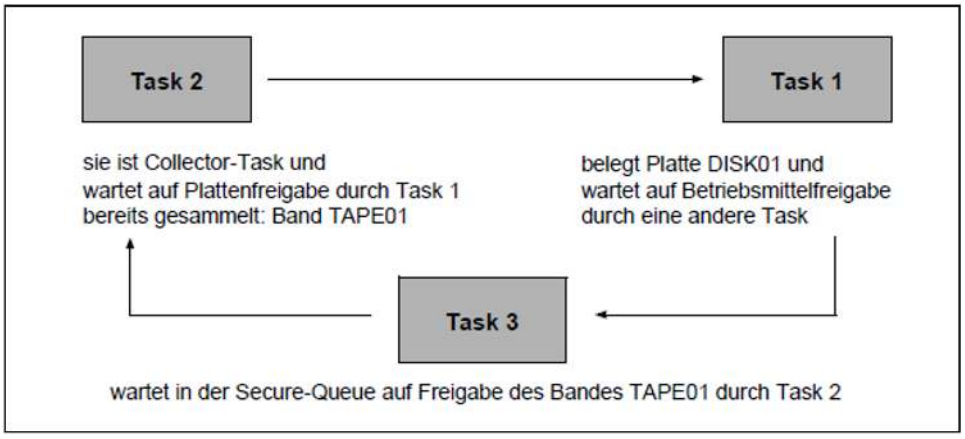


Bild 4: Deadlock-Situation wegen Collecting

Task 1 belegt die Platte DISK01 mit /CREATE-FILE FILE-NAME=abc und /ASSIGN-SYSLST TO-FILE=abc und wartet auf die Betriebsmittelfreigabe durch eine andere Task mit /SE-CURE-RESOURCE-ALLOCATION FILE=(NAME=DATEI1,ALLOC=EX).

Task 2 ist die Collector-Task, sie wartet auf die Plattenfreigabe durch Task 1 mit /SECURE-RESOURCE-ALLOCATION DISK=*PAR(VOL=DISK=01,TYPE=D3435,ALLOC=EX), TAPE=(VOL=TAPE01,TYPE=TAPE-C4),WAIT=...

Task 3 wartet in der Secure-Queue auf die Freigabe des Bandes durch Task 2 während der Belegung der Datei DATEI1 `/SECURE-RESOURCE-ALLOCATION TAPE=(VOL=TAPE01,TYPE=TAPE-C4)`.

Lösung:
In diesem Fall würde eine Etablierung von Task 3 als Collector-Task zu einer Freigabe des Bandes TAPE01 durch Task 2 führen (Task 2 verliert seine Collector-Eigenschaft). Dadurch erhält Task 3 sein angefordertes Betriebsmittel und tritt aus der Secure-Queue aus. Nach Freigabe der Betriebsmittel durch Task 3 kann die Anforderung von Task 1 erfüllt werden; nach dessen Freigabe die von Task 2.

Änderung der USER-ALLOCATION

Der Wert von USER-ALLOC für eine Platte entscheidet deren Belegbarkeit durch den Benutzer. Ist „NO“ eingestellt, werden alle Belegungsanforderungen ohne Hinweis für den Operator abgewiesen. Aus diesem Grund ist es für den Operator wichtig zu wissen, in welchen Fällen eine (implizite) Änderung des Parameters stattfindet.

1. SET-DISK-PARAMETER VOL=vsn,USER=....
2. Wenn für eine Platte USER-ALLOC=*STD definiert ist, gilt als tatsächlicher Wert derjenige, der mit SET-DISK-DEFAULTS USER=.... eingestellt wurde. Ändert sich also der Standardwert, ist auch diese Privatplatte davon betroffen.
3. Beim Entziehen einer Platte mit CHANGE-DISK-MOUNT VOL=vsn,ACTION=CANCEL wird USER-ALLOC=*NO eingestellt.
4. Das Beantworten einer Remount-Meldung mit `tsn.N` (Abweisung) hat einen impliziten Cancel der Platte zur Folge (siehe Punkt 3).

Information über Reservierungen

1. Taskspezifisch

   Das Kommando SHOW-RESOURCE-ALLOCATION (SH-RES) gibt Auskunft über folgende Betriebsmittelbelegungen:

   - Unit-Reservierungen
   - Gerätetypreservierungen
   - Band-, Plattenreservierungen (explizit durch SECURE-RESOURCE-ALLOCATION oder implizit durch Eröffnen oder Reservieren von Dateien auf privaten Datenträgern)

2. Systemglobal

   Mit SHOW-DEVICE-STATUS UNIT=*SEL-TYPE(ATTR=*FREE[,TYPE=xxx]) erhält man eine Liste aller Geräte (eines bestimmten Typs), die noch nicht explizit (durch SECURE-RESOURCE-ALLOCATION UNIT=mn) oder implizit (durch Belegung eines darauf montierten Datenträgers) belegt wurden; diese Liste spiegelt aber nicht die tatsächliche Anzahl der noch belegbaren Geräte wider; es gibt nämlich Reservierungen, die noch keinem Gerät (Unit) zugeordnet sind (wie z.B. Typreservierungen mit dem Kommando SECURE-RESOURCE-ALLOCATION).

   Eine Übersicht über die tatsächliche Anzahl von reservierten Geräten und die Art ihrer Reservierung sowie die Anzahl der noch belegbaren Geräte eines Typs kann der Ausgabe des Kommandos SHOW-DEVICE-STATUS INF=*SUM[,TYPE=xxx] entnommen werden.

   SHOW-DEVICE-STATUS UNIT=*SEL-TYPE(TYPE=xxx), INF=*TASK zeigt die belegenden oder reservierenden Tasks und die Anzahl der von ihnen reservierten Geräte.

Vorschläge zur perfomance-günstigen Einstellung von NDM-Parametern

1. Privatplatten, die im DVS-Betrieb eingesetzt werden, werden bei Dateibearbeitung auf Kommandoebene (COPY-FILE, DELETE-FILE, ADD-FILE-LINK, CREATE-FILE) durch das DVS immer wieder neu belegt und freigegeben. Haben diese Belegungen und Freigaben einer Task außerdem immer eine SVL-Neubelegung bzw. eine Freigabe der Privatplatte zur Folge, so muss neben der DVS-Ein-/Ausgabe zusätzlich immer noch eine SVL-Verwaltung (Ein-/Austragen der System-ID im SVL der Platte) stattfinden. Bleibt die Platte jedoch über die Freigabe durch den jeweiligen Benutzer hinaus noch belegt (durch weitere gültige Belegung eines anderen Benutzers oder durch ASSIGN-TIME=*OPERATOR), entfällt der nicht unerhebliche Aufwand für die SVL-Verwaltung. Aus diesem Grund sollte eine für den DVS-Betrieb eingesetzte Privatplatte so lange durch ihre ASSIGN-TIME belegt bleiben, bis sie nicht mehr gebraucht und gezielt vom Operator freigegeben wird. Erreichbar ist dies durch
   - SET-DISK-PAR VOL=vsn,ASS=*OPER
   - SET-DISK-DEFAULTS ASS=*OPER (gilt für alle Platten mit ASS=*STD)

   > ¡ Sollen auch SPECIAL-Anwendungen stattfinden, ist Alternative b) nicht zu empfehlen, da dadurch auch Platten, die für eine spätere SPECIAL-Nutzung montiert wurden, automatisch belegt werden und vor ihrer Bearbeitung durch die SPECIAL-Anwendung vom Operator explizit freigegeben werden müssen (mit SET-DISK-PAR VOL=vsn,ASS=*USER).

2. Für ein Data Center, bei dem der SPOOLOUT auf Band ausgelagert werden soll, sollte MODIFY-MOUNT-PAR UNLOAD=*NO (Voreinstellung) gesetzt sein, da ansonsten nach jedem Auslagern einer SPOOL-Datei das entsprechende Band entladen wird und ggf. vom Operator wieder startklar gemacht werden muss.

5.7 Datenträger-Überwachung

Die Geräteverwaltung unterstützt und überwacht die Platten- und Bandbenutzung, sie wickelt Montier- und Ummontiervorgänge ab und schützt benützte Bänder und Platten vor fehlerhaften Eingriffen.

Sie verwendet dazu zwei permanente Tasks, den DISK-MONITOR (DM) für die Plattenüberwachung und den TAPE-MONITOR (TM) für die Bandüberwachung. Für Funktionen, die die Verfügbarkeit der permanenten Tasks zeitweise beeinträchtigen (z.B. Bandpositionierungen), werden temporäre Tasks gestartet.

Montieren eines Datenträgers vor der Benutzung bzw. bei Belegungsanforderung

Fordert ein Benutzer einen Datenträger an, so wird an der Konsole eine Mount-Meldung ausgegeben, falls dieser noch nicht als montiert erkannt wird (`NKVD013/NKVT013`; abhängig davon, ob eine Platte (D) oder ein Band (T) montiert werden soll).

Erkennt die Datenträger-Überwachung den angeforderten Datenträger als montiert (durch Auswertung des Einschalt-Interrupts vom Gerät), so wird die ausstehende Meldung vom System automatisch beantwortet. Für den Operator entfällt dadurch die Notwendigkeit, die Mount-Meldung zu beantworten.

Ummontieren von Datenträgern während der Benutzung

Wird durch Rekonfiguration ein (defektes) Gerät weggeschaltet (Kommando DETACH), so wird der Operator bei Verfügbarkeit eines Ersatzgerätes von der Datenträger-Überwachung zu einem Ummontieren aufgefordert.

Stehen keine Ersatzgeräte zur Verfügung, so wird eine temporäre Überbelegung akzeptiert, d.h. es wird solange mit einer erneuten Mount-Aufforderung gewartet, bis ein Gerät frei oder zugeschaltet (Kommando ATTACH) wird. Der Operator kann jedoch dieses Warten durch expliziten Entzug des Datenträgers abbrechen (CANCEL-Funktion des Kommandos CHANGE-TAPE-MOUNT).

Schutz von Datenträgern gegen fehlerhafte Geräteeingriffe und Repositionierung von Bändern

Jeder Eingriff des Operators an einem Band- oder Plattengerät (Entladen des Bandes, Ausschalten des Gerätes) mit einem benutzten Datenträger führt zu einer zeitweiligen Ein-/Ausgabe-Sperre für den Benutzer.

Wird bei einer Ein-/Ausgabe-Anforderung festgestellt, dass der Datenträger nicht mehr verfügbar ist, führt dies zu einer Aufforderung an den Operator, den Datenträger wieder bereitzustellen (Remount-Meldung).

Nach einer erneuten Zuweisung des Datenträgers können die Ein-/Ausgaben fortgesetzt werden. Bei Bändern findet eine automatische Repositionierung statt.
Werden den Monitoren Hardware-Resets der Geräte gemeldet, führt dies ebenfalls zur Identifizierung der Datenträger und bei Bändern mit PHASE=IN-USE zu deren Repositionierung.

(Automatischer) Abbau einer Datenträgerüberbelegung

Wird einem belegten Band im USE-MODE=DMS das Gerät entzogen (DETACH UNIT=..., FORCE=*Y), ohne ein Ersatzgerät zur Verfügung zu haben, entsteht eine temporäre Überbelegung (SH-TAPE: Ausgabespalte ACTION=NO DEVICE), d.h. es sind mehr Datenträger dieses Gerätetyps in Benutzung, als Geräte verfügbar sind (SHOW-TAPE-STATUS: Spalte PHASE=IN-USE).

Welcher Datenträger zeitweilig nicht verfügbar sein soll, kann der Operator ggf. durch das Kommando CHANGE-TAPE-MOUNT ...,EXCHANGE=(...) beeinflussen bzw. bestimmen. Wird ein Gerät eines passenden Typs frei, so wird der Operator von der Datenträger-Überwachung automatisch zum Montieren des Datenträgers aufgefordert (mit ACTION=NO DEVICE).

5.7.1 Geräteauswahlmechanismus für Bandgeräte

Für die Auswahl durchsucht NDM die Gerätetabelle von Beginn an, d.h. in der bei der Hardware-Generierung festgelegten Reihenfolge, und wählt das erste passende, freie Gerät aus (BEST-GENERATED-DEVICE). Das hat zur Folge, dass die ersten in der Tabelle stehenden Geräte häufiger für eine Nutzung ausgewählt werden und – gerade in Verbindung mit roboterbedienten Magnetbandarchivsystemen wie ROBAR, die sich immer an den Mount-Vorschlag von NDM halten – eher abgenutzt werden, als Geräte am Ende der Tabelle.

Für Benutzer, die an einer gleichmäßigen Nutzung ihrer Bandgeräte interessiert sind, wird die Funktion „wrap around“ angeboten. Sie kann über das Kommando MODIFY-MOUNT-PARAMETER NEXT-TAPE-MOUNT=LEAST-RECENTLY-USED-DEVICE eingeschaltet werden. NDM wählt dann aus den passenden freien Geräten in der Gerätetabelle dasjenige aus, welches am längsten unbenutzt war.

Für Benutzer, die an einer auf den Ein-/-Ausgabe-Durchsatz hin optimierten Geräteauswahl interessiert sind, wird das Kommando MODIFY-MOUNT-PARAMETER ...,NEXT-TAPE-MOUNT=*BY-CONTROLLER angeboten. Aus den passenden freien Geräten kann ein Gerät ausgewählt werden, an dessen Steuerung die wenigsten Geräte in Benutzung sind. Damit wird eine gleichmäßige Verteilung der benutzten Geräte an den vorhandenen Steuerungen und Kanalpfaden erzielt. Die Geräteauswahl im NDM berücksichtigt nur die eigenen Gerätebelegungen. In Verbindung mit der IORM-Funktion DDAL (siehe "IORM: Steuerung von I/O-Ressourcen") wird die Optimierung auf alle VM2000-Gastsysteme eines Systems erweitert.

Die Standardeinstellung des Geräteauswahlmechanismus ist NEXT-TAPE-MOUNT= *BEST-GENERATED-DEVICE, d.h. NDM durchsucht die Gerätetabelle vom Beginn an, in der bei der Hardware-Generierung vorgegebenen Reihenfolge, und wählt das erste passende, freie Gerät aus.

Geräteauswahl unter Berücksichtigung von Lagerorten

Insbesondere zur Unterstützung robotergesteuerter Archive in BS2000 (Softwareprodukt ROBAR, siehe Abschnitt „Archivsysteme") wurde im NDM eine Lagerortverwaltung entwickelt.
Datenträger und die Geräte, auf denen sie montiert werden sollen, können in BS2000 Lagerorten zugeordnet werden. Die Vereinbarung von Lagerorten ermöglicht es, Datenträger und Geräte aus einer genau vorgegebenen Menge auszuwählen und zu reservieren.

Die Zuordnung von Bandgeräten zu den Lagerorten erfolgt mit dem Kommando ADD-DEVICE-DEPOT. Es sind 1024 Bandgeräte pro Lagerort erlaubt. Die Zuordnung einer VSN zu einem Lagerort wird durch das Softwareprodukt MAREN verwaltet.
Die dort festgelegten Lagerorte müssen mit den Angaben des Operators im Kommando ADD-DEVICE-DEPOT übereinstimmen.

Jeder Lagerort ist im MAREN-Katalog durch seinen Namen (max. 8 Zeichen), seinen Typ (REMOTE oder LOCAL) und seinen Bedienmodus gekennzeichnet.

> ¡ Die Arbeit mit Lagerorten ist nur in Verbindung mit MAREN (siehe "Bandverwaltung mit MAREN") möglich.

Zum Verständnis der Auswirkungen, die das Kommando ADD-DEVICE-DEPOT auf die Geräteauswahl im NDM bei Belegungsanforderungen hat, werden im Folgenden die Grundprinzipien kurz beschrieben. Dabei wird nach der Art der Belegungsanforderung (Bandgeräte-Anforderung und Band-Volume-Anforderung) unterschieden.

Geräteauswahl bei Bandgeräte-Anforderung

Die Anforderung eines Bandgerätes kann von Seiten des Benutzers auf drei Arten erfolgen:

1. Anforderung über den mnemotechnischen Gerätenamen (MN)

Die Anforderung des Gerätes ist bei Bezeichnung eines mnemotechnischen Gerätenamens explizit vorgegeben; es erfolgt keine Auswahl.

2. Anforderung mit Angabe eines Lagerortes

   Die Bandgeräteanforderung erfolgt über die Angabe eines Geräte- oder Volumetyps mit zusätzlicher Angabe eines Lagerortes.

   Erfolgt die Anforderung über das Kommando SECURE-RESOURCE-ALLOCATION, berücksichtigt NDM bei der Auswahl nur den angegebenen Lagerort. Ist hier kein Gerät verfügbar, wird der Job des Benutzers abgewiesen.

   Erfolgt die Anforderung über eine interne Schnittstelle, die von ARCHIVE genutzt wird, musste der Aufrufer den Lagerort zuvor mit Hilfe von MAREN bestimmen. Falls dieser Lagerort dem NDM bekannt ist, d.h. über das Kommando ADD-DEVICE-DEPOT eingerichtet wurde, wählt NDM ein Bandgerät aus diesem Lagerort aus. Sonst erfolgt ein Zugriff auf die Gerätemenge, die den angegebenen Typ unterstützt, jedoch keinem bestimmten Lagerort zugeordnet ist. Diese Menge wird als „Restpool“ bezeichnet.

3. Anforderung ohne Angabe eines Lagerortes

   Die Bandgeräteanforderung erfolgt über die Angabe eines Geräte- oder Volume-Typs ohne zusätzliche Angabe eines Lagerortes.

   Erfolgt die Anforderung über das Kommando SECURE-RESOURCE-ALLOCATION z.B. mit dem Operanden TYPE=TAPE-C4,...,LOCATION=*USER-DEF, versucht NDM, über MAREN einen Standard-Lagerort für den angegebenen Volume-Typ oder einen von dem angegebenen Gerätetyp unterstützten Volume-Typ zu ermitteln.
   Kennt das NDM diesen Lagerort, so wählt es ein Bandgerät aus diesem Lagerort aus. Sonst erfolgt ein Zugriff auf den „Restpool“.

   Erfolgt die Anforderung über das Kommando SECURE-RESOURCE-ALLOCATION mit dem Operanden LOCATION=*NONE, entfällt der MAREN-Aufruf und es wird unmittelbar auf den „Restpool“ zugegriffen.

Implizite Geräteauswahl bei Band-Volume-Anforderung

Die Anforderung eines Band-Volumes kann von Seiten des Benutzers auf zwei Arten erfolgen:

1. Anforderung mit Angabe der VSN

   Erfolgt die Anforderung mit Angabe der VSN, wird der in MAREN für diese VSN definierte Lagerort verwendet. Falls in MAREN für diese VSN kein Eintrag existiert, liefert MAREN entweder einen über die MAREN-Exit-Routine ermittelten oder einen Standard-Lagerort.

2. Anforderung ohne Angabe der VSN

   Die Band-Volume-Anforderung erfolgt ohne Angabe einer VSN.

   NDM versucht in diesem Fall, über MAREN einen Standard-Lagerort für den angegebenen Volume-Typ oder einen von dem angegebenen Gerätetyp unterstützten Volume-Typ zu ermitteln. Kennt das NDM diesen Lagerort, so wählt es ein Bandgerät aus diesem Lagerort aus. Sonst erfolgt ein Zugriff auf den „Restpool“.

5.8 Parallel Access Volumes (PAV, SU /390)

Im BS2000 wird jede Platte durch einen Tabelleneintrag in der PDT (Physical Device Table) repräsentiert. Jeder I/O-Auftrag für eine Platte wird vom Gerätetreiber in Form eines Kanalprogrammes an die I/O-Steuerung IOCNTRL zur Ausführung weitergereicht.

Wenn im PDT-Eintrag angezeigt wird, dass das Gerät nicht aktiv ist, wird der I/O-Auftrag gestartet. Jeder weitere I/O-Auftrag für dieses Gerät wird von IOCNTRL zunächst nur in eine Gerätewarteschlange eingereiht. Erst wenn das Gerät nicht mehr aktiv ist, wird der nächste Auftrag aus der Warteschlange gestartet.
Da zu einem Zeitpunkt also nur ein I/O-Auftrag pro Platte erlaubt ist, setzt sich die Gesamtdauer einer I/O zusammen aus der eigentlichen I/O-Dauer in der Hardware und der Wartezeit.

Hohe Wartezeiten können dadurch verursacht werden, dass:

- mehrere Anwendungen gleichzeitig mit hoher I/O-Last auf einer Platte arbeiten
- eine Anwendung ihre I/O-Aufträge asynchron an IOCNTRL abgibt

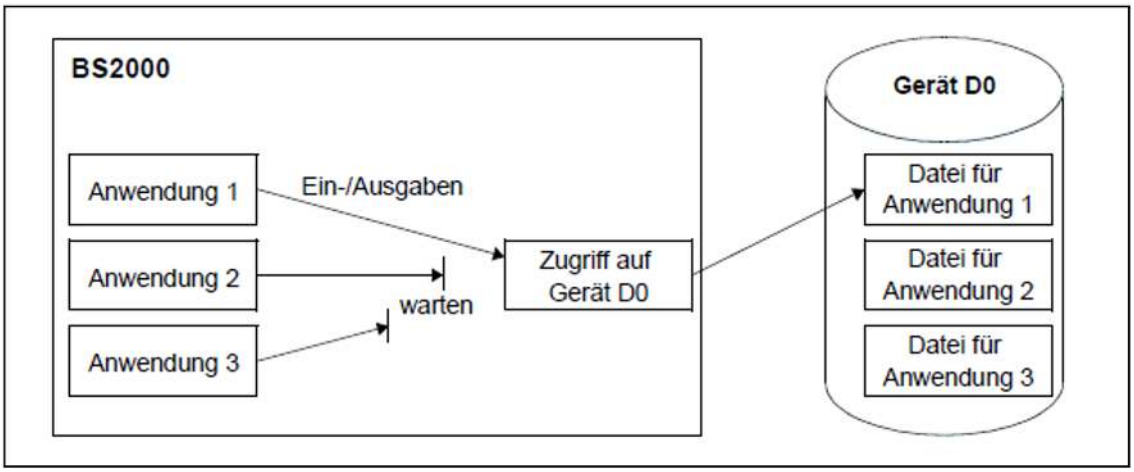


Bild 5: Mehrere Anwendungen mit I/O-Last auf einer Platte (traditionell)

> Auf der SU x86 kann IOCNTRL standardmäßig mehrere Aufträge für emulierte Platten parallel starten (RSC-Funktion). Es müssen keine besonderen Einstellungen getroffen werden.

Für die Realisierung der PAV-Funktionalität nutzt BS2000 die Tatsache aus, dass am Fibre Channel ein Gerät einen I/O-Auftrag bereits annehmen kann, während noch eine I/O aktiv ist. Im RAID-System darf jedoch das „Command Queuing“ nicht ausgeschaltet sein. PAV kann ohne Eingriffe am RAID-System in Betrieb genommen werden.

Ein logisches Parallel Access Volume wird durch ein sog. Basis-Gerät (BASE) mit bis zu 7 sog. Alias-Geräten (ALIAS) repräsentiert.

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).

Einsatzempfehlung für FastDPAV

Für hohe IO-Lasten auf SU /390 wird dringend der Einsatz von Schnellem Dynamischen PAV (FastDPAV) empfohlen.

Standard-PAV

Basis- und Alias-Geräte werden bei der Hardware-Generierung (siehe "Randbedingungen der Hardware-Generierung" in Handbuch „Systeminstallation“ [55]) als eigenständige Geräte gleichen Typs mit gleichen Logical Unit Numbers aber unterschiedlichen Unit-Adressen und unterschiedlichen mnemotechnischen Gerätebezeichnungen an der gleichen logischen Steuerung generiert.

Die Generierung von Alias-Geräten erfolgt mit IOGEN (siehe "Generieren der IO-Konfigurationsdaten (IOCF) mit IOGEN" in Handbuch „Systeminstallation“ [55]) oder im laufenden Betrieb mit dem Kommando ADD-IO-UNIT, Operand PAV-ALIAS-DEVICE.

Den Zusammenhang zwischen Basis-Gerät und Alias-Geräten ermittelt BS2000 dynamisch beim Zuschalten eines Geräts mit dem Kommando ATTACH-DEVICE. Erkennt BS2000 ein Basis-Gerät, dann wird nach zugehörigen Alias-Geräten an derselben logischen Steuerung gesucht. Diese Alias-Geräte werden implizit zugeschaltet. Alias-Geräte können auch explizit mit ATTACH-DEVICE zugeschaltet werden. Ein Alias-Geräts kann aber nicht zugeschaltet werden, wenn das zugehörige Basis-Gerät nicht zugeschaltet ist.

Beim Wegschalten eines Basis-Geräts mit dem Kommando DETACH-DEVICE werden zugehörige Alias-Geräte implizit weggeschaltet. Ein Alias-Gerät kann auch explizit mit DETACH-DEVICE weggeschaltet werden.

Wenn ein Basis-Gerät und seine Alias-Geräte zugeschaltet sind, dann kann IOCNTRL I/O-Aufträge für das PAV parallel über Basis- und Alias-Geräte starten.

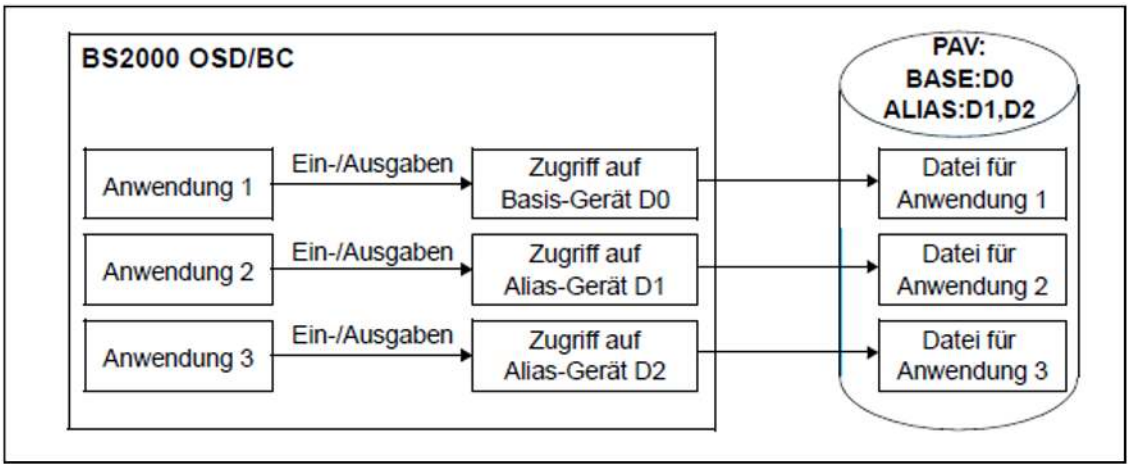


Bild 6: Mehrere Anwendungen mit I/O-Last auf einer Platte (PAV)

Die Belegung eines PAV-Volumes (Kommando IMPORT-PUBSET bzw. SET-DISK-PARAMETER) erfolgt über das Basis-Gerät. Auch die Aufträge für Lese- oder Schreib-I/Os in BS2000 erfolgen über das Basis-Gerät.
Erst IOCNTRL verteilt die I/Os in Abhängigkeit der Auslastung von Basis- und Alias-Geräten auf eines von diesen. Dabei bevorzugt IOCNTRL im native mode bei gleicher Auslastung das Basis-Gerät.

Mit dem Kommando SHOW-DEVICE-CONFIGURATION ...,INFORMATION=*PAV können Informationen über alle PAV-Geräte ausgegeben werden.

Hinweise zum Einsatz unter VM2000

Um unter VM2000 eine möglichst gleichmäßige Verteilung der I/Os für Shared-Platten zu erreichen, bevorzugt IOCNTRL in den Gastsystemen das Basis- oder ein Alias-Gerät in Abhängigkeit vom VM-Index der virtuellen Maschine.
Basis- oder Alias-Gerät können jedoch auch Gastsystem-spezifisch explizit definiert werden. Dazu steht das Kommando MODIFY-IO-UNIT UNIT-DEVICE=*DEVICE(NAME=..., STATE=*PAV-PREFERRED-DEVICE(...)) zur Verfügung. Für jedes Gastsystem kann ein bevorzugtes Alias-Gerät für die I/O-Ausführung festgelegt werden.

Ausführliche Informationen zu PAV unter VM2000 finden Sie im Handbuch „VM2000“ [60].

Hinweise zur Performance

PAV bringt enorme Verbesserungen bei TP- und Batch-Betrieb, wenn ein großer Teil der Daten im Cache des Plattenspeichersystems liegt. Cache-Hits werden gleichzeitig bedient, parallel dazu kann ein Cache-Miss mit physikalischem Plattenzugriff ausgeführt werden. Für eine stark ausgelastete Platte mit n-pfadigem Anschluss kann es daher sinnvoll sein, n-1 Alias-Geräte zu generieren. Damit kann der n-fache Durchsatz einer Platte ohne Alias-Geräte erreicht werden.

Bei der Datensicherung muss davon ausgegangen werden, dass der Cache nicht ausreicht, die Daten also beim Sichern von den Platten geholt, bzw. beim Restore auf die Platten geschrieben werden müssen. Dies geschieht innerhalb des Plattenspeichersystems hoch-optimiert durch asynchrones „Read Ahead“ bzw. „Delayed Fast Write“. Die Leistung dieser Funktionen beschränkt den Durchsatz bei der Datensicherung.

Für eine optimale Datensicherung wird ein Einsatz „großer“ Raid-Systeme mit schnellen Platten empfohlen sowie die Generierung von ausreichend vielen Alias-Geräten.

Extended-PAV (XPAV)

Mit der Funktion „Extended-PAV (XPAV)“ wird die Einschränkung, dass ein Alias-Gerät in derselben logischen Steuerung wie das Basis-Gerät liegen muss, aufgehoben. Zusätzlich zu einer vorhandenen logischen Steuerung können eine oder mehrere logische Steuerungen mit 256 Alias-Geräten konfiguriert werden.

XPAV eignet sich insbesondere für Konfigurationen, in denen PAV nicht von Anfang an eingeplant wurde oder in denen keine oder nur wenige freie Adressen für Alias-Geräte zur Verfügung stehen.

Zwei XPAV-Varianten sind möglich:

1. Wenn die vorhandene logische Steuerung und die zusätzliche Steuerung mit den Alias-Geräten an denselben Steuerungsports generiert werden, dann ist kein manueller Eingriff in der Plattensteuerung und auch keine zusätzliche Verkabelung erforderlich.
2. Wenn die vorhandene logische Steuerung und die zusätzliche Steuerung mit den Alias-Geräten an unterschiedlichen Steuerungsports generiert sind, dann müssen die Volumes in der Platten-Steuerung zusätzlichen Ports zugewiesen werden. Auch kann eine zusätzliche Verkabelung für die Alias-Geräte erforderlich werden.

Schnelles Dynamisches PAV (FastDPAV)

Für die Realisierung der Funktionalität des Schnellen Dynamischen PAV (FastDPAV) wird genutzt, dass Server Units SU /390 (ab SU710) eine Modifikation der LUN für Alias-Geräte beim Starten einer I/O unterstützen.

Vorteile von FastDPAV gegenüber Statischem PAV:

- Verlangt keine vorausschauende Planung auf die zukünftige Geräteauslastung durch Generierung geeignet vieler Alias-Geräte pro Volume.
- FastDPAV benötigt deutlich weniger Alias-Geräte, weil es nicht erforderlich ist, für jedes Volume die maximal benötigten Alias-Geräte vorzuhalten.
- Mit FastDPAV sind mehr ‚normale‘ Volumes, Clones und Snaps am Kanal möglich.
- FastDPAV ermöglicht größere Volumes bei gleicher I/O-Performance.

Beim FastDPAV wird für eine Menge von Logischen Volumes mit identischen Kanalpfaden (Verbindung vom Kanal zum Port am Plattenspeichersystem) ein Pool von FastDPAV-Alias-Geräten generiert ohne feste Zuordnung zu einem dieser Logischen Volumes. Für ein FastDPAV-Alias-Gerät wird die LUN jeweils erst beim Starten einer I/O vorgegeben/modifiziert.

Hinweise zum Einsatz

Im Native-Betrieb werden alle generierten FastDPAV-Alias-Geräte vom BS2000 automatisch für die I/O-Nutzung aktiviert.

Im VM-Betrieb werden von den generierten FastDPAV-Alias-Geräten diejenigen vom BS2000 aktiviert, die dem Gastsystem (bzw. Monitorsystem) exklusiv zugewiesen wurden. Die Zuweisung kann explizit (mit ADD-VM-DEVICES) erfolgen oder implizit (mit ATTACH-DEVICE), falls das System dafür berechtigt ist.

Bei der I/O-Ausführung berücksichtigt BS2000 nur die zuvor aktivierten FastDPAV-Alias-Geräte.

Eine Deaktivierung wird mit DETACH-DEVICE ermöglicht. Im VM-Betrieb erfolgt zusätzlich ein Entzug des Alias-Gerätes von der virtuellen Maschine, falls es nur implizit zugewiesen wurde.

Technisch gesehen kann FastDPAV auch in Kombination mit statischem PAV betrieben werden, jedoch wird der alleinige Einsatz von FastDPAV empfohlen.

5.9 Verwaltung privater Datenträger

- Bandverwaltung mit MAREN
- Einsatzmöglichkeiten privater Platten

5.9.1 Bandverwaltung mit MAREN

Für die Bandverwaltung im Data Center steht der Systembetreuung das Softwareprodukt MAREN zur Verfügung. MAREN speichert alle Informationen über die Bänder in einer ISAM-Datei, dem MAREN-Katalog.

Ohne MAREN ist das Arbeiten mit Lagerorten (siehe "Geräteauswahl unter Berücksichtigung von Lagerorten" im "Geräteauswahlmechanismus für Bandgeräte") nicht möglich.

MAREN-Katalog

Um die Informationen im MAREN-Katalog stets aktuell zu halten, arbeitet MAREN mit BS2000 zusammen. MAREN registriert jeden DVS-Zugriff. Der MAREN-Katalog wird bei jeder Verarbeitung eines Bandes aktualisiert.

Der MAREN-Katalog basiert auf den Archivnummern (VSN) der Bänder. Die Archivnummern müssen eindeutig sein, wobei kein spezielles Nummernkreisschema vorausgesetzt wird.

Freibandzuweisung und automatische Initialisierung

Eine wichtige Komponente des MAREN-Systems ist die so genannte automatische Freibandzuweisung MARENUCP. Damit können z.B. im Fall einer unspezifizierten Bandanfordung („SCRATCH“-Bänder) freie Bänder aus dem MAREN-Bandpool automatisch zugewiesen werden. Dabei werden gleichzeitig die verarbeiteten Bänder im MAREN-Katalog als reserviert gekennzeichnet.
MARENUCP prüft, ob ein Band, das angefordert und vom Operator montiert wurde, initialisiert werden muss. Falls notwendig, wird zuerst die Initialisierung durchgeführt und anschließend die Reservierung und Zuweisung vorgenommen.

Schnittstellen zur BS2000-Bandverarbeitung

Bei der Magnetband-Reservierung und -Verarbeitung über das BS2000-DVS wird automatisch das MAREN-System für folgende Aufgaben aktiv:

- Kontrolle der Zugriffsberechtigung
- Prüfung auf Verfügbarkeit
- Aktualisierung der Archiveinträge
- Ausgabe von Transportmeldungen
- Einsetzen der Geräteparameter

Der Aufruf dieser Prüfroutinen kann von folgenden Ereignissen veranlasst werden:

- Band belegen
- TFT-Eintrag erstellen
- Datei eröffnen
- Spulen wechseln
- Datei schließen
- Band freigeben

Kontrolle der Zugriffsberechtigung

Vor jeder Verarbeitung eines Bandes kontrolliert MAREN, ob der Aufrufer zugriffsberechtigt ist. Folgende Bedingungen müssen vor jedem Zugriff erfüllt sein:

- Falls das Band einer fremden Benutzerkennung zugeordnet ist, muss es im MAREN-Katalog das Attribut USER-ACCESS=*ALL-USERS besitzen
  (Ausnahme: Aufrufer unter TSOS).
- Falls das Band durch ein Kennwort geschützt ist, muss dieses Kennwort angegeben worden sein.
- Falls das Band mit einem ARCHIVE-Directory in Verbindung steht, dann darf auf dieses Band nur mit ARCHIVE zugegriffen werden.
- Bei Ausgabe auf Band muss der schreibende Zugriff erlaubt sein; das Attribut FOREIGN-READ-ONLY darf im MAREN-Katalog nicht hinterlegt sein
  (Ausnahme: Aufrufer unter TSOS).
- Bei Ausgabe auf Band muss das Dateifreigabedatum kleiner oder gleich dem Tagesdatum sein.
- Falls MAREN-Exitroutinen im Einsatz sind, müssen diese den Zugriff auf das Band erlauben.

Ist eine dieser Bedingungen nicht erfüllt, wird der Zugriff auf das betreffenden Band abgelehnt und eine entsprechende Meldung ausgegeben.

Prüfung auf Verfügbarkeit

Vor jedem Zugriff auf ein Band prüft MAREN, ob die folgenden Voraussetzungen für die Verfügbarkeit erfüllt sind:

- Die Archivnummer des Bandes muss im MAREN-Katalog eingetragen sein.
- Die Archivnummer muss laut Attribut DEVICE-TYPE ein Band bezeichnen.
- Das Band darf nicht ausgeliehen sein.
- Das Band muss für den Aufrufer reserviert sein.
- Das Band muss im lokalen Data Center verfügbar sein; d.h der in einem speziellen Feld des Archiveintrags enthaltene Lagerort darf in der so genannten Lagerorttabelle nicht als „remote“ gekennzeichnet sein.
- Das Band darf nicht gerade an einem anderen System verarbeitet werden (die Prüfung auf eine gerade laufende Verarbeitung am eigenen System wird nicht von MAREN, sondern von BS2000 durchgeführt).
- Beim Verarbeitungsmodus INPUT (außer bei Zugriff über ARCHIVE) müssen die Dateinamen im MAREN-Archiveintrag und in der TFT übereinstimmen. Dabei werden die Namensteile Katalogkennung, Benutzerkennung und Version ignoriert.

Installation und Einsatz von MAREN im Data Center sind in den Handbüchern zu „MAREN“ [31] ausführlich beschrieben.

5.9.2 Einsatzmöglichkeiten privater Platten

In BS2000 können Platten grundsätzlich als private Platte oder als gemeinschaftliche Platte genutzt werden.

Vor dem ersten Einsatz müssen die Datenträger mit dem Dienstprogramm VOLIN initialisiert werden. Dabei wird u. a. das maximal 6-stellige Datenträgerkennzeichen (Archivnummer) in den entsprechenden Kennsatz (Standard Volume Label) geschrieben. Anhand dieses Datenträgerkennzeichens (Volume Serial Number, VSN) können die privaten und gemeinschaftlichen Platten unterschieden werden.
Die VSN einer privaten Platte ist, mit der Einschränkung, dass sie nicht mit der einer gemeinschaftlichen Platte übereinstimmen darf, frei wählbar. Die Regeln und Möglichkeiten zur Vergabe einer VSN für gemeinschaftliche Platten sind im Abschnitt „Pubset-Eigenschaften" erläutert.

Für private Platten gibt es zwei Anwendungsmöglichkeiten:

- DVS-Anwendungen
  Die Platte ist durch eine oder mehrere DVS-Anwendungen belegt.
  Dabei ist der Taskbelegungsmodus standardmäßig task-shareable. Eine task-exklusive Belegung erreicht man mit dem Kommando SECURE-RESOURCE-ALLOCATION. Der Systembelegungsmodus richtet sich nach den Generierungseigenschaften des Plattengerätes, in dem die Platte montiert ist. Mit dem Operatorkommando SET-DISK-PARAMETER kann der Systembelegungsmodus generierungsunabhängig und plattenspezifisch eingestellt werden.
- Sonderanwendungen (SPECIAL-Anwendungen)
  Die Platte ist durch eine privilegierte Anwendung – z.B. VOLIN, FDDRL – belegt. Die SPECIAL-Anwendungen sind immer task- und system-exklusive Belegungen. Die Belegungsdauer wird von der SPECIAL-Anwendung bestimmt, sie kann vom Operator nicht beeinflusst werden.

5.10 IORM: Steuerung von I/O-Ressourcen

Das Dienstprogramm IORM verbessert die I/O-Eigenschaften von BS2000 im Native- und im VM2000-Betrieb.

In IORM (und dem IORM-Subsystem) 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 mit ETERNUS CS
  (Dynamic Device Allocation)
- IOLVM (SU /390):
  Begrenzung der I/O-Aufnahme einzelner VM2000-Gastsysteme
  (I/O Limit for Virtual Machines)

In IORM gibt es noch die folgende Funktion:

- DPAV (/390-Server) Einstellungen für PAV und FastDPAV:
  Die vorhandenen Aliasgeräte mit deren Auslastung können abgefragt werden, und für FastDPAV kann die Alias-Geräte-Anzahl für einzelne Platten eingeschränkt werden (Dynamic Parallel Access Volume)

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

Die IORM-Funktionen IOPT und IOLVM steuern Plattengeräte.
Die IORM-Funktion DDAL steuert Bandgeräte.

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 eingesetzt werden.
IORM arbeitet aber nicht über mehrere Server Units hinweg.

Eine detaillierte Beschreibung von IORM finden Sie im Handbuch „Dienstprogramme“ [15].

5.11 Anzeigen und prüfen der SAN-Konfiguration

Die Server Units werden mit modernen Speichersystemen über die Fibre Channel verbunden. Dabei werden die Speichersysteme in der Regel nicht direkt an den Fibre Channel Adapter einer Server Unit angeschlossen, sondern über einen **Switch**. Ein solcher FC-Switch ermöglicht gleichzeitig mehrere Verbindungen zwischen den Geräten, die an seinen Ports angeschlossen sind. An einen FC-Switch können über spezielle Anschlüsse wiederum andere Switches angeschlossen werden. Ein Netz, das aus einem oder mehreren FC-Switches gebildet wird, heißt Fabric. Ein Netz aus mehreren Speichersystemen, die mit FC-Switches verbunden sind, wird als Storage Area Network (SAN) bezeichnet.

Aus Sicht von BS2000 sind die FC-Switches transparent. BS2000 verwendet die Steuerungen und Geräte, die über Fibre Channel angeschlossen sind, ohne Informationen über die Verbindungen in der Fabric zu haben.

Wenn es zu Problemen beim Zuschalten von Geräten oder zu Fehlern im laufenden Betrieb kommt, ist es oft schwer, die Ursache hierfür zu erkennen. Eine INOP- oder NINT-Meldung der Gerätefehlerbehandlung kann durch Verbindungsstörungen an beliebiger Stelle im SAN hervorgerufen werden. Möglicherweise kann ein Gerät gar nicht erst zugeschaltet werden, weil die in BS2000 generierten Pfade oder WWPNs (World Wide Port Number) physikalisch nicht vorhanden sind, oder weil die generierten Verbindungen zwischen Kanal und Steuerung in den Switches (Zoning) nicht zugelassen werden.

FC Netzwerke im SE Manager

Informationen zu den FC Netzwerken, also zu den einzelnen FC-Fabrics, sowie den Verbindungen zwischen Server, Storage und den SAN-Komponenten werden im SE Manager unter „Hardware“ > „FC Netzwerke“ angezeigt. Zu den FC-Fabrics umfasst dies Informationen über die Fabrics und darin enthaltenen Switches. Die Verbindungen der Switches innerhalb einer Fabric, wie auch aller Ports und deren Verbindungen („Link-Neighbours “) und Zustände werden ebenfalls angezeigt.

Die Darstellung im SE Manager umfasst detailliert die kompletten FC-Verbindungen zwischen Server und Storage und eine Abbildung auf die Ressourcen in BS2000 wie auch im Storage-System.

Ausführliche Informationen dazu finden Sie im Hilfesystem des SE Manager und im Handbuch „Bedienen und Verwalten“ [57].

Die Informationen bieten Unterstützung bei:

- Erkennung von Generierungsfehlern (SU /390)
- Erkennung von Verbindungsproblemen zwischen Server und Storage
- Lokalisierung von Fehlerzuständen im SAN

6 BS2000-Benutzerverwaltung

Die BS2000-Benutzerverwaltung eines Pubsets wird über den so genannten Benutzerkatalog realisiert.

Dieses Kapitel beschreibt

- Struktur eines Benutzerkatalogs
- Führen des Benutzerkatalogs
- Sicherungskonzept für Benutzerkataloge
- Benutzerkatalog und SMS-Konzept
- Bulletin-Datei (Logon-Informationsdatei)

Details zu Pubsets finden Sie im Kapitel „Pubset-Verwaltung".

Kommando	**Bedeutung**
ADD-USER	Eintrag im Benutzerkatalog erstellen
LOCK-USER	Zugang zum System sperren
MODIFY-USER-ATTRIBUTES	Eintrag im Benutzerkatalog ändern
MODIFY-USER-PUBSET-ATTRIBUTES	Pubset-spezifische Benutzerattribute einer Benutzerkennung ändern
REMOVE-USER	Eintrag eines Benutzers im Benutzerkatalog löschen
SHOW-USER-ATTRIBUTES	Informationen aus dem Benutzerkatalog anfordern
UNLOCK-USER	Zugangssperre zum System aufheben
Makro	**Bedeutung**
RDUID	Benutzerkennung abfragen
SRMUINF	Daten aus dem Benutzerkatalog in einen Bereich übertragen

Tabelle 13: Schnittstellenübersicht zur Verwaltung des Benutzerkataloges

6.1 Struktur eines Benutzerkatalogs

Auf jedem Pubset existiert unter der Kennung TSOS die Datei SYSSRPM. Diese stellt den Benutzerkatalog dar, in dem Informationen zu Benutzerkennungen von BS2000 aufbewahrt werden.

Zusätzlich existiert unter der Kennung TSOS die Datei SYSSRPM.BACKUP (auch .BACKUP genannt). Die Datei .BACKUP ist eine Kopie des Benutzerkatalogs. Sie fungiert als Hilfsdatei, wenn der Benutzerkatalog gesichert oder mit Hilfe der Sicherung rekonstruiert werden soll.

Beide Dateien werden beim Importieren eines Pubsets von einer Systemtask geöffnet und von dieser beim Export wieder geschlossen. Damit ist ein direkter Zugriff auf die Dateien nicht möglich.

Die Datei $TSOS.SYSSRPM

Die Systemdatei $TSOS.SYSSRPM ist eine NK-ISAM-Datei.

Auf sie wird aus einer Server-Task zugegriffen (SRPM-Tasks TSN RP01 oder RP02). Mit dem Import wird die Datei durch die Server-Task eröffnet und bleibt bis zum Export im geöffneten Zustand. Ein direkter Zugriff auf $TSOS.SYSSRPM ist nicht möglich.

Die Datei $TSOS.SYSSRPM enthält Folgendes:

- Informationen zur Zugangskontrolle für Benutzerkennungen
  (auch bei erweitertem Zugangsschutz mit SECOS)
- Systemglobale Privilegien
- BS2000-Benutzergruppen

Die Datei SYSSRPM stellt den Benutzerkatalog dar.

Benutzerkatalog des Home-Pubsets

Der Benutzerkatalog des Home-Pubsets enthält **system-spezifische** Daten, wie z.B. Zugriffsrechte zu BS2000 (Benutzerkennung, Abrechnungsnummer, Kennwort der Benutzerkennung, SPOOLOUT-Klasse, Versandanschrift usw.) und die Standardkatalog-Kennung.

Benutzerkataloge der importierten Pubsets

In den Benutzerkatalogen der importierten Pubsets sind **pubset-spezifische** Daten verankert. Sie enthalten u.a. Informationen, die den einzelnen Benutzer auf dem jeweiligen Pubset betreffen, z.B.:

- den maximal zur Verfügung stehenden Speicherplatz
- ggf. die Erlaubnis, diese Grenze zu überschreiten
- Katalogkontingente

Die system-spezifischen Informationen des Home-Pubsets können auch in den Benutzerkatalogen der importierten Pubsets vorhanden sein; für Überprüfungen wird allerdings nur der Benutzerkatalog des Home-Pubsets vom System herangezogen. Die system-spezifischen Daten auf den importierten Pubsets werden nur dann benötigt, wenn diese Pubsets als Reserve-Home-Pubsets zur Systemeinleitung von BS2000 und für den Systemlauf benutzt werden.

6.2 Führen des Benutzerkatalogs

Der Benutzerkatalog wird beim erstmaligen Importieren eines Pubsets (nach dessen Erstellung mit SIR) mit dem Kommando IMPORT-PUBSET ACTUAL-JOIN=*FIRST eingerichtet. Er enthält dann standardmäßig folgende Benutzerkennungen:

TSOS	Benutzerkennung der Systemverwaltung
SYSPRIV	Benutzerkennung für die Vergabe von Privilegien bei Einsatz des Softwareprodukts SECOS
SYSDUMP	Benutzerkennung, unter der Systemdumps abgelegt werden
SYSOPR	Benutzerkennung für das Operating
SYSSPOOL	Benutzerkennung für die Spoolverwaltung
SERVICE	Benutzerkennung für den Service. Unter dieser Benutzerkennung werden vom HW- und SW-Service spezielle Programme zur Betriebssicherung eingesetzt
SYSGEN	Benutzerkennung für die Hardware-Generierung
SYSHSMS	Benutzerkennung für das HSMS-Datenarchiv
SYSSNAP	Benutzerkennung, unter der SNAP-Dumps abgelegt werden
SYSUSER	Benutzerkennung für Userdumps, die nicht unter der Benutzerkennung des Verursachers abgelegt werden können oder sollen
SYSAUDIT	Benutzerkennung für REPLOG-Verwaltung sowie für die SAT-Auswertung und SAT-Datei-Verwaltung bei Einsatz von SECOS
SYSNAC	Benutzerkennung für Network Administration Center (Netzverwaltung)
SYSROOT	Benutzerkennung für die POSIX-Verwaltung
SYSSNS	Benutzerkennung für den SPOOL Notification Service
SYSMAREN	Benutzerkennung für die MAREN-Verwaltung
SYSSOPT	Benutzerkennung für das Produkt SPACEOPT
SYSWSA	Benutzerkennung für das Web Service API

> **ACHTUNG!**
> Wird ein schon einmal importierter Pubset erneut mit ACTUAL-JOIN=*FIRST importiert, wird nicht nur der Benutzerkatalog auf seine Standardbelegung zurückgesetzt, sondern auch alle auf ihm befindlichen Daten – außer Dateien unter $TSOS – werden gelöscht.

Die Benutzerkennungen des Systems, mit Ausnahme von TSOS und SERVICE, werden bei der Initialisierung gesperrt gesetzt und können mit dem Kommando UNLOCK-USER freigegeben werden.

Benutzer eintragen und verwalten

Für jeden Benutzer, der Zugriff zum System haben soll, muss die Systembetreuung einen Eintrag im Benutzerkatalog des Home-Pubsets und dem des zugewiesenen Standard-Pubsets erstellen. Für die LOGON-Validierung ist der Eintrag im Benutzerkatalog des Home-Pubsets ausschlaggebend. In Absprache mit dem Benutzer werden ihm, neben den Daten zu seiner Identifizierung, bestimmte Betriebsmittel und Berechtigungen zugeordnet:

Identifizierung:	Benutzerkennung, Kennwort, SPOOLOUT-Klasse, Versandanschrift
Betriebsmittel:	Standard-Pubset, Benutzeradressraum, Speicherplatz auf gemeinschaftlichen Platten, CPU-Zeit, Meldungssprache, Zuordnung einer Gruppensyntaxdatei
Berechtigungen:	Überschreitung des zugewiesenen Speicherplatzes, Nutzung bestimmter Task-Attribute, Ablaufpriorität, Testprivilegien, Nutzung von Hardware- und Linkage-AUDIT, Nutzung von Net-Storage

Die Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES dienen zum Erstellen bzw. Aktualisieren von Einträgen im Benutzerkatalog.

Mit dem Kommando SHOW-USER-ATTRIBUTES kann sich die Systembetreuung die Benutzereinträge ausgeben lassen und erhält auf diese Art und Weise einen Überblick über den Inhalt des Benutzerkataloges. Ferner kann die Systembetreuung mit den Kommandos LOCK-USER und REMOVE-USER Benutzerkennungen sperren bzw. löschen und mit UNLOCK-USER wieder entsperren.

Die System-Benutzerkennungen können nicht gelöscht werden (Ausnahme: die Benutzerkennung SERVICE bei Einsatz des Produkts SECOS, siehe das SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46]) .

6.3 Sicherungskonzept für Benutzerkataloge

- Sichern des aktuellen Benutzerkatalogs eines Pubsets
- Wiedereinspielen eines gesicherten Benutzerkatalogs
- Rekonstruktion der Datei SYSSRPM
  - Rekonstruktion durchführen
  - Logging-Datei

6.3.1 Sichern des aktuellen Benutzerkatalogs eines Pubsets

Zur Sicherung des Benutzerkataloges muss die Datei $TSOS.SYSSRPM.BACKUP mit dem Subsystem ARCHIVE (Anweisung SAVE) oder HSMS (Anweisung BACKUP-FILES) gesichert werden. Diese Sicherung erfolgt für den Benutzer im Wesentlichen wie die Sicherung jeder anderen Datei, wobei der Dateiname nur als Platzhalter fungiert. Die aktuellen Daten werden während des SAVE-Laufs von der Benutzerverwaltung an ARCHIVE/HSMS übergeben, die eigentliche .BACKUP-Datei behält ihre Platzhalterfunktion und bleibt leer.
Um die gesicherten Daten wirksam werden zu lassen, muss die .BACKUP-Datei vor IPL bzw. IMCAT von der Sicherung eingelesen (mit RESTORE/RESTORE-FILES) und der IPL bzw. IMCAT entsprechend parametrisiert werden.

Wird eine Differenzsicherung der Datei gefordert (Operation SAVE ...,CHANGED=YES), so erfolgt diese in jedem Fall, da die Datei nach der letzten Bearbeitung durch ARCHIVE von der sie verwaltenden Systemtask wieder geöffnet wurde.

Ein Export für die .BACKUP-Datei wird abgewiesen.

> ¡ Da bei der Rekonstruktion eine Manipulation des Inhalts des gesicherten Benutzerkataloges nicht festgestellt werden kann, muss die Datei von der Systembetreuung durch geeignete organisatorische Maßnahmen geschützt werden.

6.3.2 Wiedereinspielen eines gesicherten Benutzerkatalogs

Zum Wiedereinspielen des Benutzerkataloges muss die Datei $TSOS.SYSSRPM.BACKUP mit Hilfe des Subsystems ARCHIVE (Operation RESTORE) wiedereingespielt werden.
Sofern bei der Sicherung der Datei $TSOS.SYSSRPM.BACKUP diese umbenannt wurde, muss diese Umbenennung bei der Wiedereinspielung rückgängig gemacht werden. Da die Datei permanent existiert, muss bei der Operation RESTORE der Operand REPLACE auf einen der Werte YES, ALL oder ALLP gesetzt werden.

> ¡ Es ist zulässig, einen gesicherten Benutzerkatalog auf einem anderen Pubset als demjenigen, von dem gesichert wurde, einzuspielen. Zu beachten ist dabei, dass Attribute, wie z.B. der Standard-Pubset, manuell aktualisiert werden müssen.

6.3.3 Rekonstruktion der Datei SYSSRPM

Eine Rekonstruktion betrifft nicht nur den Benutzerkatalog, sondern auch den Dateikatalog TSOSCAT, denn der Dateikatalog TSOSCAT führt eine eigene Liste von Benutzerkennungen, die der Abbildung Benutzerkennung zu PBN (Primary Block Number, kennzeichnet den Eigentümer der Datei) dient. Die Rekonstruktion muss die Konsistenz von Benutzerkatalog und Dateikatalog wahren. Die Benutzerstruktur des vorhandenen Benutzerkataloges SYSSRPM wird nicht in den Rekonstruktionsprozess miteinbezogen.

> ¡ Im Idealfall sollte ein Pubset mit seiner eigenen Sicherung rekonstruiert werden. Ist dies nicht der Fall, können Inkonsistenzen zwischen den Systemumgebungen zu den Zeitpunkten der Sicherung und Rekonstruktion bestehen.

Ziel ist es, die beiden Benutzerstrukturen von .BACKUP und TSOSCAT zu vereinigen und dabei mögliche Inkonsistenzen zu beseitigen.

6.3.3.1 Rekonstruktion durchführen

Die Rekonstruktion des Benutzerkataloges kann beim Startup (im Falle des Home-Pubsets) oder mit dem Kommando IMPORT-PUBSET veranlasst werden. Voraussetzung ist die erfolgreiche Wiedereinspielung eines Benutzerkataloges in die Datei .BACKUP vor dem letzten Shutdown bzw. Export.

Wird ein Pubset neu eingerichtet, kann die Rekonstruktion der Datei SYSSRPM auch durch SIR erfolgen.

Systemparameter RECONUC

Die Rekonstruktion wird durch den Systemparameter RECONUC gesteuert. Er kann über den Startup-Parameterservice eingestellt und geändert werden. Beim DIALOG-Startup besteht eine zusätzliche Möglichkeit, RECONUC zu ändern: Über die Meldung `NSI6010` werden der voreingestellte Wert für RECONUC ausgegeben und Änderungswünsche abgefragt.

Zu beachten ist, dass die Systemparameter RECONUC und STUPTYPE in Zusammenhang stehen. Sofern STUPTYPE=J oder T gewählt wurde, wird zuerst der First-Startup (mit/ohne Rücksetzen des Benutzerkatalogs) ausgeführt und anschließend der Wert für RECONUC ausgewertet. Die Meldung `NSI6220` informiert über einen ungültigen Wert für RECONUC oder STUPTYPE und das Setzen auf einen Standardwert, der in der Meldung angegeben wird.

Für RECONUC können die Werte N, B, T, A und R angegeben werden. N bedeutet keine Rekonstruktion.

Beispiel

A: Menge aller Benutzer, die in der Datei .BACKUP existieren, jedoch nicht im TSOSCAT

B: Menge aller Benutzer, die sowohl in der Datei .BACKUP also auch im TSOSCAT existieren

C: Menge aller Benutzer, die im TSOSCAT existieren, jedoch nicht in der Datei .BACKUP

Im Idealfall sollten die Mengen A und C leer sein, da sonst bei der Rekonstruktion entweder Benutzerattribute oder Dateien verloren gehen können.

Die folgende Liste zeigt, welcher Wert für welche Rekonstruktionsanforderung angegeben werden muss und wie sich das ggf. auf die Beispielmengen A, B und C auswirkt. In Klammern steht der entsprechende Wert beim Kommando IMPORT-PUBSET.

- Rekonstruktion mittels mittels BACKUP: RECONUC=B (SCOPE BACKUP)

  Bezogen auf die Mengen A, B und C bedeutet das:

  - A: Neueinrichtung mit den gesicherten Attributen.
  - B: Update der vorhandenen mit den gesicherten Benutzerattributen.
  - C: Löschen ihrer Dateien und Jobvariablen.

  Dieser Modus ist bei regelmäßiger Sicherung zu empfehlen.

- Rekonstruktion mittels TSOSCAT: RECONUC=T (SCOPE TSOSCAT)

  Bezogen auf die Mengen A, B und C bedeutet das:

  - A: keine Übernahme in die rekonstruierte Benutzerstruktur. Dadurch kann die Privilegienverteilung oder die Gruppenstruktur auf dem betreffenden Pubset dahingehend gestört werden, dass z.B. ein Benutzer, der als einziger in der .BACKUP-Datei ein bestimmtes Privileg besaß, nicht in den neuen Benutzerkatalog übernommen wird oder keine Übernahme eines Gruppenverwalters erfolgt.
  - B: Update der vorhandenen mit den gesicherten Benutzerattributen.
  - C: Einrichtung mit Standard-Attributen und den Erhalt der Dateien, Jobvariablen und Guards.

  Dieser Modus ist dann zu empfehlen, wenn der Erhalt von Dateien im Vordergrund steht und Benutzerkennungen, soweit sie zum Zeitpunkt der Sicherung bereits existierten, rekonstruiert werden sollen.
- Rekonstruktion mittels BACKUP und TSOSCAT: RECONUC=A (SCOPE ALL)

  Analog zu Scope BACKUP wird die Benutzerstruktur zum Zeitpunkt der Sicherung wiederhergestellt.

  Analog zu Scope TSOSCAT bleiben Dateien von Benutzerkennungen, die nach erstellen der Sicherung eingerichtet wurden, durch wiedereinrichten der Benutzerkennungen mit Standardattributen erhalten. Im Falle zweier großer, disjunkter Benutzerstrukturen kann dies das Fassungsvermögen des TSOSCAT mit maximal 8189 Benutzerkennungen sprengen. In diesem Fall wird der Import sofort nach Festellung dieser Tatsache abgebrochen, und der Import kann mit einem der Modi BACKUP oder TSOSCAT wiederholt werden.
- Rücksetzen des Benutzerkataloges auf Stand von TSOSCAT: RECONUC=R (RESET)

  Diese Funktion erlaubt der Systembetreuung die formale Wiederherstellung des Benutzerkataloges unter Erhalt der Dateien. Die inhaltliche Wiederherstellung (also die Wiederherstellung der Benutzerattribute) muss in einem zweiten Schritt mit Hilfe der Rekonstruktion erfolgen.

  Als Alternative im Fehlerfall bietet sich nur noch der First-Start an, bei dem alle Kennungen außer denen der Systemverwaltung und alle Dateien, die nicht der Kennung TSOS gehören, verlorengehen.
  Wurde die SYSSRPM durch einen Systemfehler zerstört, sollte man die Strategie der vollständigen Pubset-Rekonstruktion wählen, denn dieser Systemfehler könnte auch andere Dateien zerstört oder beschädigt haben.

Den ordnungsgemäßen Verlauf der Rekonstruktion kann der Operator anhand von zwei Meldungen verfolgen, von denen die erste zu Beginn die Rekonstruktionsbasis über die Katalogkennung und den Sicherungszeitpunkt dokumentiert (`SRM2017` bei Rekonstruktion mit (*BY-BACKUP) bzw. `SRM2018` bei Rekonstruktion ohne .BACKUP-Datei (*RESET)) und die zweite am Ende die Anzahl der rekonstruierten Benutzerkennungen angibt (`SRM2019` bei Rekonstruktion mit (*BY-BACKUP) bzw. `SRM2020` bei Rekonstruktion ohne .BACKUP-Datei (*RESET)).

Der Systembetreuung wird darüber hinaus eine Logging-Datei zur Verfügung gestellt, siehe "Logging-Datei".

Rekonstruktion eines defekten Benutzerkatalogs

Soll die Rekonstruktion der Wiederherstellung eines defekten Benutzerkatalogs dienen, der sich aber als Folge des Defekts nicht importieren lässt, muss zunächst eine rudimentäre Korrektur über die Funktion RESET erfolgen, bei der auf Basis der in TSOSCAT bekannten Benutzerkennungen ein neuer Benutzerkatalog aufgebaut wird. Die einzelnen Benutzerkennungen erhalten Standard-Attribute und sind alle bis auf TSOS und SERVICE gesperrt. Im Gegensatz zum First-Start bleiben auf diese Weise alle Dateien erhalten und der vollständige Neuaufbau des Pubsets kann vermieden werden.
Sofern kein Zip-Import gefordert ist, wird der defekte Benutzerkatalog nicht gelöscht, sondern für eine spätere Diagnose des Problems, das zum Reset geführt hat, unter dem Namen :catid:$TSOS.SYS.SRPM.RECON.DIAG.<datum.uhrzeit> gesichert.

> **ACHTUNG!**
> Bei der Weitergabe eines defekten Benutzerkatalogs an Dritte ist zu bedenken, dass eine unberechtigte Rekonstruktion der Benutzerdaten nicht ausgeschlossen werden kann.

Die Funktion RESET kann neben der Angabe als Rekonstruktions-Typ (siehe oben) auch als Antwort auf die Meldung `SRM2012` angefordert werden. Diese Meldung wird bei Fehlern in der Benutzerverwaltung ausgegeben, Fehler in anderen Komponenten wie z.B. der Gruppenverwaltung, führen zum Abbruch der IMPORT-PUBSET-Verarbeitung.

Auswirkung auf die Attribute

- Standard-Pubset (Default-Pubset)

  Das Attribut DEFAULT-PUBSET aller System-Benutzerkennungen erhält die Catid des rekonstruierten Pubsets. Das Attribut DEFAULT-PUBSET der übrigen Benutzerkennungen bleibt unberührt. Wurde die Sicherung einer inkompatiblen Pubset-Konfiguration rekonstruiert, muss die Systemverwaltung in eigener Verantwortung die lokal gültigen Default-Pubsets zuweisen.

- Logon-Kennwörter

  Wurde der Benutzerkatalog eines Systems ohne Kennwortverschlüsselung (Systemparameter ENCRYPT) gesichert und auf einem System mit Kennwortverschlüsselung rekonstruiert, werden alle Logon-Kennwörter verschlüsselt.
  Im umgekehrten Fall ist eine entsprechende Maßnahme unmöglich!

  Durch die Rekonstruktion erhalten alle Benutzerkennungen ihre Logon-Kennwörter zum Zeitpunkt der Sicherung zurück. Es liegt in der Verantwortung jedes einzelnen Benutzers, sich an dieses Kennwort zu erinnern. Dies gilt insbesondere für TSOS oder eine andere Benutzerkennung mit dem Privileg USER-ADMINISTRATION.

- SECOS-Attribute

  Wurde der Benutzerkatalog eines Systems mit installiertem SECOS gesichert und auf einem System ohne SECOS rekonstruiert, werden die Einstellungen der Logon-Parameter und der Privilegien aller Benutzerkennungen auf die Standard-Einstellungen zurückgesetzt. Auf diese Weise wird verhindert, dass einmal mit SECOS vorgenommene Einstellungen den Betrieb ohne SECOS behindern, ohne dass sie zurückgenommen werden können.

  Wird bei einer Rekonstruktion in einem System, in dem das Subsystem SECOS nicht verfügbar ist, eine . BACKUP-Datei verwendet (also RECONUC=B, T oder A), die in einem System gesichert wurde, in dem das Produkt SECOS verfügbar war, so werden alle Privilegien auf die Standard-Einstellungen wie bei First-Start zurückgesetzt. Sollte in diesem Fall die .BACKUP-Datei eine Privilegienverteilung enthalten, die nur in einer höheren BS2000-Version bekannt ist, (dies ist nach Versionwechseln möglich), werden dabei auch die in der aktuellen Version unbekannten Privilegien zurückgesetzt.

  Ist sowohl auf dem System des gesicherten als auch des rekonstruierten Benutzerkataloges SECOS im Einsatz, können durch die verstrichene Zeit zwischen Sicherung und Rekonstruktion Kennwort-Verfallsdaten überschritten worden sein. Die drohende Sperre der Benutzerkennung wird verhindert, indem die zum Zeitpunkt der Sicherung vorhandene Restlaufzeit rekonstruiert wird.

- SM-Pubset-Attribute

  Wurde der Benutzerkatalog eines SF- bzw. SM-Pubsets gesichert und auf einem SM- bzw. SF-Pubset rekonstruiert, so erhalten die SM-Pubset-spezifischen Attribute den Standardwert (bei einer Umwandlung von SF- nach SM-Pubset) bzw. sie werden gelöscht (bei einer Umwandlung von SM- nach SF-Pubsets).

6.3.3.2 Logging-Datei

Anhand der Logging-Datei kann die Systembetreuung das Ergebnis der Rekonstruktion überprüfen. Diese Datei wird unter der Kennung TSOS angelegt und heißt SYS.SRPM.RECON.LOG.<datum.uhrzeit>. Sie enthält folgende Informationen:

- Pubsets des gesicherten (BACKUP) bzw. zu rekonstruierenden (SYSSRPM) Benutzerkataloges:

PUBSET	Katalogkennung
TYPE	SM- oder SF-Pubset
DATE	Datum der Sicherung bzw. Rekonstruktion
TIME	Uhrzeit der Sicherung bzw. Rekonstruktion
ENCRYPT	NO/YES: Logon-Kennwort Verschlüsselung
SECOS	NO/YES: Produkt SECOS im Einsatz

- Benutzerkennung:

USERID IN SYSSRPM	
USERID	Alle bearbeiteten Benutzerkennungen
PRESENT	NO/YES: Eintrag im rekonstruierten Benutzerkatalog
USERID IN BACKUP	
PRESENT	NO/YES: Eintrag in Backup-Datei
ACTION	ADDED/REMOVED: Neuaufnahme/Löschung
USERID IN TSOSCAT	
PRESENT	NO/YES: Eintrag in TSOSCAT
ACTION	ADDED/REMOVED: Neuaufnahme/Löschung
DELETED	
FILES	Anzahl gelöschter Dateien bei SCOPE=BACKUP
JV	Anzahl gelöschter JVs bei SCOPE=BACKUP

Abhängig vom eingegebenen Kommando IMPORT-PUBSET PUBSET=..., RECONSTRUCT-USERCAT=*BY-BACKUP(SCOPE=*ALL/*BACKUP/*TSOSCAT) werden die Benutzerkennungen bei der Rekonstruktion unterschiedlich behandelt.

Beispiel

```
 USERID IN SYSSRPM | USERID IN BACKUP  | USERID IN TSOSCAT |      DELETED      |
 USERID  | PRESENT | PRESENT | ACTION  | PRESENT | ACTION  |  FILES  |   JV    |
---------+---------+---------+---------+---------+---------+---------+---------+
USERID1  |   YES   |   YES   |         |   YES   |         |         |         |
USERID2  |   YES   |   YES   |         |   NO    | ADDED   |         |         |
USERID3  |   YES   |   NO    | ADDED   |   YES   |         |         |         |
USERID4  |   NO    |   YES   | REMOVED |   NO    |         |         |         |
USERID5  |   NO    |   NO    |         |   YES   | REMOVED |       4 |       3 |
---------+---------+---------+---------+---------+---------+---------+---------+
```

- SCOPE=*ALL/*BACKUP/*TSOSCAT

  USERID1 wurde in der .BACKUP-Datei gesichert und hat bei der Rekonstruktion einen Eintrag im Dateikatalog TSOSCAT.
  USERID1 wird im Benutzerkatalog SYSSRPM mit den gesicherten Benutzerattributen rekonstruiert und behält ihren Dateibestand.

- SCOPE=*ALL/*BACKUP

  USERID2 wurde in der .BACKUP-Datei gesichert, hat aber bei der Rekonstruktion keinen Eintrag im Dateikatalog TSOSCAT.
  USERID2 wird im Benutzerkatalog SYSSRPM mit den gesicherten Benutzerattributen rekonstruiert und erhält einen leeren Eintrag im Dateikatalog.

- SCOPE=*ALL/*TSOSCAT

  USERID3 wurde nicht in der .BACKUP-Datei gesichert, hat aber bei der Rekonstruktion einen Eintrag im Dateikatalog TSOSCAT.
  USERID3 wird im Benutzerkatalog SYSSRPM mit Standardattributen neu eingerichtet und erhält ihren Dateibestand.

- SCOPE=*TSOSCAT

  USERID4 wurde in der .BACKUP-Datei gesichert, hat aber bei der Rekonstruktion keinen Eintrag im Dateikatalog TSOSCAT.
  USERID4 wird nicht in den Benutzerkatalog SYSSRPM übernommen.

- SCOPE=*BACKUP

  USERID5 wurde nicht in der .BACKUP-Datei gesichert, hat aber bei der Rekonstruktion einen Eintrag im Dateikatalog TSOSCAT.
  USERID5 wird nicht in den Benutzerkatalog SYSSRPM übernommen, ihr Eintrag im Dateikatalog wird zusammen mit ihren Dateien und JVs gelöscht.

- Zusammenfassung:

  LOGON PASSWORDS ENCRYPTED: NO/YES:
  Die Logon-Kennwörter wurden verschlüsselt

  EXPIRATION DATES UPDATED: NO/YES:
  Die Logon-Kennwort Verfallsdaten wurden angepasst

  SECOS ATTRIBUTES RESET: NO/YES:
  Die SECOS-Attribute wurden auf Standardwerte gesetzt

  USERIDS RECONSTRUCTED:
  Anzahl der Benutzerkennungen im rekonstruierten Benutzerkatalog

  USERIDS ADDED TO BACKUP:
  Anzahl der Benutzerkennungen, die im neuen Benutzerkatalog gegenüber dem Stand der .BACKUP-Datei hinzugefügt wurden; SCOPE=*ALL/*TSOSCAT

  USERIDS REMOVED FROM BACKUP:
  Anzahl der Benutzerkennungen, die im neuen Benutzerkatalog gegenüber dem Stand der .BACKUP-Datei gelöscht wurden; SCOPE=*TSOSCAT

  USERIDS ADDED TO TSOSCAT:
  Anzahl der Benutzerkennungen in TSOSCAT neu eingerichtet;
  SCOPE=*ALL/*BACKUP

  USERIDS REMOVED FROM TSOSCAT:
  Anzahl der Benutzerkennungen in TSOSCAT gelöscht; SCOPE=*BACKUP

Verhalten im Fehlerfall

Im Falle eines Systemfehlers, der zum Abbruch der Rekonstruktion und damit der Import-Verarbeitung führt, hängt die zukünftige Benutzerstruktur vom Stand der Rekonstruktion beim Abbruch ab. Dieser Stand lässt sich anhand von Konsolmeldungen bestimmen:

- vor Meldung `SRM2017`
  Die Rekonstruktion hat noch nicht begonnen, es kann ein erneuter Pubset-Import mit oder ohne Rekonstruktion gestartet werden.
- zwischen Meldung `SRM2017` und `SRM2019`
  Die Rekonstruktion ist in vollem Gange, es kann nur ein erneuter Pubset-Import ohne Rekonstruktion gestartet werden. Die Backup-Datei muss anschließend ggf. wieder eingelesen werden. Die Benutzerstruktur ist auf dem alten Stand.
- nach Meldung `SRM2019`
  Die Rekonstruktion ist so weit fortgeschritten, dass ein Rückzug nicht mehr möglich ist. Es kann ein erneuter Pubset-Import ohne Rekonstruktion gestartet werden. Die Benutzerstruktur ist auf dem neuen Stand.

6.4 Benutzerkatalog und SMS-Konzept

Die Benutzerkataloge von SF- und SM-Pubsets unterscheiden sich im Prinzip nicht.

Bei der Umwandlung von SF-Pubsets in SM-Pubsets werden die für SF-Pubsets relevanten Kontingente TEMP-SPACE-LIMIT und PUBLIC-SPACE-LIMIT auf SM-Pubsets wie folgt interpretiert:

SF-Pubset-Attribut		**SM-Pubset-Attribut**
TEMP-SPACE-LIMIT	wird zu	TEMP-SPACE-LIMITS/TOTAL-SPACE
PUBLIC-SPACE-LIMIT	wird zu	PERM-SPACE-LIMITS/S0-LEVEL-SPACE

Die zusätzlichen Kontingente für SM-Pubsets sind nur über das Kommando MODIFY-USER-PUBSET-ATTRIBUTES möglich. Die neuen Kontingente sind über den Parameter INFORMATION=PUBSET-ATTRIBUTES des Kommandos SHOW-USER-ATTRIBUTES abfragbar.

6.5 Bulletin-Datei (Logon-Informationsdatei)

Die Systembetreuung kann eine Informationsdatei einrichten, die unter dem Namen $TSOS.BULLETIN abgespeichert und automatisch an alle Benutzer nach dem Logon ausgegeben wird. $TSOS.BULLETIN muss mehrbenutzbar sein und sollte mit einem Schreib-Kennwort versehen sein oder nur Lesezugriff erlauben.

Aufbau und Inhalt der Bulletin-Datei

Es empfiehlt sich, die Bulletin-Datei wie folgt aufzubauen:

1. Allgemeine Informationen für alle Benutzer
2. Spezielle Informationen für bestimmte Zielgruppen oder einzelne Benutzer
3. Abfragbare Zusatzinformationen für alle Benutzer

Neben Nachrichten, die an alle Benutzer adressiert sind, können spezielle Informationen an bestimmte Zielgruppen oder einzelne Benutzer gesendet werden. Dem Dialogteilnehmer wird dies durch Ausgabe der folgenden Meldung angeboten:
`CONTINUE userid (Y,N)?`

Bei Eingabe von `N` wird die Ausgabe der Bulletin-Datei abgebrochen.

Die Bulletin-Datei ist so aufzubauen, dass sich der Dialogteilnehmer nach der ersten aktuellen Nachricht wahlweise noch weitere Nachrichten (Folgemitteilungen) ausgeben lassen kann. Dazu ist die folgende Abfrage vor jeder Folgemitteilung einzubauen:
`CONTINUE (Y,N)?` bzw. `CONTINUE userid (Y,N)?`

> *Hinweis zum Batch-Betrieb*
> Nach SYSLST werden alle benutzerspezifischen Nachrichten ausgegeben.
> Nach SYSOUT wird nur die erste Nachricht der Bulletin-Datei ausgegeben.
> Die aktuellsten Nachrichten sollten deshalb immer am Anfang der Bulletin-Datei stehen.

Beispiel

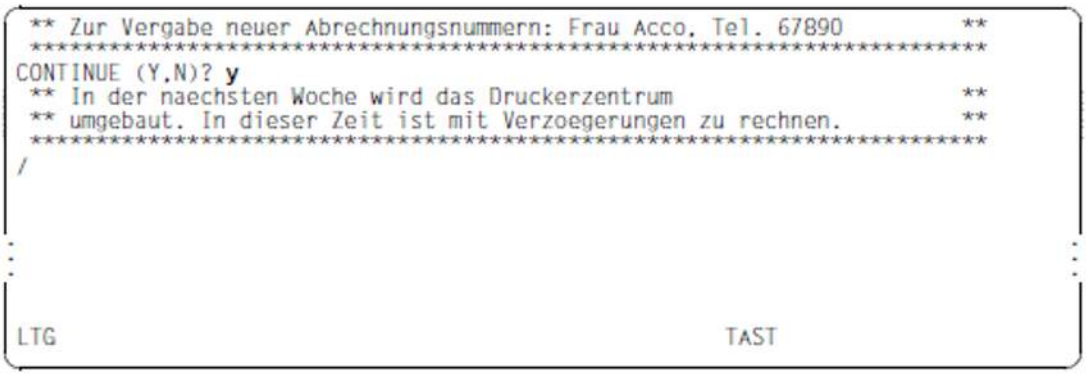
Im Bulletin soll eine allgemeine Information an alle Benutzer, abfragbare Zusatzinformationen für den Benutzer EINERLEI und ALLERLEI sowie eine abfragbare allgemeine Information an alle Benutzer enthalten:

```
/START-EDT
  ************************************************************************
  ** Willkommen am Server X/YZ !                                        **
  ** Normaler OPEN SESSION-Betrieb taeglich von 07:00 bis 21:00 Uhr.     **
  **                                                                    **
  ** Service-Nummern:  Bandansage        Tel. 12345                     **
  **                   Listenausgabe     Tel. 34512                     **
  **                   Leitstelle        Tel. 51234                     **
  ************************************************************************
  ** ACHTUNG !                                                          **
  ** Am Freitag wird der Server wegen Wartungsarbeiten                  **
  ** bereits um 17:00 runtergefahren.                                   **
  ************************************************************************
**STOP**$EINERLEI
  ** ACHTUNG !  DRINGEND !                                              **
  ** Geben Sie die von Ihnen reservierten Datentraeger MEINER1 und      **
  ** MEINER4 frei. Fuer Rueckfragen: Leitstelle, Tel. 89067             **
  ************************************************************************
**STOP**$ALLERLEI
  ** ACHTUNG !                                                          **
  ** Ihre Privatplatte PRIV03 steht momentan nicht zur Verfuegung.      **
  ** Bitte setzen Sie sich mit uns in Verbindung (Tel. 51234).          **
  ************************************************************************
**STOP**$EINERLEI
  ** Ihre Abrechnungsnummer M0815 wird ungueltig.                       **
  ** Zur Vergabe neuer Abrechnungsnummern: Frau Acco, Tel. 67890        **
  ************************************************************************
**STOP**
  ** In der naechsten Woche wird das Druckerzentrum                     **
  ** umgebaut. In dieser Zeit ist mit Verzoegerungen zu rechnen.        **
  ************************************************************************
W'BULLETIN'
HALT
```

Ausgabe 1. Bildschirm

```
% JMS0160 INSTALLATION ' SU700-60', BS2000 VERSION 'V190', HOST 'DOYZZEOX': PLE
ASE ENTER '/SET-LOGON-PARAMETERS' OR '?'
/.test logon einerlei,m0815,'denkste#'
% JMS0066 JOB 'TEST' ACCEPTED ON <date> AT <time>, TSN = 12CF
 ************************************************************************
 ** Willkommen am Server X/YZ !                                        **
 ** Normaler OPEN SESSION-Betrieb taeglich von 07:00 bis 21:00 Uhr.     **
 **                                                                    **
 ** Service-Nummern:  Bandansage        Tel. 12345                     **
 **                   Listenausgabe     Tel. 34512                     **
 **                   Leitstelle        Tel. 51234                     **
 ************************************************************************
 ** ACHTUNG !                                                          **
 ** Am Freitag wird der Server wegen Wartungsarbeiten                  **
 ** bereits um 17:00 runtergefahren.                                   **
 ************************************************************************
CONTINUE EINERLEI (Y,N)? y
 ** Achtung !  DRINGEND !                                              **
 ** Geben Sie die von Ihnen reservierten Datentraeger MEINER1 und      **
 ** MEINER4 frei. Fuer Rueckfragen: Leitstelle, Tel. 89067             **
 ************************************************************************
CONTINUE EINERLEI (Y,N)? y
 ** Ihre Abrechnungsnummer M0815 wird ungueltig.                       **
%PLEASE ACKNOwWLEDGE
LTG                                                  TAST
```

Ausgabe Folge-Bildschirm



7 POSIX-Benutzerverwaltung

Dieses Kapitel beschreibt die Schnittstellen zur Verwaltung der POSIX-Benutzerattribute einer BS2000-Benutzerkennung. Diese Schnittstellen sind Bestandteil des Bausteins SRPM, der im Softwareprodukt SECOS und in BS2000 implementiert ist. Das Softwareprodukt SECOS muss jedoch nicht installiert sein, um mit POSIX arbeiten zu können.

Jeder BS2000-Benutzer ist gleichzeitig auch POSIX-Benutzer. Außer einer BS2000-Benutzerkennung mit gültigen individuellen POSIX-Benutzerattributen sind keine weiteren Bedingungen zu erfüllen, um Zugang zu POSIX und seinen Schnittstellen zu erhalten.

Näheres zu SRPM finden Sie im Abschnitt „Beschreibung der Privilegien" und im SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46].

Die POSIX-Funktionalität in BS2000 wird ausführlich in den Handbüchern „POSIX-Kommandos“ [38] und „POSIX-Grundlagen“ [39] beschrieben.

Was ist POSIX ?

Unter POSIX (Portable Open System Interface for UNIX) versteht man eine Reihe von Standards auf UNIX-Basis. Diese Standards gewährleisten die Kompatibilität und Interoperabilität von Anwendungen in einem heterogenen Netzwerk. Ein heterogenes Netzwerk besteht aus Servern und Produkten von verschiedenen Herstellern sowie aus System- und Benutzersoftware von verschiedenen Software-Anbietern.

Der POSIX-Standard wurde vom Institute of Electrical and Electronics Engineers (IEEE) 1989 als nationaler amerikanischer Standard definiert. Anschließend wurde er vom X/OPEN-Konsortium und 1990 als internationaler Standard verabschiedet (X/OPEN Portability Guide IV, XPG4).

Dem Benutzer stehen die Bibliotheksfunktionen des XPG4-Standards über eine C-Bibliothek und eine definierte Menge von Kommandos über eine Shell (POSIX-Shell) zur Verfügung.

Mit POSIX lassen sich Anwendungsprogramme leicht portieren – unabhängig vom ausführenden Betriebssystem. Deshalb können XPG4-konforme Programme nach einer Neuübersetzung auch in BS2000 ablaufen.

Die POSIX-Programmschnittstellen werden parallel zu den BS2000-Schnittstellen angeboten. Die gemischte Nutzung von BS2000- und POSIX-Programmschnittstellen in einem Programm ist möglich.

Einige Software-Komponenten und -Produkte von BS2000 wurden funktionell erweitert, um POSIX-Dateien bearbeiten zu können. Zum Beispiel können mit SPOOL auch POSIX-Dateien ausgedruckt, sowie mit HSMS POSIX-Dateien, -Dateiverzeichnisse und -Dateisysteme gesichert und rekonstruiert werden.

Kommando	Bedeutung
ADD-POSIX-USER	Festlegen der POSIX-Benutzerattribute
ADD-USER	Erstellen eines Benutzereintrags im Benutzerkatalog [1]
MODIFY-LOGON-PROTECTION	Ändern von Schutzattributen [2]
MODIFY-POSIX-USER-ATTRIBUTES	Ändern der POSIX-Benutzerattribute einer BS2000-Benutzerkennung
MODIFY-POSIX-USER-DEFAULTS	Ändern der POSIX-Standardattribute eines Pubsets
MODIFY-USER-ATTRIBUTES	Ändern des Katalogeintrags eines Benutzers [1]

SET-LOGON-PROTECTION	Vereinbaren von Schutzattribute [2]
SHOW-LOGON-PROTECTION	Anzeigen von Schutzattributen [2]
SHOW-POSIX-STATUS	Anzeigen des POSIX-Status
SHOW-POSIX-USER-ATTRIBUTES	Anzeigen der POSIX-Benutzerattribute einer BS2000-Benutzerkennung
SHOW-POSIX-USER-DEFAULTS	Anzeigen der POSIX-Standardattribute eines Pubsets
SHOW-USER-ATTRIBUTES	Ausgabe von Informationen über die Einträge im Benutzerkatalog [1]
START-POSIX-SHELL	POSIX-Shell zur Verfügung stellen
Makro	**Bedeutung**
SRMUINF	Lesen der Daten aus dem Benutzerkatalog und Übertragen in einen vorher festgelegten Bereich

Tabelle 14: BS2000-Schnittstellen zur POSIX-Benutzerverwaltung

[1] Kommandos zum Verwalten der Abrechnungsnummer für den Zugang über einen fernen Server.

[2] Kommandos zum Verwalten der Zugangsberechtigung über einen fernen Server:
Wenn das Softwareprodukt SECOS eingesetzt wird, kann für bereits existierende BS2000-Benutzerkennungen festgelegt werden, ob sich der Benutzer eines fernen Servers mit dem UNIX-Kommando rlogin Zugang zum System (POSIX) verschaffen darf. Dazu steht der Operand POSIX-RLOGIN-ACCESS=*YES(PASSWORD-CHECK=*YES/*NO) bzw. POSIX-RLOGIN-ACCESS=*NO bei den Kommandos SET- und MODIFY-LOGON-PROTECTION zur Verfügung.

Diese und weitere POSIX-Kommandos sind in den Handbüchern „SECOS - Zugangs- und Zugriffskontrolle“ [46] und „Kommandos“ [27] beschrieben.

POSIX-Benutzerattribute

Siehe das Kapitel „POSIX-Benutzer verwalten“ im Handbuch „POSIX-Grundlagen“ [39].

Privilegierung zur Verwaltung der POSIX-Benutzerattribute

Für POSIX gibt es das Privileg POSIX-ADMINISTRATION. Inhaber dieses Privilegs werden POSIX-Verwalter genannt. Sie haben folgende Aufgaben und Rechte:

- Verwaltung der POSIX-Benutzerattribute aller BS2000-Benutzerkennungen auf allen Pubsets
- Verwaltung der POSIX-Standardattribute auf allen Pubsets
- Aufruf von privilegierten POSIX-Funktionen

Das Privileg POSIX-ADMINISTRATION ist standardmäßig an die System-Benutzerkennung SYSROOT geknüpft. Dieses Privileg kann SYSROOT nicht entzogen werden.
Der Sicherheitsbeauftragte (Privileg SECURITY-ADMINISTRATION) kann dieses Privileg auch anderen BS2000-Benutzerkennungen verleihen und entziehen.

SYSROOT ist das POSIX-Gegenstück zur Systemverwalterkennung *root* in UNIX-Systemen. SYSROOT wird beim First-Start des BS2000-Systems eingerichtet und erhält automatisch die Benutzernummer 0. SYSROOT kann keine andere Benutzernummer zugewiesen werden.

Inhaber des Privilegs USER-ADMINISTRATION erhalten zusätzlich die Berechtigung, die POSIX-Benutzerattribute und die Standardwerte für die POSIX-Benutzerattribute zu verwalten. Sie sind diesbezüglich dem POSIX-Verwalter gleichgestellt.

Die Berechtigung des Gruppenverwalters der Gruppe *UNIVERSAL wird auf die POSIX-Benutzerattribute ausgedehnt. Er ist bei der Verwaltung der POSIX-Benutzerattribute auf dem von ihm verwalteten Pubset den Inhabern des Privilegs USER-ADMINISTRATION gleichgestellt. Deshalb gelten für ihn nicht die im Folgenden aufgeführten Einschränkungen für Gruppenverwalter seiner Hierarchie.

Gruppenverwalter dürfen ebenfalls die POSIX-Benutzerattribute verwalten. Allerdings gelten für sie folgende Einschränkungen:

- Sie können nicht die POSIX-Standardattribute verwalten.
- Die Art der POSIX-Benutzerattribute, die ihrer Verwaltung unterstellt sind, hängt von ihrer Autorisierung ab (ADM-AUTHORITY).
- Der Wertebereich der POSIX-Benutzerattribute ist für sie eingeschränkt.
- Sie können nur die Gruppen- und Untergruppenmitglieder verwalten, die ihnen unterstellt sind.

Für weitere Informationen zum Thema Privilegierung siehe Abschnitt „Privilegien".

Benutzernummer einer BS2000-Benutzerkennung zuordnen

Siehe das Kapitel „POSIX-Benutzer verwalten“ im Handbuch „POSIX-Grundlagen“ [39].

BS2000- und POSIX-Gruppen verwalten

Siehe das Kapitel „POSIX-Benutzer verwalten“ im Handbuch „POSIX-Grundlagen“ [39].

Neue POSIX-Benutzer eintragen

Siehe das Kapitel „POSIX-Benutzer verwalten“ im Handbuch „POSIX-Grundlagen“ [39].

POSIX-Benutzerattribute im POSIX-Dateisystem abbilden

Das POSIX-Benutzerattribut „Benutzernummer“ ist eng mit dem POSIX-Dateisystem verbunden: Die Benutzernummer dokumentiert, wer der Eigentümer einer Datei ist. Anders als in BS2000 kann der Root-Berechtigte einer Datei oder einem Dateiverzeichnis auf einfache Weise einen neuen Eigentümer zuweisen (POSIX-Kommando *chown*).

Benutzerinformationen per Programm lesen

Siehe das Kapitel „POSIX-Benutzer verwalten“ im Handbuch „POSIX-Grundlagen“ [39].

8 Dateiverwaltung

Dateien werden in Dateikatalogen verwaltet. Die Dateikataloge selber werden über das Katalogverwaltungssystems CMS verwaltet.

8.1 Dateikatalog

Der Dateikatalog eines SF- oder eines SM-Pubsets dient als Behälter für die Katalogeinträge der dem Pubset zugehörigen Dateien auf gemeinschaftlichen Plattenspeichern und Net-Storage, den Jobvariablen, sowie auch für die Katalogeinträge von Privatplattendateien und Banddateien. Der Dateikatalog wird implizit beim Einrichten des Pubsets angelegt. Er kann von der Systembetreuung erweitert werden.

Sämtliche Benutzer- und Systemdateien sowie Jobvariablen haben einen Eintrag im Dateikatalog. Die Dateien und Jobvariablen werden über einen eindeutigen Namen identifiziert.

Die **Namensstruktur** besteht aus:

:catid:$userid.name

catid	Der jeweilige Pubset wird über die Katalogkennung (catid) adressiert. Sie ist max. vier Zeichen lang. (siehe auch Abschnitt „Pubset-Eigenschaften").
userid	Die Benutzerkennung (userid) ist maximal 8 Zeichen lang.
name	Der Name für die Datei bzw. Jobvariable darf maximal 41 Zeichen lang sein. Bei Verwendung einer mehrstelligen Katalogkennung und maximaler Länge der Benutzerkennung reduziert sich diese Zahl entsprechend.

catid, *userid* und *name* dürfen insgesamt nicht länger als 54 Zeichen sein.

Kommando	**Bedeutung**
CHECK-IMPORT-DISK-FILE	Importieren von Dateien / Dateigenerationen überprüfen
CREATE-FILE	Namen und Merkmale einer neuen Datei vereinbaren
CREATE-FILE-GENERATION	Dateigeneration einer Dateigenerationsgruppe erzeugen
CREATE-FILE-GROUP	Dateigenerationsgruppe erzeugen
DELETE-FILE	Datei löschen
DELETE-FILE-GENERATION	Dateigeneration einer Dateigenerationsgruppe löschen
DELETE-FILE-GROUP	Dateigenerationsgruppe mit zugehörigen Dateigenerationen löschen
EXPORT-FILE	Katalogeintrag für Dateien auf privaten Datenträgern oder Net-Storage-Volumes löschen
EXPORT-NODE-FILE	Katalogeintrag von Node-Files löschen (Node-Files exportieren)
IMPORT-FILE	Katalogeintrag für Dateien auf privaten Datenträgern oder Net-Storage-Volumes erstellen

IMPORT-NODE-FILE	Katalogeintrag für Node-Files erstellen (Node-Files importieren)
MODIFY-FILE-ATTRIBUTES	Merkmale einer Datei ändern
MODIFY-FILE-GENERATION-SUPPORT	Merkmale einer Dateigeneration ändern
MODIFY-FILE-GROUP-ATTRIBUTES	Merkmale einer Dateigenerationsgruppe ändern
REMOVE-CE-LOCK	Sperre des Katalogeintrags zurücksetzten
REPAIR-FILE-LOCKS	unberechtigte Dateisperre zurücksetzen
SHOW-CE-LOCK	Sperre des Katalogeintrags anzeigen
SHOW-FILE-ATTRIBUTES	Merkmale einer Datei ermitteln
SHOW-FILE-LOCKS	Dateisperren anzeigen
SHOW-PUBSET-CATALOG-ALLOCATION	Informationen über Kataloge eines Pubsets ausgeben

Tabelle 15: Kommandoübersicht zur Verwaltung des Dateikataloges und der Katalogeinträge

8.1.1 Aufbau eines Dateikataloges

Der Dateikatalog enthält für jede auf dem Pubset existierende Datei, also auch für sich selbst, einen Eintrag, in dem u.a. die Dateimerkmale, die Schutzkriterien und die Lage der Datei auf dem Datenträger vermerkt sind.

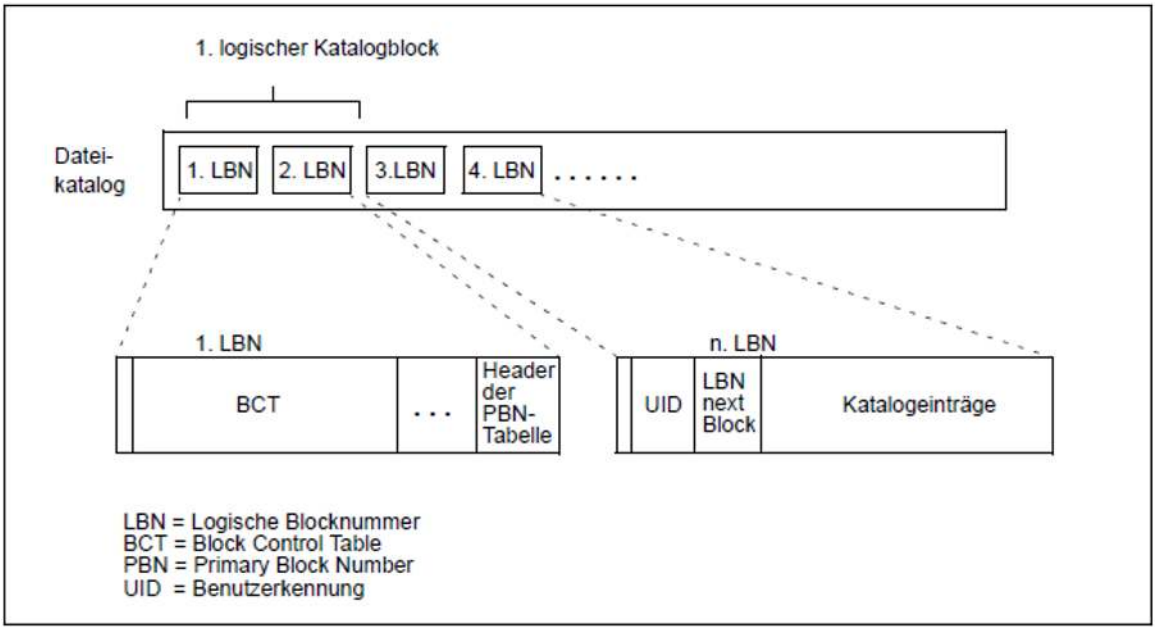


Bild 7: Aufbau des Dateikatalogs

Im ersten Katalogblock (4K-Block) sind Daten zur Verwaltung des Dateikatalogs hinterlegt. Die Katalogeinträge (Catalog Entries; CEs) stehen in den nachfolgenden Katalogblöcken. Der Katalog ist nach Benutzerkennungen strukturiert. Ein Block eines Katalogs enthält jeweils nur CEs einer Benutzerkennung; die CEs sind unsortiert sequenziell abgespeichert. Reicht der Platz im Block zur Aufnahme zusätzlicher CEs nicht mehr aus, wird ein weiterer Block angefordert und mit dem letzten Block dieser Benutzerkennung verkettet.

Alle Benutzer im Katalog werden zusammen mit dem Beginn ihrer Benutzerkette (PBN, Primary Block Number) in der PBN-Tabelle des Katalogs verwaltet.
Bei Aufnahme einer neuen Benutzerkennung in den Benutzerkatalog eines SF-Pubsets wird automatisch ein neuer Katalogblock reserviert, der die PBN dieser Kennung ist. Die logische Blocknummer (LBN) dieses Katalogblocks wird zusammen mit der Benutzerkennung in der PBN-Tabelle eingetragen.
In einem SM-Pubset wird für den neuen Benutzer zunächst nur ein Eintrag in der PBN-Tabelle des Control-Volume-Set-Katalogs erzeugt. Die PBN-Vergabe in einem SM-Pubset-Katalog erfolgt erst dann, wenn in diesem Katalog der erste CE des Benutzers abgelegt wird.

Der zweite 4K-Block des Dateikatalogs enthält immer den CE des Dateikatalogs selbst und ist der Beginn der Benutzerkette der Kennung TSOS.

Die logische Blockgröße im TSOSCAT beträgt 4KB, die logische Blockummer (LBN) wird als Nummer der logisch kleinsten, im 4-KB-Block enthaltenen PAM-Seite geführt.

Ein Katalogblock kann 1 bis 13 Katalogeinträge aufnehmen. Die Maximallänge eines Eintrags beträgt 4079 Byte, die Minimallänge (mit 1 Byte langem Dateinamen) 301 Byte, solange kein Platz zugewiesen ist. Ansonsten erhöht sich dieser Wert um die Länge der Extent-Liste. Diese beträgt mindestens 14 Byte bei Volumes < 32 GB und 16 Byte bei Volumes >= 32 GB.

Katalogformate

BS2000 OSD/BC bedient ab V11.0 nach Möglichkeit nur mehr das Katalogformat EXTRA LARGE (max. Größe 64.016 PAM-Seiten). Die bisherigen Katalogformate NORMAL und LARGE sind obsolet.

Kataloge einer Vorgängerversion im Format NORMAL oder LARGE werden automatisch in das Format EXTRA LARGE konvertiert, sobald der Pubset exklusiv oder als Master-Pubset importiert wird. Dabei werden auch die Spezialkataloge umbenannt:

- `:<catid>:$TSOS.TSOSCAT.#MIG` in `:<catid>:$TSOS.TSOSCAT.#M00`
- `:<catid>:$TSOS.TSOSCAT.#PVT` in `:<catid>:$TSOS.TSOSCAT.#J00`
- `:<catid>:$TSOS.TSOSCAT.#JVC` in `:<catid>:$TSOS.TSOSCAT.#P00`

Derart konvertierte Kataloge können auch in einer Vorgängerversion wieder verwendet werden.

Lediglich bei Import eines Shared Pubsets im Slave-Modus bleibt das vom Pubset-Master gewählte Katalogformat erhalten. Selbst dann können die Katalogformate NORMAL und LARGE aber nur auftreten, wenn der Master in einer Vorgängerversion läuft.

Dateikatalog auf SF-Pubsets

Der Dateikatalog ist in der Datei mit dem Namen `:<catid>:$TSOS.TSOSCAT` untergebracht.
Er beinhaltet sämtliche Benutzer- und Systemdateien sowie die Jobvariablen.

Dateikatalog auf SM-Pubsets

Im Gegensatz zu SF-Pubsets ist der Katalog eines SM-Pubsets aus mehreren Katalogdateien aufgebaut. Der Aufbau wird in folgendem Bild anhand eines Beispiels dargestellt. Im Beispiel besteht das SM-Pubset aus den drei Volume-Sets A, BLA und C7, die Katalogkennung ist X, das Katalog-Format ist EXTRA LARGE.

Teilkataloge bei Katalogformat EXTRA-LARGE

Beim Katalogformat EXTRA-LARGE gibt es jeweils bis zu 100 Teilkataloge, die jeweils bei einer notwendigen Katalogerweiterung angelegt werden, mit den Namen:

`:<catid>:$TSOS.TSOSCAT.#Mnn` (nn = 00 bis 99)
`:<catid>:$TSOS.TSOSCAT.#Pnn` (nn = 00 bis 99)
`:<catid>:$TSOS.TSOSCAT.#Jnn` (nn = 00 bis 99)

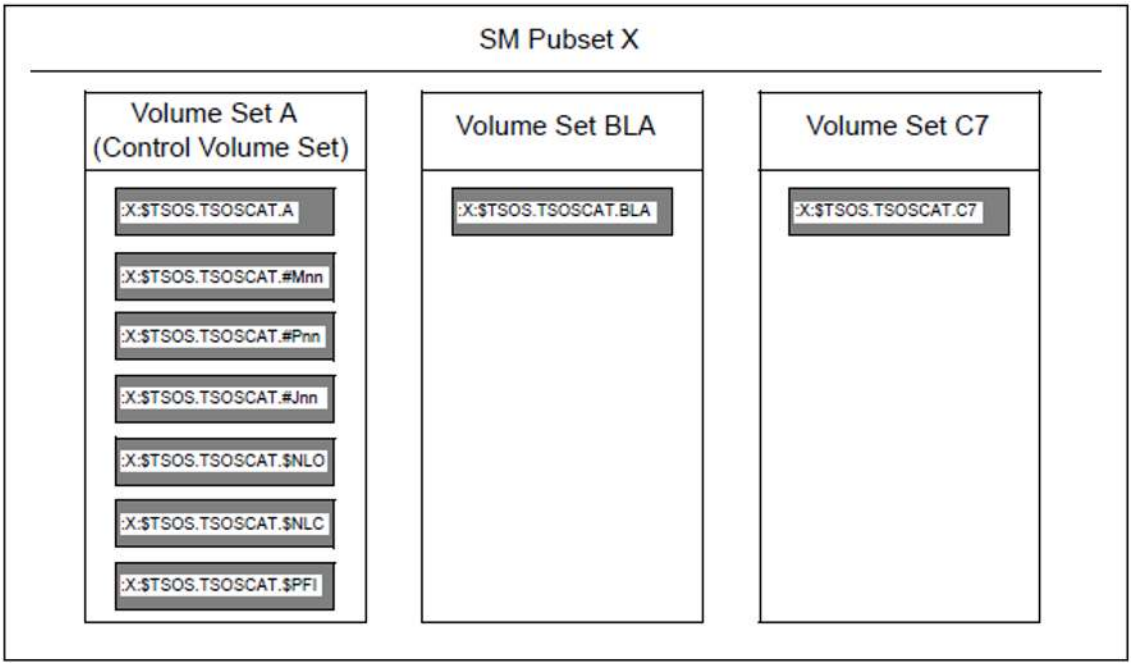


Bild 8: Aufbau des Dateikatalogs eines SM-Pubsets

Erläuterung

`:X:$TSOS.TSOSCAT.A`

`:X:$TSOS.TSOSCAT.BLA`

`:X:$TSOS.TSOSCAT.C7`

Jeder Volume-Set eines SM-Pubsets enthält eine volume-set-spezifische Katalogdatei mit den Katalogeinträgen für die auf ihm liegenden Dateien. Der Pfadname der volume-set-spezifischen Katalogdatei ergibt sich aus Pubset-Kennung und Volume-Set-Kennung wie folgt: :<catid>:$TSOS.TSOSCAT.<volume-set-id>.

Die folgenden Katalogdateien liegen alle auf dem Control-Volume-Set des SM-Pubsets:

`:X:$TSOS.TSOSCAT.#Mnn`

Katalogeinträge für auf eine Hintergrundebene migrierte Dateien sowie katalogisierte Dateien, die keinen Platz belegen.

`:X:$TSOS.TSOSCAT.#Pnn`

Katalogeinträge von Dateien auf Net-Storage, Banddateien und Privatplattendateien.

`:X:$TSOS.TSOSCAT.#Jnn`

Behälter für die Jobvariablen.

`:X:$TSOS.TSOSCAT.$PFI`

Katalog-Index-Datei, dient der Sicherung des Katalog-Index bei der Außerbetriebnahme eines SM-Pubsets. Sie ist eine den SM-Pubset-Katalogen übergeordnete Verwaltungsinstanz, die Informationen über die Lage einer Datei oder einer Jobvariablen im SM-Pubset enthält.

`:X:$TSOS.TSOSCAT.$NLO`

Namensliste (Name-List), dient der Datensicherheit. Bei Ausfall eines Volume-Sets können mit ihrer Hilfe die durch den Ausfall verlorengegangenen Daten rekonstruiert werden.
In der Namensliste existiert ein Eintrag für jede Datei, die in einem Volume-Set des SM-Pubsets existiert. Auch temporäre Dateien besitzen einen solchen Verweis.
Lediglich für Jobvariablen und Dateien, die in Katalogen auf dem Control-Volume-Set abgelegt sind, existieren keine solchen Verweise, denn mit Ausfall des Control-Volume-Sets ist auch kein Zugriff mehr auf die Namensliste möglich.
Der Eintrag für einen CE in der Namensliste besteht aus der Kennung des Katalogs, in dem der CE abgelegt ist, der Benutzerkennung des Datei-Eigentümers und dem Dateinamen.

`:X:$TSOS.TSOSCAT.$NLC`

Namensliste-Kopie (Name-List-Copy), wird zur Rekonstruktion der Namensliste nach einem Systemabsturz benötigt.
Die Namensliste und die Namensliste-Kopie sollten gleich groß angelegt werden, weil bei jeder Rekonstruktion des Katalog-Indexes die Namensliste zur Namensliste-Kopie wird und umgekehrt. Bei der Verwaltung der Namensliste könnte das CMS dann nur die kleinere der beiden Dateigrößen berücksichtigen.

Dateikatalog auf Net-Storage-Volumes

Mit BS2000 V21.0A wird, neben dem bereits existierenden Net-Storage-Volume vom Typ NETSTOR, ein zweiter Typ NETVOL eingeführt.

Während für Dateien auf einem Net-Storage-Volume vom Typ NETVOL nur die Katalogeinträge im TSOSCAT des Pubsets existieren, hat jedes Net-Storage-Volume vom Typ NETSTOR einen eigenen Dateikatalog mit dem Namen `.BS2FSCAT`.

Er enthält für alle Dateien, die auf dem Net-Storage-Volume vom Typ NETSTOR abgelegt sind, jeweils eine Kopie des Katalogeintrags im TSOSCAT des Pubsets, dem das Net-Storage-Volume zugeordnet ist.

8.1.2 Katalogdateien erweitern

Das Kommando SHOW-PUBSET-CATALOG-ALLOCATION gibt den Katalogtyp, den Belegungsgrad pro Katalog (Nutzung/Dateigröße) und die Erweiterbarkeit pro Katalog aus. Diese Ausgabe kann entweder für einen einzelnen Pubset oder für alle an einem System lokal und im Master-Modus importierten Pubsets erfolgen.

Automatische Vergrößerung des Katalogs

Das CMS erkennt, wenn ein Katalog zu 90% belegt ist und nimmt automatisch eine Katalogvergrößerung vor, sofern das ohne Wechsel des Katalogformats möglich ist.

Für Spezialkataloge des Typs EXTRA LARGE wird dann ein neuer Teilkatalog angelegt, wenn keiner der bereits existierenden Teilkataloge vergrößert werden kann.

Manuelle Vergrößerung des Katalogs

Die Katalogdateien können im laufenden Betrieb z.B. mit dem Kommando MODIFY-FILE-ATTRIBUTES ..., SPACE=... erweitert werden. Dies gilt sowohl auf SM- als auch auf SF-Pubsets. Die Vergrößerung des Katalogs wird sofort wirksam, nicht erst nach dem nächsten Import des Pubsets.

Wurde für einen SF-Pubset das Produkt SCA aktiviert, so wird SCA bei einer Erweiterung der Datei TSOSCAT implizit vom System beendet und wieder neu gestartet (zur Anpassung der SCA-Tabellenstruktur). Dies führt i.a. zu keiner Beeinträchtigung des Systems.

Für SM-Pubsets mit Katalogen des Typs EXTRA LARGE können für jeden der Spezialkataloge #M00, #J00 und #P00 bis zu 99 weitere Teilkataloge hinzugefügt werden. Dazu wird das Kommando ADD-CATALOG-FILE verwendet.

8.1.3 Performance bei Katalogzugriffen verbessern

- Der Benutzer sollte bei den Kommandos START-/LOAD-EXECUTABLE-PROGRAM, CALL-PROCEDURE, ENTER-PROCEDURE und ENTER-JOB *$dateiname* statt *dateiname* angeben.
  Gibt er z.B. START-EXECUTABLE-PROGRAM FROM-FILE=$EDT wird sofort der Bezug zu der bei DEFLUID angegebenen Benutzerkennung hergestellt.
  Bei Angabe von START-EXECUTABLE-PROGRAM FROM-FILE=EDT wird zuerst unter der aufrufenden (eigenen) Benutzerkennung nach einer Datei mit Namen EDT gesucht und danach unter der bei DEFLUID angegebenen Benutzerkennung.
- Die Systemverwaltung sollte einen geeigneten Wert für die Sekundärspeicherplatz-Zuweisung festlegen (Systemparameter DMSCALL), denn die Sekundärspeicherplatz-Zuweisung ist sehr zeitaufwändig, da bei jeder Anforderung durch die Benutzertask der entsprechende Katalogeintrag geändert wird.
- Um einer Sättigung des gemeinschaftlichen Speicherplatzes wirksam vorzubeugen, ist es ratsam, unbenutzten Speicherplatz der Benutzerdateien freizugeben.
  Dazu kann die Systemverwaltung Prozeduren erstellen.
  Siehe Abschnitt „Speicherplatz-Sättigung überwachen".

8.1.4 Zugriffssperren aufheben

Sperre des Katalogeintrags (CE-Lock) aufheben

Um die Konsistenz von Katalogeinträgen zu gewährleisten (z.B. bei der Systemkommunikation) können Systemkomponenten den Katalogeintrag einer Datei bzw. JV zeitweise gegen Änderungen sperren. Kann ein CE-Lock z.B. wegen eines Fehlers bei der Systemkommunikation nicht mehr zurückgesetzt werden, bleibt der Katalogeintrag weiter gesperrt, obwohl die Sperre nicht mehr notwendig wäre.

Solch ein „hängender“ CE-Lock behindert die Arbeit von Tasks, die diese Datei oder JV für Verarbeitungsvorgänge benötigen.

Mit dem Kommando REMOVE-CE-LOCK kann die Systembetreuung eine Sperre für den Katalogeintrag einer Datei oder JV zurücksetzen. Das Kommando wird nur ausgeführt, wenn die Task des Lock-Halters nicht mehr vorhanden ist oder sich im im Zustand PENDING INDEFINITELY befindet.
Befindet sich die Task des Lock-Halters in einem anderen Zustand, dann kann der CE-Lock nicht freigegeben werden und das Kommando wird zurückgewiesen. In diesem Fall ist zu prüfen, ob die Task des Lock-Halters beendet werden kann (z.B. Kommando CANCEL-JOB). Mit dem Kommando SHOW-CE-LOCK kann die TID dieser Task und die Sysid des Systems, in dem die Task arbeitet, ermittelt werden.

Für Shared Pubsets können die Kommandos REMOVE- und SHOW-CE-LOCK von einem beliebigen System des Verbunds eingegeben werden.

Dateisperre aufheben

Mit dem Kommando REPAIR-FILE-LOCKS hebt der Dateieigentümer bzw. die Systembetreuung (Benutzerkennung TSOS) „unberechtigte“ Dateisperren für eine Datei auf. Dabei handelt sich um Dateisperren, die nicht mehr notwendig sind, aber aus folgenden Gründen vom System nicht mehr automatisch zurückgesetzt werden konnten:

- kurzfristiger Verbindungsausfall in einem Rechnerverbund
- Systemfehler verhinderte das Rücksetzen einer Dateisperre

Die für eine Datei vorhandenen Sperren können durch einen Job, durch eine Systemkomponente oder für die Dateiübertragung gesetzt sein.

Das Kommando SHOW-FILE-LOCKS informiert den Dateieigentümer und alle zugriffsberechtigten Benutzer über die aktuell wirksamen Sperren einer Datei. Nicht angezeigt werden Dateisperren, die auf Grund des Katalogeintrags wirksam sind. Diese Sperren werden mit dem Kommando SHOW-CE-LOCK angezeigt.

8.2 ACS: Alias-Katalogsystem

Mit ACS (Alias Catalog Service) ist es möglich, auf Dateien und Jobvariablen unter zusätzlichen, in gewissen Grenzen frei wählbaren Namen zuzugreifen. Dem Benutzer wird die Möglichkeit geboten, für die von ihm benötigten Dateien/JVs Aliasnamen zu definieren und zusammen mit der Zuordnung zu realen Dateien/JVs in speziellen Katalogen, den Aliaskatalogen, zu hinterlegen. Bei der Bearbeitung der Datei/JV ist dann nur noch der Aliasname anzugeben. Die Kataloge werden tasklokal geführt.

Detaillierte Informationen zu ACS finden Sie im Handbuch „Einführung in das DVS“ [19].

Kommando	Bedeutung
ADD-ACS-SYSTEM-FILE	Neue Identifikation für eine AC-Systemdatei vereinbaren und der Identifikation einen Dateinamen zuordnen
ADD-ALIAS-CATALOG-ENTRY	Eintrag im Aliaskatalog erzeugen und Wirkungsbereich der Ersetzung festlegen
HOLD-ALIAS-SUBSTITUTION	ACS-Funktion Namensersetzung unterbrechen
LOAD-ALIAS-CATALOG	Gespeicherte Einträge aus einer AC-Datei in den Aliaskatalog übernehmen
MODIFY-ACS-OPTIONS	ACS-Optionen tasklokal oder systemweit verändern
MODIFY-ACS-SYSTEM-FILE	Definition einer AC-Systemdatei verändern
MODIFY-ALIAS-CATALOG-ENTRY	Bestehenden Eintrag im Aliaskatalog modifizieren
PURGE-ALIAS-CATALOG	Aktuellen Aliaskatalog löschen
REMOVE-ACS-SYSTEM-FILE	Definition einer AC-Systemdatei löschen
REMOVE-ALIAS-CATALOG-ENTRY	Bestehenden Eintrag im Aliaskatalog löschen
RESUME-ALIAS-SUBSTITUTION	ACS-Funktion Namensersetzung wiederaufnehmen
SET-FILE-NAME-PREFIX	Namenspräfix vereinbaren und Wirkungsbereich der Ersetzung festlegen
SHOW-ACS-OPTIONS	Gültige ACS-Optionen anzeigen
SHOW-ACS-SYSTEM-FILES	Identifikationen der AC-Systemdateien anzeigen
SHOW-ALIAS-CATALOG-ENTRY	Informationen über die Aliaskatalogeinträge anfordern
SHOW-FILE-NAME-PREFIX	Aktuellen Präfix und seinen Gültigkeitsbereich anzeigen
START-ACS	ACS systemweit aktivieren
STORE-ALIAS-CATALOG	Aliaskatalog in Datei abspeichern

Tabelle 16: Kommandoübersicht für das Alias-Katalogsystem

8.3 ADM-PFA: Administrator Performant File Access

Das bislang unter dem Namen ADM-PFA bekannte Datei-Caching wird im Handbuch zu „DAB“ [10] beschrieben.

8.4 Versenden von BS2000-Dateien per E-Mail

Mit dem Kommando MAIL-FILE oder dem Makro MAILFIL können Sie BS2000-Dateien (auch SYSLST und SYSOUT) an die angegebene Benutzerkennung versenden. Diese E-Mail wird an eine ausgewählte oder an alle Empfängeradressen versandt, die im Benutzereintrag der Benutzerkennung eingetragen sind. Die E-Mail-Adresse muss bei der Benutzerkennung eingetragen sein, siehe „E-Mail-Adressen für Benutzerkennung eintragen“.

Als Voraussetzung muss der Mail-Service des Softwareprodukts interNet Services installiert sein, denn der eigentliche Versand der E-Mail erfolgt mit dem Mail-Sender (Schnittstelle SEND-MAIL) von interNet Services.

Für die zu versendende Text-Datei wird der Zeichensatz (Dateiattribut CCS-NAME) berücksichtigt und beim Versand eine Konvertierung in einen Zeichensatz der offenen Systemwelt durchgeführt. Sie können der E-Mail einen „Betreff“ (SUBJECT) hinzufügen und festlegen, ob die Datei nach erfolgreichem Versand gelöscht werden soll.

Folgende Dateien können Sie mit diesem E-Mail-Konzept ebenfalls versenden:

- MAIL-FILE für Systemdateien SYSLST und SYSOUT, siehe "Ausgabe von SYSLST und SYSOUT mit MAIL-FILE".
- HSMS-Reports mit HSMS, siehe Handbuch „HSMS“ [24].
- Ausgabedateien von MAREN, siehe Handbuch „MAREN“ [31].

Als Absender der E-Mail wird bei erfolgreicher Auswahl einer E-Mail-Adresse über den Jobnamen diese E-Mail-Adresse verwendet, sonst die erste E-Mail-Adresse der Ablaufkennung. Zusätzlich wird für mögliche Zustellfehler die erste E-Mail-Adresse der Kennung TSOS mitgegeben. Auf diese Weise erhält der Administrator bei falschen Empfangsadressen eine sogenannte „Bounce-Mail“ und kann daraufhin die entsprechenden E-Mail-Adressen in der Benutzerverwaltung korrigieren.

E-Mail-Adressen für Benutzerkennung eintragen

E-Mail-Adressen werden mit dem Operanden EMAIL-ADDRESS der Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES im Benutzereintrag der BS2000-Benutzerkennungen eingetragen. Das Attribut MAIL-ADDRESS bleibt unverändert.

Für die Benutzerkennung TSOS muss eine E-Mail-Adresse eingetragen sein.

Mit dem Kommando SHOW-USER-ATTRIBUTES bzw. dem Makro SRMUINF werden die eingetragenen E-Mail-Adressen ausgegeben.

Nach Eintrag neuer E-Mail-Adressen sollte die Systembetreuung zur Kontrolle eine Test-Mail an die zugehörige Benutzerkennung versenden.

Ausgabe von SYSLST und SYSOUT mit MAIL-FILE

MAIL-FILE kann die aktuellen Systemdateien SYSLST und SYSOUT ausgeben. In den Kommandos EXIT-JOB, CANCEL-JOB und ENTER-PROCEDURE kann für die Systemausgabe im entsprechenden Ausgabeoperanden der Wert MAIL anstelle von PRINT angegeben werden. Außerdem kann für diese Kommandos und für das Kommando LOGOFF der Standardwert der Systemausgabe im Systemparamter SSMOUT eingestellt werden auf PRINT oder MAIL. Siehe Systemparameter SSMOUT.

Wenn bei Job-Ende kein E-Mail-Versand möglich ist, dann werden Systemdateien, die gemäß den Einstellungen als E-Mail verschickt werden müssten, ersatzweise wie bisher über SPOOL ausgegeben.

Fehleranalyse im Mail-Service des Softwareprodukts interNet Services

Der Systemverwalter kann über die Konfigurationsdatei für das Mail-Sender Backend (Datei SYSDAT.MAIL.<version>.SERVICE.OPT) und über das Kommando MODIFY-MAIL-SERVICE-PARAMETER Einstellungen für Logging und Trace von E-Mails treffen. Logging- und Trace-Datei enthalten nähere Informationen zum Versand von E-Mails, insbesondere in Fehlerfällen.

8.5 Unicode in BS2000

Unicode fasst alle weltweit bekannten Textzeichen in einem einzigen Zeichensatz zusammen. Zudem ist Unicode unabhängig von unterschiedlichen Herstellern, Systemen und Ländern.

Mit der Unicode-Unterstützung im BS2000 werden die in BS2000-Systemen verfügbaren EBCDIC-Zeichensätze um zusätzliche Zeichen erweitert, die im europäischen Sprachraum benötigt werden. Dem Anwender wird die Programmier- und Ablaufumgebung zur Verfügung gestellt, die er braucht, um seine bestehenden Anwendungen um Unicode-Datenfelder zu erweitern. Hierzu wird eine entsprechende Software-Konfiguration unter BS2000 bereitgestellt. Das Subsystem XHCS zur Unicode-Unterstützung in BS2000 wird standardmäßig geladen.

Allgemeine Informationen zu Unicode entnehmen Sie bitte der Internetseite des Unicode-Konsortiums: *http://www.unicode.org/standard/translations/german.html*.

Nähere Hinweise zum Einsatz von Unicode in BS2000 finden Sie im Handbuch „Unicode im BS2000“ [58].

9 Pubset-Verwaltung

Pubsets (Public-Volume-Sets) sind Sätze gemeinschaftlicher Platten und in BS2000 neben den privaten Datenträgern der Ablageort für Dateien. Ein Pubset enthält neben den Dateien selbst auch alle für die Dateiverwaltung erforderlichen Metadaten (Dateikatalog, Benutzerkatalog, usw.).

ℹ BS2000 bedient den so genannten Net-Storage im Rahmen von Pubsets. Die Beschreibung dazu finden Sie im Kapitel „Net-Storage-Verwaltung".

Kommando	**Bedeutung**
ADD-MASTER-CATALOG-ENTRY	Eintrag im Katalogverzeichnis MRSCAT erzeugen
ADD-NET-STORAGE-VOLUME	Net-Storage-Volume einrichten und einem lokalen Pubset zuordnen
ADD-USER	Eintrag im Benutzerkatalog erstellen und Pubset-Zugriff festlegen
CANCEL-PUBSET-EXPORT	Exportieren eines Pubsets abbrechen
CANCEL-PUBSET-IMPORT	Importieren eines Pubsets abbrechen
CHECK-PUBSET-MIRRORS	Homogenität der Pubset-Spiegelung prüfen
EXPORT-PUBSET	Zuvor importierten Pubset exportieren
FORCE-PUBSET-EXPORT	Exportieren eines Pubsets erzwingen
IMPORT-PUBSET	Importieren eines Pubsets und Behandlung des Benutzerkataloges festlegen
MODIFY-MASTER-CATALOG-ENTRY (EDIT-MASTER-CATALOG-ENTRY)	Eintrag im Katalogverzeichnis MRSCAT ändern
MODIFY-PUBSET-DEFINITION-FILE (EDIT-PUBSET-DEFINITION-FILE)	Definition eines SM-Pubsets ändern
MODIFY-PUBSET-PROCESSING	Zusammensetzung eines Pubsets ändern
MODIFY-PUBSET-RESTRICTION	Nutzungseinschränkungen für einen Pubset ändern

MODIFY-PUBSET-SPACE-DEFAULTS (EDIT-PUBSET-SPACE-DEFAULTS)	Standardwerte der Speicherplatzverwaltung ändern
MODIFY-SPACE-SATURATION-LEVELS (EDIT-SPACE-SATURATION-LEVELS)	Schwellwerte für Speicherplatzsättigung ändern
MODIFY-USER-ATTRIBUTES	Eintrag im Benutzerkatalog ändern
MODIFY-USER-PUBSET-ATTRIBUTES	SM-Pubset-Attribute einer Benutzerkennung ändern
REMOVE-MASTER-CATALOG-ENTRY	Eintrag im Katalogverzeichnis MRSCAT löschen
REMOVE-NET-STORAGE-VOLUME	Net-Storage-Volume einem lokalen Pubset entziehen
REMOVE-PUBSET-LOCK	Pubset-Lock entfernen
RESUME-PUBSET-RECONFIGURATION	Pubset-Rekonfigurationsauftrag ordnungsgemäß beenden
SET-PUBSET-ATTRIBUTES	Charakteristiken eines Pubsets festlegen
SET-SPACE-SATURATION-LEVEL	Sättigungsstufen für die Ausnutzung des Speicherplatzes auf einem SF-Pubset festlegen (wird nur noch aus Kompatibilitätsgründen unterstützt)
SHOW-MASTER-CATALOG-ENTRY	Informationen über Zustand, Belegung und Zugreifbarkeit eines Pubsets anfordern
SHOW-PUBSET-ATTRIBUTES	Informationen über Pubset-Eigenschaften anfordern
SHOW-PUBSET-DEFINITION-FILE	Informationen über die globalen und die Performance-Eigenschaften eines Volume-Sets ausgeben
SHOW-PUBSET-LOCKS	Informationen über Pubset-Locks ausgeben
SHOW-PUBSET-FILE-SERVICES	Die in einem SM-Pubset unterstützten Kombinationen von Dateieigenschaften anzeigen
SHOW-PUBSET-IMPORT-EXPORT	Information über den Bearbeitungsstand von Import-/Exportaufträgen ausgeben
SHOW-PUBSET-NET-STORAGE	Net-Storage eines Pubsets anzeigen
SHOW-PUBSET-OCCUPATION	Information über die Pubset-belegenden Jobs ausgeben
SHOW-PUBSET-PROCESSING	Information über die physikalische Pubset-Konfiguration ausgeben
SHOW-PUBSET-RESTRICTION	Information über die physikalische Pubset-Konfiguration ausgeben
SHOW-PUBSET-SPACE-ALLOCATION	Speicherplatzbelegung eines Pubsets ausgeben
SHOW-PUBSET-SPACE-DEFAULTS	Information über die Pubset-spezifischen Standardwerte für die Speicherzuweisung ausgeben

SHOW-SHARED-PUBSET	Übersicht über die Teilnehmer eines Shared-Pubset-Verbundes anfordern
SHOW-SPACE-SATURATION-LEVELS	Information über die Pubset-spezifischen Speichersättigungsstufen ausgeben
SHOW-XCS-PUBSET	Übersicht über die Teilnehmer eines XCS-Pubset-Verbundes anfordern
SHOW-XCS-OCCUPATION	Die momentan XCS-nutzenden TU-Tasks anzeigen

Tabelle 19: Kommandoübersicht zur Pubset-Verwaltung

9.1 Pubset-Konzept

Ein Pubset ist eine Zusammenfassung gemeinschaftlich genutzer Platten zu einer eigenen Einheit. Pubsets sind wegen ihrer Abgeschlossenheit weitestgehend voneinander unabhängige Objekte. Beim Ausfall eines Pubsets ist nur dieses davon betroffen.

9.1.1 Pubset-Arten

Die Pubsets werden nach Verwendungsart unterschieden:

- Der Home-Pubset, der in jedem BS2000-System vorhanden sein muss, siehe "Pubset-Namen (VSN)".
- Daten-Pubsets (auch Benutzer-Pubsets genannt), siehe "Benutzer-Pubset (Daten-Pubset)"
- Paging-Pubsets, siehe "Paging-Pubset"
- Pubsets für große Dateien und Volumes, siehe "Pubsets mit großen Objekten"
- Stand-by-Pubsets zur Erhöhung der Datensicherheit siehe "Besonderheiten für Standby-Pubsets"
- Shared Pubsets für gemeinsamen Zugriff, siehe "Shared-Pubsets"

9.1.1.1 Home-Pubset

In jedem BS2000-System muss mindestens ein Pubset existieren, der system-spezifische Daten und Dateien beinhaltet und deshalb während der gesamten BS2000-Session verfügbar sein muss. Dieser spezielle Pubset wird als Home-Pubset bezeichnet. Er kann nicht mit Net-Storage verbunden werden.

Zur Identifikation des laufenden Systems wird die Systemkennung (Sysid) des Home-Pubsets verwendet. Sie geht in die Plattenbelegung und in die Pfadgruppen-Id ein. Die Pfadgruppen-Id dient Controllern zur Unterscheidung von I/Os verschiedener Systeme.
Aus diesen Gründen ist es notwendig, dass bei parallel betriebenen Home-Pubsets die Systemkennung eindeutig vergeben wird. Bei der PUB-Notation ist dies von selbst gegeben, sofern die Pubset-Kennungen eindeutig gewählt werden. Bei Pubsets mit Punkt-Notation, die als Home-Pubsets verwendet werden sollen, muss explizit auf eine eindeutige Systemkennung geachtet werden, siehe "Shared-Pubset-Verbund einrichten".

Das Importieren des Home-Pubsets wird während der Systemeinleitung durch BS2000 selbst veranlasst. Das Exportieren des Home-Pubsets geschieht während der Systembeendigung (bei Abarbeitung des Kommandos SHUTDOWN).
Es ist möglich, auf dem Home-Pubset Benutzerdateien und Jobvariablen zu katalogisieren. Dies muss allerdings von der Systembetreuung zugelassen werden (ADD-USER- oder MODIFY-USER-Kommando).

9.1.1.2 Benutzer-Pubset (Daten-Pubset)

Diese Pubsets dienen dem DVS ausschließlich als Dateiablageorte. Sie können auch mit Net-Storage verbunden werden. Sie enthalten nur Benutzerdateien oder -Jobvariablen. Die Verfügbarkeit dieser Pubsets kann mit den Kommandos IMPORT-PUBSET und EXPORT-PUBSET jederzeit während des Systemlaufs von der Systembetreuung gesteuert werden. Ein Benutzer kann auf mehreren Pubsets zugriffsberechtigt sein.

Jedem Benutzer wird ein Standard-Pubset (Default-Pubset) zugewiesen. Auf diesem Pubset kann er (ohne Angabe der Katalogkennung) Dateien und Jobvariablen erzeugen, verarbeiten und löschen.
Für die Benutzerkennung TSOS ist unabhängig von der DEFAULT-PUBSET-Angabe im ADD-USER- bzw. MODIFY-USER-ATTRIBUTES-Kommando der Standard-Pubset immer der Home-Pubset.

9.1.1.3 Paging-Pubset

Die Pubsets, die die im Systemlauf zu verwendenden Paging-Dateien enthalten, werden als Paging-Pubsets bezeichnet. Sie werden für die gesamte BS2000-Session benötigt, können jedoch davon unabhängig – aber nur vom gleichen System – für DVS-Nutzung beliebig in und außer Betrieb genommen werden.

Paging-Pubsets werden vom DVS wie die Pubsets behandelt, auf denen keine Paging-Dateien angelegt sind: der Import-Auftrag für einen Daten-Pubset mit Paging-Datei muss explizit erfolgen (Kommando IMPORT-PUBSET). Ein automatisches Importieren bei der Systemeinleitung findet für die Paging-Pubsets nicht statt. Um die benötigten Pubsets sofort nach der Systemeinleitung zur Verfügung zu haben, sollte deren Import über den Anstoß einer RUN-Datei realisiert werden.
Die Paging-Pubsets sind im Katalogverzeichnis MRSCAT als solche gekennzeichnet und können dort nicht gelöscht werden.
Mit dem Kommando SHOW-MASTER-CATALOG-ENTRY CATALOG-ID=*ALL,SELECT= *PAGING lassen sich alle Paging-Pubsets des Systems anzeigen.

9.1.1.4 Pubsets mit großen Objekten

Volumes und Dateien, deren Größe 32 GB überschreitet, werden auch als „große Volumes“ bzw. „große Dateien“ oder gemeinsam als „große Objekte“ bezeichnet. Die max. mögliche Größe beträgt 4 TB. Zur Unterstützung großer Objekte gibt es zwei Pubset-Typen:

- Pubsets mit großen Volumes ohne große Dateien
- Pubsets mit großen Volumes und großen Dateien

Damit ist es möglich, große Volumes und große Dateien gezielt stufenweise einzuführen. Die Einführung großer Volumes ist weitestgehend transparent für existierende Programme. Bei der Einführung großer Dateien sind u.U. Anpassungen vorzunehmen.

9.1.2 Pubset-Namen (VSN)

Alle Datenträger werden in BS2000 durch einen Namen identifiziert, die so genannte VSN (Volume Serial Number).

Für Pubsets ist zwischen einstelligen Namen (VSN in PUB-Notation) und mehrstelligen Volume-Set- bzw. Pubset-Namen (VSN in Punkt-Notation) zu unterscheiden. Diese VSN-Syntax darf für private Platten nicht angewandt werden.

Innerhalb eines Rechnerverbundes müssen die Pubset-Namen eindeutig sein.

Für Net-Storage-Volumes gibt es eine spezielle Notation, siehe "Notation von Net-Storage-Volumes" im Abschnitt "Begriffe".

9.1.2.1 VSN in PUB-Notation

Eine VSN in PUB-Notation beginnt mit der festen Zeichenkette PUB (für public). Sie besteht immer aus 6 Zeichen und hat das Format **PUBpxx**:

PUB unveränderlicher Bestandteil zur Unterscheidung von privaten Platten (3 Zeichen „PUB“) = Typbezeichner

p Katalogkennung (Catid), (1 Zeichen; A..Z, 0..9)

xx Nummer innerhalb eines Pubsets/Volume-Sets, (2 Zeichen; 00..31) = Folgenummer

Mit der einstelligen Katalogkennung können maximal 36 Pubsets bzw. Volume-Sets adressiert werden, die aus bis zu 32 Platten bestehen können.

Beispiele: PUBA00, PUBA25, PUB502

9.1.2.2 VSN in Punkt-Notation

Eine VSN in Punkt-Notation besteht immer aus 6 Zeichen und hat das Format **pp[pp].[xy]z.** Dabei ist:

pp[pp] Katalogkennung (Catid), (2-4 Zeichen; je A..Z, 0..9); Präfix „PUB“ ist nicht erlaubt

. Punkt (1 Zeichen) = Typbezeichner

[xy]z Nummer innerhalb eines Pubsets/Volume-Sets (1-3 Zeichen) = Folgenummer

Ein Pubset bzw. Volume-Set in Punkt-Notation kann aus bis zu 255 Platten bestehen.

Beispiele: AA.001, AB.309, XYZ.23, OTTO.0, J19P.8

Die Katalogkennung bei SF-Pubsets korrespondiert direkt mit dem Namensteil „Katalogkennung“ in der VSN. Bei SM-Pubsets ist die Katalogkennung verschieden von allen Volume-Set-Namen dieses SM-Pubsets, und damit auch verschieden vom VSN-Namensteil „Katalogkennung“ aller Platten des Pubsets.

Maximale Anzahl von Platten pro Pubset/Volume-Set bei Punkt-Notation

Die maximale Anzahl der Platten pro Pubset/Volume-Set hängt von der Allokierungseinheit AU (ALLOCATION-UNIT) der Plattentypen und von der Länge der Folgenummer ab.

AU	Wertebereich der Folgenummer x	Wertebereich der Folgenummer y	Wertebereich der Folgenummer z	maximale Anzahl adressierbarer Platten
6 KB	0	0	A..V, 0..9	32
8 KB	A..Z, 0..9	A..Z, 0..9	A..Z, 0..9	36 / 255 [1]
64 KB	A..Z, 0..9	A..Z, 0..9	A..Z, 0..9	36 / 255 [1]

[1] bei einstelliger Folgenummer können nur maximal 36 Platten pro Pubset adressiert werden

9.1.2.3 Doppelpunkt-Notation für Spiegelplatten

Die Doppelpunkt-Notation dient dazu, Spiegelplatten eindeutig zu bezeichnen, nachdem sie von der Original-Unit abgetrennt wurden (z.B. zur Durchführung einer Sicherung mit HSMS). Spiegelplatten werden mit den Replikationsfunktionen externer Plattenspeichersysteme erstellt.

Dazu wird in einer VSN mit Punkt-Notation der Punkt in einen Doppelpunkt bzw. bei PUB-Notation die Zeichenfolge „PUB“ nach „P:B“ geändert.

Beispiele

			Doppelpunkt-Notation
Punkt-Notation:	ABC.04	wird zu	ABC:04
	XY.123	wird zu	XY:123
PUB-Notation:	PUB023	wird zu	P:B023
	PUBX88	wird zu	P:BX88

Bei der Erstellung eines Spiegel-Pubsets mit SHC-OSD kann diese Umbenennung implizit erfolgen. Es können jedoch nur SF-Pubsets implizit umbenannt werden.

9.1.2.4 Sonderfälle

Konvertierung oder Umbenennung des VSN-Formats

Mit dem Dienstprogramm PVSREN (siehe Handbuch „Dienstprogramme“ [15]) können Pubset-Notationen konvertiert oder Pubset- bzw. Volume-Set-Bezeichnungen innerhalb einer Notationsart umbenannt werden.

Eine Konvertierung von PUB-Notation in Punkt-Notation ist immer möglich.
Eine Konvertierung von Punkt-Notation in PUB-Notation ist nur dann möglich, wenn die Anzahl der Volumes des betroffenen Pubsets bzw. Volume-Sets nicht größer als 32 ist.

Snapset-Notation

Für Snap-Units, die bei der Erzeugung eines Snapsets verwendet werden, wird zur eindeutigen Benennung der VSNs die Snapset-Identifikation in Kleinbuchstaben so eingesetzt, dass einerseits aus der Snapset-Platte die zugehörige Pubset-Platte abgeleitet werden kann und andererseits die VSN der Snapset-Platte außerhalb des Namensraums der VSNs für Privatplatten und Pubsets liegt. (Details siehe "Snapset-Identifikation" im Abschnitt "Sicherungsbetrieb mit Snapsets").

9.1.2.5 Pubset-Adressierung

Die Adressierung der auf einem Pubset katalogisierten Objekte erfolgt über die Katalogkennung (Catid) durch die vollständigen Pfadnamen.

Die Catid ist identisch mit dem Namen des korrespondierenden Pubsets.
Die Catid eines SM-Pubsets muss verschieden von allen Volume-Set-Namen aller SM-Pubsets sein.

Die Adressierung von Dateien auf einem bestimmten Volume-Set eines SM-Pubsets über den Namen dieses Volume-Sets wird nicht unterstützt.

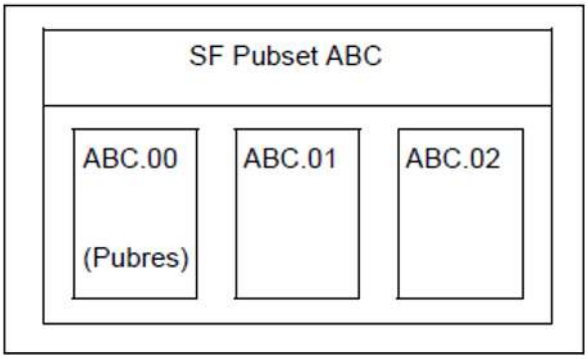
9.1.3 SF- und SM-Pubsets

Dieser Abschnitt beschreibt den Aufbau von SF- und SM-Pubsets sowie die Konfigurationsdaten, die für diese Pubsets von Bedeutung sind.

9.1.3.1 Aufbau eines SF-Pubsets

Ein Single-Feature-Pubset (SF-Pubset) besteht aus einer oder mehreren homogenen Platten, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit) übereinstimmen müssen. Die Platte mit der Nummer 0 ist die so genannte **Pubres**. Die Pubres ist die obligatorische Systemplatte eines SF-Pubsets. Sie enthält u.a. ein Verzeichnis der zum Pubset gehörenden Platten und einen Verweis auf den Beginn des Dateikatalogs TSOSCAT.

In folgendem Beispiel besteht der SF-Pubset mit der Katalogkennung ABC aus drei Platten mit den VSNs (in Punkt-Notation) ABC.00, ABC.01 und ABC.02.



Datenträgerkonfiguration eines SF-Pubsets (Beispiel)

Existiert eine Datei mit dem Dateinamen „MEINE.LISTE“ unter der Benutzerkennung „ALLERLEI“ auf einer der Platten des SF-Pubsets ABC, so lautet ihr Pfadname:
„:ABC:$ALLERLEI.MEINE.LISTE“

Eigenschaften von SF-Pubsets

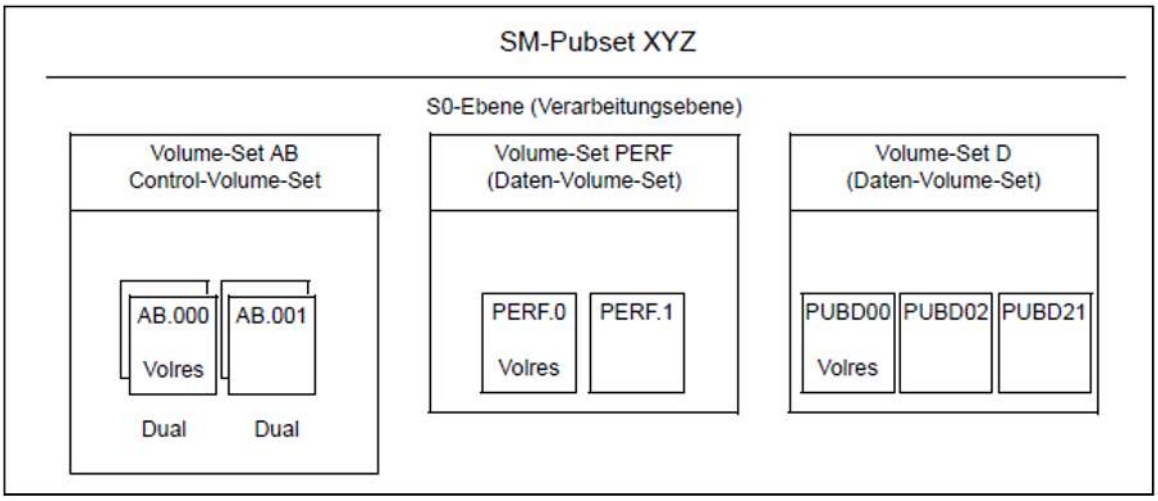
- Ein SF-Pubset repräsentiert ein begrenztes, in sich homogenes Service-Angebot. Ändern sich die Service-Anforderungen einer Datei, muss die Datei vom Benutzer auf einen anderen Pubset verlagert werden.
- SF-Pubsets können durch Hinzunahme leerer Platten vergrößert werden. Die Platteneigenschaften müssen dabei den Pubset-Eigenschaften entsprechen (siehe "SM-Pubsets einrichten und ersetzen").
- Durch Herausnehmen leerer Platten kann ein SF-Pubset verkleinert werden. Als Hilfsmittel für das Leeren von Platten wird seitens des Systems das Setzen von Allokierungseinschränkungen zur Verfügung gestellt.
- Bei Ausfall nur eines einzelnen Volumes des SF-Pubsets gilt der gesamte Pubset als ausgefallen.
- SF-Pubsets sind als Home- oder Paging-Pubsets geeignet.

9.1.3.2 Aufbau eines SM-Pubsets

Ein SM-Pubset besteht aus einem oder mehreren **Volume-Sets**, die wie bei einem SF-Pubset eine Zusammenfassung von mehreren homogenen Platten sind. Ein Volume-Set kann nur innerhalb des SM-Pubsets betrieben werden. Ein SM-Pubset kann maximal aus 255 Volume-Sets bestehen.

Unter den Volume-Sets nimmt der Control-Volume-Set eine Sonderrolle ein. Der **Control-Volume-Set** dient zur Ablage aller pubset-globalen Metadaten des SM-Pubsets. Dieser Control-Volume-Set muss deshalb immer vorhanden sein und kann nicht aus dem Pubset entfernt werden.

Das folgende Beispiel zeigt den Aufbau eines SM-Pubsets mit den drei Volume-Sets AB, PERF und D:



Datenträgerkonfiguration eines SM-Pubsets (Beispiel)

Existiert eine Datei mit dem Dateinamen „LST.TELEFON“ unter der Benutzerkennung „EINERLEI“ auf einer der Platten, die zu irgendeinem der Volume-Sets des SM-Pubsets XYZ zusammengefasst sind, so lautet ihr Pfadname: „:XYZ:$EINERLEI.LST.TELEFON“, d.h. für den Benutzer ist es i.A. nicht relevant, wie der SM-Pubset intern aufgebaut ist und wo genau seine Datei vom System abgelegt wurde.

Eigenschaften von SM-Pubsets

- Das Service-Angebot ist Volume-Set-spezifisch, d.h. jedes Volume-Set repräsentiert anhand seiner Eigenschaften einen speziellen Service-Typ innerhalb des SM-Pubsets. Volume-Sets mit gleichen Eigenschaften werden dem gleichen Service-Typ zugeordnet. Beim Neuanlegen einer Datei entscheidet das System, welches Volume-Set sich im Hinblick auf die Benutzeranforderung (Performance, Verfügbarkeit, Nutzung spezieller Anwendungen) am besten als Dateiablageort eignet.
- Dateien können innerhalb des SM-Pubset auf ein anderes Volume-Set verlagert werden, ohne die Adressierung zu ändern, d.h. die Verlagerung geschieht nicht sichtbar für den Benutzer.
- Ein SM-Pubset kann während der Betriebsphase um leere Volume-Sets erweitert oder verkleinert werden.
- Leere Volumes können zu einzelnen Volume-Sets hinzugefügt oder aus ihnen entfernt werden.

- Bei Ausfall eines einzelnen Volumes eines SM-Pubsets beschränkt sich der Schaden auf den betroffenen Volume-Set. Das hat zur Folge, dass lediglich alle auf dem betroffenen Volume-Set liegenden Dateien eventuell beschädigt sind, der restliche Teil des Pubsets ist nicht betroffen und kann ohne Unterbrechung weiterbenutzt werden.

  Bei Ausfall des Control-Volume-Sets gilt der gesamte SM-Pubset als ausgefallen. Für den Control-Volume-Set sollte deshalb durch die Systembetreuung eine besondere Ausfallsicherheit durch Nutzung von Spiegelfunktionen gewährleistet sein.
- SM-Pubsets können nicht als Home-Pubsets verwendet werden.
- Paging-Dateien können auf SM-Pubsets eingerichtet werden. Bei Nutzung der Paging-Dateien werden nur die Volume-Sets, zu denen die Paging-Platten gehören, auf Fremdbelegung überprüft.

Eigenschaften von Volume-Sets

- Ein Volume-Set wird durch eine Volume-Set-Kennung identifiziert, die den Hauptbestandteil der VSN aller zum Volume-Set gehörenden Platten bildet.
- Alle zugehörigen Platten haben ein einheitliches Plattenformat (K, NK2) und eine einheitliche Allokierungseinheit.
- Jeder Volume-Set besitzt eine sog. **Volres**, in der i.W. die physikalische Konfiguration des Volume-Sets hinterlegt ist.
- Einem Volume-Set kann maximal ein Cache-Bereich zugeschaltet werden.
- Jeder Volume-Set bildet einen Subbehälter für komplette Dateien, d.h. einzelne Dateien können sich nicht über mehrere Volume-Sets erstrecken.
- Ein Volume-Set kann folgende Betriebszustände einnehmen:

DEFECT	der Volume-Set ist defekt
DEFINED-ONLY	der Volume-Set ist nur definiert, aber nicht zugreifbar
IN-HOLD	der Volume-Set ist temporär nicht in Betrieb
NORMAL-USE	der Volume-Set ist in Betrieb und damit zugreifbar
CONTROL-VOLSET	der Volume-Set ist der Control-Volume-Set

- Der Control-Volume-Set enthält:
  - die Pubset-Konfigurationsdatei (Pubset Configuration File)
    (diese Datei enthält eine Liste aller Volume-Sets, die zum SM-Pubset gehören)
  - Spezialkataloge für Jobvariablen, migrierte Dateien, No-Space-Dateien und Dateien auf privaten Datenträgern
  - den Benutzerkatalog
  - den Guards-Katalog

SM-Pubsets mit Hintergrundebenen

Obligatorischer Bestandteil eines SM-Pubsets ist seine Verarbeitungsebene.

Pubsets können mit Hilfe des Softwareprodukts HSMS mit Hintergrundebenen ausgestattet werden. Diese bilden zusammen mit der Verarbeitungsebene bzgl. Zugriffszeit, Verfügbarkeit und Kosten eine dreistufige Speicherhierarchie:

- S0-Ebene (für die Verarbeitungsebene)

- S1-Ebene (für die online verfügbare Hintergrundebene)
- S2-Ebene (für die aus offline zugreifbaren Datenträgern gebildete Hintergrundebene)

Wenn Dateien längere Zeit nicht bearbeitet werden, ist es empfehlenswert, sie auf eine kostengünstigere Hintergrundebene zu verdrängen.
Bei SM-Pubsets wird die S1-Ebene durch einen zum Pubset gehörenden Volume-Set realisiert, der ausschließlich für diesen Zweck reserviert sein muss. Als Datenträger für die S2-Ebene dient ein aus Magnetbändern (oder MBK) bestehender Band-Pool. Die Hintergrundebenen sind einem SM-Pubset exklusiv zugeordnet und können nicht von mehreren Pubsets genutzt werden. Die für die Benutzung der Hintergrundebenen benötigten Metadaten, wie z.B. das Directory des Migrations-Archivs, befinden sich in dem SM-Pubset selbst und liegen dort auf dem Control-Volume-Set.

In folgendem Beispiel besteht der SM-Pubset mit der Katalogkennung XYZ in der Verarbeitungsebene aus drei Volume-Sets.

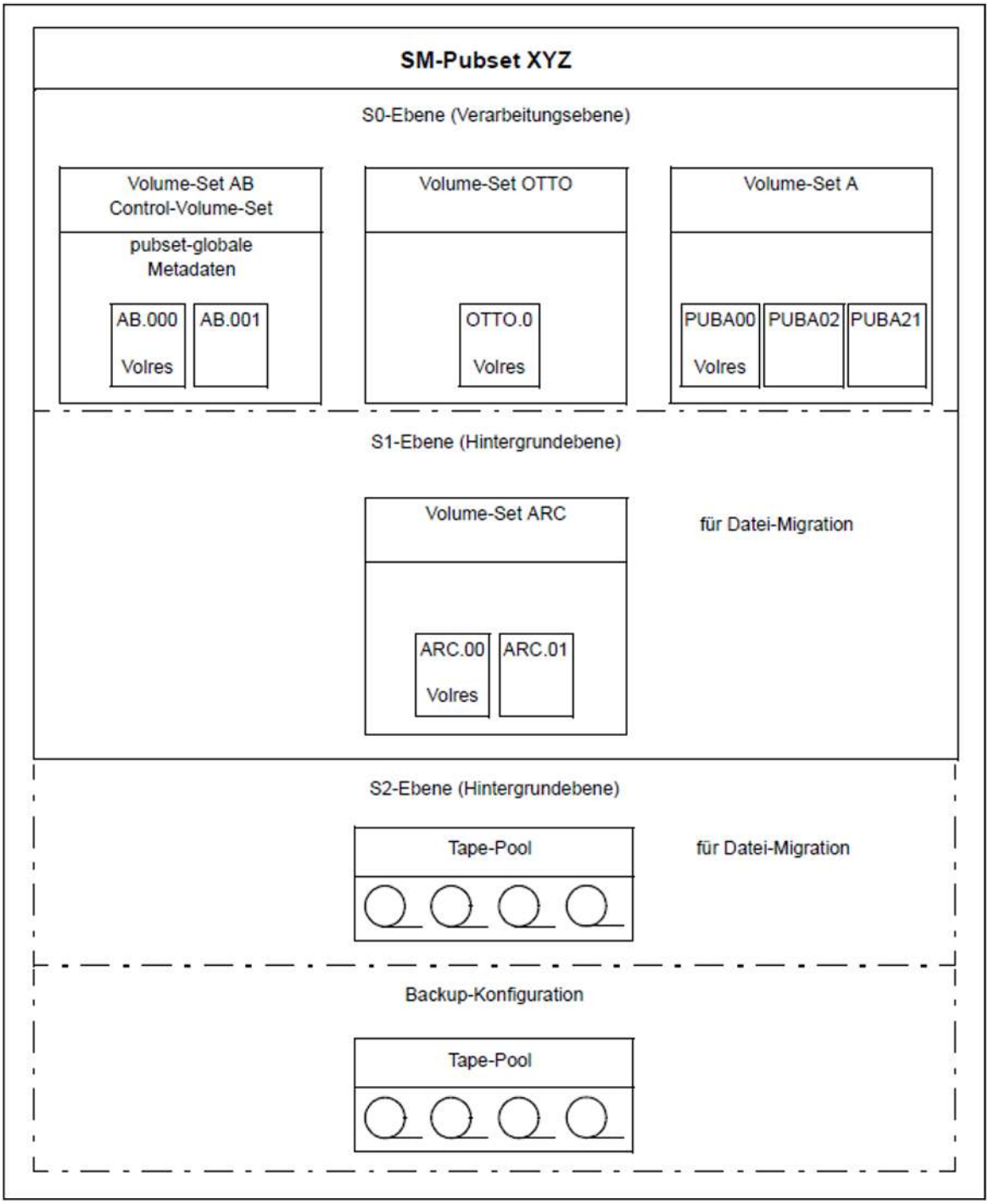


Bild 9: SM-Pubset-Struktur mit HSMS-Konfiguration

9.1.3.3 Pubset-Metadaten

Unter dem Begriff Pubset-Metadaten werden alle Daten verstanden, die letztendlich für den Pubset-Betrieb benötigt werden, also z.B. die physikalischen Konfigurationsdaten, die physikalischen und logischen Eigenschaften sowie die Betriebsparameter (Benutzerkatalog, Benutzerkontingente, usw.).

Neben den im Folgenden beschriebenen Pubset-Metadaten kann es auf SM-Pubsets noch die Systemdateien $TSOS.SYSCAT.STORCLS (Storageklassen-Katalog) und $TSOS.SYSCAT.VSETLST (Volumeset-Listen-Katalog) geben, siehe Handbuch „System-Managed Storage“ [28].

Standard-Volume-Label (SVL) von Pubres und Volres

Im SVL (Standard-Volume-Label) der Pubres eines SF-Pubsets bzw. der Volres eines Volume-Sets sind alle pubset- bzw. volume-set-globalen Informationen hinterlegt.

Der Aufbau des SVL erfolgt in zwei unterschiedlichen Satztypen:

- DMS-Record
  Es sind ein Verweis auf den Dateikatalog TSOSCAT bzw. TSOSCAT.<volume-set-id>, der Wert der Allokierungseinheit, ein Indikator über die normale oder abnormale Beendigung der letzten Pubset-Session, ein Kennzeichen über den Betriebsmodus (lokal oder shared) und Informationen über einen eventuell genutzten Cache hinterlegt.
- Pubset-Record
  Es ist die Plattenkonfiguration (Volume-Katalog) des SF-Pubsets/Volume-Sets hinterlegt. Abhängig von der max. Plattenanzahl ist der Volume-Katalog an unterschiedlichen Stellen hinterlegt:
  - Er ist im Pubset-Record selbst hinterlegt, falls die max. Plattenanzahl 32 beträgt. Der Volume-Katalog im Pubset-Record enthält die VSN, den Volume-Typ der jeweiligen Platte und den Zeitstempel der letzten Betriebsphase des Volumes.
  - Er befindet sich in einem 4KB großen Extrablock auf der Pubres/Volres, wenn die max. Plattenanzahl 255 Platten beträgt. In diesem Fall steht im Pubset-Record lediglich ein Verweis auf diesen Block.

Das Einrichten des gesamten SVL geschieht zum Zeitpunkt der Pubset-Generierung über die Dienstprogramme SIR und VOLIN.

Pubset-Konfigurationsdatei

SF-Pubsets haben keine Pubset-Konfigurationsdatei. Bei SM-Pubsets wird die Pubset-Konfigurationsdatei zum Zeitpunkt der Neugenerierung des SM-Pubsets oder Zusammenführen mehrerer SF-Pubsets zu einem SM-Pubset eingerichtet.

Die Pubset-Konfigurationsdatei ($TSOS.SYS.PUBSET.CONFIG, Pubset-Configuration-File) hat zwei Funktionen:

- Ablage sämtlicher Konfigurationsdaten eines SM-Pubsets
- Ablagemedium für die Rekonfigurationsdaten im Fall eines Rekonfigurationsauftrags für den Pubset

Die Pubset-Konfigurationsdatei ist eine ISAM-Datei und besteht aus drei Satztypen:

- Pubset-Record
  Hier finden sich alle pubset-globalen Daten des SM-Pubset, wie z.B. die Anzahl der Volume-Sets und der Name des Control-Volume-Sets.

- Volume-Set-Record
  Hiervon gibt es der Anzahl der Volume-Sets entsprechend viele Sätze. In diesen sind alle volume-set-spezifischen Betriebsdaten hinterlegt.
- Reconfiguration-Record
  Enthält genau die Informationen, die für eine Fortführung eines unterbrochenen Rekonfigurationsauftrags benötigt werden.

Änderungen der Pubset-Konfigurationsdatei sind nur möglich, wenn der SM-Pubset in Betrieb ist.

Verzeichnis MRSCAT

Der MRSCAT ist das Verzeichnis aller Pubsets, die einem BS2000-System bekannt sind. Er enthält die statischen (permanenten) und dynamischen Betriebsparameter der Pubsets. Gleichzeitig stellt er das Verzeichnis für alle Dateikataloge dar und gibt Auskunft über deren Verfügbarkeit. Der MRSCAT ist eine zentrale Datenstruktur des DVS und stets Ausgangspunkt jeglicher Dateiadressierung.

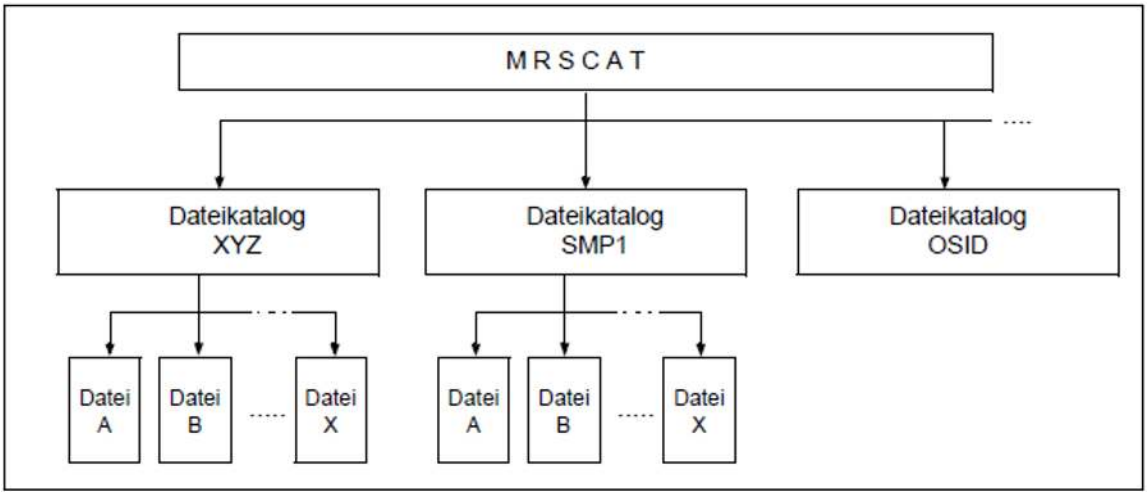


Bild 10: Katalogstruktur MRSCAT – Dateikatalog

Statische Betriebsparameter

Zur permanenten Ablage der Betriebsparameter dienen die beiden ISAM-Dateien $TSOS.SYSTEM.MRSCAT und $TSOS.SYSTEM.MRSCAT.COPY, die auf dem Home-Pubset abgelegt sind und auf konsistentem Stand gehalten werden, um bei einem eventuellen Ausfall einer der beiden Dateien nach wie vor auf gültige Daten zugreifen zu können. Eingerichtet werden die Dateien bei der Systemeinleitung, wenn ein Pubset erstmalig als Home-Pubset genutzt wird.

Zusätzlich werden MRSCAT-Informationen für den Home-Pubset und für Control-Volume-Sets von SM-Pubsets auf der ersten Seite der Katalogdatei TSOSCAT bzw. TSOSCAT.<control-volume-set-id> abgelegt. Dies ist erforderlich, um Deadlock-Situationen bei der Inbetriebnahme zu vermeiden, da ISAM-Dateizugriffe erst möglich sind, wenn der jeweilige Pubset in Betrieb ist. Auf die Katalogdatei kann dagegen bereits sehr frühzeitig mittels CMS-Funktionen zugegriffen werden.

Dynamische Betriebsparameter

Die zweite Funktion des Verzeichnisses MRSCAT ist die performante Ablage aller Pubset-Betriebsparameter während der BS2000-Session im Arbeitsspeicher. Dabei existiert pro Volume-Set, SF-Pubset oder SM-Pubset ein Eintrag im MRSCAT. Als Identifizierer eines MRSCAT-Eintrags dient die Katalogkennung (catid), die der jeweiligen Pubset- bzw. Volume-Set-Kennung entspricht.

9.1.3.4 Benutzerkontingent-Konzept

Die Systembetreuung kann mit Hilfe von Benutzer-Kontingenten den von einzelnen Benutzern auf einem Pubset beanspruchten Platz kontrollieren und einschränken.

Die Kontingente werden primär unterschieden nach Art der Dateien:

- PERM-SPACE für permanente Dateien
- TEMP-SPACE temporäre Dateien
- WORK-SPACE für Arbeitsdateien

Für diese drei Arten kann z.T. noch differenziert werden nach Attributen (Verfügbarkeit, Performance) und Speicherebenen (Verarbeitungsebene, alle Ebenen).

Die Kontingente PERM-SPACE und TEMP-SPACE gibt es bei allen Pubsets, die anderen Kontingente stehen nur für SM-Pubsets zur Verfügung.

Kontingentstruktur auf SM-Pubsets für permanente Dateien

Die einzelnen Kontingente haben folgende Bedeutung:

- TOTAL-SPACE
  Diesem Kontingent wird der Platz für alle auf dem Pubset liegenden permanenten Dateien (eines Benutzers) zugerechnet, unabhängig ob sich diese auf einer Hintergrundebene (S1 oder S2) oder auf der Verarbeitungsebene S0 befinden. Unter Hintergrundebenen sind die durch HSMS genutzten Dateiablageorte zu verstehen. Als Maßeinheit dient hier nicht der real belegte Platz, sondern der Platz, den die Datei nach dem Zurückbringen in die S0-Ebene in dieser belegen würde.
- S0-LEVEL-SPACE
  Gesamtkontingent für permanente Dateien, die sich auf der Verarbeitungsebene S0 befinden.
- HIGH-PERF-SPACE
  Kontingent für Dateien, die einen Performance-Wert *HIGH haben.
- VERY-HIGH-PERF-SPACE
  Kontingent für Dateien, die einen Performance-Wert *VERY-HIGH haben.
- HIGH-AVAILABLE-SPACE
  Kontingent für Dateien, die einen Verfügbarkeitswert *HIGH haben.

In folgenden Bild ist für permanente Dateien der Zusammenhang zwischen Dateiattributen und Kontingenten dargestellt, wobei zu beachten ist, dass die Belastung eines Kontingents implizit auch zu einer Belastung aller übergeordneten Kontingente führt.

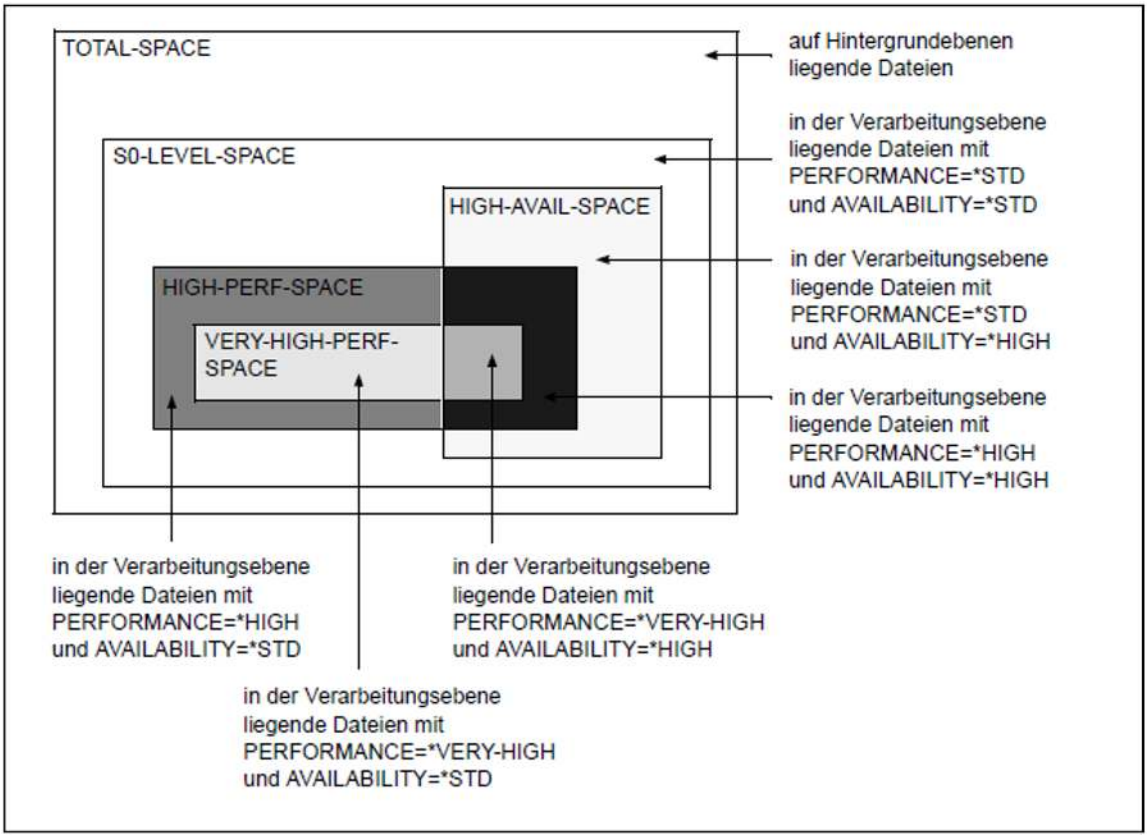


Bild 11: Kontingentstruktur für permanente Dateien in einem SM-Pubset und Zuordnung zu Dateiattributen

Beispiel zu obigem Bild

Das Anlegen einer Datei mit den Attributen AVAILABILITY=*HIGH und PERFORMANCE=*HIGH führt dazu, dass neben den Belegt-Werten der Kontingente für HIGH-AVAIL-SPACE und HIGH-PERF-SPACE auch die Belegt-Werte für S0-LEVEL-SPACE und TOTAL-SPACE um die Anzahl von Half-Pages erhöht werden, welche der Datei zugewiesen sind.

Kontingentstruktur auf SM-Pubsets für temporäre Dateien und Arbeitsdateien

Temporäre Dateien und Arbeitsdateien können nicht auf Hintergrundebenen migriert werden und können nicht das Dateiattibut AVAILABILITY=*HIGH zugewiesen bekommen.
Daher ergibt sich für temporäre Dateien und Arbeitsdateien im Vergleich zu permanenten Dateien eine vereinfachte Kontingentstruktur.

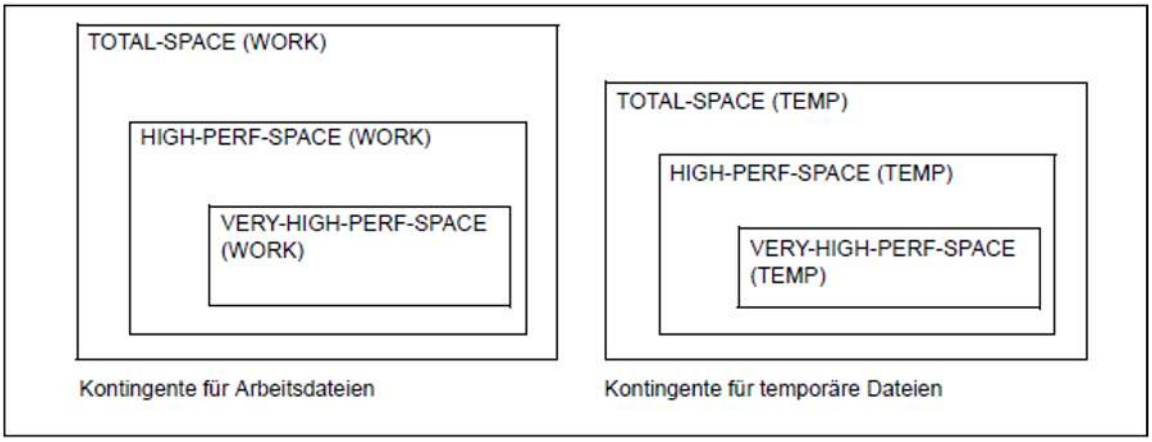


Bild 12: Kontingentstruktur für Arbeitsdateien und temporäre Dateien

Jedem Kontingent ist ein Maximalwert und ein aktueller Belegungswert (Belegt-Wert) zugeordnet. Der aktuelle Belegungswert der Kontingente ergibt sich aus den Dateien eines Benutzers und ihren Dateiattributen. Das System überwacht beim Anlegen und Vergrößern von Dateien, dass die Maximalwerte nicht überschritten werden (sofern keine Sonderberechtigung (PUBLIC-SPACE-EXCESS) vorliegt, welche das Überschreiten des zulässigen Speicherplatzes gestattet).

9.2 Pubset-Eigenschaften

Dieser Abschnitt beschreibt folgende Eigenschaften:

- Physikalische Merkmale
- Globale Eigenschaften
- Benutzerspezifische Eigenschaften
- Volume-Set- und Volume-spezifische Eigenschaften

9.2.1 Physikalische Merkmale

Physikalische Merkmale sind z.B. Plattenformate oder I/O-Eigenschaften.

9.2.1.1 Plattenformate

Die Formatierungseigenschaften eines SF-Pubsets bzw. Volume-Sets werden bereits bei der Initialisierung festgelegt und sind über die gesamte Dauer des Bestehens des SF-Pubsets/Volume-Sets unveränderbar. Alle Volumes (Platten) eines SF-Pubsets/Volume-Sets haben jeweils die gleiche Formatierungseigenschaft. Diese wird von der Systembetreuung bei der Initialisierung der Platten (mit SIR bzw. VOLIN) festgelegt:

- Dateiverarbeitung mit und ohne Nutzung des Pamkeys im Bearbeitungsmodus:
  K und NK
- minimale Allokierungseinheit (min. AU), mit der das DVS Platzzuweisungen für Dateien auf den Platten des Pubsets vornimmt: 6 KB, 8 KB oder 64 KB
- minimale E/A-Transporteinheit (min. TU), in der die Platten-Ein-/Ausgabe erfolgen soll: 2 KB oder 4 KB

Zu den aufgeführten Plattenformaten gibt es die entsprechenden Pubset-Formate, die für importierte Pubsets mit dem Kommando SHOW-PUBSET-CONFIGURATION abgefragt werden können.

Folgende Kombinationen für Plattenformate sind generell möglich:

Plattenformat	Bearbeitungsmodus	min. AU (in KB)	min. TU (in KB)	max. Plattenanzahl pro SF-Pubset/Volume-Set
K-Format	K	6	2	32
NK2-Format	NK	6	2	32
NK2-Format	NK	8	2	255
NK2-Format	NK	64	2	255

9.2.1.2 Physikalische Merkmale von Volume-Sets

Die Systembetreuung muss dafür sorgen, dass die Eigenschaftsprofile der Volume-Sets durch die physikalische Volume-Set-Konfiguration erfüllt werden.

Physikalische Eigenschaften der einzelnen Volumes

Die physikalischen Eigenschaften eines Volume-Sets sind durch die Eigenschaften der ihm zugehörigen Volumes sowie deren I/O-Konfiguration (Controller, Zugangspfade, usw.) bestimmt. Damit den Volume-Sets sinnvolle Eigenschaftsprofile zugeordnet werden können, sollte die Volume-Konfiguration der Volume-Sets homogen sein, d. h. die Systembetreuung sollte nur Volumes zu einem Volume-Set zusammenfassen, deren Eigenschaften mit dem vorgesehenen Eigenschaftsprofil harmonieren. Bei bestimmten Eigenschaften wird die Homogenität des Volume-Sets durch das System erzwungen (z.B. Größe der Allokierungseinheit, Format).

Maximale Ein-/Ausgabe-Länge

Die maximale Ein-/Ausgabe-Länge ist die maximale Anzahl von Blöcken, die bei einem I/O-Auftrag übertragen werden können. Sie wird durch den Gerätetyp bestimmt. Der kleinste Wert aller einem Volume-Set angehörenden Volumes bestimmt die maximale Ein-/Ausgabe-Länge des Volume-Sets.
Wird in einen Volume-Set ein weiteres Volume hinzugenommen, kann sich also u.U. die maximale Ein-/Ausgabe-Länge des Volume-Sets verkleinern. Die maximale Ein-/Ausgabe-Länge eines Volume-Sets wird durch bestimmte privilegierte Anwendungen (ARCHIVE, CCOPY) zur Optimierung der Ein-/Ausgaben herangezogen.

Mit dem Kommando SHOW-PUBSET-CONFIGURATION kann die maximale Ein-/Ausgabe-Länge eines Volume-Sets ausgegeben werden.

Volume-Sets als unabhängige Ausfalleinheiten

Volume-Sets eines SM-Pubsets können (mit Ausnahme des Control-Volume-Sets) als voneinander unabhängige Ausfalleinheiten eingerichtet werden.

Dazu müssen diese Volume-Sets auf physikalische Konfigurationen abgebildet werden, die hinsichtlich Ausfall unabhängig voneinander sind. Beispielsweise müssen die Zugangspfade (Kanäle, Controller, usw.) zu den Volumes von Volume-Sets, die bezüglich Ausfallsicherheit unabhängig sein sollen, voneinander entkoppelt sein. Auch die den Volume-Sets zugeordneten Cache-Konfigurationen sind zu berücksichtigen.

9.2.2 Globale Eigenschaften

Pubset-globale Einstellungen dienen vorwiegend zur Steuerung des Pubset-Betriebs:

- CMS-Einstellungen
  Sie bestimmen im Wesentlichen Anzahl und Lage der CMS-Puffer, die Einfluss auf den Betrieb der Katalogverwaltung nehmen.

- Betriebsmodus, d.h. ein Pubset kann genutzt werden:
  - im exklusiven Zugriffsmodus von einem System
  - als Shared-Pubset von mehreren Systemen gleichzeitig, siehe "Shared-Pubsets"
  - exklusiv von einem Benutzer. In diesem Fall dürfen nur diese Kennung und die Systembetreuung unter der Kennung TSOS auf diesen Pubset zugreifen.

- Standard-Einstellungen für Pubset-Space und Dateiformat
  - Für SM- und SF-Pubsets kann der Standardwert für die Größe von Primär-, Sekundär- und Maximalallokierung der Dateien gesteuert werden.
  - Für einen SM-Pubset kann der Standardwert für das Dateiformat gesteuert werden.

9.2.3 Benutzerspezifische Eigenschaften

Die Systembetreuung legt folgende benutzerspezifischen Eigenschaften fest:

- Standard-Katalogkennung (Standard-Pubset, Default Pubset)
  Jedem Benutzer wird ein eine Standard-Katalogkennung zugewiesen. Diese wird auch Standard-Pubset oder Default Pubset genannt. Auf diesem Pubset kann er (ohne Angabe der Katalogkennung) Dateien und Jobvariablen erzeugen, verarbeiten und löschen.

- Speicherplatz-Limit
  Auf dem Standard-Pubset legt die Systembetreuung für die Benutzer das Speicherplatz-Limit fest.

- Pubset-spezifische Zugriffsrechte
  Dazu gehören z.B. das Recht, Tuning-Ressourcen für die Dateibearbeitung zu nutzen, Speicherplatz direkt zu allokieren, das Kontingent-Limit zu überschreiten oder Net-Storage zu nutzen.

9.2.4 Volume-Set- und Volume-spezifische Eigenschaften

Bei Volume-Sets von SM-Pubsets und einzelne Volumes lassen sich folgende Eigenschaften festlegen:

- Verfügbarkeit und Performance
  Erhöhte Verfügbarkeit stellt die Systembetreuung, z.B. durch Einsatz von RAID-Systemen oder durch Spiegelung sicher. Das Performance-Profil wird durch den Einsatz von Caches gesteuert.

- Nutzungsarten:
  - Normalfall (STD), bei Control-Volume-Sets die einzig mögliche Nutzungsart.
  - Arbeitsdateien (WORK).
  - HSMS-Nutzung auf S1-Ebene (HSMS-CONTROLLED)

- Nutzungseinschränkungen
  - Bei Volume-Sets gibt es Nutzungseinschränkungen für das Neuanlegen von Dateien und das Bearbeiten von Dateien.
  - Für die einzelnen Volumes eines SF-Pubsets oder eines Volume-Sets können Allokierungseinschränkungen erlassen werden.

- Sättigungsschwellwerte für SF-Pubsets und Volume-Sets
  Es können mehrere Schwellwerte definiert werden. Beim Erreichen von Schwellwerten werden Konsolmeldungen ausgegeben, bei höheren Schwellwerten werden Aufträge ggf. zurückgewiesen.

- Datei-Services
  - Storage-Services (z.B. gebotene Performance, Verfügbarkeit), die mit Hilfe von Storage-Klassen und Volume-Set-Listen gesteuert werden.
  - HSMS-Management-Services (z.B. Steuerung der Dateiverdrängung auf Hintergrundebenen, Steuerung der Erstellung von Dateisicherungen), die mit Hilfe von HSMS gesteuert werden.

9.3 Pubsets einrichten

Zum Einrichten von Pubsets werden verschiedene Dienstprogramme verwendet. Alle im Folgenden erwähnten Diensprogramme sind im Handbuch „Dienstprogramme“ [15] beschrieben.

9.3.1 SF-Pubsets einrichten

Einrichten und Erweitern eines SF-Pubsets

SF-Pubsets einschließlich Home-Pubset werden mit dem Dienstprogramm SIR eingerichtet und erweitert.

Alternativ zum Erweitern von Pubsets mit SIR besteht die Möglichkeit, die neuen Platten mittels Dienstprogramm VOLIN mit der entsprechenden VSN und dem entsprechenden Format zu initialisieren. Beim nächsten Import des Pubsets mit IMPORT-PUBSET (nicht beim impliziten Import während des Startup) fragt das System, ob die neuen Platten zum Pubset hinzugenommen werden sollen.

9.3.2 SM-Pubsets einrichten und ersetzen

SM-Pubsets können durch komplettes Neueinrichten oder durch Umwandeln von bereits existierenden SF-Pubsets in SM-Pubsets eingerichtet werden.

9.3.2.1 Einrichten und Erweitern von Volume-Sets und SM-Pubsets

SM-Pubsets und Volume-Sets werden mit dem Dienstprogramm SIR eingerichtet.

SM-Pubsets und die zugehörigen Volume-Sets können zusammen in einem SIR-Lauf eingerichtet werden. Beides kann aber auch in verschiedenen SIR-Läufen erfolgen. SIR-Aktionen, die einen importierten Pubset erfordern (z.B. CREATE-PAGING-FILE), können in diesem Fall für einen einzelnen Volume-Set nicht ausgeführt werden.

9.3.2.2 SF-Pubsets in SM-Pubsets umwandeln

Damit Benutzer SM-Pubsets für einen bereits vorhandenen Dateibestand nutzen können, müssen bestehende SF-Pubsets in SM-Pubsets umgewandelt werden. Dabei können mehrere SF-Pubsets zu einem SM-Pubset zusammengefasst werden. Dabei sind folgende grundsätzlichen Aspekte zu beachten:

- Die Kennung des Pubsets ist als Catid Bestandteil des Pfadnamens der im Pubset katalogisierten Objekte. Geht ein SF-Pubset in einen SM-Pubset ein, dessen Kennung sich von der des SF-Pubsets unterscheidet, ändert sich die Adressierung der Objekte.
- Befinden sich auf verschiedenen SF-Pubsets, die in den SM-Pubset eingehen, Objekte, deren Pfadnamen sich lediglich durch die Catid unterscheiden, so ergäben sich bei der Bildung des SM-Pubsets Namenskollisionen, d. h. die Pfadnamen wären nach der Bildung des SM-Pubsets nicht mehr eindeutig. Benutzer, für welche sich nur auf einem der in den SM-Pubset eingehenden SF-Pubsets Dateien befinden, können davon nicht betroffen sein, da die Benutzerkennung Bestandteil des Pfadnamens ist.
- Geht ein SF-Pubset in einen SM-Pubset ein, dessen Kennung mehr Zeichen enthält als die des SF-Pubsets, verlängern sich die Pfadnamen der auf dem SF-Pubset liegenden Objekte. Dabei kann sich eine Pfadnamenüberlänge ergeben.

In die Umwandlung von SF-Pubsets in einen SM-Pubset sind die Verarbeitungsebene, Hintergrundebenen und Backup-Archive einzubeziehen. Eine detaillierte Beschreibung ist im Handbuch „SMS“ [28] zu finden.

Im Folgenden wird nur die Vorgehensweise bei der Verarbeitungsebene skizziert. Dabei gibt es zwei Möglichkeiten: Umwandeln mit SMPGEN oder Sichern und Wiedereinspielen.

Umwandeln mit SMPGEN (In-Place-Konvertierung)

Die In-Place-Konvertierung der Verarbeitungsebene wird durch das Dienstprogramm SMPGEN ermöglicht. Es gestattet, aus einem oder mehreren SF-Pubsets einen SM-Pubset zu bilden, wobei die SF-Pubsets jeweils in einen Volume-Set überführt werden. Die auf den SF-Pubsets liegenden Benutzerdateien bleiben bei der Zusammenführung unverändert erhalten, die Metadaten-Dateien der SF-Pubsets (wie Dateikataloge, Benutzerkataloge, Guards-Kataloge usw.) werden in entsprechende Metadaten-Dateien des SM-Pubsets umgewandelt.
Bestimmte Pubset- und Volume-Set-Merkmale, Benutzerkontingente, usw. werden für den SM-Pubset durch SMPGEN automatisch ermittelt. Falls die Systembetreuung davon abweichende Einstellungen wünscht, muss er den SM-Pubset nach der Generierung entsprechend nachbehandeln.

Zur In-Place-Konvertierung bietet SMPGEN die folgenden Anweisungen an:

- CREATE-SYSTEM-MANAGED-PUBSET für Prüfung und Konvertierung in einen neu zu erstellenden SM-Pubset
- MODIFY-SYSTEM-MANAGED-PUBSET für Prüfung und Aufnahme zusätzlicher SF-Pubsets in einen bereits existierenden SM-Pubset

Umwandeln durch Sichern und Wiedereinspielen von Dateien

Der Übergang von SF-Pubsets auf SM-Pubsets durch Sichern und Wiedereinspielen von Dateien ist vor allem dann zu empfehlen, wenn für den SM-Pubset eine andere interne Struktur gewünscht wird, als sie sich durch In-Place-Konvertierung ergeben würde. Dies ist zum Beispiel der Fall, wenn ein SF-Pubset mit vielen Volumes in einen SM-Pubset mit mehreren Volume-Sets übergeführt werden soll.

Es erfolgt eine Vollsicherung der SF-Pubsets, die in den SM-Pubset eingehen (ggf. ohne die auf Hintergrundebenen liegenden Dateien). Anschließend wird mit SIR der SM-Pubset erzeugt. Auf dem SM-Pubset werden für die bisherigen Benutzer der SF-Pubsets Benutzereinträge angelegt. Dies wird durch HSMS unterstützt, indem Sicherungen von Benutzerkatalogen von SF-Pubsets auf SM-Pubsets wieder eingespielt werden können. Dabei werden die für SM-Pubsets eingeführten Benutzerattribute (z.B. Benutzerkontingente für hochverfügbare Dateien) so voreingestellt, dass eine möglichst große Kompatibilität gegeben ist. Anschließend wird der Guards-Katalog eingerichtet, wobei auf SF-Pubsets erstellte Sicherungen benutzt werden können. Schließlich werden die Benutzerdateien wieder eingespielt.

9.3.3 Pubsets mit PVSREN erzeugen

Das Dienstprogramm PVSREN kann eigenständige Pubsets aus den Platten-Kopien eines SM- bzw. SF-Pubsets (Pubset-Kopie) innerhalb eines Systems erzeugen.

PVSREN nutzt die lokalen Replikationsfunktionen der Plattenspeichersysteme zum Erstellen von eigenständigen Pubsets, indem Pubset-Kopien umbenannt werden. Bei SM-Pubsets werden die Umbenennungsregeln für die Volume-Sets in einer Parameterdatei abgelegt, in der für jeden Volume-Set durch die Anweisung SET-NAME-OF-NEW-VOLUME-SET die neue Katalogkennung festgelegt wird.

Mit der Anweisung CREATE-PUBSET-FROM-MIRROR wird ein eigenständiger Pubset erstellt. Dabei bestimmt der Operand MIRRORING-METHOD, ob bzw. mit welcher Funktion PVSREN eine Kopie erzeugt:

- Die Angabe von MIRRORING-METHOD=*SPLIT-MIRROR setzt voraus, dass die Pubset-Kopien bereits abgetrennt sind und in Doppelpunkt-Notation (siehe "Doppelpunkt-Notation für Spiegelplatten") vorliegen. PVSREN führt in diesem Fall nur die Umbenennung zu einem eigenständigen Pubset durch.
- Mit MIRRORING-METHOD=*CLONE(...)/*SNAP(...) erzeugt PVSREN eine Kopie auf Basis von Clone-Units oder Snap-Units und benennt diese zu einem eigenständigen Pubset um.

9.4 Pubsets konfigurieren

Bevor ein Pubset in Betrieb genommen und damit genutzt werden kann, muss er konfiguriert werden, d.h.:

- Pubset in MRSCAT eintragen
- Globale Eigenschaften festlegen
- Benutzerspezifische Eigenschaften festlegen
- Volume-spezifische Eigenschaften festlegen (bei SM-Pubsets)

9.4.1 MRSCAT-Einträge verwalten

Die max. Anzahl von MRSCAT-Einträgen ist begrenzt und kann zur Optimierung der Zugriffs-Performance den Wert 128, 256, 512, 1024, 2048 oder 4096 annehmen. Mit dem Systemparameter DMCMAXP kann diese max. Anzahl festgelegt werden.

Die Systembetreuung kann mit folgenden Kommandos MRSCAT-Einträge einrichten, modifizieren und löschen:

- ADD-MASTER-CATALOG-ENTRY
  Neueinrichten von MRSCAT-Einträgen mit gleichzeitigem Initialisieren der statischen Pubset-Parameter mit Ausnahme der Cache-Konfiguration.
- MODIFY-MASTER-CATALOG-ENTRY (EDIT-MASTER-CATALOG-ENTRY)
  Änderung der statischen Pubset-Parameter mit Ausnahme der Cache-Konfiguration.
- REMOVE-MASTER-CATALOG-ENTRY
  Löschen von MRSCAT-Einträgen.

9.4.2 Globale Pubset-Eigenschaften festlegen

CMS-Einstellungen

Angaben über die CMS-Puffer können mit Hilfe der Operanden RESIDENT-BUFFERS und NUMBER-OF-BUFFERS der Kommandos ADD-/MODIFY-MASTER-CATALOG-ENTRY oder IMPORT-PUBSET gemacht werden. Diese Angaben werden erst beim Importieren des Pubsets (Kommando IMPORT-PUBSET) nach folgender Hierarchie ausgewertet:

1. Explizite Parameterangabe im Kommando IMPORT-PUBSET
2. Angaben über das Kommando ADD- bzw. MODIFY-MASTER-CATALOG-ENTRY.
   Wird nur einer der Parameter RESIDENT-BUFFERS oder NUMBER-OF-BUFFERS angegeben, gilt für den anderen jeweils der Standardwert
   (RESIDENT-BUFFERS=*NO, NUMBER-OF-BUFFERS=32) .
3. Vereinbarungen laut Systemparameter CATBUFR und BMTNUM

Betriebsmodi

Die Betriebsmodi können über die Kommandos ADD-/MODIFY-MASTER-CATALOG-ENTRY durch folgende Operanden eingestellt werden:

- ACCESS-CONTROLLED: exklusive Nutzung durch einen Benutzerkennung
- SHARED-PUBSET: Zugriff durch mehrere Systeme (Shared-Pubset)

Die Eigenschaft Shared-Pubset kann auch beim Importieren des Pubsets (Kommando IMPORT-PUBSET, Operand USE) festgelegt werden.

Standard-Einstellungen für Pubset-Space und Dateiformat

Für SM- und SF-Pubsets kann der Standardwert für die Größe von Primär-, Sekundär- und Maximalallokierung der Dateien gesteuert werden (Kommando MODIFY-PUBSET-SPACE-DEFAULTS (EDIT-PUBSET-SPACE-DEFAULTS)).

Für einen SM-Pubset kann mit dem Kommando MODIFY-PUBSET-SPACE-DEFAULTS ...,PUBSET-TYPE=*S-M (FILE-FORMAT=*K/*NK2/*NK4) der Standardwert für das Dateiformat gesteuert werden.

9.4.3 Benutzer-spezifische Einstellungen festlegen

Die Systembetreuung legt die Standard-Katalogkennung (Standard-Pubset) eines Benutzers fest und definiert für das Standard-Pubset verschiedene Grenzwerte (Speicherplatz, Anzahl Jobvariablen und Dateien) und Pubset-spezifische Zugriffsrechte.

9.4.3.1 Standard-Katalogkennung (Standard-Pubset) zuordnen

Jedem Benutzer weist die Systembetreuung mit dem Kommando ADD-USER (Operand DEFAULT-PUBSET) eine Standard-Katalogkennung zu, die den Pubset bestimmt, auf den zugegriffen wird, wenn der Benutzer beim Zugriff auf Dateien oder Jobvariablen keine Katalogkennung angibt.

Die Standard-Katalogkennungen werden im Benutzerkatalog des Home-Pubsets hinterlegt.

> ¡ Wenn die Systembetreuung eine neue Benutzerkennung einführt, die Zugriff zu mehreren Pubsets haben soll, so muss sie für jeden dieser Pubsets ein entsprechendes ADD-USER-Kommando geben. Das gilt analog für das Modifizieren und Löschen von Benutzereinträgen.

9.4.3.2 Grenzwerte und Zugriffsrechte festlegen

Die Systembetreuung legt benutzerspezifische Grenzwerte und Zugriffsrechte für ein Pubsets mit den Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES fest.

Operanden für Grenzwerte :

- PUBLIC-SPACE-LIMIT und PUBLIC-SPACE-EXCESS

  Bestimmt den maximalen Speicherplatz, den der Benutzer auf dem Pubset verbrauchen darf. PUBLIC-SPACE-EXCESS gibt an, ob dieses Limit u.U. überschritten werden darf.
- TEMP-SPACE-LIMIT

  Bestimmt den maximalen Speicherplatz, den der Benutzer auf dem Pubset für temporäre Dateien verbrauchen darf.
- JV-NUMBER-LIMIT

  Maximale Anzahl der Job-Variablen, die der Benutzer anlegen darf.
- FILE-NUMBER-LIMIT

  Maximale Anzahl der Dateien, die der Benutzer anlegen darf.

Operanden für Zugriffsrechte:

- DMS-TUNING-RESOURCES
  Dieses Recht bestimmt, ob es einem Benutzer gestattet ist, Tuning-Ressourcen für die Dateibearbeitung zu nutzen. Tuning-Ressourcen sind die Pubset-Caches und hochwertige Speicherklassen.
- PHYSICAL-ALLOCATION
  Bestimmt, ob ein Benutzer Plattenspeicherplatz direkt unter Angabe von Volume-Set, Volume oder Block anfordern darf.
- NET-STORAGE-USAGE
  Dieses Recht erlaubt oder verbietet dem Benutzer die Nutzung von Net-Storage.
  Der belegte Speicherplatz auf einem Net-Storage-Volume wird nicht auf das Public-Space-Limit des Benutzers angerechnet.

Weitere Hinweise zur Verwaltung des Benutzerkatalogs siehe Abschnitt „Zugriffsberechtigung und Benutzerkatalog verwalten"

9.4.4 Eigenschaften eines Volume-Sets festlegen

- Verfügbarkeit definieren
- Performance-Profil definieren
- Nutzungsart festlegen
- Nutzungseinschränkungen
- Sättigungsschwellwerte für Volume-Sets und SF-Pubsets festlegen
- Datei-Services definieren
- Benutzerkontingente festlegen

9.4.4.1 Verfügbarkeit definieren

Die Verfügbarkeit eines Volume-Sets definiert die Systembetreuung durch das Attribut AVAILABILITY im Kommando MODIFY-PUBSET-DEFINITION-FILE. Mögliche Werte sind STD und HIGH. Das System überprüft dabei nicht, ob die tatsächliche Eigenschaft des Volume-Sets den zugeordneten Werten entspricht. Es obliegt der Systembetreuung, der Realität entsprechende Werte zu wählen.

Der von der Systembetreuung zugewiesene Verfügbarkeitswert entspricht der langfristigen Positionierung, die dem Volume-Set innerhalb des SM-Pubsets zugedacht ist. Eine Änderung des AVAILABILITY-Werts von HIGH zu STD ist mit erheblichem organisatorischem Aufwand verbunden, da zuvor insbesondere alle auf dem Volume-Set liegenden Dateien mit dem Dateiattribut AVAILABILITY=*HIGH auf einen anderen hoch-verfügbaren Volume-Set gebracht werden müssen.

Erhöhte Verfügbarkeit eines Volume-Sets lässt sich z.B. durch den Einsatz eines geeigneten RAID-Levels im Plattenspeichersystem oder durch Volume-Spiegelung erreichen.

9.4.4.2 Performance-Profil definieren

Das Performance-Profil eines Volume-Sets definiert die Systembetreuung durch den Operanden PERFORMANCE-ATTR im Kommando MODIFY-PUBSET-DEFINITION-FILE. Es obliegt der Systembetreuung, das Performance-Profil der Volume-Sets so festzulegen, dass es die realen Verhältnisse richtig beschreibt. Mit PERFORMANCE-ATTR wird ein Performance-Spektrum (PERFORMANCE=) und die Spezifikation von Einschränkungen bzgl. der Schreibsicherheit (WRITE-CONSISTENCY=) festgelegt.

- Performance-Spektrum:
  Ein mit einem Cache ausgestatteter Volume-Set kann i.A. ein Spektrum unterschiedlicher Performance-Anforderungen von seiten der Benutzer abdecken (STD, HIGH, VERY-HIGH). Er ist damit gleichzeitig als Ablageort für Dateien mit der Anforderung PERFORMANCE=*STD, *HIGH oder *VERY-HIGH geeignet. Die unterschiedlichen Performance-Werte korrespondieren mit verschiedenen Arten der Cache-Nutzung, welche bei der Bearbeitung einer Datei möglich sind:

  PERFORMANCE=*STD: keine Cache-Nutzung

  PERFORMANCE=*HIGH: Caching mit Verdrängung

  PERFORMANCE=*VERY-HIGH: Caching ohne Verdrängung

  Die Art der Cache-Nutzung kann individuell den spezifischen Performance-Anforderungen der einzelnen Dateien angepasst werden.

  > ¡ Die gleichzeitige Zuordnung mehrerer unterschiedlicher Caches zu einem Volume-Set ist nicht möglich.

- Schreibsicherheit:
  Die Cache-Medien unterscheiden sich im Hinblick auf die Schreibsicherheit (WRITE-CONSISTENCY=*BY-CLOSE/*IMMEDIATE).
  Der Wert WRITE-CONSISTENCY=*IMMEDIATE bedeutet, dass auf dem Volume-Set erhöhte Performance uneingeschränkt geboten werden kann, WRITE-CONSISTENCY= *BY-CLOSE bringt zum Ausdruck, dass erhöhte Performance bei Schreiboperationen nur dann genutzt werden kann, wenn eine Datenkonsistenz zum CLOSE-Zeitpunkt ausreicht und dies auch so angefordert wird.
  Ein Cache gilt dann als schreibsicher, wenn für Ausgabedaten, die im Cache zwischengepuffert werden, kein höheres Ausfallrisiko besteht, als wenn sie unmittelbar auf Platte geschrieben würden.
  Nicht-schreibsichere Caches dürfen generell für die Performance-Verbesserung bei lesenden Dateizugriffen verwendet werden. Für Schreibzugriffe sind sie aber nur dann nutzbar, wenn der Benutzer Einschränkungen hinsichtlich Schreibsicherheit akzeptiert und dies durch das Dateiattribut DISK-WRITE=*BY-CLOSE zum Ausdruck bringt.

Empfehlungen für das Performance-Profil

Die in folgender Tabelle vorgeschlagenen Einstellungen für die verschiedenen Cache-Medien können folglich hinsichtlich des Performance-Spektrums nur als Orientierungshilfe dienen, da sie die Gegebenheiten konkreter Konfigurationen nicht berücksichtigen. Für die Festlegung der Schreibsicherheit bei erhöhter Performance sind hingegen Empfehlungen möglich, die in jeder Konfiguration anwendbar sind.

Volume-Set-Konfiguration	empfohlenes Performance-Profil für Performance-Spektrum (PERFORMANCE)	empfohlenes Performance-Profil für Schreibsicherheit (WRITE-CONSISTENCY) bei erhöhter Performance (PERF=HIGH/VERY-HIGH)
Volume-Set, der aus Volumes ohne besondere Performance-Eigenschaften besteht und dem kein Cache zugeordnet ist	STD	ohne Bedeutung
Volume-Set, dem ein Hauptspeicher als Cache-Medium zugeordnet ist	STD, HIGH, VERY-HIGH	nicht gegeben (BY-CLOSE)
Volume-Set, dessen Volumes an eine Cache-Steuerung angeschlossen sind	STD, HIGH	gegeben (IMMEDIATE)

Tabelle 20: Empfehlungen für Performance-Profile

9.4.4.3 Nutzungsart festlegen

Mit Hilfe des Kommandos MODIFY-PUBSET-DEFINITION-FILE, Operand USAGE, kann die Nutzungsart für die Volume-Sets eines SM-Pubsets festgelegt werden:

- STD
  Normalfall. Für den Control-Volume-Set ist STD die einzig mögliche Nutzungsart.
- WORK
  Nur Volume-Sets mit dieser Nutzungsart dienen als Ablageort für Arbeitsdateien.
- HSMS-CONTROLLED
  Volume-Sets mit dieser Nutzungsart werden benötigt, um mit Hilfe von HSMS die S1-Ebene einrichten zu können. Sie dienen ausschließlich diesem Zweck.

9.4.4.4 Nutzungseinschränkungen

Die Nutzungseinschränkungen sind im Gegensatz zur Nutzungsart auch kurzfristig änderbar. Es wird in Einschränkungen für das Neuanlegen von Dateien, für das Bearbeiten von Dateien und für das temporäre Stillegen eines Volume-Sets unterschieden.

Nutzungseinschränkungen werden mit dem Kommando MODIFY-PUBSET-RESTRICTIONS in folgenden Operanden festgelegt:

- NEW-FILE-ALLOCATION
  Einschränkungen für das Neuanlegen von Dateien.
  Möglich sind nur physikalische Allkolierung oder generelle Sperre.

  Das Zurücknehmen einer Einschränkung ist nur zulässig, wenn der Zugang zum Volume-Set nicht eingeschränkt ist.
- VOLUME-SET-ACCESS
  Einschränkungen für das Bearbeiten von Dateien auf TSOS.
- ALLOCATION-ON-VOLUME
  Nutzungseinschränkungen für einzelne Volumes.
  Mögliche sind nur physikalische Allokierung oder generelle Sperre.

9.4.4.5 Sättigungsschwellwerte für Volume-Sets und SF-Pubsets festlegen

Den einzelnen SF-Pubsets bzw. Volume-Sets eines SM-Pubsets sind durch die Systembetreuung festlegbare Schwellwerte für die Überwachung von Speicherengpässen zugeordnet. Der höchste überschrittene Sättigungs-Level bestimmt den aktuellen Sättigungszustand, siehe dazu auch den Abschnitt „Speicherplatz-Sättigung überwachen".

Die Einstellung der Schwellwerte für die einzelnen Sättigungsgrade erfolgt mit dem Kommando SET-/MODIFY-SPACE-SATURATION-LEVEL. Mit diesem Kommando können die Sättigungsstufen sowohl kurzfristig als auch dauerhaft geändert werden (Operand SCOPE=*PERMANENT/*TEMPORARY/*NEXT-PUBSET-SESSION).

Mit dem Kommando MODIFY-SPACE-SATURATION-LEVEL kann auch die System-Notreserve an Plattenspeicherplatz für einen Systemstart im Modus ZIP pubset-spezifisch eingestellt werden (Operand ZIP-LEVEL). Diese wird wie eine 6. Sättigungsstufe behandelt.

Für Pubsets bzw. Volume-Sets, für die keine expliziten Festlegungen erfolgt sind, wird die systemglobale Voreinstellung für die Sättigungsstufe 4 wirksam, die mit dem Systemparameter L4SPDEF getroffen worden ist.

Die Einstellungen der Sättigungsstufen können mit dem Kommando SHOW-SPACE-SATURATION-LEVELS für SF-Pubsets bzw. mit Operand VOLUME-SET=*ALL/<cat-id> für die Volume-Sets eines SM-Pubsets abgefragt werden.

Eine eingeschränkte Einstellmöglichkeit für die Sättigungsstufe 4 nur für SF-Pubsets bieten aus Kompatibilitätsgründen auch die Kommandos ADD- bzw. MODIFY-MASTER-CATALOG-ENTRY.

9.4.4.6 Datei-Services definieren

Aus Benutzersicht erscheint ein SM-Pubset als ein Dateibehälter, der bestimmte Services für die Dateiverwaltung bereitstellt. Diese werden als Datei-Services bezeichnet. Sie lassen sich unterteilen in

- Storage-Services (z.B. gebotene Performance, Verfügbarkeit)
- HSMS-Management-Services (z.B. Steuerung der Dateiverdrängung auf Hintergrundebenen, Steuerung der Erstellung von Dateisicherungen) .

Das Anfordern der gewünschten Datei-Services durch die Benutzer erfolgt mit Hilfe von Dateiattributen, für welche den Dateien passende Werte zugewiesen werden müssen.

Storage-Services

Die Systembetreuung kann Storage-Klassen einrichten, welche für die ablageort-relevanten Dateiattribute eine bestimmte Kombination von Attributwerten repräsentieren
(siehe Kommandos CREATE-STORAGE-CLASS und MODIFY-STORAGE-CLASS (EDIT-STORAGE-CLASS)).

Der Benutzer wählt für eine Datei die Storage-Klasse aus, welche den jeweiligen Anforderungen am besten entspricht (siehe Kommandos CREATE-FILE oder MODIFY-FILE-ATTRIBUTES, Operand STORAGE-CLASS). Er muss die Werte nicht einzeln angeben.

Der vom Benutzer für eine Datei gewünschte Storage-Service wird berücksichtigt, wenn:

- eine Datei neu angelegt wird
- eine Datei von einer Hintergrundebene in die Verarbeitungsebene zurückgeholt wird
- der Benutzer für eine bestehende Datei die ablageort-relevanten Dateiattribute so ändert, dass sie mit dem bisherigen Ablageort unverträglich sind

Dabei ermittelt das System automatisch den am besten geeigneten Volume-Set.

Die Systembetreuung kann für jeden Benutzer eine Standard-Storage-Klasse festlegen.
Diese kann Direktattribute unwirksam machen, wenn kein Recht auf physikalische Attributierung vorliegt.

Volume-Set-Listen

Das Systemverhalten kann durch Volume-Set-Listen beeinflusst werden, die die Systembetreuung einrichtet und mit Storage-Klassen verknüpft (siehe Kommandos CREATE-VOLUME-SET-LIST und CREATE-STORAGE-CLASS, Operand VOLUME-SET-LIST).

Ordnet ein Benutzer einer Datei eine Storage-Klasse zu, der eine Volume-Set-Liste zugewiesen ist, wird die Datei durch das System bevorzugt auf einem Volume-Set aus dieser Liste abgelegt. Belegt die Datei bereits Speicherplatz auf einem anderen Volume-Set, wird sie - wenn möglich - von diesem auf einen der bevorzugten Volume-Sets verlagert.

Dateien, denen keine Storage-Klasse oder eine Storage-Klasse ohne Volume-Set-Liste zugewiesen ist, werden bevorzugt auf solchen Volume-Sets abgelegt, die keiner der von der Systembetreuung eingerichteten Volume-Set-Listen angehören.

HSMS-Management-Services

Das Anfordern der Management-Services erfolgt mit Hilfe von HSMS-Management-Klassen. Sie müssen von der Systembetreuung eingerichtet werden (HSMS-Anweisung CREATE-MANAGEMENT-CLASS). Die HSMS-Management-Klassen repräsentieren bestimmte Verfahren für Dateisicherung (z.B. Sicherungshäufigkeit, Lebensdauer der Sicherungsversionen, usw.) sowie Regeln, welche die Verdrängung auf die Hintergrundebenen steuern (z.B. Verdrängbarkeit in Abhängigkeit der seit dem letzten Zugriff vergangenen Zeit).

Indem der Benutzer einer Datei eine HSMS-Management-Klasse zuweist (Kommandos CREATE-FILE oder MODIFY-FILE-ATTRIBUTES, Operand MANAGEMENT-CLASS), unterwirft er sie den Sicherungs- bzw. Verdrängungsverfahren, welche durch diese HSMS-Management-Klasse repräsentiert werden.
Der Benutzer kann bestimmte Dateien generell oder unter bestimmten Randbedingungen von der Verdrängung auf Hintergrundebenen ausschließen (Kommandos CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES, Operand MIGRATE). Ggf. muss der Benutzer dazu die Erlaubnis zur physikalischen Allokierung besitzen.

9.4.4.7 Benutzerkontingente festlegen

Die Systembetreuung legt Benutzerkontingente für ein Volume-Set mit dem Kommando MODIFY-USER-PUBSET-ATTRIBUTES fest. Dazu gibt es folgende Operanden:

- PERM-SPACE-LIMITS
  Bestimmt den maximalen Speicherplatz, den der Benutzer auf dem Volume-Set für permante Dateien verbrauchen darf.
- TEMP-SPACE-LIMITS
  Bestimmt den maximalen Speicherplatz, den der Benutzer auf dem Volume-Set für termporäre Dateien verbrauchen darf.
- WORK-SPACE-LIMITS
  Bestimmt den maximalen Speicherplatz, den der Benutzer auf dem Volume-Set für Arbeitsdateien verbrauchen darf.

Für alle Operanden können die Grenzwerte nach den Unter-Kontingenten differenziert werden (TOTAL-SPACE, S0-LEVEL-SPACE, HIGH-PERF-SPACE, VERY-HIGH-PERF-SPACE und HIGH-AVAIL-SPACE).

9.5 Pubsets in/außer Betrieb nehmen

Die In- und Außerbetriebnahme von Daten-Pubsets ist ausschließlich über Kommandoschnittstellen möglich. Der Home-Pubset wird automatisch bei der Systemeinleitung (Startup) in Betrieb bzw. bei der Systembeendigung (Shutdown) außer Betrieb genommen.

Mit dem Kommando SHOW-PUBSET-IMPORT-EXPORT kann sich die Systembetreuung über den Bearbeitungsstand laufender Import- und Export-Aufträge informieren.

Statische Pubset-Zustände

Die In- und Außerbetriebnahme eines Pubsets bewirkt jedes Mal einen Wechsel des Pubset-Zustands, der ebenfalls im MRSCAT-Eintrag hinterlegt ist und der Auskunft über die Nutzbarkeit des Pubsets gibt, siehe auch folgendes Bild:

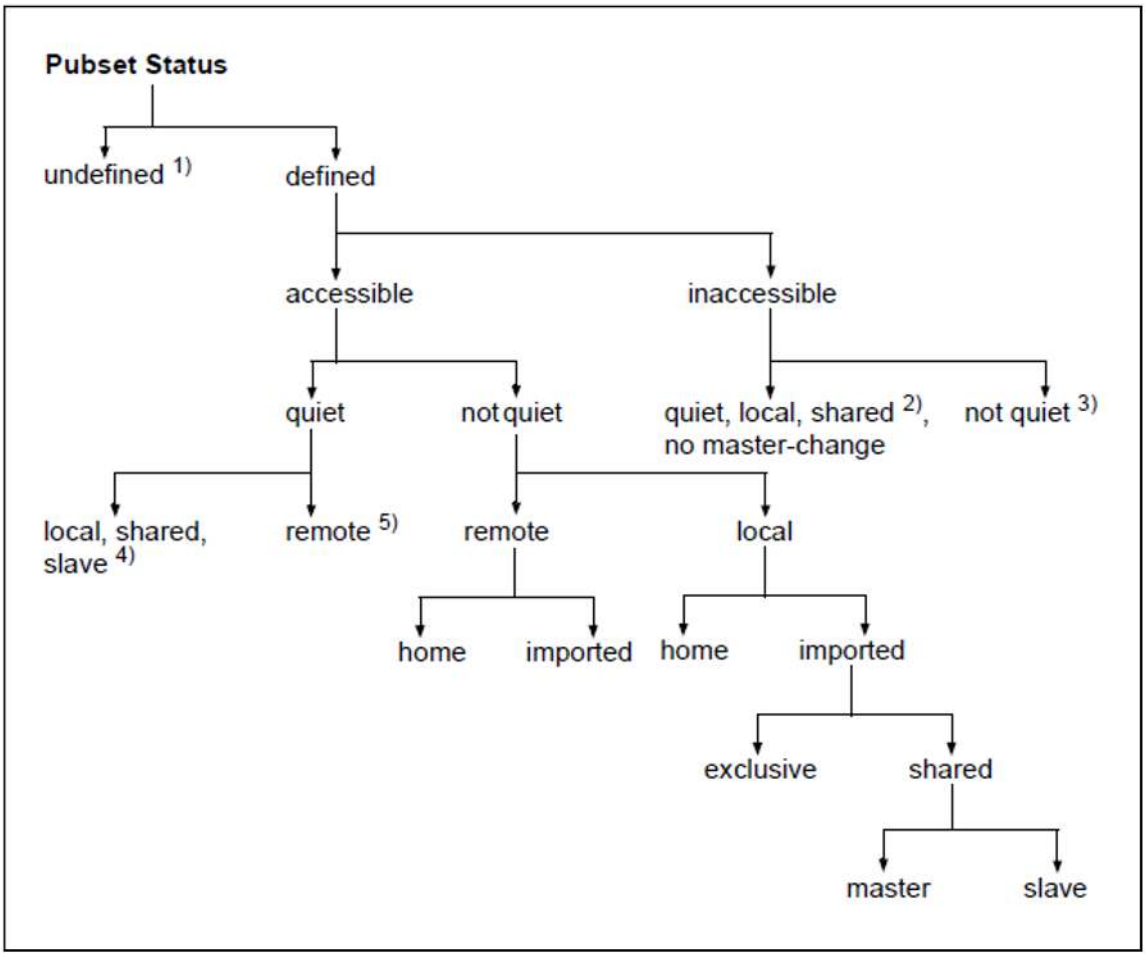


Bild 13: Statische Pubset-Zustände

1) Dieser Zustand bedeutet, dass kein MRSCAT-Eintrag definiert ist.

2) Dieser Zustand wird an einem Pubset-Slave eingenommen, wenn ein Shared Pubset nach Misslingen des Masterwechsels keinen Pubset-Master mehr hat. Der Operator muss explizit mit dem Kommando IMPORT-PUBSET SHARER-TYPE=*MASTER(MASTER-CHANGE=*YES) auf einem der Pubset-Slaves den Master-Wechsel starten.

3) Dieser Zustand wird eingenommen, wenn der Pubset nicht importiert ist.

4) Dieser Zustand wird nach MSCF-Verbindungsausfall beim Pubset-Slave eingenommen.

[5)] Dieser Zustand wird nach MSCF-Verbindungsausfall eingenommen.

9.5.1 Pubset in Betrieb nehmen

Das Inbetriebnehmen eines Pubsets wird durch das Kommando IMPORT-PUBSET eingeleitet. Grundvoraussetzung dafür sind die Hardware-Verbindungen zu den jeweiligen Platten.

> ¡ Ein Pubset, in dem Dateien auf Net-Storage katalogisiert sind, kann auch dann in Betrieb genommen werden, wenn der entsprechende Net-Storage nicht zur Verfügung steht. Es wird aber aus Verfügbarkeitsgründen empfohlen, den Net-Storage zuerst verfügbar zu machen und dann den Pubset in Betrieb zu nehmen, siehe Kapitel „Net-Storage-Verwaltung".

Eine weitere Voraussetzung ist, dass für den Pubset ein MRSCAT-Eintrag existiert, der folgende Informationen enthalten muss:

- Pubset-Kennung
- Pubset-Typ (SF- oder SM-Pubset)
- Kennung des Control-Volume-Sets (nur bei SM-Pubsets)

Alle weiteren Informationen können entweder zum Zeitpunkt der Inbetriebnahme angegeben oder aus den Ablageorten auf dem Pubset selbst ermittelt und in den jeweiligen MRSCAT-Einträgen hinterlegt werden. Die dem Pubset zugeordneten MRSCAT-Einträge müssen während der Pubset-Session erhalten bleiben. Die Definitionen in den Einträgen können zwar modifiziert werden, die Änderungen werden jedoch erst zur nächsten Pubset-Session wirksam oder müssen durch die Funktionen der Pubset-Rekonfiguration aktiviert werden.

9.5.2 Besonderheiten der Inbetriebnahme

Bei der Inbetriebnahme können nicht vorhersehbare Sondersituationen auftreten, z.B.:

- System-interne Schwierigkeiten oder Engpässe (z.B. Speicherplatzmangel)
- Verbindungsprobleme
- Volume-DefekteMehrdeutigkeit von Volume-Identifizierern
- inkonsistente Pubset-Metadaten (z.B. ungültige Zeitstempel)
- fehlerhafte Current-Master-Zuordnungen
- Schwierigkeiten beim Zuordnen der angeforderten Cache-Bereiche
- Probleme beim Konnektieren noch zugeordneter Cache-Bereiche

Bei auftretenden Problemen während des Import-Vorgangs hat die Systembetreuung die Möglichkeit den laufenden Import-Auftrag mit dem Kommando CANCEL-PUBSET-IMPORT vorzeitig abzubrechen. Im Shared-Pubset-Verbund kann insbesondere erreicht werden, dass ein laufender Import mit Master-Wechsel abgebrochen wird und der Pubset zumindest von einem anderen System aus zugreifbar wird.

Homogenität überprüfen

Mit Angabe des Operanden CHECK-PUBSET-MIRRORS=*YES im Kommando IMPORT-PUBSET wird für den zu importierenden Pubset die Homogenität der Pubset-Spiegelung überprüft. Sobald bei einem Pubset-Volume eine Inhomogenität festgestellt wird, wird der Import-Vorgang unterbrochen und die Systembetreuung hat die Wahl, Import-Vorgang abzubrechen oder fortzusetzen (siehe "Homogenität bei Pubset-Erweiterung prüfen" im Abschnitt "Dynamische Konfigurationsdaten von SF- und SM-Pubsets ändern").

Defekte Volume-Sets vom Import ausnehmen

Im Kommando IMPORT-PUBSET kann die Systembetreuung Volume-Sets (mit Ausnahme des Control-Volume-Sets) angeben, die vor der eigentlichen Inbetriebnahme automatisch aus dem Pubset ausgegliedert werden (Operand DEFECT-VOLUME-SET). Die Inbetriebnahme selbst erfolgt nur für den verbleibenden Pubset. Dabei wird ein Verzeichnis der auf dem ausgegliederten Volume-Set befindlichen Dateien erstellt, das bei einem späteren RESTORE-Aufruf (durch HSMS) verwendet werden kann.

Fehlerhafte F5-Kennsätze und TSOSCAT-Dateien reparieren

Besteht der Verdacht, dass Kontrollstrukturen eines Pubsets (z.B. die TSOSCAT-Datei oder F5-Kennsätze) fehlerhaft sind, können folgende Operanden im Kommando IMPORT-PUBSET angegeben werden:

- RECONSTRUCT-F5-LABEL=*YES: F5-Label Rekonstruktion explizit veranlassen
- REPAIR-TSOSCAT=*YES: Reparaturmodus einschalten

F5-Label-Rekontruktion

Wenn beim Import eines Pubsets während der Rekonstruktion der F5-Kennsätze in einer TSOSCAT-Datei ein Fehler in der Verkettung der TSOSCAT-Seiten einer Benutzerkennung festgestellt wird (fehlerhafte Benutzerkette), wird im weiteren Verlauf der F5-Rekonstruktion im Katalogmanagement (CMS) ein Analysemodus aktiviert. In diesem Analysemodus wird zur Diagnose die Verkettung der noch zu verarbeitenden Benutzerkennungen an der Konsole protokolliert. Die intakte Verkettung wird jeweils über die Meldung `DMS13A7` (eine oder mehrere pro Kennung) angezeigt. Tritt ein Fehler in einer Benutzerkette auf, wird dies mit der Meldung `DMS13A8` protokolliert.

Diese Information kann dazu verwendet werden, um einen Überblick über das Ausmaß des Schadens an der TSOSCAT-Datei zu erhalten. Der Import des Pubsets wird nach Ende der Analyse in jedem Fall mit Fehler beendet. Ein erfolgreicher Import des Pubsets ist i.A. erst nach Reparatur der betroffenen TSOSCAT-Datei möglich.

In jedem Fall sollten Sie Kontakt mit dem Service aufnehmen. Die Durchführung der Reparatur kann vom Systemspezialisten nach geeigneter Diagnose manuell vorgenommen oder automatisch veranlasst werden.

Reparaturmodus

Während der Import-Phase wird versucht, die beschädigten Benutzerketten durch das Ausketten defekter Blöcke oder das Abhängen des Kettenrests für das Katalogmanagement wieder zugreifbar zu machen.

> **ACHTUNG!**
> Dateien, deren Katalogeintrag auf den so abgehängten Seiten liegt, gehen verloren! Eine Information, welche Dateien betroffen sind, kann durch das System nicht bereitgestellt werden. In jedem Fall sollte vor Aktivierung der automatischen TSOSCAT-Reparatur der Service kontaktiert werden.

Der Reparatur-Modus muss zur Sicherheit (einmal) über die Meldung `DMS13AE` bestätigt werden. Durchgeführte Maßnahmen werden über die Meldungen `DMS13A9`, `DMS13AA`, `DMS13AB` und `DMS13AC` protokolliert. Ein Fehlschlagen der Reparatur (z.B. auf Grund eines permanenten HW-Fehlers) wird durch die Meldung `DMS13AD` angezeigt.

9.5.3 Pubset außer Betrieb nehmen

Mit dem Kommando EXPORT-PUBSET kann ein Pubset wieder außer Betrieb genommen werden. Voraussetzung für die Außerbetriebnahme eines Pubsets ist, dass kein Benutzer diesen Pubset noch als Ressource verwendet.

Auskunft über die Tasks, die momentanen einen Pubset benutzen, gibt das Kommando SHOW-PUBSET-OCCUPATION. Der Export-Auftrag sollte daher erst gestartet werden, wenn mit SHOW-PUBSET-OCCUPATION keine Belegung mehr angezeigt wird.

Wenn die Vermutung besteht, dass noch nicht freigegebene Dateilocks bestehen (z.B. wegen eines vorangegangenen Fehlers oder Systemabsturz), dann kann der Operator diese mit UNLOCK-DISK beseitigen (plattenweise oder für die den kompletten Pubset). Der Operator muss dabei sicherstellen, dass kein System ausgetragen wird, das noch mit der Platte arbeitet.

Besonderheiten der Außerbetriebnahme

Wenn der Operator den Pubset-Export starten möchte, obwohl der Pubset noch belegt ist, dann hat er folgende Möglichkeiten:

- Er startet den Export-Auftrag und wartet, bis alle Pubset-Beleger ihre Belegung freigegeben haben. Danach beginnt der eigentliche Pubset-Export.
  Dauert die Freigabe zu lange, kann er den Exportauftrag mit dem Kommando CANCEL-PUBSET-EXPORT abbrechen. Der Pubset ist dann wieder komplett verfügbar.
- Er startet den Export mit dem Kommando EXPORT-PUBSET ...,TERMINATE-JOBS=*YES. Alle Pubset-Beleger werden gezwungen, ihre Belegung freizugeben. Dadurch werden alle noch belegenden Tasks zwangsweise beendet, wobei in der Terminierungsphase auch die Pubset-Belegungen zurückgegeben werden.
- Letzte Alternative ist das Kommando FORCE-PUBSET-EXPORT. Hier wird trotz noch existierender Belegungen der Pubset-Export eingeleitet. Die Konsequenz ist, dass die Pubset-Verwaltungsdaten nicht korrekt abgebaut werden können, sodass es bei einer erneuten Inbetriebnahme in derselben BS2000-Session zu schweren Inkonsistenzen innerhalb der Verwaltungsdaten kommen und ein reibungsloser Pubset-Betrieb nicht gewährleistet werden kann. Diese Funktion soll also nur als äußerste Notmaßnahme eingesetzt werden. Sie wird mit der Meldung `DMS0379` an der Konsole protokolliert.

9.6 Pubsets administrieren

Die Pubset-Administration besteht aus folgende Aufgaben:

- Pubsets rekonfigurieren
- Speicherplatz-Sättigung überwachen
- Plattenspeicherplatz reorganisieren

9.6.1 Pubsets rekonfigurieren

Dieser Abschnitt beschreibt, wie Sie die Konfiguration von Pubsets ändern können und welche Kommandos dazu verwendet werden. Zur Rekonfiguration von Net-Storage-Volumes siehe "Rekonfiguration von Net-Storage-Volumes" im Abschnitt "Verwalten von Net-Storage in BS2000".

9.6.1.1 Statische Konfigurationsdaten von SM-Pubsets ändern

Unter den Begriff „statische Konfigurationsdaten“ eines SM-Pubset fallen die Definition der Volume-Set-Konfiguration und die Service-Attribute der Volume-Sets.

Die statischen Konfigurationsdaten eines SM-Pubsets werden mit Hilfe des Kommandos MODIFY-PUBSET-DEFINITION-FILE (EDIT-PUBSET-DEFINITION-FILE) verändert.
Das Kommando ermöglicht:

- ein neues Volume-Set mit seinen Eigenschaften für den Pubset zu definieren
- die Eigenschaften (Service-Attribute) eines bestehenden Volume-Sets zu ändern
- oder ein definiertes Volume-Set aus der Pubset-Konfiguration zu löschen (vorausgesetzt, es ist nicht in Betrieb).

Die Änderungen der statischen Konfigurationsdaten werden bei der Kommandobearbeitung sowohl in der Pubset-Konfigurationsdatei wie auch im MRSCAT vollzogen.

9.6.1.2 Dynamische Konfigurationsdaten von SF- und SM-Pubsets ändern

Unter „dynamischen Konfigurationsdaten“ versteht man Zusammensetzungs von Pubsets (Volumes von SF- und SM-Pubsets und Volume-Sets von SM-Pubsets) sowie Nutzungseinschränkungen und die Cache-Konfiguration.

Mit dem Kommando SHOW-PUBSET-PROCESSING kann die aktuelle Pubset-Konfiguration des Systems ausgegeben werden.

Mit dem Kommando SHOW-DEVICE-CONFIGURATION UNIT=*PUBSET-DEVICES(...) können Informationen über die zum Pubset gehörenden Platten ausgegeben werden.

Zusammensetzung eines Pubsets ändern

Die Zusammensetzung eines Pubsets kann mit dem Kommando MODIFY-PUBSET-PROCESSING geändert werden. Es gibt folgende Möglichkeiten:

- Zu einem SF-Pubset können einzelne Platten hinzugefügt bzw. daraus entfernt werden.
- Zu einem SM-Pubset können einzelne Volume-Sets hinzugefügt bzw. daraus entfernt werden.
- Zu einem Volume-Set, der Bestandteil eines SM-Pubsets ist, können einzelne Platten hinzugefügt bzw. daraus entfernt werden.

Pubres bzw. Volres können nicht aus dem SF-Pubset bzw. Volume-Set entfernt werden.

Mit Hilfe des Programms SIR zur Pubset-Generierung kann eine leere, zum Pubset oder Volume-Set passende Platte hinzugenommen werden.

Hinweise zur Volume-Set-Rekonfiguration

- Bevor ein Volume-Set für ein SM-Pubset aktiviert, d.h. in Betrieb genommen werden kann, muss es mit dem Kommando MODIFY-PUBSET-DEFINITION-FILE (EDIT-PUBSET-DEFINITION-FILE) definiert worden sein.
- Voraussetzung für das Deaktivieren eines Volume-Sets ist, dass es keine Dateien mehr beinhaltet und der betroffene Volume-Set gegen Primär-Allokierungen (siehe "Nutzung einschränken zum Entfernen von Platten und Volume-Sets" im Abschnitt "Dynamische Konfigurationsdaten von SF- und SM-Pubsets ändern") gesperrt ist.
  Eine Ausnahme hiervon ist ein defekter Volume-Set, der zwangsdeaktiviert werden kann, wodurch alle Dateien, die sich auf ihm befanden, verloren sind. Es wird deshalb beim Deaktivieren eines defekten Volume-Sets eine Liste aller auf dem Volume-Set befindlichen Dateien erstellt, die direkt als Eingabedatei für Rekonstruktionsaufträge an das Subsystem HSMS genutzt werden kann. Die Liste wird in der Datei $TSOS.SYS.PUBSET.DEFECT.<volume-set-id>.<datum.uhrzeit> abgelegt.
  Der Defektzustand eines Volume-Sets wird entweder durch das System selbst erkannt, z.B. wenn der Metadatenzugriff auf ein Volume-Set nicht mehr möglich ist, oder kann durch das Kommando MODIFY-PUBSET-RESTRICTIONS eingestellt werden.

Hinweise zur Platten-Rekonfiguration

- Bei der Hinzunahme einer leeren Platte müssen die Platteneigenschaften mit den Eigenschaften des Volume-Sets bzw. SF-Pubsets verträglich sein.
- Beim Entfernen von Platten wird zunächst geprüft, ob die Platte leer ist. Dann werden die Datenstrukturen bzgl. der Platte bereinigt und die Zugriffsrechte freigegeben.

- Mit den Kommandos ATTACH-/DETACH-DEVICE UNIT=*PUBSET-DEVICES(...) können Platten eines Pubsets zu- bzw. weggeschaltet werden. Da die Namen der zugehörigen Platten im SVL der Systemplatte verwaltet werden, müssen diese im MRSCAT eingetragen sein. Der Eintrag erfolgt jeweils beim Importieren bzw. Exportieren des Pubsets, bei Änderungen der Pubset-Zusammensetzung (MODIFY-PUBSET-PROCESSING) oder explizit mit den Kommandos ADD-, EDIT- oder MODIFY-MASTER-CATALOG-ENTRY.

  Es werden stets so viele Platten wie möglich zugeschaltet. Bei Spiegelung in einer Plattenspeichersystem-Steuerung werden nur die Standardplatten zugeschaltet. Sollen die Spiegelplatten zugeschaltet werden, muss im Operanden PUBSET die Spiegelplatte der Systemplatte (Pubres) angegeben werden.

Homogenität bei Pubset-Erweiterung prüfen

Bei der Pubset-Erweiterung mit dem Kommando MODIFY-PUBSET-PROCESSING wird bei Angabe des Operanden CHECK-PUBSET-MIRRORS=*YES eine Homogenitätsprüfung zur Spiegelung durchgeführt: Wenn im Verlauf der Pubset-Erweiterung ein Volume ermittelt wird, das unterschiedliche Spiegelungs-Eigenschaften im Vergleich zu bereits bearbeiteten Volumes aufweist, wird die beantwortbare Meldung `DMS1369` auf SYSOUT ausgegeben.

Abhängig von der Antwort des Aufrufers wird eine der folgenden Vorgehensweisen gewählt:

- Das Erweitern des Pubsets wird abgebrochen.
- Das Erweitern des Pubsets wird trotz festgestellter Inhomogenität für das gerade bearbeitete Volume des Pubsets fortgesetzt. Dabei wird für jedes weitere Volume mit unterschiedlichen Spiegelungs-Eigenschaften die Meldung `DMS136B` an der Konsole ausgegeben.

Zur Homogenitätsprüfung muss das Subsystem SHC-OSD verfügbar sein.

> Mit dem Kommando CHECK-PUBSET-MIRRORS kann die Homogenität der Pubset-Spiegelung auch vor der dynamischen Änderung der Pubset-Konfiguration geprüft werden.
>
> Bei Inbetriebnahme eines Pubsets kann die Homogenitätsprüfung im gleichnamigen Operanden des Kommandos IMPORT-PUBSET angefordert werden.

Kriterien für die Homogenität der Pubset-Spiegelung

Ein Pubset ist homogen bezüglich lokaler Spiegelung, wenn folgende Bedingungen erfüllt sind:

- Allen Volumes des Pubsets sind gleich viele Spiegelplatten zugeordnet.
- Alle Spiegelplatten des Pubsets befinden sich im gleichen Betriebszustand. Im Wesentlichen sind hier nur die Betriebszustände `ESTABLISHED` und `SPLIT` von Bedeutung; die Zustände `ESTABLISHING` und `SPLITTING` treten nur temporär als Zwischenzustände auf.
- Wenn zusätzlich mit entfernten Spiegelplatten gearbeitet wird, so befinden sich alle dem Pubset zugeordneten Spiegelplatten entweder in ihrer Gesamtheit im lokalen oder im entfernten Plattenspeichersystem, oder es gibt sowohl im lokalen wie auch im entferntern Plattenspeichersystem einen vollständigen Satz von Spiegelplatten.

Bezüglich der entfernten Spiegelung ist der Pubset homogen, wenn folgende Bedingungen erfüllt sind:

- Allen Volumes des Pubsets sind im entfernten Plattenspeichersystem gleich viele Spiegelplatten zugeordnet.
- Alle entfernten Spiegelplatten sind im gleichen Zustand `ACTIVE` oder `IN-HOLD`.
- Alle entfernten Spiegelplatten werden im gleichen Modus betrieben (synchronous, semi-synchronous oder adaptive).

- Alle entfernten Spiegelplatten befinden sich im selben entfernten Plattenspeichersystem.

Nutzung einschränken zum Entfernen von Platten und Volume-Sets

Voraussetzung für eine Reduktion der Pubset-Konfiguration ist immer, dass das herauszunehmende Objekt keine Daten mehr beinhaltet. Ein wesentliches Hilfsmittel zur Erfüllung dieser Voraussetzung ist das Kommando MODIFY-PUBSET-RESTRICTIONS.
Die Systembetreuung kann damit für das jeweilige Objekt Zugangs- und Allokierungseinschränkungen vereinbaren bzw. freigeben.

- Allokierungseinschränkungen

  Sie können für Platten oder Volume-Sets vergeben werden. Dabei können Platten gänzlich gegen Allokierungen gesperrt oder nur für physikalische Allokierung freigegeben werden. Für Volume-Sets besteht die Möglichkeit, sie gegen Primär-Allokierungen zu sperren oder sie nur für physikalische Allokierung freizugeben; wobei unter physikalischer Allokierung die Direktangabe der Volume-Set-Kennung zu verstehen ist.
- Zugangssperren

  Sie existieren ausschließlich für Volume-Sets. Durch das Setzen einer Zugangssperre bleibt jeglicher Zugriff auf eine auf dem Volume-Set befindliche Datei der Systembetreuung vorbehalten. Zugleich kann ein Volume-Set vollständig für die Pubset-Session stillgelegt werden, d.h., jeglicher Zugriff auf den Volume-Set, auch auf die darauf befindlichen Metadaten, ist verboten. Diese Zugangssperre empfiehlt sich dann, wenn ein Defekt, z.B. durch fehlgeschlagene I/O-Aufträge, für den Volume-Set abzusehen ist. Kann diese Zugangssperre nicht mehr aufgehoben werden, gilt der Volume-Set anschließend als defekt.

  Für den Control-Volume-Set ist eine Sperre zum Neuanlegen von Dateien nicht möglich, da sie zu unerwünschten Behinderungen von Systemfunktionen führen würde.

Durch das Setzen der volume-bezogenen Einschränkungen wird die Anzahl der frei verfügbaren Volumes und damit des frei verfügbaren Platzes reduziert. Es wird nicht erlaubt, wenn bereits ein gravierender Speicher-Engpass vorliegt (Überschreitung des Sättigungs-Levels 4) oder durch das Setzen die Restriktion herbeigeführt würde.

9.6.1.3 Ersetzen und Entfernen von SF-Pubsets mit gleichem Namen

Der Ersatz eines Pubsets durch einen gleichnamigen Pubset geschieht wie folgt:

1. eventuell benötigte Informationen des alten Pubsets sichern
2. den alten Pubset exportieren
3. die Platten des alten Pubsets mit dem Kommando DETACH-DEVICE (siehe Handbuch „Kommandos“ [27]) wegschalten bzw. umbenennen
4. die Platten des neuen Pubsets mit SIR generieren
5. den neuen Pubset mit IMPORT-PUBSET ..,ACTUAL-JOIN=*FIRST importieren
6. eventuell gesicherte Dateien einspielen

Der Ersatz eines Pubsets durch einen gleichnamigen Pubset aus einem anderen System geschieht in folgenden Schritten:

1. eventuell benötigte Informationen des alten Pubsets sichern
2. den alten Pubset exportieren
3. die Platten des alten Pubsets mit dem Kommando DETACH-DEVICE wegschalten bzw. umbenennen
4. optional: Im MRSCAT-Eintrag des Pubsets die MN der Pubres ändern
5. den neuen Pubset im anderen System exportieren
6. die Platten des neuen Pubsets mit dem Kommando ATTACH-DEVICE zuschalten
7. den neuen Pubset importieren

Wenn der neue Pubset als Home- und Paging-Pubset verwendet werden soll, dann müssen die Platten des alten gleichnamigen Pubsets vor der nächsten Systemeinleitung umbenannt oder physikalisch weggeschaltet werden, damit die Menge der physikalisch verfügbaren Pubsets eindeutig bleibt.

Falls dies versehentlich nicht passiert, muss die Systembetreuung bei Systemeinleitung mittels geführtem Dialog die benötigten Platten sorgfältig auswählen.

9.6.1.4 Bestehende Pubsets in große Pubsets umwandeln

Bestehende Pubsets können mit dem Kommando SET-PUBSET-ATTRIBUTES zu großen Pubsets ohne und mit Unterstützung großer Dateien hochgestuft werden.

```
/SET-PUBSET-ATTRIBUTES ...,LARGE-VOLUMES=*UNCHANGED/*ALLOWED(
                           LARGE-FILES=*UNCHANGED/*ALLOWED)
```

Dabei sind folgende Hochstufungen möglich:

Kleiner Pubset	zu	Großer Pubset ohne große Dateien
Kleiner Pubset	zu	Großer Pubset mit großen Dateien
Großer Pubset ohne große Dateien	zu	Großer Pubset mit großen Dateien

Beim Import des Pubsets werden diese Attribute in den MRSCAT übernommen und durch die entsprechenden Auskunftsfunktionen angezeigt:

- Kommando SHOW-MASTER-CATALOG-ENTRY
- Programmschnittstelle STAMCE

Zum Einrichten und Erweitern von (großen) Pubsets mit SIR siehe das Handbuch „Dienstprogramme“ [15].

> ¡ Die Erlaubnis zu großen Volumes und Dateien kann einem Pubset nicht wieder entzogen werden.

Mit den Kommandos SHOW-PUBSET-ATTRIBUTES und SHOW-MASTER-CATALOG-ENTRY können alle Pubsets mit den Attributen „große Volumes“ und „große Dateien“ angezeigt werden.

9.6.2 Zugriffsberechtigung und Benutzerkatalog verwalten

Mit Hilfe der Kommandos ADD-USER, MODIFY-USER-ATTRIBUTES, LOCK-USER, REMOVE-USER und UNLOCK-USER legt die Systembetreuung fest, welcher Benutzer auf welchen Pubset zugreifen darf und ob einem Benutzer die Zugriffsberechtigung für einen Pubset entzogen wird. Da jeder Pubset einen eigenen Benutzer- und Dateikatalog hat, kann die Systembetreuung die Benutzerkennungen in Abhängigkeit der Funktionen, die die Benutzer ausüben, auf die Pubsets aufteilen. Dabei gilt:

- Im Benutzerkatalog des Home-Pubsets müssen für sämtliche Benutzer Einträge hinterlegt sein, weil die Zugangsüberprüfungen beim Kommando SET-LOGON-PARAMETERS im Benutzerkatalog des Home-Pubsets durchgeführt werden.
- Es wird empfohlen, Benutzer im Benutzerkatalog des Home-Pubsets mit PUBLIC-SPACE-LIMIT=0, FILE-NUMBER-LIMIT=0 und JV-NUMBER-LIMIT=0 einzutragen. Die Benutzer können ungehindert auf Dateien dieses Pubsets zugreifen, sie dürfen aber keine Dateien erzeugen oder Job-Variablen verwenden.

  > ¡ Ein Benutzereintrag mit PUBLIC-SPACE-LIMIT=0 hat keine Auswirkungen auf das Erstellen von Jobvariablen (diese belegen keinen Speicherplatz) oder den Zugriff auf Dateien anderer Benutzerkennungenden (dies ist den normalen Prüfungen unterworfen: Mehrbenutzbarkeit, Kennwörter, Lese- oder Schreibzugriff usw.). Das Erstellen von Jobvariablen und Dateieinträgen auf diesem Pubset wird durch JV-NUMBER-LIMIT=0 und FILE-NUMBER-LIMIT=0 verhindert.

- Wenn ein Benutzer im Benutzerkatalog des jeweiligen Pubsets keinen Eintrag hat, dann kann er auf diesen Pubset nicht zugreifen, auch nicht auf mehrbenutzbare Dateien oder Jobvariablen anderer auf diesem Pubset geführter Benutzer. Dies ist z.B. zu empfehlen, wenn der Pubset für die Produktion benötigt wird und Benutzerkennungen lediglich Test- und Ausbildungszwecken dienen.
- Der Zugriffsschutz ist unwirksam, wenn bei der Systemeinleitung der Systemparameter FSHARING den Wert 1 erhalten hat. Dann können alle Benutzer auf sämtliche Pubsets zugreifen und benötigen dazu nicht einmal einen Eintrag im jeweiligen Benutzerkatalog.

9.6.3 Speicherplatz-Sättigung überwachen

Um Engpässe an Speicherplatz auf gemeinschaftlichen Datenträgern rechtzeitig erkennen zu können, überwacht das System den Sättigungsgrad von Pubsets. Die Systembetreuung beeinflusst diese Überwachung durch die Vereinbarung von Sättigungsstufen, siehe Abschnitt „Sättigungsschwellwerte für Volume-Sets und SF-Pubsets festlegen". Ein Sättigungsgrad ist erreicht, wenn auf einem SF-Pubset bzw. Volume-Set weniger Platz frei ist, als für die entsprechende Sättigungsstufe festgelegt worden ist..

Meldungen über Sättigungszustand

Bei der Änderung des erreichten Sättigungsgrads werden die Meldungen `EXC044i`, `DMS141C` oder `DMS1400` an der Konsole ausgegeben:

`EXC044i`	Meldung bei SF-Pubsets, wenn Sättigungsstufe i erreicht ist. `i` ist eine Ziffer von 0 bis 5.
`DMS141C`	Meldung bei SF-Pubsets beim Überschreiten des Zip-Levels.
`DMS1400`	Meldung für Volume-Sets eines SM-Pubsets. In der Meldung wird der Sättigungsgrad (0 bis 6) ausgegeben, wobei 6 für den Zip-Level steht.

Eine Sonderrolle spielt die Meldung `EXC044E` siehe „Behandlung der Speicherplatzanforderungen“ unten.

Die folgende Tablle zeigt die Maßnahmen des Operators abhängig vom Sättigungsgrad:

Grad	Auswirkungen im System	Maßnahmen des Operators
0	Normaler Systemablauf	Keine
1-3	Vorwarnung wegen beginnender Sättigung für einen Pubset bzw. Volume-Set	Systembetreuung verständigen; die Dialogteilnehmer mit dem Kommando INFORM-ALL-JOBS zum Löschen von nicht benötigten Dateien auffordern
4	Speicherplatzanforderungen von Benutzer-Jobs für einen Pubset bzw. Volume-Set werden mit Meldung `DMS0441` abgewiesen; Ausnahmen siehe unten	Systembetreuung verständigen; bei Bedarf Meldung `EXC044E` beantworten
5	Wie bei Sättigungsgrad 4; zusätzlich Einschränkungen bei Systemtasks	Systembetreuung verständigen.
6 (=ZIP)	Wie bei Sättigungsgrad 5; zusätzlich Einschränkung auf Systemlauf, der mit einem Systemstart vom Typ ZIP eingeleitet worden ist	Systembetreuung verständigen. Nachdem wieder genügend freier Speicherplatz auf den Platten geschaffen wurde, neuen Systemlauf mit normalem Typ des Systemstarts einleiten

Tabelle 21: Sättigungszustand eines Pubsets

Informationen über den Sättigungszustand eines Pubsets liefert auch das Kommando SHOW-PUBSET-SPACE-ALLOCATION.

Behandlung der Speicherplatzanforderungen

- Anforderungen, die von Benutzer-Jobs gestellt werden und deren Erfüllung zum Erreichen des Sättigungsgrades 4 führen würden, werden abgewiesen, es sei denn, der Systemparameter L4MSG ist mit dem Wert 1 belegt. In diesem Fall wird für jede Benutzer-Anforderung von Speicherplatz, die zum Erreichen des Sättigungsgrades 4, nicht jedoch auch des Sättigungsgrades 5 führen würde, folgende beantwortbare Meldung an der Konsole ausgegeben:

  ```
  EXC044E    SATURATION LEVEL 4 FOR PUBSET/VOLUMESET '(&00)' EXCEEDED.
             REQUEST FOR DISK SPACE ACCEPTED NEVERTHELESS? REPLY
             (Y=YES; N=NO)
  ```

  Die Bewilligung der Speicherplatzanforderung durch den Operator (Antwort Y) sollte restriktiv gehandhabt werden.
- Anforderungen für die Benutzerkennung TSOS werden unabhängig vom erreichten Sättigungsgrad angenommen, sofern nach ihrer Erfüllung noch eine minimale System-Notreserve an Speicherplatz vorhanden ist. Diese Notreserve wird nur bei einem Systemstart im Modus ZIP verbraucht.
- Anforderungen von Systemtasks können auch bei Sättigungsgrad 4 oder 5 erfüllt werden.
- Anforderungen für die Erstellung von Systemdumps werden abgewiesen, wenn sie zum Erreichen des Sättigungsgrades 5 führen würden.
- Bei SM-Pubsets wird versucht, Anforderungen, die sich nicht auf einen bestimmten Volume-Set des Pubsets beziehen, auf dem Volume-Set mit dem niedrigsten Sättigungsgrad zu erfüllen (bei gleich guter Eignung der Volume-Sets in Bezug auf die anderen Kriterien der Volume-Set-Auswahl). Die Anforderung wird abgewiesen, wenn auf allen Volume-Sets, die für die Allokierung in Frage kommen, die Erfüllung der Anforderung zum Erreichen eines Sättigungsgrads (wie oben beschrieben) führen würde.

Maßnahmen zur Verhinderung der Sättigung

Das Auftreten der Sättigungsgrade 4 und 5 zeigt an, dass der betroffene Pubset/Volume-Set überlastet ist. Wenn ein Pubset bzw. Volume-Set immer wieder in diesen Zustand kommt, ist er nicht ausreichend konfiguriert.

Entsprechende Maßnahmen, die durch die Systembetreuung zu treffen sind, können sein:

- Umverteilung einzelner Benutzerkennungen auf andere Pubsets
- Vergrößerung des Pubsets bzw. Volume-Sets durch Hinzunahme von Platten
- Verdrängung von nicht benutzten Dateien mit Hilfe des Softwareprodukts HSMS
- Regelmäßige Aufforderung an die Benutzer, nicht benötigten Speicherplatz freizugeben und Dateien auszulagern
- Kontrolle der Plattenspeicherplatz-Zuweisung mit dem Dienstprogramm SPCCNTRL oder dem Kommando SHOW-PUBSET-SPACE-ALLOCATION
- Sparsamer Umgang mit der Funktion „Speicherplatzüberschreitung“ (Kommando ADD-USER PUBLIC-SPACE-EXCESS=*NO/*ALLOWED)
- Rechtzeitige Reorganisation

9.6.4 Plattenspeicherplatz reorganisieren

Durch das ständige Anlegen, Löschen, Vergrößern und Verkleinern von Dateien im laufenden Betrieb kommt es auf den Volumes eines Pubsets zu einer immer stärkeren Fragmentierung des freien Speicherbereichs und der anzulegenden Dateien. Die Fragmentierung beeinträchtigt in zunehmenden Maße die Performance der Dateizugriffe und die gleichmäßige Verteilung der I/O-Last über alle Volumes des Pubsets. Die Extent-Liste in den Katalogeinträgen wird durch das zwangsläufige Anlegen mehrerer kleiner Datei-Extents beim Vergrößern der Dateien verlängert.

SPACEOPT

Das Softwareprodukt SPACEOPT bereinigt eine Fragmentierung durch die optimale Verlagerung (Reorganisation) der Datei-Extents auf den Volumes eines Pubsets. Ziel von SPACEOPT ist, auf einem Volume möglichst große zusammenhängende freie Speicherbereiche zu schaffen, damit eine Allokierung großer Dateien mit einer geringer Anzahl an Extents erfolgen kann. Des Weiteren wird die Anzahl der Extents nach Möglichkeit verringert, indem mehrere kleine Extents in einem größeren Extent zusammengefasst werden. Im Ergebnis verbessert sich die Performance der Dateizugriffe. Auf den Volumes des Pubsets wird wieder großer, zusammenhängender freier Speicherplatz geschaffen und die Gefahr des Überlaufs der Extent-Listen in den Katalogeinträgen wird beseitigt.

SPACEOPT ist im gleichnamigen Handbuch [52] beschrieben.

HSMS

Der gemeinschaftliche Plattenspeicher (Pubsets) kann durch eine Sicherung mit anschließender Rekonstruktion reorganisiert werden.
Mit folgender HSMS-Anweisung können alle Dateien wieder zusammenhängend zurückgeschrieben und damit der Speicherplatz neu organisiert werden:

```
//RESTORE-FILES REPLACE-FILES-AND-JV=*YES(REORGANIZE-SPACE=*YES)
```

Wenn zusätzlich der Operand RELEASE-UNUSED-SPACE=*YES angegeben wird, wird weiterer Speicherplatz eingespart: Die zugewiesenen (allokierten), aber nicht genutzten Seiten hinter dem Last Page Pointer der Datei werden freigegeben.

9.7 Shared-Pubsets

Bei Einsatz des Produkts HIPLEX MSCF und einer entsprechenden Hardware-Konfiguration ist der gleichzeitige und gemeinsame Zugriff über max. 16 Systeme (BS2000 native oder Gastsysteme unter VM2000) hinweg auf einen Pubset möglich. Dieser mehrbenutzbare Pubset wird „Shared-Pubset“ genannt.

Mit HIPLEX MSCF steht neben dem Shared-Pubset-Verbund eine erweiterte Verbundfunktionalität zur Verfügung: der XCS-Verbund (Cross-Coupled-System).
Der XCS-Verbund bietet eine engere Koordination der beteiligten Systeme als ein Shared-Pubset-Verbund. Ein „XCS-Pubset“ dient als zentraler Ablageort für verbundweit benötigte Daten. XCS-Pubsets werden automatisch durch das System importiert.

Das gesamte Konzept des Shared-Pubset (Hardware-Konfiguration, Verwaltung der Pubsets, Datenzugriffe, Watch-Dog-Mechanismus, Shared-Pubset- und XCS-Verbund) ist ausführlich im Handbuch „HIPLEX MSCF“ [32] beschrieben.

Mit dem Kommando SHOW-SHARED-PUBSET können Informationen über den Shared-Pubset ausgegeben werden.

> ¡ Mehrere BS2000-Systeme unter BS2000/OSD-BC ab V9.0 können gemeinsam auf Net-Storage-Volumes von Shared-Pubsets zugreifen.

9.7.1 Shared-Pubset-Verbund einrichten

Zur Realisierung des Shared-Pubset-Verbunds wird auf das MASTER/SLAVE-Prinzip zurückgegriffen. Ein System des Verbunds wird zum temporären Eigentümer des Pubsets ernannt („Pubset-Master“) und wickelt zentral alle Verwaltungsfunktionen der Metadaten ab. Alle weiteren am Verbund teilnehmenden Systeme, die Pubset-Slaves oder auch „Slave-Sharer“, richten ihre Verwaltungsanforderungen über MSCF-Funktionen an den Pubset-Master.

Systemkennung vergeben

Die Systemkennung (Sysid) identifiziert die Systeme in einem Shared-Pubset-Verbund. Beim Shared-Pubset-Betrieb ist darauf zu achten, dass die Systemkennung im Rechnerverbund eindeutig ist.

Bei der Vergabe einer Systemkennung, die intern als Synonym für den BCAM-Namen des Systems geführt wird, muss je nach Pubset-Notation unterschieden werden:

- Bei einer VSN in PUB-Notation (siehe "VSN in PUB-Notation") ist die Systemkennung mit der einstelligen Katalogkennung identisch (Sysid=Catid).
- Bei einer VSN in Punkt-Notation (siehe "VSN in Punkt-Notation") kann die Systemkennung einen numerischen Wert von 65 bis 192 annehmen.
  Die systemseitige Voreinstellung beträgt 250 (ist also ungültig).

Die Vergabe der Systemkennung erfolgt beim Einrichten des Home-Pubsets mittels SIR über den Operanden SYS-ID der Anweisung DECLARE-PUBSET. Bei existierenden Home-Pubsets kann die Systemkennung über das Kommando SET-PUBSET-ATTRIBUTES neu vergeben werden, die Änderung wird jedoch erst beim nächsten Systemstart wirksam.

Pubset-Master auswählen

Das Eintragen des gewünschten Pubset-Masters eines Shared-Pubsets kann über das Kommando SET-PUBSET-ATTRIBUTES erfolgen. Zusätzlich kann mit Hilfe dieses Kommandos auch ein Backup-Master eingetragen werden, der bei einem eventuellen Master-Ausfall dessen Funktion übernimmt. Diese Vorgehensweise wird Master-Wechsel genannt.

Im DMS-Record des SVL der Pubres des SF-Pubsets bzw. der Volres des Control-Volume-Sets gibt es zwei Felder, die Auskunft über den gewünschten und den aktuellen Pubset-Master geben. Während der Inbetriebnahme des Shared-Pubsets werden diese Felder ausgewertet und die Auswahl erfolgt in folgender Reihenfolge:

1. Ist bereits ein aktueller Pubset-Master eingetragen, gilt dieses Fremd-System als Pubset-Master; das eigene System ist dann zwangsläufig ein Pubset-Slave.
2. Bei der Inbetriebnahme selbst wurde mitgeteilt, dass das eigene System Pubset-Master werden soll.
3. Der im SVL als gewünschter Eigentümer eingetragene Pubset wird Pubset-Master.
4. Ist keine der vorangegangenen Bedingungen erfüllt, wird das System Pubset-Master, das als erstes die Inbetriebnahme des Shared-Pubsets veranlasst hat.

Das Kommando SHOW-PUBSET-ATTRIBUTES zeigt die Einstellungen des gewünschten Masters, des aktuellen Masters und des Backup-Masters an.

Pubset-Verbund aufbauen und abbauen

Der Aufbau eines Shared-Pubset-Verbunds geschieht in zwei Schritten:

1. Zunächst wird an allen beteiligten Systemen das Subsystem MSCF gestartet und die erforderlichen Verbindungen werden aufgebaut. Dabei muss mindestens die Verbindung zwischen zukünftigem Pubset-Master und Pubset-Slave bestehen, für einen eventuellen Master-Wechsel zusätzlich die Verbindung zwischen Backup-Master und Pubset-Slave.
2. An allen beteiligten Systemen wird der Shared-Pubset in Betrieb genommen.
   Die Auswahl des Pubset-Masters erfolgt nach der zuvor beschriebenen Eigentümer-Auswahl. Die Inbetriebnahme an den Pubset-Slaves kann erst dann vollständig abgearbeitet werden, wenn sie am Pubset-Master erfolgreich abgeschlossen ist.

Ein Shared-Pubset-Verbund wird implizit beim Exportieren des Shared-Pubset abgebaut.

Konfigurationsänderung

Die Konfiguration des Shared-Pubset-Verbunds kann sich dynamisch ändern. Die Ursachen für eine solche Änderung sind:

- Ein weiterer Pubset-Slave konnektiert sich durch Inbetriebnahme des Shared-Pubsets.
- Ein Pubset-Slave diskonnektiert sich durch Außerbetriebnahme des Shared-Pubsets.
- Ein Pubset-Slave oder der Pubset-Master ist ausgefallen und der Master-Wechsel wurde erfolgreich durchgeführt.

9.7.2 Shared-Pubset überwachen und Fehler beheben

- Pubset-Locks manuell beheben
- Verhalten bei Systemausfall

9.7.2.1 Pubset-Locks manuell beheben

Die Locks des Pubset-Managements (Pubset-Locks) dienen im Shared-Pubset-Verbund dazu, Pubset-Rekonfigurationsaufträge untereinander und diese mit Import- und Export-Aufträgen zu synchronisieren. Pubset-Locks werden auch verwendet, wenn verhindert werden soll, dass sich an der Volume-Konfiguration eines Pubsets etwas ändert, z.B. beim Anlegen eines Snapsets.

Typen von Pubset-Locks

Jeder Pubset-Lock wird in Form eines Lock-Eintrags entweder am Pubset-Master oder an einem der Pubset-Slaves (siehe "Shared-Pubset-Verbund einrichten") geführt. Das System, an dem der Lock-Eintrag des Locks geführt wird, wird auch als Lock-Location bezeichnet. Zum Lock-Eintrag gehören Informationen über den Typ und den Halter des Pubset-Locks (Task-Id und System-Id).

Es gibt folgende Typen von Pubset-Locks:

- PUBSET-RECONFIGURATION
  Dient dem Schutz von Pubset-Rekonfigurationsaufträgen untereinander und gegen Import- und Export-Aufträge. Dieser Pubset-Lock kann am Pubset-Master und am Pubset-Slave gesetzt sein.
- SHARED-EXCAT (werden von Export-Tasks gehalten)
  Verhindert Pubset-Rekonfigurationsaufträge, erlaubt aber parallele Pubset-Locks. Dieser Pubset-Lock kann am Pubset-Master und am Pubset-Slave gesetzt sein.
- SHARED-IMCAT (werden von Import-Tasks gehalten)
  Verhindert Pubset-Rekonfigurationsaufträge, erlaubt aber parallele Pubset-Locks. Dieser Pubset-Lock kann nur am Pubset-Master gesetzt sein.
- SHARED-MASTER-EXCAT
  Verhindert Pubset-Rekonfigurationsaufträge, erlaubt aber parallele Pubset-Locks des Typs SHARED-EXCAT. Dieser Pubset-Lock kann nur am Pubset-Master gesetzt sein.

Zur Diagnose von Lock-Zuständen und zur Behebung von fehlerhaften Lockzuständen gibt es die Kommandos SHOW-PUBSET-LOCKS und REMOVE-PUBSET-LOCK (Privileg TSOS).

Da es sich bei Pubset-Locks um interne Sperren des Pubset-Managements handelt, sollte eine manuelle Freigabe nur in abnormalen Lock-Situationen erforderlich sein, z.B. nach dem Ausfall einer MSCF-Verbindung im Shared-Pubset-Verbund. Abnormale Lock-Situationen liegen z.B. vor, wenn das Pubset-Management über interne Schnittstellen feststellen kann, dass der Task des Lock-Halters nicht mehr lebt oder sich im Zustand „pended indefinitely“ befindet.

Beispiele für normale Lock-Situationen

	vorhandener Pubset-Lock			
angeforderter Pubset-Lock	**PUBSET-RECON-FIGURATION**	**SHARED-IMCAT**	**SHARED-EXCAT**	**SHARED-MASTER-EXCAT**
PUBSET-RECON-FIGURATION	Task wartet	nicht zulässig	nicht zulässig	nicht zulässig
SHARED-IMCAT	nicht zulässig	zulässig	zulässig	nicht zulässig
SHARED-EXCAT	nicht zulässig	zulässig	zulässig	zulässig
SHARED-MASTER-EXCAT	nicht zulässig	Task wartet	zulässig	Task wartet

Tabelle 22: Kombinationen von Pubset-Locks am Pubset-Master

In den folgenden Beispielen besteht der Shared-Pubset-Verbund aus dem Pubset-Master mit der SysId „183“ und zwei Pubset-Slaves mit den SysIds „184“ und „185“.

1. Folgende Lock-Situation kann z.B. während der Bearbeitung von /MODIFY-PUBSET-PROCESSING auftreten. Das Kommando wurde am Pubset-Master oder am Pubset-Slave (in diesem Fall wird das ganze Kommando an den Pubset-Master geschickt) eingegeben. Unter dem Schutz eines PUBSET-RECONFIGURATION-Locks am Pubset-Master werden PUBSET-RECONFIGURATION-Locks an den Pubset-Slaves gesetzt und nach Beendigung der dort durchzuführenden Verarbeitung wieder freigegeben.

```
/show-pubset-locks pubset-id=puba
LOCK-TYPE      LOCK-LOCATION               LOCK-HOLDER-INFORMATION
               HOSTNAME  SYSID  SHARER-    TID       SYSID  BS2000
                                TYPE                     Version
*PUBSET-RECONF D017ZE15  183    *MASTER    1000004F  183    V20.0
*PUBSET-RECONF D017ZE16  184    *SLAVE     2000007F  184    V20.0
*PUBSET-RECONF D017ZE17  185    *SLAVE     3000009A  185    V20.0
```

   2000007F ist die TID eines MSCF-Servertasks am Pubset-Slave „184“;
   3000009A ist die TID eines MSCF-Servertasks am Pubset-Slave „185“

   Anmerkung
   Nicht zu jedem Zeitpunkt müssen die PUBSET-RECONFIGURATION-Locks am Pubset-Slave gesetzt sein. Abhängig vom Stand der jeweiligen Verarbeitung können diese Locks noch nicht gesetzt oder bereits wieder freigegeben sein.

2. Folgende Lock-Situation kann während der Bearbeitung von /EXPORT-PUBSET, das am Pubset-Slave „184“ eingegeben wurde, auftreten.

```
/show-pubset-locks pubset-id=puba
LOCK-TYPE      LOCK-LOCATION               LOCK-HOLDER-INFORMATION
               HOSTNAME  SYSID  SHARER-    TID       SYSID  BS2000
                                TYPE                     Version
*SHARED-EXCAT  D017ZE16  184    *SLAVE     2000007E  184    V20.0
*SHARED-EXCAT  D017ZE15  183    *MASTER    2000007F  184    V20.0
```

`2000007F` ist die TID eines Export-Tasks am Pubset-Slave „184“.

Anmerkung

Nicht zu jedem Zeitpunkt müssen die SHARED-EXCAT-Locks an Pubset-Slave und Pubset-Master gleichzeitig gesetzt sein. Abhängig vom Stand der Verarbeitung kann der SHARED-EXCAT-Lock auch nur an einem der beiden Systeme gesetzt sein.

Am Pubset-Slave „185“ ist in diesem Beispiel kein SHARED-EXCAT-Lock gesetzt, da dieses System vom Export-Auftrag an Pubset-Slave „184“ nicht betroffen ist.

9.7.2.2 Verhalten bei Systemausfall

Bei Ausfall eines Systems innerhalb des Shared-Pubset-Verbundes müssen die von ihm reservierten Ressourcen freigegeben oder Recovery-Maßnahmen eingeleitet werden. Alle am Shared-Pubset-Verbund beteiligten Systeme werden vom Subsystem MSCF überwacht.

Zur Systemüberwachung dienen zwei Kontrollmechanismen:

- die sog. Watch-Dog-Datei $TSOS.SYS.PVS.SHARER.CONTROL, in die alle Sharer periodisch Zeitstempel hineinschreiben (Lebendmeldungen). Fällt ein Sharer aus, kann diese Tatsache von einem anderen Sharer am Ausbleiben von dessen Lebendmeldung erkannt und entsprechende Maßnahmen eingeleitet werden.
- Beim Ausbleiben der Lebendmeldung wird die Systemverbindung überprüft, indem ein Auftrag an den betroffenen Sharer gesendet wird; dieser muss innerhalb eines bestimmten Zeitintervalls quittiert werden.

Ein Partner-Ausfall wird nur dann angenommen, wenn das Ausbleiben der Lebendmeldung durch eine erfolglose Netzwerk-/LAN-Überprüfung bestätigt wird.

Bei Ausfall des Eigentümer-Systems wird an allen abhängigen Systemen eine pubset-spezifische Jobvariable gesetzt.

Ausfall eines Pubset-Slaves

Erkennt ein Pubset-Master den Ausfall eines beteiligten Pubset-Slaves, so werden alle vom ausgefallenen Pubset-Slave reservierten Ressourcen freigegeben.

Ausfall eines Pubset-Masters

Bei Ausfall des Pubset-Masters findet, angestoßen durch den Watch-Dog-Mechanismus, ein Master-Wechsel statt. Voraussetzung für den Master-Wechsel ist, dass ein aktiver Pubset-Slave im SVL des Shared-Pubsets als Backup-Master eingetragen ist, der die neuen Master-Funktionen übernehmen soll.

Der Backup-Master wird mit dem Kommando SET-PUBSET-ATTRIBUTES BACKUP-MASTER=... im DMS-Record des SVL eingetragen. Falls kein Backup-Master eingetragen ist oder der eingetragene Backup-Master nicht aktiv ist, entscheidet der Wert des Operanden ALTERNATE-BACKUP, ob der erste aktive Pubset-Slave im SVL zum Pubset-Master wird oder der Operator explizit über das Kommando IMPORT-PUBSET SHARER-TYPE=*MASTER (MASTER-CHANGE=*YES) einen der aktiven Pubset-Slaves zum neuen Pubset-Master bestimmt oder ob der Master-Wechsel mit einem alternativen Backup-Master unterbunden werden soll.

Wenn kein Backup-Master vorgesehen ist oder die Funktion des Master-Wechsels aus einem anderem Grund scheitert, dann ist eine der folgenden Aktionen nötig:

- Alle beteiligten Pubset-Slaves nehmen den Shared-Pubset außer Betrieb und bauen den Shared-Pubset-Verbund komplett neu auf.
- Mit dem Kommando SET-PUBSET-ATTRUBUTES wird die Erlaubnis für einen nachträglichen Master-Wechsel gegeben und dieser mit dem Kommando IMPORT-PUBSET SHARER=*MASTER(MASTER-CHANGE=*YES) angestoßen.

Gründe für ein Scheitern des Master-Wechsels können sein:

- Der eingetragene Backup-Master ist nicht aktiv.
- Eine Verbindung zu einem der beteiligten Pubset-Slaves ist unterbrochen.
- Eines der am Shared-Pubset-Verbund beteiligten Systeme verwendet eine nicht-verbundkompatible Version von HIPLEX MSCF oder einen abweichenden Korrekturstand.

Nach einem erfolgreich durchgeführten Master-Wechsel können alle beteiligten Pubset-Slaves normal weiterarbeiten. Der Master-Wechsel selbst läuft für die Benutzer weitestgehend unerkannt ab.

9.7.3 Shared Pubsets im XCS-Verbund

Der XCS-Verbund (Cross-Coupled-System) bietet eine engere Koordination der beteiligten Systeme. Jedes System hat eine konsistente und vollständige Sicht des gesamten Verbunds. Der XCS-Verbund bietet damit Mechanismen zur Realisierung verteilter Anwendungen; er ist in erster Linie als Verfügbarkeits- und Lastverbund von BS2000 konzipiert. Dem Benutzer werden u.a. folgende, im DVS-Umfeld wichtige Funktionen angeboten:

- Distributed-Lock-Manager (DLM)

  Diese Funktion realisiert eine system-übergreifende Sperrenverwaltung und unterstützt damit system-übergreifende Synchronisation und Serialisierung. Sie ist Basisfunktion für SFS.

- Shared-File-System (SFS)

  Das SFS erlaubt innerhalb des XCS-Verbunds die system-übergreifende Aktualisierung von Dateien auf Shared-Pubsets, die nicht notwendig XCS-Pubsets sein müssen. Mit HIPLEX MSCF wird dieser globale Shared-Update für die block- bzw. byte-stromorientierten Zugriffsmethoden UPAM, FASTPAM und DIV unterstützt.

Ein XCS-Verbund hat weiterreichende Voraussetzungen zu erfüllen als ein Shared-Pubset-Verbund:

- ein System kann max. einem XCS-Verbund angehören
- die Teilnehmer müssen voll vermascht sein, d.h. es müssen MSCF-Verbindungen zwischen allen Systemen des Verbunds bestehen
- dem XCS-Verbund muss mindestens ein XCS-Pubset angehören, zu dem von allen Systemen aus Zugriffspfade vorhanden sein müssen

Ein XCS-Pubset dient als zentraler Ablageort für verbundweit benötigte Daten. XCS-Pubsets werden automatisch durch das System importiert.

Mit dem Kommando SHOW-XCS-PUBSET können Informationen über den XCS-Pubset ausgegeben werden.

Mit dem Kommando SHOW-XCS-OCCUPATION kann angezeigt werden, welche TU-Tasks momentan XCS nutzen.

9.8 Besonderheiten für Standby-Pubsets

Zur Erhöhung der Datensicherheit können Pubsets gespiegelt werden. Plattenspeichersysteme bieten bei Einsatz des Softwareprodukts SHC-OSD Schnittstellen zur Erstellung von Standby-Pubsets. Die Standby-Pubset können im Fehlerfall die Funktionen eines ausgefallenen Pubsets übernehmen. Somit werden Ausfallzeiten möglichst klein gehalten und die Verfügbarkeit des Pubsets bzw. des Systems erhöht.

Standby-Pubsets können für Home- und Daten-Pubsets erstellt werden. Diese werden als **Standby-Home-Pubset** oder **Standby-Daten-Pubset** bezeichnet:

- Bei Ausfall des Home-Pubsets ist das System nicht mehr ablauffähig und muss mit dem Standby-Home-Pubset neu hochgefahren werden. Anschließend kann der ausgefallene Home-Pubset rekonstruiert werden.
- Bei Ausfall eines Daten-Pubsets sind Anwendungen, die auf diesen Pubset zugreifen, u. U. nicht mehr ablauffähig. Den Nutzern dieses Pubsets kann durch Importieren des Standby-Daten-Pubsets die Wiederaufnahme der Verarbeitungsläufe auf einem bestimmten Konsistenzpunkt ermöglicht werden. Der ausgefallene Daten-Pubset kann ggf. durch die Systembetreuung rekonstruiert werden.

> ¡ Standby-Pubsets dürfen nicht eingesetzt werden, wenn im System die HSMS-Migration eingesetzt wird. Migrierte Dateien sind bei Einsatz eines Standby-Pubsets nicht mehr verfügbar!

Kommando	Bedeutung
ACTIVATE-CLONE	Clone-Paar aktivieren
RESTART-CLONE-SESSION	Clone-Paar wiedererstellen
RESTORE-FROM-CLONE	Unit von Clone-Unit rekonstruieren
SHOW-CLONE-SESSION-STATUS	Status von Clone-Paaren anzeigen
START-CLONE-SESSION	Clone-Paar erstellen
STOP-CLONE-SESSION	Clone-Paar auflösen

Tabelle 23: SHC-OSD-Kommandos für die Spiegelung von Standby-Pubsets

> ¡ Im Rahmen eines Hochverfügbarkeitskonzepts mit mehreren Plattenspeichersystemen an verschiedenen Standorten können Daten-Pubsets auch in das entfernte Plattenspeichersystem gespiegelt werden, oder Teil eines Storage-Cluster sein.
> Außerdem können mit der Funktion Snapset, die auf Snap-Funktionen der Plattenspeichersysteme beruht, Pubset-Kopien für Sicherungszwecke erzeugt werden. Zu Snapsets siehe Abschnitt „Datensicherung mit Snapsets" (Erfüllung von Sicherheitsanforderungen durch SECOS).
>
> Zu den Replikationsfunktionen der Plattenspeichersysteme siehe Handbuch „SHC-OSD“ [48].

9.8.1 Standby-Pubsets erstellen

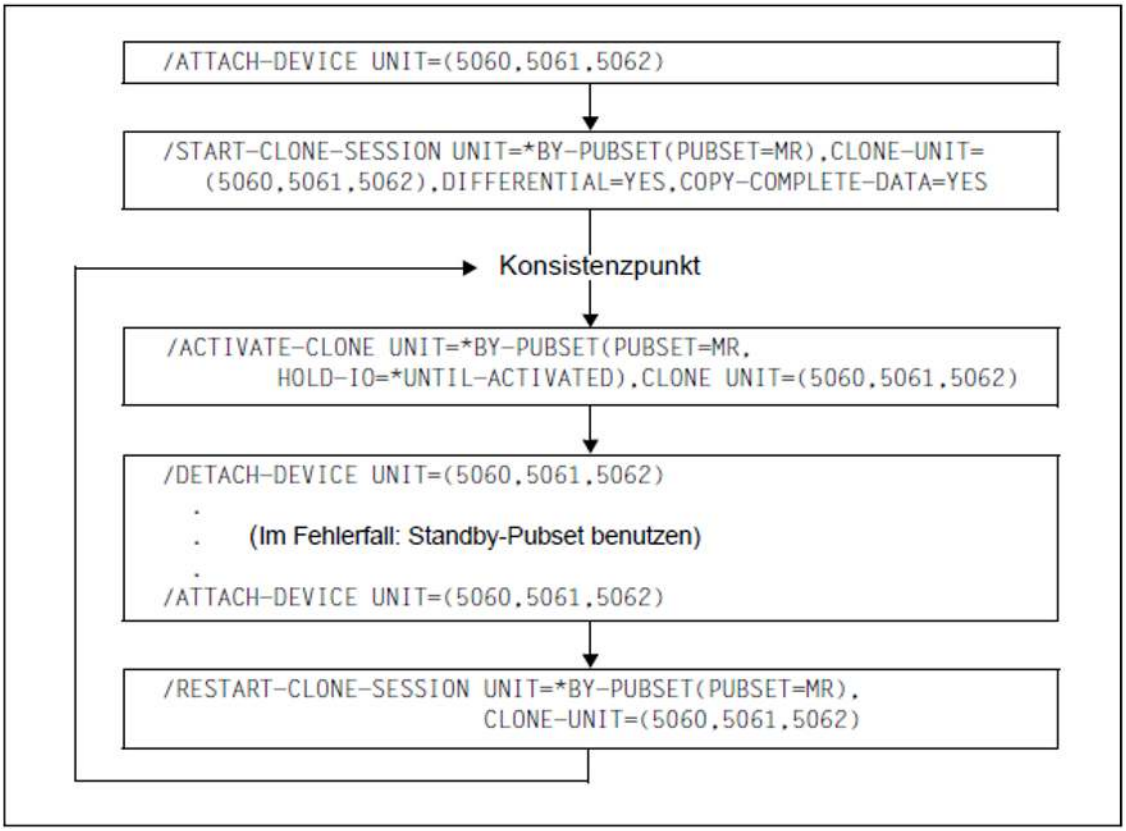
Standby-Pubsets werden mit den Replikations-Funktionen der Plattenspeichersysteme erzeugt. Die Abläufe sind dabei analog und werden in den folgenden Beispielen anhand der Replikationsmethode CLONE-TYPE=*MIRROR dargestellt. Zu unterscheiden ist, ob mit einem Standby-Pubset oder mit zwei alternierenden Standby-Pubsets gearbeitet wird.

9.8.1.1 Einzelnen Standby-Pubset einrichten

Vorgehensweise:

1. Die Platten zuschalten, die für die Spiegelung benötigt werden
   (Kommando ATTACH-DEVICE).
2. Jeder Platte (Unit) des Pubsets eine Spiegelplatte zuordnen
   (Kommando START-CLONE-SESSION).
   Dadurch werden die Daten vom Pubset und auf die Spiegelplatten kopiert und dann synchron gehalten.
3. Konsistenzpunkt erstellen.
   - Für Home-Pubsets bedeutet das:
     Host-Puffer für den Home-Pubset zurückschreiben und Anwendungen auf dem Home-Pubset anhalten oder beenden (DAB-Schreib-Caches sind auf dem Home-Pubset nicht erlaubt und müssen deshalb nicht berücksichtigt werden).
   - Für Daten-Pubsets muss eine der folgenden Aktionen ausgeführt werden:
     - Alle Dateien schließen, DAB-Schreib-Caches beenden und Host-Puffer anderer Anwendungen zurückschreiben. Dies kann dadurch erreicht werden, dass alle Anwendungen kurzzeitig beendet werden und der Pubset exportiert wird.
     - Die Anwendungen sorgen selbst für einen Konsistenzpunkt, ohne den Pubset zu exportieren (siehe Handbuch „SHC-OSD“ [48]).
4. Platten und Spiegelplatten trennen
   (Kommando ACTIVATE-CLONE)
   - Wenn ein Home-Pubset aus mehr als einer Platte besteht, so müssen alle Platten gleichzeitig getrennt werden, um die Datenkonsistenz und Rekonstruierbarkeit zu gewährleisten. Metadaten sollten sich möglichst nur auf einer logischen Platte befinden.
   - Wenn ein Daten-Pubset aus mehr als einer Platte besteht, sollten sich Metadaten (Datei-, Benutzer-, GUARDS-Kataloge usw.) möglichst nur auf einer logischen Platte befinden. Alle Platten müssen zum Splitzeitpunkt gleichzeitig getrennt werden, um die Datenkonsistenz und Rekonstruierbarkeit zu gewährleisten.
5. Die Spiegelplatten wegschalten (Kommando DETACH-DEVICE).
   Dadurch ist der Standby-Pubset erzeugt, der im Fehlerfall genutzt werden kann.
6. Die Platten des Standby-Pubsets wieder zuschalten (Kommando ATTACH-DEVICE).
7. Eine erneute Synchronisation mit dem Pubset veranlassen, damit die Daten des Standby-Pubsets aktualisiert werden
   (Kommando RESTART-CLONE-SESSION).
8. Weiter bei Schritt 3.

Beispiel mit lokaler Spiegelung über Clones



Alternierende Standby-Pubsets einrichten

Bei Verwendung von nur einem Standby-Pubset existieren Zeitabschnitte, in denen kein Standby-Pubset zur Verfügung steht (während der Aktualisierung und dem erneuten Abtrennen). Bei kritischen Anwendungen empfiehlt sich deshalb der Einsatz von zwei alternierenden Standby-Pubsets, die abwechselnd bereit stehen und aktualisiert werden.

Vorgehensweise:

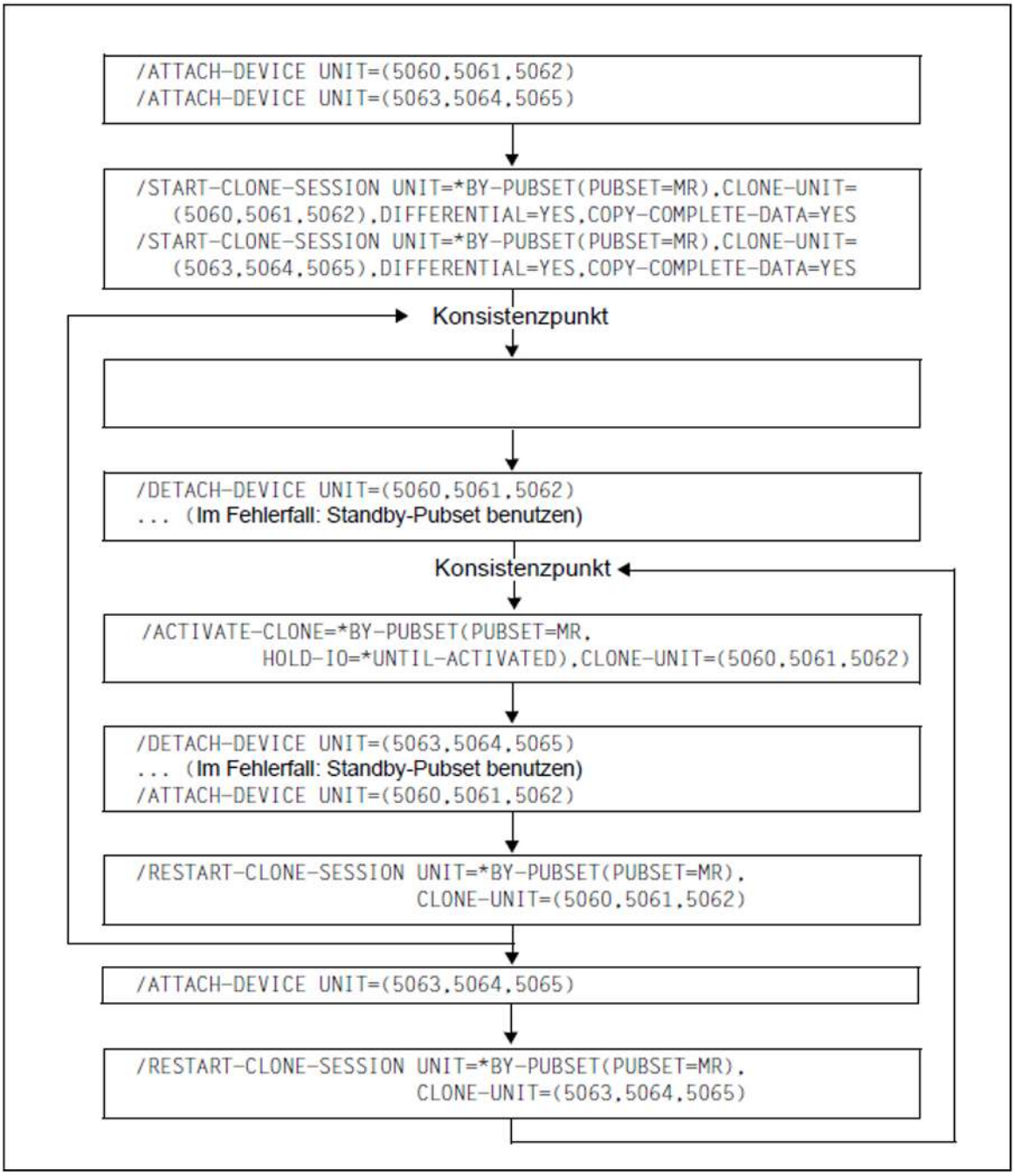
1. Die Platten zuschalten, die für die Spiegelung benötigt werden
   (Kommando ATTACH-DEVICE).
2. Jeder Platte (Unit) des Pubsets zweimal eine Spiegelplatte (Clone-Unit) zuordnen
   (Kommando START-CLONE-SESSION).
   Dadurch werden die Daten vom Pubset und auf die Spiegelplatten kopiert und dann synchron gehalten.
3. Konsistenzpunkt für den ersten Standby-Pubset erstellen.
   - Für Home-Pubsets bedeutet das:
     Host-Puffer für den Home-Pubset zurückschreiben und Anwendungen auf dem Home-Pubset anhalten oder beenden (DAB-Schreib-Caches sind auf dem Home-Pubset nicht erlaubt und müssen deshalb nicht berücksichtigt werden).
   - Für Daten-Pubsets muss eine der folgenden Aktionen ausgeführt werden:
     - Alle Dateien schließen, DAB-Schreib-Caches beenden und Host-Puffer anderer Anwendungen zurückschreiben. Dies kann dadurch erreicht werden, dass alle Anwendungen kurzzeitig beendet werden und der Pubset exportiert wird.
     - Die Anwendungen sorgen selbst für einen Konsistenzpunkt, ohne den Pubset zu exportieren (siehe Handbuch „SHC-OSD“ [48]).

4. Platten und Spiegelplatten für den ersten Standby-Pubsets trennen
   (Kommando ACTIVATE-CLONE-UNIT)

   - Wenn ein Home-Pubset aus mehr als einer Platte besteht, so müssen alle Platten gleichzeitig getrennt werden, um die Datenkonsistenz und Rekonstruierbarkeit zu gewährleisten. Metadaten sollten sich möglichst nur auf einer logischen Platte befinden.
   - Wenn ein Daten-Pubset aus mehr als einer Platte besteht, sollten sich Metadaten (Datei-, Benutzer-, Guards-Kataloge usw.) möglichst nur auf einer logischen Platte befinden. Alle Platten müssen zum Splitzeitpunkt gleichzeitig getrennt werden, um die Datenkonsistenz und Rekonstruierbarkeit zu gewährleisten.
5. Die Spiegelplatten für den ersten Standby-Pubsets wegschalten
   (Kommando DETACH-DEVICE).
   Dadurch ist der erste Standby-Pubset erzeugt, der im Fehlerfall genutzt werden kann.
6. Einen Konsistenzpunkt für den zweiten Standby-Pubset erzeugen.
7. Platten und Spiegelplatten für den zweiten Standby-Pubsets trennen
   (Kommando ACTIVATE-CLONE-UNIT)
8. Die Spiegelplatten für den zweiten Standby-Pubsets wegschalten
   (Kommando DETACH-DEVICE).
   Dadurch ist der zweite Standby-Pubset erzeugt, der aktuellere Daten als der erste hat und nun im Fehlerfall genutzt werden kann.
9. Die Platten des ersten Standby-Pubsets wieder zuschalten
   (Kommando ATTACH-DEVICE).
10. Eine erneute Synchronisation des ersten Standby-Pubsets mit dem Pubset veranlassen, damit die Daten des Standby-Pubsets aktuell bleiben

    (Kommando RESTART-CLONE-SESSION).
11. Für den ersten Standby-Pubset weiter bei Schritt 3.
12. Wenn der erste Standby-Pubset erneut abgetrennt ist, die Platten des zweiten wieder zuschalten
    (Kommando ATTACH-DEVICE).
13. Eine erneute Synchronisation des zweiten Standby-Pubsets mit dem Pubset veranlassen
    (Kommando RESTART-CLONE-SESSION).
14. Für den zweiten Standby-Pubset weiter bei Schritt 6.

Beispiel mit lokaler Spiegelung über Clones



Home-Pubset für Standby-System erzeugen

Innerhalb eines Hochverfügbarkeitsverbundes wird für den Einsatz des Standby-Systems ein eigener Home-Pubset benötigt. Dieser Home-Pubset, der den Charakter eines Stand-by-Pubsets besitzt, aber eine eigene Katalogkennung hat, kann ebenfalls mit den lokalen Replikationsfunktionen der Plattenspeichersysteme erstellt werden.

Die Erzeugung eines Home-Pubsets mit neuer Katalogkennung wird anhand eines Beispiels dargestellt. Folgende Ausgangssituation ist dabei gegeben:

- Der Pubset MR enthält ein ladefähiges BS2000 der aktuellen Version.

- Der Pubset MR besteht aus drei Platten (Units) mit den mnemotechnischen Gerätenamen 5070, 5071 und 5072.
- Das laufende BS2000-System wurde von dem Pubset MR hochgefahren.
- Für die Spiegelung stehen drei weitere Spiegelplatten zur Verfügung (5073, 5074 und 5075).

Vorgehensweise

1. Für den Pubset MR Kopien erzeugen
   (Clone-Units 5073, 5074 und 5075):

```
/START-CLONE-SESSION UNIT=*BY-PUBSET(PUBSET=MR),
   CLONE-UNIT=(5073,5074,5075),DIFFERENTIAL=YES,COPY-COMPLETE-DATA=YES
```

2. Konsistenzpunkt erstellen. Das bedeutet:
   Host-Puffer für den Home-Pubset zurückschreiben und Anwendungen auf dem Home-Pubset anhalten oder beenden (DAB-Schreib-Caches sind auf dem Home-Pubset nicht erlaubt und müssen deshalb nicht berücksichtigt werden).
3. Kopien abtrennen bzw. aktivieren und dabei den Pubset mit der neuen Katalogkennung WG erzeugen.

```
/ACTIVATE-CLONE UNIT=*BY-PUBSET(PUBSET=MR,NEW-PUBSET=WG,
   HOLD-IO=*UNTIL-ACTIVATED)
```

4. PVSREN aufrufen und mit PVSREN-Anweisungen die notwendigen Umbenennungen durchführen:

```
/PVSREN
. . .
//RENAME-PUBSET-OR-VOLUME-SET NAME=MR,NEW-NAME=WG,
  SYSID=173,MODE=*COMPLETE-SHC-RENAME
%  PVR0201 CHANGE DEFAULT CATALOG ID ENTRY 'TT' TO DEFAULT CATALOG ID
ENTRY 'WG' IN USER CATALOG OF HOME PUBSET ? (Y=YES; N=NO)?N
 %  PVR0202 CHANGE DEFAULT CATALOG ID ENTRY 'TT' TO DEFAULT CATALOG ID
ENTRY 'WG' IN USER CATALOG OF NEW PUBSET ? (Y=YES; N=NO)?Y
%  PVR0206 CHANGE CATID ENTRY 'TT' IN STANDARD IMON SCI OF HOME PUBSET TO
 CATID 'WG' ? (Y=YES; N=NO)?N
% PVR0203 READY PUBSET IN THIS BS2000 SYSTEM ? (Y=YES; N=NO)?N
//END
 %  PVR0145 PVSREN TERMINATED NORMALLY
```

Zu PVSREN siehe auch Abschnitt „SYSEAM-Speicherplatz auf Pubsets verwalten" bzw. die vollständige Beschreibung im Handbuch „Dienstprogramme“ [15].

> **i** Die aktuellen Kennwörter von wichtigen Benutzerkennungen (insbesondere TSOS) sollten aufbewahrt werden, da bei einem späteren Einsatz des neuen Home-Pubsets auf dem Standby-System der Benutzerkatalog und somit auch die Benutzerkennwörter „zurückgesetzt“ werden.
>
> Außerdem ist zu beachten dass innerhalb von Prozeduren und Programmen Dateizugriffe mit expliziter Angabe der Katalogkennung vermieden werden, da diese beim Einsatz des neuen Pubsets zu Fehlern führen.

9.9 SYSEAM-Speicherplatz auf Pubsets verwalten

SYSEAM-Dateien nehmen temporäre EAM-Dateien der Benutzer auf.

Temporär bedeutet, dass die EAM-Dateien der Benutzer nur „kurzfristig“ in einer SYSEAM-Datei gespeichert werden. Nach erfolgter Beendigung des Jobs sind die während eines Jobs erzeugten temporären EAM-Dateien wieder gelöscht.

Eine SYSEAM-Datei lässt sich auf jedem Pubset einrichten. Der Name lautet

`:<pvsid>:$TSOS.SYSEAM:`

`<pvsid>` bezeichnet den Pubset. Es existiert in diesem Falle genau eine SYSEAM-Datei.

Wenn von der Systembetreuung auf dem Standard-Pubset eines Benutzers eine SYSEAM-Datei mit entsprechendem Namen angelegt wurde, so greift der jeweilige Benutzer auf diese SYSEAM-Datei seines Standard-Pubsets zu. Existiert dagegen auf dem Standard-Pubset eines Benutzers keine SYSEAM-Datei, so erfolgt der Zugriff eines Benutzers auf die SYSEAM-Datei seines Home-Pubsets.

Werte für SYSEAM festlegen

Die Systembetreuung kann für eine SYSEAM-Datei pubset-spezifisch die folgenden Attribute festlegen, sie gelten dabei für alle auf einem Pubset erzeugten SYSEAM-Dateien in gleicher Weise:

- Minimal-Größe
- Wert für Sekundär-Allokierung
- Größe eines Cache für die SYSEAM-Datei im Klasse-4-Speicher (dieser wird allerdings ausschließlich für die SYSEAM-Datei auf dem Home-Pubset verwendet).

Diese Attribute werden mit Hilfe eines der Kommandos ADD-MASTER-CATALOG-ENTRY bzw. MODIFY-MASTER-CATALOG-ENTRY im betroffenen Master-Catalog-Entry eingetragen.

Eine SYSEAM-Datei wird (spätestens beim ersten Zugriff mit Hilfe der Zugriffsmethode EAM) mindestens in ihrer festgelegten Minimalgröße angelegt; reicht der Platz für eine SYSEAM-Datei nicht mehr aus, so wird die SYSEAM-Datei dynamisch um den Wert für ihre Sekundär-Allokierung erweitert, bis ihre Maximalgröße erreicht ist.
Bei zurückgehender EAM-Last wird überflüssig gewordener Platz in der SYSEAM-Datei in Stücken des Sekundär-Allokierungs-Wertes wieder dynamisch bis zur unteren Grenze freigegeben.

Voreinstellung durch Systemparameter

Wenn im Master-Catalog-Entry eines Pubsets keine Eintragungen für die oben genannten Kenngrößen einer SYSEAM-Datei hinterlegt sind, so werden für deren Minimal-Größe, Wert für Sekundär-Allokierung und Größe des Cache für die SYSEAM-Datei des Home-Pubset die Systemparameter EAMMIN, EAMSEC und EAMMEM wirksam. Darüber hinaus kann der für einen Benutzer allein zur Verfügung stehende Platz innerhalb einer SYSEAM-Datei mit dem Systemparameter EAMSIZ festgelegt werden.

Sie werden in der Startup-Parameterdatei im Abschnitt mit der Identifikation SYSOPT-CLASS2 festgelegt. Die Werte der genannten Systemparameter können angezeigt werden mit Hilfe des Kommandos SHOW-SYSTEM-PARAMETERS.

SYSEAM-Dateien auf Shared Pubsets

Die Behandlung von SYSEAM-Dateien auf einem Shared-Pubset wird durch den Systemparameter EAMSPVS gesteuert:

- entweder steht genau eine SYSEAM-Datei auf dem betroffenen Shared-Pubset (SPVS) für den Pubset-Master des Shared-Pubsets zur Verfügung (EAMSPVS=0). Die am Shared-Pubset-Verbund beteiligten Pubset-Slaves können diese SYSEAM-Datei nicht nutzen.
- oder es kann an jedem der am Shared-Pubset-Verbund beteiligten Systeme (d.h. sowohl am Pubset-Master als auch an jedem Pubset-Slave) mit einer SYSEAM-Datei auf diesem Shared-Pubset gearbeitet werden (EAMSPVS=1).

  Der Name der jeweiligen SYSEAM-Datei lautet `:<pvsid>:$TSOS.SYSEAM.<sysid>:.`

  `<sysid>` ist dabei eine 3-stellige Nummer und bezeichnet die Identifikation des importierenden Hosts innerhalb des Shared-Pubset-Verbunds („Systemidentifikation“). Sie kann mit Hilfe des Kommandos SHOW-PUBSET-ATTRIBUTES ermittelt werden.

  Auf dem durch `<pvsid>` bezeichneten Pubset existiert somit in diesem Falle für jedes importierende System eine SYSEAM-Datei mit dem Suffix `<sysid>`.

  Wenn in diesem Falle von der Systembetreuung eine Datei mit dem Namen `:<pvsid>:$TSOS.SYSEAM` angelegt wird, so wird diese nicht benutzt.

  Die Werte für EAMMIN, EAMSEC und EAMSIZ gelten dabei für alle auf einem Pubset erzeugten SYSEAM-Dateien in gleicher Weise.

9.10 Speed Catalog Access (SCA) einsetzen

SCA (Speed Catalog Access) ist ein Softwareprodukt zur Beschleunigung der Katalogverwaltung von SF-Pubsets. SCA ersetzt die sequenzielle Suche von Katalogeinträgen durch den direkten Zugriff. SCA verändert die Katalogstruktur nicht.

Realisiert wird diese Vorgehensweise mit Hilfe von Tabellen:

- Die Freelist-Tabelle gibt Aufschluss über freien Platz in den Katalogblöcken, Die Freelist-Tabelle wird von Routinen verwaltet, die unter der Aufrufertask ablaufen.
- Die Verweistabelle ordnet den Datei- bzw. Jobvariablennamen die logische Blocknummer des Katalogblockes zu, in dem der Eintrag steht. Die Verweistabelle wird, abhängig von der eingestellten Variante (siehe "Speed Catalog Access (SCA) einsetzen"), entweder von der SCA-Task oder von Routinen, die unter der Aufrufertask ablaufen, verwaltet.

Die SCA-Tasks verschiedener Kataloge können gleichzeitig gestartet und beendet werden.

Mit dem Einsatz von SCA kann eine Erhöhung des Gesamtdurchsatzes des Systems und eine höhere Flexibilität der Organisation im Data Center erreicht werden:

- Die Laufzeit für katalogintensive Programme (z.B. ARCHIVE) sowie das Antwortzeitverhalten bei Kommandos mit hohem Anteil an Katalogzugriffen (z.B. CREATE-FILE, MODIFY-FILE-ATTRIBUTES, IMPORT-FILE) kann erheblich verbessert werden.
- Es ist nicht mehr erforderlich ist, Dateien mit häufigem Katalogzugriff am Anfang einer Benutzerkette zu katalogisieren.

Folgendes sollte beachtet werden:

- Wesentliche Performanceverbesserungen sind nur zu erwarten, wenn große Katalogbelegungen der Benutzer vorliegen (> 60-100 Datei-Einträge pro Benutzerkennung).
- SCA benötigt zum Ablauf zusätzliche Betriebsmittel, wie z.B. virtuellen und realen Speicher, Tasks.
- Bei optimalem Aufbau der Dateikataloge kann der Einsatz von SCA auf wenige Pubsets reduziert werden.

Kommando	Bedeutung
ADD-MASTER-CATALOG-ENTRY	Eintrag im Katalogverzeichnis MRSCAT auf dem Home-Pubset erzeugen mit Festlegung, ob SCA bei IMPORT-PUBSET automatisch gestartet werden soll
ENTER-JOB	Starten und Beenden von SCA
MODIFY-MASTER-CATALOG-ENTRY (EDIT-MASTER-CATALOG-ENTRY)	Eintrag im Katalogverzeichnis MRSCAT auf dem Home-Pubset ändern mit Festlegung, ob SCA bei IMPORT-PUBSET automatisch gestartet werden soll

Tabelle 24: Kommandoübersicht zu SCA

SCA wird mit dem Installationsmonitor IMON installiert und benötigt unter der Kennung TSOS folgende Dateien

SYSLNK.SCA.<ver> bzw. SKMLNK.SCA.<ver>	SCA-Modulbibliothek für SU /390 SCA-Modulbibliothek für SU x86
SYSREP.SCA.<ver>	REP-Datei
SPEEDCAT.ENTER.START	Startprozedur
SPEEDCAT.ENTER.STOP	Beendigungsprozedur
SPEEDCAT.START	Startprogramm
SPEEDCAT.STOP	Beendigungsprogramm

SCA Ablauf-Varianten

Beim Starten von SCA für einen bestimmten Pubset kann der Aufrufer angeben, welche Variante des Speedcats geladen werden soll:

- SCA mit Taskwechsel
  Zum Starten von SCA wird eine eigene Task, die SCA-Task, aufgebaut. Sie erstellt die SCA-Tabellen und wartet auf Aufträge. Die SCA-Task kommuniziert mit der Benutzertask über eine Zweikammerbörse, um Anforderungen und Ergebnisse auszutauschen. Mit dieser Variante sind Performance-Einbußen durch den Taskwechsel in Kauf zu nehmen; der Klasse-4-Speicher wird allerdings entlastet, da die SCA-Directories im Klasse-5-Speicher gehalten werden (wo sie nur von der SCA-Task gelesen werden).

- SCA ohne Taskwechsel
  Alle SCA-Aufträge können ohne Taskwechsel durchgeführt werden. Zum Starten von SCA wird auch hier eine eigene SCA-Task aufgebaut, die sich jedoch nach Bereitstellung der Tabellen wieder beendet.
  Die Verweistabelle muss hier allen Benutzertasks zugänglich sein und wird zu diesem Zweck im Klasse-4-Speicher angelegt. Der Zugriff wird mittels exklusiver Sperren über eine Signalbörse (Einkammerbörse) reguliert, um zu verhindern, dass mehrere Tasks gleichzeitig lesend und schreibend auf die Directories zugreifen.

Implizites Starten von SCA bei IMPORT-PUBSET

Wenn SCA bereits bei der IMPORT-Verarbeitung ablauffähig sein soll, dann muss dies im MRSCAT-Eintrag des Pubsets hinterlegt werden. Mit dem Operanden START-SPEEDCAT im Kommando ADD-MASTER-CATALOG-ENTRY bzw. MODIFY-MASTER-CATALOG-ENTRY kann die Systembetreuung einen entsprechenden Eintrag erzeugen. Der Standardwert für das Starten von SCA im Kommando ADD-MASTER-CATALOG-ENTRY ist START-SPEEDCAT=*AUTOMATIC, d.h. SCA soll automatisch gestartet werden.

Explizites Starten von SCA während des Systemlaufs

1. Start mit Angabe der Katalogkennung:
   ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.START,JOB-NAME=SCA<catid> wobei <catid> die ein- oder mehrstellige Katalogkennung des Pubsets ist.
   Die SCA-Variante wird durch Beantwortung der Meldung `DMS03FE` eingestellt.

2. Start ohne Angabe der Katalogkennung:
   ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.START wobei die gewünschte Katalogkennung des Pubsets an der Konsole erfragt wird.
   Die SCA-Variante wird durch zusätzliche Beantwortung der Meldung `DMS03FE` eingestellt.

Dabei sollte beachtet werden, dass SCA erst dann für einen Pubset gestartet werden sollte, wenn der Pubset vollständig importiert ist. Das Kommando zum Starten von SCA während des Systemlaufs kann auch Bestandteil der CMDFILE sein.

Beenden von SCA

1. Beenden mit Angabe der Katalogkennung:
   ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.STOP,JOB-NAME=SCA<catid> wobei <catid> die ein- oder mehrstellige Katalogkennung des Pubsets ist.
2. Beenden ohne Angabe der Katalogkennung:
   ENTER-JOB FROM-FILE=SPEEDCAT.ENTER.STOP wobei die Katalogkennung des Pubsets an der Konsole erfragt wird.
3. Implizites Beenden:
   Die Kommandos EXPORT-PUBSET und SHUTDOWN beenden die SCA-Task bzw. geben den Klasse-4-Speicher der Verweistabelle und der Freelist frei.

Das Kommando zum Beenden von SCA kann in einer RUN-Kommandofolge abgelegt sein.

Ohne Angabe der Katalogkennung (2.) können die Kommandos auch vom Operator gegeben werden.

Fehlerbehandlung

Bei Auftreten eines Fehlers in SCA bzw. bei Hardware-bedingten Schreibfehlern im Katalog schaltet sich SCA automatisch aus und setzt den normalen Katalogzugriff wieder in Kraft. Je nach Art des Fehlers wird ein Systemdump initiiert.

SCA benutzt die SERSLOG-Funktion (siehe „Diagnosehandbuch“ [14]) zur Ausgabe von Diagnoseinformationen. Darüber hinaus werden Fehlermeldungen an der Konsole ausgegeben.

9.11 Hardware-Fehler auf Pubsets erfassen und beseitigen

Bei Ausfall eines Datenträgers bzw. bei einem partiellen Plattendefekt in einem SF-Pubset oder einem Volume-Set kann mit diesem Pubset/Volume-Set nicht mehr kontinuierlich gearbeitet werden.

Der Fehler kann nur durch eine Neuinitialisierung des betroffenen Datenträgers behoben werden. Damit verbunden ist der Verlust sämtlicher Daten, die sich auf der Platte befinden. Der Pubset kann nur nach einer vollständigen Rekonstruktion aus der Sicherung wieder in Betrieb genommen werden.
Es ist daher notwendig, solche Plattenbereiche rechtzeitig zu erkennen, um sie bei künftigen Speicherplatzanforderungen auszusparen. Nach der Erfassung des Plattenfehlers kann dann eine geeignete Maßnahme zur Beseitigung des Fehlers eingeleitet werden.

Erfassen von Hardware-Fehlern durch das DVS

Der DVS-Kontrollmechanismus sorgt bei gemeinschaftlichen Datenträgern (Pubsets) dafür, dass permanente Plattenfehler, die während eines Plattenzugriffs durch das DVS auftreten, der Allokierungskomponente bekanntgemacht und von der weiteren Vergabe von Plattenspeicher ausgenommen werden. Es wird damit sichergestellt, dass diese registrierten Plattenbereiche bei künftigen Speicherplatzanforderungen nicht mehr berücksichtigt werden und defekte Plattenbereiche nicht mehr auf andere Dateien übergehen.
Die Informationen über die defekten Plattenbereiche werden in der vom System erzeugten, pubset-spezifischen Defect-Garbage-Datei hinterlegt.
Bei der erstmaligen Registrierung eines Plattenfehlers innerhalb eines Pubsets wird die Datei auf diesem Pubset unter folgendem, vom Pubset-Typ abhängigen Pfadnamen erzeugt:

- bei SF-Pubsets: :<catid>:$TSOS.SYSDAT.DEFECT.GARBAGE.<catid#>
- bei SM-Pubsets: :<catid>:$TSOS.SYSDAT.DEFECT.GARBAGE.<volume-set-id#>

<volume-set-id#> bezeichnet die Katalogkennung des Volume-Sets, auf dem sich die Datei befindet. Ist die Katalogkennung des SF-Pubsets bzw. des Volume-Sets kleiner als vier Zeichen, werden die Variablen <catid#> und <volume-set-id#> auf vier Stellen aufgefüllt.

Die Extentliste dieser Datei besteht aus genau den Extents, die die gemeldeten defekten Plattenblöcke des Pubsets umfassen. Das bedeutet, dass die entsprechenden defekten Blöcke für die Lebensdauer der betroffenen Datei doppelt belegt sind, nämlich einmal in der Datei selbst und einmal in der Defect-Garbage-Datei.

Die Defect-Garbage-Datei darf ausschließlich durch die zuständigen Systemkomponenten bearbeitet werden, jeglicher Zugriff wird abgewiesen. Um einen Zugriff zu verhindern, wird ein Dateikennwort vergeben und das Dateischutzattribut ACCESS=READ eingestellt.
Die Defect-Garbage-Datei wird mit dem Backup-Level E versehen, d.h. sie wird nicht durch ARCHIVE oder HSMS gesichert. Damit werden inkonsistente Zustände vermieden.

Die Extentliste der Defect-Garbage-Datei hat – wie bei allen anderen Dateien – nur eine beschränkte Kapazität. Es können, je nach Plattenverteilung, ca. 140 bis 310 Extents aufgenommen werden, was dann der max. Anzahl von fehlerhaften Blöcken pro Pubset entspricht. Bei Überschreitung der Kapazität wird an der Konsole die Meldung `DMS060C` ausgegeben. Tritt dieser Fall ein, muss dringend eine Reorganisation des Pubsets mit der Beseitigung der Plattendefekte (z.B. mit VOLIN) durchgeführt werden.

Eine erzeugte Defect-Garbage-Datei kann ausschließlich durch die IMPORT-PUBSET-Verarbeitung des entsprechenden Pubsets gelöscht werden. Während der IMPORT-PUBSET-Verarbeitung wird geprüft, ob eine Defect-Garbage-Datei vorhanden ist und wenn ja, ob Blöcke daraus bereits repariert wurden. In diesem Fall können die reparierten Blöcke wieder in den normalen Prozess der Plattenspeicherplatzvergabe integriert werden. Das erfolgt implizit in der IMPORT-PUBSET-Verarbeitung über eine F5-Rekonstruktion. Existieren keine (registrierten) defekten Blöcke mehr für diesen Pubset, wird die Defect-Garbage-Datei gelöscht.

> **!** **ACHTUNG!**
> Die Defect-Garbage-Datei darf in keinem Fall von Hand gelöscht werden; dies muss ausschließlich den Mechanismen des Systems vorbehalten bleiben.
> Anderenfalls wären Datenverlust und Zerstörung der F5-Etiketten die Folge!

Bei der F5-Rekonstruktion eines Pubsets während des Import-Vorgangs kann es zu Doppelallokierungen zwischen der Defect-Garbage-Datei und anderen Dateien kommen (Meldungen `DMS0461` und `DMS0462`). Die Meldungen sind jedoch nicht als Hinweis auf Systemfehler oder Inkonsistenzen im F5-Bereich zu verstehen, sie weisen lediglich auf defekte Plattenbereiche im Allokierungsbereich der gemeldeten Dateien hin.

Eine weitere mögliche Fehlermeldung an der Konsole im laufenden Pubset-Betrieb ist `DMS0608`. In diesem Fall wurde gerade ein defekter Block gemeldet. Beim Versuch, ihn in die Defect-Garbage-Datei aufzunehmen, tritt folgender Fehler auf: Es existiert eine Datei mit der zu einer Defect-Garbage-Datei gehörenden Namensstruktur (siehe oben), ihre innere Struktur (Katalogattribute) weicht jedoch von der einer normalen Defect-Garbage-Datei ab. Die Meldung `DMS0608` muss beantwortet werden. Dabei kann versucht werden, die existierende Datei als Defect-Garbage-Datei zu verwenden. Gelingt dies nicht, kann die Funktion abgebrochen werden, d.h. der defekte Block wird nicht erfasst, oder es wird eine neue Defect-Garbage-Datei vom System erzeugt und die existierende implizit gelöscht.

Manuelle Allokierungssperre vergeben

Eine Allokierung auf defekten Platten kann für SF- und für SM-Pubsets volume-spezifisch und für SM-Pubsets auch volume-set-spezifisch verboten werden (siehe Kommando MODIFY-PUBSET-RESTRICTIONS).
Bereits beim Importieren von SM-Pubsets besteht die Möglichkeit, die Allokierung von Volume-Sets mit defekten Platten zu verhindern. Die Volume-Sets können dabei in den Zustand „defect“ oder „in hold“ versetzt werden (siehe Kommando IMPORT-PUBSET, Operanden DEFECT-VOLUME-SET bzw. IN-HOLD-VOLUME-SET).

10 Net-Storage-Verwaltung

BS2000 ermöglicht den Zugang zu UNIX-Dateisystemen (UFS) über NFS.

Dateien von BS2000 und von offenen Systemen können im Server-Netzwerk auf dem freigegebenen Speicherplatz von File-Servern abgelegt und bearbeitet werden.

10.1 Überblick

Bei der Definition von Platten unterscheidet das DVS gemeinschaftliche Platten (Public Volumes), die zu Pubsets zusammengestellt werden, und private Platten (Private Disks). Daneben gibt es noch das Speichermedium Net-Storage. Net-Storage-Volumes werden als Erweiterung zu Pubsets genutzt.
Einem Pubset können Net-Storage-Volumes zugeordnet werden. Sie sind aber nicht Bestandteil des Pubsets (IMPORT-PUBSET ist auch dann möglich, wenn auf die zugeordneten Net-Storage-Volumes nicht zugegriffen werden kann).

Jeder Pubset enthält einen Benutzerkatalog und seinen eigenen Dateikatalog. Die Angabe NET-STORAGE-USAGE=*ALLOWED im Benutzereintrag legt fest, dass der Benutzer Speicherplatz auf Net-Storage belegen darf.

Alle Datenträger werden in BS2000 durch einen Namen identifiziert, der bis zu sechs Zeichen lang sein darf. Dieser Name wird als VSN (Volume Serial Number) oder auch Archivnummer bezeichnet. Der gültige Zeichenvorrat besteht aus den alphanumerischen Zeichen A..Z, 0..9 und den Sonderzeichen Punkt, @, # und $. Pubset-Volumes sind von anderen Datenträgern durch ihre VSN unterscheidbar, wobei die VSN gemeinschaftlicher Platten einer Konvention unterliegt: Sie muss mit der Zeichenkette „PUB“ beginnen oder einen Punkt an der dritten, vierten oder fünften Stelle enthalten. Diese VSN-Syntax darf für private Datenträger und Net-Storage-Volumes nicht angewandt werden.

Standard-Namen von Net-Storage-Volumes, die einem Pubset zugeordnet sind, haben, abhängig von der Pubset-Notation, eine besondere Form, siehe "Notation von Net-Storage-Volumes" den Abschnitt „Begriffe".

Das DVS übernimmt die Speicherplatzverwaltung auf Platten. Einige Kontrollfunktionen (Zugriffsberechtigung, Sättigungssteuerung, usw.) sind nur auf einem Pubset anwendbar, nicht aber auf privaten Platten oder auf Net-Storage.

BS2000-Anwender können Net-Storage-Volumes nach ihrer Zuordnung zu einem lokalen Pubset über die Schnittstellen des DVS nutzen, wenn dies in ihrem Benutzereintrag erlaubt ist. Der verfügbare Speicherplatz auf Net-Storage ist dabei nur durch die Größe des freigegebenen Speicherbereichs auf dem Net-Server begrenzt. Siehe das Handbuch „Einführung in das DVS“ [19].

Dateien auf Net-Storage können sowohl von BS2000 als auch von offenen Systemen angelegt und gemeinsam bearbeitet werden, siehe Abschnitt „Interoperabilität auf Net-Storage". Die detaillierten Hard- und Software-Voraussetzungen dafür finden Sie in der aktuellen Freigabemitteilung von BS2000.

10.2 Begriffe

Folgende Begriffe werden in BS2000 beim Arbeiten mit Net-Storage verwendet:

Net-Server

File-Server im weltweiten Rechnernetz, der Speicherplatz (Network Attached Storage, NAS) für die Nutzung durch andere Server bereitstellt und entsprechende File-Server-Dienste anbietet.

Net-Storage

Der von einem Net-Server im Rechnernetz bereitgestellte und zur Nutzung durch fremde Server freigegebene Speicherplatz. Net-Storage kann ein Dateisystem oder auch nur ein Knoten im Dateisystem des Net-Servers sein.

Net-Storage wird mit einem BS2000-System über das Kommando MOUNT-NET-STORAGE verbunden. Der Net-Storage (genauer: das freigegebene Verzeichnis) an sich wird dabei am Net-Client eingehängt. Mit dem Kommando UMOUNT-NET-STORAGE wird der Net-Storage wieder vom BS2000-System getrennt und am Net-Client ausgehängt.

Informationen über den im BS2000-System verfügbaren Net-Storage können mit dem Kommando SHOW-NET-STORAGE angefordert werden.

Als Net-Storage-Systeme sind freigegeben: ETERNUS CS mit NAS Schnittstelle und NetApp ONTAP Systeme. Andere Net-Server benötigen eine Sonderfreigabe.

Net-Client

Realisiert den Zugriff auf Net-Storage für das nutzende Betriebssystem.

In BS2000 transformiert der Net-Client zusammen mit dem BS2000-Subsystem ONETSTOR die BS2000-Dateizugriffe in entsprechende UNIX-Dateizugriffe und führt sie über NFS auf dem Net-Server aus.

Net-Client bei SU /390 ist der HNC, bei SU x86 das Trägersystem X2000.

Net-Storage-Volume

Net-Storage-Volumes repräsentieren Net-Storage in BS2000 im Rahmen von Daten-Pubsets.

Mit dem Kommando ADD-NET-STORAGE-VOLUME werden Net-Storage-Volumes eingerichtet und einem Pubset zugeordnet. Dabei wird ein Verzeichnis im freigegebenen Dateisystem des Net-Servers angelegt und einem lokalen Daten-Pubset (SF- oder SM-Pubset) in Form eines Net-Storage-Volumes zugeordnet. Ab BS2000 V21 werden nur Net-Storage-Volumes vom Typ NETVOL erzeugt. Bereits existierende Net-Storage-Volumes aus niedrigeren BS2000-Versionen vom Typ NETSTOR werden in BS2000 V21 auch weiterhin unterstützt. Weitere Hinweise finden Sie im Kapitel Net-Storage-Volumes ab BS2000 V21 (Einführung in die Systembetreuung SE Server , #210.1).

Sowohl Net-Storage-Volumes vom Typ NETVOL als auch Net-Storage-Volumes vom Typ NETSTOR werden durch ihre Volume Serial Number (VSN) und aus Kompatibilitätsgründen den Typ NETSTOR angesprochen. Der Verzeichnisname im freigegebenen Dateisystem des Net-Servers und die VSN des Net-Storage-Volumes sind gleich.

Ein Net-Storage-Volume (genauer: das Verzeichnis mit dem Namen des Net-Storage-Volumes auf dem Net-Server) enthält folgende Dateien (siehe auch die nächste Seite):

- ein File-System-Label (Dateiname .FSL)

- bei Net-Storage-Volumes vom Typ NETSTOR einen Dateikatalog (Dateiname .BS2FSCAT) mit den Metadaten der auf Net-Storage abgelegten Dateien.
  Net-Storage-Volumes vom Typ NETVOL besitzen keinen Dateikatalog.
- die Dateien der Benutzer

Informationen über den in einem Daten-Pubset verfügbaren Net-Storage können mit dem Kommando SHOW-PUBSET-NET-STORAGE angefordert werden.

Notation von Net-Storage-Volumes

Unabhängig vom Pubset-Typ leitet sich die VSN des Net-Storage-Volumes standardmäßig von dem Pubset-Namen ab:

- Bei einem einstelligen Pubset-Namen wird die PUB-Notation (z.B. `PUBA00`) angenommen und die VSN des Net-Storage-Volumes wird durch Ersetzen des „U“ durch „@“ in der Zeichenfolge „PUB“ gebildet (z.B. `P@BA00`).
- Bei einem mehrstelligen Pubset-Namen wird die Punkt-Notation (z.B. `OTTO.0, ABC.00`) angenommen und die VSN des Net-Storage-Volumes wird durch Ersetzen des Punktes durch „@“ gebildet (z.B. `OTTO@0, ABC@00`).

Mit dieser Notation kann nur das Standard-Net-Storage-Volume im Pubset bezeichnet werden.

Mehrere Net-Storage-Volumes für einen Pubset

Wenn einem Pubset mehrere Net-Storage-Bereiche auf einem oder mehreren Net-Servern zugeordnet werden sollen, dann müssen die Net-Storage-Volumes mit unterschiedlichen VSNs zugeordnet werden. Da es pro Pubset nur ein Net-Storage-Volume mit Standard-Namen geben kann, müssen weitere Net-Storage-Volumes dem Pubset mit explizit definierten VSNs (wie bei Privatplatten) zugeordnet werden.

Net-Storage-Datei

Bezeichnet eine Datei, die auf einem Net-Storage-Volume angelegt ist. Auf Net-Storage wird zwischen den zwei Dateitypen BS2000-Datei und Node-File unterschieden.

BS2000-Datei

Bezeichnet eine Datei, die ausschließlich von BS2000 angelegt und bearbeitet wird. BS2000-Dateien auf Net-Storage (FILE-TYPE=BS2000) werden bedient. Sie liegen direkt auf einem Net-Storage-Volume. Offene Systeme dürfen ausschließlich lesend darauf zugreifen.

Node-File

Bezeichnet eine Net-Storage-Datei (FILE-TYPE=NODE-FILE), die sowohl von BS2000 als auch von offenen Systemen angelegt und bearbeitet werden kann. Sie liegen auf einem Net-Storage-Volume in einem benutzerspezifischen Verzeichnis (Name der Benutzerkennung) und die Dateinamen entsprechen den BS2000-Namenskonventionen. BS2000 bedient PAM-Node-Files und SAM-Node-Files.

10.3 Interoperabilität auf Net-Storage

Sowohl BS2000 als auch offene Systeme können sich mit demselben Net-Storage verbinden, dort Node-Files anlegen und diese wechselseitig bearbeiten:

- BS2000 kann Node-Files auf Net-Storage anlegen und bearbeiten - offene Systeme können diese Node-Files ebenfalls bearbeiten
- offene Systeme können Node-Files auf Net-Storage anlegen und bearbeiten - BS2000 kann diese Node-Files importieren und ebenfalls bearbeiten

Dies wird als **Interoperabilität** bezeichnet.

Voraussetzungen

Für das gemeinsame Bearbeiten von Node-Files müssen folgende zusätzlichen Voraussetzungen erfüllt sein:

- Net-Server: ETERNUS CS mit NAS Schnittstelle und NetApp ONTAP Systeme.

  > ¡ Andere Net-Server benötigen eine Sonderfreigabe.

- Für jeden BS2000-Benutzer, der mit Node-Files arbeiten möchte, müssen in der Benutzerverwaltung des Pubsets die POSIX-Benutzerattribute UserID (USER-NUMBER, <uid>) und GroupID (GROUP-NUMBER, <gid>) eingetragen werden (siehe "Verwalten von Net-Storage in BS2000").
- Bei Anschluss von Net-Storage über NFSv4 müssen die Benutzer in einem Verzeichnisdienst (`openLDAP` oder `AD`) mit der entsprechenden UserID und GroupID eingetragen werden. Im Net-Client und im Net-Server muss der Verzeichnisdienst konfiguriert sein. Bei Anschluss über NFSv3 ist die Verwendung eines Verzeichnisdienstes nicht nötig.

Verfahren

Node-Files werden in BS2000 im UNIX-Dateisystem (UFS) in benutzerspezifischen Verzeichnissen angelegt, siehe "Verwalten von Net-Storage in BS2000".

BS2000 bedient folgende Typen von Node-Files:

- PAM-Node-Files (BLKCTRL=NO)
- SAM-Node-Files (BLKCTRL=DATA, RECFORM=V).
  SAM-Node-Files dienen der textbasierten Verarbeitung.

  Beim Schreiben der Daten von BS2000 zum Net-Storage werden die SAM-Strukturen aus dem Datenstrom entfernt und, falls gewünscht, eine Code-Konvertierung von EBCDIC nach ASCII (ISO) durchgeführt.

  Beim Lesen der Daten vom Net-Storage zum BS2000 werden die SAM-Strukturen eingesetzt und die Daten in EBCDIC konvertiert, so dass die Zugriffsmethode SAM wieder die gewohnten SAM-Datenstrukturen vorfindet. Die Konvertierung richtet sich nach den Einstellungen von Coded-Character-Set und Net-Coded-Character-Set der Datei, die der Benutzer beim Anlegen der Datei bestimmt. Wenn der Benutzer keine Angaben macht, dann werden die Standardwerte dem Benutzereintrag im Benutzerkatalog des Pubsets entnommen, auf dem die Datei katalogisiert wird. Diese Standardwerte werden beim Anlegen eines neuen Benutzers den Systemparametern HOSTCODE bzw. NETCODE entnommen.

Node-Files, die im laufenden Betrieb von offenen Systemen angelegt werden, müssen vor ihrer Benutzung in BS2000 importiert werden. Siehe die Kommandobeschreibung von IMPORT-NODE-FILE im Handbuch „Kommandos“ [27].

Informationen über Node-Files liefert das Kommando LIST-NODE-FILES.

Node-Files können mit dem Kommando EXPORT-NODE-FILE wieder aus dem BS2000-Dateikatalog entfernt werden.

Randbedingungen

Für die Interoperabilität auf Net-Storage sind Randbedingungen zu beachten, siehe Abschnitt „Interoperabilität".

10.4 Anbindung von BS2000 an Net-Storage

BS2000 greift auf externe Speichersysteme über Fibre Channel und ein Storage Area Network (SAN) zu.

BS2000 bietet mit der Bedienung von Net-Storage auch den Zugang zu Net-Servern und dem daran angeschlossenen Network Attached Storage (NAS).

Auf SU /390 erfolgt der Zugriff zu den Net-Servern über den Net-Client HNC.
Auf SU x86 wird diese Rolle von X2000 übernommen.

Hinweise zu freigegebenen NAS-Servern finden Sie in den aktuellen Freigabemitteilungen.

10.5 Zugriff auf Net-Storage

Folgende Systeme können auf Net-Storage zugreifen:

- BS2000 Vollzugriff auf alle importierten BS2000-Dateien und Node-Files
- offene Systeme: Vollzugriff auf Node-Files, lesender Zugriff auf BS2000-Dateien

10.5.1 Zugriff von BS2000 auf Net-Storage

Der Zugriff auf eine Datei auf Net-Storage durch den BS2000-Anwender erfolgt über DVS-Schnittstellen und den Net-Client auf dem Net-Server in folgenden Schritten:

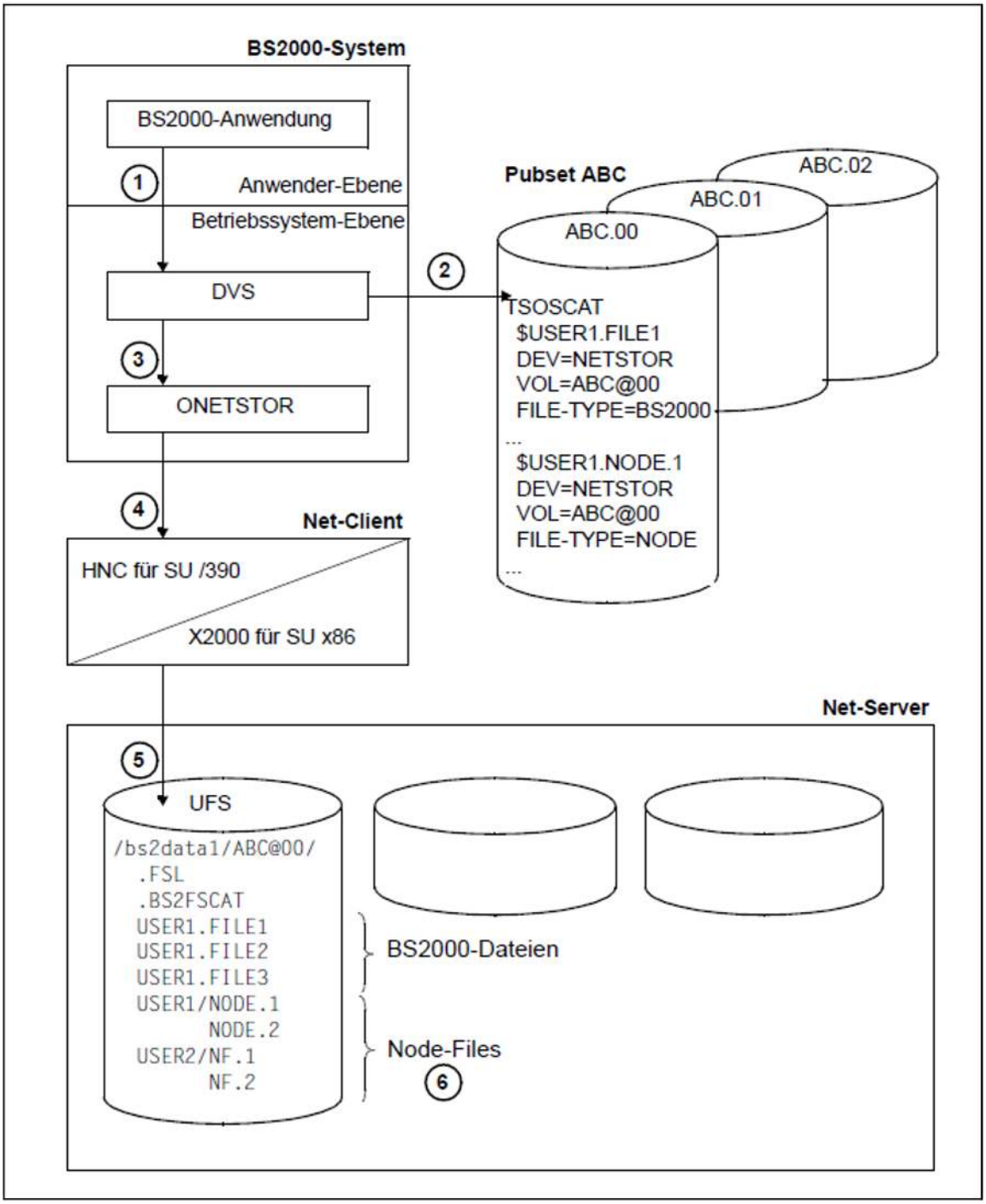


Bild 14: BS2000-Zugriff auf Net-Storage vom Typ NETSTOR

1. Eine BS2000-Anwendung unter der Benutzerkennung USER1 möchte auf die Datei FILE1, die auf Net-Storage-Volume vom Typ NETSTOR liegt und im Pubset ABC katalogisiert ist, zugreifen. Dies geschieht über die üblichen Benutzer-Schnittstellen des DVS.

2. Das DVS prüft die Existenz der Datei im Benutzer- und Dateikatalog des Pubsets ABC. Anhand der Dateiattribute DEVICE=NETSTOR und VOLUME=ABC@00 erkennt das DVS, dass es sich um eine Datei im Verzeichnis ABC@00 auf dem vom Net-Server freigegebenen Net-Storage handelt (siehe den Abschnitt „Net-Storage mit BS2000 verbinden").
3. Der eigentliche Zugriff auf die Datei FILE1 erfolgt über das BS2000-Subsystem ONETSTOR und den Net-Client.
4. Das BS2000-Subsystem ONETSTOR transformiert den BS2000-Dateizugriff in den entsprechenden Dateizugriff im UNIX-Dateisystem (UFS) und reicht ihn an den Net-Client weiter.
5. Auf dem Net-Server und dem Net-Client müssen die nötigen Vorbereitungen getroffen sein, siehe den Abschnitt „Vorbereitungen auf Net-Server und Net-Client". Am Net-Client ist das vom Net-Server freigegebenen Verzeichnis `/bs2data1` (mit Unterverzeichnis ABC@00) eingehängt, siehe Abschnitt „Net-Storage mit BS2000 verbinden". Der Dateizugriff erfolgt über NFS im UFS des Net-Servers.
6. Beim Zugriff auf ein Node-File, z.B. die Datei NODE.1, erkennt das DVS, dass es sich um ein Node-File (FILE-TYPE=NODE) im Verzeichnis ABC@00 auf dem vom Net-Server freigegebenen Net-Storage handelt. Node-Files liegen in einem benutzerspezifischen Verzeichnis, das den Namen der Benutzerkennung trägt, hier das Verzeichnis USER1. Dieses Verzeichnis wird automatisch angelegt, wenn BS2000 das erste Node-File des Benutzers anlegt.
   Damit ein Benutzer mit Node-Files arbeiten kann, müssen im Benutzereintrag des Pubsets die POSIX-Benutzerattribute UserID und GroupID eingetragen sein. Für den Benutzer USER1 ist z.B. `1005:100` eingetragen (siehe Bild 15, "Zugriff offener Systeme auf Net-Storage").

10.5.2 Zugriff offener Systeme auf Net-Storage

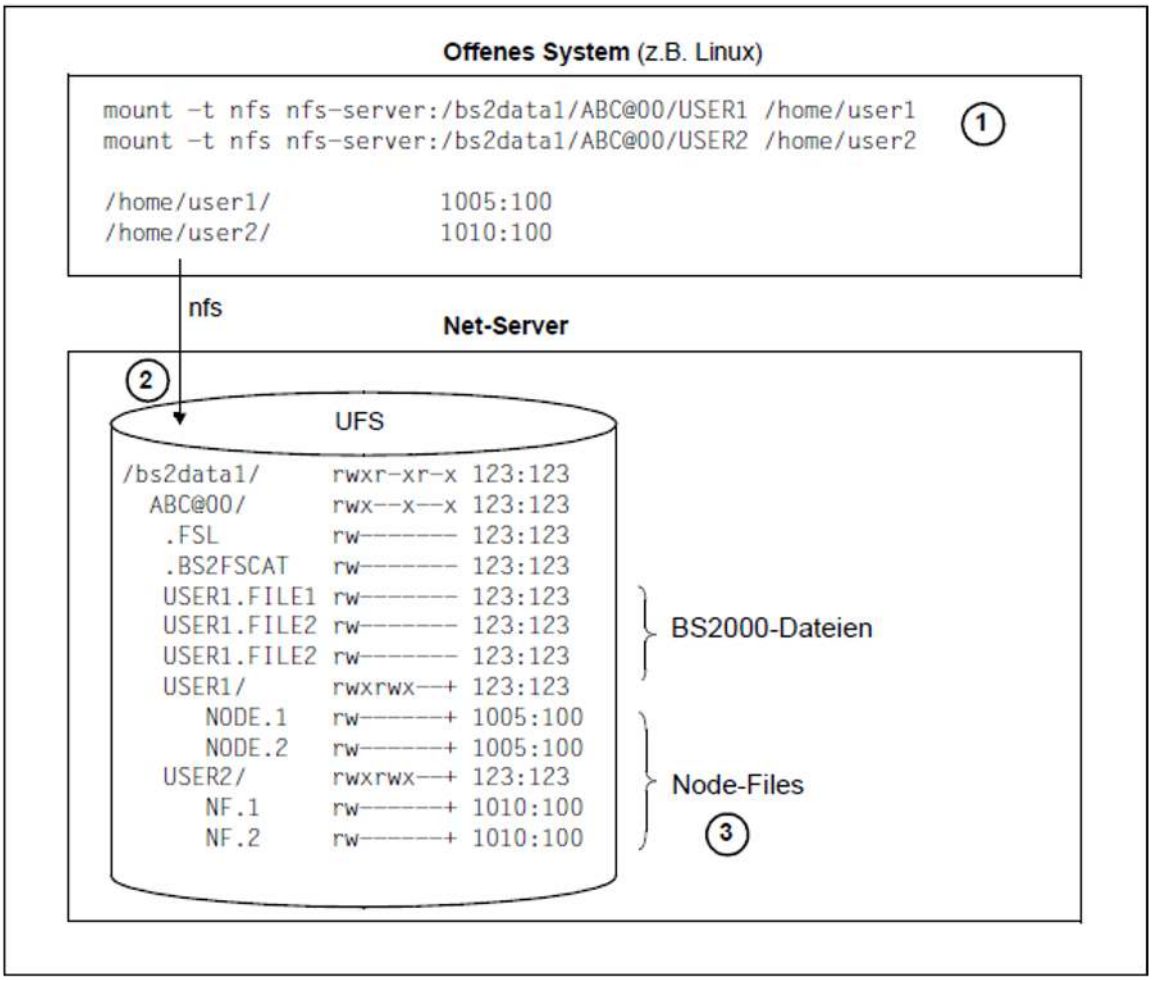


Bild 15: Zugriff offener Systeme auf Net-Storage vom Typ NETSTOR

1. Offene Systeme erhalten Zugang zu BS2000-Dateien und Node-Files auf dem Net-Server durch entsprechende Mount-Kommandos.
2. Beim Anlegen der benutzerspezifischen Verzeichnisse werden von BS2000 auch die POSIX-ACLs entsprechend versorgt. Unter Linux erhalten deshalb im Beispiel oben die Benutzer `user1` und `user2` Zugriff auf das Verzeichnis USER1 und USER2.
3. Damit ein Benutzer sowohl von BS2000 als auch vom offenen System aus lesenden und schreibenden Zugriff hat, muss für ihn auf beiden Systemen dieselbe UserId und GroupId eingetragen sein. Das Gleiche gilt für das Anlegen und Löschen von Dateien.

> ¡ Mit dem Einsatz von NFSv4 muss dies über einen Verzeichnisdienst, z.B. openLDAP oder Active Directory, gewährleistet werden, an den der NFS-Server und das offene System angeschlossen sind.

10.6 Vorbereitungen auf Net-Server und Net-Client

Der Net-Storage-Anschluss muss sowohl auf dem Net-Server als auch auf dem Net-Client konfiguriert werden.

Der Systemverwalter des Net-Servers muss das zum Montieren freigegebene Verzeichnis einem Benutzer mit UserID und GroupID (Eigentümer) zuordnen. Dieser Benutzer hat die Zugriffsberechtigung auf das freigegebene Verzeichnis.

Details dazu finden Sie in der Konfigurationsbeschreibung des Net-Servers.

> ¡ Für den BS2000-Zugriff sind stets Schreibrechte notwendig!

Die Instanz, die im Auftrag von BS2000 auf das im Net-Server freigegebene Verzeichnis zugreift, ist der bs2netsagent im Net-Client.

Der Administrator des SE Servers muss über den SE Manager am Net-Client die Zugangsberechtigung (Anwender ID und Gruppen ID) des bs2netsagent einstellen, damit der Net-Server den Zugriff auf das freigegebene Verzeichnis erlaubt. Details dazu finden Sie im Handbuch „Bedienen und Verwalten“ [57].

Da dem bs2netsagent nur eine UserId und GroupId zugeordnet werden kann, müssen auch auf den Net-Servern allen für diesen Net-Client freigebenen Verzeichnissen diese eine UserId und GroupId zugeordnet werden.

> ¡ Achten Sie bei Wartungsarbeiten (z.B. Software-Update oder Neuinstallation) und Konfigurationsänderungen unbedingt darauf, dass ein Net-Storage, der über den betreffenden Net-Client betrieben wird, vorher mit UMOUNT-NET-STORAGE von allen angeschlossenen BS2000-Systemen getrennt wird.

10.7 Net-Storage mit BS2000 verbinden

Um Net-Storage in BS2000 nutzen zu können, muss die Systembetreuung (nach den Vorbereitungen auf Net-Server und Net-Client) zunächst das BS2000-System mit dem Net-Storage verbinden sowie Net-Storage-Volumes einrichten und einem lokalen Pubset zuordnen.

- Verbinden Sie das BS2000-System mit dem Net-Storage, z.B. mit

```
/MOUNT-NET-STORAGE DIRECTORY='/bs2data1',
                   SERVER=*IP-ADDRESS(IP-ADDRESS=172.10.1.110),
                   CLIENT=system1
```

  Dieses Kommando führt folgende Bearbeitungsschritte aus:

  - Einhängen des freigegebenen Verzeichnisses am Net-Client
  - Bestandsaufnahme des freigegebenen Verzeichnisses, d.h. alle bereits vorhandenen 6-stelligen Verzeichnisnamen unterhalb dieses Verzeichnisses werden als Volume in die NDM-Tabellen aufgenommen
  - ggf. Belegung des/der Volumes

Für das Einrichten und Zuordnen von Net-Storage-Volumes haben Sie zwei Möglichkeiten:

1. Richten Sie ein neues Net-Storage-Volume ein und weisen Sie es einem lokalen Pubset zu, z.B. mit

```
/ADD-NET-STORAGE-VOLUME VOLUME=*STD, PUBSET=A,
                        DIRECTORY='/bs2data1',
                        SERVER=*IP-ADDRESS(IP-ADDRESS=172.10.1.110)
```

   Während in früheren BS2000 Versionen Net-Storage-Volumes vom Typ NETSTOR erzeugt wurden, werden ab der aktuellen BS2000 V21.0A nur Net-Storage-Volumes vom Typ NETVOL eingerichtet. Das Net-Storage-Volume steht nun als Volume mit der Standard-VSN `P@BA00` des angegebenen Pubsets für die Nutzung im BS2000-Betrieb zur Verfügung. Im Verzeichnis `bs2data1` des Net-Storage wird ein neues Unterverzeichnis `P@BA00` angelegt. In diesem Unterverzeichnis wird unter anderem das File-System-Label abgelegt.
   Das Net-Storage-Volume kann nun mit Dateien beschrieben werden.

2. Richten Sie ein bestehendes Verzeichnis auf einem Net-Storage-Volume für den BS2000-Betrieb ein, z.B. mit

```
/ADD-NET-STORAGE-VOLUME VOLUME=MYNETS, PUBSET=A,
                        DIRECTORY='/bs2data1',
                        SERVER=*IP-ADDRESS(IP-ADDRESS=172.10.1.110)
                        IMPORT=*YES(OLD-VOLUME-NAME=*SAME)
```

   Das Unterverzeichnis `MYNETS` im Verzeichnis `bs2data1` auf dem Net-Server steht nun als Net-Storage-Volume des angegebenen Pubsets für die Nutzung im BS2000-Betrieb zur Verfügung. Das Verzeichnis `MYNETS` enthält noch die Verwaltungsdaten und die darauf eingerichteten BS2000-Dateien. Bei Net-Storage Typ NETVOL sind nach erfolgreichem Import des Net-Storage-Volume die CEs bereits im TSOSCAT katalogisiert. Diese sind aber im Falle von Net-Storage-Volumes vom Typ NETSTOR (angelegt unter BS2000 < V21.0) noch nicht importiert.
   Bei Net-Storage Typ NETSTOR müssen die BS2000-Dateien des Net-Storage-Volume noch importiert werden, z.B. mit

```
/IMPORT-FILE SUPPORT=*DISK(VOLUME=MYNETS, DEVICE-TYPE=NETSTOR,
                           PUBSET=A, FILE-NAME=*ALL)
```

Die Dateien im Verzeichnis MYNETS sind nun im Katalog des Pubsets A eingetragen und können verwendet werden. Dies gilt für BS2000-Dateien. Es gilt auch für Node-Files, wenn sie bereits im Dateikatalog . BS2FSCAT des Volumes eingetragen sind.
Für Node-Files, die noch nicht im Dateikatalog .BS2FSCAT enthalten sind, kann mit dem Kommando IMPORT-NODE-FILE ein Katalogeintrag erstellt werden.

10.8 Verwalten von Net-Storage in BS2000

Die Systembetreuung kann Net-Storage über die Schnittstellen des DVS verwalten. Der Net-Storage muss dazu mit dem BS2000-System verbunden sein.

- Net-Storage-Volumes werden im Rahmen der Daten-Pubsets verwaltet.
- Informationen über den im BS2000-System verfügbaren Net-Storage liefert das Kommando SHOW-NET-STORAGE.
- Informationen über die einem Pubset zugeordneten Net-Storage-Volumes liefert das Kommando SHOW-PUBSET-NET-STORAGE.
- Informationen über die den Net-Storage belegenden Tasks des lokalen Systems liefert das Kommando SHOW-NET-STORAGE-OCCUPATION.
- Informationen über die Verzeichnisse, die von einem Net-Server für einen Net-Client und BS2000 freigegeben sind, liefert das Kommando LIST-NET-DIRECTORIES.
- Mit dem Kommando SET-NET-CLIENT-ALTERNATE wird eine Zuordnung von Net-Clients für die Hochverfügbarkeit getroffen. Es wird eine symmetrische Beziehung zwischen zwei Net-Clients geschaffen, die dafür sorgt, dass bei einem Ausfall eines der beiden Net-Clients der andere Net-Client die Net-Storage-Verbindungen übernimmt. Die Zuordnung wird mit SHOW-NET-CLIENT-ALTERNATE ausgegeben. Sie kann mit SET-NET-CLIENT-ALTERNATE wieder aufgehoben werden.
- Nachdem der Net-Storage mit dem BS2000-System verbunden ist, können alle Benutzer des Pubsets den Net-Storage nutzen, da im Benutzerkatalog standardmäßig `NET-STORAGE-USAGE=*ALLOWED` eingestellt ist. Der belegte Speicherplatz auf einem Net-Storage-Volume wird nicht auf das Public-Space-Limit des Benutzers angerechnet.

  Benutzern kann die Erlaubnis, neben den Dateien auf den lokalen Platten des Pubsets auch Dateien auf den Net-Storage-Volumes des Pubsets anzulegen und zu bearbeiten, wieder entzogen werden, z.B. mit

  ```
  /MODIFY-USER-ATTRIBUTES ...,NET-STORAGE-USAGE=*NOT-ALLOWED
  ```

  Das Kommando SHOW-USER-ATTRIBUTES zeigt diese Einstellung.
- Speed Catalog Access für Net-Storage Typ NETSTOR (SCANET) unterstützt den schnellen Zugriff auf Informationen des Dateikatalogs (Dateiname `.BS2FSCAT`) mit den Metadaten der auf Net-Storage abgelegten Dateien.
  Es wird empfohlen, das Subsystem SCANET zu starten (z.B. in der CMDFILE), wenn Net-Storage Typ NETSTOR benutzt wird oder es geplant ist, Net-Storage einzusetzen.
  Da auf Net-Storage Typ NETVOL kein Dateikatalog (Dateiname `.BS2FSCAT`) existiert, kann für diesen Typ auf das Subsystem SCANET verzichtet werden.
- Für Benutzer, die mit Node-Files arbeiten möchten, müssen im Benutzerkatalog des Pubsets die POSIX-Benutzerattribute UserID und GroupID eingetragen werden, z.B. mit folgendem Kommando:

  ```
  /MODIFY-POSIX-USER-ATTRIBUTES USER-ID=<bs2000-userid>; PUBSET=<cat-id>;
                                USER-NUMBER=<uid>; GROUP-NUMBER=<gid>
  ```

  Diese IDs müssen mit den Benutzern des offenen Systems, die mit denselben Node-Files arbeiten wollen, abgestimmt werden.
- Das Kommando IMPORT-NODE-FILE importiert einzelne Node-Files in das BS2000-System. Der Benutzer kann dabei angeben, ob die zu importierende Datei als PAM-Datei oder als SAM-Datei verarbeitet werden soll.
- Informationen über Node-Files liefert das Kommando LIST-NODE-FILES. Dabei werden nur Dateien ausgegeben, die den BS2000-Namenskonventionen für Dateien genügen.

- Node-Files können mit dem Kommando EXPORT-NODE-FILE wieder aus dem BS2000-Dateikatalog entfernt werden.
- Vor dem Einleiten des Shutdown im BS2000-System müssen die Pubsets, denen Net-Storage-Volumes zugeordnet sind, exportiert werden (siehe auch Abschnitt „Net-Storage von BS2000 trennen").

Rekonfiguration von Net-Storage-Volumes

Die Systembetreuung kann Net-Storage-Volumes in BS2000 rekonfigurieren, d.h. einem Pubset entziehen und, ohne die Dateien auf dem Net-Storage-Volume zu löschen, einem anderen Pubset zuordnen.

- Entziehen Sie das Net-Storage-Volume dem (importierten) Pubset, z.B. mit

  ```
  /REMOVE-NET-STORAGE-VOLUME ...,FILES-ON-VOLUME=*EXPORT
  ```

  Die Katalogeinträge der Dateien werden nur im lokalen Pubset gelöscht. Die Dateien und Verwaltungsdaten auf dem Net-Storage bleiben bestehen.
  Bei Net-Storage Typ NETVOL wird ein Dateikatalog (Dateiname .SCAN_TSOSCAT_CES) erstellt.
- Ordnen Sie das Net-Storage-Volume einem anderen Pubset zu, z.B. mit

  ```
  /ADD-NET-STORAGE-VOLUME ...PUBSET=B,IMPORT=*YES(...)
  ```

  Die Verwaltungsdaten auf dem Net-Storage-Volume werden aktualisiert. Wenn das zuzuordnende Net-Storage-Volume unter BS2000 < V21 erstellt wurde, handelt es sich um ein Volume vom Typ NETSTOR. Dieses kann gleichzeitig in ein Net-Storage-Volume vom Typ NETVOL umgewandelt werden: ..., IMPORT=*YES(CONVERT = *TO-NETVOL). Eine Rück-Konvertierung zum Typ NETSTOR ist nicht möglich!
- Bei Net-Storage Typ NETVOL werden alle CEs katalogisiert und der Dateikatalog (Dateiname . SCAN_TSOSCAT_CES) gelöscht.
- Bei Net-Storage Typ NETSTOR müssen Sie die Katalogeinträge für die Dateien auf dem Net-Storage-Volume im „neuen“ Pubset erstellen, z.B. mit

  ```
  /IMPORT-FILE SUPPORT=*DISK(VOLUME=<net-storage-volume>,
                             PUBSET=B,DEVICE-TYPE=NETSTOR,FILE-NAME=*ALL)
  ```

  Die Dateien des Net-Storage-Volumes sind nun im Katalog des Pubsets B eingetragen und können verwendet werden.
- Bei Net-Storage Typ NETSTOR können Sie den Import auch vorab simulieren, z.B. mit:

  ```
  /CHECK-IMPORT-DISK-FILE VOLUME=P@BA00,DEVICE-TYPE=NETSTOR,...
  ```
- Bei Net-Storage Typ NETSTOR
  In bestimmten Fällen, z.B. zur Beseitigung von Inkonsistenzen nach Ausfall eines Net-Storage, kann es sinnvoll sein, einzelne Dateien vom Pubset zu exportieren:
  Hierzu muss beim EXPORT-FILE ein Net-Storage-Volume oder der Storage-Typ *NET-STORAGE angegeben werden. (Andernfalls wirkt EXPORT-FILE nur auf Dateien auf privaten Datenträgern.)

  ```
  /EXPORT-FILE ...,SELECT=*BY-ATTRIBUTES(STORAGE-TYPE=*NET-STORAGE,...)
  ```

  Entsprechend können Sie Dateien, die in BS2000 noch nicht bekannt sind, importieren, d.h. einen Katalogeintrag im lokalen Pubset erstellen, z.B. mit:

  ```
  /IMPORT-FILE SUPPORT=*DISK(VOLUME=P@BA00,DEVICE-TYPE=NETSTOR,
                             FILE-NAME= ... )
  ```

10.9 Net-Storage von BS2000 trennen

Net-Storage kann mit UMOUNT-NET-STORAGE vom BS2000-System getrennt werden. Laufende Anwendungen können dann nicht mehr auf Dateien, die auf Net-Storage liegen, zugreifen.

Vor dem Einleiten des Shutdown im BS2000-System müssen die Daten-Pubsets, denen Net-Storage-Volumes zugeordnet sind, exportiert werden. Damit wird sichergestellt, dass auf dem Net-Server keine geöffneten Dateien mehr existieren.

BCAM darf erst nach dem Exportieren des Pubsets beendet werden. Wenn BCAM vorher beendet wird, können geöffnete Dateien auf Net-Storage (u. A. auch der Katalog) nicht mehr geschlossen und die Verbindung zwischen Net-Client und Net-Server nicht abgebaut werden. In diesem Fall ist der Zugriff zu dem Net-Storage-Volume bei einem erneuten Pubset-Import erst wieder möglich, wenn die Meldung `DMS1326` beantwortet wird.

Net-Storage kann auch im laufenden BS2000-Betrieb getrennt werden, z.B. bei einem Wechsel des Net-Servers:

- Trennen Sie den Net-Storage vom BS2000-System, z.B. mit

  ```
  /UMOUNT-NET-STORAGE DIRECTORY='/bs2data1',
                  SERVER=*IP-ADDRESS(IP-ADDRESS=172.10.1.110),
                  CLIENT=system1
  ```

  Das Kommando wird nur ausgeführt, wenn keine Belegung des Net-Storage mehr vorliegt. Belegungen durch das lokale System können Sie mit dem Kommando SHOW-NET-STORAGE-OCCUPATION überprüfen. SHOW-NET-STORAGE-OCCUPATION zeigt alle belegenden Tasks des lokalen Systems für eine Mount-Id an. Auch für Shared-Pubsets werden nur die belegenden Tasks des lokalen Systems ausgegeben.

  Im Notfall können Sie die Ausführung des Kommandos UMOUNT-NET-STORAGE mit der Option FORCE=*YES erzwingen: Der Net-Storage wird auch dann getrennt, wenn es noch belegende Tasks gibt. Dabei können jedoch Inkonsistenzen entstehen, da geöffnete Dateien nicht ordnungsgemäß geschlossen werden.

Mit dem Kommando UMOUNT-NET-STORAGE hängen Sie auch das freigegebene Verzeichnis am Net-Client aus.

10.10 Randbedingungen

- BS2000 allgemein
- Shared-Pubsets
- Platten-Spiegelung
- Interoperabilität
- Net-Storage-Volumes ab BS2000 V21

10.10.1 BS2000 allgemein

Für das Arbeiten mit Net-Storage sind in BS2000 folgende allgemeine Randbedingungen zu beachten:

- Ein Net-Storage-Volume kann nur einem Daten-Pubset zugeordnet werden. Dem Home-Pubset kann kein Net-Storage zugeordnet werden.
- Eine Datei auf Net-Storage besteht nur aus einem Extent. Die Angabe einer Volume-Liste wird akzeptiert, es wird aber nur das erste Volume verwendet.
- Eine Speicherplatzbelegung im Operanden SPACE der Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES kann zwar angegeben werden, es wird aber kein Speicherplatz in dieser Größe auf Net-Storage belegt. Die (maximale) Speicherplatzbelegung findet erst dann statt, wenn die Datei in der entsprechenden Größe gespeichert wird.
- Ein Net-Storage-Volume verhält sich in Bezug auf die Allokierung wie eine NK2-Platte mit minimaler Allokierungseinheit 8 KB.
- Absolutzuweisungen sind für Dateien auf Net-Storage nicht möglich.
- Dateien mit folgenden Eigenschaften können **nicht** auf einem Net-Storage-Volume eingerichtet werden:
  - Dateien mit PAM-Schlüssel
  - Arbeitsdateien
  - temporäre Dateien
  - Dateigenerationsgruppen
- Ein SMPGEN-Lauf wird abgebrochen, wenn Pubsets mit Dateien auf Net-Storage bearbeitet werden. Katalogeinträge von Dateien auf Net-Storage müssen auf den betroffenen Pubsets vor dem SMPGEN-Lauf entfernt oder exportiert werden, z.B. mit

  ```
  /REMOVE-NET-STORAGE-VOLUME ...,FILES-ON-VOLUME=*EXPORT
  ```
- Wenn ein Pubset, dem Net-Storage-Volumes zugeordnet sind, mit PVSREN umbenannt wird, wird auch ein evtl. vorhandenes Volume mit Standard-Namen (P@BX00, XY@000,...) entsprechend umbenannt. Net-Storage-Volumes mit benutzerdefinierten Namen werden nicht umbenannt.
- Der Administrator des File-Servers kann für Anwendungen, die auf unterschiedlichen Betriebssystemen ablaufen, den direkten Zugriff auf gemeinsame Dateien auf Net-Storage zulassen. Hierbei ist Folgendes zu beachten:
  - Anwendungen fremder Betriebssysteme (nicht BS2000) dürfen schreibend nur auf Dateien vom Typ *NODE-FILE zugreifen.
  - Zugriffsschutz in BS2000 hat nur dann eine Wirkung, wenn der Zugriff von BS2000 aus erfolgt. Sonst hat er keine Wirkung.
    Beispiel
    Das Einrichten der Mehrbenutzbarkeit einer Datei im BS2000 mit `/MODIFY-FILE-ATTRIBUTES <dateiname>,USER-ACCESS=*ALL-USERS` ist nur im BS2000 wirksam. Aus Sicht eines fremden Systems, bleibt der Zugriff weiterhin eingeschränkt.

10.10.2 Shared-Pubsets

Mehrere BS2000-Systeme können gemeinsam auf Net-Storage-Volumes von Shared-Pubsets zugreifen.

Alle Sharer müssen sich mit dem entsprechenden Net-Storage verbinden (MOUNT-NET-STORAGE).

Im Shared-Pubset-Verbund kann ein neues Net-Storage-Volume nur am Master-Rechner angelegt oder auch gelöscht werden. Wurde ein Net-Storage-Volume am Master angelegt, muss dieses Net-Storage-Volume auch an allen Slaves jeweils über das Kommando ADD-NET-STORAGE-VOLUME bekannt gemacht und belegt werden. Erst danach kann auch vom Slave-Rechner auf das Net-Storage-Volume zugegriffen werden (Status IN-USE). Diese Belegung bleibt bis zum Export des Pubsets erhalten.

Danach ist auch bei erneutem IMPORT-PUBSET an einem Slave-Rechner das Net-Storage-Volume sofort verfügbar (existierende Verbindung zum Net-Storage vorausgesetzt).

Bei einem Masterwechsel aktualisiert das System auf den zugehörigen Net-Storage-Volumes die eingetragene Master-SYSID. Dies ist aber nur für verfügbare Net-Storage-Volumes (Status: IN USE) möglich. Sollte bei einem Master-Wechsel ein Net-Storage-Volume nicht verfügbar gewesen sein (nicht verbunden oder nicht belegt (Status: ONLINE)), kommt es beim ersten Zugriff auf das Volume zur Meldung `DMS1326`, die an der Konsole des Master-Rechners ausgegeben wird und beantwortet werden muss.

10.10.3 Platten-Spiegelung

Pubsets, auf denen Net-Storage-Dateien katalogisiert sind, können z.B. mit SHC-OSD gespiegelt werden. Allerdings werden die Net-Storage-Dateien selbst nicht gespiegelt, da sie auf einem anderen Medium gespeichert sind. Nach Auflösen der Spiegelung sind somit Dateien auf Net-Storage über einen abgetrennten Spiegel nicht zu erreichen.

Es sind folgende Fälle zu unterscheiden:

- Abtrennen des Spiegels ohne Umbenennen

  Der Original-Pubset wird wieder importiert. Net-Storage-Dateien können weiter wie gewohnt über den Original-Pubset angelegt, gelöscht oder bearbeitet werden.

  Wenn der abgetrennte Spiegel wieder mit dem Original-Pubset synchronisiert wird, werden implizit auch die Katalogeinträge der Net-Storage-Dateien aktualisiert.

  Zu beachten ist, dass im Rahmen des Auftrennens der Spiegel keine Änderungen an den Verwaltungseinträgen der Net-Storage-Volumes vorgenommen werden. D.h. der Pubset, der als erster wieder importiert wird, erhält Zugriff zum betreffenden Net-Storage (existierende Verbindung zu Net-Storage vorausgesetzt).

  Wenn der abgetrennte Spiegel an einem anderen System importiert wird, wird bei Zugriff auf die zugeordneten Net-Storage-Volumes die beantwortbare Meldung DMS1326 ausgegeben, da das Volume eine fremde SYSID enthält. Wenn die Meldung mit „ja“ beantwortet wird, dann wird der Zugriff auf das Volume erlaubt. Zu beachten ist, dass falls das Volume tatsächlich an einem anderen System belegt ist, es durch die falsche Beantwortung der Frage zu Inkonsistenzen und zu Datenverlust kommen kann.

  Ist eine derartige Weiterverarbeitung des abgetrennten Spiegels geplant, wird folgendes Vorgehen empfohlen:

  1. Entziehen der dem Pubset zugeordneten Net-Storage-Volumes:

     ```
     /REMOVE-NET-STORAGE-VOLUME ..., FILES-ON-VOLUME=*EXPORT
     ```

  2. Den Pubset exportieren und den Spiegel auftrennen.
  3. Gewünschten Pubset, Original oder Spiegel, importieren.
  4. Net-Storage-Volumes wieder dem importierten/aktiven Pubset zuordnen:

     ```
     /ADD-NET-STORAGE-VOLUME ..., IMPORT=*YES
     ```

  5. Die Katalogeinträge der Dateien des Net-Storage-Volumes mit IMPORT-FILE wieder in den TSOSCAT des importierten Pubsets aufnehmen.
  6. Der abgetrennte Spiegel enthält nun keine Net-Storage-Dateien mehr und kann an einem anderen System weiterverarbeitet werden.

- Abtrennen des Spiegels mit Umbenennen

  Es wird empfohlen die Net-Storage-Volumes vor dem Abtrennen des Spiegels vom Pubset zu entziehen:

  ```
  /REMOVE-NET-STORAGE-VOLUME ..., FILES-ON-VOLUME=*EXPORT
  ```

  Nach dem Abtrennen des Spiegels kann der Net-Storage wieder dem gewünschten Pubset-Spiegel zugeordnet werden:

  ```
  /ADD-NET-STORAGE-VOLUME ..., IMPORT=*YES
  ```

  Danach müssen die Dateien vom Net-Storage-Volume importiert werden.

  Wenn gewünscht wird, dass die Dateien auf Net-Storage permanent verfügbar sind, dann kann alternativ auch folgendermaßen vorgegangen werden:

  1. Der Spiegel wird abgetrennt und dabei umbenannt. Beim Abtrennen des Spiegels werden auf diesem die Net-Storage-Volume Einträge implizit für ungültig erklärt.
  2. Beim Import dieses Pubsets werden die Net-Storage-Volume-Einträge und die Katalogeinträge der zugehörigen Net-Storage-Dateien automatisch gelöscht.
  3. Nun kann der Pubset entweder ohne Net-Storage weiterverwendet werden oder es können neue Net-Storage-Volumes angelegt werden. Es können auch vorhandene Net-Storage-Volumes wieder in Betrieb genommen werden:

     ```
     /ADD-NET-STORAGE-VOLUME ..., IMPORT=*YES
     ```
  4. Sollen die Net-Storage-Dateien und -Volumes des Original-Pubsets dem abgetrennten und bereinigten Spiegel hinzugefügt werden, müssen diese vorher vom Original exportiert werden:

     ```
     /REMOVE-NET-STORAGE FILES-ON-VOLUME=*EXPORT
     ```

10.10.4 Interoperabilität

Für Node-Files müssen folgende zusätzlichen Randbedingungen beachtet werden:

- Nur Dateien vom Typ *NODE-FILE, die sog. Node-Files, sind für die Interoperabilität mit Systemen der offenen Welt zugelassen.

  ! Dateien vom Typ *BS2000 dürfen von Systemen der offenen Welt nicht geändert werden!

- Node-Files sind strukturlos. Sie enden auf Byte-Grenze.
- Aus BS2000-Sicht können nur folgende Dateien zu Node-Files werden:
  - PAM-Dateien ohne PAM-Schlüssel (BLKCTRL=NO)
  - SAM-Dateien mit Blockkontrollinformation und variabler Satzlänge (BLKCTRL=DATA, RECFORM=V)
- Node-Files werden in benutzerspezifischen Verzeichnissen angelegt. Diese werden von BS2000 automatisch beim ersten Anlegen eines Node-File erzeugt und mit den notwendigen POSIX-ACLs versorgt. Danach können berechtigte Benutzer sowohl von BS2000 als auch von offenen Systemen auf das Verzeichnis und die darin enthaltenen Node-Files zugreifen.
- Zugriffsrechte und Schutzattribute in BS2000 können auch für Node-Files wie üblich eingestellt werden (Zugriffsrecht (ACCESS), Mehrbenutzbarkeit (USER-ACCESS), Zugriffskontrolle, Kennwörter, GUARDS, Miteigentümerschaft), siehe z.B. das Kommando MODIFY-FILE-ATTRIBUTES im Handbuch „Kommandos“ [27]. Sie gelten für BS2000, nicht aber für offene Systeme.
- Node-Files, die von offenen Systemen angelegt werden, müssen vor ihrer Benutzung mit dem Kommando IMPORT-NODE-FILES in BS2000 importiert werden.
- Das BS2000-Kommando LIST-NODE-FILES informiert einen Benutzer über die Node-Files in seinem benutzerspezifischen Verzeichnis.
- Eigentümer der Dateien ist der jeweilige BS2000-Benutzer, unter dem die Dateien im BS2000 bzw. im TSOSCAT des Pubsets, dem das Net-Storage-Volume zugeordnet ist, katalogisiert sind.
  In diesem Zusammenhang muss beachtet werden, dass im Benutzerkatalog des Pubsets mittels MODIFY-POSIX-USER-ATTRBIUTES eine sinnvolle User- und Group-Number eingetragen wird:
  - Benutzer aus offenen System erhalten Zugriff auf Node-Files durch ihre Benutzer- und Gruppennummer (uid, gid). Die pubset-spezifische POSIX User- und Group-Number eines Benutzers muss mit der uid, gid desselben Benutzers der offenen Welt abgestimmt sein, damit ein Zugriff von Systemen der offenen Welt möglich ist. Dies kann durch ein Verzeichnisdienst sichergestellt werden (z.B. openLDAP, AD)
  - Achten Sie darauf, dass verschiedene Benutzer im BS2000 verschiedene und eindeutige User- und Group-Numbers erhalten. Wird Net-Storage an mehreren Pubsets konfiguriert, müssen die User- und Group-Numbers desselben Benutzers in den verschiedenen Benutzerkatalogen identisch sein (siehe SHOW- bzw. MODIFY-POSIX-USER-ATTRIBUTES).
- Node-Files haben im UFS die Zugriffsrechte `rw- --- ---` (entspricht unter Unix user group world).
  Dies sind die für BS2000 minimal notwendigen Zugriffsrechte für den entsprechenden Benutzer.
  Ein Benutzer eines offenen Systems, der das Verzeichnis bei sich gemountet hat, kann diese Zugriffsrechte ändern. Ein BS2000-Benutzer kann dies nicht.
- Zugriffrechte des BS2000 werden allein mit BS2000-Mitteln verwaltet. Zugriffsrechte aus Sicht offener Systeme, die über den Standard hinausgehen müssen mit den Mitteln der offenen Systeme verwaltet werden, d.h.:
  - Hat ein Benutzer eines offenen Systems die Zugriffsrechte z.B. für seine Gruppe erweitert: chmod 660 <file-name> kann dies mit BS2000-Mitteln nicht rückgängig gemacht werden.

10.10.5 Net-Storage-Volumes ab BS2000 V21

Seit Einführung von BS2000 V21 gibt es neben dem bisherigen Net-Storage-Volume-Typ NETSTOR außerdem den Net-Storage-Volume-Typ NETVOL. Mit dem Kommando ADD-NET-STORAGE-VOLUME werden nur noch Net-Storage-Volumes vom Typ NETVOL angelegt. Dabei wird ein Verzeichnis im freigegebenen Dateisystem des Net-Servers eingerichtet und einem lokalen Daten-Pubset (SF- oder SM-Pubset) in Form eines Net-Storage-Volumes zugeordnet.

Bereits existierende Net-Storage-Volumes vom Typ NETSTOR werden weiterhin unterstützt. Sie können wie gewohnt weiter benutzt werden.

Beim Anlegen von Dateien spezifiziert der Benutzer wie bisher den Volume-Typen NETSTOR, um eine Datei auf Net-Storage anzulegen:

```
/CREATE-FILE FILE-NAME=:NET1:$UA.NET-FILE.1,VOLUME=NEWNET,DEVICE-TYPE=NETSTOR
```

Im folgenden Beispiel ist das angegebene Net-Storage-Volume vom Typ NETVOL (siehe DEVICE-TYPE):

```
/sh-file-attributes :NET1:$UA.NET-FILE.1,inf=*par(alloc=*yes)
%0000000004n:NET1:$UA.NET-FILE.1
%  ------------------------------ ALLOCATION   ------------------------------
%  SUPPORT    = PUB         S-ALLOC    = 4           HIGH-US-PA = 0
%  EXTENTS     VOLUME     DEVICE-TYPE      EXTENTS     VOLUME     DEVICE-TYPE
%      1       NEWNET       NETVOL
%  NUM-OF-EXT = 1
%:NET1: NET   :      1 FILE  RES=         4 FRE=         4 REL=         4 PAGES
```

Net-Storage-Volumes werden als Plattengeräte aufgefasst, daher führt die Angabe NETSTOR, STDISK oder D3435 im DEVICE-OPERANDEN des CREATE-FILE-Kommandos zum gleichen Ergebnis, wenn der Benutzer die VSN des Datenträgers angibt.

Bei Dateien, die auf Net-Volumes vom Typ NETSTOR allokiert sind, wird unter DEVICE-TYPE NETSTOR ausgegeben; entsprechendes gilt für Dateien auf Net-Storage-Volumes vom Typ NETVOL (siehe Beispiel oben).

Net-Storage-Volumes vom Typ NETSTOR können in Net-Storage-Volumes vom Typ NETVOL umgewandelt werden. Dazu dürfen die Volumes keinem Pubset zugeordnet sein (vorheriges REMOVE-NET-STORAGE-VOLUME ... FILES-ON-VOLUME=*EXPORT erforderlich). Die Konvertierungsfunktion ist im Kommando ADD-NET-STORAGE-VOLUME enthalten:

```
/ADD-NET-STORAGE-VOLUME VOLUME=NET001, PUBSET=NET1, ... , IMPORT=*YES(CONVERT=*TO-
NETVOL)
```

> ! Die Konvertierung kann nicht rückgängig gemacht werden!

Net-Storage-Volumes können mit REMOVE-NET-STORAGE-VOLUME VOLUME=..., FILES-ON-VOLUME=*EXPORT von einem Pubset abgehängt und an einen anderen Pubset angehängt werden. Bei Net-Storage-Volumes vom Typ NETVOL werden dabei automatisch die enthaltenen Dateien in den TSOSCAT des Pubsets importiert. Treten beim Import Dateinamenskonflikte auf, wird der Import abgebrochen und ev. schon angelegte Katalogeinträge wieder entfernt. In diesem Fall muss der Systemverwalter den Pubset bereinigen und den ADD-NET-STORAGE-VOLUME wiederholen. (Bei Net-Storage-Volumes vom Typ NETSTOR müssen die Dateien explizit mit dem BS2000-Kommando IMPORT-FILE importiert werden.)

11 Job- und Task-Steuerung

In diesem Kapitel werden die Job-Steuerung, die alle wartenden Aufträge (Jobs) verwaltet, und die Task-Steuerung, die alle laufenden Aufträge (Tasks) verwaltet, beschrieben.
Das Programm JMP (Jobpool Management Program) rekonstruiert aus den Informationen, die das JMS über akzeptierte Batch-Jobs im Job-Pool hinterlegt hat, ENTER-JOB-Kommandos.

Im Abschnitt „PCS: Performance-Überwachungssystem" werden die Zusammenhänge zwischen Job- und Task-Steuerung und dem Performance-Überwachungssystem PCS kurz dargestellt.

Es lassen sich grundsätzlich folgende Benutzer-Jobs unterscheiden:

- Dialog-Jobs
- Batch-Jobs (ENTER-Job oder SPOOLIN)
- SPOOLOUT-Jobs

Mit Ausnahme der SPOOLOUT-Jobs beginnen alle Benutzer-Jobs mit dem Kommando SET-LOGON-PARAMETERS und enden mit dem Kommando EXIT-JOB.
SPOOLOUT-Jobs werden entweder mit dem Kommando PRINT-DOCUMENT eingeleitet oder nach Task-Ende zur Ausgabe bereitgestellt.

Ein Job wird von der Job-Steuerung einer Job-Klasse zugeordnet und in die entsprechende Warteschlange eingereiht. Jeder Job erhält sofort bei Job-Annahme eine Auftragsnummer (TSN), mit deren Hilfe er während seiner Verweilzeit im System angesprochen werden kann. Die Job-Steuerung steuert und verwaltet alle Jobs.

Ein Job wird zur Task, wenn ihm Systemressourcen (CPU, Speicher, Geräte) zugeteilt werden. Die Task-Steuerung richtet einen Task-Control-Block (TCB) ein.
Die Task-Steuerung steuert und verwaltet alle Tasks.

Das Job-Management-System (JMS) sorgt für die Annahme und Abwicklung der Benutzer-Jobs. Die wesentlichen Aufgaben des JMS sind:

- Auftragsannahme mit Überprüfung der Zugangsberechtigung
- Job-Auswahl zur Startfreigabe
- Job-Initialisierung

Diese Aufgaben werden durch die Komponenten der Job-Steuerung (Job-Klassen und Job-Streams) realisiert.

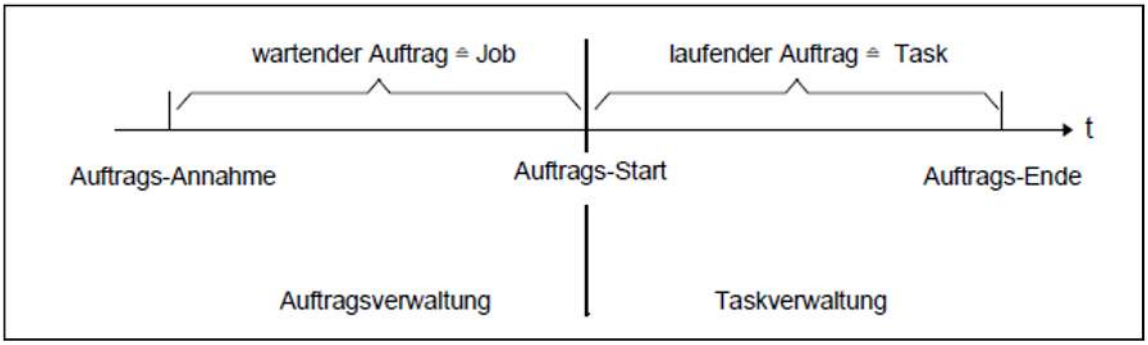


Bild 16: Zusammenhang Job-Task

11.1 Job-Steuerung

Die Job-Steuerung (Auftragssteuerung) ermöglicht eine differenzierte und damit rationellere Steuerung von Jobs. Dabei sind vor allem zwei Eigenschaften der Job-Steuerung von Bedeutung:

Die Jobs können in Job-Klassen eingeteilt werden.

Job-Klassen beschreiben ein Job-Profil und sind ein Ordnungsbegriff für Jobs, die in bestimmten Eigenschaften und Merkmalen übereinstimmen.

Die Job-Klassen werden Job-Schedulern zugeordnet.

Die Job-Klassen werden von der Systembetreuung max. 16 Job-Schedulern zugeordnet, wobei die Job-Scheduler unterschiedliche Strategien verfolgen können und entscheiden, welcher Job zum Start freigegeben wird.
Die System-Job-Klasse $SYSJC ist fest dem System-Job-Scheduler $SYSJS zugeordnet.

Im Normalfall werden alle von den Job-Schedulern zum Start freigegebenen Jobs vom Klassen-Scheduler sofort der Task-Steuerung zum Start übergeben.

Nach Aufhebung einer Überlastsituation oder einer Speichersättigung werden die Jobs der Job-Klasse bevorzugt, deren Klassenoptimum noch nicht erreicht ist und die die höchste Dringlichkeit besitzt (abhängig von der Anzahl bereits in der Job-Klasse laufender Jobs und dem Klassengewicht). Der Klassen-Scheduler bildet die Schnittstelle zur Task-Steuerung. Er fällt die Entscheidung, ob ein zum Start freigegebener Job tatsächlich abläuft.

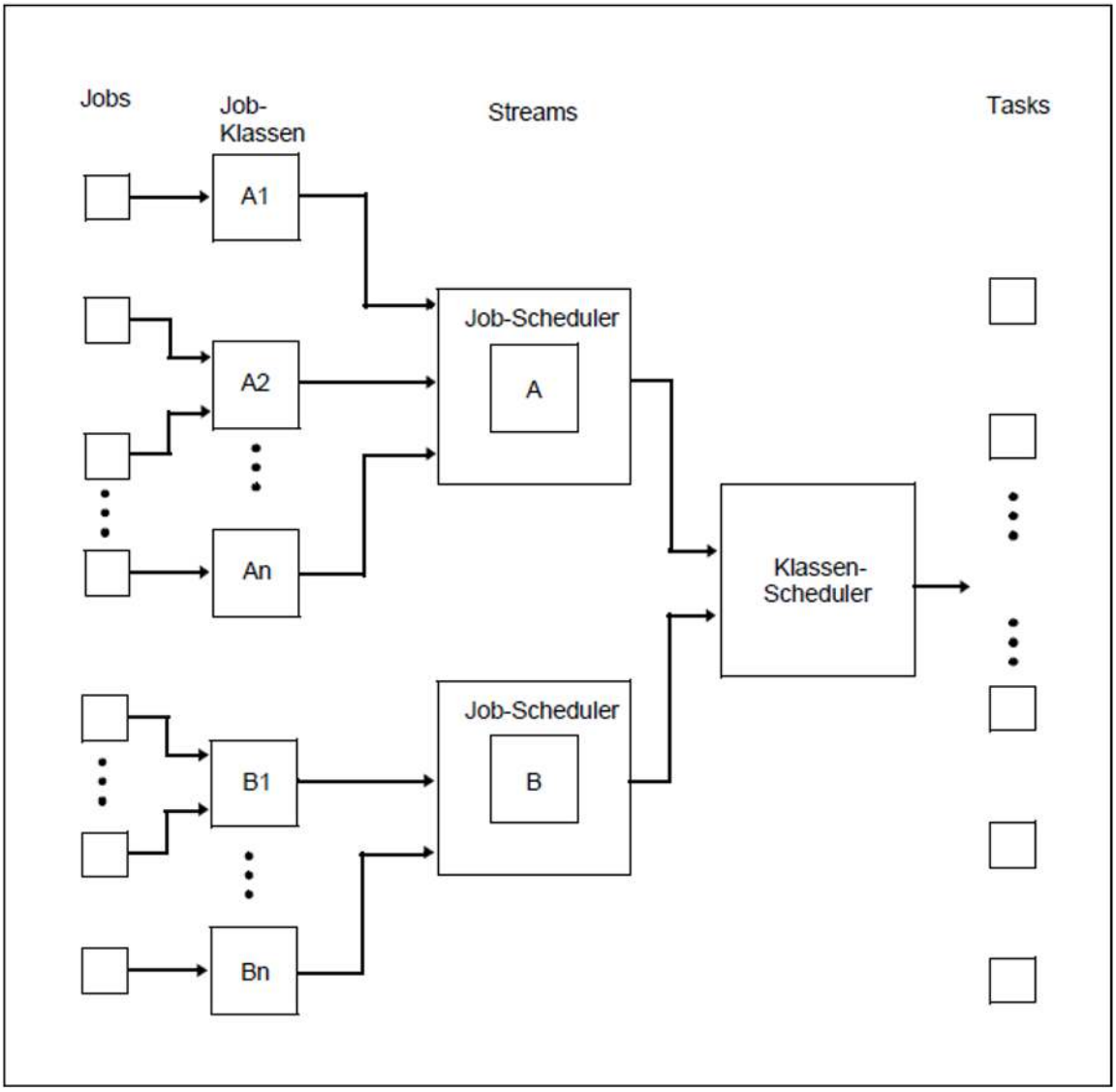


Bild 17: Komponenten der Job-Steuerung

Unterstützung des zentralen Kalenders durch das JMS

Die Startzeitpunkte von Batch-Jobs können über symbolische Daten aus dem zentralen Kalender von BS2000 angegeben werden. Dazu dient ein weiterer neuer Typ von Batch-Job: der Kalender-Job. Somit sind folgende drei Typen definiert:

der Termin-Job	ist ein einmalig ablaufender Batch-Job, dessen Startzeitpunkt von einer Zeitangabe des Benutzers abhängig ist.
der Repeat-Job	ist ein Wiederholungs-Job, dessen Wiederholungszyklus von einer Benutzerangabe (Repeat-Attribut) abhängig ist.
der Kalender-Job	ist entsprechend seiner Eigenschaft ein Termin-Job mit dem Start-Attribut *AT(...), der jedoch nach Vorgabe eines Kalenders wiederholt werden kann. Ein Kalender-Job ist somit auch ein zeitlich durch die Angaben im Kalender (oder im Operanden LIMIT der Kommandos ENTER-JOB und MODIFY-JOB) begrenzter Wiederholungs-Job.

Der erste Ablauf und die folgenden Wiederholungen eines Kalender-Jobs werden vom System gleich behandelt. Das System unterscheidet nicht – wie bei Repeat-Jobs – in „Current“ und „Next Image“ eines Kalender-Jobs.
Alle Abläufe eines Kalender-Jobs haben die gleiche TSN. Damit ist eine bessere Identifizierung und Überwachung des Kalender-Jobs möglich. Die einzelnen Abläufe werden jeweils bis zum Erreichen des Startzeitpunkts vom Job-Scheduler verwaltet: Am Ende der Terminierung des Vorgängers wird der Nachfolger eingerichtet (gleiche TSN) und dem Scheduler mit dem Start-Attribut *AT(...) und der nächsten Startzeit des symbolischen Datums aus dem Kalender übergeben.

Die Möglichkeiten zur Angabe des Startzeitpunkts richten sich nach den Vorgaben des zentralen Kalenders (z.B. ist der Wiederholungszyklus nur auf Tage und nicht auf Stunden bezogen). Siehe dazu das Handbuch „CALENDAR“ [7].

Im Gegensatz zu Repeat-Jobs können Kalender-Jobs mit Hilfe von Monitor-Jobvariablen (MONJVs) überwacht werden. Die MONJV bleibt für die gesamte Lebensdauer des Kalender-Jobs geöffnet. Zusammen mit der gleichbleibenden TSN ist somit jederzeit der Zugriff auf den Job über die MONJV gewährleistet.

Ausgefallene Wiederholungen eines Kalender-Jobs werden – entsprechend den einfachen Termin-Jobs – mit dem abgelaufenen Startdatum nachgeholt. Anschließend wird aus dem Kalender der nächste Startzeitpunkt ermittelt.

Mit Hilfe des Operanden LIMIT der Kommandos ENTER-JOB und MODIFY-JOB ist eine Begrenzung der Lebensdauer eines Kalender-Jobs möglich. Limit-Überschreitungen die sich z.B. durch ausgefallene Wiederholungen oder Verzögerungen im Scheduler ergeben, werden jedoch zugelassen.
Die Lebensdauer kann ebenfalls durch das Setzen eines Ablaufzählers begrenzt werden. Ist das Limit erreicht, wird der gesamte Wiederholungs-Job beendet.

Kommando	**Bedeutung**
CANCEL-JOB [1]	Benutzer-Job abbrechen
CHANGE-TASK-PRIORITY [1]	Job- oder Task-Priorität eines Dialog-, Batch- oder Druck-Jobs ändern
ENTER-JOB	Kommandofolge, die in einer ENTER-Datei als Batch-Job gespeichert ist, unter Angabe von Job-Klasse und Job-Priorität starten
ENTER-PROCEDURE	Kommandofolge starten, die in einer Prozedur als Batch-Job gespeichert ist
EXIT-JOB	die Task beenden
FORCE-JOB-CANCEL	Benutzer-Job abbrechen
HOLD-JOB	Benutzer-Job in den Wartezustand versetzen
HOLD-JOB-CLASS	Job-Klasse in den Wartezustand versetzen
HOLD-JOB-STREAM	Job-Stream und Job-Scheduler in den Wartezustand versetzen
MODIFY-JOB [1]	Eigenschaften eines Benutzer-Jobs ändern
MODIFY-JOB-CLASS	Grenzen und Gewichtung von Job-Klassen ändern

MODIFY-JOB-OPTIONS	Die mit ENTER-JOB, SET-LOGON-PARAMETERS und MODIFY-JOB-OPTIONS eingestellten Werte ändern
MODIFY-JOB-STREAM	Ablaufpriorität der Stream-Task und stream-spezifische Parameter ändern
MODIFY-JOB-SWITCHES	Einstellung von Auftragsschaltern ändern
MOVE-JOBS	Job-Beschreibungen exportieren bzw. importieren
RESUME-JOB	Wartezustand eines Benutzer-Jobs aufheben
RESUME-JOB-CLASS	Wartezustand einer Job-Klasse aufheben
RESUME-JOB-STREAM	Wartezustand von Job-Stream und Job-Scheduler aufheben
SET-LOGON-PARAMETERS	Einleiten eines Dialog- oder Batch-Jobs unter Angabe von Job-Klasse bzw. Job-Priorität
SHOW-JOB-CLASS	Informationen über Job-Klassen-Attribute; unterstützt die Ausgabe in S-Variable
SHOW-JOB-OPTIONS	Informationen über die mit ENTER-JOB, SET-LOGON-PARAMETERS und MODIFY-JOB-OPTIONS eingestellten Werte anfordern
SHOW-JOB-STATUS [1]	Informationen über einen Benutzer-Job anfordern
SHOW-JOB-STREAM	Beschreibung aller Job-Streams anfordern
SHOW-SYSTEM-STATUS	Informationen über Job-Klassen und Job-Streams anfordern
SHOW-USER-ATTRIBUTES	Berechtigung zu den verschiedenen Job-Klassen abfragen
SHOW-USER-STATUS [1]	Informationen über Gruppen von Benutzer-Jobs anfordern
START-JMP	Programm zur Rekonstruktion von ENTER-JOB-Kommandos starten
START-JOB-STREAM	Job-Stream und Job-Scheduler starten
STOP-JOB-STREAM	Job-Stream und Job-Scheduler beenden
Makro	**Bedeutung**
DJINF	DSECT oder Datenbereich für den Makro JINF erstellen
ENTER	ENTER-Job abarbeiten
JINF	Job-Daten anfordern
JMGDJP	DSECT oder Datenbereich für den Makro JMGJPAR erstellen
JMGJPAR	Job-Parameter anfordern

JOBINFO	Job-Daten ausgewählter Jobs anfordern
JSATTCH	Job-Scheduler mit dem JMS verbinden
JSDETCH	Job-Scheduler vom JMS lösen
JSEXPCT	JSS-Ereignisse anfordern
JSINFO	Stream-Parameter abfragen
JSRUNJB	Job zum Start übergeben
JSWAKE	Zeitereignis für Job-Scheduler initiieren
LGOFF	Job beenden
SWITCH	Benutzer- und Auftragsschalter abfragen und verändern
TMODE	Auftrags-Attribute abfragen

Tabelle 25: Schnittstellenübersicht zur Job-Steuerung

[1] Durch diese Kommandos können auch Batch-Jobs bearbeitet werden, die von der aufrufenden Benutzerkennung erzeugt wurden, aber unter einer fremden Benutzerkennung ablaufen

11.1.1 Konzept der Job-Klassen

Die Verwendung von Job-Klassen ermöglicht der Systembetreuung eine Klassifizierung von Benutzer-Jobs.

Das Konzept der Job-Klassen trägt verschiedenen Anforderungen Rechnung.

1. Über die Job-Klassen und deren Zuordnung zu den Job-Schedulern kann ein optimaler Auftragsmix erzeugt werden (z.B. viele kurzlaufende Jobs, wenige langlaufende Jobs).Letztendlich trägt dies zu einer **ausgewogenen Systemauslastung** bei. Außerdem besteht die Möglichkeit, mit Hilfe der Zugriffsrechte zu den Job-Klassen, Privilegien der Job-Steuerung unter den Benutzern zu verteilen.
2. Eine **zusätzliche Zugangskontrolle** zum System kann erreicht werden, indem die Systembetreuung den Benutzern Nutzungsrechte für bestimmte Job-Klassen gibt. Gibt der Benutzer in den Kommandos SET-LOGON-PARAMETERS bzw. ENTER-JOB keine Job-Klasse an, so wird sein Job in einer Standard-Job-Klasse geführt. Die Systembetreuung hat die Möglichkeit, Job-Klassen zu definieren, zu modifizieren oder zu löschen. Dazu steht das Dienstprogramm JMU zur Verfügung, das im Handbuch „Dienstprogramme“ [15] beschrieben ist.
3. Da sich Jobs durch die Benutzeranforderungen unterscheiden, müssen sie auch von den Job-Schedulern unterschiedlich behandelt werden.

Die Beschreibung einer Job-Klasse basiert auf einer Vielzahl von Angaben. Auf Grund vielschichtiger Möglichkeiten, Job-Eigenschaften zu kombinieren, ist es theoretisch denkbar, für jede Job-Merkmal-Kombination eine eigene Job-Klasse zu definieren.
Eine solche Vorgehensweise geht allerdings stark zu Lasten der Transparenz.

Die Systembetreuung sollte demnach die Job-Klassen nach den Kriterien definieren, die für den täglichen Produktionsbetrieb wichtig sind.

Mit dem Dienstprogramm JMU (Anweisung DEFINE-JOB-CLASS) legt die Systembetreuung u.a. folgende Eigenschaften und Vereinbarungen für eine Job-Klasse fest:

- Job-Klassenname
- Zuständiger Stream oder Default-Stream
- Dringlichkeit (Gewichtung) der Job-Klasse
- maximale Anzahl von Jobs, die in der Job-Klasse gleichzeitig ablaufen können
- Anzahl der Jobs, die idealerweise in der Job-Klasse laufen sollen
- Auftragstyp
- Job- und Task-Scheduling-Priorität
- Erlaubnis für das Starten von Repeat-Jobs
- Wiederholungsrhythmus für Repeat-Jobs
- maximal zu verbrauchende CPU-Zeit
- Start-Attribute

Während die **Job-Scheduling-Priorität** das Starten des Jobs beeinflusst, bezieht sich die **Task-Scheduling-Priorität** auf den Ablauf des gestarteten Jobs (= Ausführungspriorität).

Als mögliche Werte gelten für die Job-Steuerung die Prioritäten 1 bis 9 und für die Task-Steuerung die Prioritäten 30 bis 255.

Je niedriger der angegebene Wert ist, desto höher ist die Priorität.

Der Begriff **Auftragstyp** trifft eine Unterscheidung zwischen Batch- und Dialog-Jobs. Für beide Auftragstypen lassen sich Job-Klassen definieren, wobei allerdings zu beachten ist, dass die Dialog-Job-Klassen keinem Job-Scheduling unterliegen.
Hier wird lediglich auf die Einhaltung der Klassen-Grenzen geachtet, d.h. auf die Zahl der in der Job-Klasse geführten Dialog-Jobs und die Zugangsberechtigung zu dieser Job-Klasse.

Beispiel 1

Einteilung der Job-Klassen nach der zu verbrauchenden CPU-Zeit:

- Job-Klasse JCSHORT
  für Jobs, die nicht mehr als 5 CPU-Sekunden verbrauchen werden
- Job-Klasse JCNORMAL
  für Jobs, die nicht mehr als 500 CPU-Sekunden verbrauchen werden
- Job-Klasse JCLONG
  für Jobs, die mehr als 500 CPU-Sekunden verbrauchen werden

Beispiel 2

Einteilung der Job-Klassen nach dem Start-Zeitpunkt:

- Job-Klasse JCEXPRES
  für Jobs, die mit dem Start-Attribut IMMEDIATE ausgerüstet sind
- Job-Klasse JCNORMAL
  für Jobs, die keinen besonderen Zeitpunkt zum Start aufweisen
- Job-Klasse JCTERMIN
  für Jobs, die zu einem bestimmten Zeitpunkt (Datum/Uhrzeit) gestartet werden sollen (Termin-Jobs)

Neben den Job-Klassen für Benutzer-Jobs existiert die vordefinierte System-Job-Klasse mit Namen $SYSJC für System-Jobs. Diese System-Job-Klasse sollte den Benutzern nicht zur Verfügung stehen, da $SYSJC alle Auftragstypen zulässt und keinerlei Begrenzung hinsichtlich der Klassenattribute enthält.

Es bestehen grundsätzlich zwei Möglichkeiten, Job-Klassen zu definieren, zu modifizieren oder zu löschen:

1. Die **statische** Definition ist in der Datei $TSOS.SJMSFILE hinterlegt. Diese Datei ist die Basis für jeden Systemlauf. Sie wird mit dem Dienstprogramm JMU erzeugt und verwaltet.

   Folgende JMU-Anweisungen stehen zur Verfügung:

   ```
   DEFINE-JOB-CLASS
   MODIFY-JOB-CLASS
   DELETE-JOB-CLASS
   GRANT-JOB-CLASS-ACCESS
   SET-JOB-CLASS-DEFAULT
   SET-POSIX-JOB-CLASS-DEFAULT
   ```

   Will die Systembetreuung die Attribute einer Job-Klasse modifizieren oder die Zuordnung einer Benutzerkennung zu einer Job-Klasse in der Datei $TSOS.SJMSFILE ändern, so wirken sich diese Änderungen erst im nächsten Systemlauf aus.

2. Die **dynamische** Definition bezieht sich nur auf den aktuellen Systemlauf. Sie wird ebenfalls mit dem Dienstprogramm JMU verwaltet, wobei der Bearbeitungsmodus `SET-MODIFICATION-MODE=*SYSTEM` eingestellt sein muss. Es stehen die selben JMU-Anweisungen wie oben zur Verfügung. Bei `SET-MODIFICATION-MODE=*ALL` werden die Änderungen auch in die Datei $TSOS.SJMSFILE übernommen.

   Darüber hinaus gibt es Eingriffsmöglichkeiten über die Kommandoschnittstelle.
   Mit folgenden Kommandos reagiert die Systembetreuung kurzfristig und schnell auf Überlastsituationen, ohne die Systemdatei $TSOS.SJMSFILE ändern zu müssen:

```
HOLD-JOB-CLASS
MODIFY-JOB-CLASS
RESUME-JOB-CLASS
```

11.1.2 Job-Streams, Job- und Klassen-Scheduler

Die Job-Klassen werden unterschiedlichen Job-Schedulern zugeordnet (mit dem Dienstprogramm JMU, siehe Handbuch „Dienstprogramme“ [15]).

Jeder Job-Scheduler läuft in einem eigenen Job ab, genannt „Job-Stream“.

Gemäß den vereinbarten Job-Scheduling-Strategien entscheiden die Job-Scheduler, welcher der Jobs zum Start freigegeben wird.

Job-Streams – und damit implizit die Job-Scheduler – steuern die Auswahl der Benutzer-Jobs, indem jedem Job-Stream ein passender Job-Scheduling-Algorithmus zur Verfügung steht.

Beispiel

In der Regel hat jedes Data Center verschiedene Produktionsschwerpunkte:

1. Die Programmierung benötigt für ihre Jobs zum Übersetzen, Binden und Laden von Programmen keine speziellen Betriebsmittel.
2. Die Jobs der Fachabteilungen sind mit wenigen Ausnahmen langlaufende Jobs mit hoher Priorität und hohen Betriebsmittelanforderungen.
3. Das Data Center erstellt für administrative Zwecke eine Reihe von Jobs, die zu bestimmten Zeiten ablaufen und wiederholt werden müssen.

Es bietet sich an, für diese drei Produktionsgebiete drei Job-Streams (Job-Scheduler) mit den folgenden Job-Scheduling-Strategien zu installieren:

zu 1) Für die Jobs der Programmierung kommt ein auf dem FIFO-Prinzip basierender Job-Scheduler in Frage (FIFO = First In First Out).
Jobs zum Übersetzen, Binden von Programmen etc. können z.B. in Job-Klassen zusammengefasst werden, die einem Job-Scheduler zugeordnet sind, der nur nachts aktiv ist.
Damit kann den unterschiedlichen Belastungen, denen ein Data Center tagsüber ausgesetzt ist, Rechnung getragen werden.

zu 2) Für die Jobs der Fachabteilungen eignet sich ein auf Priorität und Betriebsmittelverbrauch basierender Job-Scheduler.

zu 3) Für die Jobs des Data Centers bietet sich ein Scheduling-Algorithmus an, der die Termin-Jobs berücksichtigt.

Die Systembetreuung bestimmt mit der JMU-Anweisung DEFINE-JOB-STREAM die Kriterien, die dem Job-Stream zu Grunde liegen sollen und gemäß denen der Job-Scheduler die Jobs dann steuert. So kann z.B. die Auftragspriorität oder die benötigte CPU-Zeit als Kriterium über den Operanden S-PAR angegeben werden.

Neben dem Scheduling-Algorithmus gibt es noch weitere Eigenschaften, die einen Job-Stream kennzeichnen:

- Ein Job-Stream muss nicht immer aktiv sein. Die Systembetreuung entscheidet, ob der Stream z.B. bereits nach dem Laden von BS2000 oder erst in der Nacht aktiv sein soll. Jobs, die für einen inaktiven Stream eingelesen wurden, werden solange in einer speziellen Warteschlange (TYPE1/DO) gesammelt, bis der Stream aktiv ist und der Job-Scheduler die Verwaltung übernimmt.
- Bei der Definition des Job-Streams wird für den Stream eine Task-Scheduling-Priorität vereinbart, mit der der Stream ab dem Startzeitpunkt geführt wird. Da jeder Job-Stream in einem eigenen Job läuft, ist auf diese Weise die Rangfolge eindeutig. Mit dem Kommando MODIFY-JOB-STREAM kann diese Rangfolge verändert werden.

- Ein Job-Stream verwaltet nur Batch-Jobs. Dialog-Jobs unterliegen keinem Job-Scheduling und werden gestartet, wenn der Benutzer das Recht hat, in der jeweiligen Job-Klasse Dialog-Jobs zu starten und wenn die Job-Klasse ihr festgelegtes Limit noch nicht erreicht hat.

Zur Definition der Job-Streams und damit implizit der Job-Scheduler stehen zwei Möglichkeiten zur Verfügung:

1. Die **statische** Definition, die in der Datei $TSOS.SJMSFILE hinterlegt ist, ist die Basis für jeden Systemlauf. Sie wird mit dem Dienstprogramm JMU erzeugt und verwaltet.

   Die dazu notwendigen JMU-Anweisungen lauten:

   ```
   DEFINE-JOB-STREAM
   DELETE-JOB-STREAM
   MODIFY-JOB-STREAM
   ```

   Änderungen in dieser Datei wirken sich erst beim nächsten Systemlauf aus.

2. Die **dynamische** Definition bezieht sich nur auf den aktuellen Systemlauf. Sie wird ebenfalls mit dem Dienstprogramm JMU verwaltet, wobei der Bearbeitungsmodus `SET-MODIFICATION-MODE=*SYSTEM` eingestellt sein muss. Es stehen die selben JMU-Anweisungen wie oben zur Verfügung. Bei `SET-MODIFICATION-MODE=*ALL` werden die Änderungen auch in die Datei $TSOS.SJMSFILE übernommen.

   Darüber hinaus gibt es Eingriffsmöglichkeiten über die Kommandoschnittstelle.

   Mit folgenden Kommandos reagiert die Systembetreuung kurzfristig und schnell auf Überlastsituationen, ohne die in der Datei $TSOS.SJMSFILE hinterlegten Definitionen ändern zu müssen:

   ```
   HOLD-JOB-STREAM
   MODIFY-JOB-STREAM
   RESUME-JOB-STREAM
   START-JOB-STREAM
   STOP-JOB-STREAM
   ```

Der Name der ENTER-Datei, die den Job-Stream definiert, ist in der Datei $TSOS.SJMSFILE hinterlegt. Der Name ist prinzipiell frei wählbar, aus Konsistenzgründen wird SYSENT.JOBSCHED.nnn empfohlen („nnn“ ist das Kennzeichen der Betriebssystemversion).

Der Job-Scheduler wird aus dieser Datei mit START-EXECUTABLE-PROGRAM aufgerufen, womit sein Name festgelegt wird.
Auch der Name des Job-Schedulers ist somit prinzipiell frei wählbar; aus Gründen der Konsistenz, für die die Systembetreuung verantwortlich ist, wird der Name SYSPRG.JOB-SCHED.nnn empfohlen („nnn“ ist das Kennzeichen der Betriebssystemversion).

Für den Stream-ENTER werden vereinbarte Logon- bzw. Logoff-Prozeduren nicht ausgeführt (zu Logon-/Logoff-Prozeduren siehe Handbuch „Dialogschnittstelle SDF“ [43]).

Von ihrer Bedeutung und Funktion sind zu unterscheiden:

- der System-Job-Scheduler oder Emergency-Job-Scheduler $SYSJS
- die Standard-Job-Scheduler
- der Klassen-Scheduler

11.1.2.1 System-Job-Scheduler

Im Gegensatz zum Standard-Scheduler ist der System- oder Emergency-Job-Scheduler Bestandteil des mit dem Betriebssystem gekoppelten Job-Streams $SYSJS.

Er läuft im Funktionszustand TPR ab und verwaltet nur Jobs, die der Systemklasse $SYSJC zugeordnet sind. Die Scheduling-Strategie, auf dem LIFO-Prinzip (Last In First Out) basierend, ist nicht beeinflussbar, das heißt, der START-Parameter wird ignoriert.

Dies ist vor allem dann von Vorteil, wenn bei der Inbetriebnahme der Job-Steuerung Fehler auftreten und wenn Fehler beim Eröffnen oder beim Zugriff auf die Dateien $TSOS.SJMSFILE oder $TSOS.SJMSFILE.WORK angezeigt werden.

Es besteht dann immer die Möglichkeit, die Dateien, entsprechend dem Fehlerfall, zu rekonstruieren und die entsprechenden Jobs zu starten.

Mit folgenden Kommandos können auch die Systemklasse $SYSJC und der System-Job-Scheduler $SYSJS bearbeitet werden:

```
HOLD-JOB-STREAM
HOLD-JOB-CLASS
RESUME-JOB-STREAM
RESUME-JOB-CLASS
SHOW-JOB-STREAM
SHOW-JOB-CLASS
```

Die Definitionen von $SYSJS und $SYSJC sind konstant und werden vom Dienstprogramm JMU nicht angesprochen.

Der System-Job-Scheduler lässt unabhängig vom Standard-Job-Scheduler während des gesamten Systemlaufs das Starten von Jobs unter der Benutzerkennung TSOS zu. Wenn die Systembetreuung für die Benutzer keine Job-Klassen definiert hat, ist die System-Job-Klasse $SYSJC die Standard-Job-Klasse für die Benutzer, und damit verbunden werden alle Benutzer-Jobs vom System-Job-Scheduler $SYSJS bearbeitet.

11.1.2.2 Standard-Job-Scheduler

Der Standard-Job-Scheduler entscheidet, welcher der Jobs der ihm zugeordneten Job-Klassen zum Start freigegeben wird.

Dabei bedient sich der Standard-Job-Scheduler einer festgelegten Job-Scheduling-Strategie, die bei der statischen Definition mit dem Dienstprogramm JMU festgelegt und mit dem Kommando MODIFY-JOB-STREAM dynamisch verändert werden kann.
Gleichzeitig können maximal 16 Standard-Job-Scheduler mit unterschiedlichen Scheduling-Strategien installiert sein.
Ein Standard-Job-Scheduler läuft in einem eigenen Job (Job-Stream) ab. Es sind Benutzerprogramme im Funktionszustand TU unter der Kennung der Systemverwaltung.

Sollten die Auftragssteuerungs-Möglichkeiten des Standard-Job-Schedulers für die Belange des Data Centers nicht ausreichen, lässt sich ein selbst entwickelter Job-Scheduler zum Einsatz bringen (siehe hierzu Abschnitt „Job-Steuerung im Data Center").

In beiden Fällen wird der Scheduler unter der Kennung der Systemverwaltung gestartet (automatisch oder mit dem Kommando START-JOB-STREAM, je nachdem was für den Operanden START in der Streamdefinition des Schedulers im SJMSFILE angegeben wurde).

Job-Scheduling-Parameter

Bei der Stream-Definition mit der JMU-Anweisung DEFINE-JOB-STREAM werden mit dem Operanden STREAM-PARAMETER (kurz S-PAR) Scheduler-spezifische Parameter definiert.

Dieser Parameter bleibt von BS2000 unberücksichtigt; er wird lediglich vom Job-Scheduler ausgewertet und liefert die Basis für die Berechnung des Scheduling-Algorithmus.

Für den Standard-Job-Scheduler werden mit dem S-PAR-Operanden folgende Parameter vereinbart:

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'

Hinweise

1. Die Definitionsvarianten der ersten drei Parameter entsprechen den in Tabelle 26 aufgeführten Scheduling-Strategien.

2. Aus der Reihe der Jobs werden die Jobs ausgewählt, für die bei der Berechnung des Scheduling-Algorithmus die niedrigste Rangfolge ermittelt wird.
   Die Anzahl der ausgewählten Jobs wird vom Parameter JOB-QUOTA bestimmt.
   Der Job-Scheduler versucht nun, diese Jobs gemäß ihrer Reihenfolge in der Warteschlange zu starten. Nach erfolgtem Start werden wiederum Jobs aus der Reihe der Jobs ausgewählt. Dies wiederholt sich so lange, bis alle Jobs gestartet sind oder das Job-Klassen-Limit erreicht ist.
3. Der LOGGING-Parameter steuert die Protokollausgabe des Standard-Job-Schedulers. Außerdem legt der LOGGING-Parameter fest, ob eventuell vom Job-Scheduler erzeugte Meldungen an der Konsole ausgegeben werden sollen.

   Zu diesen Meldungen zählen:

   - `JMS0302`: Fehler beim Ausführen eines ASSIGN-SYSFILE-Kommandos
   - `JMS0303`: Protokollausgabe wegen Speicherplatzmangel abgebrochen
   - `JMS0304`: Nicht behebbarer Fehler
   - `JMS0305`: Ungültiger Stream-Parameter
   - `JMS0306`: Job-Stream wartet auf die Verfügbarkeit der angegebenen Pubsets
   - `JMS0307`: Job-Stream wird wegen fehlender Pubsets beendet
   - `JMS0308`: Job-Stream wartet auf Verfügbarkeit des Subsystems GET-TIME
4. Der Parameter CATID-LIST vereinbart die Pubsets, die vor der Inbetriebnahme des Job-Schedulers importiert sein müssen. Der Parameter wird nur zum Zeitpunkt des Stream-Starts berücksichtigt und verzögert das Starten des Job-Schedulers, bis sämtliche, bei CATID-LIST angegebenen Pubsets verfügbar sind. Fehlt der Parameter CATID-LIST, so startet der Job-Scheduler unabhängig davon, welche Pubsets importiert sind.
   Zur Steuerung dieser Wartezeit dient der Parameter CAT-TIME.
   Kann ein Pubset innerhalb der bei CAT-TIME angegebenen Zeit nicht importiert werden, wird der Job-Scheduler nicht gestartet.
   Fehlt der Parameter CAT-TIME in der Stream-Definition, wartet das System beliebig lange auf das Importieren der benötigten Pubsets. Die Zeitangabe erfolgt beim Parameter CAT-TIME in Minuten.
5. Der Parameter MSG-OUTPUT steuert das Verhalten des Job-Schedulers bei der Meldungsausgabe. Wird der Parameter auf den Wert YES gesetzt, dann werden vom Job-Scheduler erzeugte JMS-Meldungen auch dann an der Konsole ausgegeben, wenn ansonsten keine Protokollierung gewünscht ist (LOGGING = NO). Wird der Parameter auf den Wert STD gesetzt, dann erfolgt die Meldungsausgabe wie bisher in Abhängigkeit vom Parameter LOGGING.
6. Über die Kommandoschnittstelle kann die Systembetreuung die Einstellung des Operanden S-PAR dynamisch ändern (siehe Kommando MODIFY-JOB-STREAM im Handbuch „Kommandos“ [27]).

Job-Scheduling-Algorithmus

Die Job-Scheduler übergeben ihre zum Start freigegebenen Jobs an den Klassen-Scheduler. Die Reihenfolge, innerhalb der die Jobs zum Start weitergegeben werden, wird mit folgender Formel errechnet:

$M= (S^a * P^c * R^d) / (W^b + S^{(a*b)})$

Dabei bedeutet:

M	Rangfolge
S	CPU-Zeit in Sekunden, die der Job angefordert hat
P	Job-Scheduling-Priorität
W	Wartezeit des Auftrags in Minuten nach Job-Annahme
R	Zeit in Minuten, die beim Starten des Jobs noch verbleibt, bis der späteste Startzeitpunkt erreicht ist. Voraussetzung ist die Verwendung des Start-Attributes START=WITHIN oder START=LATEST.
a,b,c	Job-Scheduler-Parameter, die den Wert 0 oder 1 annehmen können, werden von der Systembetreuung bei der Stream-Beschreibung festgelegt. Dabei entspricht: • CPU-TIME der Variablen a • WAIT-TIME der Variablen b • OB-PRIORITY der Variablen c
d	d=1, wenn der Job vom Benutzer mit den Start-Attributen START=WITHIN oder START=LATEST bereitgestellt wird. Ansonsten gilt: d=0.

> ¡ Je kleiner der Wert für M ist, desto besser ist die Position des Jobs, dies gilt unabhängig von der Job-Klasse.

Wird für mehrere Jobs der gleiche Wert M ermittelt, dann wird die Reihenfolge der zum Start freigegebenen Jobs nach dem FIFO-Prinzip bestimmt.

Im Folgenden werden die einzelnen Scheduling-Verfahren erläutert, die sich aus den verschiedenen Kombinationen der Exponentenwerte ergeben. Die Wertzuweisung für die Exponenten a, b, c erfolgt bei der Stream-Definition durch den Operanden S-PAR.

Die Scheduling-Strategien sind in der Tabelle nur mit der jeweiligen Abkürzung angegeben. Die dazu gehörende Langform ist im Anschluss an die Tabelle vermerkt.

Scheduling-Strategie	CPU-TIME (a)	WAIT-TIME (b)	JOB-PRIORITY (c)
(1) FIFO	NO	YES	NO
(2) HPF	NO	NO	YES
(3) HPA	NO	YES	YES
(4) SJF	YES	NO	NO
(5) SJP	YES	NO	YES
(6) HRN	YES	YES	NO
(7) HRP	YES	YES	YES

Tabelle 26: Scheduling-Strategien

(1) *Auswahl nach Ankunftszeit (FIFO)*

Die Strategie empfiehlt sich, wenn die CPU-Zeitanforderung der freizugebenden Jobs nicht stark voneinander abweicht.
Ist dies nicht der Fall, dann führt die ausschließliche Verwendung dieser Strategie dazu, dass langlaufende Jobs bevorzugt werden.

Auswahl der Jobs, die mit der Option START=WITHIN bzw. START=LATEST beim ENTER-JOB-Kommando versehen wurden (FIFO und R (Wartezeit)):

Basierend auf dem FIFO-Auswahlprinzip wird der STREAM-Parameter WAIT-TIME=YES ausgewertet.
Sind die Jobs mit der Start-Option START=WITHIN in das System gekommen, dann werden die Jobs erst ab Beginn des Startintervalls in die Auswahlentscheidung einbezogen. Für Jobs mit START=LATEST wächst die Wahrscheinlichkeit, gestartet zu werden, mit der Annäherung an die späteste gewünschte Startzeit.

FIFO kann bei WAIT-TIME=YES nicht garantiert werden, wenn ein Stream-Start läuft und ein TSN-Überlauf aufgetreten ist. Wurden zwei Jobs in derselben Minute gestartet und tritt gerade dann ein TSN-Überlauf auf, wird der Job mit der TSN 0AAA vor dem Job mit der letzten vergebbaren TSN (9999) gestartet.

(2) *Auswahl nach Priorität (HPF)*

Die Jobs werden gemäß ihrer extern zugewiesenen Job-Scheduling-Priorität ausgewählt. Da immer der Job zum Start freigegeben wird, der die beste Priorität nachweisen kann, ist die Bevorzugung privilegierter Jobs gewährleistet.

Auswahl der Jobs, die mit der Option START=IMMEDIATE bzw. START=AT beim ENTER-JOB-Kommando versehen wurden:
Diese Funktionen bewirken, dass der Scheduling-Algorithmus den Wert M (Rangfolge) für Jobs mit der Option START=IMMEDIATE automatisch auf 0 setzt. Für Jobs mit der Option START=AT wird M auf 0 gesetzt, wenn die angegebene Startzeit erreicht wurde. Jobs für die M=0 ermittelt wird, werden sofort zum Starten freigegeben und können von Jobs mit M>0 nicht überholt werden.

(3) *Auswahl nach Priorität und Alterung (HPA)*

Durch Einbeziehung der Wartezeit in die Auswahlstrategie wird erreicht, dass selbst bei permanentem Angebot hochpriorer Jobs auch solche mit niedriger Priorität zum Klassen-Scheduler weitergeleitet werden.

(4) *Auswahl nach Laufzeit (SJF)*

Diese Strategie bevorzugt kurzlaufende Jobs und verkürzt im Gegensatz zur FIFO-Methode die mittlere Wartezeit. Da Batch-Jobs bei Ablauf der angeforderten CPU-Zeit abgebrochen werden, kann die Strategie nicht unterlaufen werden. SJF verzögert den Start von langlaufenden Jobs und kann im Extremfall – nämlich bei genügend kurzlaufenden Jobs – dazu führen, dass langlaufende Jobs gar nicht zum Start freigegeben werden.

(5) *Auswahl nach CPU-Zeit und Priorität (SJP)*

Bei dieser Strategie werden die Methoden „Auswahl nach Laufzeit“ und „Auswahl nach Priorität“ kombiniert.
Bei Jobs mit gleicher CPU-Zeit-Anforderung ist die Priorität maßgebend; bei Jobs mit gleicher Job-Scheduling-Priorität entscheidet die angeforderte CPU-Zeit.

(6) *Auswahl nach Response Ratio (HRN)*

Response Ratio ist das Verhältnis (Wartezeit + CPU-Zeit) / CPU-Zeit und kombiniert die beiden Verfahren HPA und SJF. Obwohl wiederum die kurzlaufenden Jobs bevorzugt werden, wird jedoch die Wartezeit der Jobs berücksichtigt.

(7) *Auswahl nach Durchsatz und Priorität (HRP)*

Im Vordergrund steht die Bevorzugung von kurzlaufenden Jobs. Die Rangfolge der Jobs ist festgelegt über die jeweilige Job-Scheduling-Priorität.

Protokoll des Standard-Job-Schedulers

Der Standard-Job-Scheduler protokolliert bestimmte Ereignisse der Stream-Tasks in die Protokolldatei SYS.SCHEDLOG.<yyyy.mm.dd>.<hh.mm.ss>.<streamname>, wenn in der Stream-Definition die Protokollierung eingeschaltet ist
(in der JMU-Anweisung DEFINE-JOB-STREAM oder im Kommando MODIFY-JOB-STREAM, Operand S-PAR='..., LOGGING=YES').
Das Protokoll gibt einen Überblick über die Aktivitäten des Schedulers während eines Systemlaufs. Darüber hinaus dient es als Unterlage bei einer evtl. erforderlichen Fehleranalyse.

Ausschnitt aus einer Protokolldatei (die Nummern sind nicht Teil der Protokolldatei, sondern dienen der Zuordnung zum erläuternden Text):

```
FTS BS2000   JOB SCHEDULER JSSTD1 VERSION <version> <date> <time>    PAGE 1
**               STREAM STARTED
** JSSTD1         JSATTCH  1.
**               EVENT         10    (JS_HOLD)       16.
**               EVENT         11    (JS_RELEASE)    17.
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCDSTD
**               JSWAKE   4.                              WAKE-TIME=2147483647
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCDSTD
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCDSTD
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCDSTD
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSM
**               JSRUNJB  3.                              TSN=0FSM, JCLASS=JCBSTD,
                                                          WAIT-TIME=0, JOB-PRIO=9,
                                                          CPU-TIME=32000,
                                                          MERIT=45DBBA000000000000000001
**               EVENT          1    (J_TERMINATION)  7.  TSN=0FSM
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCBSTD
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSN
**               JSRUNJB  3.                              TSN=0FSN, JCLASS=JCBSTD,
                                                          WAIT-TIME=0, JOB-PRIO=9,
                                                          CPU-TIME=32000,
                                                          MERIT=45DBBA000000000000000002
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCDSTD
**               EVENT          1    (J_TERMINATION)  7.  TSN=0FSN
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCBSTD
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSQ
**               JSRUNJB  3.                              TSN=0FSQ, JCLASS=JCBSTD,
                                                          WAIT-TIME=0, JOB-PRIO=9,
                                                          CPU-TIME=32000,
                                                          MERIT=45DBBA000000000000000003
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSR
**               JSRUNJB  3.                              TSN=0FSR, JCLASS=JCBSTD,
                                                          WAIT-TIME=0, JOB-PRIO=9,
                                                          CPU-TIME=32000,
                                                          MERIT=45DBBA000000000000000004
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSS
**               JSWAKE   4.                              WAKE-TIME=0
** JOB_HOLD      JOB HELD                                 TSN=0FSS
**               EVENT          2    (J_HOLD)         8.  TSN=0FSS
**               JSWAKE   4.                              WAKE-TIME=10111560
**               EVENT          4    (J_CANCEL)      10.  TSN=0FSS
**               JSWAKE   4.                              WAKE-TIME=2147483647
**               EVENT               (J_INTRODUCTION) 6.  TSN=0FSS
**               JSWAKE   4.                              WAKE-TIME=0
**               EVENT         16    (JS_TIMER)      21.  DATE=<date>, TIME=<time>
**               JSWAKE   4.                              WAKE-TIME=10111500
**               EVENT          3    (J_RELEASE)      9.  TSN=0FSS
**               EVENT          1    (J_TERMINATION)  7.  TSN=0FSQ
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCBSTD
**               EVENT          1    (J_TERMINATION)  7.  TSN=0FSR
**               EVENT          9    (JC_AVAILABLE)  15.  JCLASS=JCBSTD
**               EVENT          4    (J_CANCEL)      10.  TSN=0FSS
**               JSWAKE   4.                             WAKE-TIME=2147483647
**               EVENT         13    (JS_CLOSE_IMMEDIATE) 19.
**               STREAM CLOSED
** JSSTD1         JSDETCH  2.
```

Erläuterung der Funktionen des Job-Schedulers (über die Protokollierung sichtbare Schritte sind in fett dargestellt):

1. **JSATTCH**
   Der Job-Stream informiert mit einer Meldung über seine abgeschlossene Inbetriebnahme.
2. **JSDETCH**
   Der angegebene Job-Stream befindet sich in der Beendigungsphase. Die Job-Steuerung unterstützt den Job-Scheduler nicht mehr. Darüber hinaus wird auch die Task, in der der Job-Stream realisiert ist, beendet.
3. **JSRUNJB**
   Der Job-Scheduler fordert den Klassen-Scheduler auf, den angegebenen Job zu starten.
   Falls für die Job-Klasse das CLASS-LIMIT erreicht ist, erhält der Standard-Job-Scheduler eine entsprechende Rückinformation.
4. **JSWAKE**
   Der Job-Scheduler teilt der Job-Steuerung mit, wann er wieder aktiviert werden will.
5. **JSEXPCT**

   Die Job-Steuerung erhält die Aufforderung, dem Job-Scheduler das nächste vorliegende Ereignis zu übergeben. Dazu zählen dann (siehe Nummern 6 - 21):
6. **J_INTRODUCTION**
   Der Job-Scheduler wird aufgefordert, einen Job in die Menge der von ihm verwalteten Jobs aufzunehmen.
   Dabei werden dem Job-Scheduler folgende Informationen zur Verfügung gestellt:
   Auftragsnummer, Name der Job-Klasse, Wiederholungsanzeige, CPU-Zeit, Job-Sche-duling-Priorität, Ankunftszeit des Jobs, Start-Attribut, Job-Parameter.
7. **J_TERMINATION**
   Ein vom betreffenden Job-Scheduler verwalteter Job wird nach seiner Laufzeit beendet.
8. **J_HOLD**
   Ein Job wird angehalten und steht in der entsprechenden Warteschlange.
9. **J_RELEASE**
   Ein angehaltener Job wird wieder freigegeben. Der Job-Scheduler ist aufgefordert, den Job zu berücksichtigen.
10. **J_CANCEL**
    Ein vom Job-Scheduler noch nicht gestarteter Job wird abgebrochen.
11. **J_EXPRESS**
    Annahme eines Jobs, der sobald wie möglich gestartet werden soll. Der Job erhält die Rangfolge M=0 (siehe „Job-Scheduling-Algorithmus").
12. **J_RESCHEDULE**
    Eine mit dem Kommando MODIFY-JOB vorgenommene Änderung der Job-Eigenschaften wird angezeigt.
13. **JC_HOLD**
    Die angegebene Job-Klasse wird in den Wartezustand versetzt. Der Job-Scheduler ist aufgefordert, keine Jobs der angehaltenen Job-Klasse weiter zu berücksichtigen.
14. **JC_RELEASE**
    Die angehaltene Job-Klasse und damit implizit alle Jobs dieser Job-Klasse werden wieder freigegeben.
15. **JC_AVAILABLE**
    Für die angegebene Job-Klasse ist der bei CLASS-LIMIT festgelegte Wert unterschritten worden.
16. **JS_HOLD**
    Der angegebene Job-Stream wird in den Wartezustand versetzt.

17. **JS_RELEASE**
    Der angehaltene Job-Stream wird wieder freigegeben. Er ist angewiesen, alle von ihm zu verwaltenden Jobs zu bearbeiten.
18. **JS_CLOSE QUIET**
    Anzeige, dass sich das Betriebssystem in der Beendigungsphase befindet. Dieses Ereignis hat die gleiche Wirkung wie JS_HOLD, d.h. es werden keine weiteren Jobs gestartet.
19. **JS_CLOSE IMMEDIATE**
    Nach Bearbeitung des Kommandos STOP-JOB-STREAM wird der Job-Scheduler sofort beendet.
20. **JS_CHANGE**
    Es werden Informationen über geänderte STREAM-PARAMETER ausgegeben.
21. **JS_TIMER**
    In Minutenabständen wird der Job-Scheduler angewiesen, die Start-Attribute seiner Jobs zu überprüfen.

11.1.2.3 Klassen-Scheduler

Der Klassen-Scheduler übernimmt im Rahmen der Job-Steuerung folgende Aufgaben:

Starten der Batch-Jobs

Wie das Bild 17 ("Job-Steuerung") zeigt, übergeben sämtliche Job-Scheduler ihre zum Start freigegebenen Jobs an den Klassen-Scheduler.
In Abstimmung mit der Task-Steuerung wird ein vom Job-Scheduler freigegebener Job vom Klassen-Scheduler sofort gestartet.

Überwachen der Job-Klassengrenzen

Mit der JMU-Anweisung DEFINE-JOB-CLASS und dem Operanden CLASS-LIMIT wird für jede Job-Klasse eine Grenze festgelegt, bis zu der Jobs gestartet werden können.
Die Systembetreuung hat damit – bezogen auf alle Job-Klassen – einen Job-Mix vereinbart, für dessen Einhaltung der Klassen-Scheduler verantwortlich ist. Wenn dem Klassen-Scheduler ein Job übergeben wird, überprüft er, ob das CLASS-LIMIT erreicht ist. Ist dies der Fall, wird der Job zurückgewiesen (Ausnahme: ein Job mit Start-Attribut IMMEDIATE).Sobald der Grenzwert wieder unterschritten ist, z.B. durch Job-Ende oder durch Erhöhung des Limits per Kommando MODIFY-JOB-CLASS, wird der entsprechende, die Job-Klasse verwaltende Scheduler informiert und ein Job an den Klassen-Scheduler weitergegeben.

Zwischenspeichern von Jobs bei Systemsättigung

Bei Sättigung des Seitenwechselbereichs oder des Hauptspeichers nimmt der Klassen-Scheduler von den einzelnen Schedulern zwar noch Jobs an, speichert sie aber in klassenspezifischen Warteschlangen zwischen.
Die Einreihung der Jobs in diese Warteschlangen erfolgt nach dem FIFO-Prinzip. Eine Ausnahme bilden die Jobs, die mit dem Start-Attribut IMMEDIATE (sog. Express-Jobs) versehen sind. Diese Jobs stehen in den Warteschlangen an vorderster Stelle. Treffen mehrere Express-Jobs beim Klassen-Scheduler ein, dann steht der zuletzt eingetroffene Job als erster in der Warteschlange.

Starten dieser Jobs nach Aufheben der Sättigung

Nach Aufheben der Sättigung startet der Klassen-Scheduler die Jobs, indem er eine Job-Klasse, deren Klassenoptimum noch nicht erreicht ist, mit dem kleinsten Wert M nach folgendem Algorithmus auswählt:

$M = (C + 1) / W$

Dabei bedeutet:

M	Rangfolge
C	Anzahl der momentan in der Job-Klasse laufenden Jobs
W	Gewichtung (Dringlichkeit) der Job-Klasse

Der Klassen-Scheduler startet den Job, der als erster in der Warteschlange steht.

Nach jedem Job-Start und jeder Job-Beendigung muss der Klassen-Scheduler die Rangfolge erneut berechnen.

Bei Einsatz des Softwareprodukts PCS ist der Klassen-Scheduler mit dem Performance-Controller verbunden.

11.1.3 Die Job-Steuerung während der Systemeinleitung

Für die Job-Steuerung werden in der Phase der Systemeinleitung nachstehende Funktionen ausgeführt:

- Aus der Datei SJMSFILE werden die Job-Stream- und Job-Klassen-Definitionen sowie die Klassenzugriffsrechte eingelesen.
- Von der Datei SJMSFILE (Typ ISAM) wird eine Kopie unter dem Namen SJMSFILE.WORK (Typ SAM) angelegt.
- System- und Klassen-Scheduler werden aktiviert.
- Auf der Konsole wird zusätzlich die Anzahl der aus der SJMSFILE bzw. SJMSFILE.WORK gelesenen Objekte (Streams, Klassen, Zugriffslisten usw.) ausgegeben.

Sollte beim Startup der Fall eintreten, dass die Datei SJMSFILE defekt oder nicht vorhanden ist, übernimmt die im vorherigen Systemlauf erzeugte Datei SJMSFILE.WORK deren Funktionen. Da die SJMSFILE neu erzeugt werden muss, ist es zu empfehlen, einen Batch-Job mit dem entsprechenden JMU-Lauf bereitzuhalten.

11.1.4 Job-Steuerung im Data Center

Die Job-Steuerung bietet der Systembetreuung darüber hinaus Möglichkeiten in Richtung eines Job-Schedulings im Data Center an:

1. Installieren eines eigenen Job-Schedulers

   Über die Assembler-Schnittstelle kann ein individueller Job-Scheduler erstellt werden, der als Benutzerprogramm unter der Benutzerkennung der Systemverwaltung abläuft. Die dazu erforderlichen Makros sind im Handbuch „Makroaufrufe an den Ablaufteil“ [30] beschrieben.
   Mit der JMU-Anweisung DEFINE-JOB-STREAM (Operand S-PAR) definiert die Systembetreuung zusätzliche Parameter, die in ihrer Syntax und Semantik frei wählbar sind. Sie werden nur von dem individuellen Job-Scheduler ausgewertet.

2. Benutzen einer System-Exit-Routine

   Mit der JMU-Anweisung DEFINE-JOB-CLASS (Operand JOB-PAR) definiert die Systembetreuung zusätzliche Parameter, die in ihrer Syntax und Semantik frei wählbar sind. Sie werden nur von der System-Exit-Routine (System Exit 32) ausgewertet.

Die Notwendigkeit, zusätzliche Parameter zu definieren, könnte sich für ein Data Center aus der Tatsache ergeben, dass der Standard-Job-Scheduler beispielsweise innerhalb der Job-Klassen keinen Unterschied macht zwischen:

sehr wichtigen Jobs (Klassenattribut VIMP)

wichtigen Jobs (Klassenattribut IMP)

weniger wichtigen Jobs (Klassenattribut NIMP)

Die Systembetreuung vereinbart demnach für die betroffenen Job-Klassen:

```
DEFINE-JOB-CLASS...,JOB-PAR='VIMP'/'IMP'/'NIMP'
```

11.1.5 Unterbrechungsfreie Zeitumstellung

Für die Winter-/Sommerzeit-Umstellung muss aus Sicht der Job-Steuerung das System nicht mehr heruntergefahren und wieder neu gestartet werden:

Die Job-Steuerung arbeitet intern mit der UTC-Zeit, die monoton aufsteigend ist und Zeitumstellungen nicht kennt. Extern verwendet die Job-Steuerung die für die Systeminstallation eingestellte Zeit, die sog. Systemzeit, die auf der lokalen Zeit (LT, gültige gesetzliche Zeit am Installationsort) basiert. Das bedeutet, dass der Benutzer immer mit der gesetzlichen Ortszeit arbeiten kann.

Voraussetzung für die richtige Zeitumrechnung durch die Job-Steuerung sind die korrekten System-Parameterdateien mit Angaben für die Zeitumstellung (siehe "Steuerung der Systemzeit (GTIME)"). Zeitangaben werden also immer als gesetzlich gültige Ortszeit unabhängig von Zeitsprüngen interpretiert, z.B. Startzeitpunkte für Termin-Jobs.

Die konsequente Interpretation von Zeitangaben als gesetzliche Zeit hat auch ein geändertes Start-Verhalten von Repeat-Jobs mit dem Repeat-Attribut „DAILY“ bzw. „WEEKLY“ zur Folge: Bei der Akzeptierung solcher Repeat-Jobs wird der Startzeitpunkt für die tägliche oder wöchentliche Wiederholung bestimmt. Auch dieser Startzeitpunkt wird nun konsequent als gesetzliche Zeit interpretiert. So wird z.B. ein Job, der zur Sommerzeit täglich um 17 Uhr ablief, auch nach dem Wechsel auf Winterzeit täglich um 17 Uhr gestartet.

11.1.6 JMP: Rekonstruktion von Batch-Jobs

Das Programm JMP (Jobpool Management Program) rekonstruiert aus den Informationen, die das JMS über akzeptierte Batch-Jobs im Job-Pool (in der Systemdatei SYSTEM.JOB-POOL) 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.

Anweisung	**Bedeutung**
CREATE-PROCEDURE-FILE	Prozedurdatei mit den ENTER-JOB-Kommandos erzeugen
OPEN-JOBPOOL-FILE	Job-Pool-Datei öffnen
SHOW-JOBPOOL-STATUS	Informationen über den Inhalt einer Job-Pool-Datei ausgeben
END	JMP beenden

Tabelle 27: Übersicht der JMP-Anweisungen

Eine ausführliche Beschreibung von JMP und Beispiele für rekonstruierte ENTER-JOB-Kommandos enthält das Handbuch „Dienstprogramme“ [15].

11.2 Task-Steuerung

Während die Job-Steuerung alle Jobs verwaltet, werden alle Tasks von der Task-Steuerung gesteuert und verwaltet.

Die zentralen Aufgaben der Task-Steuerung lassen sich wie folgt zusammenfassen:

- Tasks erzeugen
- Unterbrechungsanalyse und -bearbeitung
- Verwaltung der Registersicherstellungsbereiche
- Versorgung zentraler Tabellen des Ablaufteils
- Tasks aktivieren
- Tasks initiieren
- Tasks innerhalb und zwischen den Warteschlangen bewegen
- Tasks deaktivieren
- Tasks beenden

Unter dem Namen PRIOR sind nun jene Routinen der Task-Steuerung zu verstehen, die folgende Steuerungen übernehmen:

- die Steuerung von Tasks mittels Kategorien und Task-Scheduling-Prioritäten
- die Steuerung der System-Auslastung durch interne Regelfunktionen

Die Tasks sind allgemein Träger für unterbrechbare Abläufe in BS2000, wobei zwischen Benutzer- und Systemtasks unterschieden wird.

Benutzertask

Als Benutzertask werden alle – auch privilegierte – Benutzer-Jobs geführt, die von der Job-Steuerung der Task-Steuerung übergeben worden sind als:

- Dialogtask (Teilnehmerbetrieb; Task-Typ X'40')
- Batchtask (Batch-Betrieb; Task-Typ X'20')

Systemtask

Systemtasks nehmen die unterschiedlichsten Systemaufgaben wahr, wobei einige permanent sind (TCB im Klasse-1-Speicher vorgeneriert), die meisten jedoch dynamisch bei Bedarf erzeugt und wieder beendet werden.
Diesen Systemtasks sind vorgegebene Task-Identifier (TID) und TSNs zugeordnet, nach denen sie eindeutig identifiziert werden können.
Die wichtigsten Systemtasks, permanente und dynamisch (während Startup oder bei Bedarf) erzeugte, sind in den folgenden Übersichten zusammengefasst.

TID	TSN	Aufgaben
0001 0001	TSC	Emergency Job Scheduler $SYSJS für die Job-Klasse $SYSJC, in der insbesondere alle anderen Job Scheduler laufen.
0001 0003	RMM	Prüfung des Hauptspeichers bei Startup
0001 0005	PT5	Periodic Task 5: Prüft alle 10 Sekunden u.a. die LIFETIME (Wartezeitüberschreitung) beim Börsenmechanismus
0001 0006	PT6	Periodic Task 6 des Job-Managements: Für Operationen, die zu einem bestimmten Zeitpunkt („AT“) durchgeführt werden sollen; Überprüfen der Repeat-Jobs (Intervall abgelaufen ?)
0001 0007	PGE	Paging Error Recovery Task
0001 0008	UCO	Universelle Konsole UCON: Verteilt die Konsolkommandos und Meldungen an die zuständige Instanz (Operator-Task für Operator-Normalkommandos; $CONSOLE-Anwendung für Operator-Spezialkommandos); realisiert die Anwendung $CONSOLE
0001 0009	REK	Zuständig für Rekonfiguration
0001 000A	VMM	Memory Management Task: Verwaltet die Slotpools; z.B für Stacks, Börsenkontrollblöcke
0001 000B	MSG	Zuständig für Meldungswesen
0001 000C	KTT	Konsol Treiber Task: Ein-/Ausgabe-Treiber für Konsolgeräte

Tabelle 28: Permanente (preallocated) Systemtasks

TSN	gehört zu den ...	**Aufgaben**
RUNT		Operatortask: Zum Bearbeiten von aus einer RUN-Datei stammenden Operator-Normalkommandos
Xxxx		Allgemeine Arbeitstasks des Systems, z.B. MSCF-Servertask, Import-/Exporttask, Operatortask (Bearbeitung und Ausführung der Operator-Normalkommandos)
CLOG		Konsol-Logging (SYS.CONSLOG)
RP01		SRPM-Task; verwaltet den Benutzerkatalog des Home-Pubsets
RP02		SRPM-Task; verwaltet den Benutzerkatalog aller anderen Pubsets
HELT		Verwaltet die Hardware-Fehlerdatei SYS.HEL.<datum>.<uhrzeit>
HERS		HERS-Task; veranlasst HEL-Einträge bei Maschinenfehlern
SERS		Verwaltet die Fehlerdatei SYS.SERSLOG....
REPL		Führt die Loggingdatei REPLOG und protokolliert Rep before image
DSSM	Tasks der dyn. Subsystemverwaltung	Dynamic Subsystem Monitor
HTnn		Hilfs-Tasks für DSSM
NKA	Tasks des Nucleus Device Management (NDM)	Betriebsmittelverwaltung
DM		Disk-Monitor
TM		Tape-Monitor
NKS		Betriebsmittelbelegung
ACCT	Tasks des Abrechnungssystems	Bearbeitet die Abrechnungsdatei
PACT		Task für die Funktion Periodisches Accounting
BCAM	Tasks der Systemkomponente openNet Server	Nachrichtenpoolverwaltung; Verarbeitung der /BC-Kommandos
BCAH		Holder-Task für Daten, die einen /BCEND überdauern
BCAF		Dateizugriffstask (z.B. Lesen der Prozessor-Datei)
BCAC		Kryptografie-Task; behandelt ver- und entschlüsselte Jobs bei Einsatz von IPSec
BCA0		Transfer-Task für Datenverkehr über TCP/IP
BCAT		Transfer-Task für Datenverkehr über andere Protokolle
DIAA		TIAM-Task: erzeugt Dialog-Tasks (für virtuelle Hosts auch: DIAB, DIAC, ...)

DBxx		Tasks des Disk Access Buffers DAB
DRV		Task des Softwareprodukts DRV
SNAP		SNAP-Task für die Realisierung der Funktion SNAP-Dump
S	Systemtasks von SPOOL	Spool-Scheduler; erzeugt Spool-Tasks und versorgt sie mit Jobs, zuständig für die Spool-Kommandos
SPAx		Spool-Output-Writer; zuständig für SPOOLOUT
SPMG		Spool-Message-Task; zuständig für Meldungsausgabe
xxxx		Spool-Task (Device Controller Task); einem Spoolgerät zugeordnet (TSN wird vom System vergeben)
RSO		Remote SPOOL (RSO-Spool)
RSAx		RSO-Output-Writer
EQS		Verwaltung der Datei $TSOS.EQUISAMQ
M		New Message Handler
ISDS		Verwaltung der ISAM-Pools
IOTD IOTT		Tasks zur Ein-/Ausgabeverwaltung bei Rekonfiguration von Platten (IOTD) und Bändern (IOTT)
JOBP		Job-Pool-Controller-Task: Verwaltung der Job-Pool-Datei für Batch-Jobs
TSNA		TSN-File-Manager: Verwaltung der Datei TSOJOIN.TSNN
TIME TIMC		Timer-Management: Uhrensynchronisation mit externem Zeitgeber zur Sommer-/Winterzeitumstellung
DRTx		Domänen-Rekonfigurationstasks auf SU x86
MSCF MSCM WDGS	Systemtasks des Softwareprodukts MSCF	Tasks für die Realisierung der MRS-Kopplung
FTCP	Task des Softwareprodukts openFT	File Transfer Control Process (zentrale FT-Steuertask)
SM2G	Systemtasks des Softwareprodukts openSM2	Software-Monitor-Erfassungs-Task
SM2W		openSM2-Schreib-Task
SM2U		openSM2-Task für User-Messungen
SATP	Systemtasks des Softwareprodukts SECOS	SAT-Parameterdatei-Verwaltung
PRxx		GUARDS-Server-Tasks; verwalten den Guards-Katalog eines Pubsets

SATT		SECURITY-AUDIT-TRAIL-Task für die SAT-Dateiverwaltung

Tabelle 29: Wichtige, dynamisch erzeugte Systemtasks

Task-Kategorien

Jede Task ist einer Kategorie zugeordnet. Neben den vier Standardkategorien SYS, TP, DIALOG und BATCH unterstützt die Task-Steuerung 12 weitere Kategorien, deren Namen bei der Definition der Job-Klassen vereinbart werden.

Für die vier Standardkategorien gilt:

SYS	nur für Systemtasks
TP	für Transaktions-Tasks (Teilhaberbetrieb)
DIALOG	für Dialogtasks (Teilnehmerbetrieb)
BATCH	für Batch-Tasks (Batch-Betrieb)

Zusätzlich gibt es vier Task-Attribute, deren Namen identisch sind mit den Standardkategorienamen SYS, TP, DIALOG und BATCH. Den Task-Attributen sind spezielle, für das Task-Scheduling wichtige Ablaufparameter zugeordnet.

Das Task-Attribut TP zeichnet sich gegenüber den anderen Task-Attributen durch eine speziell auf die Bedürfnisse des Teilhaberbetriebs optimierte Hauptspeicher-Verwaltung aus. Dieses Attribut kann entweder durch Definition in der den Jobs zugeordneten Job-Klassen oder durch Aufruf des TINF-Makros (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30]) erlangt werden. Die dazu notwendige Berechtigung muss im Benutzerkatalog eingetragen sein.

Das Konzept der Task-Kategorien ist nur der Systemverwaltung (im Ausnahmefall auch dem Operator) über die Kommandoschnittstelle zugänglich.
Mit dem MODIFY-TASK-CATEGORIES-Kommando spezifiziert die Systembetreuung die Wichtigkeit der Kategorien untereinander für die Aktivierung (= Zuteilung des Betriebsmittels Hauptspeicher).

Zu diesem Zweck werden folgende Kategorie-Attribute vereinbart:

WEIGHT	Wichtigkeit (Gewicht)
MINMPL	Minimum an aktiv zu haltenden Tasks
MAXMPL	Maximum an aktiv zu haltenden Tasks
IO-PRIO	I/O-Priorität aller Tasks dieser Kategorie

Ob eine Task-Aktivierung, -Deaktivierung oder -Verdrängung durchgeführt wird, ist neben den Kategorie-Attributen noch abhängig von der System-Auslastung und der Priorität der betroffenen Task.

Da einer Kategorie in der Regel mehrere Tasks zugeordnet sind, ist die Kategoriensteuerung immer **gruppenspezifisch** zu betrachten. Will man eine Beurteilung der Auswirkungen der Kategoriensteuerung versuchen, so ist zwischen den folgenden Situationen zu unterscheiden:

- Im Unterlast- bzw. Normallast-Bereich hat die Kategoriensteuerung keinen nennenswerten Einfluss.
- Im Vollast- bzw. Überlast-Bereich (also bei Betriebsmittelengpässen) hat sie eine starke Wirkung und dient zur Lastbegrenzung, d.h. weniger wichtige Kategorien werden in den Hintergrund gedrängt.

Mit dem Kommando MODIFY-TASK-CATEGORIES, Operand IO-PRIORITY, kann die I/O-Priorität aller Tasks der angegebenen Katagorie bestimmt werden. Dieser Wert wird nur ausgewertet, wenn das Subsystem IORM gestartet ist. Die Voreinstellung *NONE bedeutet, dass alle Tasks dieser Kategorie eine aus der Task-Priorität abgeleitete I/O-Priorität erhalten (implizite I/O-Priorität). Die I/O-Prioritäten werden ausschließlich vom Subsystem IORM zur task-spezifischen Steuerung von Ein-/Ausgaben ausgewertet und zwar mit Ablaufsteuerung sowohl durch PRIOR als auch durch das Subsystem PCS.

Mit dem Kommando MOVE-TASK-TO-CATEGORY kann die Systembetreuung die Zuordnung einer Task zu einer Kategorie ändern, wenn etwa eine andere (bessere) Bedienung dieser Task oder auch eine Entlastung einer Kategorie erreicht werden soll (mit oder ohne Einsatz von PCS).
Zielkategorien müssen JMS-Kategorien und dürfen keine Folgekategorien sein. Die Kategorie SYS wird weder als Ausgangs- noch als Zielkategorie unterstützt.

Die Wahl geeigneter Kategorienwerte ist häufig ein Problem. Deshalb empfiehlt es sich (vor allem für große Server) über längere Zeiträume hinweg Performance-Untersuchungen durchzuführen. Die dazu notwendigen Daten liefert der Software Monitor openSM2 (siehe Handbücher „openSM2“ [49] und „SM2-PA“ [50]). Weitere Informationen enthält das „Performance-Handbuch“ [37].

Kommando	**Bedeutung**
ADD-USER	Eintrag im Benutzerkatalog erstellen mit der Vereinbarung, ob die Tasks des Benutzers deaktiviert werden dürfen oder nicht
CANCEL-JOB	Benutzerauftrag abbrechen
CHANGE-TASK-CPU-LIMIT	Maximale CPU-Zeit eines Batch-Jobs erhöhen
CHANGE-TASK-PRIORITY	Job- oder Task-Priorität ändern
ENTER-JOB	Kommandofolge, die in einer ENTER-Datei als Batch-Job gespeichert ist, unter Angabe der Task-Priorität starten
FORCE-JOB-CANCEL	Benutzerauftrag abbrechen
HOLD-TASK	Benutzerauftrag anhalten
MODIFY-TASK-CATEGORIES	Verteilung der Betriebsmittel CPU, Hauptspeicher und I/O-Prozessor auf die einzelnen Kategorien regeln, Dringlichkeiten der Kategorien festlegen und I/O-Prioritäten für alle Tasks einer Kategorie vergeben
MOVE-TASK-TO-CATEGORY	Kategorie-Zuordnung einer Task ändern
RESUME-TASK	Angehaltenen Benutzerauftrag freigeben
SET-LOGON-PARAMETERS	Dialog-/Batch-Job unter Angabe der Task-Priorität einleiten
SHOW-JOB-STATUS	Informationen über einen Benutzer-Job anfordern
SHOW-SYSTEM-STATUS	Informationen über Job-Klassen und Job-Streams anfordern
SHOW-USER-ATTRIBUTES	Maximale Task-Priorität abfragen
SHOW-USER-STATUS	Informationen über Gruppen von Benutzer-Jobs anfordern
Makro	**Bedeutung**
PASS	eine Sekunde warten
TINF	Task-Attribute lesen und ändern
TSPRIO	Runprioritäten ausgeben
VPASS	warten

Tabelle 30: Schnittstellenübersicht zur Task-Steuerung

11.2.1 Prioritätenkonzept und Warteschlangen

Die Ausführungsprioritäten lassen sich wie folgt einteilen:

Prioritäten 0 - 29	Prioritäten für Systemtasks
Prioritäten 30 - 127	Feste Task-Scheduling-Prioritäten
Prioritäten 128 - 255	Variable Task-Scheduling-Prioritäten

Mit Ausnahme der Systemprioritäten werden die Prioritäten bei der Job-Klassen-Definition und benutzerspezifisch im Benutzerkatalog festgelegt.

Gibt der Benutzer im SET-LOGON-PARAMETERS- bzw. ENTER-JOB-Kommando eine Priorität an, dann wird diese sowohl in der dem Benutzer zugewiesenen Job-Klasse als auch im Benutzerkatalog geprüft.

Die Priorität einer Task wird sowohl bei der Aktivierung als auch bei der Initiierung (Zuteilung des Betriebsmittels CPU) berücksichtigt.

Über den Systemparameter ETMFXLOW gibt es die Möglichkeit, auch einen unteren Bereich von festen Prioritäten einzurichten.

Variable Prioritäten

Charakteristisch für die variablen Prioritäten ist die dynamische Anpassung der Priorität mit dem HRN-Algorithmus (= Highest-Response ratio Next).

HRN basiert auf dem Verhältnis „Verweilzeit / verbrauchte CPU-Zeit“ unter Berücksichtigung der Start-Priorität beim SET-LOGON-PARAMETERS bzw. ENTER-JOB-Kommando oder einer extern zugewiesenen Priorität (Kommando CHANGE-TASK-PRIORITY).

Der HRN-Algorithmus bewirkt, dass Tasks, die wenig CPU-Leistung aufnehmen, und Ein-/Ausgabe-intensive Tasks bevorzugt behandelt werden, ohne dabei rechenintensive Tasks extrem stark zu benachteiligen.
Darüber hinaus wird eine angemessene Versorgung auch für Tasks mit niedriger Start-Priorität gewährleistet.

Die variable Priorität wird zu folgenden Zeitpunkten neu berechnet:

- bei jeder Aktivierung
- bei jedem PEND/UNPEND nach Q5
- bei Ablauf der Mikrozeitscheibe (einem von der CPU-Geschwindigkeit und vom unmittelbar zurückliegenden E/A-Verhalten der Task abhängigen Zeitquantum)
- periodisch (1/Sekunde)
- bei Absetzen des Kommandos MODIFY-TASK-CATEGORIES
- bei Absetzen des Kommandos CHANGE-TASK-PRIORITY

Feste Prioritäten

Die festen Prioritäten ändern sich nicht.

Für die betroffene Task stellt die feste Priorität eine starke Bevorzugung dar.
Feste Prioritäten sind für Anwendungen mit extremen Realzeitanforderungen entwickelt worden. Durch die Vergabe einer festen Priorität kann eine Performance-Verbesserung erreicht werden, jedoch unter Berücksichtigung folgender Punkte:
Feste Prioritäten engen den Entscheidungsspielraum des Systems stark ein. Zur Erzielung eines positiven Effektes müssen daher die Betriebsmittelanforderungen sämtlicher Tasks bekannt sein. Zur Aktivierung anstehende Tasks mit fester Priorität führen bei hoch ausgelasteten Betriebsmitteln zu einer sofortigen Verdrängung anderer Tasks, wodurch sich eine Überlastsituation ergeben kann. Das wiederum hat letztendlich negative Auswirkungen auf die Performance.

Feste Prioritäten sollten nur nach genauer Analyse der Last, der Betriebsmittel-Auslastung und zusammen mit lastbegrenzenden Maßnahmen (wie z.B. Beschränkung des Multiprogrammierungs-Grades der Kategorien durch den Operanden MAXIMUM-ACTIVE-TASKS im Kommando MODIFY-TASK-CATEGORIES) vergeben werden.

Zusammenfassend lässt sich Folgendes feststellen:

- Prioritäten, egal welche, beeinflussen die Task-Reihenfolge in den Warteschlangen.
- Jede Priorität, die besser ist als die „normale Priorität“ 255, beeinflusst das System unabhängig von der Lastsituation.
- Auf Grund der gesteigerten Bedeutung der Priorität kann die Systembetreuung durch Prioritätsänderung die Performance einzelner Tasks sehr leicht beeinflussen. Vor allem für leistungsschwächere Systeme mit nur wenigen Benutzertasks bietet sich an, die Tasks mit variablen Prioritäten unterschiedlicher Größenordnung zu belegen. Damit kann erreicht werden, dass jede Task die Leistung erhält, die ihrer Wichtigkeit entspricht.
- Arbeiten fast alle Tasks mit Priorität 255, genügt ein kleiner Prioritätsunterschied (z.B. 5), um eine Task deutlich zu bevorzugen. Bei einem größeren Prioritätsspektrum muss der Prioritätenunterschied entsprechend größer gewählt werden.
  Die Variation der Priorität sollte nach Last z.B. in 5er- oder 10er-Schritten vorgenommen werden, bis das gewünschte Performanceziel erreicht ist. In der Regel ist es unnötig, feste Prioritäten zu vergeben, da mit mittleren variablen Prioritäten (etwa 200) bereits gute Performancewerte erreicht werden können.
  Größere Prioritätsschritte sind bei einem permanent hohen Task-Aufkommen bzw. bei überwiegend Ein-/Ausgabe-intensiven Tasks notwendig.
- Die Zuteilung der Betriebsmittel Hauptspeicher und CPU an Benutzer- und Systemtasks wird entsprechend dem vorgegebenen Kategorien- und Prioritätenkonzept gesteuert.

Warteschlangen

Im System vorhandene Tasks können sich in fünf Zuständen befinden:

- die Task belegt eine CPU
- die Task ist aktiv und ablaufbereit
- die Task ist aktiv, aber nicht ablaufbereit
- die Task ist inaktiv und ablaufbereit
- die Task ist inaktiv und nicht ablaufbereit

Die notwendige Basis bietet das Warteschlangensystem der Task-Steuerung, das die Tasks in Abhängigkeit des jeweiligen Zustandes in eine der folgenden Warteschlangen einreiht.

Q0	Laufende Task	Task in Control
Q1	Warten auf Zuteilung einer CPU	a k t i v und ablaufbereit
Q2	Schreibtasks von openSM2	a k t i v und nicht ablaufbereit
Q3	Warten auf Seitenwechsel	a k t i v und nicht ablaufbereit
Q4	Warten bei schneller Ein-/Ausgabe, Explizite Synchronisationsfunktionen (VPASS, SOLSIG, REVNT, Locks und Auftragsbeziehungen im System), Warten auf neue Eingabe für TP-Tasks	a k t i v und nicht ablaufbereit
Q5	pro Kategorie: Warten auf Aktivierung	i n a k t i v und ablaufbereit
Q6	pro Kategorie: Warten auf Zulassung durch PCS (nur falls PCS eingeschaltet ist)	i n a k t i v und ablaufbereit
Q10	Warten im HOLD-Zustand	i n a k t i v und nicht ablaufbereit
Q11	Wartende Systemtasks	i n a k t i v und nicht ablaufbereit
Q12	Warten bei langdauernder Ein- und Ausgabe und langdauernden Synchronisationsfunktionen insbesondere Terminal-Ein-/Ausgabe im Dialog	i n a k t i v und nicht ablaufbereit
Q13	Warten n Sekunden (PASS/VPASS)	i n a k t i v und nicht ablaufbereit

Tabelle 31: Warteschlangen der Task-Steuerung

Wartet eine Task nicht auf eine Eingabe, so hat sie das primäre Ziel, die CPU zu belegen. Das setzt in der Regel mehrere Zustandswechsel voraus.

Eine Benutzertask (z.B. der Kategorie Batch) befindet sich z.B. nach einem HOLD-TASK-Kommando in der Q10. Sobald die Systembetreuung die Task mit dem Kommando RESUME-TASK wieder freigibt, verändert sie ihren Zustand.
Die Q10 wird verlassen und die Task wird in die Q5 (Subqueue für die jeweilige Kategorie, z.B. Batch) eingereiht. Dort wartet sie darauf, dass PRIOR über die Aktivierungsentscheidung auf diese Task zugreift und sie in die Q1 bringt und danach in die Q0.

Diese Zustandswechsel werden durch die PEND-/UNPEND-Routinen realisiert:
Die UNPEND-Routinen bringen die Task in Richtung CPU, die PEND-Routinen bringen die Task von einer hochwertigen Warteschlange in eine niederwertige Warteschlange.

11.2.2 Zuteilung von Ressourcen

Die Task-Steuerung muss eine Reihe von Entscheidungen treffen, wenn eine Task zum Ablauf kommen soll. Der Verwaltung der Betriebsmittel kommt dabei eine zentrale Bedeutung zu.

Erhält eine Task das Recht, Hauptspeicher zu belegen, dann heißt dies, dass die Task aktiviert wird.
Im Anschluss daran muss der Task eine CPU zugewiesen werden, d.h. die Task wird initiiert.

Kriterien für die Aktivierungsentscheidung sind:

- der Multiprogrammierungs-Grad pro Kategorie
- die Priorität
- Betriebsmittelauslastung (Hauptspeicher, CPU, Seitenwechsel)
- geleistete Systemdienste (CPU-Zeit, Anzahl der Ein- und Ausgaben)

Zeitscheiben

Die Zeitscheiben dienen der Task-Steuerung, um die Betriebsmittel CPU und Hauptspeicher allen Tasks möglichst gleichmäßig und gleichberechtigt zur Verfügung zu stellen:

- Mikrozeitscheibe (MTS) zur optimalen Nutzung der CPU
- Systemdienstzeitscheibe (SSTS) für sinnvolle Zuteilung des Hauptspeichers

Mikrozeitscheibe

Die Mikrozeitscheibe MTS ist die maximale Zeit, die eine Task das Betriebsmittel CPU ohne Unterbrechung beanspruchen kann, wenn keine Task mit höherer Priorität auf Zuteilung der CPU wartet.
Spätestens nach Ablauf der Mikrozeitscheibe erfolgt eine Unterbrechung, um die Blockierung der CPU durch eine Task zu verhindern. Besaß die Task eine variable Priorität (128-255), wird diese entsprechend des E/A-Verhaltens und der CPU-Nutzung der Task verändert.

Systemdienstzeitscheibe

Die Systemdienstzeitscheibe SSTS dokumentiert keine absolute Zeit, sondern entspricht einem Quantum an Systemdiensten (CPU-Nutzung, E/A-Rate), die eine aktive Task maximal beanspruchen darf. Nach Ablauf der Systemdienstzeitscheibe wird die Task deaktiviert, wenn Tasks der gleichen Kategorie auf Aktivierung warten.

Zuteilung des Betriebsmittels Hauptspeicher

- Aktivieren einer Task

  Die Aktivierung einer Task ist nur möglich, wenn die interne Kontrollinstanz, die ACF (Activate-Control-Function), dies erlaubt. D.h. ACF misst in periodischen Abständen die Auslastung der Betriebsmittel Hauptspeicher und CPU sowie die Seitenwechsel-Aktivitäten, bzw. wenn eine Folge von Aktivierungen durchgeführt werden soll, nur die Auslastung des Hauptspeichers.
  Je nach Betriebsmittelauslastung und definiertem Kategorien- und Prioritätenkonzept erfolgt dann anhand einer Matrix die Entscheidung, ob eine weitere Aktivierung zulässig ist oder ob eine Zwangsdeaktivierung, eine Verdrängung oder gar nichts erfolgen soll.

  Die Vorgehensweise des PRIOR bei der Aktivierung einer Task soll im Folgenden an einem Beispiel gezeigt werden:

  Insgesamt sind aktiv: 19 TP-, 9 DIALOG- und 5 BATCH-Tasks

  Auf Aktivierung warten: 2 TP-, 2 DIALOG- und 2 BATCH-Tasks

  Die Systembetreuung hat die Kategorien wie folgt definiert, der Operand IO-PRIORITY ist auf *NONE gesetzt (Voreinstellung):

```
/MODIFY-TASK-CATEGORIES CATEGORY-NAME=DIALOG,-
        WEIGHT-CODE=100,MINIMUM-ACTIVE-TASKS=25,MAXIMUM-ACTIVE-TASKS=35
/MODIFY-TASK-CATEGORIES CATEGORY-NAME=TP,-
        WEIGHT-CODE=500,MINIMUM-ACTIVE-TASKS=30,MAXIMUM-ACTIVE-TASKS=40
/MODIFY-TASK-CATEGORIES CATEGORY-NAME=BATCH,-
        WEIGHT-CODE=3,MINIMUM-ACTIVE-TASKS=5,MAXIMUM-ACTIVE-TASKS=15
```

  Die Auswahl-/Aktivierungsentscheidung trifft die Task-Steuerung in zwei Stufen.

  1. Auswahl der Kategorie, aus der eine Task aktiviert werden soll.
  2. Innerhalb dieser Kategorie wird die Task ausgewählt, die die höchste Priorität hat bzw. bei gleicher Priorität wird die Task aktiviert, die in der Warteschlange an erster Stelle steht.

Für die Auswahl der Kategorie sind folgende Entscheidungen notwendig	**Bezogen auf das Beispiel bedeutet dies**
Es werden nur die Kategorien berücksichtigt, für die **mindestens eine Task auf Aktivierung wartet**	die Bedingung trifft für alle Kategorien zu
Bevorzugt ausgewählt werden unter diesen Kategorien ○ alle Kategorien (sofern vorhanden), die MINMPL noch nicht erreicht haben. MAXMPL ist hier ohne Bedeutung. ○ wenn alle Kategorien MINMPL erreicht haben: die Kategorien, die MAXMPL noch nicht erreicht haben.	die Bedingung trifft für die Kategorien TP und DIALOG zu
Sind solche bevorzugten Kategorien vorhanden, werden alle anderen Kategorien von der weiteren Auswahl ausgeschlossen: ○ z=0 (Anzahl aktiver Tasks < MIN) ○ z=MIN (0MIN <= Anzahl aktiver Tasks < MAX) ○ z=MAX (Anzahl aktiver Tasks > MAX)	Begründung: z=0 für TP (19 < 30) z=0 für DIALOG (9 < 25) z=5 für BATCH (MIN=5)
Von diesen Kategorien wird nun die Kategorie für die Aktivierung ausgewählt, für die der niedrigste Index ermittelt wird. Die Indexberechnung erfolgt nach dem Algorithmus: *Index* = (*NAT* +1 -*z*) / *WT* Dabei bedeutet: ○ NAT = Anzahl der aktiven Task der Kategorie, für die der Indexberechnet wird ○ z = Zustandsanzeige ○ WT = Wichtigkeit (Gewicht)	Für die Kategorie TP und DIALOG muss die Indexberechnung erfolgen: $Index_{TP}$ = (19+1-0) / 500 = 20 / 500 = 0,04 $Index_{DIA}$ = (9+1-0) / 100 = 10 / 100 = 0,01
Bei Index-Gleichheit wird in der Reihenfolge TP - DIALOG - BATCH aktiviert.	

Ergebnis: Es wird die höchstpriore Task der Kategorie TP aktiviert.

Ausnahme: Tasks mit fester Priorität werden bevorzugt aktiviert.

- Deaktivieren einer Task

  Das Deaktivieren von aktiven Tasks verfolgt den Zweck, ablaufbereiten inaktiven Tasks das Betriebsmittel Hauptspeicher zur Verfügung zu stellen.
  Es geschieht, wenn für die aktiven Tasks

- eine weitere Verarbeitung nicht möglich ist
  (z.B. durch programmgesteuerte Warteaufrufe, Warten auf Eingabe von Datensichtstation im Teilnehmerbetrieb)
- die Zeit beim Warten auf Ereignisse überschritten wird
  (z.B. Warten auf Eingabe von Datensichtstation im Teilhaberbetrieb)
- ein bestimmter Umfang von System-Diensten
  (z.B. CPU-Zeit und Anzahl der Ein-/ Ausgaben) in Anspruch genommen wurden.

Ausnahme

Die Deaktivierung unterbleibt, wenn die Systembetreuung im Benutzerkatalog für den Benutzer vereinbart hat, dass seine Tasks nicht deaktiviert werden dürfen (siehe Kommando ADD-USER, Operand PRIVILEGE=INHIBIT-DEACTIVATION).

- Zwangsdeaktivieren einer Task

  Wenn die ACF-Funktion im Rahmen ihrer Messungen zur Auslastung der Betriebsmittel Engpässe feststellt, kommt es zur Zwangsdeaktivierung einer Task.
  Da die Kategorie-Merkmale MIN, MAX und WT für die Zwangsdeaktivierung herangezogen werden, kann die Systembetreuung mit einer entsprechenden Voreinstellung dieser Werte dafür sorgen, dass nur Tasks weniger wichtiger Kategorien zwangsdeaktiviert werden.

  Ausnahmen

  - Tasks mit fester Priorität
  - Tasks im Funktionszustand TPR
  - Tasks, die einen Lock halten
  - Tasks, für deren Benutzerkennung PRIVILEGE=INHIBIT-DEACTIVATION vereinbart wurde (Kommando ADD-USER), können nicht zwangsdeaktiviert werden.

- Verdrängen einer Task

  Zu einer Verdrängung kann es kommen, wenn eine Aktivierungsanforderung vorliegt und die ACF-Funktion auf Grund festgestellter Betriebsmittelengpässe keine Aktivierung zulässt. Es gibt zwei Möglichkeiten der Verdrängung:

  1. Eine aktive Task wird von einer inaktiven Task einer anderen Kategorie verdrängt.
  2. Die aktive und inaktive Task gehören der gleichen Kategorie an.

  Die Verdrängungsrate ist umso höher, je langfristiger der Überlastzustand (= Engpass an Betriebsmitteln) im System ist.

- Preemption-Control-Function (PCF)

  PCF überwacht periodisch die Verdrängungsrate.
  Meldung: `EXC0455 TASK PREEMPTION LEVEL=i`

  `i` ist eine Ziffer von 0 bis 3 und bezeichnet den Grad der Verdrängungsrate.

Grad	Auswirkungen im System	Maßnahme des Operators
0	Normaler Systemablauf	Keine
1, 2	Kurzzeitige Überlastzustände	Keine
3	Langfristige Überlastzustände Der MAX-MPL-Wert einer Katagorie wird nicht überschritten. Das System hält Hauptspeicher für Privilegierungen frei.	Reduzierung der MIN-MPL-Werte von Katagorien

  Maßnahmen zur Verhinderung zu hoher Verdrängungsraten

  - Ändern Sie die MIN-MPL- und MAX-MPL-Werte für einzelne Katagorien so ab, dass weniger Aktivierungsanforderungen entstehen.
  - Vergrößern Sie den Hauptspeicher oder reduzieren Sie die Last.

Zuteilung des Betriebsmittels CPU

Jede Task wartet nach erfolgter Aktivierung auf Zuteilung des Betriebsmittels CPU (Zentralprozessor), d.h. auf **Initiierung**.

Hat die Task das Betriebsmittel CPU erhalten, kann es auf Grund folgender Unterbrechungsursachen zu einer **Deinitiierung** kommen:

- Task will auf Synchronisationsereignis warten
- Warten auf Beendigung einer Ein-/Ausgabe-Operation
- Seitenwechsel vornehmen
- Ablauf der Mikrozeitscheibe

Auch auf Initiierungsebene kann es zu einer **Verdrängung** (Preemption) kommen.
Dies erfolgt dann, wenn eine andere aktive und ablaufbereite Task mit höherer Priorität auf Initiierung wartet. Das ist z.B. der Fall, wenn für eine Task mit höherer Priorität eine
Ein-/ Ausgabe-Operation abgeschlossen ist.

Zuteilung von I/O-Ressourcen

Die Zuteilung von I/O-Ressourcen ist abhängig von der I/O-Priorität des Tasks. Die Steuerung übernimmt IORM, siehe Abschnitt „IORM: Steuerung von I/O-Ressourcen" und Handbuch „Dienstprogramme“ [15].

11.2.3 TANGRAM: Verwaltung von affinen Task-Gruppen

Das Subsystem TANGRAM (Task and Group Affinity Management) plant für affine Task-Gruppen unter Berücksichtigung ihres Leistungsbedarfs die Zuordnung zu CPUs. Bei bestimmten Multitask-Anwendungen gibt es Gruppen von Tasks, die häufig auf größere Mengen gemeinsamer Daten schreibend zugreifen. Diese Task-Gruppen werden als „affine Task-Gruppen“ bezeichnet.
Um auf Multiprozessoren die Verluste durch Cohits in den CPU-Caches zu reduzieren, ist es günstig, solche Task-Gruppen auf eine Teilmenge der CPUs einzuschränken.

Das Einrichten der Task-Gruppen erfolgt mit dem Makro TINF (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30]). Die Verwaltung der Taskgruppen übernimmt das bei „System Ready“ automatisch gestartete Subsystem TANGBAS und zwar unabhängig davon, ob die Regelungsfunktion TANGRAM gestartet ist.

Das Subsystem TANGRAM misst die Auslastung der einzelnen CPUs, die Leistungsaufnahme der angemeldeten Task-Gruppen sowie der restlichen Tasks und legt jeweils für ein Regelintervall die für die Initiierung einer Task erlaubten CPUs fest.

Die An- und Abmeldung von Tasks zu Gruppen sowie die Prozessorzuordnung der einzelnen Gruppen kann mit COSMOS-Hooks protokolliert werden. COSMOS ist ein optionaler Bestandteil von openSM2 (siehe Messprogramme im Handbuch „openSM2“ [49]).

Der durch den Einsatz von TANGRAM erzielbare Performance-Gewinn ist abhängig vom Server (Multiprozessorgrad, Cache-Architektur), vom Lastprofil (Anteil und Strukturierung der TP-Anwendungen) und von der Server-Auslastung. Die BS2000-TP-Produkte openUTM, UDS/SQL und SESAM/SQL richten TANGRAM-Task-Gruppen ein. Multitask-Anwendungen sollten daraufhin überprüft werden, ob das Einrichten von TANGRAM-Task-Gruppen einen Performance-Gewinn bringt.

Die Arbeitsweise von TANGRAM kann durch die Angabe von Parametern in der Parameterdatei SYSSSI.TANGRAM.<version> gesteuert werden.
Die angegebenen Voreinstellungen treten in Kraft, wenn einer der Parameter nicht in der Parameterdatei gefunden wird oder keine Parameterdatei zur Verfügung steht. Im zweiten Fall wird außerdem folgende Konsolmeldung ausgegeben:

```
NTG0002 Warning: Opening of TANGRAM parameter file (&00) failed.
        DMS-Error-Code (&01). Processing continued with default parameters.
```

Um Parameter zu verändern, muss TANGRAM angehalten und mit der geänderten Parameterdatei neu gestartet werden. Bei diesem Vorgang bleiben alle eingerichteten Task-Gruppen erhalten.

Die folgenden Parameter steuern die Arbeit von TANGRAM. Die Reihenfolge der Parameter in der Parameterdatei ist beliebig, es muss jedoch jeder Parameter in einer eigenen Zeile stehen.

PERIOD=<integer 1..100>
Bestimmt den Abstand zwischen zwei aufeinander folgenden Läufen der periodischen TANGRAM-Routinen in Sekunden.
Voreinstellung: 10 Sekunden

CLEARENCE=<integer 0..100>
Gibt einen Wert zur Berechnung der max. CPU-Auslastung einer Task-Gruppe in Prozent an. Die Auslastung wird folgendermaßen berechnet: Auslastung = (1-wert).
Voreinstellung: 20 %, d.h. jede CPU sollte maximal zu 80 % von einer einzelnen Task-Gruppe belegt sein.

THRESHOLD=<integer 0..100>
Bestimmt die Schwelle, ab welcher Gesamtauslastung auf allen CPUs eine Task-Gruppe aktiv am Verfahren der Prozessor-Zuordnung durch TANGRAM teilnimmt.
Voreinstellung: 10 %

11.3 Zeitlimitierungen in BS2000

In BS2000 existieren drei verschiedene Zeitlimitierungen, die den Benutzer im Teilnehmerbetrieb betreffen: Eine benutzerkennungs-spezifische, eine task-spezifische und eine programmlauf-spezifische Zeitbegrenzung.

Eine benutzerkennungs-spezifische Zeitbegrenzung legt die Systembetreuung im Benutzereintrag fest. Von diesem festgelegten Zeitkontingent rechnet das Abrechnungssystem von BS2000 bei Task-Beendigung die verbrauchte CPU-Zeit ab.

Für eine Task kann beim Starten die maximale CPU-Zeit (Task-Time-Limit: TTL) angegeben werden.

- Für Batch-Jobs gilt:
  Die maximal angebbare CPU-Zeit ist das Minimum aus dem Zeitkontingent des Benutzereintrags und dem Maximalwert aus der Job-Klassen-Definition.
- Für Dialog-Jobs gilt:
  Die maximale CPU-Zeit wird mit dem Maximalwert aus der Job-Klassen-Definition vorgegeben, sofern der CPU-Zeit-Wert im Benutzereintrag noch größer Null ist.

Ohne Angabe einer maximalen CPU-Zeit wird der Standardwert aus der Job-Klassen-Definition oder gegebenenfalls das kleinere Zeitkontingent des Benutzereintrags als maximale CPU-Zeit der Task übernommen. Benutzer, die im Benutzereintrag das Privileg No-Time-Limit besitzen, und die Benutzerkennung TSOS können Tasks ohne Zeitbegrenzung starten (NTL-Tasks).

Da das Zeitkontingent im Benutzereintrag erst bei Beendigung der Task aktualisiert wird, kann die maximale CPU-Zeit (und ggf. die tatsächlich verbrauchte) mehrerer Tasks einer Benutzerkennung zusammen das Zeitkontingent überschreiten.

Bei dem Laden oder Starten eines Programms kann für den jeweiligen Programmlauf eine maximale Programmlaufzeit angegeben werden (Program-Time-Limit: PTL).

Bezüglich der maximalen CPU-Zeit der Task (TTL) und der maximalen Programmlaufzeit (PTL) verhalten sich Dialog- und Batch-Jobs unterschiedlich:

TTL bzw. PTL im Batch-Job erreicht

Hat der Batch-Job die maximale CPU-Zeit verbraucht und ist kein Programm geladen und war TTL < 300 Sekunden, so wird die Task mit folgender Meldung an der Konsole beendet:

```
EXC0736 ABNORMAL TASK TERMINATION.
        ERROR CODE '(CMD1011)': /HELP-MSG (CMD1011)
```

Hat der Batch-Job die maximale CPU-Zeit verbraucht und ist ein Programm geladen oder ist TTL >= 300 Sekunden, wird folgende Meldung an der Konsole ausgegeben:

```
EXC0070 Batch job with TSN (&00) has reached time runout.
```

Wenn der Systemparameter DIATTL=Q gesetzt ist (siehe "TTL bzw. PTL im Dialog-Job erreicht"), dann wird der Batch-Job beendet.

Sonst wird der Job jetzt für 20 min. angehalten, falls ein Programm geladen ist. Falls kein Programm geladen ist, wird das TTL des Jobs um 5 Minuten erhöht. Während dieser Wartezeit (5 CPU- bzw. 20 Zeit-Minuten) kann die Systembetreuung mit dem Kommando CHANGE-TASK-CPU-LIMIT die maximale CPU-Zeit des Batch-Jobs erhöhen und verhindern, dass wichtige Batch-Jobs wegen Zeitüberschreitung vorzeitig beendet werden.

Wird die maximale CPU-Zeit des Batch-Jobs innerhalb der Wartezeit von 5 bzw.
20 Minuten nicht erhöht, wird der Batch-Job (endgültig) beendet. Dabei sind folgende Fälle zu unterscheiden:

1. Wurde das TTL bzw. PTL während eines Programmlaufes erreicht, wird die Meldung EXC0072 ausgegeben oder ggf. eine STXIT-Routine abgearbeitet, die maximal weitere 30 CPU-Sekunden verbrauchen darf.
   Nach Abarbeitung der STXIT-Routine bzw. spätestens nach 30 CPU-Sekunden wird das Programm mit der Meldung EXC0073 beendet.

   Anschließend wird die Task beendet (TTL) oder Spin-Off ausgelöst (PTL).
   Ist eine Fehlerbehandlung vorgesehen, beginnt die weitere Bearbeitung bei dem nächsten SET-JOB-STEP- bzw. IF-BLOCK-ERROR-Kommando. Anderenfalls endet die Bearbeitung beim nächsten EXIT-JOB- bzw. LOGOFF-Kommando (PTL).

   ```
   EXC0072 TIME LIMIT FOR TASK/PROGRAM EXCEEDED. PROGRAM TERMINATED
   EXC0073 MAXIMUM PROGRAM RUNTIME EXCEEDED. 'STXIT' ROUTINE OR
           PROGRAM RUN USED UP 30 SECONDS.
   ```

2. Wurde das TTL während der Abarbeitung von BS2000-Kommandos erreicht, wird die Task mit folgender Meldung beendet:

   ```
   EXC0736 ABNORMAL TASK TERMINATION.
           ERROR CODE '(CMD1011)': /HELP-MSG (CMD1011)
   ```

TTL bzw. PTL im Dialog-Job erreicht

Im Normalfall ist die CPU-Zeit einer Dialogtask nicht begrenzt. Die Systembetreuung kann jedoch mit dem Systemparameter DIATTL (DIAlog Task Time Limit) die CPU-Zeit in Abhängigkeit vom TTL begrenzen. DIATTL kann die Werte Y, N und Q annehmen:

1. DIATTL=N (Standard)
   Die Dialogtask wird mit Task-Time-Runout nicht beendet.

   Bei Überschreiten des TTL wird, wenn kein Programm geladen ist, die Meldung EXC0067 ausgegeben und der Benutzer kann 30 CPU-Sekunden weiterarbeiten, bis sich der Vorgang wiederholt.

   ```
   EXC0067 CPU TIME SPECIFIED IN /SET-LOGON-PARAMETERS EXCEEDED.
           TASK CONTINUED
   ```

   Wenn ein Programm geladen ist, gilt bei Überschreiten des TTL bzw. PTL:

   - Im Prozedur-Modus wird die Meldung EXC0068 ausgegeben und die Ausführung fortgesetzt. Die Zeitbegrenzung wird um 30 CPU-Sekunden hochgesetzt, d.h. danach wiederholt sich der Vorgang.

     ```
     EXC0068 CPU TIME SPECIFIED AT PROGRAM START EXCEEDED. PROGRAM CONTINUED
     ```

   - Im interaktiven Dialog wird für ein Programm ohne definierte STXIT-Routine für die Ereignisklasse „Ende der Programmlaufzeit“ die Meldung EXC0075 ausgegeben. Der Benutzer kann angeben, ob er die Ausführung fortsetzen oder beenden will. Wird die Ausführung fortgesetzt, wiederholt sich der Vorgang nach 30 CPU-Sekunden.

     ```
     EXC0075 TIME LIMIT FOR PROGRAM RUN EXCEEDED. PROGRAM TO BE CONTINUED?
             REPLY (Y=YES; N=NO)
     ```

- In einem Programm mit einer solchen STXIT-Routine wird diese gestartet.
  Zur Abarbeitung der Routine werden der Task weitere 30 CPU-Sekunden zugestanden. Beendet die STXIT-Routine das Programm nicht, gibt sie die Kontrolle wieder an das Programm zurück. Sind die zugestandenen 30 CPU-Sekunden noch nicht aufgebraucht, läuft das Programm bis zur Überschreitung dieses neuen Zeitlimits weiter. Danach werden folgende zwei Meldungen ausgegeben:

  ```
  EXC0073 MAXIMUM PROGRAM RUNTIME EXCEEDED. 'STXIT' ROUTINE OR
          PROGRAM RUN USED UP 30 SECONDS.
  EXC0075 TIME LIMIT FOR PROGRAM RUN EXCEEDED. PROGRAM TO BE CONTINUED?
          REPLY (Y=YES; N=NO)
  ```

  Der Benutzer kann angeben, ob er die Ausführung fortsetzen oder beenden will. Wird die Ausführung fortgesetzt, wiederholt sich der Vorgang nach 30 CPU-Sekunden.

2. DIATTL=Y
   Die Dialogtask wird mit Task-Time-Runout beendet.

   Bei Überschreiten des TTL wird zunächst die Meldung EXC0076 ausgegeben:

   ```
   EXC0076 TASK CPU TIME LIMIT for DIALOG TASK exceeded.
           Task will be terminated after 30 sec extra CPU-time.
   ```

   Falls ein Programmlauf mit STXIT für Timeout vorliegt, wird diese STXIT-Routine noch nach Ablauf der mit der Meldung eingeräumten Zeit gestartet. Der STXIT-Routine werden standardmäßig noch einmal 30 CPU-Sekunden zugestanden.

3. DIATTL=Q (quick and quiet)
   Die Dialogtask wird mit Task-Time-Runout beendet (wie bei DIATTL=Y).

   Zusätzlich werden (auch für einen Batch-Job) sowohl für TTL wie auch für PTL alle zuvor erwähnten Zeitzuschläge bzw. Wartezeiten auf eine Sekunde verkürzt („quick“) sowie die auf eine Antwort wartende Meldung EXC0070 an der Konsole unterdrückt („quiet“).

11.4 PCS: Performance-Überwachungssystem

Das optionale Subsystem PCS (Performance Control System) unterstützt die Systembetreuung bei der optimalen Einstellung der Systeme. Es ermöglicht die Aufteilung der Leistung eines Systems auf die einzelnen Task-Kategorien und Tasks entsprechend den Erfordernissen der Benutzer.

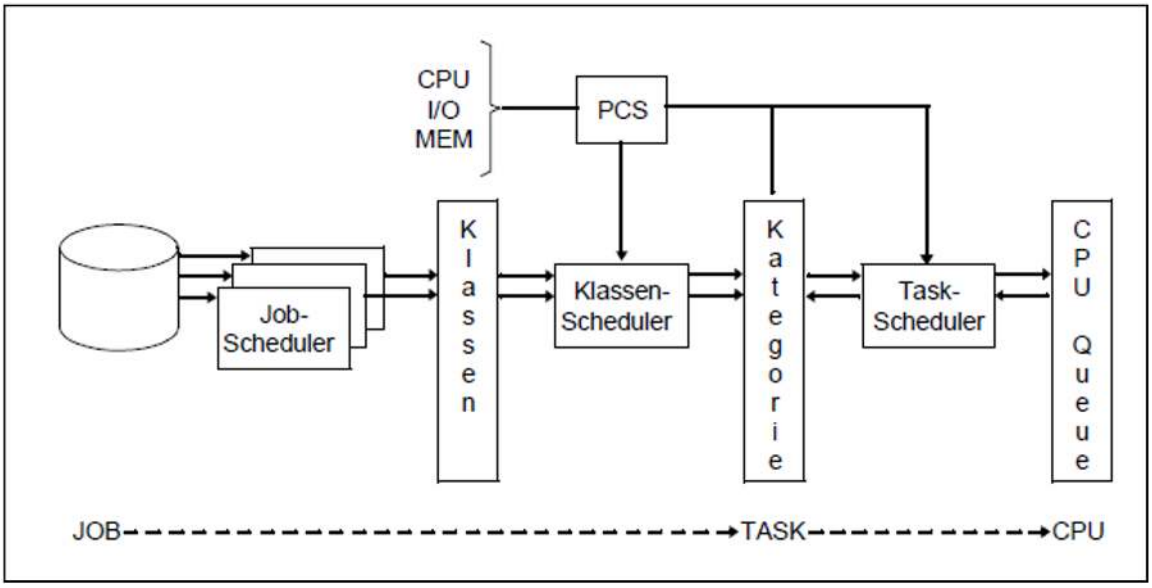


Bild 18: Job-Steuerung – PCS – Task-Steuerung

PCS regelt bei komplexem Mischbetrieb die verschiedenen Lastanteile entsprechend den kundenspezifischen Anforderungen. PCS passt in kurzen Abständen die Stellgrößen so an, dass vorgegebene Optimierungsstrategien (Antwortzeit- bzw. Durchsatz-Optimierung) und der hierfür optimale Betriebspunkt jederzeit gehalten werden.

PCS überwacht während des Betriebes die Leistungsaufnahme der Tasks und Kategorien und die Auslastung von CPU, Hauptspeicher und Ein-/Ausgabe-System.

Wenn diese Größen nicht den von der Systembetreuung vorgegebenen Führungsgrößen (PCS-Option) entsprechen, korrigiert PCS die Leistungsaufnahme durch Veränderung BS2000-interner Stellgrößen (z.B. Task-Priorität, MIN-MPL- und MAX-MPL-Werte der Katagorien).

Wenn PCS steuernd eingreift, dann gibt es in folgenden Fällen Meldungen an die Konsole aus. Die Meldungen werden auch in der CONSLOG-Datei protokolliert.

- Wenn der Job-Scheduler veranlasst wird, systemglobal oder in einer Kategorie keine Jobs mehr zu starten.
- Wenn die Task-Aktivierung bei Erreichen des MAX-MPL-Wertes eingestellt wird.

PCS ermittelt seine Regelgrößen über Zeiträume von etwa 10 Sekunden und passt die Stellgrößen der momentanen Lastsituation in diesem Zeitraum an.

PCS arbeitet mit PRIOR, dessen Steuerung der Task-Prioritäten beim Start von PCS abgeschaltet wird, und mit dem Job-Management zusammen.

Nährere Informationen zu PCS finden Sie im Handbuch „PCS“ [36].

12 Sicherheit

Dieses Kapitel beschreibt die wesentlichen Sicherheitsmechanismen von BS2000.

Dazu gehören die Zugangs- und Zugriffsschutzmechanismen von BS2000 sowie die mit dem Softwareprodukt SECOS gebotenen zusätzlichen Mechanismen. Informationen zu SECOS finden Sie im Abschnitt „Erfüllung von Sicherheitsanforderungendurch SECOS“ und in den SECOS-Handbüchern [46] und [47].

12.1 Zugangsschutz

Zugangsschutz beinhaltet alle Methoden zum Schutz eines DV-Systems vor unberechtigtem Systemzugang.

BS2000

Der Zugangsschutz mit LOGON-Kennwort im Kommando SET-LOGON-PARAMETERS ist das derzeit gebräuchlichste Authentisierungsverfahren.
Dabei wird mit dem Kommando MODIFY-USER-PROTECTION ein 8 bzw. 32 Byte langes Kennwort für die Benutzerkennung vereinbart.

Durch organisatorische Maßnahmen, die SECOS anbietet, kann der Kennwortschutz in BS2000 erheblich verbessert werden.
Dabei legt die Benutzerverwaltung für den Benutzer explizit fest, welche der folgenden Vorgaben er zwingend beachten muss:

- Minimale Länge eines Kennwortes
- Komplexität eines Kennwortes
- Lebensdauer eines Kennwortes

Erweiterter Zugangsschutz mit SECOS

SECOS bietet folgende zusätzliche Möglichkeiten für den Zugangsschutz:

- Trennung der Zugangswege
- Einschränkung des Zugangs über Terminal-Sets
- Zugangsschutz mit Guards
- Persönliche Identifizierung
- Protokollierung der Zugangsversuche
- Sperre von Terminals oder Benutzerkennungen nach einer festgelegten Anzahl von erfolglosen Zugangsversuchen
- Sperre von Benutzerkennungen bei Inaktivität

12.2 Zugriffsschutz

Zugriffsschutz bezeichnet die Regeln, nach denen in einem DV-System Subjekte auf Objekte zugreifen können und die Methoden, mit denen die Einhaltung dieser Regeln sichergestellt werden kann.

BS2000

BS2000 stellt Zugriffsschutzmechanismen für Objekte, die vom DVS (Dateiverwaltungssystem) verwaltet werden können, zur Verfügung. Das DVS stellt die Objektverwaltung für Dateien dar, die die Objekte des DVS sind.

Für Dateien gibt es folgende Schutzmechanismen:

- Begrenzter Pubset-Zugriff
  Durch Verteilung von Benutzerkennungen auf unterschiedliche Pubsets kann ein Schutz vor unerlaubtem Zugriff auf Objekten, z.B. Dateien, jeweils anderer Pubsets erreicht werden (siehe auch Kapitel „BS2000-Benutzerverwaltung" und Kapitel „Pubset-Verwaltung").
- Standard-Zugriffskontrolle (USER-ACCESS/ACCESS)
  Mit dem Zugriffsschutzmechanismus der Standard-Zugriffskontrolle (mit den Operanden ACCESS und USER-ACCESS der Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES) legt der Benutzer Zugriffsrechte für sich und andere fest (siehe "Standard-Zugriffskontrolle (ACCESS/USER-ACCESS)").
- Einfache Zugriffskontroll-Liste (Basic Access Control List, BACL)
  Mit dem Zugriffsschutzmechanismus der BACL legt der Benutzer Zugriffsrechte auf Objekte, z.B. Dateien, für eine differenzierte Menge von Subjekten fest (siehe "Einfache Zugriffskontroll-Liste (BACL)"). Die Zugriffsrechte Lesen (read), Schreiben (write) und Ausführen (exec) können für jede der Benutzerklassen Owner, Group und Others getrennt vergeben werden.
- Kennwort
  Für jede seiner Dateien kann der Benutzer Kennwörter (Lese-, Schreib- und Ausführungs-Kennwort) vereinbaren. Vor der Verarbeitung kennwortgeschützter Dateien ist das entsprechende Kennwort anzugeben. Kennwörter können auf Wunsch verschlüsselt werden (siehe "Kennwörter und Schutzfristen").
- Dateiverschlüsselung
  Der Benutzer kann eine Datei in eine verschlüsselte Form umwandeln und dabei ein Crypto-Kennwort (wie ein „normales“ Kennwort, jedoch 8 Zeichen lang) vorgeben. Der Zugang zum Dateiinhalt ist dann nur noch nach Vorgabe des richtigen Crypto-Kennwort möglich (siehe "Dateiverschlüsselung").
  Verschlüsselte Dateien haben kein Kennwort für Lesen oder Ausführen, sind aber sonst mit allen anderen Zugriffsschutzmechanismen kombinierbar.

- Schutzfrist
  Ein Benutzer kann seiner Datei eine Schutzfrist zuordnen, innerhalb der diese Datei nicht geändert werden darf (siehe "Kennwörter und Schutzfristen").

Sicherheit mit Kryptografie

Das BS2000-Subsystem CRYPT bietet kryptografische Funktionen für BS2000-Nutzer an. Es verwendet standardisierte, offene Schnittstellen zur Ver- und Entschlüsselung von Daten, siehe Handbuch „CRYPT“ [9]. CRYPT benötigt keine spezielle Hardware.

Erweiterter Zugriffsschutz mit SECOS

Das Softwareprodukt SECOS (Komponente GUARDS) stellt ein komfortables und flexibles Verwaltungsinstrument zur Verfügung, mit dem für Objekte wie Dateien, Bibliotheken, Bibliothekselemente oder weitere Objekte verschiedener Objektverwaltungen Zugriffsbedingungen definiert und bewertet werden können. Der Zugriffsschutz für die genannten Objekte wird dabei über so genannte Guards realisiert, die Zugriffsbedingungen aufnehmen. Zugriffsbedingungen werden von Objektverwaltungen (DVS, LMS, JVS, HSMS, SRPM, usw.) abgefragt. Je nach Ergebnis der Auswertung gestattet die Objektverwaltung dann dem Subjekt den Zugriff oder nicht.

Verwendung der Schutzmechanismen

Die folgende Tabelle zeigt, welche Objekttypen mit welchem Schutzmechanismus geschützt werden können:

Objekt / Schutzmechanismus		Begrenzter Pubset-Zugriff	ACCESS USER-ACCESS	BACL	Kennwort	Dateiver-schlüsselung	Schutzf
Datei[1]	**public**	+ [2]	+	+	+	+	+
	temporär	- [3]	-	-	-	+	-
	privat	-	+	+	+	-	+
	Band	-	+	-	+	-	+
Dateigenerations-gruppe	**Index public, FGen public**	-	+	+	+	+	+
	Index public, FGen Band	-	+	+	+	-	+
	Index privat, FGen privat	-	+	+	+	-	+
Jobvariable	**permanent**	+	+	+	+	-	+
	temporär	-	-	-	-	-	-
Bibliotheks-Element[4]		-	-	+	-	-	-
FITC-Port		-	-	-	-	-	-
Storage-Klassen		-	-	-	-	-	-
HSMS-Managementklassen		-	-	-	-	-	-

Tabelle 32: Schutzmechanismen für Objekte

[1] Ist Datei eine Bibliothek, siehe „Besonderheiten bei Bibliothekszugriffen"

[2] + bedeutet: Schutzmechanismus anwendbar

[3] - bedeutet: Schutzmechanismus nicht anwendbar

[4] siehe „Besonderheiten bei Bibliothekszugriffen"

Wie die Tabelle zeigt, können manche Objekte mit mehreren Schutzmechanismen geschützt werden. Von den Schutzmechanismen ACCESS/USER-ACCESS, BACL und GUARDS kann jeweils nur einer für ein Objekt wirksam sein (siehe nächsten Abschnitt). Alle übrigen Schutzmechanismen gelten zusätzlich.

Hierarchie von Datei-Schutzmechanismen

Beim gleichzeitigen Einsatz der Schutzmechanismen ACCESS/USER-ACCESS, BACL und GUARDS für dasselbe Objekt könnten widersprüchliche Situationen entstehen. Um diese zu vermeiden, gilt folgende Hierarchie:

- Wenn der Schutz eines Objektes über Guards definiert ist, dann gelten ausschließlich die in den Guards festgelegten Zugriffsbedingungen. Eine ggf. für das Objekt definierte BACL und die Schutzattribute ACCESS /USER-ACCESS bleiben unberücksichtigt.
- Wenn kein Guardschutz für ein Objekt definiert ist, aber eine BACL, dann gilt die in der BACL festgelegte Schutzeinstellung. Die Schutzattribute ACCESS und USER-ACCESS bleiben unberücksichtigt.
- Wenn der Schutz eines Objektes weder mit Guards noch mit BACL geregelt ist, dann werden für den Schutzmechanismus die Schutzattribute ACCESS und USER-ACCESS herangezogen.

In allen Fällen gelten zusätzlich der begrenzte Pubset-Zugriff, der Kennwortschutz, die Dateiverschlüsselung und die Schutzfrist.

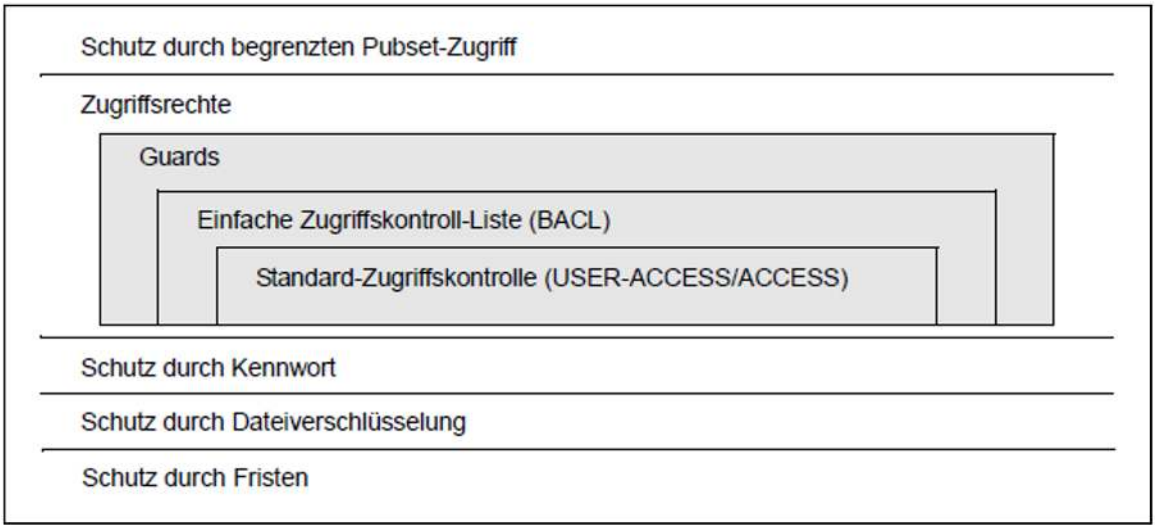


Bild 19: Schutzmechanismen für Dateien auf einem Pubset

Jede Datei und jede Jobvariable kann durch einen oder mehrere Schutzmechanismen gesichert werden. Die Zugriffsrechte sind mit folgender Priorität wirksam:

1. Guards
2. BACL
3. Standard-Zugriffskontrolle

Von den drei Schutzmechanismen wird genau einer bei jeder Überprüfung einer Zugriffsberechtigung herangezogen. Das ist derjenige, der in der Hierarchie am höchsten steht **und** der aktiviert ist.

Besonderheiten bei Bibliothekszugriffen

PLAM-Bibliotheks**dateien** können als Ganzes wie eine Datei geschützt werden. Davon unabhängig können Bibliotheks**elemente** mit der LMS-Anweisung MODIFY-ELEMENT-PROTECTION geschützt werden.

Beim Zugriff auf Bibliotheken und Bibliothekselemente ist daher Folgendes zu beachten:

- Der Zugriff auf einzelne Bibliothekselemente wird durch die mit MODIFY-ELEMENT-PROTECTION festgelegten Schutzmechanismen geregelt. Unabhängig vom Elementschutz ist dieser Zugriff jedoch nur möglich, wenn die Bibliotheksdatei in ihrer Gesamtheit gelesen werden darf.
- Beim Zugriff auf eine Bibliothek in ihrer Gesamtheit (mit ARCHIVE, mit File-Transfer oder mit dem DVS-Kommando COPY-FILE) gilt Folgendes:
  - Ist die Bibliothek weder durch eine BACL noch durch Guards geschützt, kann auf sie zugegriffen werden wie auf eine beliebige Datei.
  - Für den Zugriff auf eine Bibliothek, die durch eine BACL oder ein Guard geschützt ist, können die Zugriffsregelungen folgender Tabelle entnommen werden:

		Bibliothek enthält mindestens ein Element, das mit einer BACL oder einem Guard geschützt ist	Bibliothek enthält **kein** Element, das mit einer BACL oder einem Guard geschützt ist
Zugriff durch	**Eigentümer**	* [1]	*
	Miteigentümer	*	*
	Andere	Zugriff verboten	*

Tabelle 33: Zugriffsregelungen beim Zugriff auf Bibliotheken

[1] * bedeutet: Zugriff hängt von den Zugriffsregelungen der gesamten Bibliothek ab

12.3 Zugriffsschutz in BS2000

Kommando	Bedeutung
ADD-CRYPTO-PASSWORD	Crypto-Kennwort zur Entschlüsselung von verschlüsselten Dateiinhalten wird in der Kennwort-Tabelle der Task hinterlegt
ADD-PASSWORD	Tasklokale Eingabe eines Kennworts in die Kennwort-Tabelle, um Zugriff zu kennwortgeschützten Dateien zu ermöglichen
CREATE-FILE	Name und Merkmale für eine neue Datei vereinbaren
CREATE-FILE-GROUP	Name und Merkmale für eine neue Dateigenerationsgruppe vereinbaren
CREATE-JV	Neue Jobvariable erzeugen
DECRYPT-FILE	Verschlüsselte Datei in unverschlüsselte Datei umwandeln
ENCRYPT-FILE	Unverschlüsselte Datei in verschlüsselte Datei umwandeln und Crypto-Kennwort vereinbaren
MODIFY-FILE-ATTRIBUTES	Festlegen der Schutzmerkmale einer Datei
MODIFY-FILE-GROUP-ATTRIBUTES	Festlegen der Schutzmerkmale einer Dateigenerationsgruppe
MODIFY-JV-ATTRIBUTES	Festlegen der Schutzmerkmale einer Jobvariablen
REMOVE-CRYPTO-PASSWORD	Crypto-Kennwort wird aus der Kennwort-Tabelle der laufenden Task gelöscht
REMOVE-PASSWORD	Kennwort aus der Kennwort-Tabelle löschen
SHOW-FILE-ATTRIBUTES	Information über die Schutzmerkmale einer Datei
SHOW-JV-ATTRIBUTES	Information über die Schutzmerkmale einer Jobvariablen

Tabelle 34: Kommandoübersicht für die Zugriffskontrolle in BS2000

12.3.1 Kennwörter und Schutzfristen

Bei der Erzeugung oder Bearbeitung einer Datei mit dem Kommando CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES können für jede Zugriffsart (READ, WRITE, EXEC) Kennwörter vergeben werden (Operand `PROTECTION=*PAR(<zugriffsart>-PASSWORD=...)`).

`READ-PASSWORD=`	Anzugeben für Lesen (`*NONE`: kein Kennwort erforderlich)
`WRITE-PASSWORD=`	Anzugeben für Schreiben (`*NONE`: kein Kennwort erforderlich)
`EXEC-PASSWORD=`	Anzugeben für Ausführen (`*NONE`: kein Kennwort erforderlich)

Bei der Änderung der Dateiattribute mit dem Kommando MODIFY-FILE-ATTRIBUTES sind die zuvor vergebenen Kennwörter gemäß der geltenden Kennworthierarchie (WRITE-PASSWORD – READ-PASSWORD – EXEC-PASSWORD) anzugeben.

Bei der Bearbeitung einer Datei (Kommandos ADD-FILE-LINK EXPIRATION-DATE= ... oder MODIFY-FILE-ATTRIBUTES ...,PROTECTION=*PARAMETERS(EXPIRATION-DATE=...) kann eine Schutzfrist vergeben bzw. geändert werden.

Für die Zeitdauer dieser Schutzfrist kann ein möglicher Schreibzugriff (durch das Attribut „w“ in der einfachen Zugriffskontroll-Liste einer Datei oder ein auf Grund von ACCESS= *WRITE bei Standardzugriffskontrolle) unwirksam gemacht werden. Die Datei kann dann nur gelesen werden.

> ¡ Wenn eine mit einem Lese- und/oder Ausführungskennwort geschützte Datei umgewandelt wird, verliert die Datei dabei den Schutz durch Lese- und/oder Ausführungskennwort.

12.3.2 Dateiverschlüsselung

Durch die Dateiverschlüsselung mit Crypto-Kennwort ist es möglich, den Inhalt einer Datei vor unbefugtem Zugriff zu schützen – auch gegenüber Personen mit TSOS-Privileg und auch gegen den physikalischen Zugriff auf Platten und Sicherungsbänder.

Eine verschlüsselte Datei wird durch Umwandlung einer normalen Datei mit dem Kommando ENCRYPT-FILE erzeugt. Dabei erfolgt die Festlegung des Crypto-Kennworts und das im Systemparameter FILECRYP eingestellte Verschlüsselungsverfahren wird in den Katalogeintrag der Datei übernommen.

Das Kommando DECRYPT-FILE macht aus verschlüsselten Dateien wieder unverschlüsselte. Dies ist nur nach Vorgabe des korrekten Cypto-Kennworts möglich.

Crypto-Kennwort

Ein Crypto-Kennwort ist max. 8 Zeichen lang, ohne Unterscheidung von Groß- und Kleinschreibung.

Zur Vorgabe der Zugriffsberechtigung auf eine verschlüsselte Datei wird das zugehörige Crypto-Kennwort mit dem Kommando ADD-CRYPTO-PASSWORD in der tasklokalen Crypto-Kennworttabelle hinterlegt.

Das explizite Entfernen eines Crypto-Kennworts aus der Kennworttabelle erfolgt mit dem Kommando REMOVE-CRYPTO-PASSWORD (sonst implizit beim Beenden der Session).

> ¡ Ohne Angabe des korrekten Crypto-Kennworts ist eine Entschlüsselung unmöglich. Gegen Verlust des Crypto-Kennworts werden folgende Maßnahmen empfohlen:
>
> - Hinterlegen der festgelegten Crypto-Kennworte an einem sicheren Ort
> - Anzahl der verwendeten Crypto-Kennwörter klein halten. Dazu gehört:
>   - Zusammengehörige verschlüsselte Dateien immer mit dem gleichen Crypto-Kennwort schützen (Benutzer: ENCRYPT-FILE mit Angabe einer Referenzdatei).
>   - Neben bestimmten verschlüsselten Musterdateien nur verschlüsselte Dateien mit gleichen Crypto-Kennworten wie diese Musterdateien erlauben (Systemparameter FREFCRYP).

Arbeiten mit verschlüsselten Dateien

Beim **homogenen Transfer** einer verschlüsselten Datei wird der verschlüsselte Inhalt einszu-eins in eine Empfangsdatei übertragen, die dieselben Verschlüsselungsattribute hat wie die Sendedatei.

Dieser homogene Transfer wird benutzt bei:

- homogenem COPY-FILE
- Sichern und Rekonstruieren (SAVE/RESTORE) mit HSMS/ARCHIVE
- Verdrängen und Zurückholen (MIGRATION/RECALL) mit HSMS
- Export und Import mit HSMS/ARCHIVE
- Verlagern von Dateien innerhalb eines SM-Pubsets auf einen anderen Volume-Set
- homogener File-Transfer

Beim homogenen Transfer findet also keine Entschlüsselung statt und somit wird auch kein Key und kein Crypto-Kennwort benötigt. Dadurch kann insbesondere die Systembetreuung bei Transfer-Aktionen mit verschlüsselten Dateien arbeiten wie mit unverschlüsselten Dateien.

Bei der Ausgabe des Kommandos SHOW-FILE-ATTRIBUTES zeigt das Feld ENCRYPTION im Abschnitt SECURITY an, ob und mit welchem Verschlüsselungsverfahren (AES oder DES) eine Datei verschlüsselt ist. Eine Dateiselektion wird entsprechend der Werte des Operanden ENCRYPTION angeboten.

Bei Fern-Dateizugriff über RFA werden die Kommandos ADD- und REMOVE-CRYPTO-PASSWORD von der Aufrufertask automatisch an alle angeschlossenen RFA-Partnerprozesse weitergeleitet.

DAB unterstützt das Lese-Caching von verschlüsselten und unverschlüsselten Dateien, nicht jedoch das Schreib-Caching.
Lese-Caching ist für verschlüsselte Dateien günstig, um die durch das Verschlüsselungsverfahren verlängerten Zugriffszeiten zu verkürzen.

Einschränkungen und Besonderheiten

- Durch die Bindung an das Subsystem openCrypt ist ab „System Ready“ der Zugriff auf verschlüsselte Dateien möglich.
- Nicht verschlüsselt werden Jobvariablen, Banddateien, EAM-Dateien, Dateien auf Privatplatte und TSOS-Dateien auf dem Home Pubset.
- Verschlüsselte Dateien können nicht gedruckt werden. Eine verschlüsselte Datei muss erst entschlüsselt werden.
- Verschlüsselte Dateien haben kein Kennwort für Lesen und Ausführen. Sie können jedoch ein Schreib-Kennwort haben und sind auch mit den anderen Zugriffsschutzmechanismen kombinierbar.
- Die Änderung des Crypto-Kennworts mit dem Kommando MODIFY-FILE-ATTRIBUTES ist nicht möglich. Änderungen von READ- oder EXEC-PASSWORD werden ignoriert.
- Bei Einsatz von PAMCONV oder bei Aufruf des Kommandos REPAIR-DISK-FILE für eine verschlüsselte Datei ist die Vorgabe des Crypto-Kennworts erforderlich.

Möglichkeiten der Steuerung über Systemparameter

1. FILECRYP

   Der Systemparameter FILECRYP bestimmt das Verschlüsselungsverfahren beim Umwandeln in eine verschlüsselter Datei mit dem Kommando ENCRYPT-FILE.
   Es werden die Verschlüsselungsverfahren AES (Voreinstellung) und DES unterstützt.

   - Der aktuelle Wert des Systemparameters FILECRYP wird bei ENCRYPT-FILE in die Verschlüsselungsmerkmale der Datei übernommen.
   - Beim Zugriff auf eine bereits verschlüsselte Datei wird das Verschlüsselungsverfahren nicht dem Systemparameter FILECRYP entnommen, sondern dem Katalogeintrag der Datei.
   - Eine Änderung des Systemparameters FILECRYP wird für eine zum Zeitpunkt der Änderung bereits verschlüsselte Datei erst dann wirksam, wenn die Datei entschlüsselt und anschließend erneut verschlüsselt wird.

   Beim Shared-Pubset-Betrieb mit verschlüsselten Dateien sollte die Wahl der einzelnen Verschlüsselungsverfahren im Systemparameter FILECRYP der beteiligten Systeme einheitlich sein.
2. FREFCRYP

Zur kontrollierten Vergabe und zahlenmäßigen Begrenzung der verwendeten Crypto-Kennwörter steht der Systemparameter FREFCRYP zur Verfügung.
Enthält - falls nicht leer - eine ausgewählte Benutzerkennung. Nur für Dateien von dieser Benutzerkennung ist dann die Umwandlung in verschlüsselte Dateien (Kommando ENCRYPT-FILE) mit freier Definition eines Crypto-Kennworts möglich.
Bei der Umwandlung von Dateien anderer Benutzerkennungen muss dann eine schon verschlüsselte Referenzdatei angegeben werden. Die Menge aller verwendeten Crypto-Kennwörter wird damit begrenzt auf die der verschlüsselten Dateien von der ausgewählten Benutzerkennung.

3. PWACTIVE

   Mit dem Systemparameter PWACTIVE wird festgelegt, wie viele Crypto-Kennwörter eine Crypto-Kennworttabelle maximal enthalten darf. Wird der Grenzwert erreicht, wird die Meldung `DMS0691` ausgegeben und es kann für die laufende Task kein weiteres Crypto-Kennwort mehr eingegeben werden, bevor nicht mindestens eines der Crypto-Kennwörter aus der Crypto-Kennworttabelle wieder ausgetragen wurde.

4. PWENTERD

   Mit dem Systemparameter PWENTERD wird festgelegt, wie viele Crypto-Kennwörter pro Task eingegeben werden dürfen. Wird der Grenzwert erreicht, wird die Meldung `DMS0692` ausgegeben und es kann für die laufende Task kein weiteres Crypto-Kennwort mehr eingegeben werden.

5. PWERRORS

   Mit dem Systemparameter PWERRORS wird festgelegt, wie viele Crypto-Fehlzugriffsversuche unter einer Task maximal toleriert werden. Wird der Grenzwert erreicht, wird ggf. ein SAT-Eintrag geschrieben, die Meldung `DMS0693` auf die Konsole ausgegeben und die Task abnormal beendet.

6. PWPENTI

   Mit dem Systemparameter PWPENTI wird eine Festlegung für die Zeitstrafe der zu tolerierenden Crypto-Fehlzugriffsversuche getroffen.

> **i** Alle oben aufgeführten Grenzwerte gelten sowohl für die Eingabe von Crypto-Kennwörtern als auch für die Eingabe von Datei-Kennwörtern (READ, WRITE, EXEC). Im Gegensatz zu den Datei-Kennwörtern, bei deren Eingabe die privilegierte Systembetreuung das Sonderrecht besitzt, die festgelegten Grenzwerte zu überschreiten, werden bei der Eingabe von Crypto-Kennwörtern keinerlei Sonderrechte gewährt.
>
> Crypto-Kennworttabellen und Crypto-Kennwortzähler werden getrennt von den Datei-Kennworttabellen und den Datei-Kennwortzählern geführt. Ist zum Beispiel unter einer Task die maximal mögliche Anzahl von READ-, WRITE- und EXEC-Kennworteingaben erreicht, sind durchaus weitere Crypto-Kennworteingaben möglich und umgekehrt.
>
> Der Parameter ENCRYPT wirkt sich nur aus auf die Verschlüsselung von Dateikennworten, nicht auf die von Crypto-Kennworten. Diese werden stets mit Einweg-Verschlüsselung in der Crypto-Kennworttabelle hinterlegt.

Für weitere Hinweise zur Nutzung von verschlüsselten Dateien siehe Handbuch „Einführung in das DVS“ [19].

12.3.3 Standard-Zugriffskontrolle (ACCESS/USER-ACCESS)

Gesteuert wird die Standard-Zugriffskontrolle über die Operanden ACCESS und USER-ACCESS der CREATE- und MODIFY-Kommandos.

Die Standard-Zugriffskontrolle über die Schutzattribute ACCESS und USER-ACCESS bietet sich dann an, wenn ein höherer Zugriffsschutz durch die einfache Zugriffskontroll-Liste nicht gewünscht wird.

Ist weder eine BACL noch ein Guard definiert, wird die Standard-Zugriffskontrolle wirksam. Zusätzlich werden immer die Kennwörter und die Schutzfrist überprüft.

Schutzattribut **ACCESS**

Mit dem Schutzattribut ACCESS kann für ein Objekt Schreib- oder Leserecht festgelegt werden. Das Schreibrecht schließt hierbei das Leserecht ein.

Schutzattribut **USER-ACCESS**

Mit dem Schutzattribut USER-ACCESS kann festgelegt werden, ob auf eine Datei nur der Eigentümer (*USER-ONLY) oder auch alle übrigen Benutzer (*ALL-USERS) – ausschließlich oder einschließlich der Benutzerkennung (en) mit dem Privileg HARDWARE-MAINTENANCE für die Online-Wartung) – zugreifen dürfen.

Anmerkung für Benutzerkennungen mit dem Privileg HARDWARE-MAINTENANCE

Ein Dateizugriff wird einer Benutzerkennung mit diesem Privileg nur dann erlaubt, wenn folgende Bedingungen erfüllt sind:

- Falls die Datei mit Guards geschützt ist, müssen im Guard Zugriffsbedingungen festgelegt sein, die der privilegierten Benutzerkennung den Zugriff erlauben.
- Falls die Datei nicht mit Guards geschützt ist, aber durch eine einfache Zugriffskontroll-Liste (BACL), muss diese der privilegierten Benutzerkennung den Zugriff erlauben.
- Wenn die Datei weder mit Guards noch mit einer BACL geschützt ist, muss USER-ACCESS=*SPECIAL gesetzt sein.

12.3.4 Einfache Zugriffskontroll-Liste (BACL)

Eine Stufe über den Schutzattributen ACCESS/USER-ACCESS liegt in der Hierarchie der Zugriffsschutzmechanismen die einfache Zugriffskontroll-Liste (Basic Access Control List, BACL). Sie ist für ein Objekt dann wirksam, wenn für das Objekt kein Guards-Schutz definiert ist. Kennwortschutz und Schutzfrist sind zusätzlich wirksam.

Mit einer BACL können für den Objekt-Eigentümer, für die Mitglieder seiner Benutzergruppe und für alle anderen Benutzer unterschiedliche Zugriffsrechte definiert werden. Eine Festlegung von Zugriffsrechten für einzelne Benutzerkennungen ist mit diesem Zugriffsschutzmechanismus jedoch nicht möglich.

Eine einfache Zugriffskontroll-Liste für Dateien wird mit dem Operanden BASIC-ACL der Kommandos CREATE-FILE oder MODIFY-FILE-ATTRIBUTES definiert.

Einfache Zugriffskontroll-Listen für Jobvariablen können entsprechend mit den Kommandos CREATE-JV oder MODIFY-JV-ATTRIBUTES definiert werden.

Benutzerklassen

Aufbauend auf dem Konzept der Benutzergruppen werden Benutzerklassen für den Zugriff auf Objekte festgelegt. Die Benutzerklassen unterteilen die Menge aller Benutzer jeweils in die Teilmengen OWNER, GROUP und OTHERS.

OWNER	der Eigentümer eines Objekts, also die Benutzerkennung, unter der die Datei oder Jobvariable katalogisiert ist, sowie Miteigentümer, die mit Hilfe des Miteigentümerschutzes festgelegt wurden
GROUP	alle Benutzerkennungen der Benutzergruppe, der der Eigentümer angehört, mit Ausnahme des Eigentümers selbst und der Miteigentümer
OTHERS	alle übrigen Benutzer mit Ausnahme der Miteigentümer

Zur Festlegung der Benutzerklasse wird die Definition der Gruppenstruktur auf dem Home-Pubset herangezogen.

Hinweise zur Benutzerklasse GROUP

Alle Benutzer, die keiner explizit eingerichteten Gruppe zugeordnet sind, sind automatisch Mitglied der implizit eingerichteten Gruppe *UNIVERSAL. Dies gilt insbesondere dann, wenn gar keine Gruppen explizit eingerichtet wurden. In diesem Fall sind alle Systembenutzer Mitglied derselben Gruppe. Bei der Auswertung einer BACL erhalten daher alle zugreifenden Benutzerkennungen außer dem Objekteigentümer selbst die Zugriffsrechte aus dem GROUP-Eintrag und nicht die des OTHERS-Eintrags.

Für Mitglieder der Benutzergruppe *UNIVERSAL wird daher dringend empfohlen, für die Benutzerklassen GROUP und OTHERS die gleichen Zugriffsrechte zu vergeben.

Zugriffsrechte

In einer BACL sind neun Zugriffsberechtigungen für eine Datei festgelegt. Der Datei können für jede der drei Benutzerklassen OWNER, GROUP, OTHERS drei Zugriffstypen separat zugeordnet werden:

- Lesen (R)
- Schreiben (W)
- Ausführen (X)

Keines dieser Zugriffsrechte schließt ein anderes ein.

Auswertung der einfachen Zugriffskontroll-Liste

1. Ist die Benutzerkennung, die den Zugriff wünscht, der Eigentümer oder ein Miteigentümer des Objekts bzw. TSOS, gelten die unter OWNER abgespeicherten Zugriffsrechte.
2. Gehört die Benutzerkennung der Benutzergruppe des Eigentümers an, gelten die unter GROUP abgespeicherten Zugriffsrechte.
3. Für alle anderen Benutzerkennungen gelten die unter OTHERS abgespeicherten Zugriffsrechte.

Beispiel

OWNER = R W X

GROUP = R W -

OTHERS = R - -

Der Eigentümer dieser Datei darf auf die Datei lesend, schreibend und ausführend zugreifen. Die Gruppe des Dateieigentümers darf die Datei lesen und in die Datei schreiben. Der Rest darf die Datei nur lesen.

12.4 Privilegien

Systemprivilegien (kurz: Privilegien) beschreiben das Recht, bestimmte Aufgabenbereiche der Systembetreuung mit den dazu erforderlichen Systemfunktionen unter einer bestimmten Benutzerkennung abzuwickeln.

Privilegien können an unterschiedliche Benutzerkennungen vergeben werden. Diese Verteilung bewirkt zum einen eine Entlastung der Systembetreuung. Zum anderen wird dadurch die Sicherheit im Rahmen der Systembetreuung erhöht, weil z.B. der Kreis derer kleiner wird, die für anfallende Routinearbeiten das TSOS-Kennwort kennen müssen.

Standardmäßig sind die Privilegien an vordefinierte Benutzerkennungen vergeben, siehe Tabelle 36 ("Privilegienverteilung"). Die Standard-Privilegienverteilung kann nur mit SECOS geändert werden.

Jedes Kommando muss in einer der aktivierten System-Syntaxdateien deklariert und seine Verwendung ausdrücklich zugelassen sein. Jedes an SDF übergebene Kommando (Benutzer-, Systemverwaltungs- oder Operatorkommando) wird vor der Verarbeitung daraufhin überprüft, ob der Aufrufer mit dem zur Ausführung nötigen Privileg ausgestattet ist, im Falle der Operatorkommandos ist dies z.B. das Privileg OPERATING.

Die folgende Übersicht zeigt anhand einiger Kommandos die Aufteilung der Berechtigungen, Benutzer-, Operator- und Systemverwaltungskommandos an BS2000 zu geben:

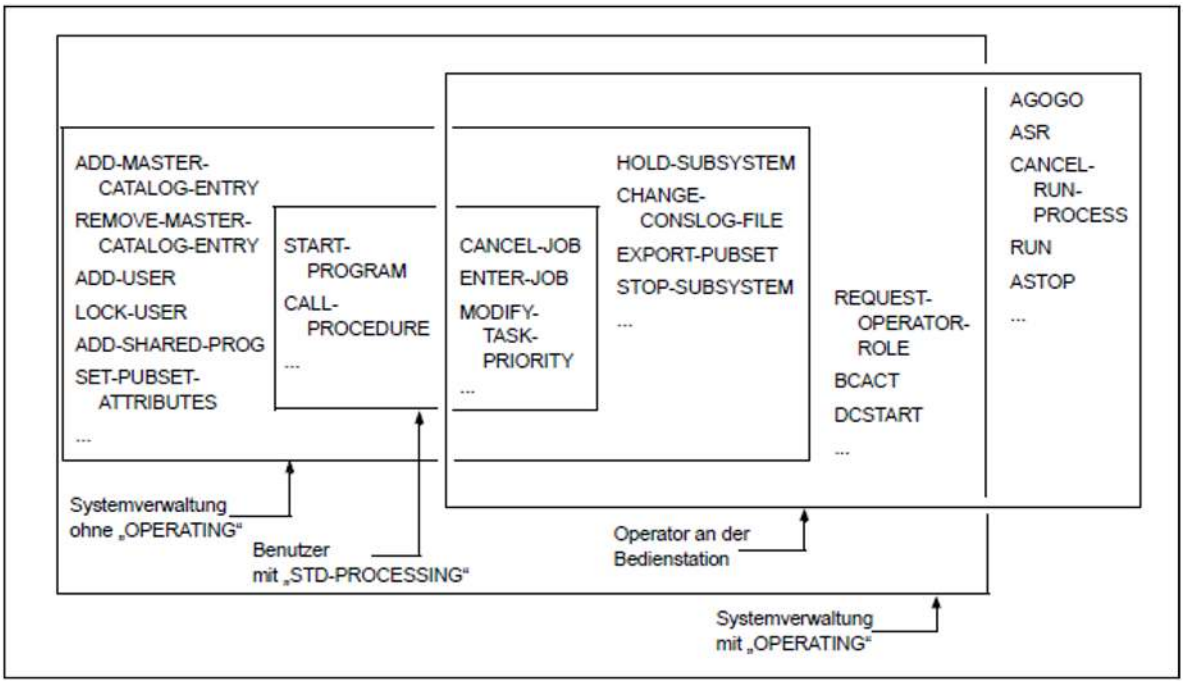


Bild 20: Berechtigungen zur Eingabe von Kommandos

Es gibt also einige Kommandos, die sowohl vom Operator (über Konsole) als auch von der Systemverwaltung (unter der Benutzerkennung TSOS) eingegeben werden können. Die Aufgabenteilung zwischen Systemverwaltung und Operator ist deshalb nicht starr. Hier ist ein gewisser Spielraum in der Organisation des Data Centers vorgesehen. In jedem Fall muss die Zusammenarbeit zwischen Systemverwaltung und Operator sehr eng sein.

Privilegienverteilung mit SECOS

Jede Benutzerkennung in BS2000 ist mit mindestens einem Privileg ausgestattet. Standardmäßig ist das das Privileg STD-PROCESSING, d.h. das Recht, die Benutzerkommandos auszuführen.

Eine Benutzerkennung kann mehrere Privilegien besitzen und ein Privileg kann mit SECOS an mehrere Benutzerkennungen vergeben werden.

Mit SECOS können einzelne Privilegien zu bestimmten Aufgabenbereichen gruppiert werden. Eine solche Gruppierung wird durch die Definition von „Sammelprivilegien“ realisiert, denen die verschiedenen (Einzel-) Privilegien zugeordnet werden können.

Die Privilegien einer Benutzerkennung werden im Benutzerkatalog (SYSSRPM) hinterlegt. Die Privilegienverteilung im Benutzerkatalog des Home-Pubsets ist systemweit wirksam. Der Benutzerkatalog eines Pubsets wird beim Importieren dieses Pubsets geöffnet und bleibt offen bis zum Exportieren des Pubsets.

Die einzelnen Privilegien sind im Abschnitt „Beschreibung der Privilegien" beschrieben.

12.4.1 Privilegien der Benutzerkennung TSOS

Das Privileg **TSOS** ist fest an die Benutzerkennung TSOS geknüpft und kann weder dieser Benutzerkennung entzogen noch an eine andere Benutzerkennung vergeben werden.
Welche Privilegien der Benutzerkennung TSOS standardmäßig zugewiesen sind, kann der Tabelle 36 ("Privilegienverteilung") entnommen werden.

> ¡ Gemäß BS2000-Konvention beginnen Produktdateien oder vom System unter der Benutzerkennung TSOS angelegte Dateien mit der Zeichenfolge „SYS“ (für SU /390) oder „SKM“ und „SKU“ (für SU x86). Es muss darauf geachtet werden, dass sie nicht versehentlich überschrieben oder gelöscht werden.

Benutzerkommandos

Die Benutzerkennung TSOS ist Miteigentümer aller Dateien und Jobvariablen sämtlicher Benutzer, solange ihr diese Miteigentümerschaft nicht durch den Eigentümer selbst entzogen wird.

Der Systembetreuung steht der volle Umfang der Benutzerkommandos zur Verfügung.

TSOS ist berechtigt, auf alle Einträge im Datei- oder Benutzerkatalog lesend und schreibend zuzugreifen. Der Kennwortschutz von Benutzerdateien ist für die Systembetreuung insofern aufgehoben, als TSOS berechtigt ist, entweder die Kennwörter zu ignorieren oder bei Bedarf zu ermitteln.
Mit dem Einsatz von SECOS kann der Eigentümer seine sicherheitskritischen Dateien mit einem Bedingungs-Guard versehen, der TSOS den Zugriff verbietet.

Der Zugang zu allen Dateien schließt auch temporäre Dateien ein, die die Systembetreuung unter jeder beliebigen Katalog- und Benutzerkennung einrichten darf. Diese Dateien werden jedoch **nicht** automatisch bei EXIT-JOB bzw. LOGOFF gelöscht. Für das Löschen dieser Dateien ist die Systembetreuung selbst verantwortlich. Temporäre Dateien werden bei einer teilqualifizierten Angabe von Dateinamen entsprechend ihrer internen Darstellung berücksichtigt.

Kommandos mit anderen Privilegien als TSOS

Die Benutzerkennung TSOS kann nur die Kommandos ausführen, für die sie auch das entsprechende Privileg besitzt.

Insbesondere können unter TSOS keine Operatorkommandos eingegeben werden, da TSOS standardmäßig nicht das Privileg OPERATING besitzt.
Die Standard-Privilegienverteilung kann nur mit SECOS geändert werden.

Makros

Die Privilegien für das Einrichten und Verändern von Dateien auf Kommandoebene gelten ebenfalls bei den entsprechenden Makroaufrufen.
Die ausführliche Beschreibung der privilegierten Operanden der Makros ist den Handbüchern „Makroaufrufe an den Ablaufteil“ [30] und „DVS Makros“ [20] zu entnehmen.

Jobvariablen

Die nachfolgenden Kommandos stehen nur im Zusammenhang mit dem Softwareprodukt JV zur Verfügung und sind im Handbuch „Jobvariablen“ [26] ausführlich beschrieben.

Die Benutzerkennung TSOS ist Miteigentümer aller Dateien und Jobvariablen sämtlicher Benutzer, solange ihr diese Miteigentümerschaft nicht durch den Eigentümer selbst entzogen wird.

Die Systembetreuung kann mit den folgenden Kommandos Jobvariablen anderer Benutzerkennungen bearbeiten:

```
CREATE-JV
DELETE-JV
MODIFY-JV
MODIFY-JV-ATTRIBUTES
MODIFY-JV-CONDITIONALLY
REMOVE-JV-LINK
SET-JV-LINK
SHOW-JV
SHOW-JV-ATTRIBUTES
SHOW-JV-LINK
SHOW-CJC-STATUS
```

Eventuell vereinbarter Dateischutz in Form von Kennwörtern wird nur dem privilegierten Aufrufer unter der Benutzerkennung TSOS angezeigt.

12.4.2 Privilegien für das Operating

Das Privileg **OPERATING** ist standardmäßig der Benutzerkennung SYSOPR zugeordnet. Die Standard-Privilegienverteilung kann nur mit SECOS geändert werden.

Damit ist es möglich, Operating sowohl von physikalischen und logischen Konsolen als auch von Benutzertasks mit dem Privileg OPERATING zu betreiben.
Für Konsolen bleibt das Kommandoberechtigungskonzept über Berechtigungsschlüssel unverändert erhalten. Für Benutzertasks spielt es keine Rolle mehr (Ausnahme: Kommandos, die über die UCON-Task abgearbeitet werden und denen der Berechtigungsschlüssel $ zugeordnet ist, siehe auch "Kommandodefinition").

Der Systembetreuung ist es freigestellt, bestimmte Kommandos, die der Operator grundsätzlich verwenden dürfte, in der Systemsyntaxdatei zu sperren (siehe auch Handbuch „SDF-A“ [44]).
Eine Task mit dem Privileg OPERATING bietet die meisten Operator-Kommandos an, also auch diejenigen, welche bisher ausschließlich über Konsole eingegeben werden konnten. Nicht angeboten werden z.B. die Kommandos CONSOLE, ASR, RUN, AGOGO.

Jedes von einer Konsole eingegebene Operatorkommando wird durch eine OPR-Task verwaltet, die sich beim Ausführen eines Kommandos gegenüber SDF und dem System als privilegiert ausweist. Jede Anforderung des Operators (von der OPR-Task zur Syntax- und Privilegienprüfung an SDF übergebene Kommandos) wird vor der Verarbeitung daraufhin überprüft, ob der Aufrufer mit dem zur Ausführung nötigen Privileg OPERATING ausgestattet ist.
Die OPR-Task einer physikalischen Konsole (bei NBCONOPI=NO) und einer logischen Konsole besitzt immer das Privileg OPERATING.

Der Operator kann die Kommandos über die Konsole eingeben.
Die Kurzform ist, falls vorgesehen, in der Kommandosyntax angegeben.
Zum Aufbau der Kommandoeingaben und Meldungen siehe Kapitel „Operatorfunktionen" und Kapitel „Automatisierung von Operatorfunktionen".

Die Kommandos sind im Handbuch „Kommandos“ [27] beschrieben. Die BCAM-Kommandos für den Operator (z.B. BCACT, DCSTART) sind im Handbuch „BCAM“ [4] beschrieben.

Unter bestimmten Voraussetzungen kann der Operator auch einige Systemverwaltungs- bzw. Benutzerkommandos verwenden (siehe Abschnitt „Ausübung von Systemverwaltungsfunktionen durch den Operator").

12.4.3 Beschreibung der Privilegien

BS2000 kennt folgende Privilegien. Sie sind im Anschluss an diese Tabelle beschrieben.

Verwaltungsbereich	**Name des Privilegs**
ACS-Verwaltung	ACS-ADMINISTRATION
(ist nicht vordefiniert, wird von der Systembetreuung bestimmt)	CUSTOMER-PRIVILEGE-1 : CUSTOMER-PRIVILEGE-8
File-Transfer-Verwaltung	FT-ADMINISTRATION
FTAC-Verwaltung	FTAC-ADMINISTRATION
Systemglobale Guard-Administration	GUARD-ADMINISTRATION
Hardware-Online-Wartung	HARDWARE-MAINTENANCE
HSMS-Verwaltung	HSMS-ADMINISTRATION
Netzverwaltung (obsolet)	NET-ADMINISTRATION
Notification-Service-Administration	NOTIFICATION-ADMINISTRATION
Operating	OPERATING
POSIX-Benutzerverwaltung	POSIX-ADMINISTRATION
SPOOL-Verwaltung	PRINT-SERVICE-ADMINISTRATION
PROP-XT-Verwaltung	PROP-ADMINISTRATION
Audit-Datei-Auswertung	SAT-FILE-EVALUATION
Audit-Datei-Verwaltung	SAT-FILE-MANAGEMENT
Sicherheitsverwaltung	SECURITY-ADMINISTRATION
Ausführen von Benutzerkommandos	STD-PROCESSING
Subsystem-Verwaltung	SUBSYSTEM-MANAGEMENT
Software-Monitor-Verwaltung	SW-MONITOR-ADMINISTRATION
Bandverwaltung	TAPE-ADMINISTRATION
Encryption-Key-Verwaltung für Bänder	TAPE-KEY-ADMINISTRATION
TSOS	TSOS
Benutzerverwaltung	USER-ADMINISTRATION

VM-Verwaltung	VIRTUAL-MACHINE-ADMINISTRATION
VM2000-Verwaltung	VM2000-ADMINISTRATION

Tabelle 35: Verwaltungsbereiche und dazugehörende Privilegien

ACS-Verwaltung

Die mit dem Privileg **ACS-ADMINISTRATION** ausgestattete Benutzerkennung darf im Rahmen der Funktion ACS (Alias Catalog Service) folgende Rechte wahrnehmen:

- systemglobale Voreinstellungen und Einschränkungen für die Benutzung eines Aliaskataloges festlegen
- die Vereinbarungen hinsichtlich der ACS-Systemdateien treffen bzw. ändern
- erweiterte Funktionen einzelner ACS-Kommandos wahrnehmen

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Der Rechte- und Funktionsumfang der Benutzerkennung mit dem Privileg ACS-ADMINISTRATION ist im Abschnitt „ACS: Alias-Katalogsystem" beschrieben.

CUSTOMER-PRIVILEGE-1..8

Durch die Vergabe der Privilegien **CUSTOMER-PRIVILEGE-1** oder CUSTOMER-PRIVILEGE-2 oder CUSTOMER-PRIVILEGE-3 u.s.w. bis **CUSTOMER-PRIVILEGE-8** ist es der Systembetreuung möglich, den Zugang zu Kommandos und Anweisungen für bestimmte Benutzerkennungen flexibel zu gestalten.

Standardmäßig sind diese Privilegien an keine Benutzerkennung vergeben.

File-Transfer-Verwaltung

Die File-Transfer-Verwaltung (das ist die Benutzerkennung mit dem Privileg **FT-ADMINISTRATION**) darf das Auftrags- und Netzbeschreibungsbuch des Softwareprodukts openFT verwalten (siehe Handbuch „openFT“ [23]).

FTAC-Verwaltung

Die FTAC-Verwaltung (das ist die Benutzerkennung mit dem Privileg **FTAC-ADMINISTRATION**) darf die Schutzfunktionen des Softwareprodukts openFT-AC verwalten (siehe Handbuch „openFT“ [23]).

Systemglobale Guard-Administration (GUARD-ADMINISTRATION)

Die systemglobale Guard-Administration mit dem Privileg **GUARD-ADMINISTRATION** hat das Recht, auf allen lokalen Pubsets Aktionen zur Verwaltung von Guards beliebigen Typs durchzuführen und mit dem Programm GUARDS-SAVE benutzerkennungsübergreifend Guards zu sichern und zu rekonstrurieren. Das heißt, eine Benutzerkennung mit diesem Privileg ist Miteigentümer aller Guards im System.
Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Hardware-Online-Wartung

Das Privileg **HARDWARE-MAINTENANCE** beinhaltet das Recht zur Durchführung der Hardware-Online-Wartung. Es umfasst insbesondere folgende Funktionen:

- Führen der Hardware-Fehlerstatistikdatei
- Ablauf von Statistik- und Verfolgungs-Programmen unter Steuerung von BS2000 simultan zu den Benutzerprogrammen

Das Privileg ist standardmäßig der Benutzerkennung SERVICE zugeordnet.

Wird das Privileg HARDWARE-MAINTENANCE an eine beliebige Benutzerkennung vergeben, ist Folgendes zu beachten:

- Eine Benutzerkennung mit dem Privileg HARDWARE-MAINTENANCE ist nur dann zum Zugriff auf Dateien fremder Benutzerkennungen berechtigt, wenn Folgendes gilt:
  - Falls die Datei mit Guards geschützt ist, müssen im Guard Zugriffsbedingungen festgelegt sein, die der privilegierten Benutzerkennung den Zugriff erlauben.
  - Falls die Datei nicht mit Guards, aber durch eine einfache Zugriffskontroll-Liste geschützt ist, muss diese der privilegierten Benutzerkennung den Zugriff erlauben.
  - Wenn die Datei weder mit Guards noch mit einer BACL geschützt ist, muss USER-ACCESS=*SPECIAL gesetzt sein.
- Es muss dafür gesorgt werden, dass diese Benutzerkennung alle benötigten Zugriffsrechte hat.

Weitere Informationen über die Hardware-Online-Auswertung finden Sie im „Diagnosehandbuch“ [14].

HSMS-Verwaltung

Die Benutzerkennung mit dem Privileg **HSMS-ADMINISTRATION** darf in BS2000 Aktionen zur Verwaltung des „Hierarchical Storage Management System“ durchführen (siehe Handbuch „HSMS“ [24]).

Das Privileg ist standardmäßig den Benutzerkennungen TSOS und SYSHSMS zugeordnet.

Netzverwaltung

Ein Benutzer-Job mit dem Privileg **NET-ADMINISTRATION** darf Netzverwaltungsfunktionen ausüben, u.a. die Redefinition von Stationsnamen.

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Notification-Service-Administration

Das Privileg **NOTIFICATION-ADMINISTRATION** berechtigt zur Konfiguration des Notification Service, d. h. es erlaubt die Produkte festzulegen, die den Notification Service nutzen dürfen und welche Methoden zur Benachrichtigung dabei unterstützt werden.

Das Privileg ist standardmäßig den Benutzerkennungen TSOS und SYSSNS zugeordnet.

Notification Service in BS2000 ist ein Mechanismus, mit dem Benutzer beim Auftreten bestimmter Ereignisse benachrichtigt werden können. Derzeit wird diese Funktionalität von SPOOL genutzt. Ein Benutzer kann per Mail benachrichtigt werden, wenn bei seinen Druckaufträgen bestimmte Ereignisse, z.B. Job-Beendigung, eintreten. Siehe auch das Handbuch „SNS“ [51].

Operating

Das Privileg **OPERATING** erlaubt, Aufgaben des Operators wahrzunehmen. Somit können Operator-Funktionen auch von Benutzerarbeitsplätzen ausgeführt werden.
Zur Basissystembedienung ist jedoch die Bedienung von (der normalen) Konsole weiterhin unablässig.

Das Privileg ist standardmäßig der Benutzerkennung SYSOPR zugeordnet.
Siehe auch Abschnitt „Privilegien für das Operating".

POSIX-Benutzerverwaltung

Das Privileg **POSIX-ADMINISTRATION** schützt den Zugriff auf POSIX-Attribute, die durch die BS2000-Benutzerverwaltung verwaltet werden, sowie das Tool zum Installieren zusätzlicher Produkte in POSIX. Die POSIX-Attribute können u.a. folgendermaßen geschützt werden:

- durch die Verwaltung der POSIX-Benutzerattribute aller Benutzerkennungen auf allen lokalen Pubsets. Diese Berechtigung stellt eine Untermenge des Privilegs USER-ADMINISTRATION dar.
- durch die Nutzung privilegierter Teilfunktionen des POSIX-SVC

Das Privileg ist standardmäßig der Benutzerkennung SYSROOT zugeordnet und kann dieser nicht entzogen werden.

Siehe auch Kapitel „POSIX-Benutzerverwaltung".

SPOOL-Verwaltung

Das Privileg **PRINT-SERVICE-ADMINISTRATION** erlaubt, SPOOL-Verwaltungsaufgaben durchzuführen. Dazu gehört:

- Start und Stopp von SPOOL-Geräten wie Drucker oder Bänder
- Veränderung der SPOOL-Parameter mit dem Dienstprogramm SPSERVE
- Veränderung von Drucksteuerdateien mit dem Dienstprogramm PRM
- Verteilter Zugriff auf vernetzte Drucker mit Dprint
- Verwaltung der Druckaufträge aller Benutzer

Das Privileg ist standardmäßig den Benutzerkennungen TSOS, SYSSNS und SYSSPOOL zugeordnet.

PROP-XT-Verwaltung

Das Privileg **PROP-ADMINISTRATION** erlaubt, PROP-XT-Systemkommandos auszuführen. Zur PROP-XT-Verwaltung siehe das Handbuch „PROP-XT“ [40].

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Auswertung der Audit-Dateien

Das Privileg **SAT-FILE-EVALUATION** gewährt das Recht, die SATLOG-Dateien und die CONSLOG-Dateien auszuwerten. Für den Inhaber des Privilegs SAT-FILE-EVALUATION ist die SAT-Protokollierung standardmäßig eingeschaltet, kann aber explizit wieder abgeschaltet werden.

Das Privileg ist standardmäßig der Benutzerkennung SYSAUDIT zugeordnet.

Verwaltung der Audit-Dateien

Die Audit-Datei-Verwaltung benötigt das Privileg **SAT-FILE-MANAGEMENT** und darf folgende Aktionen durchführen:

- die von SAT (Security Audit Trail) erzeugten Dateien verwalten (insbesondere die SATLOG-Datei mit dem Kommando CHANGE-SAT-FILE umschalten)
- mit dem Kommando SET-REPLOG-READ-MARK den aktuellen Stand der Rep-Logging-Datei anfordern
- Informationen über den aktuellen Stand der EVENTING-Auswahl einholen
- die SATLOG-Dateien und die CONSLOG-Dateien auswerten

Für den Inhaber des Privilegs SAT-FILE-MANAGEMENT ist die SAT-Protokollierung standardmäßig eingeschaltet, kann aber explizit wieder abgeschaltet werden.
Das Privileg ist standardmäßig der Benutzerkennung SYSAUDIT zugeordnet.

Sicherheitsverwaltung

Die Sicherheitsverwaltung (der Sicherheitsbeauftragte) hat das Recht zur Privilegienverwaltung, zur Verwaltung der Operator-Rollen und zum Schalten der Protokollierung (Auditing) mit SAT. Für die Benutzerkennung mit dem Privileg **SECURITY-ADMINISTRATION** ist die Protokollierung mit SAT immer eingeschaltet und kann nicht abgeschaltet werden.

Die Privilegienverwaltung darf die einzelnen Privilegien verwalten, d.h.

- Vergabe von Privilegien an Benutzerkennungen auf allen Pubsets
- Entzug von Privilegien für Benutzerkennungen auf allen Pubsets
- Vergabe und Entzug von Sammelprivilegien auf allen Pubsets
- Abfrage von Informationen über Sammelprivilegien

Hinsichtlich der Verwaltung der Operator-Rollen kann der Sicherheitsbeauftragte:

- Operator-Rollen definieren, modifizieren und löschen
- Operator-Rollen an Benutzerkennungen vergeben oder entziehen
- Informationen über die aktuelle Definition und Verteilung der Operator-Rollen abfragen

Das Privileg SECURITY-ADMINISTRATION kann nicht per Kommando an eine Benutzerkennung vergeben bzw. einer Benutzerkennung entzogen werden. Das Privileg ist in einem System ohne SECOS fest an die Benutzerkennung SYSPRIV geknüpft (die wie TSOS immer im Benutzerkatalog SYSSRPM enthalten ist), in einem System mit SECOS kann eine beliebige Benutzerkennung über den Startup-Parameterservice mit dem Privileg ausgestattet werden (siehe SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46]).

Ausführen von Benutzerkommandos

Mit dem Privileg **STD-PROCESSING** wird das Recht vergeben, die im Handbuch „Kommandos“ [27] beschriebenen Kommandos sowie die nichtprivilegierten Kommandos von Softwareprodukten, die nicht zu BS2000 OSD/BC gehören, auszuführen.

Beim Neueinrichten einer Benutzerkennung (mit ADD-USER) wird dieser Benutzerkennung standardmäßig das Privileg STD-PROCESSING zugewiesen. Es ist außerdem standardmäßig an die vom System bei First-Start erzeugten Benutzerkennungen vergeben, mit Ausnahme der Benutzerkennungen SERVICE, SYSAUDIT und SYSPRIV.

Obwohl gilt, dass eine Benutzerkennung, die mit einem Privileg ausgestattet ist, nicht gelöscht werden kann, wird beim Löschen einer Benutzerkennung (mit REMOVE-USER) der „Besitz“ des Privilegs STD-PROCESSING nicht berücksichtigt, d.h eine Benutzerkennung kann gelöscht werden, wenn ihr außer diesem Privileg kein anderes mehr zugeordnet ist.

Subsystem-Verwaltung

Die mit dem Privileg **SUBSYSTEM-MANAGEMENT** ausgestattete Benutzerkennung darf die Aufgaben zur globalen Verwaltung der Subsysteme der Dynamischen Subsystemverwaltung **DSSM** ausführen.

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Software-Monitor-Verwaltung

Eine Benutzerkennung, die mit dem Privileg **SW-MONITOR-ADMINISTRATION** ausgestattet ist, darf die Software-Monitore openSM2 und COSMOS starten, beenden und administrieren (siehe Handbuch „openSM2“ [49]).

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

Bandverwaltung

Die Bandverwaltung (das ist die Benutzerkennung, die mit dem Privileg **TAPE-ADMINISTRATION** ausgestattet ist) darf die Administrationsfunktionen des Softwareprodukts MAREN ausführen. D.h. sie darf den MAREN-Katalog verwalteen (siehe Handbuch „MAREN“ [31]).

Das Privileg ist standardmäßig den Benutzerkennungen TSOS und SYSMAREN zugeordnet.

Encryption-Key-Verwaltung für Bänder

Die Encryption-Key-Verwaltung (das ist die Benutzerkennung, die mit dem Privileg **TAPE-KEY-ADMINISTRATION** ausgestattet ist) darf die Anweisungen des Programms MARENEKM (MAREN Encryption Key Manager) ausführen. D.h. sie darf die Encryption Keys für Bänder verwalten (siehe Handbuch „MAREN“ [31]).

Das Privileg ist standardmäßig der Benutzerkennung SYSMAREN zugeordnet.

TSOS

Das Privileg **TSOS** erlaubt es der Benutzerkennung TSOS, Systemverwaltungsfunktionen auszuführen.

Das Privileg TSOS ist fest an die Benutzerkennung TSOS geknüpft und kann weder dieser Benutzerkennung entzogen noch an eine andere Benutzerkennung vergeben werden.

Benutzerverwaltung

Die systemglobale Benutzerverwaltung (das sind alle mit dem Privileg **USER-ADMINISTRATION** ausgestatteten Benutzerkennungen) darf auf allen lokalen Pubsets (für alle Benutzer oder Benutzergruppen) eine Aktion zur Benutzer- oder Benutzergruppenverwaltung durchführen.
Für die Vergabe von Betriebsmitteln und Rechten des Benutzerkatalog-Eintrags (wie z.B. START-IMMEDIATE, NO-CPU-LIMIT,...) an Benutzerkennungen und Benutzergruppen gibt es keine Begrenzungen.

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

VM-Verwaltung

Das Privileg **VIRTUAL-MACHINE-ADMINISTRATION** erlaubt, eine virtuelle Maschine (VM) mit Ausnahme der Monitor-VM zu bedienen. Es dürfen also VM-Administrator-Kommandos für die eigene VM gegeben werden (siehe Handbuch „VM2000“ [60]).

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

VM2000-Verwaltung

Das Privileg **VM2000-ADMINISTRATION** erlaubt, die Monitor-VM zu administrieren und sämtliche VM2000-Kommandos einzugeben (siehe Handbuch „VM2000“ [60]).

Das Privileg ist standardmäßig der Benutzerkennung TSOS zugeordnet.

12.4.4 Privilegienverteilung

Fährt man ein System mit First-Start hoch, wird der Benutzerkatalog SYSSRPM neu erzeugt. Standardmäßig haben vordefinierte Benutzerkennungen bestimmte Privilegien:

Privileg	**TSOS**	**SERVICES**	**SYSAUDIT**	**SYSHSMS**	**SYSMAREN**	**SYSOPR**	**SYSPRIV**[1]	**SYSROOT**	**SYSSNS**	**SYSSPOOL**	**SYSWSA**	**Andere**[2]
ACS-ADMINISTRATION	X[3]	-[4]	-	-	-	-	-	-	-	-	-	-
CUSTOMER-PRIVILEGE-1...8	-	-	-	-	-	-	-	-	-	-	-	-
FT-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
FTAC-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
GUARD-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
HARDWARE-MAINTENANCE	-	X	-	-	-	-	-	-	-	-	-	-
HSMS-ADMINISTRATION	X	-	-	X	-	-	-	-	-	-	-	-
NET-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
NOTIFICATION-ADMINISTRATION	X	-	-	-	-	-	-	-	X	-	-	-
OPERATING	-	-	-	-	-	X	-	-	-	-	X	-
POSIX-ADMINISTRATION	-	-	-	-	-	-	-	X	-	-	-	-
PRINT-SERVICE-ADMINISTRATION	X	-	-	-	-	-	-	-	X	X	-	-
PROP-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
SAT-FILE-EVALUATION	-	-	X	-	-	-	-	-	-	-	-	-
SAT-FILE-MANAGEMENT	-	-	X	-	-	-	-	-	-	-	-	-
SECURITY-ADMINISTRATION	-	-	-	-	-	-	X	-	-	-	-	-
STD-PROCESSING	X	-	-	X	X	X	-	X	X	X	X	X
SUBSYSTEM-MANAGEMENT	X	-	-	-	-	-	-	-	-	-	-	-
SW-MONITOR-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
TAPE-ADMINISTRATION	X	-	-	-	X	-	-	-	-	-	-	-
TAPE-KEY-ADMINISTRATION	-	-	-	-	X	-	-	-	-	-	-	-
TSOS	X	-	-	-	-	-	-	-	-	-	-	-
USER-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-

VIRTUAL-MACHINE-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	-	-
VM2000-ADMINISTRATION	X	-	-	-	-	-	-	-	-	-	X	-

Tabelle 36: Privilegienverteilung nach First Start (Standard-Privilegienverteilung)

[1] Wenn in der Startup-Parameterdatei eine Benutzerkennung ungleich SYSPRIV als Benutzerkennung des Sicherheitsbeauftragten angegeben wurde, so gilt diese Spalte für eben diese Benutzerkennung. Die Benutzerkennung SYSPRIV ist in diesem Fall unter „Andere“ einzuordnen.

[2] die Systemkennungen SYSDUMP, SYSGEN, SYSNAC, SYSSOPT, SYSSNAP, SYSUSER

[3] X bedeutet: Der Benutzerkennung ist das Privileg standardmäßig zugeordnet

[4] - bedeutet: Der Benutzerkennung ist das Privileg standardmäßig nicht zugeordnet

Privilegienverteilung nach Nicht-First-Start mit SECOS

Die Standard-Privilegienverteilung kann nur mit SECOS geändert werden.

Erfolgt nach einem Shutdown in BS2000 ein Startup im System derselben Version mit Kaltstart, Warmstart, SELECTIVE-Start oder ZIP-Start, so ist die Privilegienverteilung dieselbe wie vor dem letzten Shutdown.

Sofern mit dem Startup jedoch gleichzeitig erstmalig ein Versionswechsel aus einer kleineren Version in die aktuelle Version von BS2000 erfolgt (Versionshochstieg), werden neue Privilegien an die Benutzerkennungen verteilt, an die sie auch bei First-Start vergeben worden wären.

Privilegienverteilung nach Pubset-Import

Die Privilegienverteilung nach einem Pubset-Import mit ACTUAL-JOIN=*FIRST entspricht der nach einem First-Start, ein Pubset-Import anderer Art der nach einem Nicht-First-Start.

Erfolgt gleichzeitig eine Rekonstruktion des Benutzerkatalogs, werden für alle die Benutzer die Privilegien rekonstruiert, die aus dem alten Benutzerkatalog (gesichert in der Datei $TSOS.SYSSRPM.BACKUP) in den neuen übernommen werden. Diesen Benutzern sind also nach Abschluss der Rekonstruktion dieselben Privilegien zugeordnet wie zum Zeitpunkt der Sicherung. Welche Benutzerkennungen letztendlich rekonstruiert werden, ist abhängig vom Rekonstruktionstyp.

Gehört der alte Benutzerkatalog zu einem System mit SECOS-Funktionalität und soll er auf einem System ohne SECOS rekonstruiert werden, so werden in den neuen Benutzerkatalog für alle Benutzer die entsprechenden Standardwerte für die Privilegierung eingetragen. Das entspricht einer Privilegienverteilung nach First-Start.

12.5 Betriebsmittel für Benutzer begrenzen

Die Benutzerverwaltung kann für Benutzerkennungen Vorgaben machen, um die Nutzung folgender Betriebsmittel zu begrenzen bzw. deren Mißbrauch zu verhindern:

- Nutzung der Plattenspeicherkapazität auf den Pubsets
- Nutzung des Hauptspeichers im Server
- Nutzung der Leistung des Servers (CPU-Kapazität)

Die benutzerbezogene Zuteilung dieser Betriebsmittel erfolgt jeweils auf Kommandoebene mit ADD-/MODIFY-USER.

Die tatsächliche Verteilungs-Steuerung und -Kontrolle dieser Betriebsmittel im Rahmen der vorgegebenen Werte erfolgt im Betriebssystem (z.B. Verwaltung der Taskkategorien, Steuerung durch PCS, Verwaltung von Job-Streams und Jobklassen,...).

> ¡ Eine analoge Steuerung für Benutzergruppen ist mit SECOS möglich.

Die Nutzung der systemglobalen Betriebsmittel kann für Benutzerkennungen durch die in der folgenden Tabelle dargestellten Kommandos begrenzt werden.

genutzte Betriebsmittel	**Art des Betriebsmittels**	**Kommando**	**Operanden**
Plattenspeicherkapazität auf den Pubsets	Public-Space	`ADD-USER/ MODIFY-USER-ATTRIBUTES`	`PUBSET=..., PUBLIC-SPACE-LIMIT=..., PUBLIC-SPACE-EXCESS=..., FILE-NUMBER-LIMIT=...., JV-NUMBER-LIMIT=..., TEMP-FILE-SPACE=..., DMS-TUNING-RESOURCES=...`
Hauptspeicher im Server	Adressraum	`ADD-USER/ MODIFY-USER-ATTRIBUTES`	`ADDRESS-SPACE-LIMIT=...`
	Hauptspeicher	`ADD-USER/ MODIFY-USER-ATTRIBUTES`	`RESIDENT-PAGES=...`
	Task-(De-) Aktivierung	`ADD-USER/ MODIFY-USER-ATTRIBUTES`	`ACCOUNT-ATTRIBUTES= (MAX-ALLOWED-CATEGORY=..., PRIVILEGE=NO / PARAMETERS (INHIBIT-DEACTIVATION=...))`
Leistung des Servers (CPU-Kapazität)	CPU-Limit	`ADD-USER/ MODIFY-USER-ATTRIBUTES`	`ACCOUNT-ATTRIBUTES= (CPU-LIMIT=..., PRIVILEGE=NO/PAR-(NO-CPU LIMIT=...))`

	zulässige Ablaufpriorität	ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (MAXIMUM-RUN-PRIORITY=...)
	zulässige Taskkategorien	ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (MAX-ALLOWED-CATEGORY=...)
	Scheduling-Recht	ADD-USER/ MODIFY-USER-ATTRIBUTES	ACCOUNT-ATTRIBUTES= (PRIVILEGE=NO/PARAMETERS-(START- IMMEDIATE=...))
	Performance-Maßnahmen	ADD-USER/ MODIFY-USER-ATTRIBUTES	DMS-TUNING-RESOURCES=...

12.6 Erfüllung von Sicherheitsanforderungen durch SECOS

Mit dem Softwareprodukt SECOS wird BS2000 um die Funktionen erweitert, die einen sicheren Betrieb nach den geforderten Sicherheitskriterien gewährleisten.

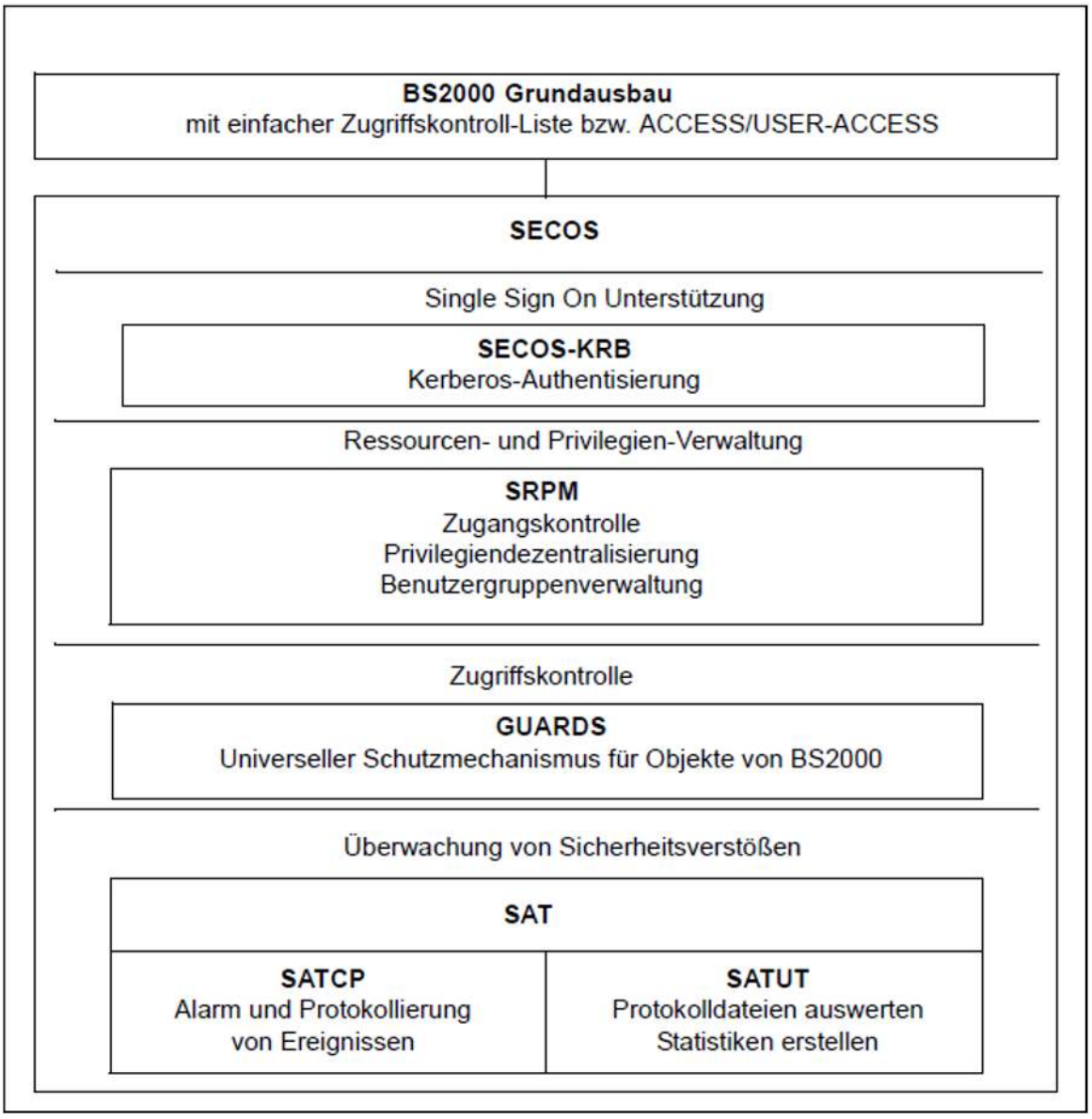


Bild 21: Funktionseinheiten des Sicherheitspakets

Komponenten von SECOS

SRPM (System Resources and Privileges Management) bietet folgende Möglichkeiten:

- Dezentralisierung von Privilegien der Systembetreuung. Damit ist eine Bündelung oder Entbündelung von Aufgaben der Systembetreuung möglich, d.h. es kann eine dem jeweiligen Data Center angemessene Aufgabenverteilung vorgenommen werden. Die Privilegienvergabe erfolgt unter der Benutzerkennung des Sicherheitsbeauftragten (siehe Privileg SECURITY-ADMINISTRATION, "Beschreibung der Privilegien").
- Identifikation und Authentisierung von Benutzern durch erweiterten Zugangsschutz (Trennung und Kontrolle der Zugangswege, Terminal-Sets) und Kennwortschutz (Lebensdauerbegrenzung, minimale Länge, Komplexität).
- Einrichtung von Benutzergruppen, die einen differenzierten Zugriffsschutz ermöglichen. Zugleich können dadurch Verwaltungsaufgaben durch Gruppenverwalter dezentral wahrgenommen werden. Dies bedeutet eine Entlastung der Systembetreuung von trivialen Aufgaben.

GUARDS (General Usable Access Control Administration System) ermöglicht dem Benutzer, den Zugriff zu seinen Datenobjekten von sog. „Zugriffsbedingungen“ abhängig zu machen. Um seine Datenobjekte vor missbräuchlichem Zugriff zu schützen, legt der Benutzer Zugriffsbedingungen fest, die ein Subjekt erfüllen muss, bevor es auf die Datenobjekte zugreifen darf.

GUARDDEF (Default Protection, Standardschutz) dient der Vergabe von Standardattributwerten für Dateien und Jobvariablen. Diese Werte können wahlweise für das Anlegen oder Modifizieren dieser Objekte vorgegeben werden. Die Einstellungen können von der Systembetreuung (TSOS) jeweils pubset-weit und von jedem Benutzer für seine eigenen Objekte unter seiner Benutzerkennung vorgenommen werden.
GUARDDEF nutzt GUARDS zur Ablage der Einstellungen.

Mit **GUARDCOO** wird der Mechanismus zur Miteigentümerschaft (Co-owner Protection, Dateien und Jobvariablen können unter einer anderen Benutzerkennung angelegt und mitverwaltet werden) verfeinert.
Dabei kann die Miteigentümerschaft für unterschiedliche Namensbereiche der Objekte sowohl der Benutzerkennung TSOS entzogen als auch anderen Benutzerkennungen oder den Inhabern bestimmter Privilegien gewährt werden.
GUARDCOO nutzt GUARDS zur Ablage der Einstellungen.

SAT (Security Audit Trail) ermöglicht Beweissicherung durch Protokollierung sicherheitsrelevanter Ereignisse in eine besonders geschützte Datei (SATLOG-Datei). Ausschließlich der Sicherheitsbeauftragte hat das Recht, die SAT-Protokollierung zu aktivieren/deaktivieren und die SAT-Protokollierung für Benutzerkennungen und protokollierbare Ereignisse ein- bzw. auszuschalten.

Mit der Komponente SATUT lassen sich die SATLOG-Dateien von einem besonders autorisierten Benutzer mit den Privilegien SAT-FILE-MANAGEMENT oder SAT-FILE-EVALUATION auswerten. Die Auswertung kann so eingestellt werden, dass z.B. ein Rückblick auf spezielle Verarbeitungsschritte oder auf Aktionen von bestimmten Benutzerkennungen ausgegeben wird. Auf diese Weise lässt sich z.B. eine missbräuchliche Benutzung des Systems oder der unerlaubte Zugriff auf gesicherte Daten entdecken.

Für SSO (Single Sign On) im BS2000 wird Kerberos eingesetzt.
Mit **SECOS-KRB** (Kerberos-Authentisierung) wird SSO im BS2000 realisiert.
Kerberos ist ein SSO-Sicherheitssystem (Single Sign On), das auf kryptographischen Verschlüsselungsverfahren basiert. Bei einer Authentisierung mit Kerberos werden keine Kennwörter im Klartext über das Netzwerk gesendet. Dadurch wird das Abfangen von Kennwörtern im Netzwerk verhindert.

Das Softwareprodukt SECOS, seine einzelnen Komponenten und deren Installation und Integration in BS2000 sind in den SECOS-Handbüchern [46] und [47] ausführlich beschrieben.

13 Datensicherung

Dieses Kapitel stellt Kriterien zur Auswahl von Sicherungskonzepten in BS2000, Datensicherungsfunktionen in BS2000 sowie Softwareprodukte zur Datensicherung (ARCHIVE, HSMS und FDDRL) vor und informiert über Möglichkeiten zur Rekonstruktion von Dateien und Datenträgern.

Zum Schluss des Kapitels werden die von BS2000 unterstützten Archivsysteme mit dem sie steuernden Softwareprodukt ROBAR kurz beschrieben.

13.1 Kriterien zur Auswahl von Sicherungskonzepten

Jedes Data Center widmet dem Thema Datensicherung besondere Aufmerksamkeit. Die Notwendigkeit für ein lückenloses Sicherungskonzept ist begründet in

- der Forderung nach hoher Datenverfügbarkeit
- der Möglichkeit, auf Datenbestände zuzugreifen, die aus Kapazitätsgründen nicht auf gemeinschaftlichen Datenträgern gespeichert sind
- der Verbesserung der Antwortzeiten, indem sowohl gemeinschaftliche als auch private Datenträger reorganisiert werden
- dem Wunsch bzw. der Notwendigkeit, Datenbestände in ein anderes Data Center zu transportieren

Um diesen Forderungen gerecht zu werden, muss in jedem Data Center eine vorbeugende und regelmäßige Sicherung aller sicherungswürdigen Daten durchgeführt werden.

Auswahl der Daten

Die Auswahl der Daten, die bei einer Sicherung erfasst werden sollen, orientiert sich i.A. an ihrer Sicherungswürdigkeit. So kann unterschieden werden zwischen Produktionsdaten, die für die laufende Produktion erforderlich sind und einer steten Änderung unterliegen, und reinen Testdaten, die jederzeit reproduzierbar sind. Zudem können die Systemdaten, die sich im laufenden Betrieb normalerweise nicht ändern, von regelmäßigen Sicherungen ausgeklammert werden. Hier genügt eine vollständige Sicherungskopie, um im Rekonstruktionsfall den aktuellen Stand wieder herzustellen.

Zeitpunkt und Häufigkeit der Sicherung

Das Sicherungsverfahren muss so konzipiert sein, dass es sowohl der Forderung nach Datensicherheit als auch der Forderung nach Verfügbarkeit der Anwendungen Rechnung trägt. So sollten z.B. langwährende Sicherungen – logische oder physikalische Gesamtsicherungen – gezielt in lastarme Zeiten verlegt werden, um den Ausfall an nutzbarer Rechenzeit zu minimieren.
Das Sicherungsverfahren ist ferner so zu wählen, dass redundante Sicherungsbestände vermieden werden. Das heißt, dass jeder Änderungsstand einer Datei nur einmal erfasst werden sollte.

Sicherungsarten

- **Gesamtsicherung**

  Bei einer Gesamtsicherung werden alle durch sonstige Auswahlkriterien bestimmten Dateien erfasst, die zum Zeitpunkt der Sicherung geschlossen sind. Diese Dateien werden vollständig gesichert, unabhängig davon, ob sie sich seit der letzten Sicherung geändert haben oder nicht.

  Die Gesamtsicherung wird oft auch „Vollsicherung“ genannt.
  Jede physikalische Sicherung ist stets eine Gesamtsicherung.

- **Differenzsicherung**

  Bei der Differenzsicherung werden nur die Dateien gesichert, die sich seit der letzten Sicherung geändert haben oder die neu erstellt wurden. Diese Dateien werden vollständig gesichert. Voraussetzung ist, dass mit der gleichen Directory-Datei gearbeitet wird, in der die entsprechenden Informationen bezüglich der Dateiversion hinterlegt sind.

- **partielle Sicherung**

  Bei der partiellen Sicherung (einer besonderen Form der Differenzsicherung) wird für ausgewählte Dateien nur eine Sicherung der PAM-Seiten durchgeführt, die nach der letzten vollständigen Sicherung der Datei verändert wurden. Die anderen Dateien werden bei dieser Sicherung überhaupt nicht gesichert.
  Zur Rekonstruktion einer partiell gesicherten Datei sind jeweils die letzte partielle Sicherung und die letzte Gesamtsicherung erforderlich.

  Die partielle Sicherung wird oft auch „Teilsicherung“ genannt.

Umfang der Sicherung

Den Umfang der Sicherung bestimmt die Systembetreuung abhängig von den Kriterien Datenbestand, Datenkonsistenz, Server-Last und Konfiguration.

- **Datenbestand**

  Die Anzahl und Größe der zu verwaltenden und zu sichernden Daten beeinflusst den Umfang der Sicherung.
  Bei einem geringen Datenbestand kann die Systembetreuung auf Teil- und Differenzsicherungen verzichten und regelmäßig eine logische oder physikalische Gesamtsicherung des Systems vornehmen. Dabei werden zwar die unveränderten Daten auch immer wieder gesichert, das gesamte „Datenpaket“ ist aber ständig konsistent und muss bei Bedarf nicht aus verschiedenen Sicherungsständen rekonstruiert werden.
  Bei umfangreichen Datenbeständen muss die Systembetreuung ein ausgefeilteres Konzept zur Datensicherung erarbeiten. Hierbei kann eine Sicherung aller Dateien sukzessive erfolgen, d.h. durch Aneinanderreihung mehrerer partieller und Differenzsicherungen, und in z.B. wöchentlichen Rhythmen eine logische oder physikalische Gesamtsicherung durchgeführt werden.
  Dateien können verschiedenen Backup-Levels oder Management-Klassen zugeordnet werden. Diese Zuordnungen können dann von der Systembetreuung zur Eingrenzung der Sicherung verwendet werden.

- **Datenkonsistenz**

  Der Datenbestand muss im Hinblick auf den Umfang der Sicherung auch einer qualitativen Überprüfung unterzogen werden.

  Qualitative Merkmale der System- und Benutzerdateien sind z.B.:

  - Anzahl der Zugriffe
  - Häufigkeit der Änderungen
  - Umfang der Änderungen
  - Austauschhäufigkeit von Programmversionen
  - Zuordnung zu einem Backup-Level oder einer Management-Klasse

  Beschränkt sich der Datenbestand im Wesentlichen auf konstante, stabile Versionen, die selten oder nicht verändert werden, kann der aktuelle Stand auch aus einer länger zurückliegenden Sicherungsversion schnell rekonstruiert werden. Bei häufig wechselnden Programmversionen und rasch wechselndem Dateienbestand muss die Systembetreuung einen Datenverlust durch ein entsprechend abgestuftes Sicherungskonzept verhindern.

- **Server-Last**

  Durch eine genaue Analyse der Server-Last kann die Systembetreuung nicht nur Zeitpunkt und Häufigkeit der Sicherungen bestimmen, sondern auch Rückschlüsse auf den Umfang der zu sichernden Daten ziehen.
  Das Sicherungskonzept könnte idealerweise so gewählt werden, dass in einer relativ lastfreien Periode eine umfangreiche Vollsicherung, in einer Periode mittlerer Last partielle und Differenzsicherungen durchgeführt werden, unter Voll-Last jedoch auf eine Sicherung gänzlich verzichtet wird.

- **Konfiguration**

  Die Konfiguration des Servers und damit auch die für die Datensicherung verfügbare Hardware beeinflusst die Erstellung des Sicherungskonzeptes und den Umfang der einzelnen Sicherungen.
  Stehen der Systembetreuung genügend periphere Geräte zur Verfügung, kann dadurch die Sicherungsdauer verkürzt und die Menge der Sicherungsdaten erhöht werden.
  - Durch Nutzung der Subtask-Funktion der Dienstprogramme ARCHIVE und FDDRL können Teilaufträge auf verschiedene Geräte gelenkt und die Sicherungen parallel abgewickelt werden.
  - Die Verteilung der Nutzerkreise auf einzelne Pubsets (MPVS) erleichtert den gezielten Zugriff auf Teilmengen der zu sichernden Daten.
  - Bereits bei der Generierung des Systems kann die Systembetreuung durch eine performance-günstige Konstellation „Peripherie zu Kanäle zu IOP“ die Voraussetzung für die beschleunigte Sicherung auch umfangreicher Datenmengen schaffen.

Ausfallzeit durch Sicherung

Bei der Sicherung der Daten einer Anwendung ist die Anwendung selber für eine bestimmte Zeit nicht verfügbar. Diese Ausfallzeit wird vor allem durch Umfang und Zeitdauer der Sicherung bestimmt. Durch die Sicherung von abtrennbaren Kopien (z.B. auf Spiegelplatten) kann die Ausfallzeit auf das Erstellen der Kopie verkürzt werden. Die Anwendung selber kann nach der kurzen Kopierphase auf den Originaldateien weiterarbeiten, während die zeitaufwendigere Sicherung von der abgetrennten Kopie erfolgt.

Logische und physikalische Sicherung

Bei einer **logischen Datensicherung** werden einzelne Dateien und Jobvariablen von einem oder mehreren Datenträgern gelesen und zusammenhängend, also in logischen Einheiten, auf andere Datenträger geschrieben.

Für die logische Datensicherung stehen in BS2000 die Softwareprodukte ARCHIVE (siehe Handbuch „ARCHIVE“ [3]) und HSMS (siehe Handbuch „HSMS“ [24]) zur Verfügung.

HSMS bietet dem Benutzer die vier Grundfunktionen:

- Datensicherung (Backup)
- Langzeitarchivierung (Archival)
- Verdrängung (Migration)
- Datentransfer (Export/Import)

HSMS setzt dabei auf dem Softwareprodukt ARCHIVE auf. Die meisten Funktionen, die früher über ARCHIVE aufgerufen wurden, stehen in HSMS kompatibel zur Verfügung.

Zur Verkürzung der Ausfallzeit wird in HSMS die Backup-Sicherung mit CCOPY (Concurrent Copy) von einer Kopie geboten.

Bei einer **physikalischen Datensicherung** werden keine einzelnen Dateien gesichert, sondern ganze Datenträger gemäß ihrer physikalischen Struktur. Dabei werden sämtliche Dateien eines Datenträgers, einschließlich der Datenträgerkennsätze, blockweise in der physikalischen Reihenfolge auf einen zweiten Datenträger geschrieben. Dieser ist dann identisch mit dem Originaldatenträger.
Für die physikalische Datensicherung steht in BS2000 das Softwareprodukt FDDRL (siehe Handbuch „FDDRL“ [22]) zur Verfügung.

Eine Mischung von physikalischer und logischer Sicherung stellt die **Snapset-Sicherung** dar: Sie ist eine Pubset-Sicherung, bei der für jede Pubset-Platte eine entsprechende Kopie auf einer Snap-Unit erstellt wird. Aus dieser Pubset-Sicherung können einzelne Dateien und Jobvariablen als logische Einheiten gelesen werden. Abhängig vom Storage-System kann auch der gesamte Pubset rekonstruiert werden. Die Funktionen für die Snapset-Sicherung und den Zugriff auf die gesicherten Daten (auf Ebene von Pubset, Datei und Job-variable) stehen in BS2000 zur Verfügung.

Verschlüsselte Dateien werden bei allen Sicherungsarten in verschlüsselter Form abgespeichert.

Die grundsätzlichen Unterschiede bei der Sicherung von Datenträgern (physikalische Sicherung), der Sicherung von Dateien und Jobvariablen (logische Sicherung) und der Snapset-Sicherung zeigt folgende Übersicht:

	Logische Sicherung	**Physikalische Sicherung**	**Snapset-Sicherung**
Was wird gesichert?	Dateien, Katalogeinträge, Jobvariablen	komplette Datenträger, d. h. private und gemeinschaftliche Platten	kompletter Pubset (plattenweise)
Wer sichert?	Systembetreuung bzw. HSMS-Adminstration: sämtliche Benutzer- und Systemdateien; Benutzer: nur seine eigenen Dateien	Systembetreuung bzw. HSMS-Adminstration	Systembetreuung bzw. HSMS-Adminstration
Wann wird gesichert?	regelmäßig bei gestoppter Anwendung	regelmäßig bei excatiertem Pubset	regelmäßig im laufenden Betrieb (importierter Pubset)
Hilfsmittel	Dienstprogramme ARCHIVE und HSMS	Dienstprogramm FDDRL	Snapset-Kommandos in BS2000

Besonderheiten	• Voll- und Differenzsicherung; komfortable Methoden zur Objektauswahl • Verkürzung der Ausfallzeit durch Sichern von Clone-Units	• Sichern nur der belegten Blöcke • Verkürzung der Ausfallzeit durch Sichern von Clone-Units	• Zugriff auf gesicherte Dateien, Katalogeinträge, Jobvariablen (zur Information bzw. Rekonstruktion) ist „online“ möglich für alle Benutzer gemäß ihrer DVS-Zugriffsrechte • für Systembetreuung Rekonstruktion des kompletten Pubsets • Keine Ausfallzeit • Sicherung/Rekonstruktion im Minutenzeitraum
Sicherungsdatenträger	MBK, Platte und Net-Storage-Volume	MBK und Platte	Snap-Units (im Storage-System)

Tabelle 37: Logische und physikalische Sicherung

13.2 Datensicherung mit Snapsets

BS2000 bietet eine integrierte Lösung zur Sicherung eines Pubsets im laufenden Betrieb auf sogenannte Snap-Units eines externen Storage-Systems. Diese Funktion wird unter dem Namen **Snapset** angeboten. Sie steht für die Storage-Systeme ETERNUS DX/AF von FUJITSU zur Verfügung.

BS2000 unterstützt auch die Datensicherung durch Replikationsfunktionen externer Storage-Systeme. Siehe dazu das Handbuch „SHC-OSD“ [48].

Prinzip der Snapsets

Ein Snapset ist die unmittelbar verfügbare Sicherung eines Pubsets auf sogenannte Snap-Units eines Storage-Systems. Ein Snapset dient der Wiederherstellung verlorener Daten (z.B. nach versehentlichem Löschen). Auch der komplette Pubset kann auf den Stand zum Zeitpunkt der Erstellung des Snapsets zurückgesetzt werden. Es können sowohl SM- als auch SF-Pubsets gesichert werden. Snapsets werden im laufenden Pubset-Betrieb von der Systembetreuung oder vom HSMS-Administrator erzeugt und später bei Bedarf wieder gelöscht. Für einen Pubset können maximal 52 Snapsets existieren (sofern diese Maximalzahl vom Storage-System unterstützt wird).

Innerhalb des Storage-Systems nutzen Snap-Units ein blockweises Differenzverfahren und benötigen nur für die vom Original abweichenden Blöcke einen eigenen Speicherplatz.

Dadurch wird die Sicherung auf Platte mit Snap-Technik auch wirtschaftlich vertretbar:
Wenn z.B. von einem Pubset der Kapazität 100 GB an einem Tag 10 Sicherungen erstellt werden, dann würde mit einer Clone-Technik 1 TB zusätzlicher Plattenspeicherplatz benötigt werden. Bei der Nutzung von Snapsets hingegen hängt der benötigte zusätzliche Plattenspeicherplatz vom Änderungsvolumen des Pubsets ab. Bei einem Änderungsvolumen von 20% pro Tag entspricht dies in obigem Beispiel einem zusätzlichen Plattenspeicherplatz von nur 0,2 TB (20% * 10 Sicherungen * 100 GB).

Snapsets stehen ausschließlich für Restore-Zwecke zur Verfügung (d.h. nur Lesezugriff):
Sie werden bei der Erzeugung automatisch in Betrieb genommen. Bereits vorhandene Snapsets werden in der Regel beim Importieren des zugehörigen Pubsets in Betrieb genommen.

Mit folgenden Funktionen können Benutzer auf Snapsets zugreifen:

- Informationen über vorhandene Snapsets (d.h. Pubset-Sicherungen) einholen
- Informationen über rekonstruierbare Dateien und Jobvariablen einholen
- Dateien und Jobvariablen rekonstruieren
- Pubset rekonstruieren (nur privilegierter Benutzer)

Folgende Softwareprodukte ermöglichen ebenfalls den Zugriff zu Snapsets:

- LMS ermöglicht das Selektieren und Kopieren von Elementen aus PLAM-Bibliotheken auf Snapsets (siehe Handbuch „LMS“ [29]).
- Mit HSMS können Dateien und Jobvariablen von einem Snapset in ein Backup-Archiv übernommen werden (siehe "Sicherung vom Snapset in ein Backup-Archiv" im Kapitel "Rekonstruktion von Dateien und Jobvariablen" bzw. Handbuch „HSMS“ [24]).

Die Snapset-Funktionen sind im Shared-Pubset-Betrieb in allen beteiligten Systemen verfügbar. Die Voraussetzungen dazu (BS2000-Versionen der Pubset-Sharer und von SHC-OSD) sind in den aktuellen BS2000-Freigabemitteilungen beschrieben.

In BS2000 können Sie bis zu 52 Snapsets verwenden. Außerdem werden zum Zugriff auf Snapsets Programmschnittstellen angeboten, die funktionell den zuvor erwähnten Kommando-Schnittstellen entsprechen (siehe Handbuch „DVS-Makros“ [20]).

VM2000 unterstützt die Nutzung von Snapsets durch die Gastsysteme (siehe Handbuch „VM2000“ [60]).

Einsatzbereich von Snapsets

Snapsets dienen kurzfristig als Backup bei Verlust von Dateien bzw. Jobvariablen eines Pubsets. Auch ein gesamter Pubset kann mit dem Kommando RESTORE-PUBSET-FROM-SNAPSET schnell auf einen bestimmten Stand zurückgesetzt werden. Für Langzeitsicherungen oder Datentransfer sind Snapsets nicht geeignet. Die bei Snapsets verwendeten Snap-Units sind normalerweise lokale Kopien innerhalb eines Storage-Systems und bieten daher auch keinen Schutz bei Ausfall der Storage-Systems. Es wird daher dringend empfohlen, auch Snapsets mit HSMS zu sichern (siehe "Sicherung vom Snapset in ein Backup-Archiv" im Kapitel "Rekonstruktion von Dateien und Jobvariablen").

Privatplatten (bzw. Dateien auf Privatplatten) können nicht auf Snapsets gesichert werden.

Der Betrieb von Snapsets kann mit einer gleichzeitigen Nutzung von Plattenkopien auf Basis von lokalen Spiegeln im gleichen Pubset kombiniert werden, sofern dies im jeweiligen Storage-System möglich ist.
Ein Mischbetrieb von Snapsets (/CREATE-SNAPSET) und Snapshots (/START-SNAP-SESSION, /ACTIVATE-SNAP) wird nicht unterstützt .

Diese anderen Nutzungsarten sind in den Handbüchern „HSMS“ [24] und „SHC-OSD“ [48] beschrieben.

Einschränkungen:

- Die Rekonstruktion des Pubsets von einem Snapset ist nicht möglich, wenn für diesen Pubset noch eine Clone-Session aktiv ist.

Schnittstellenübersicht zur Arbeit mit Snapsets

Alle Kommandos für den Snapset-Betrieb, die Sicherung auf Snapset bzw. die Rekonstruktion von Snapsets werden im SDF-Anwendungsbereich „SNAPSET“ angeboten. Diese Kommandos sind im Handbuch „Kommandos“ [27] beschrieben.

Kommando	Funktion
ADAPT-SNAPSET-ACCESS [1]	passt den Zugriff auf Snapsets bei remote Replikation an
CHECK-SNAPSET-CONFIGURATION [2]	prüft die Snapset-Konfiguration und aktiviert vorhandene Snapsets
CREATE-SNAPSET [2]	erzeugt einen Snapset
DELETE-SNAPSET [2]	löscht einen Snapset
LIST-FILE-FROM-SNAPSET [3]	informiert über Dateien auf einem Snapset
LIST-JV-FROM-SNAPSET [3]	informiert über Jobvariablen auf einem Snapset
RESTORE-FILE-FROM-SNAPSET [3]	rekonstruiert Dateien auf Basis eines Snapsets
RESTORE-JV-FROM-SNAPSET [3]	rekonstruiert Jobvariablen auf Basis eines Snapsets
RESTORE-PUBSET-FROM-SNAPSET [2]	rekonstruiert einen kompletten Pubset auf Basis eines Snapsets
SET-PUBSET-ATTRIBUTES [2]	bestimmt die maximal erlaubte Anzahl von Snapsets für den Pubset (Operand SNAPSET-LIMIT)
SET-SNAPSET-PARAMETER [2]	vereinbart die Snapset-Umgebung
SHOW-PUBSET-ATTRIBUTES	informiert u.a. über das Snapset-Limit
SHOW-SNAPSET-CONFIGURATION	informiert über vorhandene Snapsets und die Snapset-Konfiguration

Tabelle 38: Schnittstellenübersicht zur Arbeit mit Snapsets

[1] Privileg TSOS erforderlich

[2] Privileg TSOS oder HSMS-ADMINISTRATION erforderlich

[3] Für diese Kommandos gibt es auch eine Programmschnittstelle, siehe Handbuch „DVS-Makros“ [20].

13.2.1 Snapset-Betrieb vorbereiten

Den Snapset-Betrieb für einen Pubset aufzunehmen bedeutet, dass für den Pubset Kopien (Sicherungen) auf Snapsets erstellt werden, die für die Rekonstruktion von Dateien bzw. Jobvariablen oder des gesamten Pubsets zur Verfügung stehen.

Begriffe

Snap-Unit

Die lokalen Replikationsfunktionen der Storage-Systeme erstellen einen „Snapshot“ einer logischen Unit (ggf. auch mehrere). Der Snapshot, Snap-Unit genannt, ist eine logische Kopie der Original-Unit zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“): Während die Daten auf der Original-Unit verändert werden, behält die Snap-Unit den Stand der Daten zum Zeitpunkt der Snapshot-Erstellung.

Im BS2000 werden die Snap-Units als normale Plattengeräte (Typ D3435) generiert mit gleicher Kapazität wie die jeweilige Original-Unit.

Für den Snapset-Betrieb müssen die dafür geplanten Snap-Units auf Thin Devices (TDEV) oder Flex Volumes (FDEV) vorab mit dem Dienstprogramm VOLIN (siehe „Dienstprogramme“ [15]) initialisiert werden. Als VSN für diese Volumes wird die Sondernotation S#<mn> erwartet, z.B. S#5234. <mn> ist dabei die Geräte-Mnemonic.

> ¡ Aufgrund der Flexibilität wird im Storage-System die Nutzung der Snap-Units als Thin Devices (TDEV) oder Flex Volumes (FDEV), also im BS2000 die Nutzung von S#<mn> Volumes empfohlen.

Für den Snapset-Betrieb können auch speziell konfigurierte Units, sog. Snap-Units oder auch Snap-Device-Volumes (SDVs) verwendet werden. Diese werden in der Regel von einem qualifizierten Techniker direkt am Storage-System eingerichtet. Original-Unit und Snap-Unit müssen dabei im selben Storage-System liegen.

Save Pool

Der tatsächlich benötigte Speicherplatz für die zu sichernden Originaldaten wird im Falle von Snap-Device-Volumes aus einem sogenannten Save-Pool bereitgestellt. Im Speichersystem werden Save Pools mit entsprechender Kapazität konfiguriert.

Die Realisierung von Save-Pools ist abhängig von den verwendeten Snap-Units:

- im S#-Volume Betrieb
  Hier gibt es keinen separaten Save-Pool.
  Der benötigte Speicherplatz wird für die zugeordneten Snap-Units bedient, die der Überlaufkontrolle des jeweiligen „Thin Provisioning Pool" unterliegen.
- im SDV-Betrieb
  Hier gibt es einen einzigen Save-Pool für alle Snapsets.
  Ein Save-Pool wird im Storage-System mit geeigneter Kapazität konfiguriert. Daraus wird der tatsächlich benötigte Speicherplatz für die zu sichernden Originaldaten bereitgestellt.
  Bei Überlauf des Save-Pools sind ***alle*** Snapsets unbrauchbar, die hier Speicherplatz belegen. Die Funktionen von SHC-OSD zur Überwachung von Save Pools sollten deshalb hier dringend genutzt werden.

Hinweise zum Betrieb mit Save-Pool:

Der tatsächlich benötigte Speicherplatz (für die zu sichernden Originaldaten) wird dabei in der Regel aus einem speziell dafür vorgehaltenen Speicherbereich auf dem Storage-System, dem sogenannten Save-Pool, bestritten. Die benötigte Kapazität eines Save-Pools ist von folgenden Faktoren abhängig:

- Datenvolumen der Anwendung für die Snapsets erzeugt wurden
- Anzahl der Snap-Sessions pro Original
- Änderungsvolumen auf dem Original (und den zugehörigen Snap-Units)

Beim Löschen eines Snapsets wird der jeweilige Bereich im Save-Pool wieder freigegeben.

Save-Pool / Thin-Pool Monitoring

Zur Überwachung des Füllgrades eines Save-Pools bzw. Thin-Pools stellt SHC-OSD entsprechende Monitoring-Funktionen zur Verfügung. Bei Erreichen oder Ändern gewisser Grenzwerte werden Warnmeldungen an der Konsole ausgegeben.
Die Systemadministration sollte in jedem Fall angemessen darauf reagieren und wieder Platz auf dem Save-Pool schaffen bzw. diesen bei Bedarf auch erweitern. Die Grenzwerte selbst werden am Storage-System oder auch in der SHC-OSD Parameterdatei eingestellt.

> **ACHTUNG!**
> Bei einem Überlauf gehen die bestehenden Snapsets verloren.
> Konkret gehen die jeweiligen Snap-Sessions in den Zustand „FAILED“ und können anschließend nur noch gelöscht werden.

Beispiel für die Vorbereitungen

Bevor der Snapset-Betrieb für einen Pubset aufgenommen werden kann, sind entsprechende Vorbereitungen zu treffen. Die Vorgehensweise soll an folgendem Beispiel veranschaulicht werden.

- Ausgangssituation und Bedarfsplanung

  Der Pubset A besteht aus 8 Platten (`PUBA00` bis `PUBA07`). Er soll an jedem Arbeitstag zweimal (z.B. mittags und abends) gesichert werden. Der Zugriff auf die Sicherungen der letzten 6 Arbeitstage soll gewährleistet sein.

  Für 2 x 6 Pubset-Sicherungen müssen dann 12 Snapsets zur Verfügung stehen. Pro Snapset wird die gleiche Anzahl an Snap-Units benötigt wie die Anzahl der Platten in Pubset A. Insgesamt werden 8 x 12 = 96 Snap-Units benötigt.

- Snap-Units im Storage-System bereitstellen

  Die benötigten 96 Snap-Units sind im selben Storage-System wie die Original-Platten des Pubset A mit gleichem Typ und gleicher Größe bereitzustellen.
  Wenn bereits eine Pubset-Erweiterung abzusehen ist, sollten auch dafür schon zusätzliche Snap-Units in ausreichender Zahl eingerichtet werden.

- Snap-Units in BS2000 konfigurieren

  Alle 96 Snap-Units müssen wie normale Platten (Typ D3435) in die Hardware-Generierung der beteiligten BS2000-Systeme aufgenommen werden, praktisch also in alle Hardware-Generierungen, in denen auch die Original-Platten des Pubsets A eingetragen sind (siehe Handbuch „Systeminstallation“ [55]).

  An SU /390 können die Platten auch im laufenden Betrieb in die aktive I/O-Konfiguration aufgenommen werden. Die Aufnahme von emulierten Platten D3435 in die X2000-Konfiguration einer SU x86 ist im Handbuch „Bedienen und Verwalten“ [57] beschrieben.

Hinweis für VM2000

Im Storage-System eingerichtete Snap-Units werden von VM2000 auch als Snap-Units erkannt. Sie werden automatisch beim Pubset-Import zugeschaltet und unter VM2000 implizit der VM zugeordnet, wenn die VM das Privileg AUTO-SNAP-ASSIGNMENT besitzt. Eine VM erhält beim Initialisieren mit dem VM2000-Kommando CREATE-VM standardmäßig dieses Privileg. Es erlaubt dem Gastsystem auf der VM, sich die Snap-Units eines Snapsets implizit zuzuordnen, ohne dass VM und Gerät für die implizite Gerätezuordnung vorbereitet sind (d.h. das VM-Privileg und das Geräte-Attribut ASSIGN-BY-GUEST sind in diesem Fall nicht notwendig).

Ausnahme

Werden S#<mn> Volumes als Snap-Units verwendet, dann werden diese von VM2000 nicht als Snap-Units erkannt und auch nicht angezeigt. Sie können trotzdem einer VM mit dem VM-Privileg AUTO-SNAP-ASSIGNMENT implizit zugeordnet werden.

- Snapset-Limit setzen und ggf. Save Pool zuweisen (in BS2000)

  Das Snapset-Limit ist im SVL der Pubres eingetragen. Es legt fest, wie viele Snapsets maximal für den Pubset zulässig sind. Bei einem Pubset ohne Snapset-Betrieb ist das Snapset-Limit 0.
  Da für den Pubset A maximal 12 Snapsets angelegt werden sollen, setzt die Systembetreuung das Snapset-Limit auf 12:

  ```
  /SET-PUBSET-ATTRIBUTES PUBSET=A,SNAPSET-LIMIT=12
  ```

Snapset-Betrieb bei Katastrophenschutz mit remote Replikation

Wird der Pubset A wegen Katastrophenschutz in einem lokalen Storage-System **und** in einem remote Storage-System betrieben, ist für den Snapset-Betrieb Folgendes zu beachten:

- Im Normalfall werden die Snapsets **nur** im lokalen Storage-System eingerichtet. Sie stehen bei Ausfall des lokalen Storage-Systems nicht mehr zur Verfügung.
- Sollen die Snapsets für den Katastrophenfall auch im remote Storage-System zur Verfügung stehen, müssen sie in **beiden** Storage-Systemen erzeugt werden. Dazu muss im remote Storage-System die gleiche Anzahl Snap-Units wie im lokalen Storage-System bereitgestellt und und in die Hardware-Generierung der beteiligten BS2000-Systeme aufgenommen werden.

  Zusätzlich muss die Systembetreuung die Verarbeitungsumgebung für die Snapsets entsprechend einstellen:

  ```
  /SET-SNAPSET-PARAMETER PUBSET=A,REMOTE-COPY=*YES
  ```

  Bei einer Umschaltung des Plattenzugriffs zwischen Source- und Target-Controller wird der Zugriff auf die dem Pubset zugeordneten Snapsets nicht automatisch mit umgeschaltet.
  Das Kommando ADAPT-SNAPSET-ACCESS stellt sicher, dass die zugeordneten Snapsets nach einer solchen Umschaltung weiter zur Verfügung stehen, ohne dass der Pubset exportiert werden muss:

  ```
  /ADAPT-SNAPSET-ACCESS PUBSET=A
  ```

13.2.2 Sicherungsbetrieb mit Snapsets

Sicherungsbetrieb mit Snapsets

Sind die Vorbereitungen abgeschlossen, kann der Snapset-Betrieb (im Beispiel für den Pubset A) aufgenommen werden, d.h. die Systembetreuung bzw. die HSMS-Administration kann zu den geplanten Zeitpunkten Pubset-Kopien (=Snapsets) erstellen.

Beispiel

Pubset A soll an jedem Arbeitstag zweimal (mittags und abends) gesichert werden.

Snapset erstellen

Jeweils zu dem gewünschten Sicherungszeitpunkt erstellt die Systembetreuung bzw. die HSMS-Administration mit dem Kommando CREATE-SNAPSET einen Snapset:

```
/CREATE-SNAPSET PUBSET=A
```

Das Kommando kann im Shared-Pubset-Betrieb an allen beteiligten Systemen eingegeben werden.
Zu einem Zeitpunkt kann für den gleichen Pubset nur ein Kommando CREATE-SNAPSET gegeben werden. Parallele Aufrufe werden serialisiert.
Als Voreinstellung des Kommandos wird bei erreichtem Snapset-Limit implizit der älteste Snapset gelöscht. Optional kann im Operanden DELETE-EARLIEST auch vereinbart
werden, dass in jedem Fall der älteste Snapset implizit gelöscht wird oder dass in keinem Fall implizit gelöscht wird. Der neu erzeugte Snapset wird automatisch in Betrieb genommen und steht damit allen Benutzern für Restore-Zwecke zur Verfügung.

Snapset-Identifikation

Jeder neu erzeugte Snapset erhält eine eindeutige Snapset-Identifikation. Für die maximal 52 möglichen Snapsets werden fortlaufend in alphabetischer Reihenfolge zuerst die Kleinbuchstaben „a“ bis „z“, dann die Großbuchstaben „A“ bis „Z“ vergeben. Nach Erreichen von „Z“ beginnt die Vergabe wieder mit „a“ (die zeitliche Reihenfolge der Snapset-Erzeugung entspricht deshalb nicht immer der alphabetischen Reihenfolge der Snapset-Ids). An der Kommandoschnittstelle muss die Groß-/Kleinschreibung beachtet werden.

Jede Snap-Unit erhält eine VSN, die aus der VSN der Originalplatte und aus der Snapset-Identifikation gebildet wird.

Beispiele zur Verdeutlichung der Abbildungsregel

VSN in PUB-Notation und Snapset-Id als Kleinbuchstabe, z.B. „a“: PUBX12 --> PaaX12

VSN in PUB-Notation und Snapset-Id als Großbuchstabe z.B. „A“: PUBX12 --> aaBX12

VSN in Punkt-Notation und Snapset-Id als Kleinbuchstabe, z.B. „a“:
AB.123 --> ABa123; ABC.12 --> ABCa12; ABCD.1 --> ABCDa1

VSN in Punkt-Notation und Snapset-Id als Großbuchstabe, z.B. „A“:
AB.123 --> AB123a; ABC.12 --> aABC12; ABCD.1 --> AaBCD1

Verwaltung und Inbetriebnahme von Snapsets

Jeder neu erzeugte Snapset besitzt eine eindeutige Snapset-Identifikation (siehe "Snapset-Identifikation"). Die Beschreibungen aller Snapsets eines Pubset werden im Snapset-Katalog hinterlegt und verwaltet. Der Snapset-Katalog wird im zugehörigen Pubset unter dem Namen $TSOS.SYSCAT.SNAPSET abgelegt und enthält folgende Informationen:

- die Verarbeitungsumgebung für die Snapsets des Pubsets wie die Nutzung von remote Replikation und Angaben zum Save Pool
- Beschreibung der einzelnen Snapsets mit Angaben wie Snapset-Identifikation, Erstellungsdatum und MNs der Snap-Units im lokalen Storage-System und ggf. auch der Snap-Units im remote Storage-System.

Der Snapset-Katalog wird, falls er noch nicht existiert, mit Erstellen des Snapsets eingerichtet. Der Snapset-Katalog liegt bei einem SM-Pubset auf der Volres des Control-Volume-Sets und bei einem SF-Pubset auf der zugehörigen Pubres.

Snapsets in und außer Betrieb nehmen

Bestehende Snapsets werden in der Regel beim Importieren des zugehörigen Pubsets in Betrieb genommen.

Vorhandene Snapsets werden nicht in Betrieb genommen, wenn das Subsystem SHC-OSD zum Zeitpunkt des Pubset-Imports noch nicht aktiv ist (insbesondere werden vorhandene Snapsets des Home-Pubsets beim Hochfahren des Systems nicht automatisch in Betrieb genommen). Sobald SHC-OSD aktiv ist, werden die Snapsets bei Aufruf des Kommandos SHOW-SNAPSET-CONFIGURATION nachträglich aktiviert.

Wenn Snapsets beim Import wegen Hardware- oder Konfigurationsproblemen nicht in Betrieb genommen werden können, dann verbleiben diese zunächst im Zustand „nicht zugreifbar“. Nach Fehlerbehebung können diese Snapsets mit dem Kommando CHECK-SNAPSET-CONFIGURATION wieder zugreifbar gemacht werden.

Beim Neuanlegen mit dem Kommando CREATE-SNAPSET wird ein Snapset automatisch in Betrieb genommen.

Dabei werden folgende Aktionen durchgeführt:

- Snapset-Katalog öffnen (durch GCF)
- alle Snap-Units zuweisen (ATTACH)
  In einer Konfiguration mit remote Replikation werden die „passenden“ Snap-Units zugewiesen. Im Normalzustand bei aktiver remote Replikation sind das die Snap-Units im lokalen Storage-System. Bei getrennten Spiegeln erfolgt die Auswahl anhand des importierten Original-Pubsets:
  - Beim Pubset am lokalen Storage-System werden nur die Snap-Units am lokalen Storage-System verwendet.
  - Beim Pubset am remote Storage-System werden nur die Snap-Units am remote Storage-System verwendet.

  Unter VM2000 sind für die implizite Gerätezuordnung Randbedingungen zu beachten, siehe Abschnitt „Hinweis für VM2000“ im Kapitel "Snapset-Betrieb vorbereiten".
- Snapset im Storage-System anlegen
- CCOPY-Session für den Snapset aufbauen

Beim Exportieren des Pubsets werden die zugehörigen Snapsets außer Betrieb genommen. Die beim Importieren ausgeführten Aktionen werden in umgekehrter Reihenfolge durchgeführt.

Snapset löschen

Unabhängig von der Snap-Erstellung kann die Systembetreuung bzw. die HSMS-Administration für einen importierten Pubset einen vorhandenen Snapset auch explizit löschen:

```
/DELETE-SNAPSET PUBSET=A[,SNAPSET=*EARLIEST]
```

Hier wird der älteste Snapset des Pubsets A gelöscht.
Die Angabe SNAPSET=*EARLIEST kann auch entfallen (Voreinstellung).

Der zu löschende Snapset kann auch über seine Snapset-Identifikation oder als relative Angabe zum aktuellen Stand des Pubsets (mit -n, wobei -1 den jüngsten Snapset bezeichnet) angegeben werden. Mit dem Löschen eines Snapsets steht der entsprechende Sicherungsstand des Pubsets nicht mehr zur Verfügung. Gerade laufende Restore-Vorgänge für einen zu löschenden Snapset werden mit dem Löschen des Snapsets abgebrochen.

Im Betrieb mit ETERNUS DX/AF kann jeweils nur der älteste Snapset gelöscht werden. Dies ist zu beachten, wenn ein zu löschender Snapset explizit über die Snapset-Id oder über das relative Alter angegeben wird.

Snapset-Betrieb beenden

Mit der Angabe SNAPSET=*ALL werden alle Snapsets gelöscht und der Snapset-Betrieb für den Pubset beendet. Dieser Aufruf beinhaltet folgende Maßnahmen:

- alle Snapsets löschen
- Snapset-Katalog schließen und löschen
- Snapset-Limit des Pubsets auf Null setzen

Snapsets anzeigen

Das Kommando SHOW-SNAPSET-CONFIGURATION gibt für einen Pubset Informationen zu den jeweiligen Snapsets aus, siehe Handbuch „Kommandos“ [27]:

```
/SHOW-SNAPSET-CONFIGURATION PUBSET=<cat-id>,SNAPSET=*ALL
```

Dabei werden sowohl Pubset-globale als auch Snapset-spezifische Informationen angezeigt.

Falls der privilegierte Benutzer (Privileg TSOS, OPERATING, HSMS-Administrator) Informationen über einen bestimmten Snapset anfordert, werden zusätzlich die VSNs der Pubset-Volumes und die MNs der zugeordneten Snap-Units des lokalen Stroage-Systems ausgegeben (bei remote Replikation ggf. auch für das remote Storage-System).

Hinweise und Einschränkungen

Kein Snapset-Betrieb

Der Snapset-Betrieb für einen Pubset ist – wie alle anderen über SHC-OSD angebotenen Replikationsfunktionen – nicht verträglich mit DAB-Schreib-Caching für diesen Pubset. Das Kommando CREATE-SNAPSET wird bei DAB-Schreib-Caching abgewiesen.

Snapset-Betrieb mit remote Replikation

Wenn Snapsets auch auf dem remote Storage-System betrieben werden sollen (siehe Abschnitt „Snapset-Betrieb bei Katastrophenschutz mit remote Replikation“ im Kapitel "Snapset-Betrieb vorbereiten"), sind folgende Einschränkungen zu beachten:

- Die Snapset-Erzeugung auf dem remote Storage-System ist nicht möglich, wenn die remote Replikation unterbrochen ist (entweder mit dem Kommando HOLD-REMOTE-COPY oder verursacht durch unterbrochene Links), wenn die remote Spiegelplatte noch nicht synchronisiert ist oder bei asynchroner remote Replikation.
- Wenn die Snapset-Erzeugung auf einem der beteiligten Storage-Systeme (lokal oder remote) scheitert, so wird auch auf dem anderen Storage-System kein Snapset erstellt.
- Wenn der Zugriff auf die Platten des Pubsets dynamisch zwischen Source- und Target-System umgeschaltet wird (z.B. durch SHC-OSD-Kommandos), dann wird der Zugriff auf die dem Pubset zugeordneten Snapsets nicht automatisch mit umgeschaltet.
  Das Kommando ADAPT-SNAPSET-ACCESS stellt sicher, dass die zugeordneten Snapsets nach einer solchen Umschaltung weiter zur Verfügung stehen, ohne dass der Pubset exportiert werden muss.

  Beim Kommandoaufruf wird geprüft, ob der Zugriff auf die zugeordneten Snapsets im gleichen Controller erfolgt wie der Zugriff auf die Platten des Pubsets. Wenn dies nicht der Fall ist, dann wird die Umschaltung für die dem Pubset zugeordneten Snapsets nachgebildet:
  - Die aktuell zugeschalteten Snapsets werden außer Betrieb genommen.
  - Anschließend werden die Snapsets des lokalen Controllers zugeschaltet.

Betriebsmittelengpass und seine Folgen

Der benötigte Speicherplatz für die zu sichernden Originaldaten wird aus den entsprechend zugeordneten Snap-Units und ggf. einem Save-Pool bereitgestellt. Hinweise dazu und die möglichen Folgen bei einem Überlauf finden Sie im Abschnitt „Snapset-Betrieb vorbereiten".

Zur Vermeidung eines hohen Speicherbedarfs für die zu sichernden Originaldaten sollten folgende Maßnahmen getroffen werden:

- keine Paging-Dateien in Pubsets mit Snapset-Betrieb erzeugen
- den Pubset möglichst wenig mit dem Softwareprodukt SPACEOPT reorganisieren

Änderung der Pubset-Konfiguration

Bevor eine der folgenden Änderungen der Pubset-Konfiguration durchgeführt wird, muss der Snapset-Betrieb beendet werden (bedeutet Löschen aller Snapsets):

- vor dem Umbenennen eines Pubsets
- vor dem Überführen eines SF-Pubsets in einen Volume-Set eines SM-Pubsets
- vor dem Wechsel auf andere Pubset-Platten, d.h. beim Übergang von K- auf NK-Format, oder bei einem physikalischen Umzug mit FDDRL (Save/Restore)

Bei Vergrößerung des Pubsets um weitere Platten muss auch die Menge der passenden Snap-Units entsprechend erweitert werden (siehe Abschnitt „Snapset-Betrieb vorbereiten").

Bei der Reduzierung des Pubsets um Platten werden vorhandene Snapsets ungültig. Sie können nicht mehr zur Rekonstruktion herangezogen werden. Sie werden nach einer Neuinitialisierung der remote Platten beim nächsten Importieren des Pubset aus dem Snapset-Katalog entfernt. Vorhandene Snapsets müssen daher vor dem Herauskonfigurieren von Platten mit HSMS gesichert werden, falls die dort abgelegten Daten noch benötigt werden. Unmittelbar nach der Konfigurationsänderung sollte ein neuer Snapset erzeugt werden, damit die Rekonstruktion von Dateien und Jobvariablen sowie des gesamten Pubsets möglich bleibt.

Migrierte Dateien

Damit migrierte Dateien nach der Rekonstruktion von Snapsets noch benutzbar sind, müssen die zugehörigen Bänder noch vorhanden sein. Sie dürfen nicht freigegeben oder schon neu verwendet worden sein.
Wenn vorhandene Snapsets einen Zeitraum von **n** zurückliegenden Tagen für die Rekonstruktion abdecken, dann sollte die Rekonstruktion der migrierten Dateien mit HSMS stets nur für einen Zeitraum vor diesen **n** Tagen durchgeführt werden.

13.2.3 Rekonstruktion von Dateien und Jobvariablen

Rekonstruktion von Dateien und Jobvariablen

Sobald der Snapset-Betrieb für einen Pubset aufgenommen wurde, stehen die Pubset-Kopien auf den erstellten Snapsets allen Benutzern für die Rekonstruktion zur Verfügung. Die Snapsets sind betriebsbereit, wenn der Pubset importiert ist. Der nichtprivilegierte Benutzer hat dabei Zugriff auf Dateien und Jobvariablen gemäß der DVS-Zugriffsrechte.

Informationen über alle betriebsbereiten Snapsets erhalten die Benutzer über das Kommando SHOW-SNAPSET-CONFIGURATION, wobei der privilegierte Benutzer auch
genauere Informationen über Snapset-Platten erhält:

```
/SHOW-SNAPSET-CONFIGURATION PUBSET=A[,SNAPSET=*ALL]
```

Vor Rekonstruktion einer Datei bzw. Jobvariable kann der Benutzer feststellen, welcher Sicherungsstand, d.h. welcher Snapset, verwendet werden soll.

Informationen über gesicherte Dateien/Jobvariablen anzeigen

Eine Übersicht über Dateien bzw. Jobvariablen, die in einer Snapset-Sicherung vorhanden und damit aus dieser Sicherung rekonstruiert werden können, erhält der Benutzer mit dem Kommando LIST-FILE-FROM-SNAPSET bzw. LIST-JV-FROM-SNAPSET. Die Kommandos sind an allen Pubset-Sharern parallel und synchron ausführbar.

Der nichtprivilegierte Benutzer erhält Informationen über Dateien und Jobvariablen, auf die er gemäß DVS-Rechten zugreifen kann (vergleichbar mit SHOW-FILE-ATTRIBUTES bzw. SHOW-JV-ATTRIBUTES). Die Systembetreuung hat entsprechend die erweiterten Rechte.

Zur Selektion einer Menge von Dateien/Jobvariablen ist Teilqualifizierung und die Angabe von Musterzeichen im Datei-/Jobvariablennamen, nicht aber in der Katalog- und/oder
Benutzerkennung erlaubt.

Die ausgegebenen Informationen enthalten neben den Namen noch die relevanten Attribute für die Rekonstruktion: Größe, Sonderkennzeichen für migrierte Dateien und Banddateien, Erstellungsdatum und Datum des letzten Zugriffs (bzw. Ablaufdatum bei Jobvariablen), sowie den Dateistatus. Der Dateistatus zeigt an, ob die Datei zum Zeitpunkt der Sicherung schreibgeöffnet war (STATE=OPENED bzw. CLOSED). Bei Dateien, die nicht rekonstruiert werden können (z.B. Dateien mit der Sicherungshäufigkeit BACKUP-CLASS=E oder spezielle Systemdateien), wird STATE=NOREST angezeigt.

Dateien/Jobvariablen rekonstruieren

Dateien bzw. Jobvariablen kann der Benutzer mit dem Kommando RESTORE-FILE-FROM-SNAPSET bzw. RESTORE-JV-FROM-SNAPSET von einem Snapset rekonstruieren. Die Kommandos sind an allen Pubset-Sharern parallel und synchron ausführbar.

Der nichtprivilegierte Benutzer kann auf Dateien und Jobvariablen gemäß seiner DVS-Rechte zugreifen (Leserechte für die Originaldatei und Eigentümerrechte für die Zieldatei).
Die Systembetreuung hat entsprechend die erweiterten Rechte.

Zur Selektion einer Menge von Dateien/Jobvariablen ist Teilqualifizierung und die Angabe von Musterzeichen im Datei-/Jobvariablennamen, nicht aber in der Katalog- und/oder Benutzerkennung erlaubt.

Die angegebenen Dateien/Jobvariablen werden von einem bestimmten Snapset (mit dem Operanden SNAPSET=*LATEST ist der jüngste Snapset voreingestellt) rekonstruiert.
Bei der Rekonstruktion können aber auch ein Snapset-Bereich (Operand SNAPSET=*INTERVAL) oder alle Snapsets (Operand SNAPSET=*ALL) einbezogen werden. In diesem Fall werden die Dateien/Jobvariablen auf den jeweils neuesten Stand dieser Snapsets hergestellt.

Dateien und Jobvariablen werden, wie beim HSMS-Restore aus Backup-Archiven, mit ihren ursprünglichen Katalogattributen wiederhergestellt. Insbesondere entsprechen der Erstellungs- und Änderungszeitpunkt sowie die Schutzattribute dem Original zum Siche-rungszeitpunkt, d.h. zum Zeitpunkt der Snapset-Erstellung.
In einem Protokoll können wahlweise nur die wegen Fehler nicht rekonstruierten Dateien/Jobvariablen oder auch alle rekonstruierten Dateien/Jobvariablen aufgelistet werden.

Zusätzliche Optionen bei der Rekonstruktion

Beim Erzeugen eines Snapsets werden alle Dateien und Jobvariablen des Pubsets gleichzeitig gesichert. Ebenso werden alle Seiten einer Datei gleichzeitig gesichert. Dadurch sind schreibgeöffnete Dateien crash-consistent gesichert. Diese Dateien werden standardmäßig nicht rekonstruiert (außer Dateien mit dem Indikator ONLINE-SAVE). Die Rekonstruktion solcher Dateien kann explizit angefordert werden (Operand RESTORE-OPEN-FILES=*YES). Im Falle von ISAM-Dateien kann eine Nachbearbeitung mit dem Kommando REPAIR-DISK-FILE notwendig sein. Dauernd geöffnete Systemdateien für SRPM, GUARDS und GCF werden bei der Snapset-Erzeugung in einen konsistenten Zustand gebracht, um später eine ordentliche Rekonstruktion zu ermöglichen.

Dateien und Jobvariablen können unter einem neuen Namen rekonstruiert werden (Operand NEW-FILE-NAME bzw. NEW-JV-NAME). Die Umbenennung erfolgt entweder durch Angabe einer anderen Benutzerkennung oder eines Dateinamenspräfix. Der nichtprivilegierte Benutzer kann eine andere Benutzerkennung angeben, wenn er Miteigentümer dieser Kennung ist. Die eigene Benutzerkennung kann er angeben, wenn er eine fremde Datei (Lesezugriff ist erforderlich) auf der eigenen Kennung rekonstruieren will.

Bereits bestehende Dateien und Jobvariablen werden standardmäßig nicht überschrieben, also nicht rekonstruiert. Das Rekonstruieren dieser Dateien/Jobvariablen kann explizit angefordert werden (Operand REPLACE=*YES).
Das Rekonstruieren erfordert die Berechtigung zum Löschen (z.B. Schreibkennwort). Der privilegierte Aufrufer kann den Schreibschutz explizit ignorieren (Operand IGNORE-PROTECTION=*YES).

Einschränkungen

Einzelne Dateigenerationen können nicht rekonstruiert werden, nur ganze Dateigenerationsgruppen.

Für migrierte Dateien und Banddateien werden nur die Katalogeinträge wiederhergestellt. Eine Umbenennung ist in diesem Fall nicht möglich (Ausnahme: Der privilegierte Benutzer unter TSOS kann bei einer Banddatei eine andere Benutzerkennung angeben).

PLAM-Bibliothekselemente rekonstruieren

In LMS kann mit der Anweisung OPEN-LIBRARY eine PLAM-Bibliothek auf einem Snapset im Lesemodus geöffnet werden. Elemente dieser Bibliothek lassen sich dann über die entsprechenden LMS-Anweisungen anzeigen, kopieren oder selektieren.

Sicherung vom Snapset in ein Backup-Archiv

Mit der HSMS-Anweisung BACKUP-FILES können Dateien/Jobvariablen, die auf einen Snapset gesichert wurden, in ein Backup-Archiv übernommen werden. Mit der folgenden BACKUP-Anweisung können z.B. alle Dateien und Jobvariablen der Benutzerkennung USER1, die in der letzten Snapset-Sicherung des Pubsets A gesichert wurden, in ein Backup-Archiv übernommen werden:

```
//BACKUP-FILES FILE-NAMES=$USER1., JV-NAMES=$USER1.,
               ARCHIV-NAME=<archiv>,CONCURRENT-COPY=*YES(
                WORK-FILE-NAME=*FROM-SNAPSET(PUBSET-ID=A,SNAPSET-ID=-1)
```

Diese Funktion steht auch nichtprivilegierten Benutzern zur Verfügung.

Die Übernahme vom Snapset in ein Backup-Archiv ist zu jedem beliebigen Zeitpunkt zwischen Erzeugen und Löschen des Snapsets möglich. Sie erfolgt praktisch „offline“ zum laufenden Pubset-Betrieb. Die in HSMS neu erzeugte Sicherungsversion erhält als Sicherungszeitpunkt dann nicht das aktuelle Datum des BACKUP-FILES-Aufrufs, sondern das Erstellungsdatum des Snapsets. Im Backup-Archiv darf jedoch noch keine neuere Sicherungsversion existieren.

13.2.4 Rekonstruktion von Pubsets

Rekonstruktion von Pubsets

Die Systembetreuung kann mit dem Kommando RESTORE-PUBSET-FROM-SNAPSET auch den gesamten Pubset auf Volume-Basis aus einem vorhandenen Snapset des
Pubsets rekonstruieren. Diese Funktion kann nicht im laufenden Pubset-Betrieb ausgeführt werden. Der Pubset muss exportiert sein, d.h. er darf weder lokal noch remote in einem
anderen System importiert sein. Außerdem muss der Pubset noch auf den Originalplatten existieren und die ursprünglichen Pubset-Eigenschaften aufweisen (siehe auch „Änderungder Pubset-Konfiguration“ im Kapitel ''Sicherungsbetrieb mit Snapsets'').

Beim Rekonstruieren wird der Pubset im Storage-System auf den Stand des jeweiligen Snapsets gebracht. Der Pubset besitzt dann den Stand wie unmittelbar vor dem Erzeugen
dieses Snapsets.

Der Pubset ist bis zum Abschluss der Rekonstruktion gegen die Inbetriebnahme durch /IMPORT-PUBSET geschützt.

Während der Rekonstruktion wird auf dem Home-Pubset eine temporäre Kopie des Snapset-Katalogs des zu rekonstruierenden Pubsets mit dem Dateinamen $TSOS.SYSWRK.SNAPSET.<catid> angelegt. Diese Datei wird nur für die Rekonstruktion benötigt und erst nach Abschluss der Rekonstruktion wieder gelöscht. Während der Rekonstruktion ist sie gegen Löschen geschützt, da sie zum Wiederaufsetzen einer unterbrochenen Rekonstruktion dient (s.u.). Wenn der Pubset auf eine andere Weise rekonstruiert wird, so wird die Datei beim nächsten Pubset-Import gelöscht.

Nach einem Abbruch des Systemlaufs während der Rekonstruktion befindet sich der Pubset in einem Zwischenzustand, der einer teilweisen Rekonstruktion entspricht. In diesem Fall kann die Rekonstruktion durch einen erneuten Kommandoaufruf beendet werden. Die Rekonstruktion kann jedoch nur in der gleichen Ablaufumgebung wieder aufgesetzt werden, d.h. mit dem gleichen Home-Pubset, da sich dort die erforderliche Kopie des Snapset-Katalogs befindet.

Im Betrieb mit ETERNUS DX/AF Systemen kann die Rekonstruktion eines Pubsets auf jedem beliebigen Snapset-Stand durchgeführt werden. Alle Snapset-Stände, auch der zur Rekonstruktion verwendete Stand, bleiben erhalten und können weiter genutzt werden.

Hinweis zum Operanden SNAPSET des Kommandos RESTORE-PUBSET-FROM-SNAPSET

- SNAPSET = *LATEST
  Während der Funktionsausführung wird der Pubset *nicht* implizit importiert. Die F5-Label-Rekonstruktion erfolgt dann, wenn der wiederhergestellte Pubset importiert wird .
- SNAPSET = <name 1..1 with-low> / <integer -52..-1> (explizite Angabe)
  Während der Funktionsausführung wird der Pubset implizit importiert und am Ende wieder exportiert. Dabei kommt es implizit auch zur F5-Label-Rekonstruktion.

Änderung der Pubset-Konfiguration

Wenn der Pubset nach der letzten Snapset-Sicherung um Platten reduziert wurde und wenn diese Platten mit VOLIN neu initialisiert wurden, dann ist die Rekonstruktion des Pubsets nicht mehr möglich. Das Kommando RESTORE-PUBSET-FROM-SNAPSET wird in diesem Fall abgewiesen.

Die Möglichkeit der Rekonstruktion des Pubsets nach einer Pubset-Reduzierung kann mit einer der folgenden Maßnahmen erhalten werden:

- weggefallene Platten nicht neu initialisieren
- unmittelbar einen neuen Snapset erzeugen

> ¡ VOLIN erlaubt keine Initialisierung von Snap-Units und reagiert bei der Initialisierung von Originalplatten mit einer Rückfrage, wenn noch Snap-Units verkettet sind.

13.3 Rekonstruktion von Dateien und Datenträgern

Rekonstruktion von Dateien und Datenträgern

Daten können sowohl auf Datei- als auch auf Datenträgerebene gesichert und rekonstruiert werden. Es gibt mehrere Gründe, die eine Rekonstruktion von Dateien bzw. Platten erforderlich machen:

- Eine oder mehrere Dateien müssen aus den Sicherungsbeständen eingespielt werden, weil sie aus Kapazitätsgründen nicht online gehalten werden konnten.
- Das gesamte System muss nach einem Ausfall rekonstruiert werden.
- Eine Platte muss rekonstruiert werden.

Das Ziel der Rekonstruktion ist die schnelle Bereitstellung der gesicherten Dateien. Je nachdem, ob es sich bei der ausgefallenen Platte um eine gemeinschaftliche oder
private Platte handelt, sind unterschiedliche Maßnahmen zu ergreifen.

Rekonstruktion einzelner Dateien

Verliert ein Benutzer eine Datei durch fehlerhafte Bedienung – wie z.B. durch versehentliches Löschen – gibt es verschiedene Möglichkeiten der Rekonstruktion durch Rückgriff auf
die letzte Sicherungskopie.

- Rückgriff auf die logische Sicherung der Datei

  Die Datei ist logisch auf einem privaten Datenträger gesichert. Die Sicherungskopie ist zu übernehmen und die inzwischen an der Originaldatei vorgenommenen Änderungen
  sind nachzutragen.
- Rückgriff auf die physikalische Sicherung der Platte oder des Bandes, auf der/dem die Datei liegt (Ausnahmefall)

  Die letzte physikalische Sicherung mit FDDRL liegt auf einer Sicherungsplatte vor oder muss vom Sicherungsband auf eine Platte eingespielt werden. Die betreffende Datei
  wird von dieser Platte auf die Originalplatte kopiert. Die inzwischen vorgenommenen Änderungen in der ersetzten Datei sind in der Sicherungskopie nachzutragen.

Rekonstruktion einer gemeinschaftlichen Platte

Nach dem Ausfall einer gemeinschaftlichen Platte muss zunächst dafür gesorgt werden, dass eine funktionsfähige Platte, die auf einem Reservegerät montiert wird, zur Verfügung steht.

Bei Ausfall einer gemeinschaftlichen Platte muss der gesamte Pubset rekonstruiert werden. Der Rekonstruktionsaufwand hängt vom Umfang und der Aktualität der zur Verfügung stehenden letzten physikalischen Sicherung ab.
Diese Sicherung ist zunächst vom Band einzuspielen. Je nach Stand dieser physikalischen Sicherung müssen noch logische Sicherungen eingelesen werden, um den letzten aktuellen Stand zu erhalten.

Danach sind die Katalogeinträge der Benutzerdateien, die auf privaten Platten liegen, zu importieren oder die letzte logische Sicherung einzuspielen. Dies kann mit einer Kommandofolge erfolgen, die ein IMPORT-FILE-Kommando für jede montierte private Platte enthält.

Rekonstruktion eines SM-Pubsets

Ist in einem SM-Pubset der Control-Volume-Set nicht mehr zugreifbar, fällt der gesamte SM-Pubset aus.

Um die Daten auf den anderen Volume-Sets weiterverarbeiten zu können, kann der exklusive Import eines Volume-Sets erfolgen, der dieses zu einem SF-Pubset mit gleicher Katalogkennung konvertiert. Dazu wird im Kommando IMPORT-PUBSET der Operand `USE=*EXCLUSIVE (CONVERT-VOLUME-SET=*YES)` zur Verfügung gestellt.

Control-Volume-Sets sind von einer Konvertierung ausgeschlossen.

Die Daten und Metadaten des ausgefallenen Control-Volume-Sets müssen nach Bereitstellung eines zusätzlichen SF-Pubsets aus der Sicherung rekonstruiert werden. Danach kann mit dem Dienstprogramm SMPGEN aus den konvertierten SF-Pubsets und dem ebenfalls als SF-Pubset vorliegenden rekonstruierten Control-Volume-Set ein neuer SM-Pubset erzeugt werden.

Rekonstruktion eines Pubsets

Wenn ein Pubset (SM oder SF), der im Snapset-Betrieb gesichert wurde, nicht mehr importiert werden kann, lässt sich der Pubset mit dem Kommando RESTORE-PUBSET-FROM-SNAPSET auf dem Stand des jüngsten Snapsets rekonstruieren (siehe Abschnitt „Rekonstruktion von Pubsets").

Rekonstruktion einer privaten Platte

Nach Bereitstellen eines intakten Plattengerätes müssen folgende Arbeiten durchgeführt werden:

Die private Platte ist mit dem Dienstprogramm VOLIN zu initialisieren (siehe Handbuch „Dienstprogramme“ [15]).
Bei der Rekonstruktion durch FDDRL ist die Sicherung vom Band einzuspielen. Je nach Stand dieser physikalischen Sicherung kann noch das Einlesen logischer Sicherungen nötig sein, um den aktuellen Stand wieder herzustellen.
Im Anschluss daran sind alle Katalogeinträge von Dateien, die im Katalog einen Eintrag für den betreffenden Datenträger hatten, zu löschen (mit dem Kommando EXPORT-FILE).

Danach können die Dateien mit ARCHIVE wieder rekonstruiert werden. Dabei ist zu beachten, dass bei Dateien, die auf mehreren Privatplatten hinterlegt sind, alle betroffenen Datenträger rekonstruiert werden müssen. Ansonsten ist aber jede ARCHIVE-Sicherung logisch, d.h. jede Datei kann einzeln rekonstruiert werden. Hier ist empfehlenswert, die Privatplatten gezielt – je Platte eine FILES-Anweisung in ARCHIVE – zu sichern; die Rekonstruktion kann dann über die zugehörige SVID erfolgen. Zudem kann in der FILES-Anweisung der Operand ORIGIN genutzt werden, mit dem die VSN des Datenträgers, dessen Dateien rekonstruiert werden sollen, angegeben wird.

Das folgende Bild zeigt eine Rekonstruktion nach Systemfehler.

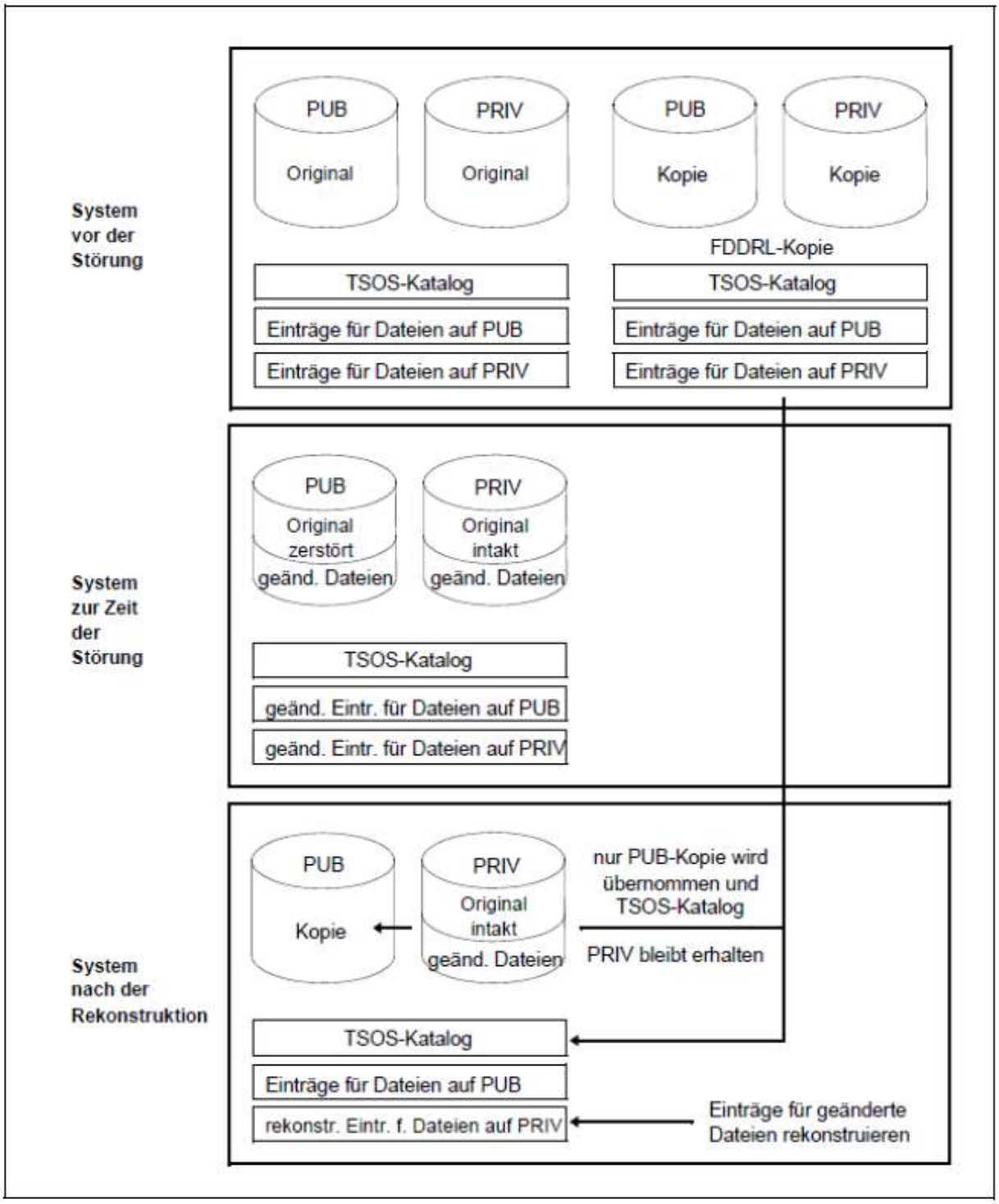


Bild 22: Rekonstruktion nach Systemfehler

13.4 Archivsysteme

Archivsysteme

Die roboterunterstützte Automatisierung der Peripheriebedienung zeichnet sich durch folgende Vorteile aus:

- Verlegung zeitaufwändiger Sicherungsläufe in unbediente Betriebszeiten
- rationelle Nutzung der Kassettenlaufwerke
- Erhöhung der Datensicherheit durch Installation des Archivierungssystems in abgesicherten und zugangskontrollierten Räumen
- Standardisierung der Aktionen und Antworten auf Bandanforderungen
- Entlastung der Plattenperipherie bei vollautomatischem Zugriff auf externe Bestände durch Auslagerung wenig benutzter Online-Daten von Platte auf Kassette

BS2000 unterstützt reale Archivsysteme. Sie werden über das Softwareprodukt ROBAR (siehe Handbuch „ROBAR “ [41]) gesteuert.

Die Bänder können von MAREN verwaltet werden. Die Entscheidung, welches Archivsystem betroffen ist, trifft MAREN anhand des Lagerortes (zu Lagerorten siehe auch "Geräteauswahlmechanismus für Bandgeräte").

Zu MAREN siehe auch den Abschnitt „Bandverwaltung mit MAREN" und das Handbuch „MAREN“ [31].

Neben realen Archivsystemen kann auch das virtuelle Archivsystem **ETERNUS CS** zum Einsatz kommen. Mit ETERNUS CS wird dem realen Bandrobotersystem (mit den realen Laufwerken und Kassetten) ein virtuelles Bandrobotersystem vorgelagert. Auf diese Weise ist eine völlige Entkopplung zwischen Server und realem Archiv realisiert. Das virtuelle Bandrobotersystem kennt sogenannte virtuelle (logische) Laufwerke und virtuelle (logische) Volumes. Das Kernstück besteht dabei im wesentlichen aus einem Plattensystem als Daten-Cache, so dass nicht nur ein extrem schneller Zugriff auf die Daten gewährleistet wird, sondern auch aufgrund der hohen Anzahl zu generierender virtueller Laufwerke (bis zu 1024) und logischer Volumes (bis zu 1.500.000) bestehende Engpässe bei einem realen Archivsystem beseitigt werden.

Zu ETERNUS CS siehe auch das Handbuch „ETERNUS CS“ [8].

14 Accounting

Das Abrechnungssystem von BS2000 hat die Aufgabe, Daten für die Benutzerabrechnung und die Betriebsabrechnung zu sammeln und bereitzustellen.

Durch die für die Benutzerabrechnung erfassten Daten kann das Data Center den Benutzern die in Anspruch genommenen Betriebsmittel und Dienstleistungen in Rechnung stellen.

Die gesammelten Daten für die Betriebsabrechnung geben eine zeitlich lückenlose Auskunft über die Benutzung und Verfügbarkeit des gesamten Server-Systems.

Die Daten werden in Form von Abrechnungssätzen in eine eigens dafür vorgesehene Abrechnungsdatei geschrieben. Die Abrechnungsdatei wird mit speziellen Programmen ausgewertet.

Am Ende des Kapitels sind alle Abrechnungssätze von BS2000 in einer Übersicht dargestellt (eine ausführliche Beschreibung der Abrechnungssätze und ihrer Struktur finden Sie im Handbuch „Abrechnungssätze“ [1]).

14.1 Steuerung des Abrechnungssystems

Das gesamte Abrechnungssystem von BS2000 wird von der Systembetreuung gesteuert. Diese bestimmt den Zeitpunkt, an dem das Abrechnungssystem gestartet werden soll, vereinbart den Namen der Abrechnungsdatei und legt Namen und Anzahl der Abrechnungssätze und Satzerweiterungen fest, die in der Abrechnungsdatei erfasst werden sollen.
Die Systembetreuung legt ferner den Zyklus und den Umfang der periodischen Erfassung fest, die bestimmte Abrechnungsätze und Jobklassen umfasst.

Der festgelegte Ablauf kann mit der Accounting-Exit-Routine derart modifiziert werden, dass der Abrechnungssatz geändert, unterdrückt oder durch weitere Abrechnungssätze ergänzt wird (siehe Handbuch „System-Exits“ [56]).
Die Accounting-Exit-Routine wird aktiviert, bevor ein Abrechnungssatz in die Abrechnungsdatei geschrieben wird.

Mit den Kommandos START- und STOP-ACCOUNTING wird das Abrechnungssystem aktiviert bzw. beendet.

Über die Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES kann für jeden Benutzer die Anzahl der benutzerspezifischen Datensätze festgelegt werden, die in die Abrechnungsdatei geschrieben werden darf.

Mit dem Abrechnungssystem kann die Systembetreuung dynamisch Abrechnungssätze ganz oder teilweise ein- und auszuschalten und Einfluss auf den Umfang der einzelnen Abrechnungssätze nehmen.
Mit dem Kommando MODIFY-ACCOUNTING-PARAMETERS können nicht benötigte Abrechnungssätze und Satzerweiterungen ausgeschaltet werden.

Mit dem Kommando SHOW-ACCOUNTING-STATUS informiert sich die Systembetreuung über folgende Daten

- Zustand des Abrechnungssystems (eingeschaltet oder ausgeschaltet)
- Name der aktuellen Abrechnungsdatei
- Namen der Folgedateien
- ein- und ausgeschaltete Sätze bzw. Satzerweiterungen
- Häufigkeit der periodischen Erfassung bestimmter Abrechnungssätze
- Namen der Jobklassen, über die zyklisch Daten erfasst werden

Kommando	**Bedeutung**
ADD-USER	Abrechnungsspezifische Benutzerkennungs-Einträge erstellen
CHANGE-ACCOUNTING-FILE	Abrechnungsdatei wechseln
CREATE-FILE	Vereinbaren der Dateimerkmale der Abrechnungsdatei oder Anlegen der Abrechnungsdatei auf Band oder privater Platte
MODIFY-ACCOUNTING-PARAMETERS	Parameter der Systemabrechnung festlegen
MODIFY-DEFAULT-ACCOUNT	Standard-Abrechnungsnummern für BS2000- und POSIX-Zugang vereinbaren
MODIFY-FILE-ATTRIBUTES	Vereinbaren der Dateimerkmale der Abrechnungsdatei
MODIFY-USER-ATTRIBUTES	Abrechnungsspezifische Benutzerkennungs-Einträge ändern
SHOW-ACCOUNTING-STATUS	Informationen über das Abrechnungssystem anfordern
SHOW-USER-ATTRIBUTES	Abrechnungsspezifische Informationen aus dem Benutzerkatalog anfordern
START-ACCOUNTING	Abrechnungssystem aktivieren
STOP-ACCOUNTING	Abrechnungssystem beenden
WRITE-ACCOUNTING-RECORD	Benutzer-Abrechnungssatz schreiben
Makro	**Bedeutung**
ARDS	DSECT zur Beschreibung der Abrechnungssätze erzeugen
AREC	Benutzer-Abrechnungssatz schreiben
ASPC	Erfassung der momentanen Speicherplatzbelegung auf gemeinschaftlichen oder privaten Datenträgern in Form von Accounting-Abrechnungssätzen
SRMUINF	Daten für die Abrechnung aus dem Benutzerkatalog lesen

Tabelle 39: Schnittstellenübersicht für die Systemabrechnung

14.2 Abrechnungsdatei

Der Dateiname für die Abrechnungsdatei kann bei der Systemeinleitung über den Parameterservice oder während des Systemlaufs festgelegt werden.

- Systemeinleitung

  Das Abrechnungssystem kann im Rahmen der Systemeinleitung aktiviert werden, indem die dazu notwendigen Anweisungen über den Parameterservice mitgeteilt werden. Die Aktivierung des Abrechnungssystems im Rahmen des Parameterservice ist im Abschnitt „Starten des Abrechnungssystems (ACCOUNT)" beschrieben.
- BS2000-Systemlauf

  Die Systembetreuung kann die Abrechnungsroutine im laufenden BS2000 mit dem Kommando START-ACCOUNTING einschalten. Das ist dann sinnvoll, wenn das Abrechnungssystem z.B. zu Testzwecken für eine Weile ausgeschaltet war.
  Soll die aktuelle Abrechnungsdatei ausgewertet werden, wird über das Kommando CHANGE-ACCOUNTING-FILE die aktuelle Abrechnungsdatei geschlossen und eine neue Datei eröffnet.

Automatische Generierung des Dateinamens

Die Dateinamen für die Abrechnungsdateien können vom System automatisch generiert werden. Der Dateiname setzt sich aus dem Präfix $TSOS.SYS.ACCOUNT. und einem für alle Loggingverfahren gültigen Suffix yy.mm.dd.xxx.nn bzw. yyyy-mm-dd.xxx.nn zusammen (vgl. Dateiname der Protokolldatei CONSLOG und SERSLOG-Datei).

Dabei bedeutet:

yyyy-mm-dd oder yy.mm.dd	Datum, an dem die Abrechnungsdatei eingerichtet wird (ob das Jahr zwei- oder vierstellig ausgegeben wird, ist abhängig vom Systemparameter FMTYFNLG)
xxx	Nummer des aktuellen Systemlaufs
nn	laufende Nummer der Abrechnungsdatei (01-99; beginnt beim STARTUP immer mit 01)

Die Systembetreuung kann die Voreinstellung für das Präfix $TSOS.SYS.ACCOUNT ändern, wenn sie bestimmte Namenskonventionen im Data Center berücksichtigen muss.

Wenn an Stelle eines vollqualifizierten Dateinamens ein teilqualifizierter Dateiname angegeben wird, sind folgende Punkte zu beachten:

- Ein teilqualifizierter Dateiname wird automatisch durch das Suffix yy.mm.dd.xxx.nn bzw. yyyy-mm-dd.xxx.nn ergänzt. Aus der Länge des Suffixes ergibt sich ein maximal 26 Zeichen langer teilqualifizierter Dateiname (ohne Benutzerkennung); bei einer Katalogkennung mit mehr als einem Zeichen reduziert sich dieser Wert um die Anzahl der Mehrstellen.
- Wurde der Dateiname ohne Benutzerkennung angegeben, dann wird $TSOS. dem Dateinamen vorangestellt.
- Enthält der teilqualifizierte Dateiname nur die Benutzerkennung, wird der Name erweitert auf $userid.SYS.ACCOUNT.<date>.xxx.nn (<date>= yy.mm.dd oder yyyy-mm-dd).

Mögliche Fehler beim Eröffnen des Abrechnungssystems

- Bei Syntaxfehlern in den Parametersätzen während der Systemeinleitung fordert eine Fehler-Routine des Parameterservices den Operator an der Konsole auf, den fehlerhaften Parameter zu korrigieren oder zu ignorieren.
- Für die zu eröffnende Datei wird vom Abrechnungssystem ein FILE-Makro abgesetzt. Tritt dabei ein Fehler auf, dann wird eine Abrechnungsdatei mit dem Standardnamen $TSOS.SYS.ACCOUNT.<date>.xxx.nn und den entsprechenden Standardattributen angelegt.
- Dies gilt auch für den Fall, dass die Datei, die eröffnet werden sollte, bereits existiert und mit dem Schutzattribut ACCESS=READ katalogisiert ist.
  Zu beachten ist, dass in vorhergehenden Betriebssystemversionen die Abrechnungsdateien mit ACCESS=READ katalogisiert wurden.
- Nach maximal drei erfolglosen Versuchen, die Datei mit dem Standardnamen zu eröffnen, erscheint eine Meldung auf der Konsole (keine Antwort notwendig), und BS2000 wird ohne Abrechnungssystem geladen.
  Die Systembetreuung kann das Abrechnungssystem im aktuellen Systemlauf per Kommando einschalten.

14.2.1 Merkmale und Inhalt der Abrechnungsdatei

Die Abrechnungsdatei ist eine SAM-Datei mit variabler Satzlänge. Die übrigen Dateimerkmale kann die Systembetreuung sowohl über die Kommandos CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES als auch über die Abrechnungskommandos START-ACCOUNTING und CHANGE-ACCOUNTING-FILE vereinbaren.

Folgende Punkte sind zu beachten:

- Als Standardwerte gelten für

SPACE:	48 PAM-Blöcke Primär- und 48 PAM-Blöcke Sekundärzuweisung
BUFFER-LENGTH:	1 PAM-Block bzw. 2 PAM-Blöcke bei NK4-Pubsets
VOLUME:	wird vom DVS festgelegt

- Wird der VOLUME-Operand bei den Abrechnungskommandos vereinbart, dürfen nur Datenträger vereinbart werden, die vom Datenverwaltungssystem von BS2000 automatisch unterstützt werden und keine zusätzlichen gerätespezifischen Angaben benötigen.
  Soll die Abrechnungsdatei auf **Band** oder **privater Platte** angelegt werden, muss die Datei vor dem Starten des Abrechnungssystems mit dem Kommando CREATE-FILE angelegt werden. Gehört im Falle einer **gemeinschaftlichen Platte** diese nicht dem Standard-Pubset der vorgesehenen Benutzerkennung an, muss die Katalogkennung des Datenträgers im Dateinamen mit angegeben werden.
- Die Abrechnungsdatei wird im EXTEND-Modus eröffnet. Das bedeutet, dass eine vorhandene Datei fortgeschrieben wird.
- Eine neue Abrechnungsdatei wird mit den Schutzattributen ACCESS=WRITE und USER-ACCESS=OWNER-ONLY katalogisiert.
  Existiert die Abrechnungsdatei bereits, dann wird keine Veränderung der Attribute vorgenommen. Die Systembetreuung muss demzufolge darauf achten, dass die Abrechnungsdatei mit den entsprechenden Attributen versehen wird.
- Wird für eine Datei nur ein Katalogeintrag angelegt und noch kein Speicherplatz reserviert (Kommando CREATE-FILE mit Operand SUPPORT=NONE), so tritt beim Versuch, diese Datei als Abrechnungsdatei zu eröffnen, ein Fehler auf.
  Um das zu verhindern, muss die Systembetreuung dieser Datei vor dem Eröffnen primären Speicherplatz zuweisen.

Die Abrechnungsroutine schreibt folgende Informationen in die Abrechnungsdatei:

- Taskabrechnung
- Programmabrechnung
- Geräte- und Datenträgerabrechnung
- Speicherplatzabrechnung
- SPOOLOUT-Abrechnung
- Benutzer-spezifische Abrechnung
- Betriebsabrechnung
- Betriebsmittelverfügbarkeit
- Auftragnehmer-Leistungs-Abrechnung
- Abrechnung entbündelter Produkte
- Abrechnung von Subsystemen und Subsystemtasks

Die Informationen zur Programmabrechnung und zur Speicherplatzabrechnung werden von der Abrechnungsroutine zusätzlich periodisch erfasst. Damit ist eine genauere Abrechnung lange laufender Programme und bestimmter Jobklassen, die dieser periodischen Überwachung unterliegen, möglich.

Die Abrechnungsdaten stehen in der Abrechnungsdatei in einzelnen Abrechnungssätzen. Die Abrechnungssätze haben eine variable Satzlänge; sie beträgt maximal 492 Byte ohne Satzlängenfeld und 496 Byte mit Satzlängenfeld.

Die Struktur eines Abrechnungssatzes ist ausführlich im Handbuch „Abrechnungssätze“ [1] beschrieben.

14.2.2 Arten von Abrechnungsdaten

Programmabrechnung mit Verbrauchsstempeln

Der Betriebsmittelverbrauch eines Programmlaufs wird in zwei Abrechnungssätzen beim Start und bei der Beendigung des Programms erfasst, die den jeweils aktuellen Stand der Verbrauchswerte (*Verbrauchsstempel*) seit Beginn des Tasklaufs enthalten.
Ein Tasklauf wird vom Abrechnungssystem in die Ereignisse Task-Start, Programm-Start, Programm-Beendigung und Task-Beendigung aufgeteilt. Ein Auswertungsprogramm kann durch Differenzbildung der zugehörigen Ereignisdaten sowohl den Betriebsmittelverbrauch des Programms als auch die Verbrauchsdaten des zwischen zwei Programmen liegenden Kommando-Modus berechnen. Zur genaueren Abrechnung lange laufender Programme bietet das Abrechnungssystem auch einen dritten Satz an, der in periodischen Abständen Daten in die Abrechnungsdatei schreibt und Aufschluss über die durchschnittlichen Verbrauchswerte während der Laufzeit gibt.

Abrechnungsdaten für die Benutzerabrechnung

Für die Benutzerabrechnung stellt das Abrechnungssystem eine Reihe von Abrechnungsdaten zur Verfügung, die im Einzelnen der genauen Beschreibung der Abrechnungssätze im Anhang zu entnehmen sind. Folgende Daten werden für eine Benutzerkennung bereitgestellt:

- Daten zu den Basis-Betriebsmitteln
  - CPU-Zeit
  - E/A-System (Anzahl der transportierten Blöcke oder Byte)
  - Hauptspeicher (Working-Set-Integral)
  - Service Units (Summe aus den Einzelwerten CPU-Zeit, Ein-/Ausgaben und Hauptspeicher, berücksichtigt die CPU-Leistung)
- Erweiterte Gerätebelegungsdaten mit Zeitstempel und Gerätetyp
- Belegungsdaten des Datenträgers
- Job-Informationen und Task-Ablauf-Informationen
- Erweiterte Programm-Informationen
  - Interner Programmname
  - Beendigungsgrund
  - Speicherbelegungsintegral für Klasse-5- und Klasse-6-Speicher
- Spoolout-Informationen
  - Typ des Ausgabegerätes
  - Anzahl gedruckter Zeilen und Seiten
- Abrechnung von Speicherplatz-Änderungen

Speicherplatz-Abrechnung

Für die Abrechnung der permanenten Speicherplatz-Belegung auf gemeinschaftlichen und privaten Platten bietet das Abrechnungssystem eine Schnittstelle an, die eine Erfassung über Stichproben ermöglicht. Diese Stichproben können dabei von einem Benutzerprogramm (über Makro ASPC) unter der privilegierten Benutzerkennung der Systembetreuung gesteuert werden.
Der Makro ASPC schreibt dabei für jeden lokal verfügbaren Pubset einen oder mehrere Abrechnungssätze in die Abrechnungsdatei, wobei jeder Satz folgende Daten enthält:

- die Katalogkennung des Pubsets
- den Zeitpunkt der Bestandsaufnahme
- einen Indikator über die Vollständigkeit des Satzes
- die Benutzerkennungen des Pubsets
- die Anzahl der belegten PAM-Blöcke pro Benutzerkennung

Der Makro ASPC ist im Handbuch „Makroaufrufe an den Ablaufteil“ [30] beschrieben.

Diese Datensätze können auch periodisch in der Abrechnungsdatei erfasst werden. Die Parameter für diese zyklische Überwachung und Erhebung steuert die Systembetreuung mit entsprechenden Vorgaben auf Kommandoebene.

Betriebsabrechnung

Die Verfügbarkeit des Gesamtsystems wird durch einen Abrechnungs-Eröffnungs- und einen Abrechnungs-Abschlusssatz dokumentiert. Der Eröffnungssatz wird nach jeder Eröffnung einer Abrechnungsdatei geschrieben, der Abschlusssatz vor jedem ordnungsgemäßen Schließen der Abrechnungsdatei.
Diese Sätze enthalten im Kennzeichnungsteil bzw. in der Grundinformation:

- die Bezeichnung des Servers
- den Systemnamen und die Systemversion
- ein Kennzeichen für Startup oder Shutdown

Bei einem Wechsel der Abrechnungsdatei erhält die neue Datei einen Verweis auf ihre Vorgängerin.
Die Verfügbarkeit des Betriebsmittels CPU wird in periodisch aufgezeichneten Sätzen erfasst, die die Task-Zeiten (TU + TPR), Interrupt-Handling-Zeiten (SIH) und IDLE-Zeiten der CPU enthalten. Die Hauptspeichergröße wird im ersten Abrechnungs-Eröffnungssatz während der Systemeinleitung oder nach einem START-ACCOUNTING-Kommando aufgezeichnet.

Abrechnung von Systemdiensten in Auftragnehmertasks

Die Task ist grundsätzlich die größte Einheit, die für die Benutzerabrechnung herangezogen wird.

Systemfunktionen, die in einer Systemtask oder an einem anderen Server ausgeführt werden und einen nennenswerten Verbrauch an Betriebsmitteln aufweisen, werden in eigenen Abrechnungssätzen erfasst.

Folgende vom Benutzer angeforderte Leistungen werden in entsprechenden Leistungsgrößen ermittelt:

- Spoolout
- Userdumps
- RFA-Verbindungen

Für Systemfunktionen, bei denen der Betriebsmittelverbrauch für einen einzelnen Auftrag verhältnismäßig gering ist, die aber häufig aufgerufen werden, sind die Leistungsgrößen in den Einträgen der Task- und Programm-Abrechnungsätze integriert, z.B.:

- Anzahl Platten-Ein-/Ausgaben
- Anzahl Terminal-Ein-/Ausgaben
- Anzahl Katalogzugriffe

Die Abrechnungssätze sind in den Tabellen ab "Übersichten über die Abrechnungssätze" dargestellt.

14.3 Übersichten über die Abrechnungssätze

Die Abrechnungssätze sind einmal nach Zielgruppe (siehe unten) und einmal alphabetisch nach Gruppenzusammengehörigkeit (siehe folgende Seiten) geordnet.

Abrechnungssatz für die Benutzerabrechnung	**max. Länge in Byte**
JOBS (Job-Start-Abrechnungssatz)	314
PACC (Periodischer Programm-Abrechnungssatz)	402
PDMP (Userdump-Abrechnungssatz)	78
PRGS (Programm-Start-Abrechnungssatz)	476
PRGT (Programm-Beendigungs-Abrechnungssatz)	528
SPLO (SPOOLOUT-Abrechnungssatz)	474
TASK (Task-Abrechnungssatz)	400
TDEV (Geräte- und Datenträger-Abrechnungssatz)	492
TATR (Task-Attribut-Änderungs-Abrechnungssatz)	70
Abrechnungssatz für die Speicherplatz-Abrechnung	
DALC (Speicherplatz-Allokierungs-Abrechnungssatz)	492
DSPC (Speicherplatz-Bestandsaufnahme-Abrechnungssatz)	492
DSPP (Privatplatten-Speicherplatz-Bestandsaufnahme-Abrechnungssatz)	492
benutzerspezifische Abrechnungssätze	
UACC (Benutzer-Verbrauchsstempel-Abrechnungssatz)	382
UDAT (Benutzerdaten-Abrechnungssatz)	326
frei definierte Benutzer-Abrechnungssätze	496 (mit SLF)
Abrechnungssatz für die Betriebsabrechnung	
ACLS (Abrechnungs-Abschluss-Satz)	610 (mit PUT)
AOPN (Abrechnungs-Eröffnungs-Satz)	698 (mit PUT)
RCPU (CPU-Verfügbarkeits-Abrechnungssatz)	70
RSRV (Auftragnehmer-Leistungs-Abrechnungssatz)	[1]
Abrechnungssatz für DSSM	
ESMC (Subsystem-Initialisierungs-Abrechnungssatz)	54

ESMD (Subsystem-Beendigungs-Abrechnungssatz)	54

Tabelle 40: Übersicht über die Abrechnungssätze von BS2000

[1] Die Satzlänge hängt ab von der Auftragnehmer-Art

In den folgenden Tabellen bedeuten in der Spalte „Standardeinstellung“: IE/IA = implizit eingeschaltet/implizit ausgeschaltet, EE/EA = explizit eingeschaltet/explizit ausgeschaltet.

Satzkennung	Erweiterungskennung	Bedeutung	Ausschaltbarkeit	Standardeinstellung
Gruppe A: Abrechnungsdaten des Abrechnungssystems				
ACLS		Abrechnungs-Abschluss-Satz	Nein	IE
ACLS	FN	Name der Nachfolgerdatei	Nein	IE
AOPN		Abrechnungs-Eröffnungssatz	Nein	IE
AOPN	FN	Name der Vorgängerdatei	Nein	IE
AOPN	MM	Hauptspeicherausbau und Arbeitsspeichergröße	Nein	IE
Gruppe C: reserviert				
Gruppe D: Abrechnungsdaten des DVS				
DALC		Speicherplatz-Allokierungs-Satz	Ja	EA
DALC	FN	Speicherplatz-Änderungsdaten	Nein	IE
DRFA [1]		RFA-Session-Abrechnungssatz	Ja	EA
DRFA [1]	ST	RFA-Session-Beendigungsgrund	Nein	IE
DRVR [1]		Abrechnungsdaten von DRV	Ja	EA
DSPC		Speicherplatz-Bestandsaufnahme-Satz	Ja	IE
DSPC	SP	Speicherplatz-Belegungsdaten	Nein	IE
DSPP		Speicherplatz-Bestandsaufnahme-Satz für Privatplatten	Ja	IE
DSPP	PS	Speicherplatz-Belegungsdaten für Privatplatten	Nein	IE

Gruppe E: Abrechnungssätze von DSSM				
ESMC		Subsystem-Initialisierungs-Abrechnungssatz	Ja	IE
ESMD		Subsystem-Beendigungs-Abrechnungssatz	Ja	IE
Gruppe F: Abrechnungsdaten von File-Transfer				
FTRO [1]			Ja	IE
FTRO [1]	FN	Name der übertragenen Datei	Ja	IE
FTRO [1]	MN	Name der Bibliothek	Ja	IE
FTRO [1]	YY	Jahrhundertanteil bei Zeitangaben	Ja	IE
FTRO [1]	MS	Rechenzeit-Verbrauch	Ja	IE
Gruppe G: reserviert				
Gruppe H: Abrechnungsdaten des Hierarchischen Speicher-Management-Systems				
HSMS [1]		HSMS-Abrechnungssatz	Ja	EA
HSMS [1]	IO	Ein-/Ausgaben pro Gerätegruppe	Ja	EA
Gruppe I: reserviert				
Gruppe J: Abrechnungsdaten der Auftragsverwaltung				
JOBS		Job-Start-Abrechnungssatz	Ja	IE
JOBS	JO	Job-Herkunft	Ja	IE
JOBS	JD	Job-Eigenschaften	Ja	EA
JOBS	JR	Betriebsmittel-Anforderung	Ja	EA
JOBS	JP	Job-Parameter	Ja	EA
Gruppe K reserviert				
Gruppe L: reserviert				
Gruppe M: reserviert				
Gruppe N: reserviert				
Gruppe O: reserviert				
Gruppe P: Programm-Abrechnungsdaten				

PACC		Periodischer Programm-Abrechnungssatz	Ja	EA
PACC	PD	Zeitpunkt des vorhergehenden PACC-Satzes	Ja	IE
PACC	MA	Hintergrundspeicherbelegung	Ja	IE
PACC	IO	E/A und Datenmenge je Gerätegruppe	Ja	IE
PACC	TI	Terminal-Ein-/Ausgaben	Ja	IE
PACC	CA	Anzahl der Katalogzugriffe	Ja	IE
PACC	PC	Performance-Controller-Daten	Ja	IE
PACC	ID	Benutzer-Kennzeichnung	Ja	IE
PDMP		Userdump-Abrechnungssatz	Ja	IE
PRGS		Programm-Start-Abrechnungssatz	Ja	IE
PRGS	PN	Interner Programmname	Ja	IE
PRGS	MA	Hintergrundspeicherbelegung	Ja	IE
PRGS	IO	E/A und Datenmenge je Gerätegruppe	Ja	IE
PRGS	TI	Terminal-Ein-/Ausgaben	Ja	EA
PRGS	CA	Anzahl der Katalogzugriffe	Ja	EA
PRGS	PC	Performance-Controller-Daten	Ja	EA
PRGS	ID	Benutzer-Kennzeichnung	Ja	EA
PRGT		Programm-Beendigungs-Abrechnungssatz	Ja	IE
PRGT	PT	Programm-Beendigungs-Grund	Ja	IE
PRGT	MA	Hintergrundspeicherbelegung	Ja	IE
PRGT	IO	E/A und Datenmenge je Gerätegruppe	Ja	IE
PRGT	TI	Terminal-Ein-/Ausgaben	Ja	EA

PRGT	CA	Anzahl der Katalogzugriffe	Ja	EA
PRGT	PC	Performance-Controller-Daten	Ja	EA
PRGT	EI	Externe Programm-Identifikation	Nein	IE
PRGT	ID	Benutzer-Kennzeichnung	Ja	EA
Gruppe Q: reserviert				
Gruppe R: Betriebsmittel-Verfügbarkeitsdaten				
RCPU		CPU-Verfügbarkeits-Abrechnungssatz	Ja	EA
RSRV		Auftragnehmer-Leistungs-Abrechnungssatz	Ja	EA
RSRV	RD	Betriebsmittel-Beschreibung	Ja	IE
RSRV	SV	Leistungsdaten	Ja	IE
Gruppe S: SPOOL-Abrechnungsdaten				
SPLO		SPOOLOUT-Abrechnungssatz	Ja	IE
SPLO	OT	SPOOLOUT-Beendigungsgrund	Ja	IE
SPLO	OC	SPOOLOUT-Erzeugung	Ja	IE
SPLO	OI	SPOOLOUT-Anstoß	Ja	IE
SPLO	IN	SPOOLOUT-Eingabeband	Ja	IE
SPLO	OM	Ausgabemedium	Ja	IE
SPLO	FN	SPOOLOUT-Dateiname	Ja	IE
Gruppe T: Task-Abrechnungdaten				
TASK		Task-Abrechnungssatz	Nein	IE
TASK	TT	Task-Beendigungs-Grund	Ja	IE
TASK	MA	Hintergrundspeicherbelegung	Ja	IE
TASK	IO	E/A und Datenmenge je Gerätegruppe	Ja	IE
TASK	TI	Terminal-Ein-/Ausgaben	Ja	EA
TASK	CA	Anzahl der Katalogzugriffe	Ja	EA

TASK	PC	Performance-Controller-Daten	Ja	EA
TASK	ID	Benutzer-Kennzeichnung	Ja	EA
TATR		Task-Attribut-Änderungs-Abrechnungssatz	Ja	EA
TDEV		Geräte- und Datenträger-Belegungssatz	[2]	IE
TDEV	D	Unit-Record-Geräte-Belegungsdaten	Ja	IE
TDEV	DV	Volume-Geräte-Belegungsdaten	Ja	IE
TDEV	VU	Datenträger-Belegungsdaten	Ja	EA
TDEV	ID	Benutzer-Kennzeichnung	Ja	EA
Gruppe U: Unprivilegierte Benutzer-Abrechnungsdaten und UTM-Abrechnungsdaten				
UACC		Benutzer-Verbrauchsstempel-Satz	Ja	IE
UACC	ID	Benutzer-Kennzeichnung	Nein	EE
UACC	MA	Hintergrundspeicherbelegung	Ja	IA
UACC	IO	E/A und Datenmenge je Gerätegruppe	Ja	IA
UACC	TI	Terminal-Ein-/Ausgaben	Ja	IA
UACC	CA	Anzahl der Katalogzugriffe	Ja	IA
UACC	PC	Performance-Controller-Daten	Ja	IA
UDAT		Benutzerdaten-Satz	Ja	IE
UDAT	' '[3]	Benutzerdaten-String	Nein	IE
UTMA [1]		UTM-Abrechnungssatz	Ja	IE
UTMK [1]		UTM-Kalkulationssatz	Ja	IE
Gruppe V: Abrechnungsdaten für virtuelle Maschinen				
VACD [1]		VM2000-Abrechnungssatz für Geräte	Ja	IE
VACD [1]	DV	Geräteerweiterung	Nein	IE

VACM [1]		VM2000-Abrechnungssatz für Betriebsmittel	Ja	IE
Gruppe W: reserviert				
Gruppe X: reserviert für Benutzer-Abrechnungssätze				
Gruppe Y: reserviert für Benutzer-Abrechnungssätze				
Gruppe Z: reserviert für Benutzer-Abrechnungssätze				

[1] Diese Abrechnungssätze sind in den jeweiligen Handbüchern der Produkte beschrieben

[2] Wenn alle Satzerweiterungen dieser Abrechnungssätze ausgeschaltet sind, ist die Satzart implizit ausgeschaltet

[3] Erweiterungskennung: zwei Leerzeichen

> **i** Die Unterscheidung bei der Standardeinstellung zwischen explizit/implizit ein- oder ausgeschaltet gilt nur für Satzerweiterungen. Dabei ist zu beachten, dass, wenn in einem Abrechnungssatz eine Satzerweiterung explizit (per Kommando) eingeschaltet ist, alle anderen Satzerweiterungen implizit ausgeschaltet sind und umgekehrt.
> Die einzelnen Abrechnungssätze sind ausführlich im Handbuch „Abrechnungssätze“ [1] beschrieben.

15 Systemmeldungen

Dieses Kapitel gibt eine Einführung zu den Systemmeldungen von BS2000.

Die BS2000-Systemmeldungen sind ein wesentliches Element der Kommunikation zwischen Anwender und Betriebssystem. Sie informieren den Anwender über Fehlersituationen, die während der Bearbeitung seines Jobs auftreten, über die erfolgreiche Beendigung eines Arbeitsabschnittes oder fordern zu einer Maßnahme/Eingabe auf. BS2000-Systemmeldungen haben einen definierten Aufbau, sind in Meldungsdateien gespeichert und werden vom BS2000-Meldungssystem verwaltet.

Kommando	**Bedeutung**
HELP-MSG-INFORMATION	Text einer Systemmeldung und Erläuterungen zu dieser Meldung anfordern
MODIFY-MIP-PARAMETERS	MIP-Parameterdatei erzeugen oder ändern und/oder Meldungsdateien dem System zuschalten oder entziehen
MODIFY-MSG-ATTRIBUTES	Sprache der Meldungsausgabe vereinbaren (tasklokal)
MODIFY-MSG-FILE-ASSIGNMENT	Meldungsdateien für den Systemlauf festlegen
SHOW-MIP-PARAMETERS	Einträge der MIP-Parameterdatei oder Informationen über die Meldungsdateien für den Systemlauf ausgeben
SHOW-MSG-FILE-ASSIGNMENT	Informationen über die Meldungsdateien für den Systemlauf anzeigen
Makro	**Bedeutung**
MSG7 / MSG7X	Meldung ausgeben
MSGSINIT / MSGSMOD	Meldungsdatei sperren oder dem Meldungssystem hinzufügen
MSGSHOW	Informationen über system- oder taskspezifische Meldungen ausgeben

Tabelle 41: Schnittstellenübersicht zum Meldungswesen

15.1 BS2000-Meldungssystem

Kernstück des BS2000-Meldungssystems ist die Systemkomponente MIP (Message Improvement Processing). MIP erhält von einem gerade ablaufenden Modul einer anderen Systemkomponente, eines Subsystems oder eines Benutzerprogramms über den Makro MSG7/MSG7X die Aufforderung, eine bestimmte Meldung zu einem definierten Ausgabeort (Konsole, Datensichtstation) zu senden (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30]) oder in eine S-Variable auszugeben (siehe Abschnitt „Ausgabe von Meldungen in S-Variablen“).

Die Systemkomponente MIP übernimmt bei der Meldungsverarbeitung folgende Aufgaben:

1. Über den Meldungsschlüssel und den Kennbuchstaben für die Sprache wird in den zur Verfügung stehenden Meldungsdateien nach der entsprechenden Meldung gesucht. Die Meldung ist als „Meldungsgerüst“ in der Meldungsdatei abgespeichert.
2. Das „Meldungsgerüst“ wird vervollständigt, d.h. im Meldungstext vorhandene Inserts werden durch aktuelle Werte ersetzt. Diese werden MIP über den Makro MSG7/MSG7X mitgeteilt.
3. Die vollständige Meldung wird zum angegebenen Ausgabeort gesendet.

15.2 Meldungsdateien

Meldungen sind in Meldungsdateien gespeichert. Diese Dateien sind als ISAM-Dateien organisiert, wobei der Meldungsschlüssel zugleich der ISAM-Schlüssel ist.

Meldungsdateien werden mit dem Dienstprogramm MSGMAKER erstellt und bearbeitet, siehe Handbuch „Dienstprogramme“ [15]. Dies gilt sowohl für Meldungsdateien, die vom Hersteller ausgeliefert werden, als auch kundeneigene Meldungsdateien.

15.2.1 Systemmeldungsdateien

Bereits bei der Systemeinleitung können verschiedene Angaben zum Meldungssystem über den Parameterservice vereinbart werden:
Neben der Anzahl der Meldungsdateien, die über die Zugriffsmethode DLAM verarbeitet werden sollen (MSGDLAM) und der Sprache, in der standardmäßig die Meldungsausgabe erfolgen soll (MSGLPRI), legt die Systembetreuung im Parametersatz SYSOPT-CLASS2 vor allem die Anzahl und die Dateinamen der Systemmeldungsdateien fest, die durch den Parameterservice vereinbart werden. Hierbei gilt für die Systemparameter MSGFIL01 und MSGFIL02 folgende Standardeinstellung: MSGFIL01=SYSMES.BS2CP.version bzw. MSGFIL02=SYSMES.EKP.01. Diese Dateien werden bei der Installation mit IMON automatisch eingerichtet.

Die Systembetreuung kann weitere 13 Meldungsdateien vollqualifiziert angeben (MSGFIL03-MSGFIL15).

Es ist möglich, die weiteren Meldungsdateien dynamisch in einer MIP-Parameterdatei zu halten. Diese Parameterdatei kann im Gegensatz zu den im Parametersatz SYSOPT-CLASS2 stehenden Angaben um Meldungsdateien erweitert bzw. reduziert werden.
Die MIP-Parameterdatei wird von der Systembetreuung mit dem Kommando MODIFY-MIP-PARAMETERS angelegt /verwaltet und bekommt standardmäßig den Namen
SYSPAR.MIP.<version>. Sie wird auf der Kennung TSOS abgelegt und hat folgenden Aufbau:

```
MIPPAR=SYSTEM
* this must be the first line
*
* comment lines begin with an "*"
*
MSGFILE=$.SYSMES.product_1.version
MSGFILE=$.SYSMES.product_2.version
:
```

Die MIP-Parameterdatei kann beliebig viele Vereinbarungen enthalten. Da BS2000 insgesamt nur 255 Meldungsdateien verwalten kann, werden überzählige Angaben ignoriert. Es wird eine entsprechende Warnung an der Konsole ausgegeben.

Die MIP-Parameterdatei darf leer sein, muss aber zum Zeitpunkt der Systemeinleitung immer katalogisiert sein. (Ist sie leer, wird ein neuer Dateiname erfragt. Für diesen Fall ist es von Vorteil, eine MIP-Parameterdatei zu haben, in deren erster Zeile „MIPPAR=SYSTEM“ steht.) Eine mit dem Kommando MODIFY-MIP-PARAMETERS eingefügte Meldungsdatei wird an den Beginn der MIP-Parameterdatei geschrieben, gleich nach der Eröffnungszeile und evtl. nachfolgenden Kommentarzeilen. Die MIP-Parameterdatei kann mit dem Kommando SHOW-MIP-PARAMETERS gelesen werden.

Temporär kann mit den Kommandos MODIFY-MSG-FILE-ASSIGNMENT oder MODIFY-MIP-PARAMETERS ..., SCOPE=*TEMPORARY die Menge der Meldungsdateien während der BS2000-Session beeinflusst werden.

Somit hat die Systembetreuung drei Möglichkeiten , die Meldungsdateien für die BS2000-Session festzulegen:

- zum Startup-Zeitpunkt im Parameterservice mit den Systemparametern
- während Startup, aber nachdem MIP geladen wurde, mit der MIP-Parameterdatei
- während der BS2000-Session mit den Kommandos MODIFY-MIP-PARAMETERS und MODIFY-MSG-FILE-ASSIGNMENT

Bei Beendigung des Systemlaufs werden alle Meldungsdateien geschlossen.

15.2.2 Benutzermeldungsdateien

Der nichtprivilegierte Anwender kann für seine Task Meldungsdateien hinzufügen oder entfernen (/MODIFY-MSG-FILE-ASSIGNMENT) und die Sprache der Meldungsausgabe festlegen. Er kann somit auf Fehlersituationen mit eigenen Meldungen reagieren.

Die entsprechenden Kommandos und Makros sind in den Handbüchern „Kommandos“ [27] und „Makroaufrufe an den Ablaufteil“ [30] beschrieben.

15.3 Garantierte Meldungen

Meldungsschlüssel und Inserts diverser Systemmeldungen werden für zukünftige Subsystemversionen bzw. Versionen von BS2000 als unveränderliche Meldungs-Bestandteile garantiert, d.h. weder Meldungsschlüssel noch Nummern und Semantik der Inserts werden in künftigen Versionen verändert. Entsprechende Meldungen werden als **garantierte Meldungen** bezeichnet. Die Auszeichnung dieser Meldungen erfolgt über das Dienstprogramm MSGMAKER.
MIP erzeugt für garantierte Meldungen S-Variablen, über die direkt auf bestimmte Meldungsdaten zugegriffen werden kann; das Ausgabelayout der Meldungen muss nicht bekannt sein. Weitere Informationen hierzu siehe Abschnitt „Ausgabe von Meldungen in S-Variablen".

Garantierte Meldungen werden durch das Meldungsattribut „Garantie“ (siehe "Attribute der Meldungseinheit") dokumentiert.

Beispiel

Die Meldung `JMS0066`, eine garantierte Systemmeldung, wird nach erfolgreicher Bearbeitung des Kommandos ENTER-JOB ausgegeben. Sie enthält vier Inserts und hat folgendes Ausgabelayout:

```
% JMS0066 JOB '(&00)' ACCEPTED ON (&01) AT (&02), TSN = (&03)
```

Über die Inserts wird der Auftragsname - (&00) -, das Datum - (&01) -, die Uhrzeit - (&02) - und die TSN - (&03) - des Jobs ausgegeben. Die Zuordnung der Insertnummern zu ihrem Inhalt ist garantiert (d.h. über die Insertnummer (&03) dieser Meldung wird immer eine TSN des Jobs (&00) ausgegeben werden). Darüber hinaus ist garantiert, dass bei einem erfolgreich abgesetzten Kommando ENTER-JOB immer die Meldung `JMS0066` ausgegeben wird.
Nicht garantiert ist dagegen der Meldungstext, der in einer folgenden Systemversion evtl. auf folgende Art geändert werden kann:

```
% JMS0066 TSN (&03): JOB '(&00)' ACCEPTED AT (&02) ON (&01)
```

Die Reihenfolge der Inserts hat sich zwar geändert, die logischen Zuordnungen der Inserts bleiben jedoch ebenso erhalten wie die Meldungsaussage der gesamten Meldung.

- [ ] Folgende Meldungen von BS2000 sind garantierte Meldungen:

ACS0000	BLP0163	BLP0467	BLP0980	BLS0152	BLS0451
ACS0001	BLP0165	BLP0468	BLP0981	BLS0153	BLS0452
ACS0014	BLP0166	BLP0469	BLP0990	BLS0160	BLS0453
ACS0019	BLP0167	BLP0470	BLP0991	BLS0162	BLS0454
ACS0032	BLP0168	BLP0500	BLP0992	BLS0163	BLS0455
ACS0046	BLP0170	BLP0501	BLP0993	BLS0165	BLS0456
ACS0047	BLP0171	BLP0502	BLP0994	BLS0166	BLS0457
ACS0048	BLP0230	BLP0505	BLP0995	BLS0167	BLS0461
ACS0049	BLP0231	BLP0506	BLP0996	BLS0168	BLS0462
BLP0002	BLP0232	BLP0507	BLP0997	BLS0170	BLS0464
BLP0005	BLP0233	BLP0508	BLS0002	BLS0171	BLS0465
BLP0006	BLP0234	BLP0509	BLS0005	BLS0230	BLS0467
BLP0007	BLP0235	BLP0510	BLS0006	BLS0231	BLS0468
BLP0008	BLP0310	BLP0511	BLS0007	BLS0232	BLS0469
BLP0009	BLP0311	BLP0512	BLS0008	BLS0233	BLS0470
BLP0010	BLP0330	BLP0513	BLS0009	BLS0234	BLS0500
BLP0011	BLP0331	BLP0514	BLS0010	BLS0235	BLS0501
BLP0012	BLP0332	BLP0515	BLS0011	BLS0310	BLS0502
BLP0013	BLP0333	BLP0516	BLS0012	BLS0311	BLS0503
BLP0014	BLP0334	BLP0517	BLS0013	BLS0330	BLS0505
BLP0016	BLP0335	BLP0518	BLS0014	BLS0331	BLS0506
BLP0034	BLP0337	BLP0520	BLS0016	BLS0332	BLS0507
BLP0035	BLP0338	BLP0521	BLS0017	BLS0333	BLS0508
BLP0040	BLP0339	BLP0522	BLS0018	BLS0334	BLS0509
BLP0041	BLP0340	BLP0523	BLS0034	BLS0335	BLS0510
BLP0053	BLP0341	BLP0524	BLS0035	BLS0337	BLS0511
BLP0054	BLP0342	BLP0529	BLS0040	BLS0338	BLS0512
BLP0055	BLP0345	BLP0530	BLS0041	BLS0339	BLS0513
BLP0060	BLP0420	BLP0532	BLS0053	BLS0340	BLS0514
BLP0061	BLP0421	BLP0533	BLS0054	BLS0341	BLS0515
BLP0100	BLP0422	BLP0534	BLS0055	BLS0342	BLS0516
BLP0101	BLP0424	BLP0536	BLS0060	BLS0345	BLS0517
BLP0102	BLP0425	BLP0540	BLS0061	BLS0354	BLS0518
BLP0104	BLP0450	BLP0550	BLS0100	BLS0355	BLS0520
BLP0139	BLP0451	BLP0551	BLS0101	BLS0356	BLS0521
BLP0140	BLP0452	BLP0552	BLS0102	BLS0357	BLS0522
BLP0142	BLP0453	BLP0553	BLS0104	BLS0358	BLS0523
BLP0150	BLP0454	BLP0554	BLS0106	BLS0420	BLS0524
BLP0151	BLP0456	BLP0555	BLS0139	BLS0421	BLS0525
BLP0152	BLP0457	BLP0560	BLS0140	BLS0422	BLS0526
BLP0153	BLP0461	BLP0570	BLS0142	BLS0424	BLS0527
BLP0160	BLP0462	BLP0977	BLS0150	BLS0425	BLS0529
BLP0162	BLP0464	BLP0979	BLS0151	BLS0450	BLS0530

BLS0531	CMD0201	DMS03BE	DMS1D08	EMM2822	EMM2895
BLS0532	CMD0211	DMS051A	DMS1D09	EMM2823	EMM2897
BLS0533	CMD0214	DMS051B	DMS1D0A	EMM2824	EMM2898
BLS0534	CMD0216	DMS051C	DMS1D0B	EMM2825	ESM0201
BLS0535	CMD0221	DMS0533	DMS1D0C	EMM2826	ESM0202
BLS0536	CMD0300	DMS053C	DMS1D0D	EMM2828	ESM0203
BLS0537	CMD0302	DMS053F	DMS1D0E	EMM2829	ESM0204
BLS0538	CMD0490	DMS0574	DMS1D0F	EMM2830	ESM0205
BLS0539	CMD0500	DMS059C	DMS1D10	EMM2832	ESM0206
BLS0540	CMD0508	DMS05BF	DMS1D11	EMM2834	ESM0207
BLS0550	CMD0509	DMS05C3	DMS1D12	EMM2835	ESM0208
BLS0551	CMD0555	DMS05C6	DMS1D13	EMM2836	ESM0209
BLS0552	CMD0557	DMS05F8	DMS1D1F	EMM2837	ESM0210
BLS0553	CMD0601	DMS060A	DSP0006	EMM2838	ESM0211
BLS0554	CMD0671	DMS060B	DSP0007	EMM2839	ESM0212
BLS0555	CMD0672	DMS0666	ECCCOPY	EMM2840	ESM0213
BLS0560	CMD0674	DMS0681	EMM2303	EMM2842	ESM0214
BLS0570	CMD0678	DMS0682	EMM2304	EMM2843	ESM0215
BLS0977	CMD0679	DMS0683	EMM2305	EMM2844	ESM0216
BLS0979	CMD0680	DMS0684	EMM2307	EMM2846	ESM0218
BLS0980	CMD0681	DMS06D5	EMM2314	EMM2847	ESM0219
BLS0981	CMD0682	DMS0800	EMM2316	EMM2848	ESM0220
BLS0990	CMD0687	DMS0801	EMM2317	EMM2850	ESM0221
BLS0991	CMD0688	DMS0808	EMM2320	EMM2851	ESM0222
BLS0992	CMD0689	DMS0811	EMM2350	EMM2853	ESM0223
BLS0993	CMD0690	DMS0812	EMM2800	EMM2855	ESM0224
BLS0994	CMD2203	DMS0815	EMM2801	EMM2857	ESM0225
BLS0995	CMS0002	DMS0816	EMM2802	EMM2858	ESM0226
BLS0996	DMS0350	DMS0949	EMM2803	EMM2860	ESM0228
BLS0997	DMS0351	DMS09B5	EMM2804	EMM2863	ESM0230
CJC0002	DMS0352	DMS09B6	EMM2805	EMM2865	ESM0231
CJC0010	DMS0355	DMS0E49	EMM2806	EMM2867	ESM0232
CJC0011	DMS035A	DMS1343	EMM2807	EMM2870	ESM0233
CJC0020	DMS035B	DMS1373	EMM2808	EMM2877	ESM0235
CJC0021	DMS035C	DMS1374	EMM2809	EMM2878	ESM0237
CJC0022	DMS035D	DMS1376	EMM2810	EMM2880	ESM0238
CJC0025	DMS0362	DMS13C9	EMM2811	EMM2881	ESM0239
CJC0050	DMS0363	DMS13E2	EMM2812	EMM2882	ESM0240
CJC0051	DMS0364	DMS1D01	EMM2813	EMM2885	ESM0241
CJC0070	DMS036B	DMS1D02	EMM2814	EMM2886	ESM0242
CJC0072	DMS036C	DMS1D03	EMM2815	EMM2888	ESM0244
CJC0076	DMS036D	DMS1D04	EMM2817	EMM2889	ESM0245
CMD0001	DMS036E	DMS1D05	EMM2818	EMM2891	ESM0246
CMD0002	DMS037B	DMS1D06	EMM2819	EMM2892	ESM0249
CMD0093	DMS0382	DMS1D07	EMM2821	EMM2894	ESM0252

ESM0253	EXC0315	IDA0300	IMO9029	JMS0024	LMC0093
ESM0254	EXC0316	IDA0301	IMO9030	JMS0025	LMC0095
ESM0256	EXC0317	IDH0101	IMO9031	JMS0043	LMC0102
ESM0257	EXC0318	IDH0102	IMO9032	JMS0044	LMC0129
ESM0259	EXC031A	IDH0103	IMO9033	JMS0045	LMC0151
ESM0260	EXC031F	IDH0104	IMO9034	JMS0046	LMC0163
ESM0262	EXC032A	IDH0105	IMO9035	JMS0066	LMC0199
ESM0267	EXC032B	IDH0106	IMO9036	JMS0067	LMC0201
ESM0269	EXC0354	IDH0107	IMO9037	JMS0100	LMC0211
ESM0288	EXC0371	IDH0108	IMO9038	JMS0101	LMC0213
ESM0292	EXC0419	IDH0109	IMO9039	JMS0105	LMC0214
ESM0322	EXC0707	IDH0110	IMO9090	JMS0110	LMC0238
ESM0332	EXC0713	IDH0111	IMO9091	JMS0111	LMC0274
ESM0432	EXC0714	IDH0112	IMO9092	JMS0113	LMC0286
ESM0600	EXC0755	IDH0113	IMO9100	JMS0130	LMC0301
ESM0603	EXC0773	IDH0114	IMO9101	JMS0131	LMC0302
ESM0604	EXC0861	IDH0115	IMO9103	JMS0133	LMC0303
ESM0610	EXC0862	IDH0116	IMO9200	JMS0143	LMC0304
ESM0612	EXC0863	IMO9001	IMO9201	JMS0171	LMC0310
ESM0613	EXC0896	IMO9002	IMO9202	JMS0264	LMC0311
ESM0614	EXC0897	IMO9003	IMO9203	JMS0265	LMC0312
ESM0615	EXC0908	IMO9004	IMO9204	JMS0268	LMC0411
ESM0670	EXC0926	IMO9005	IMO9205	JMS0522	LMC0412
ESM0671	EXC0927	IMO9006	IMO9206	JPM0204	LMC0413
ESM0672	EXC0930	IMO9007	IMO9207	JPM0205	LMC0509
ESM0673	EXC0931	IMO9008	IMO9208	JPM0500	LMC0510
ESM0674	EXC0936	IMO9009	IMO9209	JVS04A2	LMC0712
ESM0675	GCF1014	IMO9011	IMO9210	JVS04A3	LMC0714
EXC0012	HEL0001	IMO9012	IMO9211	JVS04D0	LMC0721
EXC0123	HEL0002	IMO9013	IMO9212	JVS04D1	LMC1002
EXC0155	HEL0003	IMO9014	IMO9213	JVS04D2	LMC1003
EXC0156	HEL0004	IMO9015	IMO9214	JVS04D3	LMC1004
EXC0269	HEL0005	IMO9016	IMO9215	JVS04D4	MCA0005
EXC0300	HEL0006	IMO9017	JDS0301	JVS04D5	NAM0001
EXC0303	HEL0010	IMO9018	JDS0302	JVS04D6	NAM0002
EXC0304	HEL0011	IMO9019	JDS0303	JVS04D8	NAM0003
EXC0305	HEL0012	IMO9020	JDS0304	JVS0465	NAM0004
EXC0307	HEL0100	IMO9021	JDS0308	JVS0468	NAM0005
EXC0309	HEL0101	IMO9022	JDS0309	JVS0469	NAM0006
EXC030B	HEL0102	IMO9023	JDS0310	LMC0020	NAM0007
EXC030D	HEL0105	IMO9024	JDS0314	LMC0035	NAM0008
EXC0310	HEL0106	IMO9025	JDS0322	LMC0036	NAM0009
EXC0311	HEL0108	IMO9026	JMS0021	LMC0041	LMC0071
EXC0312	HEL0109	IMO9027	JMS0022	LMC0053	LMC0081
EXC0313	HEL0110	IMO9028	JMS0023	LMC0064	LMC0084

NAM0012	NBR0961	NKD0026	NKR0173	NKV0001	SDP0099
NAM0013	NBR0962	NKD0027	NKR0174	NKV0004	SDP0140
NAM0014	NBR1070	NKD0028	NKR0175	NKV0005	SDP0517
NAM0016	NBR1073	NKD0030	NKR0176	NKV0006	SDP1008
NAM0020	NBR1074	NKD0031	NKR0177	NMH1102	SDP1018
NAM0021	NDI0541	NKD0032	NKR0178	NMH1103	SDP1030
NAM0022	NER0000	NKD0033	NKR0179	NMH1104	SPF0300
NAM0023	NER1040	NKD0034	NKR0180	NMH1105	SPF0680
NAM0024	NER1060	NKD0035	NKR0181	NMH1106	SPF0681
NAM0025	NER1500	NKD0036	NKR0182	NMH1108	SPF0682
NAM0026	NER1510	NKD0037	NKR0188	NMH1109	SPF0688
NAM0027	NKA0094	NKD0038	NKR0189	NMH1116	SPF0689
NAM0028	NKA0096	NKD0039	NKR0191	NMH1117	SRM0003
NAM0029	NKD0002	NKD0040	NKR0192	NMH1120	SRM0004
NAM0030	NKD0005	NKD0041	NKR0193	NMH1123	SRM0016
NAM0031	NKD0006	NKR0018	NKR0194	NMH1124	SRM2101
NAM0032	NKD0007	NKR0073	NKR0195	NMH1154	SRM2103
NAM0033	NKD0009	NKR0074	NKR0196	NMH1155	SRM2151
NAM0034	NKD0010	NKR0075	NKR0197	NMH1165	SRM2210
NAM0035	NKD0011	NKR0124	NKR0201	NMH1166	SRM2211
NAM0036	NKD0013	NKR0125	NKR0202	NMH1180	SRM2401
NAM0037	NKA0094	NKR0147	NKS0008	NMH1181	SRM3102
NAM0039	NKA0096	NKR0148	NKS0010	NMH1183	SRM3999
NAM0040	NKD0002	NKR0149	NKS0011	PDT0200	SRM6001
NAM0042	NKD0005	NKR0150	NKS0012	PDT0201	SRM6010
NAM0043	NKD0006	NKR0151	NKS0014	PDT0202	SRM6020
NAM0044	NKD0007	NKR0160	NKS0015	PDT0203	SRM6030
NAM0045	NKD0009	NKR0161	NKS0036	PDT0204	SRM6040
NAM0048	NKD0010	NKR0162	NKS0046	PDT0205	SSM2012
NAM3001	NKD0011	NKR0163	NKS0047	PDT0206	SSM2039
NAM3003	NKD0013	NKR0164	NKS0053	PDT0207	SSM2052
NBR0031	NKD0014	NKR0165	NKS0055	PDT0208	SSM2053
NBR0032	NKD0016	NKR0166	NKS0056	PDT0209	SSM3056
NBR0033	NKD0018	NKR0167	NKS0057	PDT0210	SSM3100
NBR0034	NKD0019	NKR0168	NKS0062	PDT0213	SSM3101
NBR0200	NKD0021	NKR0169	NKS0063	PDT0214	SSM3284
NBR0724	NKD0023	NKR0170	NKS0068	SDP0090	
NBR0874	NKD0024	NKR0171	NKS0069	SDP0091	
NBR0960	NKD0025	NKR0172	NKS0071	SDP0092	

15.4 Meldungssuche

Die Meldungssuche erfolgt zuerst im task-eigenen Speicherbereich; die Suchkriterien sind der Meldungsschlüssel und das im Makro angegebene Sprachkennzeichen. Bleibt im Taskbereich die Suche ohne Erfolg, wird auf Systemebene weitergesucht. Existiert auch dort keine Meldung zu dem angegebenen Meldungsschlüssel und der Sprache, wird die Meldungssuche zuerst mit der taskweit eingestellten Sprache, dann mit der ersten und schließlich mit der zweiten systemweit eingestellten Sprache fortgesetzt. Bleibt die Suche ohne Erfolg, wird die Meldung „MESSAGE UNDEFINED“ ausgegeben.
Bild 23 skizziert diesen Vorgang, ohne alle in der Beschreibung genannten Einzelheiten abzubilden.

Meldungssuche auf Taskebene

Auf Taskebene wird als Erstes der task-eigene Meldungspuffer durchsucht, in welchem die zuletzt ausgegebenen Meldungen abgespeichert sind. Ist die angeforderte Meldung dort abgelegt, wird sie aufbereitet und ausgegeben. Andernfalls wird überprüft, ob die Bereichszuordnungsliste (Class List) die angegebene Meldungsklasse enthält. Die Class List enthält zu allen „taskbekannten“ Meldungsklassen die Namen der entsprechenden Meldungsdateien. Besonders häufig auszugebende Meldungen werden in der Meldungsdatei mit DLAM-Zugriff spezifiziert. Sie werden zum Zeitpunkt der Verknüpfung der Meldungsdatei mit der Task in den Task-DLAM-Bereich übertragen. Ist die Meldungsklasse vorhanden, so wird im nächsten Schritt die Meldungsdatei nach dem Meldungsschlüssel und dem Sprachkennzeichen durchsucht.
Wird die Meldung gefunden, so wird der taskeigene Meldungspuffer mit dem „Meldungsgerüst“ aktualisiert, das „Meldungsgerüst“ mit den vom Makro MSG7/MSG7X übergebenen aktuellen Einfügungen ergänzt und die Meldung ausgegeben.

Meldungssuche auf Systemebene

Ist die Meldung in keiner task-eigenen Meldungsdatei zu finden, wird der Suchvorgang auf Systemebene im systemeigenen Meldungspuffer fortgesetzt. Entspricht auch hier keine Meldung den Vorgaben, wird die MIP-Task aufgerufen, um mit der Meldungssuche in den Systemmeldungsdateien fortzufahren. Auch hier wird, analog zur Task-Ebene, zuerst der DLAM-Bereich überprüft, bevor die Meldungsdateien durchsucht werden.

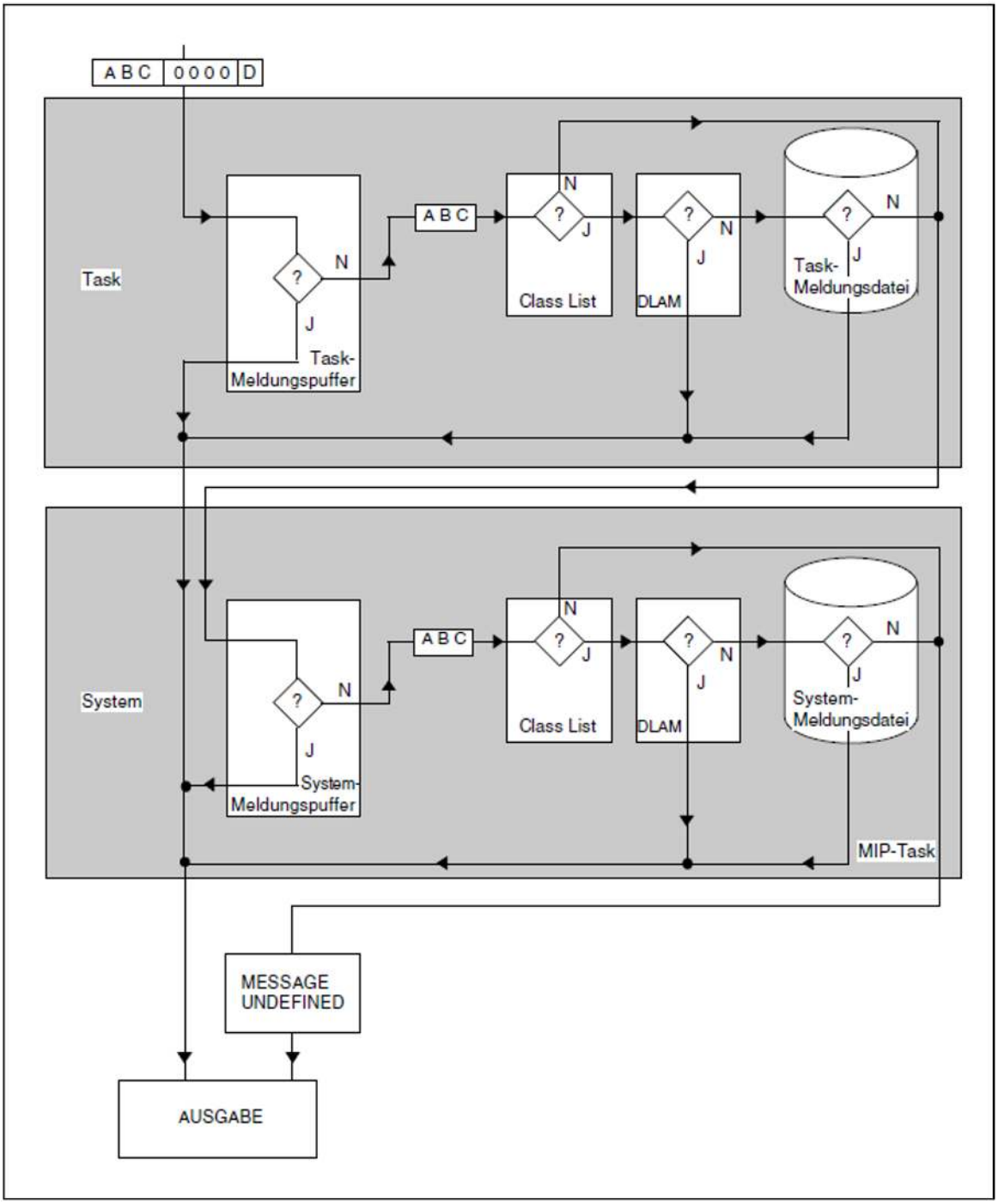


Bild 23: Meldungssuche im BS2000

15.5 Struktur einer Meldungseinheit

Jede mit dem Dienstprogramm MSGMAKER erstellte Meldungsdatei setzt sich aus einzelnen Meldungseinheiten zusammen, auf die über einen siebenstelligen Meldungsschlüssel zugegriffen wird. Die Meldungseinheit besteht aus Meldungsattributen, Meldungstext und ggf., um ein in der Meldung zitiertes Problem näher zu erläutern und mögliche Maßnahmen aufzuzeigen, einem Bedeutungs- und Maßnahmetext. Alle Texte können in maximal acht Sprachen eingetragen werden. Jede Sprache wird durch ein einstelliges Buchstabenkürzel, dem Kennbuchstaben für die Sprache, dargestellt.
Über die zugeordneten Meldungsattribute kann die Meldungsausgabe gesteuert werden.

Es folgt ist der Aufbau einer Meldungsdatei, wie sie mit dem Kommando `/SHOW-FILE <meldungsdatei>` zu sehen ist. Nach der Identifikationszeile erscheinen Meldungen, die für den DLAM-Zugriff spezifiziert sind. Sie werden zum Zeitpunkt der Verknüpfung der Meldungsdatei mit dem System/der Task in den System/Task-DLAM-Bereich übertragen.

```
ID   MSG C    TESTPRODUKT   V1.1

DLAA       0002

AAA0001D  U   Meldungstext in Deutsch zu AAA0001 mit ersetzten Inserts
AAA0001E  U   Meldungstext in Englisch zu AAA0001 mit ersetzten Inserts
AAA0002D  U   Meldungstext in Deutsch zu AAA0002 mit ersetzten Inserts
AAA0002E  U   Meldungstext in Englisch zu AAA0002 mit ersetzten Inserts

AAA0001        Insertname
AAA0002        Insertname

AAA0001D  U   Bedeutungstext in Deutsch zu AAA0001
AAA0001E  U   Bedeutungstext in Englisch zu AAA0001
AAA0002D  U   Bedeutungstext in Deutsch zu AAA0002
AAA0002E  U   Bedeutungstext in Englisch zu AAA0002

AAA0001D  U   Massnahmetext in Deutsch zu AAA0001
AAA0001E  U   Massnahmetext in Englisch zu AAA0001
AAA0002D  U   Massnahmetext in Deutsch zu AAA0002
AAA0002E  U   Massnahmetext in Englisch zu AAA0002

AAA   AAA0001AAA0002 Meldungsverantwortlicher Team Kommentar
```

Bild 24: Ausgabe einer Meldungsdatei mit dem Kommando /SHOW-FILE

15.6 Meldungsschlüssel

Der Meldungsschlüssel ist siebenstellig, wobei die ersten drei Buchstaben für die Meldungsklasse und die letzten vier Zeichen für die Meldungsnummer stehen.
Die Meldungsklasse bezeichnet eine BS2000-Komponente, ein Subsystem, Modul o. Ä.; die Meldungsnummer dient der laufenden Nummerierung innerhalb einer Meldungklasse.In der Regel besteht die Meldungsnummer aus vier Ziffern. Meldungen, deren Meldungsnummern auch Buchstaben enthalten, werden in die Meldungsdatei vor solchen Meldungen eingefügt, welche Ziffern beinhalten.

Beispiel

Die Meldungsschlüssel ABC123A bis ABC123Z werden vor den Meldungsschlüsseln ABC1230 bis ABC1239 in die Meldungsdatei eingefügt.

15.6.1 Kennbuchstabe für die Sprache

Ein alphabetisches Zeichen dient zur Kennzeichnung der Sprache, in welcher Meldungs-, Bedeutungs- und Maßnahmetext erstellt sind. Der Kennbuchstabe D wird für Deutsch und E für Englisch verwendet, alle weiteren Sprachen sind mit beliebigen Buchstaben abkürzbar. Maximal acht Sprachen können definiert werden. Der Kennbuchstabe dient in Verbindung mit dem Meldungsschlüssel zur Identifikation eines Meldungs-, Bedeutungs- und Maßnahmetextes innerhalb der Meldungseinheit.

15.6.2 Attribute der Meldungseinheit

MIP-Zugriffsmethode für die Meldung

Es gibt folgende MIP-Zugriffsmethoden für Meldungen: ISAM, DLAM, LOCAL-DLAM, BAMR und MINIMIP. Die Zugriffsmethoden LOCAL-DLAM, BAMR und MINIMIP sind für den internen Gebrauch reserviert.

Ausgabeziel der Meldung

Dokumentiert ein oder mehrere Ausgabeziele der Meldung. Die Angabe wird nicht ausgewertet, sollte jedoch mit dem Operanden DEST des Makros MSG7/MSG7X übereinstimmen.

Berechtigungsschlüssel (Routing-Code)

Ist das Ausgabeziel einer Meldung eine Konsole, erhält sie ein zusätzliches, einstelliges Kennzeichen, den so genannten Berechtigungsschlüssel (BS), auch Routing-Code (RC) genannt. Er wird bei den Meldungen als Zielangabe ausgewertet, wenn neben der Hauptkonsole auch Nebenkonsolen verwendet werden.

Beim Betrieb mit mehreren Konsolen werden diese bestimmten Aufgabengebieten zugeteilt. Über den Berechtigungsschlüssel können dann bestimmte Meldungen vom System den richtigen Konsolen zugeordnet werden.
Der mit MSGMAKER definierte Berechtigungsschlüssel wird nur verwendet, wenn dem Operanden UCDEST im Makro MSG7/MSG7X kein Wert zugewiesen wird.

BS / RC	Aufgabengebiet
A	Systemverwaltung
C	Bedienung des Datenkommunikationssystems
D	Bedienung von Plattenspeichern
E	Allgemeine Aufgaben und Befugnisse
G	Geräteverwaltung
H	Hardware-Wartung
J	Steuerung und Kontrolle von Jobs
K	OPR-Kommandoverwaltung
N	Überwachung von Remote-Spool
O	Bedienung von Schnelldruckern
P	Steuerung von Tasks
R	Überwachung und Steuerung des Systemverhaltens
S	Steuerung des SPOOLOUT-Betriebs, SPOOL laden
T	Bedienung von Magnetbandgeräten
U	Dateiverwaltung

W,X,Y,Z	Zur freien Verfügung für den Benutzer

0	Meldungen mit diesem Berechtigungsschlüssel sind nicht bestellbar. Meldungen ohne Antwort werden nur in der CONSLOG-Datei hinterlegt. Meldungen mit Antwort (Fragen) werden der Hauptkonsole zugestellt.
9	VM2000-Verwaltung
#	POSIX
*	Hauptkonsole
@	Hauptkonsole (nur bei Meldungen, die eine Antwort erfordern; Meldungen ohne Antwort werden in der CONSLOG-Datei hinterlegt, aber nicht ausgegeben)
$	Sonderbedeutung; wird nicht für Meldungsausgaben verwendet
B,F,I,L,M,Q,V, 1,2,3,4,5,6,7,8	Reserviert für künftige Systemerweiterungen

Tabelle 42: Aufgabengebiete und Berechtigungsschlüssel (BS / RC) von Meldungen

Siehe auch Tabelle 45 ("Aufgabengebiete und ihre Aufteilung auf Konsolen").

Weight Code

Der Weight Code bestimmt das Meldungsgewicht.

Ist das Ausgabeziel einer Meldung eine Konsole, muss ein Weight Code vergeben werden, andernfalls ist eine Zuweisung ohne Bedeutung. Meldungen, die nicht an eine Konsole gehen, kann WEIGHT=*NONE zugewiesen werden. Wird eine Meldung mit anders lautender Zielangabe trotzdem an eine Konsole geschickt, erhält sie den Standard Weight Code 99.

Werte von 0 bis 99 können vergeben werden, wobei Weight Code 99 einer Meldung äußerste Wichtigkeit bescheinigt. Die Meldung kann nicht unterdrückt werden.
Erfordert die Meldung eine Antwort (Operand REPLY im Makro MSG7/MSG7X) sollte der Weight Code 90 oder 98 zugewiesen werden.

Zur Filterung von Meldungen können an der Konsole per Kommando vier Filterstufen eingestellt werden, welche den Weight Codes der Meldungen folgendermaßen zugeordnet sind:

Filterstufe	Weight Code	Standardwerte des Herstellers
1	0 - 19	Keine
2	20 - 39	30
3	40 - 59	50
4	60 - 79	70
nicht unterdrückbar	80 - 99	90,98,99

ℹ Die in den ausgelieferten Meldungsdateien eingestellten Meldungsgewichte (Vorbesetzung) können von der Systembetreuung über die MSGMAKER-Anweisung //MODIFY-MSG geändert werden.

Filterstufe 1

Der Benutzer kann nachträglich Meldungen, die seiner Meinung nach bedeutungslos sind, mit einem Gewicht dieses Bereichs versehen und durch Zuordnung der Filterstufe 1 für die betreffende Konsole/den berechtigten Benutzer-Job eine dortige Ausgabe verhindern.

Filterstufe 2

Eine Meldung mit einem Weight Code aus diesem Bereich ist rein informativ und kann ohne Gefährdung des sicheren Betriebs unterdrückt werden.

Filterstufe 3

Eine Meldung aus diesem Bereich ist informativ. Der sichere Betrieb wird nicht gefährdet, wenn sie unterdrückt wird. Der erhöhte Wert erklärt sich durch einen Zusatzaspekt, durch den eine Ausgabe ratsam sein kann, z.B. Protokollierung aus Datenschutzgründen, Bearbeitung durch einen berechtigten Benutzer-Job.

Filterstufe 4

Eine Meldung mit einem Weight Code aus diesem Bereich ist informativ und erfordert keine Antwort. Aber die Information ist für den Operator wichtig. Ihre Unterdrückung könnte unter Umständen negative Auswirkungen auf den Systemlauf haben.

Weight Code-Bereich 80 - 99

Eine Meldung mit einem Weight Code dieses Bereichs erfordert entweder eine Antwort oder ist von größter Wichtigkeit.

- Weight Code 90
  Mit diesem Gewicht werden Antwortmeldungen versehen, für die eventuell später in Verbindung mit gesicherten Automatisierungsfunktionen eine Unterdrückung denkbar wäre. Zum aktuellen Zeitpunkt sind sie nicht unterdrückbar.
- Weight Code 98
  Eine Meldung mit diesem Weight Code erfordert eine Antwort und darf in keinem Fall unterdrückt werden. Weight Code 98 ist von der Wichtigkeit her gleichzusetzen mit dem Weight Code 99.
- Weight Code 99
  Eine Meldung mit diesem Weight Code ist von höchster Wichtigkeit und darf in keinem Fall unterdrückt werden.

Garantie

Das Meldungsattribut „Garantie“ wird von MIP ausgewertet.

Das Attribut garantiert, dass bei einer damit versehenen Meldung weder Meldungsschlüssel noch Inserts (Nummern und Semantik) in den kommenden Subsystemversionen bzw. Versionen von BS2000 OSD/BC geändert werden.

MIP erzeugt für garantierte Meldungen S-Variablen. Nähere Informationen hierzu siehe Abschnitt „Ausgabe von Meldungen in S-Variablen".

15.6.3 Meldungstext

Zu jedem definierten Meldungsschlüssel muss ein Meldungstext erstellt werden, der in acht Sprachen abgefasst werden kann. In den Meldungstext können maximal 30 Inserts (siehe unten) eingefügt werden.
Der Meldungstext ist mit dem Meldungsschlüssel und Kennbuchstaben für die Sprache in der Meldungsdatei abgespeichert (siehe Bild 24 ("Struktur einer Meldungseinheit")).
Bei der Erstellung eines Meldungstextes sind folgende Besonderheiten zu beachten:

- Trennungszeichen „^“

  Der Meldungstext wird als fortlaufende Textzeile eingegeben. Um eine strukturierte Ausgabe zu erhalten, kann das Trennungszeichen „^“ an jeder beliebigen Stelle und beliebig oft in den Meldungstext eingefügt werden. Text, der auf dieses Trennungszeichen folgt, wird in der jeweils nächsten Zeile ausgegeben. Mit der SHOW-Funktion von MSGMAKER kann die Textausgabe überprüft werden.

- Beschränkung des Zeichenvorrats

  Die Meldungsausgabe kann auf unterschiedlichen 7-Bit-Terminals erfolgen. Abweichende Codier-Tabellen haben eine doppelte Belegung von Zeichen zur Folge.

  > **i** Folgende Zeichen sind sowohl im Meldungstext als auch im Bedeutungs- und Maßnahmetext verboten, wenn die Texte in den Standardsprachen Deutsch und Englisch erstellt sind. MSGMAKER gibt eine Warnung aus, falls eines dieser Zeichen verwendet wurde. Die Einschränkungen gelten nicht für alle zusätzlich definierten Sprachen.

Code	**Zeichen**						
Hex-Code X' __'	4F	BB	BC	BD	FB	FD	FF
Internationaler EBCDIC.DF.03	|	\	[	]	<	>	~
Deutscher EBCDIC.DF.03	ö	Ö	Ä	Ü	ä	ü	ß

  Leerzeichen am Ende des Meldungstextes werden unterdrückt. Der letzte Satz oder Satzteil eines Meldungstextes schließt ohne Satzzeichen ab.

- Inserts (Einfügungen)

  Definition von Inserts

  Maximal 30 verschiedene Inserts können sprachübergreifend in den Meldungstexten einer Meldungseinheit definiert werden.
  Die Zeichenfolgen (&00), (&01) bis (&29) werden in den Meldungstext eingefügt, wobei 00 bis 29 die Nummern der Inserts darstellen. Sie können im Text in beliebiger Reihenfolge auftreten (00, dann 02, dann 01) und müssen nicht lückenlos definiert werden (00, 08, 11).
  Entscheidend für die Meldungsausgabe ist, dass die definierten Inserts im Makro MSG7/MSG7X aufgeführt werden. Enthält der Meldungstext mehr Inserts als im Makroaufruf angegeben, wird für jede übrige Einfügung der Standard-Text übernommen. Ist kein Standard-Text vorhanden, unterscheidet man zwei Fälle:

  1. Die Nummer des übrigen Inserts (z.B. (&02)) liegt zwischen den Insertnummern, die im Makroaufruf angegeben sind (z.B. (&00),(&01),(&03)). In diesem Fall wird ein Leerstring ausgegeben.
  2. Die Nummer des übrigen Inserts (z.B. (&03)) ist höher als die Insertnummern, die im Makroaufruf angegeben sind (z.B. (&00),(&01),(&02)). Der Insert wird nicht ersetzt, d.h. es wird (&03) ausgegeben.

  Ist im Makro MSG7/MSG7X eine Angabe zu einer Insertnummer gemacht, die im Meldungstext nicht definiert ist, ignoriert MIP die Angabe im Makro MSG7/MSG7X. Es erscheint keine Fehlermeldung.
  Die mit MSGMAKER in der Meldung angegebenen Standard-Texte ersetzen die definierten Inserts nur, wenn im Makro MSG7/MSG7X kein aktueller Text dafür vereinbart ist.

  Insertnamen

  Für jeden Insert kann zusätzlich ein mnemotechnischer Insertname vergeben werden. Insertnamen werden zur Bildung von S-Variablen verwendet.
  Die vergebenen Insertnamen werden immer in Großbuchstaben umgewandelt. Weitere Informationen zur Bildung von S-Variablen siehe Abschnitt „Ausgabe von Meldungen in S-Variablen".

  Standard-Text für Insert

  Text, der an Stelle der Inserts (&nn) in den Ausgabetext eingesetzt wird. Der Standard-Text ist Inhalt der S-Variablen I0 (bis I29).

  Automatic help

  Die automatische Hilfefunktion (Funktion /HELP-MSG-INFORMATION) wird für den Teil des Meldungsschlüssels aktiviert, der als Insertwert geliefert wird.

  > **i** *Darstellung von Inserts in Anweisungen* Wird ein Insert mit einer Anweisung (//ADD-MSG bzw. //MODIFY-MSG) in einem Meldungstext vereinbart, muss das Zeichen & doppelt eingegeben werden.

- Länge des Meldungstextes

  Der Meldungstext kann maximal 220 Zeichen, einschließlich der Zeichenfolgen (&00) ... (&29), lang sein. Die aktuellen Inhalte, welche bei der Meldungsausgabe diese Zeichenfolgen ersetzen sollen, sind in dieser Längenangabe nicht inbegriffen. Der vollständige Meldungstext kann folglich mehr als 220 Zeichen umfassen.

  Beispiel für einen Meldungstext

  ```
  MSG-TEXT =THIS MESSAGE WILL BE SENT TO THE DESTINATION "(&04)",^
  IN LANGUAGE "(&00)", "(&01)" OR "(&02)"
  ```

  Maximale Textlänge bei der Meldungsausgabe auf die Datensichtstation

  Die maximale Textlänge, die von MSG7/MSG7X ausgegeben wird, darf 4079 Zeichen nicht übersteigen. Dies beinhaltet den Meldungstext (ohne Meldungsschlüssel), die Insertwerte (ggf. die Standardwerte), die Datums- und Zeitangabe (wenn gewünscht).

  Generell gilt: Übersteigt ein Meldungstext die maximale Textlänge, so wird dieser Text bei der Ausgabe abgeschnitten. Dabei gibt MSGMAKER keine Warnung aus.

  Maximale Textlänge bei der Meldungsausgabe auf die Konsole

  Die maximale Textlänge, die von MSG7/MSG7X an die Konsole ausgegeben wird, beträgt 230 Zeichen, einschließlich des vorangestellten Prozentzeichens % und des Meldungsschlüssels.

  MSGMAKER überprüft die Länge des Meldungstextes und der Standard-Inserts. Bei Überschreitung der Textlänge von 230 Zeichen gibt MSGMAKER eine Warnung aus.

  Bedingt durch die aktuellen Insertwerte oder die Optionen des Makros MSG7/MSG7X kann die Textlänge variieren. Bei Angabe von TEXT-ONLY wird der Meldungsschlüssel nicht ausgegeben, durch Angabe von DATESTAMP und TIMESTAMP kann die Datums- und Zeitangabe vor dem Meldungstext variieren.

15.6.4 Bedeutungs-/Maßnahmetext

Zur Erläuterung des Meldungstextes kann ein erklärender Text formuliert werden, der auf Anforderung ausgegeben wird. Weiterhin können Antworten oder Maßnahmen angeboten werden, mit deren Hilfe das im Meldungstext beschriebene Problem gelöst werden kann. Bedeutungs- und Maßnahmetexte werden in der vorher festgelegten Sprache ausgegeben.

Über das Kommando /HELP-MSG-INFORMATION wird ein solcher Bedeutungs- und Maßnahmetext nach SYSOUT ausgegeben (siehe "Kommando HELP-MSG-INFORMATION").
Sowohl Bedeutungs- als auch Maßnahmetext können aus maximal 256 Zeilen zu je 74 Zeichen bestehen. Inserts sowie das Trennungszeichen „^“ werden im Text akzeptiert, haben jedoch nicht die Sonderbedeutung, die sie im Meldungstext besitzen.
Die letzte Zeile des Bedeutungs- und Maßnahmetextes sollte mit einem Punkt enden.

Analog zum Meldungstext gelten die Beschränkungen des Zeichenvorrats.

15.7 Meldungsausgabe

- Ausgabeformen der Meldungen
- Kommando HELP-MSG-INFORMATION
- Meldungen, die eine Antwort erfordern
- Ausgabe von Meldungen an Benutzerprogramme
- Ausgabe von Meldungen in S-Variablen

15.7.1 Ausgabeformen der Meldungen

Als Ausgabeform für Meldungen kann generell gewählt werden:

- **Normalform**
  umfasst den Meldungsschlüssel, Meldungstext ggf. mit aktuellen Einfügungen
- **Kurzform**
  umfasst den Meldungsschlüssel und ggf. die aktuellen Einfügungen, aber keinen weiteren Meldungstext
- **Langform**
  umfasst die Normalform plus Erläuterungstext (Bedeutungs- und Maßnahmetext)

Die Ausgabeform der Meldungen kann der Anwender über den Operanden INFORMAT-ON-LEVEL des Kommandos /MODIFY-JOB-OPTIONS ändern.
Vereinbarungen, die mit /SET-LOGON-PARAMETERS oder /ENTER-JOB getroffen wurden und die die Protokollierung eines Jobs betreffen, werden überschrieben.

INFORMATION-LEVEL=*MEDIUM bewirkt die ungekürzte Ausgabe der Meldungen (Normalform).

INFORMATION-LEVEL=*MINIMUM bewirkt die Ausgabe der Meldungen in codierter Form (Kurzform).

Beispiel

Zur Demonstration der unterschiedlichen Ausgabeformen von Systemmeldungen wird im Job JOB.TEST versucht, die Attribute der nicht katalogisierten Datei ABCD auszugeben. Die Systemmeldungen werden in Normal- und Kurzform in die Datei SYSOUT.JOB.TEST protokolliert.
Der Job wird mit dem Kommando /ENTER-JOB FROM-FILE=JOB.TEST gestartet.

Job JOB.TEST

```
/SET-LOGON-PARAMETERS
/ASSIGN-SYSOUT TO=SYSOUT.JOB.TEST
/SHOW-FILE-ATTRIBUTES FILE-NAME=ABCD
/SET-JOB-STEP
/MODIFY-JOB-OPTIONS INFORMATION-LEVEL=*MINIMUM
/SHOW-FILE-ATTRIBUTES FILE-NAME=ABCD
/EXIT-JOB
```

SYSOUT-Protokoll SYSOUT.JOB.TEST

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=ABCD
%  DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET '2OSG'. COMMAND TERMINATED
%  CMD0205 ERROR IN PRECEDING COMMAND OR PROGRAM AND PROCEDURE STEP
TERMINATION                                                                1.
/SET-JOB-STEP
/MODIFY-JOB-OPTIONS INFORMATION-LEVEL=*MINIMUM                             2.
/SHOW-FILE-ATTRIBUTES FILE-NAME=ABCD
%  DMS0533 2OSG
%  CMD0205                                                                 3.
/EXIT-JOB
%  EXC0419 /LOGOFF AT <time> ON <date> FOR TSN '2EZD'
%  EXC0421 CPU TIME USED: 0.5752
```

1. Ausgabe der Systemmeldungen in Normalform, da INFORMATION-LEVEL= *MEDIUM Voreinstellung ist.
2. Ausgabeform wird auf INFORMATION-LEVEL=*MINIMUM gesetzt.
3. Ausgabe der Systemmeldungen in codierter Kurzform.

15.7.2 Kommando HELP-MSG-INFORMATION

Dieses Kommando wird verwendet, um sich zu einer erhaltenen Meldung Erläuterungstexte (Bedeutungs- und Maßnahmetext) ausgeben zu lassen. Das Kommando kann ohne und mit Operanden abgesetzt werden (siehe Handbuch „Kommandos“ [27]).

/HELP-MSG-INFORMATION ohne Operanden gibt die vom System zuletzt ausgegebene Meldung (ggf. mit aktuellen Einfügungen) zusammen mit dem Bedeutungs- und Maßnahmetext in der gleichen Sprache nochmals aus.

/HELP-MSG-INFORMATION mit Operanden ermöglicht die gezielte Angabe eines Meldungsschlüssels (Operand MSG-IDENTIFICATION), des Informationsumfangs (Operand INFORMATION-LEVEL) und der Sprache (Operand LANGUAGE).

Beispiel

```
/help-msg-information msg-id=dms0533
%  DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET '(&00)'. COMMAND TERMINATED
%  ? This message is issued by DMS commands. The requested file is not
%    cataloged in the requested pubset.
%  RESPONSE : NONE %                                                    1.
/help-msg-information msg-id=dms0533,language=d
%  DMS0533 ANGEGEBENE DATEI IN PUBSET '(&00)' NICHT GEFUNDEN. KOMMANDO
%  BEENDET
%  ? Diese Meldung wird von DVS-Kommandos ausgegeben. Die angeforderte Datei
%    ist im gewuenschten Pubset nicht katalogisiert.
%  MASSNAHME : KEINE                                                  2.
/help-msg-information msg-id=dms0533,inf-level=*medium
%  DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET '(&00)'.
%  COMMAND TERMINATED                                                 3.
/help-msg-information msg-id=dms0533,inf-level=*minimum
%  DMS0533 (&00)                                                      4.
```

1. Ausgabe der Systemmeldung in Langform. INFORMATION-LEVEL=*MAXIMUM ist Voreinstellung.
2. Englisch wurde als Standardwert für die Sprache festgelegt. Wird die Ausgabe der Meldung in Deutsch gewünscht, muss dies extra angegeben werden.
   Mit dem Kommando /MODIFY-MSG-ATTRIBUTES kann der Anwender für seine Task die Sprache vereinbaren, in der die Meldungen ausgegeben werden sollen.
3. Ausgabe der Systemmeldung in Normalform, Sprache Englisch ist Standardwert.
4. Ausgabe der Systemmeldung in codierter Kurzform.

15.7.3 Meldungen, die eine Antwort erfordern

Bei Meldungen, die eine Stellungnahme erfordern, bietet MIP als Standard-Antwort ein Fragezeichen an. Das Fragezeichen ist ein reserviertes Schlüsselwort bei MIP. Setzt der Benutzer ein Fragezeichen ab, gibt MIP den Bedeutungs- und Maßnahmetext zu dieser Meldung aus und wiederholt die Frage.

```
%  CMD0213 TERMINATE PROGRAM? (Y=YES; N=NO)
?
                                                                      1.
%  ? The user requested program termination by hitting a
%    function key or by NEXT line input.
%  ! Y: causes the END statement to be executed
%    N: termination request is ignored
%  CMD0213 TERMINATE PROGRAM? (Y=YES; N=NO)                           2.
```

1. Der Benutzer gibt ein Fragezeichen ein. MIP gibt den Bedeutungs- und Maßnahmetext zu dieser Meldung aus.
2. Der im Anschluss daran erneut ausgegebene Meldungstext lässt sich vom Anwender nun leichter beantworten.

15.7.4 Ausgabe von Meldungen an Benutzerprogramme

Benutzern ist es möglich, sich mit dem Makroaufruf MSG7/MSG7X Meldungen vom System in einen Speicherbereich ihres Benutzerprogramms schreiben zu lassen (Operand BUFFER). Nähere Informationen siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30].

15.7.5 Ausgabe von Meldungen in S-Variablen

Meldungen können in die Systemdateien SYSOUT bzw. SYSLST ausgegeben werden. Das Softwareprodukt SDF-P bietet ergänzend hierzu die Möglichkeit, **garantierte Meldungen** in strukturierte S-Variablen auszugeben (siehe auch Abschnitt „Garantierte Meldungen").
SDF-P übernimmt dabei die Steuerung des S-Variablenstroms SYSMSG, über den die garantierten Meldungen in S-Variablen umgelenkt werden. Nähere Informationen zum Konzept der strukturierten S-Variablenströmen siehe Handbuch „SDF-P“ [45].
Die Systemkomponente MIP erzeugt die S-Variablen, deren Werte als Eingabedaten in S-Prozeduren weiterverwendet werden können.

Das Bild unten zeigt sowohl die Ausgabe nach SYSOUT durch den Systemdateien-Manager als auch die Umlenkung der Ausgabeinformation in die S-Variablenströme SYSMSG (und SYSINF), welche durch SDF-P gesteuert wird. Weitere Informationen zu SYSINF siehe Handbuch „Kommandos“ [27].

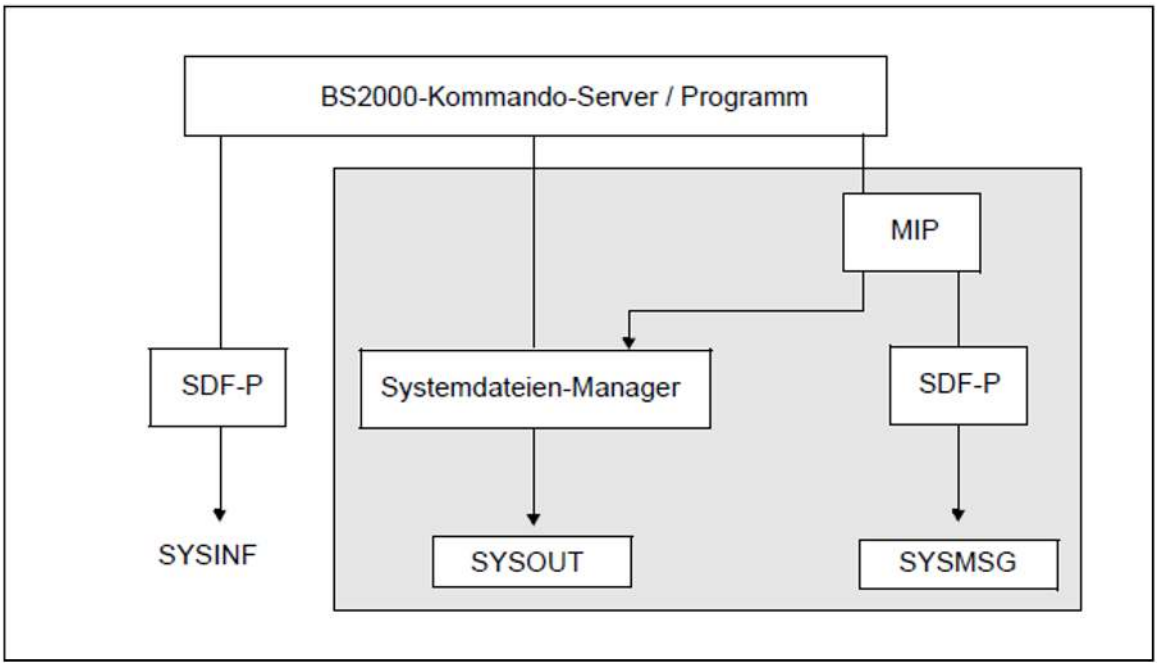


Bild 25: Ausgabe von Meldungen in den S-Variablenstrom SYSMSG und nach SYSOUT

Dem Benutzer erwachsen aus der Verwendung von S-Variablen entscheidende Vorteile:

- Die S-Variablen erlauben einen einfachen Zugriff auf eine garantierte Meldung, da die Information strukturiert abgespeichert ist. Der Anwender ist unabhängig vom Ausgabelayout der Meldung.
- Die Namen der S-Variablen und ihre inhaltliche Zuordnung sind auch in kommenden Subsystemversionen bzw. Versionen von BS2000 garantiert.
- Der Inhalt der S-Variablen kann als Eingabe in S-Prozeduren weiterverwendet werden.

Folgende Angaben sind erforderlich, um garantierte Meldungen in eine strukturierte S-Variable auszugeben:

1. Der Anwender muss eine Listenvariable vom Typ „Struktur“ deklarieren.

   ```
   /DECLARE-VARIABLE <var-name>(TYPE=*STRUCTURE),MULTIPLE-ELEMENT=*LIST
   ```

2. Der Anwender muss den S-Variablenstrom SYSMSG für die strukturierte Ausgabe in S-Variablen zuweisen. Eine der folgenden Vereinbarungen muss getroffen werden:

Auf Kommandoebene:

- Kommando EXECUTE-CMD vereinbart die strukturierte Ausgabe in S-Variablen für ein Kommando.

  `/EXEC-CMD (<cmd-name>),MSG-STRUCTURE-OUTPUT=<var-name>`

- Kommando ASSIGN-STREAM; die Zuweisung des S-Variablenstroms ist für alle nachfolgenden Kommandos gültig, solange bis die Vereinbarung wieder zurückgenommen wird (`ASSIGN-STREAM SYSMSG,TO=*STD`).

  `/ASSIGN-STREAM STREAM-NAME=SYSMSG,TO=*VARIABLE(<var-name>)`

Auf Programmebene:

Der Makro CMD vereinbart die strukturierte Ausgabe in S-Variablen für mehrere Kommandos.

`CMD '<cmd-name>',...,VER=4,MSGVAR@=<adr>,MSGVARL=<len>,[,MSGEXT=YES]`

`<adr>` ist die symbolische Adresse, die den Namen der S-Variablen enthält. Makro CMD, siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30].

3. Im Makro MSG7X muss das Ziel der Meldungsausgabe vereinbart werden.
   Wird DESTINATION=SYSOUT vereinbart, werden garantierte Meldungen immer in S-Variablen ausgegeben, vorausgesetzt der Benutzer hat eine S-Variable deklariert. Darüber hinaus kann dem Operanden BUFFER ein benutzereigener Speicher und BUFFUSE=EXTERNAL zugewiesen werden.
   Makro MSG7X, siehe Handbuch „Makroaufrufe an den Ablaufteil“ [30].
4. Mit MSGMAKER muss die Meldung folgendermaßen ausgezeichnet werden:
   a. Die auszugebende Meldung muss als garantiert ausgezeichnet sein (//ADD-MSG MSG-ID=..., WARRANTY=YES). Für garantierte Meldungen erzeugt MIP folgende S-Variablen:
      - MSG-ID:
        enthält den Meldungsschlüssel.
      - REPLY:
        enthält die Antwort auf die Meldung, sofern diese vom Benutzer eine Stellungnahme fordert.
      - I0 bis I29:
        Standard-Namen der Inserts; I0 bis I29 werden standardmäßig von MIP als S-Variablennamen vergeben, wenn der Benutzer mit MSGMAKER keine Insertnamen (siehe b)) vereinbart hat.
      - MSG-TEXT:
        enthält den Meldungstext inklusive aller ersetzten Inserts.
   b. Für die Inserts der auszugebenden Meldung werden Namen vereinbart (//ADD-MSG MSG-ID=...,INSERT-ATTRIBUTES=...(NAME=...)).
      Ist die Meldung nicht garantiert, wird jedoch mindestens einem Insert der auszugebenden Meldung ein Name zugewiesen, so werden folgende S-Variablen erzeugt:
      - <insert-name>:
        Mit MSGMAKER vereinbarter Name wird in Großbuchstaben umgewandelt und von MIP als S-Variablenname verwendet.
      - MSG-ID:
        enthält den Meldungsschlüssel.
      - REPLY:
        enthält die Antwort auf eine Meldung.
      - MSG-TEXT:
        enthält den Meldungstext inklusive aller ersetzten Inserts.

> Die von MIP erzeugten S-Variablen sind vom Datentyp STRING.

Die endgültige, von MIP ausgegebene S-Variable setzt sich aus zwei Teilnamen zusammen:

- Name der vom Benutzer deklarierten S-Variablen: <var-name>
  (<var-name> ist der erste Teilname der S-Variablen)
- MIP liefert den zweiten Teilnamen (MSG-ID, I0 bis I29, <insert-name>, REPLY, MSG-TEXT)

Die beiden Teilnamen bilden, durch einen Punkt voneinander getrennt, die Namen der S-Variablen

- <var-name>.MSG-ID
- <var-name>.I0 ... <var-name>.I29
- <var-name>.<insert-name>
- <var-name>.REPLY
- <var-name>.MSG-TEXT

Alle Variablen (MSG-ID, I0, ...), die es zu einer garantierten Meldung gibt, gehören einem Listenelement an. Jede weitere Meldung wird als neues Listenelement aufgenommen. Dies ist der Fall, wenn zu einem Kommando mehrere garantierte Meldungen existieren (Aufruf des Kommandos mit /EXECUTE-CMD) oder wenn mit dem Kommando /ASSIGN-STREAM allgemein die Ausgabe garantierter Meldungen in eine S-Variable umgelenkt wird.

Beispiel

Das folgende Beispiel zeigt die Umlenkung der Meldungsausgabe in eine S-Variable.
Die Attribute der nicht existierenden Datei „TEST“ sollen ausgegeben werden. Die dadurch verursachte Fehlermeldung DMS0533, eine garantierte Meldung, wird in die S-Variable MSG ausgegeben:

```
/DECLARE-VARIABLE NAME=MSG(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC)),
   MULTIPLE-ELEMENTS=*LIST                                          1.
/EXECUTE-CMD CMD=(SHOW-FILE-ATTRIBUTES FILE-NAME=TEST),TEXT-OUTPUT=*NONE,
   MSG-STRUCTURE-OUTPUT=MSG(WRITE-MODE=*EXTEND)                     2.
/SHOW-VAR MSG                                                       3.
MSG(*LIST).MSG-TEXT = %  DMS0533 REQUESTED FILE NOT CATALOGED IN
   PUBSET '1SBZ'. COMMAND TERMINATED
MSG(*LIST).MSG-ID = DMS0533
MSG(*LIST).I0 = 1SBZ                                                4.
```

1. Die S-Variable MSG wird als dynamisch erweiterbare Listenvariable deklariert.
2. Mit dem Kommando /EXECUTE-CMD wird die Ausgabe des Kommandos /SHOW-FILE-ATTRIBUTES in die S-Variable MSG umgelenkt. Die Ausgabe nach SYSOUT wird unterdrückt.
3. Mit SHOW-VAR MSG wird die Listenvariable MSG ausgegeben.
4. MIP erzeugt für diese garantierte Meldung die S-Variablen MSG-TEXT, MSG-ID und I0; die Werte dieser S-Variablen bilden das erste Listenelement der S-Variablen MSG. Wird das Kommando /EXECUTE-CMD wiederholt abgesetzt, wird jede weitere Meldung als neues Listenelement angefügt, da WRITE-MODE=*EXTEND angegeben ist (siehe 2.).

16 Operatorfunktionen

Die Tätigkeiten, die ein Operator während des Systemlaufs auszuführen hat, stammen aus mehreren Aufgabengebieten. Zu diesen zählen z.B.:

- die Überwachung und Steuerung des Systemverhaltens
- die Überwachung und Steuerung von Anwendungen
- die Steuerung des Datenkommunikationssystems
- die Bedienung von Peripheriegeräten

Operating und Konsolen

Die Arbeitsplätze, von denen die Bedienung des Systems – das sog. Operating – erfolgt, werden Konsolen genannt. In den BS2000-Systemmeldungen werden dafür auch noch die Synonyme „Bedienstation“, „Bedienplatz“ und „Bedienungsplatz“ verwendet.

Mit dem Start des Datenkommunikationssystems eröffnet das System automatisch die BCAM-Anwendung $CONSOLE. Benutzerprogramme, die sich an $CONSOLE anschließen, können auch Funktionen des Operating wahrnehmen. Sie werden deshalb ebenfalls Konsolen genannt.

Zur Unterscheidung dienen die Begriffe „physikalische Konsole“ und „logische Konsole“.

Das Operating kann getrennt nach Aufgabengebieten auf mehrere Konsolen verteilt werden. Dies können auch logische Konsolen sein, wie sie zum Beispiel von den Softwareprodukten MAREN und ROBAR verwendet werden, um Operating-Aufgaben für Bänder zu verteilen.

Das Privileg OPERATING kann mit dem SECOS-Kommando SET-PRIVILEGE einer beliebigen Benutzerkennung zugewiesen werden. Auch von einer Benutzertask dieser Benutzerkennung kann das Operating erfolgen.

Physikalische Konsolen

Physikalische Konsolen waren in der Vergangenheit spezielle Hardware-Geräte, die direkt an den BS2000-Server angeschlossen waren.

Heute werden diese Geräte im SE Manager oder in VM2000 emuliert. Die Emulation eines Konsol-Bildschirms am SE Manager erfolgt durch das KVP (Konsol-Verteil-Programm), siehe Handbuch „Bedienen und Verwalten“ [57].

Jeder physikalischen Konsole ist eine eindeutige mnemotechnische Gerätebezeichnung (Mnemonik, MN) zugeordnet. Die Emulationen im SE Manager oder in VM2000 erlauben das Öffnen mehrerer Instanzen auf einer physikalischen Konsole.

Von physikalischen Konsolen aus sind die Eingabe von Kommandos und der Austausch von Nachrichten mit dem BS2000-System auch in den Betriebsphasen möglich, in denen BCAM im angesprochenen BS2000-System nicht zur Verfügung steht.

Genau eine der physikalischen Konsolen ist die Hauptkonsole. Sie verfügt über Sonderrechte. Standardmäßig ist dies die IPL-Konsole, von der die Systemeinleitung stattgefunden hat. Die Eigenschaft Hauptkonsole kann während des Systemlaufs aber auch auf eine andere physikalische Konsole übergehen.

In Abhängigkeit von dem Systemparameter NBCONOPI können zwei Betriebsmodi eingestellt werden: Ist eine Anmeldung (das so genannte Operator-Logon) mit einer Operator-Kennung für physikalische Konsolen erforderlich (NBCONOPI=Y), werden die Berechtigungen der physikalischen Konsolen mit Hilfe der Operator-Rollen gesteuert.

Ist kein Operator-Logon für physikalische Konsolen erlaubt (NBCONOPI=N), dann werden Berechtigungsschlüssel (Routing-Codes) über den Parameterservice (Parametersatz
OPR) oder das ASR-Kommando erteilt.

Als Operator-Kennung wird eine beliebige Benutzerkennung mit dem Privileg OPERATING bezeichnet.

> Bei NBCONOPI=Y arbeitet der Operator unter einer eigenen Benutzerkennung und verfügt über einen eigenen Dateiraum. Für RUN-Dateien erfolgt die Ergänzung des Dateinamens mit dieser Benutzerkennung und nicht mehr mit TSOS.

Wenn das Operator-Logon eingeschaltet ist (NBCONOPI=Y), dann ist nach „System Ready“ Operating von einer physikalischen Konsole nur noch nach erfolgreicher Anmeldung unter einer Kennung mit dem Privileg OPERATING möglich (mit SET-LOGON-PARAMETERS).

Vor dem Logon ist nur das Kommando SHOW-PENDING-MSG zur Anzeige offener Fragen erlaubt. Nach erfolgreicher Anmeldung hat der Operator noch keine Berechtigungen in Form von von Berechtigungsschlüsseln (Routing-Codes). Diese besorgt er sich mit dem Kommando REQUEST-OPERATOR-ROLE. Mit dem Kommando SHOW-OPERATOR-ROLE werden die aktuellen Berechtigungen der eigenen Kennung angezeigt.

Zur Abmeldung vom System ist das Kommando EXIT-JOB einzugeben. Es impliziert auch die Rückgabe aller Berechtigungsschlüssel (RELEASE-OPERATOR-ROLE OPERATOR-ROLE=*ALL).

Die Kennung SYSOPR mit den Privilegien OPERATING und STD-PROCESSING ist immer im System vorhanden.

Logische Konsolen bzw. $CONSOLE-Anwendungen

„Logische Konsolen“ sind Benutzerprogramme, die mit Hilfe von DCAM oder sonstigen Netzwerkfunktionen eine Verbindung mit der Systemanwendung $CONSOLE hergestellt haben und über diese Verbindung mit der UCON-Task kommunizieren. Deswegen werden sie auch als „$CONSOLE-Anwendungen“ bezeichnet. In Abhängigkeit von der Art der Anmeldung bei $CONSOLE unterscheidet man zwischen „unberechtigten“ und „berechtigten“ Anwendungen.

Logische Konsolen können nur betrieben werden, wenn BCAM im BS2000-System zur Verfügung steht.

Berechtigte logische Konsolen haben - analog der MN von physikalischen Konsolen - einen eindeutigen Berechtigungsnamen, anhand dessen sie identifiziert werden können. Dabei wird zwischen statisch generierten Berechtigungsnamen und dynamisch zugewiesenen Berechtigungsnamen unterschieden.

Mit berechtigten $CONSOLE-Anwendungen können bestimmte Aufgaben der Systembedienung automatisiert werden. Sie werden dann z.B. als „programmierter Operator“ oder „automatischer Operator“ bezeichnet. Sie können aber auch eine interaktive Systembedienung über eine Netzwerkverbindung ermöglichen.

Die Beschreibung der $CONSOLE-Schnittstelle für logische Konsolen erfolgt im Kapitel „Automatisierung von Operatorfunktionen".

16.1 Benutzertask mit OPERATING-Privileg

Die Bedienung des Systems kann auch von Benutzertasks betrieben werden, die das Privileg OPERATING besitzen. Mit SECOS kann das Privileg OPERATING an eine beliebige Kennung vergeben werden. Die System-Benutzerkennung SYSOPR besitzt standardmäßig die Privilegien OPERATING und STD-PROCESSING.

Der von einem Operator-Arbeitsplatz (bisher nur Konsolen) bekannte Datenstrom lässt sich logisch in einen Dialogstrom (Dialog Stream) entsprechend der aktiven Systembedienung und einen Ereignisstrom (Event Stream) entsprechend der reaktiven Systembedienung aufteilen.

Der **Dialogstrom** ist der kommandogebundene Anteil des Datenstroms eines Operator-Arbeitsplatz. Er umfasst die Abwicklung des Kommandodialogs, d.h. die Eingabe von Kommandos und die Ausgabe der kommandogebundenen Meldungen sowie u. U. das Beantworten von Zusatzinformationsanforderungen der Kommandobearbeitung.
Die systeminterne SCI-Schnittstelle (Synchronous Console Interface) ermöglicht die Eingabe aller OPERATING-Kommandos aus einer Benutzertask.

Die Kommandoeingabe im Dialogstrom erfolgt synchron, d.h. die Eingabe eines Kommandos ist erst nach Beendigung des vorhergehenden möglich. Dies ist ein wesentlicher Unterschied zum Kommandodialog an Konsolen, wo die Eingabe der Kommandos asynchron erfolgt, d.h. unabhängig von der Beendigung evtl. vorausgegangener Kommandos ist Kommandoeingabe möglich. Die synchrone Arbeitsweise des Dialogstroms ermöglicht den Einsatz von SDF-P für Operating-Prozeduren in Benutzertasks mit Privileg OPERATING.

Der **Ereignisstrom** ist der systembedingte, asynchrone Anteil des Datenstroms eines Operator-Arbeitsplatzes. Er umfasst im Wesentlichen freilaufende Systemmeldungen (von den Konsolen bekannt), auf welche der Operator ggf. reagieren muss. Die Integration dieser Funktionalität in einer Benutzertask erfolgt im Rahmen des **Ereignisstrom-Service** (ESS, Event-Stream-Service), der auch für Benutzer ohne OPERATING-Privileg neue Möglichkeiten eröffnet. Ein wesentliches Merkmal des Ereignisstrom-Service ist die Möglichkeit, Nachrichten des Systems asynchron, also sofort nach deren Erzeugung, an einer Datensichtstation innerhalb einer Dialogtask zu präsentieren.

Zu diesem Zweck wird eine zentrale System-Ereignisstrom-Datei (System Event Stream File, SESF) angelegt, in welcher alle System-Ereignisse von der Systemtask SEST aufgezeichnet werden.
Der Name der System-Ereignisstrom-Datei ist $SYSAUDIT.SYSLOG.ESS.SYSTEM.
Sie wird beim Hochfahren des Systems angelegt und während der Systembeendigung wieder gelöscht.
Über den Systemparameter NBESSIZE lässt sich die Maximalgröße der System-Ereignis-strom-Datei einstellen.
Die Voreinstellung beträgt 40000 PAM-Seiten. Die max. Aufbewahrungsdauer der aufgezeichneten Ereignisse beträgt 96 Stunden. Nach Ablauf dieser Zeit werden sie aus Platzgründen gelöscht.

System-Ereignisse sind grundsätzlich alle freilaufenden, über Routing-Code verteilten und nicht kommandogebundenen Systemmeldungen, die auch in der CONSLOG-Datei enthalten sind. Mit Hilfe des Kommandos ASSIGN-SYSEVENT können sich Benutzertasks mit dem Privileg OPERATING einen eigenen System-Ereignisstrom zuordnen. Für diese werden zusätzlich an die eigene Task gerichtete asynchrone Nachrichten und – abhängig von der im Kommando ASSIGN-SYSEVENT getroffenen Auswahl – auch Anteile des eigenen Kommandodialogs als System-Ereignisse in der System-Ereignisstrom-Datei aufgezeichnet.

Die Anzeige der aufgezeichneten Ereignisse übernimmt eine Präsentationsfunktion mit maskenorientierter Darstellung unter Nutzung des Subsystems FHS-TPR. Mit dem Kommando SHOW-SYSEVENT-LOG wird die Präsentationsfunktion des Ereignisstroms gestartet. In Benutzertasks mit Privileg OPERATING können alle im System vorhandenen System-Ereignisströme angezeigt werden. An die eigene Task gerichtete, asynchrone Nachrichten und protokollierte Anteile des Kommandodialogs werden grundsätzlich angezeigt. Der Umfang der zusätzlich dargestellten System-Ereignisse wird durch das so genannte „Betrachterprofil“ bestimmt, das vor Aufruf der Präsentationsfunktion eingestellt werden muss. Das Betrachterprofil wird durch die Gesamtheit der mit Hilfe der Operator-Rollen eingestellten Routing-Code-Menge und den lokalen Einstellungen zur Meldungsunterdrückung, Meldungsfilterung und Meldungsbestellung bestimmt.

Die Präsentationsfunktion bietet einen dynamischen und einen statischen Modus an. Der dynamische Modus ermöglicht die unverzügliche Darstellung neu eintreffender, asynchroner Ereignisse (wie von der Konsole bekannt), bietet aber keine Dialogfunktion.
Der statische Modus der Präsentationsfunktion bietet einen maskengesteuerten Benutzerdialog mit Möglichkeiten zum Blättern und Positionieren innerhalb des gesamten Ereignisstroms, mit einer Suchfunktion, einer Funktion zum Beantworten offener Systemfragen und ein Hilfesystem. In Abhängigkeit der tasklokal eingestellten Meldungssprache erfolgt die Präsentation in Englisch oder Deutsch.

Zur Aufbereitung der Ausgabemasken nutzt die Präsentationsfunktion das Subsystem FHS-TPR, welches in der Regel ab „System Ready“ verfügbar ist. Der notwendige Zugriff auf die FHS-Maskenbibliotheken SYSFHS.BS2CP.<version>.E bzw.
SYSFHS.BS2CP.<version>.D und SYSFHS.FHS-TPR.<version>.E bzw.
SYSFHS.FHS-TPR.<version>.D wird bei der Systeminstallation ermöglicht.

Das Kommando SHOW-SYSEVENT-LOG-ATTRIBUTES erteilt Auskünfte über die Eigenschaften und Attribute der zugreifbaren Ereignisströme.

Beliebige Benutzertasks können sich eigene Benutzer-Ereignisströme (User Event Stream) zuordnen, die keinerlei Operating-Funktionalität enthalten.
Für diese erfolgt allein die Aufzeichnung der mittels Kommando ASSIGN-SYSEVENT eingestellten Ereignisse in einer privaten Benutzer-Ereignisstrom-Datei (UESF, User Event Stream File). Die Datei wird unter der eigenen Kennung angelegt und hat den Namen SYSLOG.ESS.USER.<log-id>.

Benutzer-Ereignisströme eignen sich vor allem zum Aufzeichnen und Darstellen von asynchronen Nachrichten an die Benutzer (z.B. INFORM-JOB bzw. INFORM-ALL-JOBS) und zum Überwachen von Abläufen, die sich hinter einem Auftragskommando (z.B. ENTER-JOB) verbergen. Die Präsentationsfunktion wird ebenfalls mit dem Kommando SHOW-SYSEVENT-LOG gestartet. Alle vorhandenen Benutzer-Ereignisströme der eigenen Kennung können angezeigt werden.

Bei der Systembedienung von Konsolen ist der Operator an eine gemischte Darstellung von Dialogstrom und Ereignisstrom gewöhnt. Bei der Systembedienung aus einer Benutzertask ist diese Mischform der Darstellung aus technischen Gründen nicht möglich.
Zur Realisierung eines komfortablen Arbeitsplatzes sind mindestens zwei Dialogtasks an zwei Terminals oder – innerhalb einer Terminal-Emulation – die Mehrfenstertechnik zu empfehlen. Eine Dialogtask dient der Bedienung des Dialogstroms, die andere Dialogtask der Präsentation des Ereignisstroms. Sind weitere Dialogtasks bzw. Präsentationsfenster verfügbar, können mehrere Präsentationsfunktionen mit unterschiedlichen Betrachterprofilen entsprechend unterschiedlichen Arbeitsgebieten parallel dargestellt werden. Steht nur eine einzige Dialogtask zur Verfügung, ist trotzdem Systembedienung aus der Benutzertask möglich, nur ist in diesem Fall das Wechseln zwischen Dialogstrom und Ereignisstrom (Starten und Beenden der Präsentationsfunktion je nach Bedarf) notwendig.

16.1.1 Bereitstellung einer Kennung für das Operating

Zur Bereitstellung der Operating-Funktionalität von physikalischen Konsolen (bei eingeschaltetem Operator-Logon), $CONSOLE-Anwendungen mit dynamischen Berechtigungsnamen und aus Benutzertasks sind die im Folgenden beschriebenen Vorbereitungen zu treffen. Zur Nutzung der vorhandenen System-Benutzerkennung SYSOPR ist nur das Einrichten und Zuordnen von Operator-Rollen notwendig.

Unter der Kennung TSOS

- Einrichten einer Kennung:

  `/ADD-USER USER-ID=OPEROPER, LOGON-PASSWORD=C'12345678', ACCOUNT=account`
- Wenn das kostenpflichtige Produkt SECOS im Einsatz ist, kann für $CONSOLE-Anwendungen mit dynamischen Berechtigungsnamen ein erweiterter Zugangsschutz mit folgendem Kommando festgelegt werden:

```
/MODIFY-LOGON-PROTECTION USER-ID=OPEROPER, ...,
                         OPERATOR-ACCESS-TERM= ...,
                         OPERATOR-ACCESS-PROG= ...
```

Unter der Kennung SYSPRIV

- Erteilen des Privilegs OPERATING (für $CONSOLE-Anwendungen nicht unbedingt erforderlich):

  `/SET-PRIVILEGE PRIVILEGE=OPERATING, USER-ID=OPEROPER`
- Einrichten von Operator-Rollen:

  `/CREATE-OPERATOR-ROLE OPERATOR-ROLE=OPERROLE, ROUTING-CODES=<list>`
- Zuordnen der Operator-Rollen zu der Kennung:

  `/MODIFY-OPERATOR-ATTRIBUTES USER-ID=OPEROPER,ADD-OPERATOR-ROLE=OPERROLE`
- Falls gewünscht, zusätzliches Zuordnen der Operator-Rollen zur Kennung des System-Operators:

  `/MODIFY-OPERATOR-ATTRIBUTES USER-ID=SYSOPR,ADD-OPERATOR-ROLE=OPERROLE`

Nach diesen Vorbereitungen und der Eingabe des Kommandos SET-LOGON-PARAMETERS mit Benutzerkennung, Abrechnungsnummer und Kennwort ist es für einen Operator unter Berücksichtigung der jeweiligen Besonderheiten möglich, von einer physikalischen Konsole, von einer $CONSOLE-Anwendung mit dynamischem Berechtigungsnamen oder von einer Benutzertask aus Operating zu betreiben. Die Berechtigungsschlüssel fordert der Operator mit dem Kommando REQUEST-OPERATOR-ROLE an.

16.1.2 Nutzung des Ereignisstrom-Service für Operating aus Benutzertasks

Die Nutzung für einen komfortablen Operator-Arbeitsplatz basiert auf mindestens zwei Benutzertasks. Eine Benutzertask (Benutzertask-1) wird zur Eingabe von Kommandos und zum Empfang der zugehörigen Kommandoergebnisse verwendet (Dialogstrom). Die zweite Benutzertask (Dialogtask-2) ist notwendigerweise eine Dialogtask und dient der Präsentation der asynchronen Systemmeldungen (Ereignisstrom).

Dialogstrom (Benutzertask-1):

- Die Eingabe aller mit dem Privileg OPERATING geschützten Kommandos ist erlaubt.

  Die Berechtigungsprüfung erfolgt anhand der SDF-Syntax-Dateien (einschließlich einer evtl. vorgeschalteten Gruppen-Syntax-Datei).

  Bei Kommandos, die über die interne SCI-Schnittstelle der UCON-Task in einer Operator-Task (Umgebung) abgewickelt werden, erfolgt keine Berechtigungsprüfung anhand der Authorization-Codes in den Kommandotabellen der UCON-Task.
  Ausnahme: Mit dem Authorization-Code „$“ geschützte Kommandos sind auch aus Benutzertasks verboten.
- Einrichten eines System-Ereignisstroms zum Empfang asynchroner Systemmeldungen:

  ```
  /ASSIGN-SYSEVENT TO=*SYSTEM (LOG-ID=OPER),ADD-SYNCH-EVENTS=*NONE
  ```

  Mit ADD-SYNCH-EVENTS=*ALL kann zusätzlich der gesamte Kommandodialog der Benutzertask-1 im Ereignisstrom OPER protokolliert werden. Mit dem Kommando ASSIGN-SYSEVENT ..., ADD-SYNCH-EVENTS wird der Umfang der zu protokollierenden Ereignisse (Nachrichten) eingestellt.

  Die an die Benutzertask-1 direkt gesendeten asynchronen Nachrichten werden grundsätzlich protokolliert. Je nach Einstellung (Kommando MODIFY-MSG-OPTIONS) werden diese zusätzlich auch an der Datensichtstation ausgegeben. Die Ausgabe an der Datensichtstation kann durch folgende Einstellung unterdrückt werden:

  ```
  /MODIFY-MSG-OPTIONS OPERATOR-BROADCAST=*NO,OPERATOR-MSG=*NO,SYSTEM-MSG=*NO
  ```
- Die Präsentation des Ereignisstroms in derselben Task ist möglich, aber nicht empfehlenswert, da während der Präsentation keine Kommandoeingabe möglich ist.
- Die Präsentation des Ereignisstroms erfolgt aus der zentralen System-Ereignisstrom-Datei. Diese enthält – im Rahmen der durch den Systemparameter NBESSIZE festgelegten Größenordnung – alle notwendigen (bei der Präsentation auswählbaren) Nachrichten der letzten Stunden des Systemlaufs (max. 96 Stunden).

Präsentation des Ereignisstroms (Dialogtask-2):

- Die Dialogtask-2, ebenfalls unter einer Kennung mit Privileg OPERATING, dient nur der Präsentation des Ereignisstroms. Der in Benutzertask-1 zugewiesene Ereignisstrom ist für Dialogtasks aller Kennungen mit Privileg OPERATING zugreifbar und anhand seines Namens (LOG-ID) eindeutig identifizierbar.
- Für die Präsentation werden aus der Gesamtheit der aufgezeichneten Ereignisse über die tasklokale Einstellung des Betrachterprofils die darzustellenden Ereignisse anhand der nachfolgenden Kriterien ausgewählt. Die Aufzählung erfolgt in der Reihenfolge ihrer Priorität, d.h. sofern ein Kriterium erfüllt ist, werden die folgenden Ereignisse nicht mehr geprüft:
  1. Protokollierte Anteile des Kommandodialogs von Benutzertask-1 werden immer dargestellt.
  2. Direkt an die Benutzertask-1 gesendete, asynchrone Nachrichten werden dargestellt.
  3. Anhand ihrer MSG-ID unterdrückte System-Nachrichten (Kommandos SET- und RESET-MSG-SUPPRESSION) werden nicht dargestellt, sofern es keine Fragen sind.
  4. Anhand ihrer MSG-ID bestellte System-Nachrichten (Kommando MODIFY-MSG-SUBSCRIPTION) werden dargestellt.
  5. Mit Hilfe der eingestellten Operator-Rollen (Kommandos REQUEST- und RELEASE-OPERATOR-ROLE) werden über Routing-Code verteilte Systemmeldungen zur Darstellung ausgewählt.
  6. Bei über Routing-Code verteilten Meldungen werden die eingestellten RoutingCode-spezifischen Filterstufen (Kommandos ADD- und REMOVE-CONSOLE-FILTER) gegen das Meldungsgewicht geprüft.
  7. Ereignisstrom-spezifische Meldungen (z.B. erfolgreich durchgeführte Recovery oder Reorganisation) werden unabhängig von taskspezifischen Einstellungen immer angezeigt.

- Gestartet wird die Präsentationsfunktion mit dem Kommando

  `/SHOW-SYSEVENT-LOG LOG-ID=OPER(TYPE=*SYSTEM)`

  Die Präsentationsfunktion meldet sich mit ihrer Hauptmaske im statischen Modus und bietet Eingabemöglichkeiten über Menü, Kommandozeile und F-Tasten. Die aktuelle Maske positioniert immer auf die jüngsten Ereignisse.

  Folgende Funktionen werden im statischen Modus angeboten:

  - Positionieren und Blättern innerhalb des Ereignisstroms
  - Suchen
  - Beantworten offener Systemfragen
  - Wechseln in den dynamischen Modus
  - Bereitstellen eines Hilfesystems

  Der dynamische Modus bietet die für die Systembedienung wichtigste Funktionalität:
  die sofortige Ausgabe asynchron eintreffender Ereignisse wie von der Konsole bekannt. Die einzig mögliche Eingabe ist die K2-Taste zur Rückkehr in den statischen Modus.

Wird nur die Präsentation der System-Ereignisse ohne „Privatanteile“ gewünscht, ist das Einrichten eines System-Ereignisstroms mit dem Kommando ASSIGN-SYSEVENT nicht erforderlich. Der direkte Aufruf der Präsentationsfunktion mit

`/SHOW-SYSEVENT-LOG LOG-ID=*SYSTEM-MSG-ONLY`

zeigt die dem „Betrachterprofil“ des Operators entsprechende Auswahl an System-Ereignissen.

Für eine ausführliche Beschreibung der Präsentationsfunktion wird auf die Beschreibung des Kommandos SHOW-SYSEVENT-LOG verwiesen.

16.1.3 Nutzung des Ereignisstrom-Service für beliebige Benutzertasks

Die Funktionalität des Ereignisstrom-Service ist für jede beliebige Benutzertask nutzbar, jedoch um die spezifische Operating-Funktionalität eingeschränkt. Es ist sinnvoll, pro Arbeitsplatz mindestens zwei Benutzertasks zu verwenden. Eine Benutzertask (Benutzertask1) wird zur Eingabe von Kommandos und zum Empfang der zugehörigen Kommandoergebnisse verwendet. Die zweite Benutzertask (Dialogtask-2) ist notwendigerweise eine Dialogtask und dient der Präsentation asynchroner, an die Benutzertask gerichteter Nachrichten (z.B. INFORM-ALL-JOBS), sowie von Anteilen des Kommandodialogs von Benutzertask-1 je nach Einstellung.

Für jeden Benutzer-Ereignisstrom erfolgt die Aufzeichnung der Ereignisse in einer eigenen Benutzer-Ereignisstrom-Datei in einer eigenen Batchtask der gleichen Kennung. Der Name der Datei ist SYSLOG.ESS.USER.log-id. Den Suffix des Dateinamens bildet der Name (LOG-ID) des Ereignisstroms. Der Dateizugriff ist nur über die Präsentationsfunktion möglich. Die Lebensdauer der Batchtask entspricht der aktiven Phase des Ereignisstroms (vom Starten bis zum Beenden bzw. Schließen). Die Benutzer-Ereignisstrom-Datei wird beim erstmaligen Starten des Ereignisstroms erzeugt und beim endgültigen Beenden gelöscht.

Der Name des Ereignisstroms (LOG-ID) ist innerhalb der Kennung eindeutig. Der Zugriff auf einen Benutzer-Ereignisstrom ist von allen Dialogtasks der gleichen Kennung möglich. Es gibt keine Sonderberechtigung für die Kennung TSOS.

Dialogstrom (Benutzertask-1):

- Einrichten eines Benutzer-Ereignisstroms zum Empfang asynchroner Nachrichten:

  ```
  /ASSIGN-SYSEVENT TO=*USER (LOG-ID=MYES), ADD-SYNCH-EVENTS=*NONE
  ```

  Mit ADD-SYNCH-EVENTS=*ALL kann zusätzlich der gesamte Kommandodialog der Benutzertask-1 im Ereignisstrom MYES protokolliert werden. D.h. mit Hilfe des Kommandos ASSIGN-SYSEVENT und dem Operanden ADD-SYNCH-EVENTS wird der Umfang der zu protokollierenden Ereignisse (Nachrichten) eingestellt.

  Die an die Benutzertask-1 direkt gesendeten asynchronen Nachrichten werden grundsätzlich protokolliert. Je nach Einstellung (Kommando MODIFY-MSG-OPTIONS) werden diese zusätzlich auch an der Datensichtstation ausgegeben. Die Ausgabe an der Datensichtstation kann durch folgende Einstellung unterdrückt werden.

  ```
  /MODIFY-MSG-OPTIONS OPERATOR-BROADCAST=*NO.OPERATOR-MSG=*NO,SYSTEM-MSG=*NO
  ```

- Die Präsentation des Ereignisstroms in derselben Task ist möglich, aber nicht empfehlenswert, da während der Präsentation keine Kommandoeingabe möglich ist.

Präsentation des Ereignisstroms (Dialogtask-2):

- Die Dialogtask-2, ebenfalls unter der gleichen Kennung dient nur der Präsentation des Ereignisstroms. Der in Benutzertask-1 zugewiesene Ereignisstrom ist anhand seines Namens (LOG-ID) eindeutig identifizierbar.
- Es werden immer alle aufgezeichneten Ereignisse angezeigt, d.h. es gibt kein „Betrachterprofil“ für Benutzer-Ereignisströme.
- Gestartet wird die Präsentationsfunktion mit dem Kommando

  ```
  /SHOW-SYSEVENT-LOG LOG-ID=MYES
  ```

  Die Präsentationsfunktion meldet sich mit ihrer Hauptmaske im statischen Modus und bietet Eingabemöglichkeiten über Menü, Kommandozeile und F-Tasten. Die aktuelle Maske positioniert immer auf die jüngsten Ereignisse.

  Folgende Funktionen werden im statischen Modus angeboten:

  - Positionieren und Blättern innerhalb des Ereignisstroms
  - Suchen
  - Wechseln in den dynamischen Modus
  - Bereitstellen eines Hilfesystems

Für eine ausführliche Beschreibung der Präsentationsfunktion wird auf die Beschreibung des Kommandos SHOW-SYSEVENT-LOG verwiesen.

16.2 Kommandoeingabe über Konsole

Die Tastaturfelder sind bei den verschiedenen Server-Typen unterschiedlich.

Eingabe der Kommandos

1. EINGEB-Taste drücken
2. Eingabe eines Schrägstriches (/), dem das Kommando und die erforderlichen Operanden folgen
3. ETX-Taste (DÜ1-Taste oder ENTER-Taste) drücken

Formate der Kommandos

```
/[.cid] cmd[ operands]
```

.	Punkt
cid	Kommandoauftragskennzeichen Das cid beginnt mit einem Buchstaben oder einem der Zeichen @, $, #, an das sich noch 7 Zeichen (A-Z, 0-9, oder @,$,#) anschließen können. Es entspricht formal den Job-Namen bei Benutzerkommandos (siehe Handbuch „Kommandos“ [27]). Das cid dient der Auftragsklammerung bei Operatorkommandos. Die ersten drei Zeichen des cid erscheinen als mid (Meldungsauftragskennzeichen, siehe „Format der Meldungen“, "Nachrichtenformate") in der Kommandobeendigungsmeldung, die vom System für alle Operatorkommandos gesendet wird und in allen Kommandoergebnissen.
cmd	Bezeichnung eines Operator- oder Spezialkommandos
''	Leerzeichen
operands	Operanden des Kommandos

Die Kommandos können abgekürzt werden, solange sie eindeutig bleiben.

Beispiel

```
1. /.MINE CANCEL-JOB *TSN(1532)

2. /.YOURS SPECIAL OP1=<value1>,OP2=<value2>
```

Das Kommando unter 1. ist ein gewöhnliches Operatorkommando,
das Kommando unter 2. ist ein Spezialkommando.

16.3 Nachrichten

- Emergency-Nachrichten
- Verständigung zwischen System und Operator
- Nachrichtenformate
- Steuerung der Nachrichtenzustellung
- Nachrichtenaustausch der Operator

16.3.1 Emergency-Nachrichten

Schwerwiegende Systemprobleme, wie z.B. Speichermangel oder Hardware-Fehler bei Systemplatten, können die BS2000-Komponenten treffen, die für die Ein-/Ausgabe von Konsolnachrichten zuständig sind. In Situationen, in denen die normalen Ausgabewege blockiert sind, wählt das System für wichtige Nachrichten einen alternativen Ausgabeweg, den sog. „Emergency-Pfad“.

Ausgaben über den Emergency-Pfad werden Emergency-Meldungen bzw. Emergency-Fragen genannt. Sie werden zunächst über interne Service-Schnittstellen direkt an der Hauptkonsole des Systems ausgegeben.

Formate

%E.text	Emergency-Meldung
?E.text	Emergency-Frage

Anschließend werden – wenn möglich – die Nachrichten der regulären Nachrichtenverarbeitung übergeben, siehe die folgenden Abschnitte.

Unabhängig von der Ausgabe an logischen Konsolen oder der Anzeige durch das Kommando SHOW-PENDING-MSG ist das Beantworten von Emergency-Fragen nur an einer physikalischen Konsole möglich, an der die Meldung ausgegeben wurde.

Format der Antwort

E.text	Antwort auf Emergency-Frage

16.3.2 Verständigung zwischen System und Operator

Das System und seine Benutzer können Nachrichten an die Konsolen senden und Antworten von dort erhalten, wobei der „Benutzer“ auch ein Operator sein kann, der selbst an einer Konsole sitzt. Inhalt und Bedeutung dieser Nachrichten sind naturgemäß sehr verschieden. Um entscheiden zu können, welche Nachrichten wie behandelt werden müssen, unterscheidet man sie nach Typen und Eigenschaften.

Nachrichtentypen

Um dem Empfänger möglichst schnell einen Anhaltspunkt über Art und Wichtigkeit der Nachrichten zu geben, stellt das System eine Typeinteilung bereit. Jede auf der Konsole ausgegebene Nachricht wird genau einem dieser Typen zugeordnet, und jeder dieser Typen hat ein eindeutiges Typkennzeichen, das mit ausgegeben wird.

Folgende Nachrichtentypen sind definiert:

% „Meldung“: Informative Nachricht zur Kenntnisnahme durch den Operator.

? „Frage“: Nachricht, in der zur Eingabe einer Antwort aufgefordert wird.

. „Antwort“: Antwort des Operators auf eine Frage.

; „Anweisung“: Nachricht, mit der der Operator aufgefordert wird, etwas zu tun. Aus der internen Sicht des Systems handelt es sich bei Anweisungen ebenfalls um Fragen, jedoch mit dem Unterschied, dass die Antwort nicht vom Operator gegeben werden kann, sondern nur vom Absender der Frage selbst.

/ „Kommando“: Typkennzeichen für die Eingabe eines Kommandos.

+ „Kommandoergebnis“: Nachricht eines Kommandoservers an den Absender des Kommandos.

& „Kommandozusatzinformationsanforderung“: Nachricht, in der der Absender eines Kommandos zur Eingabe weiterer Informationen aufgefordert wird.

: „Kommandozusatzinformation“: Antwort des Kommandogebers auf eine Kommandozusatzinformationsanforderung.

! „Kommandoende“: Nachricht, mit der dem Eingeber eines Kommandos das Ende der Ablaufüberwachung durch das System mitgeteilt wird. Im Normalfall ist dies identisch mit der Beendigung des Kommandos, es gibt jedoch Ausnahmen. Näheres siehe Abschnitt „Kommandobeendigungsmeldungen".

* „Rückweisung“: Nachricht, mit der UCON eine fehlerhafte Eingabe zurückweist. Alle möglichen Rückweisungen sind im Abschnitt „Rückmeldungen an physikalischen und logischen Konsolen" zusammengestellt.

Nachrichteneigenschaften

Für die Nachrichtenbehandlungsmechanismen und -kommandos des Systems sind ferner einige weitere, nicht immer aus dem Typ ableitbare Eigenschaften einer Nachricht entscheidend. Es werden unterschieden:

- „beantwortbare“ und „unbeantwortbare“ Nachrichten
- „kommandogebundene“ und „kommandoungebundene“ Nachrichten
- „gerichtete“ und „ungerichtete“ Nachrichten

Beantwortbare und unbeantwortbare Nachrichten

„Beantwortbar“ sind alle Nachrichten der Typen:

- Frage
- Kommandozusatzinformationsanforderung
- Anweisung

„Unbeantwortbar“ sind alle Nachrichten anderer Typen.

Wichtig ist diese Nachrichteneigenschaft z.B. bei der Ausgabe, wo beantwortbare Meldungen hervorgehoben (auf Farbbildschirmen z.B. rot anstatt grün) erscheinen und mit dem Kommando SHOW-PENDING-MSG wieder angezeigt werden können.

Kommandogebundene und kommandoungebundene Nachrichten

„Kommandogebunden“ sind – neben dem Kommando selbst – alle Nachrichten der Typen

- Kommandoergebnis
- Kommandozusatzinformationsanforderung
- Kommandozusatzinformation
- Kommandoende

„Kommandoungebunden“ sind alle Nachrichten anderer Typen.

Wichtig ist diese Nachrichteneigenschaft bei der Programmierung von Kommandoservern für Operator-Spezialkommandos (siehe "Operator-Spezialkommandos in berechtigten Benutzerprogrammen").

Gerichtete und ungerichtete Nachrichten

„Gerichtete“ Nachrichten sind an einen exakt definierten Empfänger gerichtet.

„Ungerichtete“ Nachrichten haben dagegen einen Routing-Code als Empfängerangabe. Sie werden vom System an alle Konsolen verteilt, die den angegebenen Routing-Code innehaben – das können sehr viele sein, aber auch gar keine.

Im Gegensatz zu den anderen Nachrichteneigenschaften ist der Nachrichtentyp hier kein eindeutiges Kriterium: Zwar sind z.B. Antworten (an den Fragesteller) und alle kommandogebundenen Nachrichten grundsätzlich gerichtet, die anderen Nachrichtentypen können jedoch sowohl gerichtet als auch ungerichtet sein. Zuverlässig erkennen lässt sich das nur in der CONSLOG, wo der Empfänger jeder Nachricht protokolliert wird, und in berechtigten Benutzerprogrammen, die ihre Nachrichten in erweiterter Form empfangen (siehe "Berechtigte Benutzerprogramme mit Operatorfunktionen").

Eine Sonderbehandlung gilt hier für Emergency-Nachrichten: Aus technischen Gründen gehen sie zunächst gerichtet an die Hauptkonsole. Anschließend werden sie, sofern noch möglich, ungerichtet über den Routing-Code * verteilt, jedoch nur an logische, nicht an physikalische Konsolen.

Wichtig ist diese Nachrichteneigenschaft z.B. bei der Meldungsfilterung (Kommandos ADD-/REMOVE-CONSOLE-FILTER): Dieser Mechanismus wirkt nur auf ungerichtete Meldungen mit dem Nachrichtentyp „%“ (Meldung).

Ausgabeformat

Alle Nachrichten werden bei der Ausgabe auf Konsole mit einem einheitlichen, 17 Zeichen langen Vorspann versehen.
Ausnahme: Antworten auf Fragen des Operators erhalten keine Uhrzeit-Angabe.

Spalte 1	Typkennzeichen der Nachricht: mögliche Werte siehe "Verständigung zwischen System und Operator"; Ausnahme: bei Emergency-Nachrichten nur „?“ und „Y“.
Spalte 2-5	Absender der Nachricht; möglich Notationen: • (mn): Absender ist die physikalische Konsole mit der MN „mn“ • name: Absender ist die logische Konsole mit dem Berechtigungsnamen „name“ • tsn: Absender ist die Task mit der TSN „tsn“
Spalte 6	Bindestrich (Minuszeichen)
Spalte 7-9	Meldungsauftragskennzeichen: 3-stelliges, alphanumerisches Kennzeichen der Nachricht
Spalte 10	Trennzeichen: Ist der Systemparameter SECSTART=N, so ist das Trennzeichen immer ein Punkt. Ist SECSTART=Y, so ist das Trennzeichen dann ein „#“, wenn die Nachricht vom System selbst stammt, also „vertrauenswürdig“ ist. Andere Nachrichten, z.B. vom Benutzer mit INFORM-OPERATOR ausgegebene, erhalten als Trennzeichen einen Punkt.
Spalte 11-16	Uhrzeit im Format: hhmmss (bei Antworten folgt ab Spalte 11 der Antworttext)
Spalte 17	Leerzeichen
Spalte 18ff	Nachrichtentext

Beispiel

```
% 6C5-000.122342 % BLS0519 PROGRAM 'EDT' LOADED
```

16.3.3 Nachrichtenformate

Folgende Formate werden beschrieben:

- Format beim Empfangen von Meldungen im Dialog zwischen System und Benutzer
- Format beim Senden von Antworten
- Format der kommandogebundenen Antworten
- Format bei der Eingabe von Operatorkommandos
- Format der Kommandoergebnisse und Kommandobeendigungsmeldungen

Format beim Empfangen von Meldungen im Dialog zwischen System und Benutzer

```
{?|%|;}tsn-mid{.|#}hhmmss text
```

%	Meldung, Antwort nicht erforderlich
?	Meldung, die eine Antwort verlangt, die auch vom Operator gegeben werden kann
;	Meldung, die eine Antwort verlangt, die nicht vom Operator gegeben werden kann
tsn	Auftragsnummer der Task, die die Systemmeldung verursacht hat
-	Bindestrich
mid	Meldungsauftragskennzeichen, Standardwert = 000 (A..Z, 0-9, @, #, $)
.	Punkt als Trennzeichen. Der Sender ist ein Benutzerprogramm.
#	Nummernzeichen als Trennzeichen. Der Sender ist ein Systemmodul. Ob Punkt oder „#“ ist abhängig vom Systemparameter SECSTART.
hhmmss	Tageszeit (6 Ziffern; hh = Stunde, mm = Minute, ss = Sekunde)
text	Beliebiger Meldungstext

Beispiel

```
%TSN1-000#142423 % NBR0970 OPERATOR TASK WITH TSN 'XAAA' CREATED FOR
                   CONSOLE 'K1'
?TSN9-000#142753 % NBR0820 SHOULD K1 BECOME MAIN OPERATOR TERMINAL ?
;TSN4-001#142823 % XYZ0123 SWITCH OFF DEVICE L1 URGENTLY
```

Meldungen des Typs „?“ und „;“ können mit dem Kommando SHOW-PENDING-MSG wieder angezeigt werden.

Meldungen des Typs „;“ können nur vom Meldungssender wieder aus der Warteschlange der offenen Fragen entfernt werden.
Der Operator hat nur indirekt Einfluss darauf, indem er die vom Sender verlangte Reaktion durchführt (im obigen Beispiel das Gerät L1 abschaltet).

Format beim Senden von Antworten

```
dest [-mid].[text]
```

dest	Ziel der Antwort. Dies ist der Absender der zu beantwortenden Meldung.
-	Bindestrich
mid	Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $); führende Nullen dürfen entfallen (mid muss der Meldung, die beantwortet werden soll, entsprechen)
.	Punkt, Kennzeichen für Antwort
text	Beliebiger Antworttext Bei Antworten auf Systemmeldungen sind nur bestimmte Texte zulässig. Hilfe zu einzelnen Meldungen erhalten Sie mit /HELP-MSG-INFORMATION.

Beispiel

```
TSN9-000.YES
```

```
TSN9.YES
```

Format der kommandogebundenen Antworten

```
dest [-mid]:[text]
```

dest	Ziel der Kommandozusatzinformation Dies ist der Absender der zu beantwortenden Meldung.
-	Bindestrich
mid	Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $) (muss der Meldung, die beantwortet werden soll, entsprechen)
:	Doppelpunkt, Kennzeichen für Kommandozusatzinformation
text	Beliebiger Antworttext Bei Antworten auf Systemmeldungen sind nur bestimmte Texte zulässig. Hilfe zu einzelnen Meldungen erhalten Sie mit /HELP-MSG-INFORMATION.

Beispiel

```
TSN3-TUR:Y
```

Format bei der Eingabe von Operatorkommandos

```
/[.cid] cmd[ operands]
```

/	Schrägstrich. Kennzeichen für Kommando
.	Punkt. Kennzeichen, dass ein „cid“ angegeben ist
cid	Kommandoauftragskennzeichen Das cid beginnt mit einem Buchstaben oder einem der Zeichen @, $, #, an das sich noch 7 Zeichen (A-Z, 0-9, oder @,$,#) anschließen können. Es entspricht formal den Job-Namen bei Benutzerkommandos (siehe Handbuch „Kommandos“ [27]). Das cid dient der Auftragsklammerung bei Operatorkommandos. Die ersten drei Zeichen des cid erscheinen als mid (Meldungsauftragskennzeichen, siehe "Nachrichtenformate") in allen kommandogebundenen Nachrichten (Kommandoergebnisse, Zusatzinfo-Anforderungen und Zusatzinfos).
cmd	Operationsbezeichnung eines Operatorkommandos
' '	Leerzeichen
operands	Operanden des Kommandos

Beispiele

```
/.MINE CANCEL-JOB *TSN(1532)
/SPECIAL-CMD OP1=<value1>,OP2=<value2>
/.EVAN16 SHOW-USER-ATTRIBUTES USER-IDENTIFICATION=RZ01
/.E SHOW-USER-ATTRIBUTES USER-IDENTIFICATION=RZ01
```

Im Gegensatz zum Meldungsauftragskennzeichen werden bei der Eingabe von Kommandoauftragskennzeichen nur bis zu acht Zeichen zugelassen. Ein- oder zweistellige Kommandoauftragskennzeichen werden auf bis zu drei Stellen nach rechts mit Nullen ergänzt. Kommandoauftragskennzeichen müssen mit einen Buchstaben oder den Sonderzeichen $,#,@ beginnen. Die ersten drei Zeichen des Kommandoauftragskennzeichens werden als Meldungsauftragskennzeichen verwendet.

Format der Kommandoergebnisse und Kommandobeendigungsmeldungen

```
{+|&|!}{tsn|an}-mid{.|#}hhmmss text
```

+	Kommandoergebnis, Antwort nicht erforderlich
&	Kommandozusatzinformationanforderung, der Absender verlangt eine Antwort
!	Kommandoende, Antwort nicht erforderlich
tsn	Auftragsnummer der Task, in der die Kommandobearbeitung stattfindet
an	Berechtigungsname des berechtigten Benutzerprogramms, das die Kommandobearbeitung durchführt (Operator-Spezialkommando)
-	Bindestrich
mid	Meldungsauftragskennzeichen, Standardwert = 000 (A..Z, 0-9, @, #, $)
.	Punkt als Trennzeichen
#	Nummernzeichen als Trennzeichen Ob als Trennzeichen der Punkt oder das Zeichen „#“ verwendet wird, ist abhängig vom Systemparameter SECSTART.
hhmmss	Tageszeit (6 Ziffern; hh = Stunde, mm = Minute, ss = Sekunde).
text	Beliebiger Meldungstext

Beispiele

```
&TSN3-TUR.142623  %  EXC0422 PROCESSING OF /TURN TO BE CONTINUED? REPLY
                            (Y=YES; N=NO)
+TSN2-ASR.142523  %  NBR0825 THIS CONSOLE IS (K3)
!TSN2-ASR.142723  %  NBR0740 COMMAND COMPLETED 'ASR' (RESULT: SC2=00,
                            SC1=00, MC=CMD0001); DATE :<date>
```

16.3.4 Steuerung der Nachrichtenzustellung

Die Nachrichten, die an eine Konsole gesendet werden können, lassen sich nach mehreren Kriterien unterscheiden (siehe auch Abschnitt „Nachrichten").

Um die Zustellung der Nachrichten zu steuern, stehen derzeit vier Mechanismen zur Verfügung:

- NOINF-Funktion
- Nachrichtenunterdrückung durch Filterung
- Meldungsbestellung anhand der Meldungsschlüssel
- Nachrichtenunterdrückung anhand der Meldungsschlüssel

> **i** Auf das Konsollogging (Funktion CONSLOG) hat keiner der vier Mechanismen Einfluss.
> Für die Funktionen zur Nachrichtenunterdrückung gilt: Beantwortbare Nachrichten sind grundsätzlich nicht unterdrückbar.

Kommando	**Bedeutung**
ADD-CONSOLE-FILTER	Ausgabe bestimmter Meldungsgruppen unterdrücken oder Konsole in den NOINF-Zustand versetzen
ASR	Konsole in den NOINF-Zustand versetzen
MODIFY-MSG-SUBSCRIPTION	Meldungsbestellung verfügen oder zurücknehmen
REMOVE-CONSOLE-FILTER	Mit ADD-CONSOLE-FILTER eingestellte Filterstufen aufheben
RESET-MSG-SUPPRESSION	Unterdrückung der Meldungsausgabe aufheben
SET-MSG-SUPPRESSION	Gezielt die Ausgabe einzelner Meldungen auf eine Konsole oder ein berechtigtes Benutzerprogramm unterdrücken
SHOW-CONSOLE-FILTER	Einstellung der Filterstufen anzeigen
SHOW-MSG-SUBSCRIPTION	Meldungsbestellung und Unterdrückung nicht bestellter Meldungen anzeigen
SHOW-MSG-SUPPRESSION	Informationen über Meldungsunterdrückung anfordern
OPR-Anweisung	**Bedeutung**
SET-FILTER	Ausgabe bestimmter Meldungsgruppen unterdrücken; Filterstufen einstellen

Tabelle 43: Schnittstellenübersicht zur Steuerung der Nachrichtenzustellung

Der Empfängerkreis einer Meldung wird auf folgende Weise ermittelt:

- Wenn es eine an eine bestimmte Konsole gerichtete Meldung ist, so wird diese Konsole in den Empfängerkreis der Meldung eingetragen.
- Wenn es eine über einen Routing-Code verteilte (d.h. ungerichtete) Meldung ist, dann werden alle Konsolen in den Empfängerkreis eingetragen, die Inhaber dieses Routing-Codes sind.
- Wenn es eine ungerichtete Meldung und keine Frage ist und es Konsolen gibt, die die Unterdrückung aller nicht explizit bestellten ungerichteteten Meldungen gefordert haben (mit den Kommandos MODIFY-MSG-SUBSCRIPTION ..., DELIVER-OTHER-MSG=*NO oder ASR NOINF), dann werden diese Konsolen aus dem Empfängerkreis gestrichen.
- Wenn es eine ungerichtete Meldung ist und es Konsolen gibt, die eine Unterdrückung von Meldungen eines betreffenden Gewichts und Routing-Codes gefordert haben (Kommando ADD-CONSOLE-FILTER), dann werden diese Konsolen aus dem Empfängerkreis gestrichen.
- Wenn es eine ungerichtete Meldung ist und sie einen Meldungsschlüssel hat, und wenn es keine Frage oder eine sonstige, als „nicht bestellbar“ gekennzeichnete Meldung ist, und wenn es Meldungsbestellungen (mit dem MODIFY-MSG-SUBSCRIPTION-Kommando) für den betreffenden Meldungsschlüssel (oder Teile von ihm) gibt, dann werden die Konsolen, für die diese Meldungsbestellungen gelten, in den Empfängerkreis eingetragen.
- Wenn die Meldung einen Meldungsschlüssel hat und keine Frage ist, und wenn es Konsolen gibt, die die Unterdrückung von Meldungen mit diesem Meldungsschlüssel gefordert haben (Kommando SET-MSG-SUPPRESSION), dann werden diese Konsolen aus dem Empfängerkreis gestrichen.

Die höchste Priorität hat die Unterdrückung der Meldungszustellung anhand der Meldungsschlüssel. Eine Konsole kann also eine Gruppe von Meldungen mit dem Kommando MODIFY-MSG-SUBSCRIPTION bestellen (z.B. alle Meldungen mit der Meldungsklasse NBR) und aus dieser Menge einzelne Meldungsschlüssel (z.B. `NBR0740`) mit dem Kommando SET-MSG-SUPPRESSION ausklammern.

NOINF-Funktion

Mit dem Kommando ASR kann eine Konsole in den NOINF-Zustand versetzt werden.
In diesem Zustand werden alle unbeantwortbaren, zu verteilenden Meldungen (durch „%“ gekennzeichnet und über einen Routing-Code zu senden) unterdrückt.

Dieselbe Wirkung erreicht man mit dem Kommando ADD-CONSOLE-FILTER, wenn für eine Konsole für alle Routing-Codes alle 5 Filterstufen beantragt werden, oder wenn das Kommando MODIFY-MSG-SUBSCRIPTION DELIVER-OTHER-MSG=*NO eingegeben wird.

Nachrichtenunterdrückung durch Filterung

Der Operator hat die Möglichkeit, durch Einstellen von Filterstufen (mit dem Kommando ADD-CONSOLE-FILTER oder die Parameterservice-Anweisung SET-FILTER) die Ausgabe bestimmter Meldungsgruppen zu unterdrücken.

Es handelt sich dabei grundsätzlich um unbeantwortbare, zu verteilende Meldungen (durch „%“ gekennzeichnet und über einen Routing-Code zu senden), die über ein Gewicht verfügen, also ihren Ursprung in einer Meldungsdatei haben und über den Makro MSG7 oder MSG7X produziert wurden.

Es können 5 Filterstufen angegeben werden, die den „Weight Codes“ der Meldungen (Meldungsgewicht) folgendermaßen zugeordnet sind:

Weight Code	Filterstufe	Bedeutung
bis 19	1	Frei für Kunden
20 - 39	2	Rein informative Meldungen
40 - 59	3	Erhöht informative Meldungen
60 - 79	4	Wichtige informative Meldungen
80 - 99	5	Sehr wichtige informative Meldungen

Z.B. werden bei eingeschalteter Filterstufe 2 alle Meldungen mit den Codes 20 bis 39 unterdrückt.

Die Einstellung der Filterstufen kann mit dem Kommando SHOW-CONSOLE-FILTER angezeigt werden. Mit dem Kommando ADD-CONSOLE-FILTER eingestellte Filterstufen können mit dem Kommando REMOVE-CONSOLE-FILTER wieder aufgehoben werden; mit der Parameterservice-Anweisung SET-FILTER eingestellte Filterstufen können nicht wieder zurückgenommen werden.

Welche Weight Codes den einzelnen Meldungen zugeordnet sind, ist in den Meldungsdateien beschrieben. Meldungsdateien können mit dem Dienstprogramm MSGMAKER angesehen werden. Die Systembetreuung kann außerdem mit MSGMAKER das Gewicht einer Meldung verändern.

Meldungsbestellung anhand der Meldungsschlüssel

Für Konsolen ist es möglich, Meldungen anhand ihrer Meldungsschlüssel explizit anzufordern. Es werden dann u. U. nur diese bestellten Meldungen an der jeweiligen Konsole ausgegeben.

Die Meldungsbestellung erfolgt mit dem Kommando MODIFY-MSG-SUBSCRIPTION und der Angabe der vollständigen oder beliebig reduzierten Meldungsschlüssel.

Der Operand ADD-MSG-ID fügt Meldungen (zum bereits bestehenden Meldungsbestell-Umfang) hinzu, mit dem Operanden REMOVE-MSG-ID können zuvor bestellte Meldungen aus dem Meldungsbestell-Umfang wieder herausgenommen werden.
Ein weiterer Operand entscheidet, ob alle nicht explizit angeforderten Meldungen an dieser Konsole unterdrückt werden sollen. Mit DELIVER-OTHER-MSG=*NO werden nur die mit ADD-MSG-ID bestellten Meldungen sowie Fragen und explizit an diese Konsole gerichtete Meldungen zugestellt (das entspricht der ASR NOINF-Funktion). Meldungsbestellungen haben keinen Einfluss auf die Zustellung der betreffenden Meldungen an anderen Konsolen.

Die Eingabe der Kommandos zur Meldungsbestellung ist an allen Typen von Konsolen zulässig, also sowohl an physikalischen Konsolen, logischen Konsolen ($CONSOLE-Anwendungen) und Dialogtasks mit dem Privileg OPERATING.
An physikalischen und logischen Konsolen beeinflussen die Kommandos die Meldungszustellung an der Konsole, von der aus sie eingegeben wurden.
In Dialogtasks mit dem Privileg OPERATING wirken die Kommandos auf einen System-Ereignisstrom (siehe auch "Benutzertask mit OPERATING-Privileg").

Nachrichtenunterdrückung anhand der Meldungsschlüssel

Um gezielt die Ausgabe einzelner Meldungen auf eine Konsole oder ein berechtigtes Benutzerprogramm zu unterdrücken, steht dem Operator das Kommando SET-MSG-SUPPRESSION zur Verfügung. Die Meldungsausgabe von zu bis zu 1000 verschiedenen Meldungsnummern kann unterdrückt werden.

Die Meldung muss eine Nummer haben, sie muss also ihren Ursprung in einer Meldungsdatei haben und über den Makro MSG7 oder MSG7X produziert worden sein. Es können alle Typen von unbeantwortbaren Nachrichten (Kennzeichen %, + und !) unterdrückt werden.

Die Ausgabe von Meldungen einer angegebenen Meldungsnummer kann mit dem Kommando RESET-MSG-SUPPRESSION wieder zugelassen werden, die Vereinbarungen können mit SHOW-MSG-SUPPRESSION aufgelistet werden.
Diese Feinfilterung über die Angabe von Meldungsnummern ist an allen Konsolen bzw. berechtigten Benutzerprogrammen zulässig; eine Vereinbarung für andere Konsolen bzw. Benutzerprogramme kann allerdings nur von der Hauptkonsole aus getroffen werden.

Durch die Vergabe entsprechender Werte für den Parameter OPR innerhalb der BS2000-Startup-Parameterdatei kann der Meldungsfluss bereits bei der Systemeinleitung reduziert werden (siehe Abschnitt „Konfiguration und Meldungsunterdrückung an Konsolen (OPR)").

16.3.5 Nachrichtenaustausch der Operator

Die Operator haben die Möglichkeit, Meldungen über die Konsolen auszutauschen.

Von einer gegebenen Konsole aus lässt sich dabei eine Meldung an folgende Ziele senden:

- an eine andere Konsole (mit dem mnemotechnischen Gerätenamen).
- an Konsolen, denen ein bestimmtes Aufgabengebiet zugeordnet ist (mit dem Berechtigungsschlüssel, siehe Abschnitt „Aufgabengebiete und ihre Aufteilung auf Konsolen").

Zwischen je zwei syntaktischen Einheiten dürfen Leerzeichen stehen. Beim Senden einer Meldung dürfen z.B. zwischen den syntakischen Einheiten „an“ und „-mid“ sowie „%“ Leerzeichen stehen. Siehe auch die jeweiligen Beispiele.

Folgende Formate werden beschrieben:

- Format beim Senden von Operatornachrichten
- Format beim Empfangen von Operatormeldungen
- Format beim Senden von Operatorantworten
- Format beim Empfangen von Operatorantworten

Format beim Senden von Operatornachrichten

```
{<rc | an | (mn)}[-mid] {% | ?}text
```

<	Spitze Klammer auf zur Einleitung des Berechtigungsschlüssels
rc	Berechtigungsschlüssel (1 Zeichen) Die Meldung geht an alle zugeordneten Konsolen und berechtigten Benutzerprogramme. Der Schlüssel * bedeutet: Die Meldung geht *in jedem Fall auch* an die Hauptkonsole.
an	Berechtigungsname eines berechtigten Benutzerprogramms (4 Zeichen). Die Meldung geht an das berechtigte Benutzerprogramm, das mit diesem Berechtigungsnamen verknüpft ist.
mn	Mnemotechnischer Gerätename einer Konsole (2 Zeichen) Die Meldung geht an die bezeichnete Konsole.
-	Bindestrich (Minuszeichen)
mid	Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $); Die letzten drei Zeichen werden als mid für die Ausgabe übernommen. Bei Angabe von weniger als drei Zeichen wird von links mit Nullen aufgefüllt.
%	Prozentzeichen; Kennzeichen, dass die Meldung eine reine Mitteilung ist (keine Antwort erforderlich)
?	Fragezeichen; Kennzeichen, dass die Meldung eine Frage ist, die vom Empfänger eine Antwort verlangt
text	Beliebiger Text

Format beim Empfangen von Operatormeldungen

```
{% | ?}{an | (mn)}-mid.hhmmss text
```

%	Prozentzeichen; Antwort nicht erforderlich
?	Fragezeichen; der Absender verlangt eine Antwort
an	Berechtigungsname des berechtigten Benutzerprogramms, von dem die Meldung stammt (4 Zeichen)
mn	Mnemotechnischer Gerätename der Konsole, von der die Meldung stammt
-	Bindestrich
mid	Meldungsauftragskennzeichen, Standardwert = 000 (A..Z, 0-9, @, #, $)
.	Punkt als Trennzeichen
hhmmss	Tageszeit (6 Ziffern; hh = Stunde, mm = Minute, ss = Sekunde)
text	Beliebiger Meldungstext

Beispiele

```
Eingabe von (K3): <A % THIS IS A MESSAGE WITH NO MID
Ausgabe an *)   : %(K3)-000.142423 THIS IS A MESSAGE WITH NO MID
Eingabe von (K4): <*-AKZ% THIS IS A MESSAGE WITH AN MID
Ausgabe an **)  : %(K4)-AKZ.142523 THIS IS A MESSAGE WITH AN MID
Eingabe von (K4): <*-123456789 % THIS IS A MESSAGE WITH A LONG MID
Ausgabe an **)  : %(K4)-789.142623 THIS IS A MESSAGE WITH A LONG MID
Eingabe von (K4): <*-@#$ % THIS IS A MESSAGE WITH SPECIAL CHARACTERS IN THE
MID
Ausgabe an **)  : %(K4)-@#$.142623 THIS IS A MESSAGE WITH SPECIAL CHARACTERS
IN THE MJID
Eingabe von (K4): <*-1 % THIS IS A MESSAGE WITH A SHORT MID
Ausgabe an **)  : %(K4)-001.142623 THIS IS A MESSAGE WITH A SHORT MID
```

*) Ausgabe erfolgt an alle Inhaber des Routing-Codes A

**) Ausgabe erfolgt an alle Inhaber des Routing-Codes *

> Aus Kompatibiltätsgründen werden die letzten drei Zeichen des Meldungsauftragskennzeichens zur Ausgabe verwendet. Ein- und zweistelligen Meldungsauftragskennzeichen werden führende Nullen vorangestellt. Es sind auch Meldungsauftragskennzeichen mit mehr als acht Zeichen zulässig. Das erste Zeichen darf eine Ziffer oder eines der Zeichen @, $, # sein.
> Diese Bedingungen für das Meldungsauftragskennzeichen entsprechen nicht den Vorschriften für das Kommandoauftragskennzeichen (siehe „Format bei der Eingabe von Operatorkommandos“ ("Nachrichtenformate")).

Format beim Senden von Operatorantworten

```
dest [-mid].[text]
```

dest Ziel der Antwort/Kommandozusatzinformation
Dies ist der Absender der zu beantwortenden Meldung.

\- Bindestrich

mid Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $); führende Nullen dürfen entfallen

(mid muss der Meldung, die beantwortet werden soll, entsprechen)

. Punkt; Kennzeichen für Antwort auf Fragen

text Beliebiger Antworttext

Bei Antworten auf Systemmeldungen sind nur bestimmte Texte zulässig.

Hilfe zu einzelnen Meldungen erhalten Sie mit /HELP-MSG-INFORMATION.

Format beim Empfangen von Operatorantworten

```
.{rc | (mn)}[-mid].[text]
```

. Punkt als Nachrichtentypkennzeichen einer Operator-Antwort

rc Berechtigungsname des antwortenden berechtigten Benutzerprogramms (4 Zeichen)

mn Mnemotechnischer Gerätename der antwortenden Konsole (2 Zeichen)

\- Bindestrich

mid Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $); wird bis zu 3 Zeichen mit führenden Nullen ergänzt. Standardwert = 000

. Punkt als Trennzeichen

text Beliebiger Antworttext

Beispiel

```
Fragesteller   Eingabe: (C2)-1? TO BE OR NOT TO BE ?
Frageempfänger Ausgabe: (C1)-001.092312 TO BE OR NOT TO BE ?
               Eingabe: (C1)-1. TO BE - THAT IS THE ANSWER
Fragesteller   Ausgabe: .(C2)-001. TO BE - THAT IS THE ANSWER
```

16.4 Verwendung mehrerer Konsolen

Dieser Abschnitt beschreibt nur die Systembedienung von physikalischen und logischen Konsolen. Die Bedienung des Systems kann auch von Benutzertasks mit dem Privileg OPERATING betrieben werden.

BS2000 bietet die Möglichkeit, mehrere Konsolen einzusetzen und ihnen die Aufgabengebiete so zuzuordnen, wie es der Betrieb eine Server Unit erfordert (siehe Abschnitt „Hauptkonsole und Nebenkonsolen"). Es können bis zu 24 physikalische und 192 logische Konsolen betrieben werden.

BS2000 gestattet darüber hinaus dem Personal eines Data Centers, Datenstationen während des Systemlaufs für die Systembedienung zu verwenden. Dazu kann das Softwareprodukt OMNIS verwendet oder es können Automatisierungs-Programme erstellt werden (siehe auch Abschnitt „Softwareprodukte OMNIS und PROP-XT").

Ein Schutzmechanismus (mit Kennwörtern) verhindert dabei den Missbrauch dieser Funktion durch nichtprivilegierte Benutzer.

Der Einsatz dieser Funktion ist zum Beispiel dann angebracht, wenn die Systemverwaltung – ausnahmsweise – in das Operating eingreifen möchte, Teile des Operating automatisiert werden sollen oder gleichartige Aufgaben für mehrere Server von einem Operator bearbeitet werden sollen.

Zuordnung von Aufgabengebieten zu Konsolen

Die Aufgabengebiete lassen sich auf die Konsolen folgendermaßen verteilen:

- Jedes Aufgabengebiet ist einer Konsole eindeutig zugeordnet: dies ist die strenge Trennung von Arbeiten nach Aufgabengebieten, wobei manchen Konsolen durchaus mehrere Aufgabengebiete zugeordnet sein dürfen.
- Einzelne Aufgabengebiete sind an mehreren Konsolen zu bearbeiten: dies ist eine unscharfe Aufteilung der Arbeiten nach Aufgabengebieten; sie ist nur in Sonderfällen sinnvoll.

Eine Übersicht über die Aufgabengebiete enthält der Abschnitt „Aufgabengebiete und ihre Aufteilung auf Konsolen".

Die Zuweisung von Routing-Codes (Zuordnung zu Aufgabengebieten) erfolgt:

- bei logischen Konsolen mit generierten Berechtigungsnamen über den Parameterservice bzw. das ASR-Kommando
- bei logischen Konsolen mit dynamischen Berechtigungsnamen über Operator-Rollen
- bei physikalischen Konsolen über eine der beiden Möglichkeiten in Abhängigkeit vom Systemparameter NBCONOPI

Auswirkungen der Zuordnungen im Systemlauf

Die Zuordnung von Aufgabengebieten zu Konsolen sind im Betrieb daran zu erkennen, dass Meldungen, die zu einem bestimmten Aufgabengebiet gehören, nur an den zugeordneten Konsolen erscheinen. Umgekehrt lassen sich Kommandos, die zu diesem Aufgabengebiet gehören, an den zugeordneten Konsolen eingeben.
Ist der für die Eingabe eines bestimmten Kommandos notwendige Berechtigungsschlüssel keiner Konsole zugeordnet, so kann von der Konsole mit der Funktion „Hauptkonsole“ dennoch dieses Kommando eingegeben werden, falls NBCONOPI=N eingestellt ist. Bei NBCONOPI=Y sind auch der Hauptkonsole nur diejenigen Kommandos erlaubt, deren Berechtigung sie explizit besitzt. Soll ein Kommando in einem System generell nicht eingebbar sein, so ist es im Parameterservice mit dem Berechtigungsschlüssel $ zu schützen.

Wenn logische Konsolen (berechtigte Benutzerprogramme) verwendet werden, so gelten die obenstehenden Aussagen für physikalische und logische Konsolen.

Nachrichten zwischen den Konsolen

Die Operatoren haben die Möglichkeit, Nachrichten über die Konsolen auszutauschen.

Von einer gegebenen Konsole aus lässt sich dabei eine Nachricht an folgende Ziele senden:

- eine andere Konsole (mit dem mnemotechnischen Gerätenamen)
- an Konsolen, denen ein bestimmtes Aufgabengebiet zugeordnet ist (mit dem Berechtigungsschlüssel, siehe Abschnitt „Aufgabengebiete und ihre Aufteilung auf Konsolen").

Die Formate der Nachrichten sind im Abschnitt „Nachrichtenformate" zu finden.

Kommando	Bedeutung
REQUEST-MAIN-CONSOLE-FUNCTION	Anfordern der Eigenschaft „Hauptkonsole“
SHOW-CONSOLE-STATUS	Anzeigen von Konfiguration und Status von Konsolen

Tabelle 44: Kommandoübersicht zur Verwendung mehrerer Konsolen

16.4.1 Hauptkonsole und Nebenkonsolen

Konsolen werden unterschieden in Hauptkonsole und – sofern vorhanden – Nebenkonsolen.

Hauptkonsole ist zunächst immer diejenige, von der aus das System geladen wird oder diejenige, die als erste operable Konsole erkannt wird.

Die verschiedenen Konsolfenster, die am SE Manager geöffnet werden, sind nur verschiedene Ansichten derselben Konsole (zu erkennen an der 2-stelligen Konsol-Mnemonik in der Statuszeile).

Haupt- und Nebenkonsolen haben keine technischen Unterschiede. BS2000 unterscheidet sie nur nach ihrer Funktion.

Hauptkonsole

Diese Funktion ist zu jedem Zeitpunkt genau einer Konsole zugeordnet. Bei Ausfall dieser Konsole wird sie vom System automatisch einer anderen verfügbaren Konsole übertragen. Mit dem Kommando REQUEST-MAIN-CONSOLE-FUNCTIONS kann der Operator die Funktion auch gezielt einer anderen Konsole übergeben. Die jeweils als Hauptkonsole wirksame Konsole hat folgende Eigenschaften:

- Sie erhält den Routing-Code * – auch dann, wenn er ihr nicht explizit zugewiesen wurde.
- Sie erhält alle so genannten „Emergency-Nachrichten“.
- Von ihr aus können alle Fragen und alle Anforderungen zu Kommando-Zusatzinformationen beantwortet werden, auch Emergency-Fragen (Ausnahme: an SCI-Kommandos gebundene und „;“-Fragen).

bei NBCONOPI=N

- Sie ist berechtigt, alle Kommandos auszuführen, deren Routing-Code keiner Konsole zugeordnet ist.
- Von dieser Konsole lässt sich die Zuordnung der Nebenkonsolen sowie das Zu- und Wegschalten von Konsolen per Kommando regeln.
- Von dieser Konsole aus lässt sich die Zuordnung von Aufgabengebieten zu anderen Konsolen jederzeit ändern.
- Sie erhält in der Regel alle Meldungen, die sich keiner anderen Konsole zuordnen lassen (siehe auch Systemparameter MSGDEST).

zusätzlich bei NBCONOPI=Y

- Von dieser Konsole lässt sich die Zuordnung von Aufgabengebieten zu logischen Konsolen mit generiertem Berechtigungsnamen mit dem ASR-Kommando ändern.
- Sie erhält alle Fragen, die an ihrem gewünschten Ausgabeort nicht ausgegeben werden konnten.

Wie bei allen Konsolen richten sich die Befugnisse der Hauptkonsole danach, welche Aufgabengebiete ihr zugeordnet sind.

> **i** Bis zum Ende des Ladevorgangs fällt die Funktion Hauptkonsole mit dem Gerät zusammen, von dem aus das System geladen wurde.
> Die Funktion Hauptkonsole lässt sich nur einer physikalischen Konsole zuordnen, nicht aber einer logischen Konsole (einem berechtigten Benutzerprogramm). Lediglich das Recht zur Beantwortung offener Fragen kann mit dem Systemparameter NBREPLY an den Routing-Code * (Stern) geknüpft werden.

Nebenkonsole

Alle Konsolen, die nicht die Funktion Hauptkonsole haben, werden vom System als Nebenkonsole angesehen. Die Befugnisse einer Nebenkonsole richten sich danach, welche Aufgabengebiete ihr zugeordnet sind.

Umschalten der Funktion Hauptkonsole

Das System schaltet die Funktion Hauptkonsole in folgenden Situationen um:

- Auf Anforderung eines Operators.
  Von der Konsole aus, die die Funktion Hauptkonsole erhalten soll, ist die Umschaltung anzufordern (mit dem Kommando REQUEST-MAIN-CONSOLE-FUNCTIONS). Das System verlangt daraufhin von der derzeitigen Hauptkonsole eine Bestätigung. Trifft diese ein, wird die Umschaltung vorgenommen. Andernfalls weist das System die Anforderung zurück.
- Bei Ausfall der derzeitigen Hauptkonsole.
  In diesem Fall schaltet das System automatisch auf eine anderen Konsole um: Für NBCONOPI=N wird die im Parameterservice vordefinierte Ersatz-Konsole verwendet (siehe Abschnitt „Anweisung DEFINE-CONSOLE"). Diese Auswahl kann vom Operator durch das CONSOLE-Kommando beeinflusst werden.
  Für NBCONOPI=Y wird die neue Hauptkonsole durch einen internen Algorithmus ausgewählt.

16.4.2 Ersatz-Konsolen

Für jede Konsole kann bei NBCONOPI=N eine Ersatz-Konsole definiert werden. Fällt eine Konsole aus, wird die ihr zugeordnete Ersatz-Konsole automatisch aktiviert.
Werden Ersatz-Konsolen nicht im Parameterservice definiert, so werden sie vom System durch einen internen Algorithmus ausgewählt.
Der Operator hat die Möglichkeit, die Zuordnungen zu verändern oder auf Ersatz-Konsolen umzuschalten (siehe Kommando CONSOLE).

Bei NBCONOPI=Y gibt es die Funktion „Ersatz-Konsole“ nicht.

16.4.3 Aufgabengebiete und ihre Aufteilung auf Konsolen

Aufgabengebiete bei der Bedienung von BS2000

Eine Übersicht über die Aufgabengebiete bei der Bedienung von BS2000 gibt die nachfolgende Tabelle. In der Tabelle sind auch – in alphabetischer Reihenfolge – die Kommandos angegeben, die zu den einzelnen Aufgabengebieten gehören. Diese Aufteilung der Kommandos gilt standardmäßig; sie kann bei der Systemeinleitung geändert werden (für jedes Kommando gesondert, siehe Abschnitt „Anweisung SET-CMD-CODE").

Für einige Kommandos ist die standardmäßige Zuordnung zu einem Aufgabengebiet und damit auch die Zuordnung eines Berechtigungsschlüssels vom Systemparameter NBCONOPI abhängig. Einige Kommandos, die bei NBCONOPI=N standardmäßig mit dem Berechtigungsschlüssel $ geschützt sind, erhalten bei eingeschaltetem Operator-Logon (NBCONOPI=Y) einen anderen Berechtigungsschlüssel. Sie werden in der Tabelle gesondert ausgewiesen.

Jedes Aufgabengebiet ist mit einem Berechtigungsschlüssel (BS), auch Routing-Code (RC) genannt, versehen, der bei der Zuordnung jeweils anzugeben ist.
Die Bedeutung dieser Berechtigungsschlüssel ist ab "Beschreibung der einzelnen Aufgabengebiete" beschrieben. Über die Zuordnung von Aufgabengebieten zu Konsolen gibt der Abschnitt „Zuordnung von Aufgabengebieten zu Konsolen" Auskunft.

Eine Zuordnung „Operatorkommandos zu Berechtigungsschlüssel“ finden Sie im Handbuch „Kommandos“ [27].

> ISP-Kommandos können aus Kompatibilitätsgründen noch angegeben werden. Ausführlich beschrieben sind diese Kommandos nur noch im Handbuch „Operator-Kommandos (ISP-Format)“ [35].

Die folgende Tabelle gilt i.A. nur bei NBCONOPI=N. Ausnahmen sind gesondert ausgezeichnet.

BS / RC	Aufgabengebiet	Standard für Kommando
A	Systemverwaltung	CHANGE-SERSLOG-FILE, SHOW-SERSLOG-STATUS, START-/STOP-SERSLOG
C	Bedienung des Datenkommunikationssystems	alle BCAM-Kommandos, z.B.: DCSTART, BCEND
	Bedienung und Steuerung des File Transfer	openFT-Kommandos, siehe Handbuch „openFT“ [23], z.B. ADD-FT-PARTNER, START-/STOP-FT
D	Bedienung von Plattenspeichern	CHANGE-DISK-MOUNT, CHECK-DISK-MOUNT, SET-DISK-DEFAULTS, SET-DISK-PARAMETER, UNLOCK-DISK

E	Allgemeine Aufgaben und Befugnisse	CANCEL-RUN-PROCESS, INFORM-ALL-JOBS, INFORM-JOB, MODIFY-MSG-FILE-ASSIGNMENT,
	Bei **NBCONOPI=Y** bekommen einige Kommandos, die sonst den BS $ haben, den BS E zugewiesen.	MODIFY-MSG-SUBSCRIPTION, PROTECT-FITC-APPLICATION, REQUEST-MAIN-CONSOLE-FUNCTIONS, SHOW-CJC-STATUS, SHOW-DEVICE-CONFIGURATION, SHOW-DEVICE-DEPOT, SHOW-DEVICE-STATUS, SHOW-DISK-DEFAULTS, SHOW-DISK-STATUS, SHOW-JOB-STATUS, SHOW-MASTER-CATALOG-ENTRY, SHOW-MOUNT-PARAMETER, SHOW-PUBSET-ATTRIBUTES, SHOW-RESOURCE-ALLOCATION, SHOW-RESOURCE-REQUESTS, SHOW-SHARED-PUBSET, SHOW-SYSTEM-STATUS, SHOW-TAPE-STATUS, SHOW-USER-STATUS SHOW-XCS-PUBSET, START-/STOP-DIALOG-APPLICATION
G	Geräteverwaltung	ADD-IO-UNIT, ATTACH-DEVICE, DETACH-DEVICE, INCLUDE-DEVICE-CONNECTION, MODIFY-IO-UNIT, MODIFY-MOUNT-PARAMETER, MOUNT-NET-STORAGE, REMOVE-DEVICE-CONNECTION, REMOVE-IO-UNIT, SHOW-NET-STORAGE, START-/STOP-CONFIGURATION-UPDATE, UMOUNT-NET-STORAGE, UNLOCK-DEVICE
H	Hardware-Wartung	
J	Steuerung und Kontrolle von Jobs	HOLD-JOB, HOLD-JOB-CLASS, HOLD-JOB-STREAM, HOLD-TASK, MODIFY-JOB,

	Bei **NBCONOPI=Y** bekommen einige Kommandos, die sonst den BS $ haben, den BS J zugewiesen.	MODIFY-JOB-CLASS, MODIFY-JOB-STREAM, MODIFY-JV, MODIFY-RESOURCE-COLLECTION, RESUME-JOB, RESUME-JOB-CLASS, RESUME-JOB-STREAM, RESUME-TASK, SHOW-JV, START-/STOP-JOB-STREAM, START-/STOP-RESOURCE-COLLECTION
K	OPR-Kommandoverwaltung	CONNECT-CMD-SERVER
N	Überwachung von Remote-Spool	REDIRECT-PRINT-JOB
O	Bedienung von Druckern	HOLD-PRINT-JOB (Alias HOLD-SPOOLOUT), RESUME-PRINT-JOB (Alias RESUME-SPOOLOUT)
P	Steuerung von Tasks	CANCEL-JOB, CHANGE-TASK-CPU-LIMIT, CHANGE-TASK-PRIORITY, ENTER-JOB, FORCE-JOB-CANCEL, INFORM-PROGRAM, SEND-MSG

R	Überwachung und Steuerung des Systemverhaltens	ACTIVATE-SNAPSHOT, CANCEL-PUBSET-EXPORT/-IMPORT, CHANGE-CONSLOG-FILE, CREATE-/DELETE-PAGING-FILE, DEACTIVATE-SNAPSHOT, EXPORT-PUBSET, EXTEND-PAGING-AREA, FORCE-PUBSET-EXPORT, HOLD-SUBSYSTEM, IMPORT-PUBSET, MODIFY-MEMORY-PARAMETERS, MODIFY-MSCF-CONNECTION, MODIFY-MSCF-ENVIRONMENT, MODIFY-NSM-ENVIRONMENT, MODIFY-PAGING-AREA-ATTRIBUTES, MODIFY-PCS-OPTION, MODIFY-SYSTEM-BIAS, MODIFY-TASK-CATEGORIES, REDUCE-PAGING-AREA, RELEASE-CLUSTER-RECOVERY-LOCK, RESUME-SUBSYSTEM, SET-DSSM-OPTIONS, SET-MSCF-ENVIRONMENT, SET-RESTART-OPTIONS, SET-XCS-PUBSET, SHOW-MEMORY-CONFIGURATION, SHOW-PAGING-CONFIGURATION, SHOW-PCS-OPTION, SHOW-RESTART-OPTIONS, SHOW-SNAPSHOT-STATUS, SHOW-SUBSYSTEM-STATUS, SHOW-TRACE-STATUS, SHUTDOWN, START-/STOP-MSCF-CONNECTION, START-/STOP-SUBSYSTEM, START-/STOP-TRACE
S	Steuerung des SPOOLOUT-Betriebs SPOOL laden	MODIFY-PRINTER-OUTPUT-STATUS, MODIFY-TAPE-OUTPUT-STATUS, SHOW-ACTIVE-SPOOL-DEVICES, SHOW-PRINT-JOB-STATUS, START-/STOP-PRINTER-OUTPUT, START-/STOP-TAPE-OUTPUT, START-/STOP-TAPE-REPLAY

T	Bedienung von Magnetbandgeräten	ADD-DEVICE-DEPOT, CHANGE-TAPE-MOUNT, CHECK-TAPE-MOUNT, REMOVE-DEVICE-DEPOT, UNLOAD-TAPE
U	Dateiverwaltung	
W,X,Y,Z	zur freien Verfügung	
B,F,I,L,M, Q,V, 0..8	reserviert	
9	VM2000-Verwaltung	alle VM2000-Kommandos
#	POSIX	

@	Sonderbedeutung (keine Berechtigung nötig)	ADD-CONSOLE-FILTER, DISCONNECT-CMD-SERVER, GRANT-PROP-CONNECTION, HELP-MSG-INFORMATION, MODIFY-CONSOLE-OPTIONS, MODIFY-USER-PROTECTION, RELEASE-OPERATOR-ROLE, REMOVE-CONSOLE-FILTER, REQUEST-OPERATOR-ROLE, RESET-MSG-BUFFER, RESET-MSG-SUPPRESSION, SET-MSG-SUPPRESSION, SHOW-CMD-ATTRIBUTES, SHOW-CONSLOG-ATTRIBUTES, SHOW-CONSOLE-FILTER, SHOW-CONSOLE-OPTIONS, SHOW-CONSOLE-STATUS, SHOW-FILE-ATTRIBUTES, SHOW-IOCF, SHOW-MSCF-CONFIGURATION, SHOW-MSG-SUBSCRIPTION, SHOW-MSG-SUPPRESSION, SHOW-NSM-CONFIGURATION, SHOW-OPERATOR-ATTRIBUTES, SHOW-OPERATOR-ROLE, SHOW-PENDING-MSG, SHOW-PUBSET-CACHE-ATTRIBUTES, SHOW-PUBSET-DEFINITION-FILE, SHOW-PUBSET-IMPORT-EXPORT, SHOW-PUBSET-OCCUPATION SHOW-PUBSET-PROCESSING, SHOW-PUBSET-RESTRICTION, SHOW-PUBSET-SPACE-ALLOCATION, SHOW-PUBSET-SPACE-DEFAULTS, SHOW-SDF-PARAMETERS, SHOW-SPACE-SATURATION-LEVELS, SHOW-SYSTEM-INFORMATION, SHOW-SYSTEM-PARAMETERS, SHOW-XCS-OCCUPATION
	zusätzlich bei **NBCONOPI=Y**	EXIT-JOB, SET-LOGON-PARAMETERS
*	Hauptkonsole	CONSOLE (bei NBCONOPI=Y ist dieses Kommando nicht mehr erlaubt) MODIFY-SDF-PARAMETERS

$	Sonderbedeutung (Kommando ist generell gesperrt)	ADD-MASTER-CATALOG-ENTRY, ADD-USER, LOCK-USER, MODIFY-MASTER-CATALOG-ENTRY, MODIFY-SPACE-SATURATION-LEVELS, MODIFY-USER-ATTRIBUTES, MODIFY-USER-PUBSET-ATTRIBUTES, PRINT-DOCUMENT (PRINT-FILE), REMOVE-MASTER-CATALOG-ENTRY, REMOVE-USER, SHOW-FILE-TRANSFER, SHOW-PRINTER-POOLS, SHOW-SPOOL-CHARACTER-SETS SHOW-SPOOL-DEVICES SHOW-SPOOL-FORMS SHOW-SPOOL-PARAMETERS SHOW-USER-ATTRIBUTES, UNLOCK-USER WRITE-SPOOL-TAPE
	bei **NBCONOPI=Y** ändert sich der BS von $ nach E	ADD-/SET-FILE-LINK, ADD-PASSWORD, COPY-FILE, CREATE-FILE, CREATE-FILE-GENERATION, CREATE-FILE-GROUP, CREATE-TAPE-SET, DELETE-FILE, DELETE-FILE-GENERATION, DELETE-FILE-GROUP, DELETE-SYSTEM-FILE, DELETE-TAPE-SET, EXPORT-FILE, EXTEND-TAPE-SET, IMPORT-FILE, MODIFY-FILE-ATTRIBUTES, MODIFY-FILE-GENERATION-SUPPORT, MODIFY-FILE-GROUP-ATTRIBUTES, REMOVE-PASSWORD
	bei **NBCONOPI=Y** ändert sich der BS von $ nach J	CREATE-JV, DELETE-JV, MODIFY-JV-ATTRIBUTES, REMOVE-JV-LINK, SET-JV-LINK, SHOW-JV-ATTRIBUTES, SHOW-JV-LINK

Tabelle 45: Berechtigungsschlüssel (BS / RC) - Aufgabengebiete - Kommandos

Beschreibung der einzelnen Aufgabengebiete

A Systemverwaltung

Dieses Aufgabengebiet umfasst das Aktivieren bzw. Deaktivieren des Software-Error-Logging und das Wechseln der SERSLOG-Datei.

C Bedienung des Datenkommunikationssystems

Dieses Aufgabengebiet umfasst z.B. folgende Tätigkeiten:

- Starten des Datenkommunikationssystems
- Definieren und Aktivieren von Kommunikationspartnern
- Aktivieren von Leitungen
- Definieren und Aktivieren von Routen zu Kommunikationspartnern

Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die auf Fehler im Datenkommunikationssystem hinweisen.

D Bedienung von Plattenspeichern

Dieses Aufgabengebiet umfasst das Bereitstellen von Datenträgern (Platten).
Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die auf Unstimmigkeiten bei den Archivnummern oder auf andere Fehler hinweisen.

E Allgemeine Aufgaben und Befugnisse

Dieses Aufgabengebiet umfasst keine genau festgelegten Tätigkeiten; vielmehr sollten die dazugehörigen Kommandos jeder Konsole zur Verfügung stehen.

G Geräteverwaltung

Dieses Aufgabengebiet umfasst z.B. folgende Tätigkeiten:

- Wegschalten eines Gerätes vom System
- Zuschalten eines Gerätes zum System
- Dynamische I/O-Konfigurationsänderung
- Net-Storage verbinden und trennen

Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die auf Unstimmigkeiten hinweisen.

H Hardware-Wartung

Jede zugeordnete Konsole erhält Meldungen, die auf Fehler im Server hinweisen und die vom Wartungspersonal ausgewertet werden.

J Steuerung und Kontrolle von Jobs

Jede zugeordnete Konsole erhält Meldungen, die den Beginn, Besonderheiten oder Fehler bei der Bearbeitung von Jobs anzeigen, so dass der Operator den Ablauf von Jobs verfolgen kann.

K	OPR-Kommandoverwaltung Berechtigte Benutzerprogramme können Operatorkommandos definieren. Sie können die so definierten Kommandos auch wieder löschen, und sie können sich für die Bearbeitung bereits eingetragener Operatorkommandos zuständig erklären und die Zuständigkeit auch wieder kündigen.
N	Überwachung von Remote-Spool
O	Bedienung von Druckern
P	Steuerung von Tasks Dieses Aufgabengebiet umfasst z.B. folgende Tätigkeiten: • Festlegen von Prioritäten für einzelne Jobs • Starten von Batch-Jobs • Abbrechen von Benutzer-Jobs
R	Überwachung und Steuerung des Systemverhaltens Dieses Aufgabengebiet umfasst z.B. folgende Tätigkeiten: • Ändern der kategorie-spezifischen Vorgaben zu Auftrags-Anzahl und I/O-Priorität • Beenden des Systemlaufs Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die Systemfehler oder Server-Fehler anzeigen oder die anzeigen, dass ein Benutzer mehr Speicherplatz auf gemeinschaftlichen Datenträgern angefordert hat, als ihm zusteht.
S	Steuerung des SPOOLOUT-Betriebs Dieses Aufgabengebiet umfasst z.B. die Zuordnung von Ausgabegeräten zu SPOOLOUT-Jobs. Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die auf Unstimmigkeiten hinweisen.
T	Bedienung von Magnetbandgeräten Dieses Aufgabengebiet umfasst das Bereitstellen bzw. das Wechseln von Datenträgern. Jede zugeordnete Konsole erhält darüber hinaus Meldungen, die auf Fehler hinweisen.
U	Dateiverwaltung
W,X,Y,Z	Zur freien Verfügung. Diese Aufgabengebiete stehen dem Benutzer für Sonderzwecke frei zur Verfügung (z.B. für Spezialkommandos).
B,F,I,L,M,Q,V,	Reserviert für künftige Systemerweiterungen.
1,2,3,4,5,6,7,8	Diese Aufgabengebiete sind derzeit nicht definiert, können aber in zukünftigen Versionen definiert werden.
9	Reserviert für VM2000-Verwaltung
#	POSIX

Berechtigungsschlüssel mit Sonderbedeutungen

0 Meldungen mit diesem Berechtigungsschlüssel sind nicht bestellbar. Meldungen ohne Antwort werden nur in der CONSLOG-Datei hinterlegt. Meldungen mit Antwort (Fragen) werden der Hauptkonsole zugestellt.

* Hauptkonsole
Dieses Aufgabengebiet ist immer auch der aktuellen Hauptkonsole zugeordnet. Ob ein Kommando mit diesem Berechtigungsschlüssel tatsächlich von der Hauptkonsole eingegeben wurde, wird nur beim Kommando CONSOLE geprüft.

Darüber hinaus erhält jede zugeordnete Konsole Meldungen, die über den Zustand von Abrechnungsdateien Auskunft geben.

@ Meldungen mit diesem Berechtigungsschlüssel sind bestellbar (Kommando MODIFY-MSG-SUBSCRIPTION). Meldungen ohne Antwort werden nur in der CONSLOG-Datei hinterlegt. Meldungen mit Antwort (Fragen) werden der Hauptkonsole zugestellt. Kommandos mit diesem Berechtigungsschlüssel sind ungeschützt; jeder Operator hat die Berechtigung, sie einzugeben.

Das Zuweisen des Schlüssels @ an Konsolen hat daher praktisch keinen Sinn. Sinnvoll hingegen ist die Zuordnung des Schlüssels an Meldungen oder Kommandos.

$ Ein mit dem Schlüssel $ geschütztes Kommando ist grundsätzlich keinem Benutzer erlaubt. Der Berechtigungsschlüssel $ kann zwar beliebigen Konsolen und berechtigten Benutzerprogrammen zugeordnet werden, aber er berechtigt nicht zur Eingabe eines mit $ geschützten Kommandos. Als Verteilerschlüssel gibt es keine Sonderbedeutung für den Schlüssel $.

Zuordnung von Aufgabengebieten zu Konsolen

Zum Zuordnen von Aufgabengebieten zu Konsolen gibt es folgende Möglichkeiten:

NBCONOPI=N

- Beim Systemstart

  Über die Startup-Parameterdatei können während der Startup-Phase physikalischen oder logischen Konsolen mit der SET-CODE-Anweisung (SET-CODE CODE=rc,CONSOLE=mn) Aufgabengebiete zugeordnet werden (siehe "Anweisung SET-CODE").

- Während des Systemlaufs

  Von der jeweiligen Hauptkonsole aus lässt sich mit dem Kommando ASR jede gewünschte Zuordnung für physikalische Konsolen sowie für logische Konsolen mit generierten Berechtigungsnamen herstellen.
  In Abhängigkeit von den Systemparametern ASRSW1 und ASRSW2 kann man auch von einer anderen physikalischen Konsole oder von einem logischen Konsole mit generiertem Berechtigungsnamen (sofern dort das Kommando ASR zugelassen ist) mit dem Kommando ASR die Zuordnung einstellen.
  ASR-Kommandos zur Veränderung der Zuteilung von Aufgabengebieten werden abgewiesen, wenn sie von berechtigten Benutzerprogrammen mit dynamischen Berechtigungsnamen oder in Bezug auf solche eingegeben werden. Berechtigte Benutzerprogramme mit dynamischen Berechtigungsnamen ordnen sich ihre Aufgabengebiete mit dem Kommando REQUEST-OPERATOR-ROLE selbst zu.

  Die Funktion „Hauptkonsole“ lässt sich nur dann umschalten, wenn die bisherige Hauptkonsole zustimmt.

  Fällt die jeweilige Hauptkonsole aus, so schaltet das System die Funktion „Hauptkonsole“ automatisch auf eine andere Konsole um; dabei wird die im Parameterservice vordefinierte Ersatz-Konsole verwendet (siehe Abschnitt „Anweisung DEFINE-CONSOLE"). Diese Auswahl kann vom Operator durch das CONSOLE-Kommando beeinflusst werden.

NBCONOPI=Y

Die in der Startup-Parameterdatei mit der Anweisung SET-CODE der IPL-Konsole zugewiesenen Routing-Codes werden als „implizite Routing-Codes“ bezeichnet.
Ihre Wirkung wird über den Systemparameter NBIMPRCA gesteuert.
Bei NBIMPRCA=Y wirken sie während des gesamten Systemlaufs, bei NBIMPRCA=N nur bis „System Ready“ und ab Systembeendigung (nach erfolgtem Shutdown).

„Implizite Routing-Codes“ wirken grundsätzlich nur als Routing-Codes (zur Verteilung von Meldungen) und nicht als Berechtigungsschlüssel (zur Eingabe von Kommandos).

- Beim Systemstart und nach Systembeendigung

  Die Hauptkonsole erhält alle Meldungen der Routing-Codes, die als „implizite Routing-Codes“ festgelegt wurden.

- Während des Systemlaufs (bis Shutdown)

  Bei NBIMPRCA=Y erhält die Hauptkonsole alle Meldungen der „impliziten Routing-Codes“, sofern diese keiner anderen Konsole explizit zugeordnet sind.

  Eine Konsole kann sein:

  - operabel / inoperabel
  - aktiv / inaktiv
  - berechtigt / unberechtigt

  Eine Konsole ist inoperabel, wenn sie systemintern als „defekt“ verwaltet wird, was auch für nicht-existierende Geräte oder Geräte im Zustand DETACHED gilt. Alle anderen Konsolen sind operabel.
  Eine operable Konsole wird durch das Kommando SET-LOGON-PARAMETERS aktiv und nach dem Kommando EXIT-JOB wieder inaktiv.
  Eine aktive Konsole erhält durch das Kommando REQUEST-OPERATOR-ROLE verschiedene Berechtigungsschlüssel, die ihre Berechtigungsmenge darstellen.
  Zur Eingabe von @-Kommandos sind aktive Konsolen immer berechtigt.
  Inaktiven Konsolen stehen nur folgende Kommandos zur Verfügung:
  SET-LOGON-PARAMETERS und SHOW-PENDING-MSG.

  Die erste Konsole, an der das Kommando SET-LOGON-PARAMETERS erfolgreich eingegeben wurde, erhält die Eigenschaft „Hauptkonsole“. Sie ist nun aktiv, hat aber nur die Berechtigung zur Eingabe von @-Kommandos.

Die Funktion „Hauptkonsole“ lässt sich nur dann umschalten, wenn die bisherige Hauptkonsole zustimmt.

Fällt die jeweilige Hauptkonsole aus, so schaltet das System die Funktion „Hauptkonsole“ automatisch auf eine andere Konsole um; dabei wird zunächst in der Menge der aktiven und danach in der Menge der operablen Konsolen gesucht. Auch ein EXIT-JOB-Kommando von der Hauptkonsole leitet diese Suche ein.
Mit ASR-Berechtigung kann die Hauptkonsole jede gewünschte Zuordnung von Routing-Codes an berechtigte Benutzerprogramme mit generiertem Berechtigungsnamen herstellen.

Hinweise

Das Zuordnen einer Konsole zum Aufgabengebiet @ hat praktisch keine Wirkung.

Es ist auch möglich, einen Berechtigungsschlüssel (z.B. A, C, R, U usw.) überhaupt keiner Konsole zuzuordnen. Wenn Berechtigungsschlüssel keiner Konsole zugeordnet sind, geschieht Folgendes:

- Meldungen, die keine Antwort erfordern, werden auf keiner Konsole ausgegeben. Sie werden nur in die Protokolldatei geschrieben (beim Schlüssel @ ist dies immer der Fall).
- Meldungen, die eine Antwort des Operators erfordern, werden an der Hauptkonsole ausgegeben. Auch über den Schlüssel @ verteilte Meldungen werden an der Hauptkonsole ausgegeben, wenn sie eine Antwort des Operators erfordern.
- Kommandos mit Berechtigungsschlüsseln, die keiner Konsole zugeordnet sind, werden bei NBCONOPI=N vom System akzeptiert, wenn sie von der Hauptkonsole eingegeben werden.
  Ausnahme: Kommandos mit dem Schlüssel @ werden immer akzeptiert.

16.5 Rückmeldungen

- Kommandobeendigungsmeldungen
- Rückmeldungen an physikalischen Konsolen
- Rückmeldungen an physikalischen und logischen Konsolen

16.5.1 Kommandobeendigungsmeldungen

Kommandobeendigungsmeldungen erkennt man am Meldungstyp „!“ als erstes Zeichen der Ausgabe. Folgende Kommandobeendigungsmeldungen sind möglich:

`NBR0740` `COMMAND COMPLETED 'ASTOP' ; RESULT: (SC2=00, SC1=00, MC=CMD0001); DATE:<date>`
erscheint, wenn das Kommando fehlerfrei ausgeführt wurde.

`NBR0741` `COMMAND SUCCESS UNCERTAIN 'SHOW-CONSLOG'; RESULT: (SC2=02, SC1=00, MC=NBR0741)`
erscheint, wenn während der Kommandobearbeitung ein Fehler auftrat, der die Terminierung der Operatortask zur Folge hatte.

`NBR0741` `COMMAND SUCCESS UNCERTAIN 'SPECIAL'; RESULT: (SC2=02, SC1=64, MC=NBR0741)`
erscheint, wenn während der Kommandobearbeitung in einem berechtigten Benutzerprogramm die Verbindung zur DCAM-Anwendung $CONSOLE abgebaut wurde.

`NBR0742` `COMMAND NOT ALLOWED 'ASR'; RESULT: (SC2=00, SC1=64, MC=CMD0216)`
erscheint, wenn die Konsole, von der aus das Kommando ausgegeben wurde, nicht berechtigt ist, dieses Kommando auszugeben.

`NBR0743` `COMMAND NOT AVAILABLE 'SPECIAL'; RESULT: (SC2=00, SC1=130, MC=CMD0200)`
erscheint, wenn die Routine, die das Kommando verarbeitet, nicht im System ist, oder wenn zwischen der DCAM-Anwendung $CONSOLE und dem berechtigten Benutzerprogramm, das das Kommando ausführen soll, keine Verbindung besteht.
Beispiel: `/.FSTAT DADM xxx`
Falls TDADM nicht geladen ist, wird dieses Kommando mit der Meldung `NBR0743` zurückgewiesen.

`NBR0744` `COMMAND NOT FOUND 'ADR'; RESULT: (SC2=00, SC1=64, MC=NBR0744)`
erscheint, wenn ein Kommando eingegeben wurde, das kein Operator-Kommando ist.
Beispiel: `/.A1234567 ADR HELP`

`NBR0745` `COMMAND ID INVALID '.1WRONGID'; RESULT: (SC2=00, SC1=01, MC=NBR0745)`
erscheint, wenn ein Kommandoauftragskennzeichen ein ungültiges Zeichen enthält.
Beispiel: `/.1WRONGID ASR HELP`

`NBR0746` `COMMAND CANCELLED 'RUN'; RESULT: (SC2=02, SC1=00, MC=NBR0746)`
erscheint, wenn ein RUN-Kommando mit dem Kommando CANCEL-RUN-PROCESS zurückgenommen wurde oder wenn wegen Abbruch der Verbindung zur Konsole die zugehörige Operator-Task terminiert wurde, bevor das Kommando von ihr bearbeitet werden konnte.

`NBR0747` `COMMAND ID AMBIGUOUS 'ABC'; RESULT: (SC2=00, SC1=64, MC=NBR0747)`
erscheint, wenn ein Kommando mit einem nicht eindeutigen Kommandoauftragskennzeichen eingegeben wurde.
Das Kommandoauftragskennzeichen muss nur bei Kommandos eindeutig sein, die eine Kommandozusatzinformation verlangen können und deren Bearbeitung in einem berechtigten Benutzerprogramm erfolgt.

`NBR0748 SYSTEM SHUTDOWN IN PROGRESS. COMMAND IGNORED 'ASR'; RESULT: (SC2=02, SC1=00, MC=CMD0198)`
erscheint, wenn während der Systembeendigung noch Kommandos (SHOW-PENDING-MSG ausgenommen) eingegeben werden.

`NBR0749 COMMAND FROM /RUN FILE CANCELLED: 'RUN'; RESULT: (SC2=02, SC1=00, MC=NBR0749)`
erscheint, wenn ein Kommando aus der RUN-Datei zwar gelesen, aber wegen eines vorausgegangenen CANCEL-RUN-PROCESS-Kommandos nicht mehr ausgeführt wird.

16.5.2 Rückmeldungen an physikalischen Konsolen

`NBC0050 CONSOLE ERROR - TRY AGAIN`
erscheint, wenn während des Lesens ein Fehler auftrat.

`NBC0051 TIMEOUT - INPUT IGNORED`
erscheint, wenn die vorhergehende Eingabeoperation wegen eines Fehlers von der Software abgebrochen wird.

`NBC0052 INPUT NULL - IGNORED`
erscheint, wenn der Operator nur ein `END OF TEXT` eingegeben hat.

`NBC0053 RECIPIENT NOT AVAILABLE - INPUT IGNORED`
erscheint, wenn eine Eingabe nicht an den Empfänger weitergereicht werden kann, da dieser nicht verfügbar ist.

`NBC0054 UCON NOT ACTIVE - INPUT IGNORED`
erscheint, wenn in der Zeit zwischen IPL-Ende und UCON-Start Kommandoeingaben oder Eingaben, die nicht mit `P.` beginnen, gemacht wurden.

`NBC0055 INVALID PREFIX ENTERED - NO SUCH ANSWER EXPECTED`
erscheint, wenn ein Eingabetext mit `P.` oder `E.` beginnt, aber keine Frage mit einem dieser Präfixe offen ist.

`NBC0058 UCON BUSY - INPUT IGNORED`
erscheint, wenn eine Eingabe wegen UCON-Problemen nicht weitergegeben werden konnte.

`NBC0059 NO PF1-, PF2- OR ENTER-KEY USED - INPUT IGNORED`
erscheint, wenn eine Eingabe nicht mit der PF1-, PF2- oder ENTER-Taste abgeschlossen worden ist.

`NBC0070 CIO TESTS THIS CONSOLE`
erscheint, wenn das System sich aus einem Zustand ohne funktionsfähige Konsole erholt hat. Sind alle Konsolen defekt, so werden sie zyklisch mit dieser Meldung getestet. Die erste Konsole, die wieder operabel ist, gibt diesen Text aus.

16.5.3 Rückmeldungen an physikalischen und logischen Konsolen

REJ1 TEXT NOT RECOGNIZED
erscheint unter folgenden Bedingungen:

- wenn kein Empfänger für die Meldung angegeben ist
- wenn der Empfänger mit mehr als 4 Zeichen angegeben ist
- wenn nach dem Empfänger keines der Zeichen „/ % ? . : -“ und kein Leerzeichen folgt oder falls nach dem Empfänger mit „-“ ein Auftragskennzeichen eingeleitet wird, aber in der Zeichenfolge dahinter keines der folgenden Zeichen „/ % ? . :“ enthalten ist
- wenn nur ein „/“ oder nur ein „/“ und ein Kommandoauftragskennzeichen (cid) eingegeben werden

REJ5 DESTINATION NOT FOUND
erscheint, wenn eine Meldung einen falschen mnemotechnischen Gerätenamen für eine physikalische Konsole oder einen falschen Berechtigungsnamen enthält.

REJ6 NO MATCH FOR REPLY
erscheint, wenn die passende Frage zu der gesendeten Antwortmeldung nicht gefunden werden kann. Die Zurückweisung erfolgt auch dann, wenn versucht wird, eine Kommandozusatzinformationanforderung mit Punkt und eine Frage mit Doppelpunkt zu beantworten. Meldungen des Meldungstyps „;“ können weder von physikalischen noch von logischen Konsolen beantwortet werden, sie werden vom System beim Eintreffen eines erwarteten Ereignisses zurückgezogen.

REJ7 USER NOT ALLOWED TO REPLY
erscheint, wenn ein nicht berechtigter Benutzer versucht, auf eine Fragemeldung eine Antwortmeldung zu geben.

REJ8 CONSOLE OFF - INPUT IGNORED
erscheint, wenn sich die Konsole software-mäßig im Zustand SWITCHED OFF befindet (siehe Kommando CONSOLE, Handbuch „Kommandos“ [27]).

REJ9 CONSOLE mn INOPERABLE - INPUT IGNORED
erscheint, wenn die Konsole als defekt gekennzeichnet ist, d.h. wenn das Hardcopy-Gerät ausgefallen ist und der Operand FORCED-HARDCOPY=YES gesetzt ist (siehe Kommando MODIFY-CONSOLE-OPTIONS, Handbuch „Kommandos“ [27]).

REJ10 QUESTION IDENTIFICATION ALREADY EXISTING
erscheint, wenn von einer Konsole eine Frage eingegeben wird, deren Nummer bereits einer anderen, von derselben Konsole gestellten und noch nicht beantworteten Frage, gegeben wurde.

REJ11 MESSAGE CANNOT BE MATCHED TO AN ORDER
erscheint, wenn sich ein berechtigtes Benutzerprogramm in seiner Eigenschaft als Kommandobearbeiter auf eine falsche Kommandoumgebung bezieht.

`REJ12` `NBMHE/NBMAP ERROR IN PROTOCOL`
erscheint, wenn eine $CONSOLE-Anwendung, die mit V02 angeschlossen ist, eine Nachricht mit einem formal falschen NBMHE oder NBMAP sendet.

> Bei den Meldungen `REJ1` und `REJ5` bis `REJ7` wird zusätzlich gemeldet, welcher Text nicht erkannt, bzw. welche Meldung, Frage oder Antwort abgewiesen wurde.

Folgende Meldungen wurden durch Kommandoabschlussmeldungen ersetzt:

`REJ2` `COMMAND NOT ALLOWED`
wird ersetzt durch die Kommandobeendigungsmeldung `NBR0742.`

`NBR0742` `COMMAND NOT ALLOWED '...'; (RESULT: SC2=00, SC1=64, MC=CMD0216)`
erscheint, wenn die Konsole, von der aus das Kommando gegeben wurde, nicht berechtigt ist, dieses Kommando zu geben.

`REJ3` `COMMAND NOT AVAILABLE`
wird ersetzt durch die Kommandobeendigungsmeldung `NBR0743.`

`NBR0743` `COMMAND NOT AVAILABLE '...'; (RESULT: SC2=00, SC1=130, MC=CMD0200)`
erscheint, wenn die Routine, die das Kommando verarbeitet, nicht im System ist oder wenn zwischen der DCAM-Anwendung $CONSOLE und dem berechtigten Benutzerprogramm, das das Kommando ausführen soll, keine Verbindung besteht.

`REJ4` `COMMAND NOT FOUND`
wird ersetzt durch die Kommandobeendigungsmeldung `NBR0744.`

`NBR0744` `COMMAND NOT FOUND '...'; (RESULT: SC2=00, SC1=64, MC=NBR0744)`
erscheint, wenn ein Kommando eingegeben wurde, das kein Operator-Kommando ist.

16.6 Ablösung der Kommandos STATUS MSG und ASR

Die Kommandos STATUS MSG und ASR wurden bereits in BS2000/OSD-BC V3.0 durch die folgenden Kommandos abgelöst, werden jedoch aus Kompatibilitätsgründen weiterhin unterstützt.
Das Kommando ASR steht dem Operator nur noch im eingeschränkten Funktionsumfang und an der Hauptkonsole nicht mehr automatisch zur Verfügung. Mit dem ASR-Kommando können Berechtigungsschlüssel (Routing-Codes) auch nur noch an Anwendungen mit generiertem Berechtigungsnamen vergeben werden.

Kommando STATUS	**Ersatz-Kommando**	**BS**
MSG	SHOW-PENDING-MSG	@
MSG, ALL	SHOW-PENDING-MSG DESTINATION=*ANY	@

Kommando ASR	**Ersatz-Kommando**	**BS**
ADD, CODE=...	REQUEST-OPERATOR-ROLE	@
ADD, CODE=..., FILTER=....	ADD-CONSOLE-FILTER FILTER=...,ROUTING-CODE=...	@
ADD, CODE=..., FILTER=ALL	ADD-CONSOLE-FILTER FILTER=(1,2,3,4),- ROUTING-CODE=...	@
ADD, CONSOLE=...	hat kein Ersatz-Kommando	--
DELETE, CODE=...	RELEASE-OPERATOR-ROLE	@
DELETE, CODE=..., FILTER=...	REMOVE-CONSOLE-FILTER FILTER=...,- ROUTING-CODE=...	@
DELETE, CODE=..., FILTER=ALL	REMOVE-CONSOLE-FILTER FILTER=(1,2,3,4),- ROUTING-CODE=...	@
DELETE, CONSOLE=...	hat kein Ersatz-Kommando	--
DESTINATION	SHOW-SYSTEM-PARAMETERS PARAM=MSGDEST	R
HELP	SHOW-CONSOLE-STATUS CONSOLE=*OWN	@
HELP, CODE=...	SHOW-CONSOLE-STATUS CONSOLE=*ALL [1]	@

HELP, CODE=..., FILTER=...	SHOW-CONSOLE-FILTER SELECT=*FILTER(...) oder SHOW-CONSOLE-FILTER SELECT=*ROUTING-CODE(...)	@
HELP, COMMAND=....	SHOW-CMD-ATTRIBUTES	@
HELP, CONSOLE=...	SHOW-CONSOLE-STATUS CONSOLE=...	@
HELP, CONSOLE=ALL	SHOW-CONSOLE-STATUS CONSOLE=*ALL	@
HELP, CONSOLE=..., FILTER=...	SHOW-CONSOLE-FILTER SELECT=*FILTER(...) [2]	@
INF	REMOVE-CONSOLE-FILTER FILTER=*ALL,- ROUTING-CODE=*ALL oder MODIFY-MSG-SUBSCRIPTION DELIVER-OTHER-MSG=*YES	@ E
MAIN	REQUEST-MAIN-CONSOLE-FUNCTIONS	E
NOINF	ADD-CONSOLE-FILTER FILTER=ALL,- ROUTING-CODE=*ALL oder MODIFY-MSG-SUBSCRIPTION DELIVER-OTHER-MSG=- *NO	@ E
PRIMARY	hat kein Ersatz-Kommando	--
SUPPRESS	RESET-MSG-BUFFER SENDER=...	@

[1] wird nicht direkt abgebildet, die Information ist aber im angegebenen Kommando enthalten

[2] nur bezogen auf die eingebende Konsole

17 Automatisierung von Operatorfunktionen

- Berechtigte Benutzerprogramme mit Operatorfunktionen
  - Anschlüsse mit generierten Berechtigungsnamen
  - Anschlüsse mit dynamischen Berechtigungsnamen
  - Austausch von Nachrichten
  - Nachrichtenformate
- Operator-Spezialkommandos in berechtigten Benutzerprogrammen
  - Kommandodefinition
  - Nachrichtenformate
- Softwareprodukte OMNIS und PROP-XT
- Kommandodateien für den Operator
  - Ausführen und Abbrechen einer Kommandodatei
  - Aufbau von Kommandodateien
- Ausübung von Systemverwaltungsfunktionen durch den Operator

17.1 Berechtigte Benutzerprogramme mit Operatorfunktionen

Mit der $CONSOLE-Schnittstelle von BS2000 kann sich der Operator von solchen Arbeiten befreien, die sich programmgesteuert ausführen lassen, z.B.

- regelmäßiges Abfragen bestimmter Lastwerte des Systems (z.B. Anzahl von Jobs nach unterschiedlichen Kriterien, Auslastung des Hauptspeichers);
  Veranlassen geeigneter Maßnahmen, falls erforderlich
- regelmäßiges Prüfen, ob unbeantwortete Meldungen ausstehen; Nachrichten an die Konsolen senden
- Geräteverwaltung

Tätigkeiten dieser Art lassen sich auf Benutzerprogramme übertragen, die sich als BCAM-/DCAM-Anwendungen dem Operating-System gegenüber als berechtigte $CONSOLE-Anwendungen ausgewiesen haben und als logische Konsolen fungieren. Diese als Konsole fungierenden Benutzerprogramme werden im Allgemeinen als **berechtigte Benutzerprogramme** bezeichnet. Sie sind berechtigt, Operatorfunktionen entsprechend ihrem Berechtigungsprofil (Routing-Code-Menge) auszuführen.

Ein unberechtigtes Benutzerprogramm darf lediglich Meldungen an das System senden und ggf. Antworten erhalten. Es handelt sich dabei um eine BCAM-/DCAM-Anwendung, die sich zwar an $CONSOLE anschließt, aber keinen speziellen (berechtigten) Verbindungsaufbau beantragt hat.

Berechtigte Programme können alle herkömmlichen Aktivitäten eines Operators ausführen.

Darüber hinaus kann ein berechtigtes Programm als Kommandoausführungsinstanz für Operator-Spezialkommandos auftreten (mit der Parameter-Anweisung ADD-CMD-ENTRY oder durch das Kommando CONNECT-CMD-SERVER definiert), siehe "Operator-Spezialkommandos in berechtigten Benutzerprogrammen".

Bei der Anmeldung eines eigenen Operator-Kommandos mit CONNECT-CMD-SERVER ist die Angabe COMPLETION-CONTROL=*YES möglich. Dabei ist zu beachten, dass – im Unterschied zu Konsolen – die Kommandoeingabe in einer Benutzertask synchron erfolgt.

Die Eingabe ist also erst nach beendeter Ausführung des vorherigen Kommandos möglich. Deshalb sollte das Kommando entweder eine kurze Bearbeitungsdauer aufweisen oder mit COMPLETION-CONTROL=*NO angemeldet werden.

> ¡ Vom Hersteller werden auf dieser Schnittstelle basierende Produkte mit eigener Oberfläche angeboten, z. B. OMNIS und PROP-XT (siehe "Softwareprodukte OMNIS und PROP-XT"), MAREN (siehe „MAREN“ [31]) und ROBAR (siehe „ROBAR“ [41]).
>
> Im Betriebsmodus NBCONOPI=Y besitzt die Operator-Task für ein berechtigtes Benutzerprogramm mit dynamischem Berechtigungsnamen die Privilegien seines Benutzers und das Privileg OPERATING.

Für berechtigte Benutzerprogramme gilt dieselbe Einteilung der Systembedienung in Aufgabengebiete wie für physikalische Konsolen.
Jedes Benutzerprogramm, das Operatorfunktionen ausüben will, hat sich mit der zentralen Task der Systembedienung (unter dem Applikationsnamen $CONSOLE) zu verbinden, zu „konnektieren“. Falls Operatorfunktionen ausgeübt werden sollen, die eine Berechtigung (Berechtigungsschlüssel, Routing-Code) voraussetzen, muss das Benutzerprogramm auch seine Berechtigung für diese Konnektierung nachweisen.

Die Verknüpfung eines Benutzerprogramms mit einem **Berechtigungsnamen** verleiht diesem Programm ein Merkmal, das mit dem mnemotechnischen Gerätenamen einer physikalischen Konsole vergleichbar ist.

Für berechtigte Benutzerprogramme, die sich anhand dieses Berechtigungsnamens an das System anschließen (generierter Berechtigungsname), muss im OPR-Parameterservice mit SET-CODE bzw. ADD-CMD-ENTRY ein Eintrag in die Berechtigungsnamentabelle erzeugt werden. Ein Berechtigungsname besteht aus 4 alphanumerischen Zeichen (Werte: A-Z, 0-9, oder @, $, #), wobei das erste Zeichen keine Ziffer und kein „#“ sein darf. „@“ als erstes Zeichen sollte nicht verwendet werden.
Vom System werden immer 512 Berechtigungsnamen erzeugt. Im Parameterservice können maximal 384 Berechtigungsnamen (generierte Berechtigungsnamen) angegeben werden. Die restlichen Berechtigungsnamen werden vom System vergeben (dynamische Berechtigungsnamen, @001 bis @512, sofern noch nicht vergeben).

Auf diese Weise lassen sich berechtigte Benutzerprogramme mit Namen bezeichnen, die die gleiche Lebensdauer haben wie die mnemotechnischen Gerätenamen der physikalischen Konsolen.

Berechtigungsnamen haben folgende Eigenschaften:

- Die Lebensdauer von Berechtigungsnamen ist gleich der Lebensdauer des Systems und damit gleich der Lebensdauer der Gerätedefinitionen für die physikalischen Konsolen.
- Sobald ein Benutzerprogramm mit einem Berechtigungsnamen verknüpft ist, gilt es als berechtigtes Benutzerprogramm. Die Verknüpfung ist vom Programm zu veranlassen und gilt, bis das Benutzerprogramm die Verknüpfung aufhebt, bis das Programm beendet wird oder bis UCON die Verbindung abbricht.
- Die Zuordnung von Aufgabengebieten für berechtigte Benutzerprogramme lässt sich, abhängig von der Anschlussart, entweder über den Berechtigungsnamen oder über Operator-Rollen herstellen.
- Nachdem ein Anschluss vom System akzeptiert wurde, ist ein berechtigtes Benutzerprogramm nur am Berechtigungsnamen als Sender oder als Empfänger von Nachrichten zu erkennen.

Einem Berechtigungsnamen kann eine Kommandobearbeitungsfunktion zugeordnet werden. Soll ein berechtigtes Benutzerprogramm Kommandos ausführen, so muss dem System bekannt sein, für welche Kommandos es als Kommandobearbeiter (Kommando-Server für Spezial-Kommandos) fungiert. Für berechtigte Anwendungen mit generiertem Berechtigungsnamen erfolgt die Zuordnung Kommandoname/Berechtigungsname im OPR-Parameterservice mit dem Satz ADD-CMD-ENTRY.
Daneben steht **allen** berechtigten Benutzerprogrammen zur Verknüpfung von berechtigten Anwendungen mit Operatorkommandos das Kommando CONNECT-CMD-SERVER zur Verfügung. Zum Lösen der Verknüpfung dient das Kommando DISCONNECT-CMD-SERVER (siehe Handbuch „Kommandos“ [27]).
Beide Kommandos sind standardmäßig mit dem Berechtigungsschlüssel K geschützt und nur für berechtigte Benutzerprogramme zulässig.

Die Systembetreuung erzeugt für jedes berechtigte Benutzerprogramm, das sich an das System anschließen darf, einen Eintrag im Benutzerkatalog. Sie sollte dabei Kennwörter zum Schutz vor unbefugter Aufnahme einer Verbindung zur Anwendung $CONSOLE (UCON) vereinbaren. Die Einträge in den Benutzerkatalog erfolgen mit dem Kommando ADD-USER; als Benutzerkennung wird bei Anwendungen mit generiertem Berechtigungsnamen der Berechtigungsname, bei Anwendungen mit dynamischem Berechtigungsnamen die OPERID verwendet.

Beispiel für einen Eintrag in den Benutzerkatalog

```
/ADD-USER USER-ID=RUDI,PROTECTION-ATTRIBUTES=(-
LOGON-PASSWORD=C'FOX#HOLE'),ACCOUNT-ATTRIBUTES=(ACCOUNT=K0815)
```

Als Benutzerkennung wird für Benutzerprogramme mit generiertem Berechtigungsnamen der Berechtigungsname verwendet.

```
/ADD-USER USER-ID=ISOLDE,PROTECTION-ATTRIBUTES=(-
LOGON-PASSWORD=C'MUSTARD!'),ACCOUNT-ATTRIBUTES=(ACCOUNT=K0815)
```

Als Benutzerkennung wird für Benutzerprogramme mit dynamischem Berechtigungsnamen die OPERID verwendet.

Ein sedezimales Kennwort ist nicht erlaubt.

Ein berechtigtes Benutzerprogramm darf prinzipiell jedes Operatorkommando verwenden; davon ausgenommen sind nur die Kommandos, deren Eingabeort eigenen Einschränkungen unterliegt (z.B. REQUEST-MAIN-CONSOLE-FUNCTIONS). Ein berechtigtes Benutzerprogramm kann nicht Ersatz- oder Hauptkonsole sein.
Ist in der zugeteilten Operator-Rolle die Berechtigung für das Kommando CONNECT-CMD-SERVER enthalten – standardmäßig ist das der Berechtigungsschlüssel K – so kann sich das berechtigte Benutzerprogramm mit diesem Kommando für die Bearbeitung von Operatorkommandos zuständig erklären.
Das System reicht Operatorkommandos, für deren Ausführung sich ein berechtigtes Benutzerprogramm zuständig erklärt hat, ohne syntaktische Analyse an das Programm weiter.

ACHTUNG!
Inhaber des Berechtigungsschlüssels K könnten versehentlich wichtige Operatorkommandos vereinnahmen und z.B. außer Kraft setzen. Die Systembetreuung sollte deshalb diese Berechtigung nur an ausgewählte Benutzerprozesse vergeben.

Aufbau der Berechtigungsnamentabelle

Den maximal 384, im Startup-Parameterservice mit dem Parametersatz OPR erzeugten, Berechtigungsnamen (generierte Berechtigungsnamen) werden vom System für berechtigte Benutzerprogramme, die sich anhand einer Operator-Identifikation anschließen, solange eindeutige Berechtigungsnamen hinzugefügt, bis die maximale Anzahl von 512 Einträgen erreicht ist.

Die Systembetreuung definiert mit dem Kommando CREATE-OPERATOR-ROLE die Liste von Berechtigungsschlüsseln, die zu einer Operator-Rolle zusammengefasst und dem angegebenen Pubset bekannt gemacht werden. Mit dem Kommando MODIFY-OPERATOR-ATTRIBUTES wird die Zuordnung von Operator-Rollen zu einer Operator-Identifikation hergestellt bzw. geändert.

Die Konnektierung als berechtigtes Benutzerprogramm kann auf zwei verschiedene Arten erfolgen:

- als Anschluss mit generiertem Berechtigungsnamen (Benutzerprogramme alter Art); siehe "Anschlüsse mit generierten Berechtigungsnamen"
- als Anschluss mit dynamischem Berechtigungsnamen; siehe "Anschlüsse mit dynamischen Berechtigungsnamen"

Kommando	**Bedeutung**
ADD-USER	Einträge in den Benutzerkatalog
CREATE-OPERATOR-ROLE	Eine oder mehrere Operator-Rollen definieren
DELETE-OPERATOR-ROLE	Löscht die Definition einer Operator-Rolle
EXIT-JOB	Beenden der $CONSOLE-Anwendung inkl. Verbindungsabbau
MODIFY-OPERATOR-ATTRIBUTES	Zuordnung von Operator-Rollen zu einer Operator-Identifikation herstellen bzw. ändern
RELEASE-OPERATOR-ROLE	Beantragt die Rückgabe eines spezifischen Tätigkeitsbereichs
REQUEST-OPERATOR-ROLE	Zuweisung einer oder mehrerer Operator-Rollen beantragen
SHOW-OPERATOR-ROLE	Zeigt Informationen über Operator-Rollen an
Makro	**Bedeutung**
NBMAP	Nachspann der Nachricht beschreiben (siehe "Nachrichtenformate")
NBMHE	Format des Nachrichtenheaders beschreiben (siehe "Nachrichtenformate")

Tabelle 46: Schnittstellenübersicht zu „Berechtigten Benutzerprogrammen mit Operatorfunktionen“

17.1.1 Anschlüsse mit generierten Berechtigungsnamen

Ein statischer Berechtigungsname kann im Startup-Parameterservice, Parametersatz OPR, mit den Anweisungen SET-CODE oder ADD-CMD-ENTRY generiert werden, siehe Abschnitt „Konfiguration und Meldungsunterdrückung an Konsolen (OPR)". Die Anweisung SET-CODE definiert die Zuordnung von Berechtigungsschlüsseln zu einer $CONSOLE-Anwendung mit diesem Berechtigungsnamen. Die Zuordnung kann mit dem ASR-Kommandos verändert werden.

Die Verbindung zu einem so generierten Berechtigungsnamen wird durch die Angabe folgender Parameter in der Verbindungsnachricht realisiert:

```
[NAME=]<name 4..4>
[,[PASSWORD=]C'password 1..8']
[,[PROTVERS=]Vn]
[,[DISCON=]{YES/NO}]
```

NAME=<name 4..4>
Der angegebene Name muss mit einem der mit den Anweisungen SET-CODE oder ADD-CMD-ENTRY im Startup-Parameterservice eingerichteten Berechtigungsnamen übereinstimmen.
Zusätzlich muss im Benutzerkatalog des Home-Pubsets eine gleichnamige Benutzerkennung existieren. Die Kennung sollte mit einem Kennwort geschützt sein. Sie darf auch gesperrt sein (Kommando LOCK-USER). Sie benötigt keine SECOS-Zugangsberechtigungen. Der Zugang erfolgt in der Zugangsklasse UCON.

PASSWORD=<password 1..8>
Ein Kennwort schützt den Zugang zu $CONSOLE mit diesem Berechtigungsnamen. Das im Benutzerkatalog der Benutzerkennung mit diesem Berechtigungsnamen eingetragene Kennwort ist anzugeben.
Ein sedezimales Kennwort ist nicht erlaubt. Das Kennwort darf max. 8 Zeichen lang sein.

PROTVERS und DISCON
Siehe die Beschreibung auf "Anschlüsse mit dynamischen Berechtigungsnamen".

17.1.2 Anschlüsse mit dynamischen Berechtigungsnamen

Anschlüsse mit generiertem Berechtigungsnamen genügen nicht mehr den strengen Sicherheitsanforderungen an ein System, da die Aufgabengebiete, d.h. eine Summe von Berechtigungsschlüsseln, immer nur einem Gerät oder einem berechtigten Benutzerprogramm, nicht aber einer Person zugeordnet werden können.

Diese Personalisierung wird bei Anschlüssen mit dynamischen Berechtigungsnamen gewährleistet, da bei der Konnektierung eine Operator-Identifikation (entspricht einer Benutzerkennung) angegeben werden muss. Unmittelbar nach der Konnektierung verfügt das berechtigte Benutzerprogramm über keinerlei Berechtigungen, kann also nur Operatorfunktionen, die keine Berechtigungen voraussetzen, und Kommandos mit dem Berechtigungsschlüssel @ nutzen. Erst mit dem Kommando REQUEST-OPERATOR-ROLE (Berechtigungsschlüssel @) können Berechtigungen beantragt werden, wobei die Vergabe einer OPERID-bezogenen Prüfung unterliegt.
Eine Operator-Rolle entspricht einem Aufgabengebiet und ist eine von der Systembetreuung festgelegte Menge von Berechtigungsschlüsseln, wobei beliebige Kombinationen der insgesamt 40 Berechtigungsschlüssel möglich sind. Ein berechtigtes Benutzerprogramm kann mehrere Operator-Rollen beantragen. Die logische Summe der Operator-Rollen ergibt dann das Berechtigungsprofil eines berechtigten Benutzerprogramms.

Mit der Festlegung von Operator-Rollen durch die Systembetreuung können auch Operatorfunktionen **personenbezogen** aufgeteilt und berechtigten Benutzerprogrammen zugewiesen werden.

Verbindungsaufbau mit dynamischem Berechtigungsnamen

Der Verbindungsaufbau einer berechtigten Anwendung mit dynamischem Berechtigungsnamen erfolgt grundsätzlich in mehreren Schritten. Die meisten davon sind nur optional, abhängig von den in der Verbindungsnachricht gemachten Angaben und davon, wie die verwendete Operator-Kennung geschützt ist.

Von $CONSOLE (Systemtask UCON) kann nicht überprüft werden, ob die Angabe $CONSOLE oder @CONSOLE in der ankommenden Verbindungsanforderung der Realität entspricht ($CONSOLE für die Datensichtstation als Endbenutzer, @CONSOLE für ein Programm als Endbenutzer), da diese Verbindungsanforderung immer von einem Vermittlerprogramm (z.B. OMNIS) eingereicht wird.
Die Systembetreuung sollte darauf achten, dass nur solche Programme als Vermittler eingesetzt werden, die die Korrektheit der Eingabe (erstes Zeichen „$“ oder „@“) sicherstellen. Er kann ausgewählten Kennungen den Zugang zu BS2000 ausschließlich von der Datensichtstation erlauben (siehe Kommando SET-LOGON-PROTECTION).

- Erster Schritt: Verbindungaufforderung

  Das berechtigte Benutzerprogramm sendet zunächst an $CONSOLE des gewünschten Systems eine Verbindungsanforderung mit folgender Struktur (x=$ oder x=@, je nach Art des Programmanschlusses, siehe auch Abschnitt „Format der Verbindungsnachricht für dynamische Berechtigungsnamen"):

  ```
  NAME=xCONSOLE[,OPERID=<name>][,PASSWORD=<password>][,PROTVERS=V<integer>]
  ```

  Der Anschluss wird **vorläufig** akzeptiert, wenn folgende Kriterien erfüllt sind:

  - es sind nicht alle zu vergebenden Berechtigungsnamen belegt
  - der Systemparameter NBBAPRIV erlaubt die Anmeldung

  Mit der vorläufigen Akzeptierung wird ein Begleittext gesendet, dem die von $CONSOLE (UCON) akzeptierte Protokollversion (PROTVERS) und die BS2000-Version zu entnehmen sind. Wenn die beantragte Protokollversion größer als die höchste Version ist, die von $CONSOLE unterstützt wird, enthält der Begleittext den maximalen Wert, den UCON akzeptieren kann.

Struktur des Begleittextes:

00 - 18 DC	C'CONNECTION ACCEPTED'	19 Byte
19 DC	C' ' (Leerzeichen)	1 Byte
20 - 21 DC	C'nn' (Protokollversionsnummer)	2 Byte
22 DC	C',' (Komma)	1 Byte
23 - 26 DC	C'nn.n' (BS2000-Versionsnummer)	4 Byts

Für BS2000 OS DX wird als BS2000-Versionsnummer C'21.0' ausgegeben.

Unabhängig von der in der Verbindungsanforderung beantragten Version des Protokolls, werden ab diesem Zeitpunkt, bis zur endgültigen Akzeptierung oder Ablehnung der Verbindung, alle Nachrichten in beide Richtungen mit einem Header versehen. Das Protokoll muss strikt eingehalten werden; unerwartete Eingaben während des Anmeldungsdialogs führen zur Zurückweisung des Konnektierungsversuches.

- Zweiter Schritt: OPR-LOGON-REQUEST

  Wenn der Operand OPERID in der Verbindungsnachricht fehlt, wird er vom System durch die Nachricht `PLEASE LOGON` angefordert.
  Das berechtigte Benutzerprogramm hat nun mit der Nachricht zu antworten
  (die ISP-Form `LOGON <name>,,<kennwort>` ist zulässig):

  `SET-LOGON-PARAMETERS USER-ID=<name>[,PASSWORD=<password>]`

  Das im Kommando SET-LOGON-PARAMETERS angegebene Kennwort wird für die weiteren Überprüfungen benutzt. Das in der Verbindungsnachricht eventuell angegebene Kennwort wird grundsätzlich ignoriert, wenn ein nachträgliches Kommando SET-LOGON-PARAMETERS angefordert wurde.

- Dritter Schritt: Prüfung der Schutzkriterien

  Anhand der bis hierher erhaltenen Information werden vom System die im Benutzerkatalog hinterlegten Schutzkriterien der Operator-Identifikation abgefragt.

  Wird bei der Überprüfung der Authentisierungsdaten festgestellt, dass für diese Operator-Identifikation ein Anschluss in der angegebenen Betriebsart (Attribute OPERATOR-ACCESS-TERMINAL bzw. OPERATOR-ACCESS-PROGRAM im Benutzerkatalog) nicht erlaubt ist, wird die Verbindung wieder abgebaut.

  Wird bei der Überprüfung der Authentisierungsdaten festgestellt, dass der Anschluss nicht durch Kennwort geschützt ist, wird der Schritt 4 übersprungen.

- Vierter Schritt: Anforderung des Kennworts

  Wenn das Kennwort bis jetzt noch nicht angegeben worden ist, wird es vom System mit der Nachricht `PLEASE ENTER PASSWORD` angefordert.
  Das berechtigte Benutzerprogramm hat nun mit dem Kennwort zu antworten.
  Unter OMNIS erfolgt die Eingabe dunkelgesteuert.

- Fünfter Schritt: Prüfung der Zugangsberechtigung

  Nachdem das berechtigte Benutzerprogramm alle notwendigen Daten an das System weitergeleitet hat, muss es noch die Bestätigung des Anschlusses an das System abwarten, bevor es mit dem normalen Nachrichtenverkehr, gemäß der in der Verbindungsanforderung geforderten Protokollversion, beginnen kann.

  Das System überprüft die Anschlussdaten und akzeptiert **endgültig** den Anschluss, wenn das angegebene Kennwort korrekt ist bzw. kein Kennwort erforderlich ist und OPERID und PASSWORD im Benutzerkatalogeintrag zusammenpassen
  (wenn die OPERID mit Kennwort zu authentisieren ist).

  Die Bestätigung enthält folgenden, an der Datensichtstation auszugebenden Text:

  ```
  CONNECTION REQUEST ACCEPTED, APPLICATION NAME = @xxx
  ```

Das angeschlossene berechtigte Benutzerprogramm kann zunächst nur einige nicht schützenswerte Kommandos eingeben (Schloss @). Um Operatorfunktionen auszuführen, muss es mit dem Kommando REQUEST-OPERATOR-ROLE eine oder mehrere Operator-Rollen beantragen.

Format der Verbindungsnachricht für dynamische Berechtigungsnamen

```
[NAME=]xCONSOLE [,[OPERID=]name]
[,[PASSWORD=]C'password 1..8']
[,[PROTVERS=]Vn]
[,[DISCON=]{YES/NO}]
```

Die Verbindungsanforderung kann auch ohne Schlüsselworte angegeben werden.

xCONSOLE
kennzeichnet die Art des Programmanschlusses im sicheren System
(siehe Kommando SET-LOGON-PROTECTION im SECOS-Handbuch „Zugangs- und Zugriffskontrolle“ [46]). Für **x** kann „$“ oder „@“ eingesetzt werden:

$CONSOLE
zeigt an, dass das berechtigte Benutzerprogramm im Dialog mit einer Person arbeitet. Die Prüfung der Zugangsberechtigung erfolgt gegen die mit SET-/MODIFY-LOGON-PROTECTION, Operand OPERATOR-ACCESS-TERM, getroffenen Festlegungen.

@CONSOLE
zeigt an, dass das berechtigte Benutzerprogramm im Batch-Betrieb arbeitet.
Die Prüfung der Zugangsberechtigung erfolgt gegen die mit SET-/MODIFY-LOGON-PROTECTION, Operand OPERATOR-ACCESS-PROGRAM, getroffenen Festlegungen.

OPERID=<name>
Name einer Operator-Identfikation. Eine Operator-Identifikation ist eine Benutzerkennung für Operatoren. Einen funktionellen Unterschied zu einer gewöhnlichen Benutzerkennung gibt es zunächst nicht. Erst wenn mit dem Kommando SET-LOGON-PROTECTION einer Benutzerkennung Operatorberechtigungen zugeordnet wurden, kann man sinnvollerweise von einer Operator-Identifikation sprechen.

PASSWORD=C'<password 1..8>'
Für eine Operator-Identifikation kann (wie für jede Benutzerkennung) ein Kennwort vereinbart werden. Das im Benutzerkatalog einer Operator-Identifikation eingetragene Kennwort ist anzugeben.

Ein sedezimales Kennwort ist nicht erlaubt. Das Kennwort darf max. 8 Zeichen lang sein.

PROTVERS=Vn
bezeichnet die Versionsnummer der Schnittstelle zwischen der Anwendung und $CONSOLE (n = 00, 01, 02). Jede Versionsnummer entspricht einem eindeutigen Kommunikationsprotokoll, das von $CONSOLE unterstützt und eingehalten wird.

Der Standardwert ist 00. In diesem Fall erfolgt der Nachrichtenverkehr zwischen System und berechtigtem Benutzerprogramm in beiden Richtungen ohne einen Header.

Bei der Version 01 empfängt das berechtigte Benutzerprogramm vom System die Nachricht und einen Header. Der Nachricht folgt bei Meldungen aus der Meldungsdatei, die mit dem Makro MSG7 oder MSG7X ausgegeben werden, ein Nachspann, der die Nachricht zur programmierbaren Auswertung „datenorientiert“ enthält. Die Datenstruktur des Headers wird vom Makro NBMHE festgelegt, die des Nachspanns vom Makro NBMAP (siehe "Makro NBMHE" und "Makro NBMAP" im Kapitel "Nachrichtenformate"). Der Nachrichtenverkehr vom berechtigten Benutzerprogramm zum System wird, wie bei der Protokollversion 00, ohne Header durchgeführt.

Bei der Version 02 wird der Nachrichtenverkehr in beiden Richtungen mit Header durchgeführt. Berechtigte Benutzerprogramme sind damit in der Lage, Nachrichten als Kommandoergebnisse und Zusatzinformationen zu kennzeichnen und dem System das Ende einer Kommandobearbeitung mitzuteilen.

Fehlt in der Verbindungsnachricht der Operand PROTVERS, so wird der Standardwert 00 eingesetzt. Wurde in PROTVERS ein zu hoher Wert angegeben, so wird die höchste vom System bediente Version im Begleittext der Verbindungsakzeptierung eingetragen. Die beantragte Versionsnummer der Schnittstelle kann also herabgesetzt werden. Ist das berechtigte Benutzerprogramm mit der herabgesetzten Versionsnummer nicht einverstanden, kann es die Verbindung abbauen.

DISCON
legt fest, wie das System reagiert, wenn das berechtigte Benutzerprogamm mit dem Empfang von Meldungen so weit in den Rückstand gerät, dass die durch den Systemparameter NBRCSCK[N] festgelegte Zeit überschritten wird.

DISCON=YES
Voreinstellung: Das System baut die Verbindung zum berechtigten Benutzerprogramm ab.

DISCON=NO
Das berechtigte Benutzerprogramm wird nicht diskonnektiert. Statt dessen werden alle Meldungen, die zur Ausgabe an das berechtigte Benutzerprogramm anstehen, gelöscht und durch eine einzelne Meldung `NBR0601` ersetzt.

```
NBR0601 SOME MESSAGES DISCARDED SINCE THE DESTINATION APPLICATION DID NOT
        RECEIVE THEM JUST IN TIME
```

Verbindungsabbau

Zum Beenden der $CONSOLE-Anwendung einschließlich Verbindungsabbau wird (mit NBCONOPI=Y) das Kommando EXIT-JOB angeboten. Damit wird implizit auch die Funktion von RELEASE-OPERATOR-ROLE OPERATOR-ROLE=*ALL realisiert und damit die Rückgabe aller Berechtigungsschlüssel.

17.1.3 Austausch von Nachrichten

Jedes berechtigte Benutzerprogramm tauscht seine Nachrichten über die DCAM-Anwendungen aus. DCAM bietet Schnittstellen für die Programmiersprachen COBOL und Assembler an:

- Wenn das zu lösende Problem keine besonderen Anforderungen an die Übermittlung von Nachrichten stellt, so lässt sich COBOL einsetzen.
- Ist aber eine asynchrone Verarbeitung eintreffender Meldungen erforderlich, so ist in Assembler (mit Makroaufrufen zur ereignisgesteuerten Verarbeitung) zu programmieren. Einzelheiten sind den Handbüchern „DCAM-Programmschnittstellen“ [13], „DCAM-Cobol-Aufrufe“ [11], „DCAM-Makroaufrufe“ [12] und „Makroaufrufe an den Ablaufteil“ [30] zu entnehmen.

Grobstruktur eines berechtigten Benutzerprogramms

Das folgende Bild gibt zunächst eine Übersicht, wie ein berechtigtes Benutzerprogramm Nachrichten mit dem System austauscht.

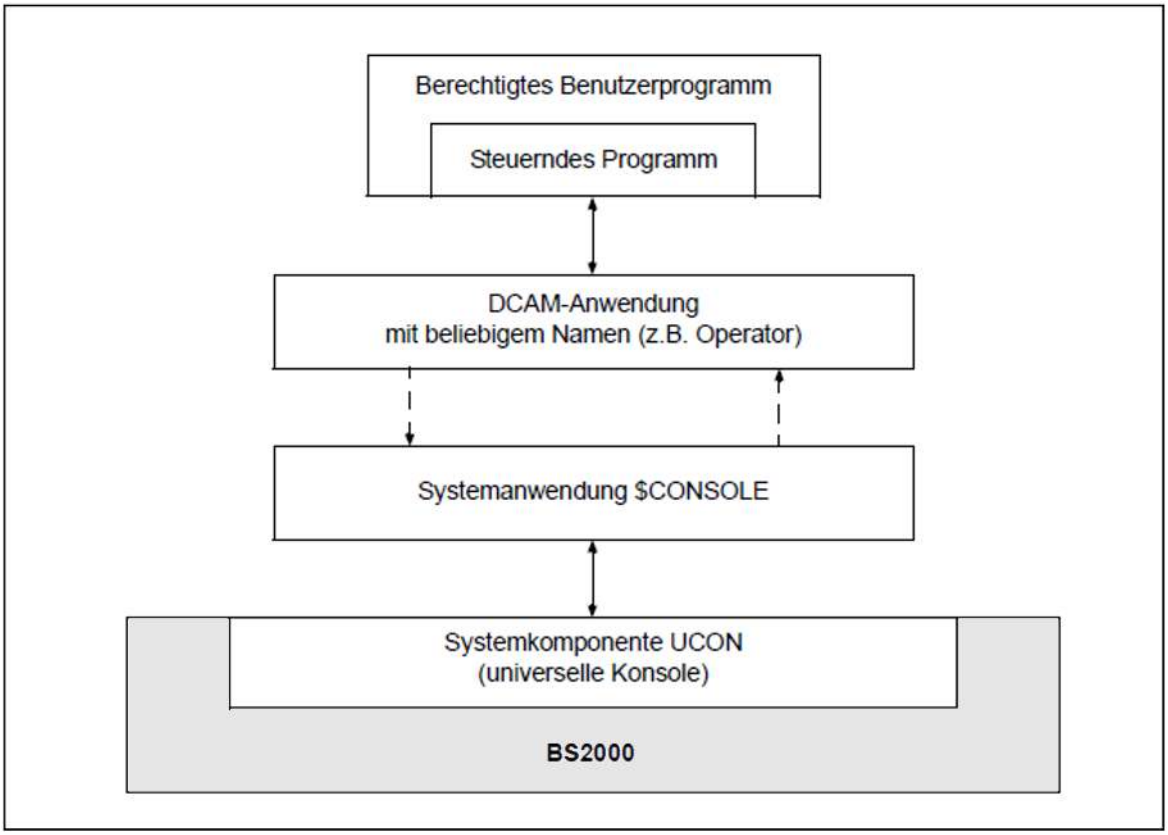


Bild 26: Nachrichtenaustausch zwischen BS2000 und berechtigtem Benutzerprogramm

Da jedes berechtigte Benutzerprogramm seine Nachrichten über DCAM-Anwendungen mit dem System austauscht, ergeben sich zwei Grobstrukturen für solche Programme: synchrone und asynchrone Verarbeitung.

- Synchrone Verarbeitung

  Die Grobstruktur für ein Programm, das eintreffende Meldungen synchron verarbeitet, zeigt das folgende Bild. Dabei kann der Inhalt der Schleife je nach gegebener Aufgabenstellung anders angeordnet sein.

  In Klammern ist jeweils der zu verwendende COBOL- bzw. Makroaufruf angegeben.

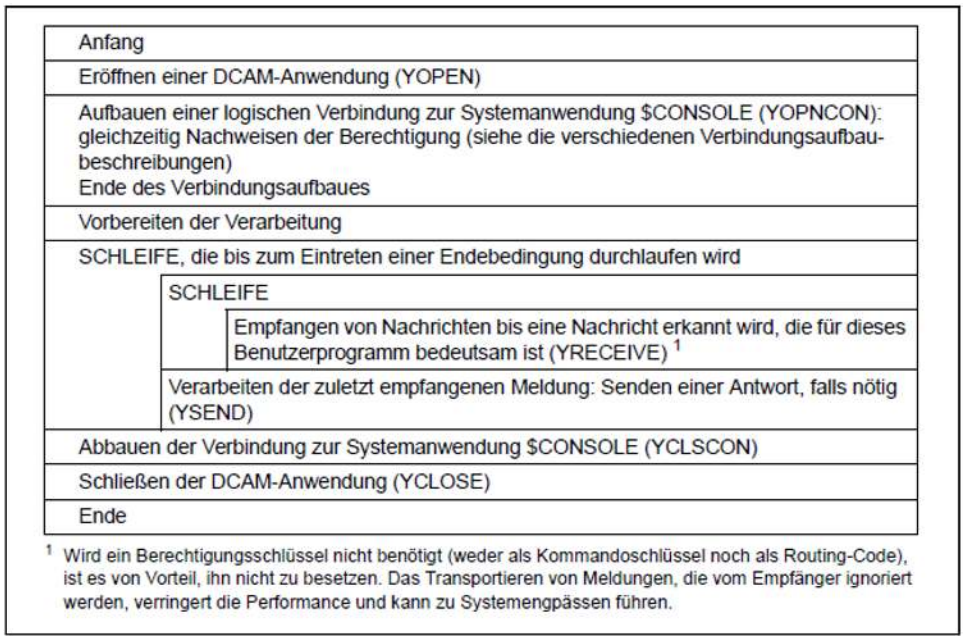


Bild 27: Grobstruktur für ein berechtigtes Benutzerprogramm (Meldungen synchron verarbeiten)

- Asynchrone Verarbeitung

  Die Grobstruktur für ein Programm, das eintreffende Meldungen asynchron verarbeitet, unterscheidet sich von der vorstehenden im Wesentlichen durch den Inhalt der Schleife und eine geringe Änderung im Schritt „Vorbereiten der Verarbeitung“: Ereigniskennungen sind einzurichten und Contingency-Definitionen dem System bekanntzugeben; siehe Makroaufrufe ENAEI und ENACO im Handbuch „Makroaufrufe an den Ablaufteil“ [30]).

  Die Schleife enthält hier nur die Anforderung einer Nachricht; wenn die Nachricht eintrifft, lässt sie sich in einem Contingency-Prozess verarbeiten, das Senden einer Antwort eingeschlossen.
  In die Schleife lassen sich weitere Tätigkeiten einbeziehen, die nicht zum Verarbeiten von Nachrichten gehören.
  Das asynchrone Verarbeiten von Nachrichten ist deshalb bei Aufgabenstellungen sinnvoll, die zusätzliche Koordinierungstätigkeiten erfordern oder bei denen mehrere Kommunikationspartner zu bedienen sind.

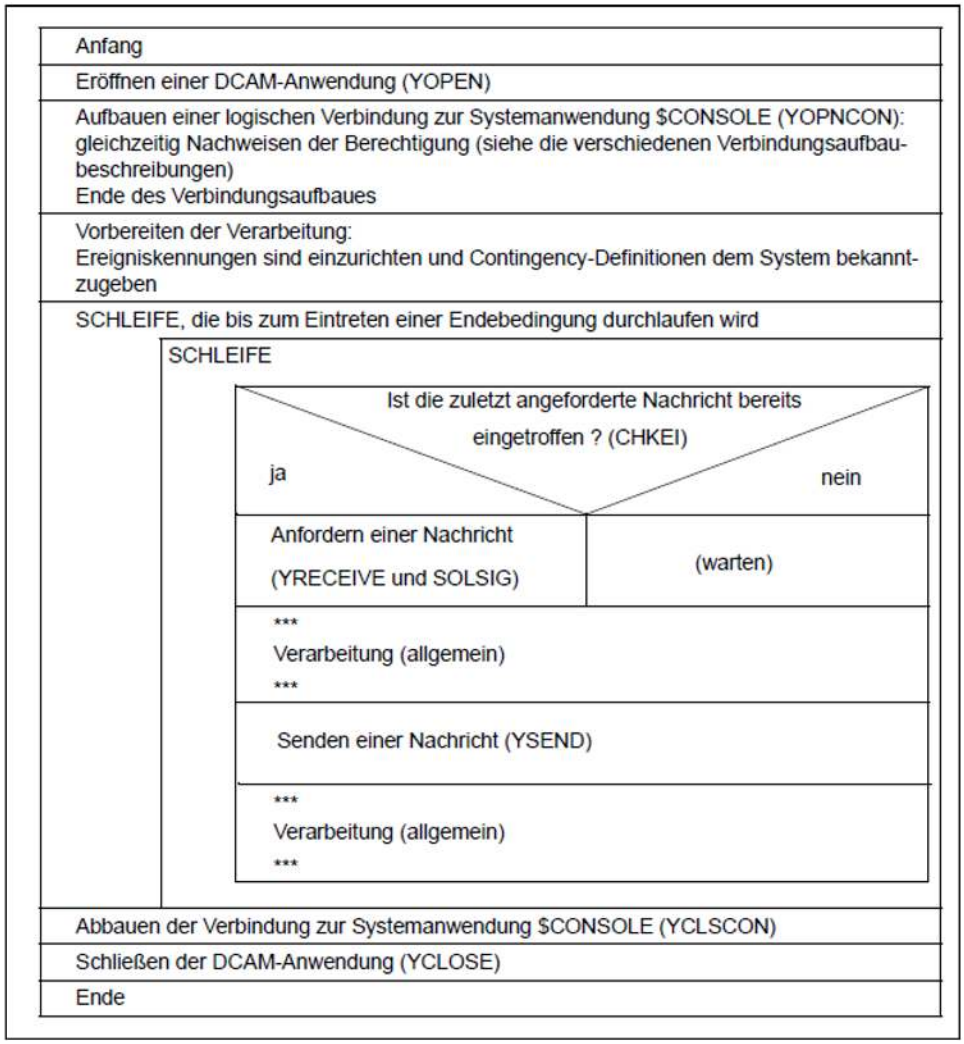


Bild 28: Grobstruktur für ein berechtigtes Benutzerprogramm (Meldungen asynchron verarbeiten)

Überwachung der Abnahme von Nachrichten

Von der UCON-Task wird zyklisch (jede Minute) überprüft, ob berechtigte Benutzerprogramme die an sie verschickten Nachrichten entgegennehmen. Wird festgestellt, dass die nächste zur Ausgabe anstehende Meldung schon länger als eine Minute auf den Empfang durch das berechtigte Benutzerprogramm wartet, wird das mit der Meldung `NBR0600` an der Konsole gemeldet.

Wird ferner festgestellt, dass die nächste Meldung länger als die durch den Systemparameter NBRCSCK bzw. NBRCSCKN festgelegte maximale Wartezeit zur Ausgabe ansteht, reagiert das System entsprechend der DISCON-Angabe in der Verbindungsnachricht (siehe "Anschlüsse mit dynamischen Berechtigungsnamen"):

DISCON=YES: Die Verbindung des betreffenden berechtigten Benutzerprogrammes zur UCON-Task ($CONSOLE) wird abgebrochen.

DISCON=NO: Alle Meldungen, die zur Ausgabe an das berechtigte Benutzerprogramm anstehen, werden gelöscht und durch eine einzelne Meldung `NBR0601` ersetzt.

17.1.4 Nachrichtenformate

Nachrichtenformate und Protokollversionen

Jede Nachricht, die ein berechtigtes Benutzerprogramm sendet, darf höchstens 31 KB lang sein. Längere Nachrichten werden mit `REJ1` abgewiesen. Nachrichten, die das Benutzerprogramm empfangen soll, sind damit maximal 31 KB groß plus die Länge des ggfs. hinzugefügten Headers (siehe "Makro NBMHE"), denn von anderen Absendern stammende Nachrichten sind sowieso viel kleiner (so beträgt die maximale Länge einer mit MSG7X ausgegebenen Meldung z.B. 4 KB).

Der Text dieser Nachrichten hat grundsätzlich das gleiche Format wie bei Ein-/Ausgaben an physikalischen Konsolen (siehe voriges Kapitel). Zusätzliche Formate ergeben sich ausschließlich, wenn das berechtigte Benutzerprogramm als Server eines Operator-Spezialkommandos fungiert. Diese Erweiterungen sind in ab "Operator-Spezialkommandos in berechtigten Benutzerprogrammen" beschrieben.
Für die automatische Auswertung der empfangenen Meldungen ist ein reines Textformat jedoch ungünstig, da jede Einzelinformationen – z.B. der Wert eines bestimmten Inserts in einer Meldung – in dem Text erst gesucht werden müsste. Das Format von Ausgabetexten ist aber keine garantierte, bei Versionswechsel unverändert bleibende Schnittstelle.
Das System bietet daher an, mit berechtigten Benutzerprogrammen formatierte Nachrichten auszutauschen, die neben dem Text zusätzliche Informationen enthalten. Gesteuert wird dies durch die Protokollversion, die das berechtigte Benutzerprogramm beim Verbindungsaufbau zu $CONSOLE beantragt. Dieser Vorgang ist für Anschlüsse unter generiertem und unter dynamischem Berechtigungsnamen unterschiedlich und kann den entsprechenden Abschnitten auf "Anschlüsse mit generierten Berechtigungsnamen" entnommen werden.
Die Bedeutung der ausgewählten Protokollversion ist jedoch in allen Fällen gleich.

Protokollversion 0

Hinter dem Begriff Protokollversion 0 verbirgt sich die einfache Vereinbarung, dass sich der Nachrichtenaustausch zwischen System und berechtigtem Benutzerprogramm in beide Richtungen auf den Nachrichtentext beschränkt. Zusätzliche Informationen werden nicht ausgetauscht.

Protokollversion 1

Vereinbart das berechtigte Benutzerprogramm mit der $CONSOLE-Schnittstelle die Protokollversion 1, erwartet das System als Eingabe weiterhin nur Texte der bekannten Formate. Beim Senden von Nachrichten durch das berechtigte Benutzerprogramm bestehen also keinerlei Unterschiede zu Protokollversion 0.
Nachrichten vom System erhält das berechtigte Benutzerprogramm jedoch in erweiterter Form, bestehend aus einem Nachrichtenkopf (Header), dem eigentlichen Text, so wie er bei Protokollversion 0 netto übertragen worden wäre, sowie, falls es sich bei der Nachricht um eine mit MSG7[X] ausgegebene Meldung handelt, ein Füllbyte mit nachfolgender datenorientierten Beschreibung („Mapping-Format“) des Meldungsaufbaus.

Eine DSECT zur Beschreibung des Nachrichtenkopfes kann durch den Makro NBMHE erzeugt werden, der im Anschluss beschrieben ist.
Aufbau und Inhalt der datenorientierten Meldungsbeschreibung sind der Beschreibung des Makros NBMAP auf "Makros NBMAP" zu entnehmen.

Protokollversion 2

Vereinbart das berechtigte Benutzerprogramm mit der $CONSOLE-Schnittstelle die Protokollversion 2, erhält es alle Nachrichten vom System in erweiterter Form. Bezüglich des Empfangs von Meldungen durch das berechtigte Benutzerprogramm besteht also kein Unterschied zu Protokollversion 1.
Vom berechtigten Benutzerprogramm an das System gesendete Nachrichten müssen hier jedoch die gleiche, erweiterte Form aufweisen; die Übertragung des reinen Nachrichtentextes reicht nicht mehr aus.
Im Regelfall bringt die Erzeugung des Nachrichtenkopfes für das berechtigte Benutzerprogramm lediglich zusätzlichen Aufwand (und Fehlerquellen), aber keinen besonderen Nutzen mit. Notwendig ist die Verwendung der Protokollversion 2 ausschließlich, wenn das berechtigte Benutzerprogramm als Server für ein verwaltbares Operator-Spezialkommando dienen soll. Die dann geltenden Besonderheiten sind ab "Operator-Spezialkommandos in berechtigten Benutzerprogrammen" beschrieben.

Makro NBMHE

Der Makro NBMHE beschreibt das Format des Nachrichtenheaders. Die Assembler-Schnittstelle hat folgendes Aufrufformat:

Makro	Operanden
NBMHE	MF=D/C, PREFIX=prefix, MACID=macid, VER=1/2/3

Für den Benutzer ist die Angabe VER=1 nicht erlaubt!

Der Header hat folgenden Aufbau:

Parameter	V	DS	AT	LH	L2	L1	D	AK	SP	BC	RC2	RC1	FL	UID
gültig für Version	123	123	123	123	123	123	123	23	23	23	23	23	3	3

Parameter	Bedeutung
V	Versionsnummer des Makros NBMHE, der diese Struktur generiert; 1 Byte
DS	Dialogstatusbyte; 1 Byte X'80' Anmeldungsdialog läuft X'40' Verbindung vorläufig akzeptiert X'10' endgültige Akzeptierung der Anmeldung X'08' Anmeldung abgewiesen X'04' Operator-Logon angefordert X'02' Kennwort angefordert
AT	Auftragstyp; 1 Byte Werte bei Makro V01: % ? / . * ; zusätzliche Werte ab Makro V02: + ! & :
LH	Länge des Headers; 2 Byte
L2	Distanz zum datenorientierten Teil; 2 Byte (L2=0000, falls kein datenorientierter Teil (mit NBMAP beschrieben) vorhanden ist)
L1	Länge des textorientierten Teils; 2 Byte
D	Ziel einer Meldung; 4 Byte
Erweiterung für NBMHE Version 02 und 03:	
AK	Auftragskennzeichen abhängig vom Auftragstyp; 4 Byte
SP	Indikator für Senderprofil; 1 Byte
BC	Berechtigungscodemenge; 5 Byte
RC2	Kommandoreturncode Subcode2 zum Verschicken an bzw. beim Empfang von Kommandogeber; 1 Byte
RC1	Kommandoreturncode Subcode1 zum Verschicken an bzw. beim Empfang von Kommandogeber; 1 Byte
Erweiterung für NBMHE Version 03:	
FL	Flag-Byte, das den Typ des Kommandogebers ausgibt; 1 Byte
	X'80' X'40' X'20'
UID	Benutzerkennung des Kommandogebers; 8 Byte

Hinweise zu den einzelnen Feldern

Die meisten Felder des NBMHE sind nur in bestimmten Zusammenhängen relevant, die in der Tabelle aufgeführt sind. Sollte der jeweilige Zusammenhang nicht gegeben sein, so bedeutet dies bei vom System empfangenen Nachrichten einheitlich, dass der Feldinhalt undefiniert ist, und bei an das System gesendeten Nachrichten, dass der Feldinhalt vom System nicht ausgewertet wird, also beliebig gewählt werden kann.
Ferner ist der von der Art des verwendeten Berechtigungsnamens anhängige Anmeldedialog des berechtigten Benutzerprogrammes aus Gründen der Übersichtlichkeit separat beschrieben, siehe "Anschlüsse mit generierten Berechtigungsnamen". Die Tabelle bezieht sich auf den – von der Berechtigungsnamensart unabhängigen – Nachrichtenaustausch nach erfolgreichem Verbindungsaufbau (Dialogstatusbyte = 0).

V Immer relevant
Die Versionsnummer wird durch den Parameter VER des NBMHE-Makros bestimmt. Das System verwendet für seine Nachrichten stets die neueste Version. Das Benutzerprogramm ist jedoch nicht verpflichtet, beim Senden von Nachrichten die gleiche Version zu verwenden; mit Ausnahme von Version 1 akzeptiert das System alle kleineren Versionen kompatibel und eventuelle größere Versionen mit der Einschränkung, dass neu hinzugekommene Felder natürlich nicht berücksichtigt werden können.

DS Immer relevant
Das Dialogstatusbyte zeigt an, in welcher Phase des Anmeldungsdialogs sich das berechtigte Benutzerprogramm befindet. Von Null verschiedene Werte können ausschließlich während des Anmeldedialogs vorkommen.
Nach erfolgreichem Verbindungsaufbau ist das Byte immer 0.

AT Immer relevant
Typkennzeichen der Nachricht (siehe "Nachrichtenaustausch der Operator").

LH Länge des Headers, gleichzeitig – da der Nachrichtentext unmittelbar hinter dem Header folgt – die Distanz vom Anfang der Nachricht zum Text.
Das Feld muss mit der angegebenen NBMHE-Version übereinstimmen (d.h. 25 Byte für Version 2, 34 Byte für Version 3). Um auch für künftige Versionen korrekt arbeiten zu können, muss die Länge trotzdem immer dem Längenfeld entnommen werden.

L2 (DLL2) Immer relevant
Distanz vom Anfang der Nachricht zum datenorientierten Teil (NBMAP).
Ist kein datenorientierter Teil vorhanden, ist die Distanz 0.
Wichtig: Ist ein datenorientierter Teil vorhanden, darf er nicht unmittelbar hinter dem Nachrichtentext liegen, sondern muss durch mindestens ein Füllbyte von ihm getrennt sein. L2 ist also zwangsläufig größer als LH + L1.

L1 (DLL1) Immer relevant

D Nur relevant bei empfangenen Nachrichten

AK Nur relevant beim Empfang von Kommandos (AT = „ / “) und Kommandozusatzinformationen (AT = „ : “) sowie beim Senden von Nachrichten im Rahmen der Kommandobehandlung (näheres siehe „Operator-Spezialkommandos“, ab "Operator-Spezialkommandos in berechtigten Benutzerprogrammen").

SP Nur relevant beim Senden von Nachrichten
Dieses Byte muss immer auf Null gesetzt werden. Andere Werte sind für Systemanwendungen reserviert.

BC Nur relevant, wenn SP ungleich Null ist (also ausschließlich für Systemanwendungen).

RC2 Nur bei Nachrichten vom Typ „Kommandoende“ (AT = „ ! “) relevant

RC1 Nur bei Nachrichten vom Typ „Kommandoende“ (AT = „ ! “) relevant

FL Der Begriff „SCI-Task“ bezeichnet hier eine Benutzertask mit dem zur Eingabe des Kommandos notwendigen Privileg (bei selbstdefinierten Operator-Spezialkommandos muss dies nicht das OPERATING-Privileg sein).

UID Nur beim Empfang von Kommandos (AT = „ / “) relevant

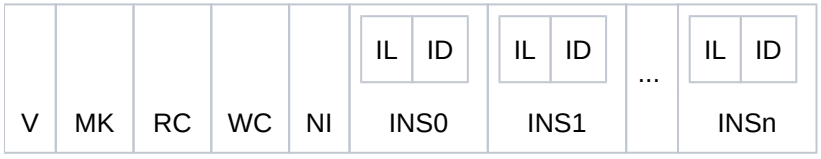
> Die Versionsnummer des Headers ist nicht mit der Protokollversionsnummer der Verbindung zu verwechseln.
>
> Das Feld LH befindet sich in allen Versionen auf Distanz X'03' im Header. Ihm ist die Distanz zum Beginn des Folgetextes zu entnehmen, da der Folgetext unmittelbar hinter NBMHE beginnt.

Makro NBMAP

Der Makro **NBMAP** beschreibt den Nachspann der Nachricht. Die Assembler-Schnittstelle hat folgendes Aufrufformat:

Makro	Operanden
NBMAP	MF=D/C/L,PREFIX=chars 1, MACID=macid 1..3, VERSION=1 ,MSGKEY=chars 7..7,MSGRC=chars 1..1 ,MSGWEIGHT=msgweight,MSGINSNUM=integer 0..15 ,INSERTS=((length1,distance1),...,(length,distancex))

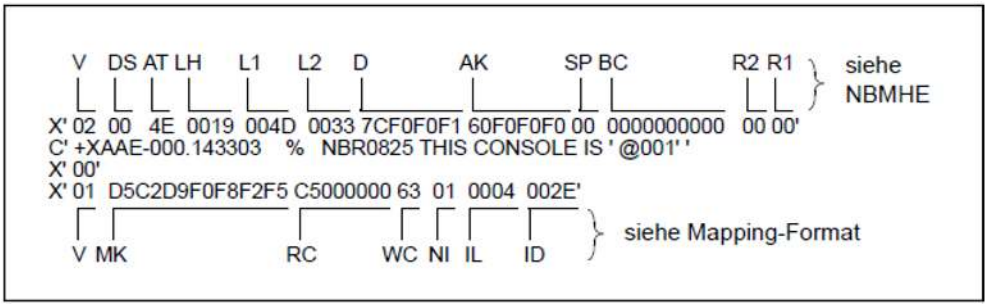
Die Struktur hat folgenden Aufbau:



Parameter	Bedeutung
V	Versionsnummer des Makros NBMAP, der diese Struktur generiert; 1 Byte
MK	Meldungsschlüssel; 7 Byte
RC	Routing-Code; 4 Byte (linksbündig, restliche Stellen mit 0 aufgefüllt)
WC	Gewicht der Meldung; 1 Byte
NI	Anzahl der Inserts; 1 Byte

IL	Länge des aktuellen Inserts; 2 Byte
ID	Distanz des aktuellen Inserts ab Anfang des Meldungstextes; 2 Byte
INS0 ....INSn	0...n Inserts der Meldung

Beispiel mit Nutzung des Nachrichten-Headers V02



17.2 Operator-Spezialkommandos in berechtigten Benutzerprogrammen

Ein Spezialkommando ist ein besonderes Operatorkommando, dessen Format und Funktion die Systembetreuung festlegt.
Das System leitet ein empfangenes Spezialkommando einem berechtigten Benutzerprogramm zu, das das Kommando auswertet und die gewünschten Tätigkeiten ausführt. Die Bearbeitung von Spezialkommandos ist somit eine spezielle Anwendung von berechtigten Benutzerprogrammen.

Es gilt:

- Für jedes Spezialkommando ist eine Operationsbezeichnung (= Name) festzulegen. Diese darf nicht mit den Operationsbezeichnungen der gewöhnlichen Operatorkommandos übereinstimmen. Die Operationsbezeichnung „CANCEL“ ist z.B. als Operationsbezeichnung für ein Spezialkommando unzulässig.
- Jedes Spezialkommando ist einem Aufgabengebiet zuzuteilen (das Kommando wird mit einem Berechtigungsschlüssel versehen). Falls ein Spezialkommando sich keinem der bestehenden Aufgabengebiete zuordnen lässt, ist es einem privaten Aufgabengebiet zuzuteilen (Berechtigungsschlüssel: W, X, Y, Z).
- Um die Eingabe eines Spezialkommandos auch aus einer Benutzertask mit Privileg OPERATING zu ermöglichen, ist es in einer SDF-Syntaxdatei mit SDF-A zu definieren. Diese SDF-Syntaxdatei muss der Benutzertask vorgeschaltet werden.
  Mit ADD-CMD ...,IMPLEMENTOR=APPLICATION wird festgelegt, dass die Kommandobearbeitung in einem berechtigten Benutzerprogramm erfolgt.

Für das Verhalten von Operator-Spezialkommandos aus der Sicht des Benutzers ist maßgebend, ob die Kommandos die Eigenschaft der Verwaltbarkeit besitzen.
Man unterscheidet verwaltbare und nicht verwaltbare Operator-Spezialkommandos.

Verwaltbare Spezialkommandos

Sie werden vom System genauso verwaltet wie gewöhnliche Operator-Kommandos.
Der Austausch von kommandogebundenen Nachrichten ist möglich.
Die Kommando-Endemeldung wird – außer in Fehlerfällen – von der kommandoausführenden $CONSOLE-Anwendung veranlasst und enthält einen auswertbaren Returncode.

Nicht-verwaltbare Spezialkommandos

Sie werden vom System als „beendet“ angesehen, sobald sie (mit DCAM) erfolgreich an die kommandoausführende $CONSOLE-Anwendung abgeschickt wurden. Der Austausch von kommandogebundenen Meldungen ist sonst nicht möglich. Verwendet der Operator ein nicht-verwaltbares Spezialkommando, so erhält er die Meldung

```
NBR0740 COMMAND TERMINATED '<cmd-name>':
        RESULT: (SC=00, SC1=00,MC=NBR0768); DATE:<date>
```

Das bedeutet jedoch nur, dass das Kommando an den zuständigen Server weitergegeben wurde. Das endgültige Kommandoergebnis findet sich in keiner Abschlussmeldung und Kommandoauftragskennzeichen finden sich nicht als Meldungsauftragskennzeichen in den Kommandoergebnissen wieder.

Kommando	Bedeutung
CONNECT-CMD-SERVER	Verknüpfung von berechtigten Anwendungen mit Operatorkommandos herstellen
DISCONNECT-CMD-SERVER	Verknüpfung von berechtigten Anwendungen mit Operatorkommandos lösen
SHOW-CMD-ATTRIBUTES	Informationen über Operatorkommandos ausgeben

Tabelle 47: Kommandoübersicht Operator-Spezialkommandos

17.2.1 Kommandodefinition

Alle Eingaben, die an einer Konsole gemacht werden, gehen zunächst an UCON, die zentrale Konsoltask des Systems. Handelt es sich bei der Eingabe um ein Kommando, muss UCON zunächst die an den Routing-Code des Kommandos geknüpfte Berechtigungsprüfung durchführen. Erst wenn die bestanden ist, darf das Kommando an den zuständigen Kommandoserver weitergeleitet werden.
Da der Routing-Code eines Kommandos (und bei Operator-Spezialkommandos auch der Kommandoserver) jedoch nicht Bestandteil der SDF-Syntaxdatei ist, muss sich UCON eine eigene Kommandotabelle halten, in der die benötigten Informationen enthalten sind. Kommandos, die in der UCON-Kommandotabelle nicht enthalten sind, können an Konsolen nicht ausgeführt werden. Operator-Spezialkommandos müssen daher als Erstes in diese Tabelle eingetragen werden.

Statische Einträge

Eine Möglichkeit zur Eintragung von Operator-Spezialkommandos in die UCON-Kommandotabelle ist die Parameterservice-Anweisung ADD-CMD-ENTRY (siehe "Anweisung ADD-CMD-ENTRY"). Die dadurch erzeugten Einträge sind statisch, d.h. bleiben für die gesamte Dauer der Session erhalten. Als zuständiger Kommandoserver kann daher nur eine $CONSOLE-Anwendung mit (statisch) generiertem Berechtigungsnamen angegeben werden (ist der angegebene Berechtigungsname noch nicht bekannt, wird er automatisch generiert).
Ferner besteht hier keine Möglichkeit, das Kommando als verwaltbar zu deklarieren.

Soll ein Operator-Spezialkommando verwaltbar ablaufen oder von einer $CONSOLE-Applikation mit dynamischen Berechtigungsnamen ausgeführt werden, muss der Eintrag in die Kommandotabelle mit Hilfe des Kommandos CONNECT-CMD-SERVER dynamisch erfolgen.

Dynamische Einträge

Eine $CONSOLE-Anwendung, die als Server für ein Operator-Spezialkommando fungieren soll, kann sich selbst mit dem Kommando CONNECT-CMD-SERVER dynamisch bei UCON als Kommandoserver anmelden. UCON erzeugt dann automatisch einen entsprechenden Eintrag in der Kommandotabelle.
Zu beachten ist dabei, dass pro Kommando bis zu vier Einträge in der Kommandotabelle erlaubt sind. CONNECT-CMD-SERVER kann also auch für bereits eingetragene Kommandos (auch Systemkommandos) abgesetzt werden (wobei allerdings nicht alle Eigenschaften des Kommandos geändert werden können, siehe Kommandobeschreibung). Der neue Eintrag „überlagert“ dann den alten, und die Kommandos werden fortan der $CONSOLE-Anwendung zugestellt, die den neuesten Eintrag besitzt. Welche Einträge alles existieren, lässt sich mit dem Kommando SHOW-CMD-ATTRIBUTES anzeigen.

Ein dynamischer Eintrag in der UCON-Kommandotabelle wird automatisch wieder entfernt, wenn die Verbindung von $CONSOLE zum zuständigen Kommandoserver abgebaut wird. Ferner kann die Anwendung ihn mit Hilfe des Kommandos DISCONNECT-CMD-SERVER explizit löschen.
Ist der entfernte Eintrag der einzige für dieses Kommando gewesen, ist das Kommando anschließend nicht mehr bekannt. Gibt es weitere Einträge, so wird (bzw. bleibt) der neueste der noch existierenden Einträge wirksam; der Eigentümer dieses Eintrags bekommt die Kommandos zugestellt.

Definition in der Syntaxdatei

SDF-A bietet die Möglichkeit, Syntaxdatei-Einträge für Operator-Spezialkommandos zu erstellen (Anweisung ADD-CMD <name>, IMPLEMENTOR= *APPLICATION). Ein solcher Syntaxdatei-Eintrag ist notwendig, wenn das Kommando Kennwörter enthält, die nicht in der CONSLOG protokolliert werden sollen. Ein solches Kommando ist mit PASSWORD-POSSIBLE=*YES in die UCON-Kommandotabelle einzutragen. Die CLOG-Task ruft dann vor der Protokollierung in die CONSLOG SDF auf und lässt alle als „geheim“ (SDF-A: ADD-OPERAND <name>, SECRET-PROMPT=*YES) gekennzeichneten Operanden entfernen. Existiert kein Syntaxdatei-Eintrag, wird das Kommando unverändert im Klartext protokolliert.

Darüber hinaus können Operator-Spezialkommandos, für die ein Syntaxdatei-Eintrag vorhanden ist, nicht nur an Konsolen verwendet werden, sondern in allen Benutzertasks, die gemäß den in der Syntaxdatei eingetragenen Kommandoprivilegien dazu berechtigt sind. Eine Berechtigungsprüfung anhand des Routing-Codes findet bei Eingabe in einer Benutzertask nicht statt.

Soll ein Operator-Spezialkommando durch Eintrag in die Systemsyntaxdatei einem bestimmten Benutzerkreis zur Verfügung gestellt werden, ist folgende Besonderheit zu beachten:
Der Syntaxdatei-Eintrag ersetzt den Eintrag in die UCON-Kommandotabelle nicht. Kommandos, die zwar in der Systemsyntaxdatei, aber nicht in der UCON-Kommandotabelle eingetragen sind, werden mit der Meldung `NBR0744` zurückgewiesen. Ebenso wird mit zusätzlichen, UCON unbekannten Alias-Namen eines eigentlich bekannten Kommandos verfahren.

Das Privileg OPERATING kann an beliebige Benutzerkennungen vergeben werden. Damit verbunden ist die Erlaubnis, fast alle sog. Operator-Kommandos nutzen zu können.

17.2.2 Nachrichtenformate

Folgende Formate werden beschrieben:

- Format der Kommandos beim Empfangen von Operatorkommandos
- Format beim Senden von Kommandoergebnissen und Kommando-Zusatzinformationen
- Format beim Senden von Kommandoende-Nachrichten
- Format beim Empfang von Kommando-Zusatzinformationen

Format der Kommandos beim Empfangen von Operatorkommandos

```
[header]/{tsn | an | (mn)}:  cmd[ [operands]
```

header	Bei Anwendungen, die sich an $CONSOLE mit der Protokollversion 01 oder 02 konnektiert haben, enthält das Format einen n Zeichen langen Header, der durch den Makro NBMHE beschrieben wird.
/	Nachrichtentyp-Kennzeichen (Schrägstrich)
tsn	Auftragsnummer der Benutzertask, von der das Operator-Spezialkommando eingegeben wurde
an	Berechtigungsname des berechtigten Benutzerprogramms, von dem das Kommando eingegeben wurde (4 Zeichen)
mn	Mnemotechnischer Gerätename der physikalischen Konsole, an der das Kommando eingegeben wurde (2 Zeichen)
:	Doppelpunkt als Trennzeichen
' '	Leerzeichen
cmd	Bezeichnung eines Kommandos
operands	Operanden des Kommandos

Die Auftragsidentifikation, die ein berechtigtes Benutzerprogramm beim Empfangen des Kommandos im Header mitbekommt, ist eine vom System vergebene, eindeutige Auftragsidentifikation. Sie ist nicht identisch mit dem bei der Kommadoeingabe vergebenen Auftragskennzeichen.
Die interne Auftragsidentifikation ist für das berechtigte Benutzerprogramm transparent. Das System verwaltet das Kommando anhand dieser Auftragsidentifikation und sorgt dafür, dass die in Zusammenhang mit dem Kommando gesendeten Kommandoergebnisse, die Zusatzanforderungen und die Kommandoendemeldung stets mit dem vom Kommadogeber eingegebenen Auftragskennzeichen versorgt werden und den Kommandogeber erreichen.
Vorrausetzung ist, dass im Header die interne Auftragsidentifikation und der Auftragstyp (Kommandoergebnis (+), Kommando-Zusatzinformation-Anforderung (&) oder Kommandobeendigung (!)) richtig versorgt sind.

Beispiele

1. ohne Header

   ```
   /OPR1: SPECI OP1=VALUE1,OP2=VALUE2
   ```

2. mit Header

   ```
   <header>/OPR1: SPECI OP1=VALUE1,OP2=VALUE2
   ```

   Zur Datenstruktur des Headers siehe Makro NBMHE ("Nachrichtenformate").

Format beim Senden von Kommandoergebnissen und Kommando-Zusatzinformationen

```
<header> text [<mapping-format>]
```

header	Im Header muss der entsprechende Auftragstyp gekennzeichnet und das Feld Auftragsidentifikation mit der aus dem Header des Kommandos entnommenen Auftragsidentifikation versorgt werden. Das System sorgt dafür, dass der Text an der Kommandoeingabestelle mit dem im Kommando enthaltenen Auftragskennzeichen erscheint.
text	formatfreier (beliebiger) Text oder Nachrichtentext im MSG7X-Format
mapping-format	Sendet eine $-CONSOLE-Anwendung Nachrichten des Typs „+“ oder „&“ unter Angabe eines Mapping-Formats (siehe Makro NBMAP, "Nachrichtenformate"), und war der Absender des Kommandos eine Benutzertask, so ignoriert das System für die Meldungsausgabe den angegebenen Nachrichtentext. Allerdings wird der angegebene Nachrichtentext zur Aufnahme in die CONSLOG herangezogen. Er sollte daher MSG7X-Format haben. Desweiteren werden systeminterne Prüfungen (z.B. die Meldungsnummer an Distanz 4 im Nachrichtentext) durchgeführt, die i.d.R. nur bei einem MSG7X-Format erfolgreich sein können. Der Absender des Kommandos erhält als Nachricht die Ausgaben, die der Makro MSG7X mit den im Mapping-Format angegebenen Parametern erzeugt. Dabei wird automatisch die in der Umgebung des Kommandosenders gültige Sprache verwendet. Der Zweck dieses MSG7X-Aufrufes ist es also, die sprachlich erwünschte Nachrichtentext-Ausgabe zu erreichen.

Format beim Senden von Kommandoende-Nachrichten

```
header maincode
```

header — Im Header muss der Auftragstyp „!“ eingetragen und die Felder RC1 und RC2 mit dem Subcode1 bzw. dem Subcode2 des Kommandoergebnisses versorgt werden.
Ferner muss das Feld Auftragsidentifikation mit der aus dem Header des Kommandos entnommenen Auftragsidentifikation versorgt werden. Das System sorgt dafür, dass die Meldung `NBR0740` an der Kommandoeingabestelle ausgegeben wird.

```
NBR0740 COMMAND COMPLETED 'xxxx'; (RESULT:....); DATE:.....
```

Die Meldung enthält auch das bei der Eingabe des Kommandos eingegebene Auftragskennzeichen und ist mit einem Ausrufezeichen gekennzeichnet.

maincode — 7 Zeichen langer Maincode des Kommandoergebnisses

> Nach dem Verschicken einer Kommandobeendigungsanzeige darf ein berechtigtes Benutzerprogramm die interne Auftragsidentifikation nicht mehr weiter verwenden. Aufträge für Kommandos, deren Beendigung bereits angezeigt wurde, oder Aufträge mit falscher Auftragsidentifikation werden vom System mit der Meldung `REJ11` abgewiesen.

Format beim Empfang von Kommando-Zusatzinformationen

```
header :{tsn | an | (mn)}[-mid]:[text]
```

header	n Zeichen langer Header, der durch den Makro NBMHE beschrieben wird
:	Nachrichtentyp-Kennzeichen (Doppelpunkt)
an	Berechtigungsname des antwortenden berechtigten Benutzerprogramms (4 Zeichen)
mn	Mnemotechnischer Gerätename der antwortenden physikalischen Konsole (2 Zeichen)
tsn	Auftragsnummer der antwortenden Benutzertask
-	Bindestrich
mid	Meldungsauftragskennzeichen (A..Z, 0-9, @, #, $); wird bis zu 3 Zeichen mit führenden Nullen ergänzt. Standardwert = 000
:	Doppelpunkt als Trennzeichen
text	beliebiger Antworttext

17.3 Softwareprodukte OMNIS und PROP-XT

Das Softwareprodukt **OMNIS** ist ein Steuersystem zur zentralen Bedienung mehrerer Anwendungen in BS2000.

In BCAM kann eine Datenstation prinzipiell zu einem Zeitpunkt nur eine einzige Verbindung zu einem Kommunikationspartner unterhalten. OMNIS ist ein Programm, das es ermöglicht, diese Einschränkung zu umgehen.

OMNIS ermöglicht die gleichzeitige Verbindungsaufnahme von einer Datenstation zu mehreren Partnern in einem Rechnerverbund. Über OMNIS kann man mit einem Partner auch mehrfach in Verbindung treten. Außerdem können Sie mit OMNIS auch 9750-Datenstationen für Anwendungen verwenden, die Datenstationen eigentlich gar nicht unterstützen, wie z.B. UCON.

OMNIS ist im Handbuch „OMNIS/OMNIS-MENU“ [33] ausführlich beschrieben.

PROP-XT ist ein Produkt im Rahmen der Automatisierung im Data Center für das Aufgabengebiet der Server-Steuerung.
Es ermöglicht programmiertes Operating durch vom Benutzer mit den komfortablen Sprachmitteln von SDF-P erstellte Administrationsprozeduren.

PROP-XT ist im Handbuch „PROP-XT“ [40] ausführlich beschrieben.

17.4 Kommandodateien für den Operator

Kommandodateien beim Systemstart

Nach abgeschlossener Systemeinleitung hat der Operator gewöhnlich den Dialogbetrieb vorzubereiten. Er nimmt unter anderem das Datenkommunikationssystem in Betrieb und leitet die Teilhaberanwendungen ein.

> ¡ Das Starten des Datenkommunikationssystems kann bereits vor „System Ready“ erfolgen. Zu diesem Zweck können in der Startup-Parameterdatei (Abschnitt BCAM) die entsprechenden BCAM-Kommandos eingetragen werden. Siehe dazu auch das Handbuch „BCAM“ [4].

Die Kommandos, die zur Vorbereitung des Dialogbetriebs erforderlich sind, lassen sich in einer Kommandodatei hinterlegen.
Wenn der Name dieser Kommandodatei bereits im Startup-Parameterservice (Systemparameter CMDFILE) festgelegt wurde, so bearbeitet das System die Datei automatisch unmittelbar nach dem Systemstart.
Der Operator kann entweder durch Änderung des Systemparameters CMDFILE mittels Parameterservice oder beim DIALOG-Startup eine andere Kommandodatei ausführen lassen oder manuell Kommandos eingeben (siehe auch Systemparameter NBRUNUID).

Kommandodateien bei häufig vorkommenden Kommandofolgen

Kommandofolgen, die der Operator wiederholt einzugeben hat – beispielsweise bei häufig wiederkehrenden Aufträgen – lassen sich in Kommandodateien speichern.

Der Operator startet die Ausführung einer Kommandodatei mit einem RUN-Kommando. Dies kann entweder von der physikalischen Konsole aus erfolgen, oder das RUN-Kommando wird von einem berechtigten Benutzerprogramm abgesetzt. In beiden Fällen muss darauf geachtet werden, dass der aufrufenden physikalischen Konsole oder dem berechtigten Benutzerprogramm alle Aufgabengebiete zugeordnet sind, auf die sich die Kommandos in der Kommandodatei beziehen. Neben den Kommandodateien kann die Systembetreuung ENTER-Jobs aufbauen, die der Operator von der Konsole startet. Da Kommandofolgen in ENTER-Jobs Benutzerkommandos enthalten und nach dem Start keinerlei besondere Maßnahmen seitens des Operators erfordern, wird im Folgenden nicht weiter auf diese Kommandofolgen eingegangen.

Das Anbieten der Operator-Funktionalität in beliebigen Benutzertasks mit dem Privileg OPERATING ermöglicht auch das Bereitstellen von beliebigen Prozeduren (mit und ohne SDF-P) für den Operator. Diese Kommandodateien können eine beliebige Mischung aus Operator- und Benutzerkommandos enthalten. Das Starten der Prozeduren mit CALL-PROCEDURE und ENTER-PROCEDURE ist jedoch nur aus Benutzertasks möglich und nicht von physikalischen Konsolen oder berechtigten Anwendungen. Die Funktionalität entspricht in vollem Umfang den Prozeduren in Benutzertasks und wird hier nicht beschrieben.

Die nachstehenden Ausführungen beziehen sich also ausschließlich auf Operator-Kommandofolgen.

Kommando	**Bedeutung**
AGOGO	Kommandodatei fortsetzen
ASTOP	Bearbeitung der Kommandodatei anhalten und die Bedingung zum Fortsetzen festlegen
CANCEL-RUN-PROCESS	Abbrechen einer Kommandodatei
ENTER-JOB	Starten eines ENTER-Jobs
RUN	Kommandodatei starten
SHOW-PENDING-MSG	Wiederholtes Anzeigen von Systemmeldungen

Tabelle 48: Kommandoübersicht zu Kommandodateien

17.4.1 Ausführen und Abbrechen einer Kommandodatei

Ausführen einer Kommandodatei

Eine Kommandodatei lässt sich von jeder physikalischen Konsole und von jedem berechtigten Benutzerprogramm aufrufen, sofern sie den Berechtigungsschlüssel E besitzen (siehe Abschnitt „Berechtigte Benutzerprogramme mit Operatorfunktionen"). Der aufrufenden Konsole bzw. dem Benutzerprogramm müssen jedoch alle Aufgabengebiete zugeordnet sein, auf die sich die Kommandos in der Kommandodatei beziehen (über den Berechtigungsschlüssel).

Das System bearbeitet zu einem gegebenen Zeitpunkt höchstens eine Kommandodatei. Empfängt es während der Bearbeitung einer Kommandodatei ein weiteres RUN-Kommando, so verzögert es die Ausführung dieses Kommandos so lange, bis die aktuelle Kommandodatei abgearbeitet ist.

Dabei geht das System wie folgt vor:

- Stehen mehrere RUN-Kommandos in einer Kommandofolge, so werden sie in der Reihenfolge bearbeitet, wie sie in der Kommandofolge stehen.
- Wird eine RUN-Prozedur mit dem Kommando CANCEL-RUN-PROCESS gestoppt, so werden auch alle in der RUN-Datei enthaltenen RUN-Kommandos nicht mehr ausgeführt.

Wenn die Ausführung einer Kommandodatei gestartet wird, zeigt dies Meldung `NBR1000` an. Der Aufrufer einer Kommandodatei erhält nur Meldungen, die beim Ausführen der einzelnen Kommandos entstehen; die Kommandos selbst werden ihm i.d.R. nicht angezeigt.
Will der Kommandogeber das Kommando aus der RUN-Datei auf der Konsole mitprotokollieren, kann er dies mit dem Systemparameter NBRUNSP=Y erreichen. Unabhängig vom Systemparameter NBRUNSP werden Kommandos aus RUN-Dateien in der Protokolldatei (CONSLOG) festgehalten.

Enthält eine Kommandodatei Spezialkommandos, so läuft sie nur dann korrekt ab, wenn das berechtigte Benutzerprogramm bereits gestartet worden ist, das die Spezialkommandos empfangen und ausführen soll.

Der Ablauf einer Kommandodatei, die Spezialkommandos enthält, muss daher mit dem Start des berechtigten Benutzerprogramms koordiniert werden.
Dazu sind folgende Aktionen von der RUN-Prozedur aus zu veranlassen:

1. berechtigte Benutzerprogramme zur Bearbeitung von Spezialkommandos starten
   (mit ENTER-JOB-Kommandos)
2. Bearbeitung der Kommandodatei anhalten und die Bedingung zum Fortsetzen festlegen
   (mit einem ASTOP-Kommando)

Um die Anlaufzeit auszunutzen, die die Tasks benötigen, stellt man die ENTER-JOB-Kommandos am günstigsten an den Anfang der Kommandodatei und das ASTOP-Kommando unmittelbar vor das erste Spezialkommando.

Die Bearbeitung der Kommandodatei wird fortgesetzt, wenn eine der folgenden Bedingungen erfüllt ist:

- Es sind so viele AGOGO-Kommandos eingetroffen, wie im ASTOP-Kommando als Bedingung festgelegt ist.
- Die Wartezeit ist verstrichen, ohne dass genügend viele AGOGO-Kommandos eingetroffen sind.

Der Zeitraum, für den die Kommandobearbeitung bei einem ASTOP-Kommando unterbrochen wird, kann mit dem Systemparameter NBRUNWT festgelegt werden.
Er liegt zwischen 10 und 255 Sekunden und beträgt standardmäßig 180 Sekunden.

Beispiel für Spezialkommando

Die Systembetreuung hat für das von ihr betreute System das Spezialkommando SPECI vorgesehen, welches in der Kommandodatei nach dem Systemstart benutzt werden soll.

Diese Datei muss unter anderem folgende Kommandos enthalten:

```
/ENTER-JOB SPEC.CMD-PROC
  :
/ASTOP
/SPECI
   :
```

Die Datei SPEC.CMD-PROC muss dabei eine geeignete Kommandofolge zum Starten des Benutzerprogramms enthalten, das für die Verarbeitung des SPECI-Kommandos erstellt wurde. Die Datei sollte zudem das Kommando AGOGO enthalten, das zusammen mit dem Kommando ASTOP der RUN-Prozedur die Synchronisierung herstellt.

> Einige Kommandos, wie z.B. SHOW-PENDING-MSG, werden vom System mit höherer Priorität bearbeitet als andere Kommandos. Wird ein solches Kommando in einer Kommandodatei angegeben, so kann es andere, vor ihm stehende Kommandos der Kommandodatei „überholen“.

Das Zusammenspiel zwischen der Bearbeitung einer Kommandodatei und den Aktionen eines einzelnen berechtigten Benutzerprogramms zeigt das folgende Bild.

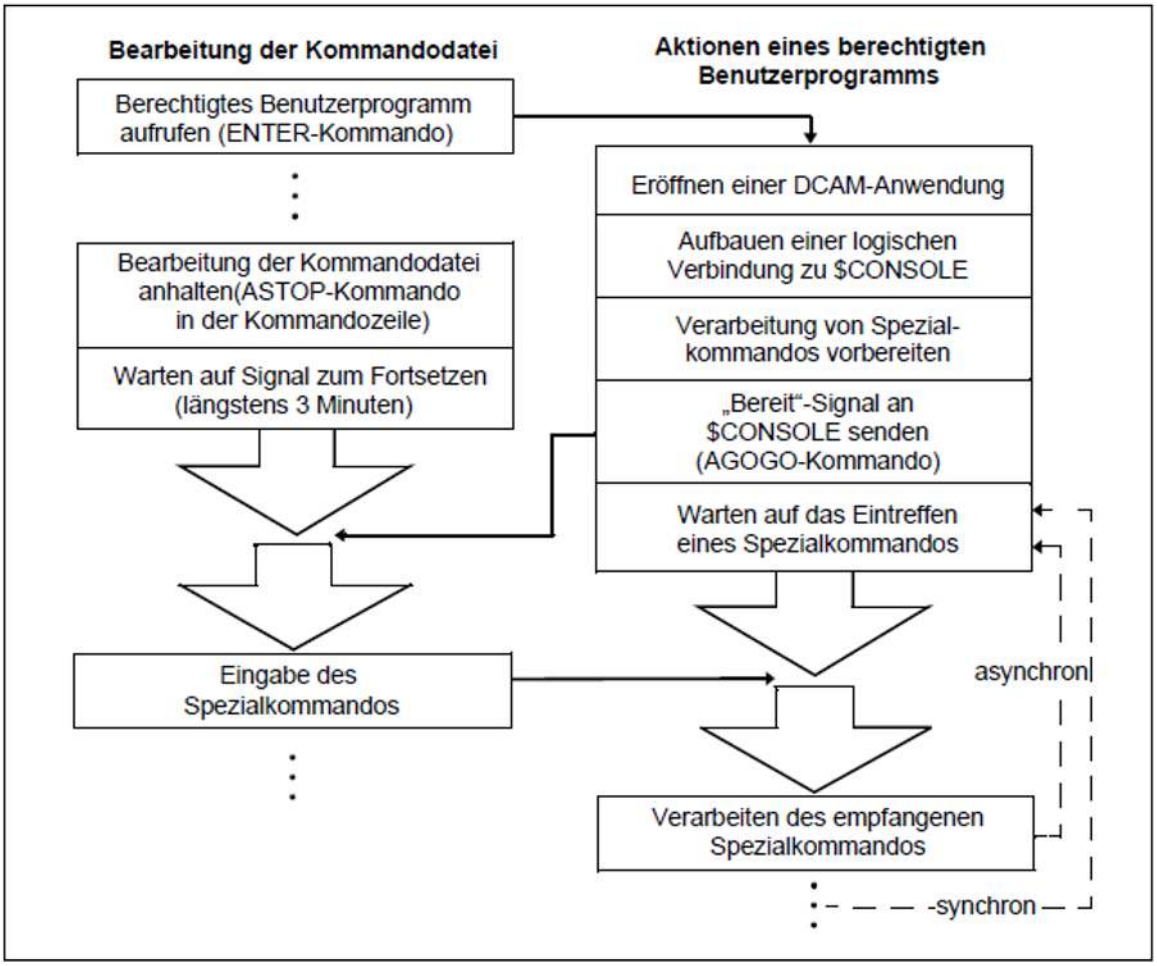


Bild 29: Bearbeiten einer Kommandodatei

Abbrechen einer Kommandodatei

Das Kommando CANCEL-RUN-PROCESS bricht die Bearbeitung einer gestarteten Kommandodatei ab. Zu seiner Ausführung muss der Operator eine RUN-ID angeben, die jeder gestarteten Kommandodatei nach dem Kommando RUN zugeteilt wird. Kommandodateien, die rekursiv aus einer anderen RUN-Sequenz angestartet werden, erhalten dieselbe RUN-ID wie die erzeugende Kommandodatei. Im Falle des Abbruchs einer Kommandodatei mit CANCEL-RUN-PROCESS werden automatisch auch alle erzeugten Kommandodateien abgebrochen. Das Kommando muss nicht notwendigerweise von derselben Konsole eingegeben werden, von dem aus das Kommando RUN gestartet wurde.
Es setzt allerdings den Berechtigungsschlüssel E voraus.

17.4.2 Aufbau von Kommandodateien

Eine Kommandodatei, die vom Operator an der Konsole gestartet wird, ist eine SAM- oder ISAM-Datei bzw. ein Bibliothekselement (Typ J) auf gemeinschaftlichen Datenträgern. Die Datensätze dürfen beliebige Operatorkommandos enthalten, unter bestimmten Voraussetzungen auch privilegierte und Benutzerkommandos. Die RUN-Kommandofolge darf kein SET-LOGON-PARAMETERS- oder EXIT-JOB-Kommando enthalten. Die Datensätze der Kommandodatei haben entweder feste oder variable Satzlänge, wobei die maximale Satzlänge allerdings 201 Byte beträgt. Es werden keine Fortsetzungszeilen unterstützt.
Die Systembetreuung kann die Kommandodatei durch die Vergabe eines Ausführungskennwortes gegen unberechtigte Ausführung durch den Operator schützen. Die Kommandodatei wird vom System nur dann verarbeitet, wenn der Operator im RUN-Kommando das passende Kennwort angibt. Das mögliche Kennwort wird in der CONSLOG-Datei nicht protokolliert.

RUN-Kommandofolgen können beliebig viele Kommandos aufnehmen. Die Systembetreuung sollte beim Aufbau der Kommandofolge nach spätestens 30 Kommandos das Kommando ASTOP einfügen, um einen strukturierten Ablauf der Kommandos zu gewährleisten. Eine Unterbrechung ist auch dann ratsam, wenn die Ausführung eines Kommandos die vollständige Abarbeitung des vorhergehenden bzw. den Aufruf abhängiger Funktionen bedingt. Stehen beispielsweise 10 Kommandos in der RUN-Kommandodatei, werden diese Kommandos hintereinander eingelesen und bearbeitet. Wird in der Kommandofolge das Kommando SHOW-PENDING-MSG ausgeführt, sollte die Prozedur nach diesem Kommando mit einem ASTOP-Kommando unterbrochen werden. Der Operator kann daraufhin evtl. ausstehende Fragen beantworten. Nach Ablauf der Wartezeit oder einem AGOGO wird die Kommandofolge weiter bearbeitet.

Ein Vorteil einer mit ASTOP-Kommandos gegliederten RUN-Datei ist, dass die Kommandofolge während der Ausführung nicht unnötig Klasse-4-Speicher belegt.
Das System liest eine so aufgebaute und umfangreiche Kommandodatei abschnittsweise in den Klasse-4-Speicher, d.h. von einem ASTOP-Kommando bis zum folgenden, und führt jeweils den eingelesenen Dateiabschnitt aus, bevor es einen neuen einliest.

Sind DSSM-Kommandos Bestandteile der Kommandodatei, sollte die Systembetreuung Folgendes beachten:
Eine eigene Task arbeitet die DSSM-Kommandos ab. Kommandos, die erst eingegeben werden dürfen, wenn das jeweilige Subsystem geladen ist, müssen warten, bis die DSSM-Kommandos abgearbeitet sind. Entsprechend ist auch an dieser Stelle das Kommando ASTOP in die Kommandodatei aufzunehmen.

Die Kommandodatei CMDFILE (Bibliothekselemente sind nicht zulässig) kann eine Folge von Operatorkommandos enthalten, die bei jedem Systemstart unverändert ablaufen. Sie enthält z.B. Aufrufe für die folgende Kommandofolge:

DCSOF	Kommandodatei (Start Option File) zur Inbetriebnahme des Datenkommunikationssystems (DCS)

Beispiel für eine CMDFILE

```
/MODIFY-JOB-CLASS CLASS-NAME=JCDIALOG,C-L=0                                  1.
/MODIFY-JOB-CLASS CLASS-NAME=TSOSDIA,C-L=0                                   2.
/MODIFY-JOB-CLASS CLASS-NAME=JCBATCH,C-L=40                                  3.
/SET-MSG-SUPPRESSION MSG-ID=(BLS0980,JMS0154),CONSOLE=C0                     4.
/DCSTART DCSOF=SOF.ACT.DCM8                                                  5.
/ASTOP                                                                       6.
/MOD-TASK-CAT CATEG-NAME=DIALOG,WEIGHT-CODE=400,MIN-ACT=35,MAX-ACT=50        7.
/MOD-TASK-CAT CATEG-NAME=BATCH,WEIGHT-CODE=10,MIN-ACT=5,MAX-ACT=20           8.
/MOD-TASK-CAT CATEG-NAME=TP,WEIGHT-CODE=500,MIN-ACT=20,MAX-ACT=40            9.
/IMPORT-PUBSET PUBSET=L                                                     10.
/ATTACH-DEVICE (X3,X4,X5,S1,S3,K7,X0,Y3)                                    11.
/SET-DISK-P V=(PRI009,PRI010,PRI037,PRI038,SESAMA),USER=SHARE,ASS=OPER      12.
/SET-DISK-P V=(WORK11,SLED),USER=SHARE,ASS=OPER                              13.
/START-SUBSYS SUBSYS-NAME=MSCF,SUBSYS-PARAM='FILE=$TSOS.RFA.MSCFSTART'      14.
/START-SUBSYS SUBSYS-NAME=SPOOL,SUBSYS-PARAM='RSO=STD'                      15.
/ASTOP                                                                      16.
/MODIFY-JOB-CLASS CLASS-NAME=JCDIALOG,CLASS-LIMIT=20                        17.
/MODIFY-JOB-CLASS CLASS-NAME=TSOSDIA,CLASS-LIMIT=4                          18.
```

1. Keine Jobs für die Jobklasse JCDIALOG zulassen.
2. Keine Jobs für die Jobklasse TSOSDIA zulassen.
3. Maximal 40 Jobs für die Jobklasse JCBATCH zulassen.
4. Meldungen mit den Meldungsnummern `BLS0980` oder `JMS0154` werden an der Konsole C0 unterdrückt.
5. Laden des Datenkommunikationssystems.
6. Kommandodatei anhalten, bis das Datenkommunikationssystem geladen ist.
7. Anzahl und Dringlichkeit von Tasks für die Kategorie DIALOG festlegen.
8. Anzahl und Dringlichkeit von Tasks für die Kategorie BATCH festlegen.
9. Anzahl und Dringlichkeit von Tasks für die Kategorie TP festlegen.
10. Pubset L importieren.
11. Geräte dem System zuschalten.
12. Vorgaben setzen für die Überwachung von Platten.
13. Vorgaben setzen für die Überwachung von Platten.
14. MSCF-Kommunikation aktivieren.
15. Laden des Subsystems SPOOL mit automatischem Laden von RSO.
16. Kommandodatei anhalten, bis SPOOL geladen ist.
17. Maximal 20 Jobs für die Jobklasse JCDIALOG zulassen.
18. Maximal 4 Jobs für die Jobklasse TSOSDIA zulassen.

Mit dem Systemparameter NBCONOPI=Y kann ein System beantragt werden, bei dem auch bei physikalischen Konsolen ein Logon nötig ist.
In einem solchen System ist also z.B. die Eingabe eines RUN-Kommandos an einer physikalischen Konsole nur nach einem erfolgreichen SET-LOGON-PARAMETERS und einem entsprechenden REQUEST-OPERATOR-ROLE möglich.

Wird bei Verwendung des Systemparameters NBCONOPI=Y der Dateiname der RUN-Datei ohne Benutzerkennung angegeben, wird die Datei zuerst unter der Kennung des Operators, dann unter der im Systemparameter NBRUNUID angegebenen Kennung und zuletzt unter der Benutzerkennung TSOS gesucht. Da das Privileg OPERATING das Privileg TSOS nicht mehr voraussetzt, muss die RUN-Datei ggf. „shareable“ gemacht werden.

Bei Verwendung des Systemparameters NBCONOPI=Y kann nicht jeder Operator die Kommandos AGOGO oder CANCEL-RUN-PROCESS für jede RUN-Sequenz eingeben, auch wenn er über die notwendigen Berechtigungsschlüssel verfügt.
Die beiden Kommandos sind nur dann zulässig, wenn mindestens eine der folgenden Bedingungen erfüllt ist:

- die Benutzerkennung des Aufrufers stimmt mit der Kennung überein, unter der das RUN-Kommando abgesetzt wurde
- die Benutzerkennung des Aufrufers ist TSOS
- das RUN-Kommando wurde von einem berechtigten Benutzerprogramm mit generiertem Berechtigungsnamen eingegeben
- der Aufrufer des RUN-Kommandos hat sich vom System abgemeldet

Beim Hochfahren des Systems wird die CMDFILE abgearbeitet und dabei so behandelt wie ein abgesetztes RUN-Kommando auf diese Datei von der Hauptkonsole. Hier ist vorher kein SET-LOGON-PARAMETERS- und REQUEST-OPERATOR-ROLE-Kommando nötig, denn für diese CMDFILE und nur bei Startup wird die Berechtigungsprüfung anhand von Berechtigungsschlüsseln so durchgeführt, als wären der Hauptkonsole alle Berechtigungsschlüssel zugewiesen. Alle aus der CMDFILE aufgerufenen RUN-Kommandos genießen ebenfalls dieses Privileg.

Das Kommando RUN und die mit ihm logisch verbundenen Kommandos (z.B. AGOGO und CANCEL-RUN-PROCESS) sind einer Benutzertask auch mit OPERATING-Privileg nicht erlaubt. Hier sind SDF-P-Prozeduren anwendbar, in denen auch Kommandos des Privilegs OPERATING benutzt werden können.

17.5 Ausübung von Systemverwaltungsfunktionen durch den Operator

Dem Operator an Konsolen werden über das Kommando SHOW-CMD-ATTRIBUTES alle zu diesem Zeitpunkt bekannten Kommandos genannt. Kommandos, die den Schlüssel $ haben, sind jedoch verboten (Eingabe wird abgewiesen).

In der Menge der $-Kommandos sind auch Kommandos enthalten, die keinerlei Systembetreuungsfunktion anbieten. Sollen einzelne dieser Kommandos an Konsolen erlaubt werden, müssen sie bereits zum Zeitpunkt der Systemeinleitung mit Hilfe des Parameterservice (Parametersatz OPR, siehe "Konfiguration und Meldungsunterdrückung an Konsolen (OPR)") einem anderen Berechtigungsschlüssel zugeordnet werden.

Dazu ist für jedes Kommando eine SET-CMD-CODE-Anweisung erforderlich.
Der Berechtigungsschlüssel ist von der Systembetreuung festzulegen.

Beispiel

Das Kommando mit der Bezeichnung SHOW-FILE-ATTRIBUTES soll dem Aufgabengebiet „allgemeine Aufgaben“ (Berechtigungsschlüssel E) zugeordnet werden.
Die Bedienplätze K1 und K2 sollen „allgemeine Aufgaben und Befugnisse“ ausführen dürfen.

```
...
SET-CMD-CODE AUTHORIZATION-CODE=E,CMD-NAME=SHOW-FILE-ATTRIBUTES
SET-CODE CODE=E,CONSOLE=(K1,K2)
...
```

Folgende Kommandos werden vom Hersteller standardmäßig mit dem Berechtigungsschlüssel $ ausgeliefert:

```
ADD-FILE-LINK
ADD-MASTER-CATALOG-ENTRY
ADD-PASSWORD
ADD-USER

COPY-FILE
CREATE-FILE
CREATE-FILE-GENERATION
CREATE-FILE-GROUP
CREATE-JV
CREATE-TAPE-SET

DELETE-FILE
DELETE-FILE-GENERATION
DELETE-FILE-GROUP
DELETE-JV
DELETE-SYSTEM-FILE
DELETE-TAPE-SET

EXPORT-FILE
EXTEND-TAPE-SET

IMPORT-FILE

LOCK-USER

MODIFY-FILE-ATTRIBUTES
MODIFY-FILE-GENERATION-SUPPORT
MODIFY-FILE-GROUP-ATTRIBUTES
MODIFY-JV-ATTRIBUTES
MODIFY-MASTER-CATALOG-ENTRY
MODIFY-SPACE-SATURATION-LEVELS
MODIFY-USER-ATTRIBUTES
MODIFY-USER-PUBSET-ATTRIBUTES

PRINT-DOCUMENT

REMOVE-JV-LINK
REMOVE-MASTER-CATALOG-ENTRY
REMOVE-PASSWORD
REMOVE-USER

SET-JV-LINK
SHOW-FILE-TRANSFER
SHOW-JV-ATTRIBUTES
SHOW-JV-LINK
SHOW-SPOOL-CHARACTER-SETS
SHOW-SPOOL-DEVICES

SHOW-SPOOL-FORMS
SHOW-SPOOL-PARAMETERS
SHOW-USER-ATTRIBUTES

UNLOCK-USER

WRITE-SPOOL-TAPE
```

Dagegen sind die Kommandos SHOW-FILE-ATTRIBUTES und SHOW-SDF-PARAMETERS (voreingestellter Berechtigungsschlüssel @) sowie MODIFY-SDF-PARAMETERS (voreingestellter Berechtigungsschlüssel *) für den Operator standardmäßig verfügbar. Sie werden benötigt, um im Notfall von der Konsole aus die Syntaxdatei-Konfiguration wieder so weit herzustellen, dass neue Anmeldungen im Dialog möglich sind.

Wenn dies nicht gewünscht wird, dann können sie durch Zuweisen des Berechtigungsschlüssels $ verboten werden.

18 Systemzeit-Verwaltung

Auf einem SE Server gibt es unterschiedliche Uhren und damit Zeiten:

- Die BS2000-Systemzeit, basierend auf dem TODR (Time of Day Register).
  Auf SU /390 ist das TODR eine eigenständige Uhr.
  Auf SU x86 wird das TODR durch X2000 emuliert.
- Die Zeit des Serviceprozessors (SVP-Zeit), die von BS2000 durch den Befehl „Store Real Clock“ ermittelt werden kann.
  Auf SU /390 ist die SVP-Uhr eine eigenständige Uhr, die von der Management Unit auf ihre eigene Zeit synchronisiert wird.
  Auf SU x86 wird sie von X2000 emuliert und mit der Zeit des Trägersystems versorgt.
- Die Zeit der Management Unit.
  Die Management Unit kann mit externen Zeitgebern synchronisiert sein.
- Die Zeit des Trägersystems an SU x86, die über NTP mit der Zeit der Management Unit synchronisiert wird.
  Sie wird von X2000 als SVP-Zeit emuliert.

Darüber hinaus kann BS2000 in einen Rechnerverbund integriert sein, in dem die Zeit der Mitglieder synchronisiert wird.

Das vorliegende Kapitel erläutert diese Zusammenhänge. Es gibt einen Überblick über die Verwaltung der Systemzeit in BS2000 und beinhaltet die Initialisierung und die Synchronisation der Systemzeit sowie die Sommer-/Winterzeit-Umstellung und Spezialfälle.

Zur Zeitbestimmung bei der Systemeinleitung siehe "Zeitbestimmung bei der Systemeinleitung".

18.1 Systemzeit

Als Systemzeit für eine BS2000-Session dient die lokale gesetzliche Zeit. Alle Zeitstempel im privilegierten Zustand (TPR/SIH/MER) basieren auf dieser Zeit. Sie wird über die Zeit-Informationsdienste (GTIME, GDATE, GETOD) an Benutzerprogramme (TU) geliefert.

Einige Datumsangaben (insbesondere die Zeitstempel in den Dateikatalogen) erfolgen in UTC-Zeit. Einige HW-nahe, performance-kritische Komponenten nutzen den mit STCK (Store Clock) ermittelten TODR-Wert.

Die aktuelle Systemzeit wird folgendermaßen berechnet:

- Ermittlung des aktuellen STCK-Wertes
- Subtraktion des TODR-Korrekturwertes, resultierend aus Zeitkorrekturen von abgeschlossenen Synchronisationsaufträgen und dem initialen IPL-Korrekturwert
- Berücksichtigung eines aktuell aktiven Synchronisationsauftrages

Ein durch STCK ermittelter Zeitwert stimmt also nicht mit demjenigen der offiziellen Informationsdienste überein und gibt somit nicht die aktuelle Zeit wieder.

Mit dem Kommando SHOW-SYSTEM-INFORMATION INFORMATION=*SYSTEM-TIME-PARAMETER können Informationen über die Parameter der Zeiteinstellung eingeholt werden.

18.1.1 TODR als HW-Uhr

Das Time-of-Day-Register (TODR) ist ein 64 Bit breites Register mit folgenden Eigenschaften:

- Es wird im Bereich [Bit 63, Bit 12] pro Mikrosekunde um den Wert 1 erhöht und realisiert damit eine Uhr.
- Es kann durch den privilegierten Befehl Set Clock (SCK) geändert werden.
- Es kann durch den nichtprivilegierten Befehl Store Clock (STCK) gelesen werden.

Durch die Breite der Uhr im TODR ist auch die Größe der darstellbaren Zeitspanne begrenzt. BS2000 erweitert die Zeitdarstellung im TODR kompatibel durch so die genannte Epoche, siehe Abschnitt „TODR-Epoche".

Während des ersten BS2000-IPL wird das TODR initialisiert. Mit der Standard-Epoche 00 ist dies gleichbedeutend mit dem Wert „Anzahl Mikrosekunden seit 1.1.1900 00:00:00“. Für andere Epocheneinstellungen muss das TODR gemäß der Epoche interpretiert werden, um einen korrekten Zeitwert zu erhalten, siehe Abschnitt „TODR-Epoche".

Auf Server-Units SU /390 wird diese Initialisierung bei jedem weiteren IPL wiederholt.
Auf Server-Units SU x86 findet diese Initialisierung nur einmalig statt. Sollte auf SU x86 eine erneute Initialisierung des TODR notwendig sein, muss dazu im Native-Betrieb die Server Unit neu gestartet werden, während unter VM2000 die VM neu erzeugt werden muss (z.B. mit /DELETE-VM und /CREATE-VM).

Virtuelles TODR auf SU /390 unter VM2000

Die VM2000-Firmware erlaubt, für jede VM ein eigenes virtuelles TODR zu führen.
Das HW-TODR der SU /390 steht nur dem Hypervisor zur Verfügung. Alle TODRs takten gleich. Zum HW-Kontext einer VM gehört ein VM-spezifischer Korrekturwert des virtuellen TODR gegenüber dem HW-TODR.

Der Hypervisor initialisiert bei SU /390 das HW-TODR mit dem Wert
„Monitorzeit – 24 Stunden“.

Emuliertes TODR auf SU x86

Auf SU x86 werden Set Clock und Store Clock von der Firmware emuliert. Das emulierte TODR taktet unabhängig von der Zeit des Trägersystems.
Alle VM2000-Gastsysteme haben ein eigenes virtuelles TODR.

18.1.2 TODR-Epoche

BS2000 erweitert das TODR um eine TODR-Epoche. Damit kann eine Systemzeit dargestellt werden, die nicht in dem bisherigen Standardbereich (von 01.01.1900 00:00:00.000000 bis 17.09.2042 23:53:47.370495) liegt, welcher der Standard-Epoche 00 entspricht.

Die TODR-Epoche für den aktuellen Systemlauf wird im Startup-Parameterservice, Parametersatz GTIME, Parameter EPOCH eingestellt, siehe Abschnitt „Steuerung der Systemzeit (GTIME)" . Sie kann während des Systemlaufs nicht verändert werden.

Die Epocheneinstellung wird in den Benutzer-Makros CTIME (Rechnen mit Zeitstempeln) und GTIME (Datum und Uhrzeit anfordern) automatisch berücksichtigt, siehe Handbuch „Makroaufrufe an den Ablaufteil [30]. Der Wert der aktuellen Epocheneinstellung kann mit dem Benutzermakro GTIME und mit Kommando SHOW-SYSTEM-INFORMATION ermittelt werden.

> **!** Damit die Funktionen GTIME und CTIME das TODR auch nach dessen Überlauf im Jahr 2042 korrekt interpretieren, wird rechtzeitig vor diesem Zeitpunkt die Standard-Epoche von 00 auf 08 geändert werden. Damit wird die Darstellung eines Datumsbereiches von 1971 bis 2114 ermöglicht (siehe Tabelle unten).
>
> Für Applikationen, welche Zeitstempel im TODR-Format benutzen, kann das zu Inkompatibilitäten führen, und sie müssen angepasst werden, bevor die Epoche geändert wird. Deswegen wird dringend empfohlen, in Applikationen auf das Speichern von Zeitstempeln im TODR-Format zu verzichten, um problemlose Wechsel der TODR-Epoche zu ermöglichen.

Interpretation von TODR-Zeistempeln mit den TODR-Epochen 00 und 08

TODR-Zeitstempel (hex)	Interpretation als Datum und Uhrzeit	
	EPOCH=00	EPOCH=08
00000000 00000000	01.01.1900 00:00:00,000000	17.09.2042 23:53:47,370496
7FFFFFFF FFFFF000	11.05.1971 11:56:53,685247	26.01.2114 11:50:41,055743
80000000 00000000	11.05.1971 11:56:53,685248	11.05.1971 11:56:53,685248
FFFFFFFF FFFFF000	17.09.2042 23:53:47,370495	17.09.2042 23:53:47,370495

Tabelle 49: Vergleich der derzeitigen Standard-Epoche 00 mit der zukünftigen Standard-Epoche 08

Zeitstempelformat TODX

Als Alternative zum TODR-Format gibt es im BS2000 das Zeitstempelformat TODX. TODX besteht aus einem Doppelwort und enthält die Anzahl der Mikrosekunden seit 01.01.1900 00:00:00,000000 Uhr. Anders als das TODR-Format kann das TODX-Format eine viel größere Zeitspanne abdecken (vom Jahr 1900 bis zum Jahr 4317).

Das Format TODX wird in den Benutzer-Makros CTIME (Rechnen mit Zeitstempeln) und GTIME (Datum und Uhrzeit anfordern) verwendet bzw. geliefert, siehe Handbuch "Makroaufrufe an den Ablaufteil" [30]. Es ist vorteilhaft bei der Epochen-unabhängigen Darstellung eines Zeitstempels und beim Rechnen mit Zeitstempeln aus unterschiedlichen Epochen.

Innerhalb der derzeitigen Standard-Epoche 00 entspricht ein TODX-Zeitstempel einem TODR-Zeitstempel, dessen Inhalt um 12 Bit nach rechts geschoben ist.

Programme, die selbst mit dem TODR rechnen, können den Algorithmus verwenden, der im Abschnitt "Umwandlung von TODR-Zeitstempeln in TODX-Zeitstempeln" beschriebenen wird.

Umwandlung von TODR-Zeitstempeln in TODX-Zeitstempel

Die folgenden Algorithmen verwenden einen TODR-Wert und den zugehörigen Epochen-Wert zur Berechnung der Anzahl der Mikrosekunden seit 01.01.1900 00:00:00,000000, also den TODX-Wert.

Verwendete Variablen und Operatoren:

```
TODX    :  TODX-Wert, d.h. Anzahl der Mikrosesekunden seit 01.01.1900 00:00:00,000000
TODR    :  TODR-Wert
EPD     :  Wert der Epocheneinstellung (1 Byte)
EPC     :  Epoch-Counter (erste Sedezimalziffer des EPD)
EPO     :  Epoch-Offset (zweite Sedezimalziffer des EPD)
x >> n  :  Verschiebe den Wert x um n Bits nach rechtes
x & y   :  UND-Operation (bitweise)
```

Algorithmus für die derzeitige Standard-Epoche 00:

```
TODX := TODR >> 12;
```

Algorithmus für die zukünftige Standard-Epoche 08:

```
TODX := TODR >> 12;
if ((TODX >> 48) < 8)
  TODX := TODX + x'0010000000000000';
```

Algorithmus für eine beliebige Epocheneinstellung:

```
TODX := TODR >> 12;
EPC  := EPD >> 4;
EPO  := EPD & x'0F';
if ((TODX >> 48) < EPO)
  EPC := EPC + 1;
TODX := TODX + EPC * x'0010000000000000';
```

Wertebereiche weiterer TODR-Epochen

EPOCH	Datum und Uhrzeit	
	von	bis
00	01.01.1900 00:00:00,000000	17.09.2042 23:53:47,370495
01	02.12.1908 19:29:36,710656	19.08.2051 19:23:24,081151
02	03.11.1917 14:59:13,421312	20.07.2060 14:53:00,791807
03	05.10.1926 10:28:50,131968	21.06.2069 10:22:37,502463
04	06.09.1935 05:58:26,842624	23.05.2078 05:52:14,213119
05	07.08.1944 01:28:03,553280	24.04.2087 01:21:50,923775
06	08.07.1953 20:57:40,263936	24.03.2096 20:51:27,634431
07	09.06.1962 16:27:16,974592	24.02.2105 16:21:04,345087
08	11.05.1971 11:56:53,685248	26.01.2114 11:50:41,055743
09	11.04.1980 07:26:30,395904	28.12.2122 07:20:17,766399
0A	13.03.1989 02:56:07,106560	29.11.2131 02:49:54,477055
0B	11.02.1998 22.25:43,817216	29.10.2140 22:19:31,187711
0C	13.01.2007 17:55:20,527872	30.09.2149 17:49:07,898367
0D	15.12.2015 13:24:57,238528	01.09.2158 13:18:44,609023
0E	15.11.2024 08:54:33,949184	03.08.2167 08:48:21,319679
0F	17.10.2033 04:24:10,659840	04.07.2176 04:17:58,030335

18.1.3 TODR-Korrekturwerte

TODR-Korrekturwerte des Zeitmanagements haben folgende Ziele:

1. Die Systemzeit kann ohne Änderung des TODR an externe Zeitgeber bzw. an eine Verbundzeit angepasst werden; damit ist ein Rückstellen der Systemzeit möglich.
2. Bei Sommer-/Winterzeitumstellung ermöglicht dies einen Sprung der Systemzeit ohne Beeinflussung des TODR.

Insbesondere ein Rückstellen des TODR verbietet sich im laufenden System, da viele Algorithmen im Betriebssystem wie in der Benutzersoftware die Monotonie des TODR unterstellen.

Nach jedem Abschluss eines Synchronisationsauftrages der Systemzeit wird der aktuell gültige TODR-Korrekturwert um diesen Auftragswert bei einem Rückstellen erhöht bzw. bei einem Vorstellen erniedrigt. Angemeldete Realzeit-Timer (STXIT bzw. systemintern) werden entsprechend angepasst. Bei der Bereitstellung der aktuellen Systemzeit im Rahmen von GTIME, GDATE oder GETOD wird der aktuelle TODR-Korrekturwert berücksichtigt.

18.1.4 Synchronisation der Systemzeit mit externen Zeitgebern oder im Verbund

BS2000 bietet eine privilegierte sowie eine POSIX-Schnittstelle „ADJUST-TIME“ zur Anpassung der Systemzeit (nicht des HW-TODR!) an externe Uhren. Diese Schnittstelle wird genutzt für:

- Synchronisation in einem NTP-Verbund
- Synchronisation mit einem XCS-Verbund
- Synchronisation mit der SVP-Zeit (SU /390)
- Synchronisation mit der Trägersystem-Zeit (SU x86)

Ein solcher Synchronisationsauftrag führt für einen Zeitraum, der viermal so lang ist wie der geforderte Anpassungswert, zu einem langsameren bzw. schnelleren Takten der Systemzeit. Hierbei wird die Nebenbedingung eingehalten, dass zwei aufeinander folgende GTIME-Aufrufe einer Task weiterhin monoton steigende Zeitwerte liefern.

Während der Anpassungsphase wird die zu berechnende Systemzeit um den bereits durchgeführten Synchronisationsanteil zusätzlich korrigiert:

$(-t - t_a) / (t_e - t_a) * d$

Dabei bedeutet:

t	aktuelle Zeit
t_a	Zeitpunkt des Synchronisationsbeginns
t_e	Zeitpunkt des Abschlusses der Synchronisation
d	Anpassungswert

Erst mit Abschluss der Synchronisation wird der TODR-Korrekturwert um den gewünschten Anpassungswert korrigiert.

Zwischen den einzelnen Instanzen, die Synchronisationsaufträge stellen dürfen, ist eine Prioritätenreihenfolge entsprechend folgender Auflistung festgelegt. Sind mehrere Instanzen aktiv, werden nur Aufträge derjenigen mit höchster Priorität ausgeführt.

Die Prioritäten sind in folgender absteigenden Reihenfolge festgelegt
(1. = höchste Priorität, 2. = zweithöchste Priorität, usw.):

1. BS2000 ist Teilnehmer eines NTP-Verbundes, der mit einem mit einem "vertrauenswürdigen" externen Zeitgeber konfiguriert ist (z.B. einem NTP-Server, dessen stratum-Wert nicht größer als 4 ist) konfiguriert ist.
2. Die Management Unit, die die SVP-/Trägersystem-Zeit der Server Units synchronisiert, ist mit einem "vertrauenswürdigen" externen Zeitgeber verbunden (stratum-Wert des NTP-Servers ist nicht größer als 4).
3. BS2000 ist Teilnehmer eines XCS-Verbundes.
4. BS2000 ist Teilnehmer eines NTP-Verbundes, der ohne externen Zeitgeber konfiguriert ist.

5. Die Management Unit, die die SVP-/Trägersystem-Zeit der Server Units synchronisiert, ist entweder mit keinem externen Zeitgeber verbunden, oder sie ist mit einem "nicht vertrauenswürdigen" Zeitgeber verbunden (stratum-Wert des NTP-Servers ist größer als 4).

ℹ Bei der Zeitsynchronisation auf Basis der SVP-Uhr (Punkte 2 und 5) ist zu beachten, dass die SVP-Uhr nur eine Auflösung im Sekundenbereich hat. Dadurch kann es zwischen SVP-Uhr und BS2000-Zeit sowie an SU /390 zusätzlich auch zwischen SVP-Uhr und der Zeit der Management Unit jeweils zu Abweichungen von ± 1 Sekunde kommen. In Summe ergibt sich damit für die BS2000-Systemzeit an SU /390 eine maximale Abweichung von ± 2 Sekunden zur Zeitquelle, an SU x86 von ± 1 Sekunde.
In einem NTP- oder XCS-Verbund (Punkte 1, 3 und 4) erfolgt die Übertragung und Synchronisation der Zeit mit wesentlich höherer Auflösung. Die Abweichung der BS2000-Systemzeit von der maßgebenden Zeitquelle liegt im Millisekunden-Bereich.

18.1.5 Subsystem GET-TIME

BS2000 bietet mehrere Schnittstellen zur Ermittlung der aktuellen Systemzeit: GTIME, GDATE, GETOD. Laufzeitroutinen von Hochsprachen mit analoger Semantik setzen entsprechend darauf auf.

GTIME bietet die Systemzeit (lokale Zeit sowie UTC-Zeit) im Binär-, ISO4- und TODR/TODX-Format an. Es ist zu beachten, dass sich die Systemzeit im TODR-Format wegen der TODR-Korrekturwerte von einem gleichzeitig ermittelten STCK-Wert unterscheidet.

Der Server für die langfristig garantierte Schnittstelle GTIME ist sowohl als Bestandteil von BS2000 OSD/BC als auch als nichtprivilegiertes TU-Subsystem GET-TIME realisiert, um bei Aufrufen aus TU eine SVC-freie, performante Bearbeitung zu bieten. Der GTIME-Aufruf aus TU hat für die Bereitstellung der Systemzeit im TODR-Format eine Pfadlänge von weniger als 200 Befehlen. Der Einsatz von TASKDATE (siehe "TASKDATE: Testen in simulierter Zeit") bzw. die Ermittlung der XCS-Zeit (siehe "Synchronisation in einem XCS-Verbund") führt zu einer Bearbeitung im TPR. Das Subsystem enthält weiterhin den Server für CTIME, der Makroschnittstelle zur Arithmetik mit Zeitstempeln und zur Konvertierung von Zeitstempeln bezüglich Format und Zeitbasis. Die Datenbasis für die beiden Dienste (TODR-Korrekturwerte, Daten eines aktiven Synchronisationsvorganges sowie Sommer-/ Winterzeit-Umstellzeitpunkte) sind sowohl für den Betriebssystem-Modul im Systemadressraum als auch TU-schreibgeschützt im Subsystem abgelegt.

Das Subsystem GET-TIME ist obligatorisch, wird vor „System Ready“ automatisch gestartet und besitzt zwei, über die explizite Auswahl eines der zwei ausgelieferten Subsystemkataloge gesteuerte Ausprägungen:

- Laden unterhalb 16 MB (Standard), falls in dem System Nicht-XS-fähige Programme im Einsatz sind
- Laden oberhalb 16 MB (zur Entlastung des unteren Adressraumes), falls nur XS-fähige Programme das Subsystem nutzen

Neben dem Subsystem GET-TIME gibt es auf SU x86 noch das nur Betriebssystem-intern genutzte Subsystem GET-TIMX.

GET-TIME Startup-Parameter

Die Startup-Parameterdatei enthält einen Abschnitt für GET-TIME (Parametersatz GTIME).
Dieser beschreibt für eine oder mehrere Zeitzonen folgende Information:

- Information über die Zeitzone (d.h. Differenz zur UTC-Zeit)
- Sommerzeit-Information (Differenz Sommerzeit / Winterzeit, vergangene und zukünftige Umstellzeitpunkte)

Diese Information wird benötigt für

- Zeitkonvertierungen Lokale-Zeit / UTC (und umgekehrt) für beliebige Zeitstempel durch die Funktion CTIME. Je mehr Umstellzeitpunkte verfügbar sind, um so größer ist der Zeitbereich, in dem CTIME eine korrekte Umrechnung Lokale-Zeit / UTC vornehmen kann.
- Bestimmung von Sommerzeit / Winterzeit bei Startup
- Umstellung der Systemzeit zum nächsten Termin einer Sommerzeit- / Winterzeit-Umstellung

Bei SU /390 ist die Zeitzone des Standortes des Servers über SVP-Frame einzustellen. Bei SU x86 ist diese Information im Trägersystem bekannt. Startup ermittelt sich dieses Datum und wählt damit vorrangig den korrespondierenden Zeitzonen-Parameterabschnitt aus.

Für weitere Hinweise zum Startup-Parameter GTIME, insbesondere zum Umfang der bereitgestellten Umstellungszeitpunkte, siehe Abschnitt „Steuerung der Systemzeit (GTIME)".

18.2 SVP-Zeit

Der SVP hat aus BS2000-Sicht eine eigene Uhr, die über die privilegierte Schnittstelle „Store Real Clock“ vom Betriebssystem abgefragt werden kann. Die SVP-Uhr muss immer in lokaler Zeit laufen. „Store Real Clock“ stellt die SVP-Zeit mit Sekunden-Auflösung bereit. Die „Store Real Clock“-Zusatzinformation ist abhängig von der Architektur der Server Unit:

- Zeitzone
- Anzeige: mit externem Zeitgeber synchronisiert

Server Unit	Zusatzinformation
SU /390	Zeitzone, Anzeige: • Synchronisation durch die MU aktiv / ausgefallen • MU mit externem Zeitgeber synchronisiert
SU x86	Zeitzone, Anzeige: • MU mit externem Zeitgeber synchronisiert

Je nach Architektur der Server Unit ist diese Uhr als Hardware vorhanden oder durch X2000 emuliert, sie taktet jedoch in beiden Fällen unabhängig vom TODR.

Server Unit	Art der Einbettung der SVP-Uhr
SU /390	Eigenständige SVP-Uhr vorhanden, von MU synchronisiert. Die MU kann mit einem externem Zeitgeber oder mit anderen Systemen mittels NTP synchronisiert sein.
SU x86	SVP-Uhr wird von X2000 mit Hilfe der Zeit des Trägersystems emuliert. Die Zeit des Trägersystems wird von der Management Unit synchronisiert.

Hinweise zum Einsatz unter VM2000

Auf SU /390 wird unter VM2000 der „Store Real Clock“ eines Gastsystems vom Hypervisor emuliert. Im Normalfall wird hierzu die Systemzeit des Monitorsystems herangezogen. Es gibt folgende Ausnahmen, die zur Ausführung eines „Store Real Clock“-Befehls führen:

- Das Monitorsystem wurde mit einer explizit vom Operator eingegebenen Zeit gestartet.
- Das Monitorsystem nimmt an einem XCS-Verbund teil.
- Das Monitorsystem nimmt an einem NTP-Verbund teil, der nicht mit einem externen Zeitgeber konfiguriert ist.
- Das Monitorsystem ist zum Zeitpunkt der Befehlsausführung nicht erreichbar (z.B. bei Restart im Monitorsystem).

Auf SU x86 wird der „Store Real Clock“ auch unter VM2000 wie native von X2000 emuliert.

18.3 Initialisierung der Systemzeit

Die Systemzeit wird während IPL initialisiert, indem das TODR geladen wird.

Zeitquellen für Systemzeit-Initialisierung

Als Zeitquellen für die Initialisierung wird die über die Schnittstelle „Store Real Clock“ ermittelte SVP-Zeit übernommen.

Weiter gibt es folgenden Sonderfall:
Das System wird im DIALOG-Modus mit Option UNLOCK hochgefahren und der Operator gibt eine explizite Zeitangabe ein.

Die verwendete Zeitquelle wird als Insert der Startup-Meldung `NSI1163` protokolliert.

Die Initialisierung der Sommerzeit-/Winterzeit-Einstellung erfolgt mittels der SVP-Information. Liegt dort die Information nicht vor oder wurde eine manuelle Zeitangabe (mittels UNLOCK-Option) gemacht, kann eine doppelte Stunde bei Sommer-/Winterzeitwechsel nicht eindeutig ermittelt werden. Startup unterstellt dann Winterzeit.

Zur Initialisierung der Systemzeit siehe auch Abschnitt „Zeitbestimmung bei der Systemeinleitung".

18.4 Synchronisation der Systemzeit

- Synchronisation mit der SVP-Uhr auf SU /390
- Synchronisation mit der Zeit des Trägersystems auf SU x86
- Synchronisation in einem NTP-Verbund
- Synchronisation in einem XCS-Verbund
- Generelle Empfehlung für den XCS-Verbund
- Synchronisationsanzeigen bei /SHOW-SYSTEM-INFORMATION

18.4.1 Synchronisation mit der SVP-Uhr auf SU /390

Die Management Unit synchronisiert die SVP-Uhr auf die eigene Zeit.

Wenn keine andere Synchronisations-Instanz in BS2000 vorhanden ist, dann findet auf SU /390 eine Synchronisation der Systemzeit mit der SVP-Uhr und damit implizit mit der Zeit der Management Unit statt. Dazu wird periodisch in BS2000 ein Zeitvergleich SVP-Zeit gegen BS2000-Systemzeit vorgenommen und eine entsprechende Anpassung der BS2000-Systemzeit eingeleitet.

> In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
> `SYNCHRONIZATION=SKP-X`.

Die MU kann mit einem externen Zeitgeber verbunden sein. Die MU zeigt dem BS2000-System die Existenz eines externen Zeitgebers an, wenn die NTP-Zeit ein *stratum* 4 hat.

> In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
> `SYNCHRONIZATION=SERVER-CONN-EXT-REF`.

Zusätzlich kann in BS2000 ein NTP-Verbund mit externem Zeitgeber konfiguriert werden, der dann die höchstwertige Synchronisationsinstanz darstellt (siehe Abschnitt „Synchronisation in einem NTP-Verbund").

18.4.2 Synchronisation mit der Zeit des Trägersystems auf SU x86

Wenn keine andere Synchronisations-Instanz in BS2000 vorhanden ist, dann findet auf SU x86 automatisch eine Synchronisation der Systemzeit mit der SVP-Uhr, also mit der Zeit des Trägersystems statt. Das Trägersystem erhält über NTP die Zeit der Management Unit.

- In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
  `SYNCHRONIZATION=X2000`.

Die MU kann mit einem externen Zeitgeber verbunden sein. Diese externen Zeitgeber werden in der BS2000-Zeit nur dann sichtbar, wenn ihre Qualität bis zum Trägersystem erhalten bleibt (d.h.: *stratum* des NTP-Servers im Trägersystem 4).

- In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
  `SYNCHRONIZATION=SERVER-CONN-EXT-REF.`

18.4.3 Synchronisation in einem NTP-Verbund

Das Software-System NTP (Network Time Protocol) organisiert für die an einem derartigen Verbund beteiligten Systeme eine gemeinsame Zeit. BS2000 kann über POSIX Teilnehmer eines solchen Verbundes sein.
Jeder Verbundteilnehmer fragt die Zeit ausgewählter Systeme unter Berücksichtigung von Übertragungslaufzeiten ab und ermittelt daraus eine Verbundzeit, auf welche er seine Systemzeit synchronisiert. NTP nutzt dazu die POSIX-Schnittstellenausprägung von „ADJUST-TIME“. Server mit externen Zeitgebern können als so genannte „Time-Server“ deklariert werden, die ihre Systemzeit nicht an eine Verbundzeit anpassen, sondern umgekehrt: deren Zeit dient für die anderen Verbundteilnehmer als ausschließliche Quelle zur Ermittlung der Verbundzeit.

In BS2000 können folgende zwei NTP-Ausprägungen bezüglich der Priorisierung als Synchronisationsinstanz der Systemzeit zum Einsatz kommen. Die Unterscheidung erfolgt anhand der NTP-Zeitqualität *stratum*.

- STD-Einsatz
  Wenn die NTP-Zeit ein *stratum* > 4 hat, dann nimmt die BS2000-Zeitverwaltung an, dass der NTP-Server nicht über eine zuverlässige Zeitquelle verfügt, und weist ihm eine entsprechend niedrige Priorität zu. Wenn eine Zeitquelle (Zeitgeber an Management Unit oder XCS-Verbund, siehe Abschnitt „Synchronisation der Systemzeit mit externen Zeitgebern oder im Verbund") mit höherer Priorität existiert, dann lässt die BS2000-Zeitverwaltung keine Synchronisationsaufträge von NTP zu. Nur wenn keine Zeitquelle mit höherer Priorität existiert, dann wird die BS2000-Systemzeit mit der NTP-Zeit synchronisiert.

  > In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
  `SYNCHRONIZATION=DCE/NTP`.

- NTP-Verbund mit Anschluss an einen externen Zeitgeber
  Wenn die NTP-Zeit ein *stratum* <= 4 hat, dann nimmt die BS2000-Zeitverwaltung an, dass der NTP-Server über eine zuverlässige Zeitquelle verfügt, und weist ihm höchste Priorität zu. Die BS2000-Systemzeit wird mit der NTP-Zeit synchronisiert. Die Genauigkeit der NTP-Zeit liegt im Millisekunden-Bereich.

  > In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
  `SYNCHRONIZATION=BS2-CONN-EXT-REF`.

Eine Sommer/Winterzeit-Umstellung findet zu der durch NTP gesteuerten Systemzeit statt. NTP kann über eine solche Zeitumstellung hinweg im Einsatz bleiben.

Hinweise zum Einsatz unter VM2000

Ein VM2000-Gastsystem (auch das VM2000-Monitorsystem) kann unabhängig von den restlichen virtuellen Maschinen eines unter VM2000 betriebenen Servers an einem NTP-Verbund beteiligt sein. Ist das Monitorsystem Teilnehmer eines derartigen Verbundes, hängt bei SU /390 die Emulation der an die Gastsysteme gelieferten SVP-Zeit („Store-Real-Clock“) von der NTP-Ausprägung ab:

- Bei NTP mit Anschluss an einen externen Zeitgeber wird den Gastsystemen die Monitorzeit bereitgestellt. Die Teilnahme der Monitor-VM an einem derartigen Verbund garantiert für alle Gastsysteme ebenfalls die Verbundzeit, allerdings mit einer Genauigkeit von ± 1 Sekunde. Wird eine gleich hohe Genauigkeit gefordert, muss das Gastsystem als expliziter Verbundteilnehmer agieren.
- Bei NTP-Standardkonfiguration im Monitorsystem erhalten die Gastsysteme die echte SVP-Zeit. Demzufolge müssen bei Bedarf alle Gastsysteme Verbundteilnehmer sein.

Auf SU x86 nimmt die Dom0 mit Linux/X2000 am NTP-Verbund mit Anschluss an die MU teil. Dies garantiert die NTP-Verbundzeit für alle Gastsysteme, auch für das Monitorsystem, allerdings mit einer Genauigkeit von ± 1 Sekunde. Wenn eine höhere Genauigkeit gefordert wird, dann muss das System als expliziter Verbundteilnehmer agieren.

18.4.4 Synchronisation in einem XCS-Verbund

Ein XCS-Verbund ist eine Ausprägung der in BS2000 verfügbaren HIPLEX-MSCF-Funktionalität. Er ermöglicht insbesondere den Ablauf verteilter Anwendungen und umfasst folgende Funktionalität:

- automatische Ausfallerkennung und Rekonfiguration
- Lock-Management für verteilte Anwendungen (DLM)
- einheitliche Zeit im Verbund

Die Synchronisierung der Systemzeit im Verbund ist durch das Subsystem XCS-TIME, das beim XCS-Start initiiert wird, realisiert und basiert auf dem Konzept der XCS-Zeit, einer zweiten Zeit neben der Systemzeit. Die XCS-Zeit wird über GTIME (allerdings mit SVC) im TODR-Format und UTC-basiert bereitgestellt.

XCS-Zeit

Verteilte Anwendungen benötigen oft neben einer Lock-Funktion eine verbundweit monotone Zeit, z.B. um lokal erstellte Logging-Dateien später chronologisch mischen zu können. Die XCS-Zeit garantiert bei zwei aufeinander folgenden Abfragen auf beliebigen Systemen diese Monotonie, falls die Zeitermittlung unter dem gleichen DLM-Lock erfolgt:

System A	`GET-LOCK (Lock-Name=X1,exklusiv)` für Datenblock x
	`GTIME(XCS-TIME)`
	`RELEASE-LOCK(Lock-Name=X1)`
System B:	`GET-LOCK (Lock-Name=X1,exklusiv)` für den gleichen Datenblock x
	`GTIME(XCS-TIME)`
	`RELEASE-LOCK(Lock-Name=X1)`

Die beiden GTIME-Aufrufe liefern echt monotone Zeitwerte, womit die Logging-Information für Manipulationen am Datenblock x chronologisch unterschieden werden kann.

Jedes System verwaltet auf folgende Weise eine eigene XCS-Zeit:

1. Bei exklusiver Lock-Zuteilung durch ein System A, das Verwalter eines DLM-Locks ist, an ein zweites System B wird zusätzlich zur Zuteilungsquittung das um den Wert „eine Mikrosekunde“ inkrementierte Maximum aus XCS-Zeit und Systemzeit an den Lock-Requestor geschickt.
2. System B empfängt die Lock-Zuteilung von System A

   Das Maximum der mit Lock-Zuteilung gelieferten Zeit des Systems A und der XCS-Zeit des Systems B sowie dessen aktueller Systemzeit bildet die neue aktuelle XCS-Zeit des Systems B.
3. GTIME-Aufruf zum Erhalt der XCS-Zeit durch eine Anwendung

   Das um den Wert „eine Mikrosekunde“ inkrementierte Maximum aus der aktuellen Systemzeit und der XCS-Zeit des Systems wird zur aktuellen XCS-Zeit und dem Aufrufer bereitgestellt.

Dieses Verfahren hält die XCS-Zeit aller beteiligten Systeme etwa gleich der Systemzeit des Systems mit der am weitesten vorausgehenden Zeit.

> In der Ausgabe des Kommandos SHOW-SYSTEM-INFORMATION findet sich im Abschnitt SYSTEM-TIME-PARAMETER die entsprechende Information:
> `SYNCHRONIZATI-ON=XCS`.

Um auch die Systemzeiten möglichst identisch zu halten (um z.B. eine weitestgehende korrekte Chronologie von dezentralen System-Logging-Daten zu erhalten), synchronisiert in allen beteiligten Systemen das Subsystem XCS-TIME die Systemzeit auf die XCS-Zeit des Systems. Die Systemzeit und die XCS-Zeit der einzelnen Server sind damit nur annähernd gleich; eine Garantie für eine maximale Abweichung gibt es nicht.

Der Eintritt eines Servers in einen Verbund wird solange verzögert, bis dessen Systemzeit an die der bereits aktiven Partner angepasst ist. Ist der Zeitunterschied bei Eintritt größer als 15 Minuten, wird der Eintritt abgewiesen.

Externe Zeitgeber und XCS-Verbund

Ein XCS-Verbund kann zusätzlich mit einem externen Zeitgeber synchronisiert werden.

> Es wird empfohlen, einen externen Zeitgeber entweder an alle oder an nur einen Teilnehmer des XCS-Verbunds anzuschließen.
> *Begründung*
> Wenn nicht alle Teilnehmer des Verbunds an diesen Zeitgeber angeschlossen sind, dann überwachen die angeschlossenen Teilnehmer die Zeiten der nicht angeschlossenen Teilnehmer und verschicken bei Bedarf Zeitanpassungs-Aufträge an diese.
> Beim Empfänger können Anpassungsaufträge verschiedener Absender aufgrund unterschiedlicher Laufzeiten jedoch widersprüchlich erscheinen. Dies kann zu erhöhtem CPU-Zeit-Bedarf und zu Zeitabweichungen führen.

Hinweise zum Einsatz unter VM2000

Ein VM2000-Gastsystem (auch das Monitorsystem) kann unabhängig von den restlichen virtuellen Maschinen des Servers an einem XCS-Verbund beteiligt sein. Ist das Monitorsystem Teilnehmer eines derartigen Verbundes, basiert die Emulation der an die Gastsysteme gelieferten SVP-Zeit (Store-Real-Clock) auf der tatsächlichen SVP-Zeit und nicht auf der XCS-synchronisierten Systemzeit des Monitorsystems.

18.4.5 Generelle Empfehlung für den XCS-Verbund

Die MUs aller Server, auf denen an einem XCS-Verbund (mit oder ohne externe Zeitgeber) beteiligte Systeme laufen, sollten alle in einen NTP-Verbund aufgenommen werden.

18.4.6 Synchronisationsanzeigen bei /SHOW-SYSTEM-INFORMATION

Die folgende Tabelle stellt die möglichen Anzeigen des Kommandos SHOW-SYSTEM-INFORMATION im Abschnitt SYSTEM-TIME-PARAMETER für den Wert von SYNCHRONIZATION auf beiden Server-Typen nach Prioritäten aufsteigend dar. Beim Vorhandensein mehrerer Zeitgeber bestimmt die höchste Priorität unter ihnen den Wert von SYNCHRONIZATION.

SYNCHRONIZATION =	**SU /390**	**SU x86**
*NONE	Fehlerfall: SVP-Uhr zur Zeit nicht verfügbar oder Differenz zwischen Systemzeit und Zeitgeber zu groß für eine Synchronisation!	Fehlerfall: SVP-Uhr zur Zeit nicht verfügbar oder Differenz zwischen Systemzeit und Zeitgeber zu groß für eine Synchronisation!
X2000	entfällt	Standardwert (ohne externe Synchronisation)
SKP-X	Standardwert (ohne externe Synchronisation)	entfällt
DCE/NTP	Synchronisation durch NTP-Verbund (in BS2000)	Synchronisation durch NTP-Verbund (in BS2000)
XCS	Synchronisation durch XCS-Verbund	Synchronisation durch XCS-Verbund
SERVER-CONN-EXT-REF	Externer Zeitgeber über die MU oder (für VM2000-Gastsysteme) im Monitorsystem angeschlossen	Externer Zeitgeber über die MU oder (für VM2000-Gastsysteme) im Monitorsystem angeschlossen
BS2-CONN-EXT-REF	Externer Zeitgeber über NTP (BS2000) angeschlossen	Externer Zeitgeber über NTP (BS2000) angeschlossen

18.5 Unterbrechungsfreie Sommer-/Winterzeitumstellung

Die Zeitumstellung kann unterbrechungsfrei erfolgen, d.h. das System wird kontinuierlich über eine Zeitumstellung hinweg betrieben.

An den beiden Terminen der Zeitumstellung wird die Systemzeit entsprechend angepasst: d.h. in unserer mitteleuropäischen Zeitzone wird die Systemzeit im Frühjahr (meist am letzten Märzwochenende) um 1 Stunde vorgestellt, im Herbst (am letzten Oktoberwochenende) um 1 Stunde zurückgestellt.

Das Verstellen erfolgt in einem Schritt und damit anders als die sonstigen Zeitsynchronisationen: Das TODR wird nicht geändert, es läuft monoton weiter. Stattdessen wird der TODR-Korrekturwert entsprechend manipuliert, um die verstellte Systemzeit zu erhalten. Diese Umstellung wird dadurch ausgelöst, dass die Systemtask TIME bei Startup (und nach jeder Umstellung) einen Timer mit dem nächsten Umstelltermin aufzieht.

Umstellungszeitpunkte werden über den GTIME-Abschnitt der Startup-Parameterdatei bekanntgegeben. Fehlen dort zukünftige Umstellzeitpunkte, wird eine entsprechende Warnung ausgegeben, es sei denn, das Schlüsselwort SINGLE ist hinter dem letzten Umstellungszeitpunkt angegeben. Umstellungszeitpunkte können auch im laufenden BS2000-Betrieb mit den Kommandos ADD-/MODIFY-/REMOVE-/SHOW-CHANGE-DATE verwaltet werden. Änderungen von Umstellungszeitpunkten durch die Kommandos müssen aber manuell in die Parameterdatei eingetragen werden, wenn sie im nächsten Systemlauf gültig sein sollen.

Für Regionen, in denen es heute ein Wechsel zwischen Sommer- und Winterzeit gibt, für die Zukunft aber die Einführung einer festen, unveränderlichen Zeitzone geplant ist, sollte das Schlüsselwort SINGLE hinter dem letzten Umstellungszeitpunkt in der Parameterdatei eingetragen werden.

Falls die Anwendungs-Software noch nicht vollständig von Zeitermittlung mit STCK auf GTIME umgestellt ist, kann die Systemzeitumstellung durch Angabe von DIFF=0:00 im Parameterservice unterbunden werden. Das System muss dann neu gestartet werden.

Hinweise

- Die Schnittstelle CTIME stellt zusätzliche Warnungen für Eingabe- bzw. Ausgabezeitstempel in der doppelten bzw. übersprungenen Stunde bereit. Die mit GTIME gelieferte Systemzeit (lokale Zeit) macht zum Umstellungszeitpunkt einen entsprechenden Sprung. Anwendungen, die mit Zeitstempeln rechnen bzw. diese vergleichen, müssen negative Zeitsprünge der lokalen Zeit verarbeiten können. Es wird entweder eine Verfahrensumstellung auf UTC-Zeit oder die zusätzliche Mitführung der Sommerzeit-/Winterzeit-Anzeige empfohlen, welche bei GTIME/CTIME ebenfalls bereitgestellt wird. Die aktuelle Zeit darf nicht durch direkten TODR-Zugriff ermittelt werden.
- Benutzen Benutzerprogramme statt GTIME noch den STCK-Befehl und eigene Umrechnungsalgorithmen für Zeitstempel, sind sie auf die lokale Zeit im TODR angewiesen. In diesem Fall muss das System bei Zeitumstellung wie bisher neu geladen werden.
- Sind alle Benutzerprogramme schon auf GTIME zur Zeitabfrage umgestellt worden oder verrechnen sie STCK-Werte nur zu Differenzen, kann die unterbrechungsfreie Zeitumstellung genutzt werden.

Verhalten des Job-Managements bei Zeitumstellung

- Umstellung von Winter- auf Sommerzeit
  - Termin-Jobs

    Durch die Umrechnung der Startzeit in UTC und anschließender Rückkonvertierung kommt ein um eine Stunde verschobener Startzeitpunkt zustande.

    Beispiel: ENTER <datei>,START=AT(02:30) führt zum Start um 03:30 (LTI).
    Diese Startzeit vererbt sich auf die Nachfolgejobs, wenn der Job der erste von mehreren Repeat-Jobs ist.
  - Repeat-Jobs mit Periode DAILY/WEEKLY:

    Repeat-Job-Wiederholungen, die in diesem Intervall starten müssten, fallen aus. Stattdessen wird der nächste Nachfolger genommen, das nächste Starten erfolgt also 1 Tag oder 1 Woche später.
  - Repeat-Jobs mit Stunden-/Minute-Perioden

    Die Startzeiten der Jobwiederholungen werden einfach durch Addition der Minuten der Periode auf die letzte Startzeit berechnet (in JMS-Zeit, UTC), sie überspringen also das Intervall einfach.

    Beispiel Startzeitpunkte: ... 01:45 , 01:55 , 03:05 , 03:15 ... (LTI)
- Umstellung von Sommer- auf Winterzeit
  - Termin-Jobs

    Die interne Umrechnung der Startzeit eines Termin-Jobs in UTC unterstellt Winterzeit bei der Umrechnung; der Start erfolgt dann entsprechend in der zweite Stunde des Intervalls.
  - Repeat-Jobs mit Periode DAILY/WEEKLY:

    Repeat-Job-Wiederholungen, die in diesem Intervall starten müssten, werden ebenso behandelt, starten also im zweiten Intervall.
  - Repeat-Jobs mit Stunden-/Minute-Perioden

    Die Startzeiten der Jobwiederholungen werden einfach durch Addition der Minuten der Periode auf die letzte Startzeit berechnet (in JMS-Zeit, also UTC), sie laufen also ggf. in jedem der beiden Intervalle.

Hinweise zum Verhalten von AVAS bei Zeitumstellung

- Umstellung von Winter- auf Sommerzeit

  Die für den Zeitraum 2.00 – 3.00 Uhr geplanten Netze werden plötzlich entsprechend den Einstellungen für Verspätungen (NET-DELAY-SOLUTION) gestartet und laufen dann normal ab. Kurz nach 3.00 Uhr kommt es somit vermehrt zu AVAS-Aktivitäten, da alle Netze, die in der verlorenen Stunde gestartet werden sollen, sofort behandelt werden.
- Umstellung von Sommer- auf Winterzeit

  Die (seit 2.00 Uhr) bereits gestarteten Netze laufen normal weiter, da sie unabhängig von der Zeit nach dem Vorläufer-Nachfolger-Prinzip gestartet werden. Nach dem Zurückstellen der Zeit werden bis 3.00 Uhr keine Netze gestartet. Mit dem Zeitstempel in der Journaldatei kann es u.U. beim Sortieren Probleme geben, da die Startzeiten von Nachfolgejobs vor denen der Vorläufer liegen können.

18.6 Systemstart mit spezieller Systemzeit

Die Systemzeit kann mit einer beliebigen Zeit initialisiert werden. Der Systemstart ist dazu im DIALOG-Modus mit der IPL-Option UNLOCK durchzuführen. Über zu beantwortende Meldungen kann der Operator eine beliebige Zeit angeben. Die SVP-Zeit bleibt davon unberührt.

Unter VM2000 kann eine beliebige VM mit fiktiver Systemzeit und entsprechend versetzter UTC-Zeit betrieben werden. Allerdings muss jeder Systemstart dieses Gastsystems im DIALOG-Modus und der IPL-Option UNLOCK erfolgen.

Es findet keine Synchronisation mit einem evtl. vorhandenen externen Zeitgeber statt, wenn die manuell eingestellte Zeit mehr als 15 Minuten von der SVP-Zeit abweicht. Die Systemzeit taktet mit der HW-TODR-eigenen Qualität.
Da die veränderte Systemzeit auch für Zeitstempel im Systemkatalog genutzt wird, kann z.B. bei vorgestellter Systemzeit und anschließender Wiedernutzung des Pubsets in der Jetzt-Zeit der Dateizugriff wegen noch nicht abgelaufenem EXPIRATION-DATE verwehrt sein. Anwendungsspezifische Daten mit Zeitstempeln sind nach einer derartigen zweifachen Portierung ebenfalls beliebig inkonsistent und können zu undefiniertem Programmverhalten führen (insbesondere bei Datenbanken).

Einen speziellen Fall stellt der permanente Betrieb eines Systems mit fremder Zeitzone dar. Die Umstellung auf eine andere Zeitzone im Monitorsystem bzw. native funktioniert nur, falls im SVP bzw. im Trägersystem auch die neue Zeitzone eingestellt und anschließend der Startup mit IMPL durchgeführt wird.
Ansonsten (also inbesondere bei fremder Zeitzone für ein Gastsystem) darf das System die vom SVP bei einem Startup über „Store Real Clock“ erhaltene Zeit nicht als Systemzeit verwenden. Die korrekte, zur gewünschten Zeitzone passende Systemzeit kann nur noch durch manuelle Eingabe entstehen. Dazu muss jeder Startup dieses Systems als Dialog-Startup mit der Option UNLOCK erfolgen.
Darüber hinaus ist zu beachten:

- Enthält die GTIME-Parameterdatei nur die Informationen der gewünschten Zeitzone, kann der Makro CTIME im laufenden Betrieb keine Umrechnungen für fremde Zeitzonen durchführen.
- Enthält die GTIME-Parameterdatei Angaben zu mehreren Zeitzonen, ist nach manueller Eingabe der Systemzeit nicht ersichtlich, welche davon die richtige ist. Die Zeitinitialisierungsroutine führt dann mit dem Operator einen Meldungsdialog, welcher die Eingabe einer Zone, nicht aber einer Sommer-/Winterzeit-Differenz bzw. von Umstellungszeitpunkten für diese Zone, ermöglicht.
  Die eingegebene Zone wird im SVL hinterlegt, ein sinnvolles Arbeiten ist mit einem derartigen System aber nicht möglich. Erst beim nächsten Startup – natürlich im Modus DIALOG UNLOCK – wird statt der SVP-Zone die im SVL hinterlegte Zone verwendet und damit die umgestellte Parameterdatei auswertbar.

18.7 TASKDATE: Testen in simulierter Zeit

Das Tool TASKDATE wurde zum Test von datum- oder zeitsensitiven Anwendungen entwickelt.

Es ermöglicht einer Task, solange sie sich im nichtprivilegierten Zustand (TU) befindet, in versetzter Zeit zu laufen. Dazu werden alle Zeitabfragen aus TU (GTIME, GDATE, GETOD sowie Zeitservices der Hochsprachen – nicht STCK) mit einer vorher über das Kommando SET-TASK-CLOCK festgelegten, versetzten Zeit vom System befriedigt. Die simulierte Zeit kann optional an Folgetasks (ENTER) vererbt werden. Das System, Katalogeinträge der Tasks sowie Jobstart-Zeitpunkte basieren weiterhin auf der Systemzeit. Das Tool ist als DSSM-Subsystem sowie als Erweiterung des BS2000-Zeitmanagements, JMS, SDF-P, JV und openUTM realisiert. Das Tool kann im laufenden Betrieb mittels einer mit ausgelieferten Startprozedur gestartet werden.

19 Anhang

Im Anhang finden Sie folgende Übersichten und Tabellen:

- Zeichensatz bei Ein-/Ausgabe über Konsole
- Übersicht zu Testprivilegien

Spezielle Übersichten, die sich nicht mehr in diesem Anhang befinden, finden sich an anderer Stelle oder in anderen Handbüchern:

- Klasse-2-Systemparameter (kurz: Systemparameter),
  siehe das Kommando SHOW-SYSTEM-PARAMETERS im Handbuch „Kommandos“ [27]
- Gerätetyp-Tabelle, Volumetyp-Tabelle, Organisation der Plattenspeicher
  (siehe Handbuch „Systeminstallation“ [55])
- Übersicht über die Kommandos mit den dazugehörenden Privilegien, soweit sie Bestandteil von BS2000 OSD /BC und im Handbuch „Kommandos“ beschrieben sind,
  siehe Handbuch „Kommandos“ [27]
- Übersicht über die Operatorkommandos mit ihren Berechtigungsschlüsseln,
  siehe Abschnitt „Aufgabengebiete und ihre Aufteilung auf Konsolen"

19.1 Zeichensatz bei Ein-/Ausgabe über Konsole

Bei **Eingaben** über Konsole werden alle auf der jeweiligen Tastatur eingebbaren Zeichen unterstützt. Sie werden intern vom Zeichencode der Konsole in EDCDIC.DF.04 umgewandelt.

Bei **Ausgaben** über Konsole wird der Zeichencode umgesetzt in einen Zeichencode, der auf der jeweiligen Konsole darstellbar ist. Alle nichtabdruckbaren Zeichen, außer X'15', werden in Leerzeichen X'40' umgewandelt.

19.2 Übersicht zu Testprivilegien

Die Testprivilegien steuern die Ausführbarkeit von Software- und Hardware-Diagnosetätigkeiten in BS2000. Der Hauptnutzer ist die Testhilfe AID, siehe Handbuch „AID“ [2].

Die Testprivilegien unterteilen sich in Lese- und Schreibprivilegien. Das Schreibprivileg eines Benutzers darf nicht größer sein als sein Lese-Privileg. Lese- und Schreibprivilegien sind jeweils hierarchisch definiert von 1 (niedrig) bis 9 (hoch). Eine Privilegierung mit dem Wert n umfasst implizit die Zugriffsberechtigungen der niedrigeren Privilegienstufen 1 bis n-1.

Die maximalen Testprivilegien für eine Benutzerkennung werden über die privilegierten Kommandos ADD-USER bzw. MODIFY-USER ATTRIBUTES festgelegt. Systemweite Höchstwerte für die Lese- und Schreibprivilegien können über die Systemparameter RDTESTPR und WRTESTPR definiert werden.
Die taskspezifische Einstellung erfolgt über das Kommando MODIFY-TEST-OPTIONS.

Informationen über die aktuellen Testprivilegien liefert das Kommando SHOW-TEST-OPTIONS.

Die nachfolgenden Tabellen geben eine Übersicht über die Testprivilegien bei AID und bei weiteren Software-Diagnoseprodukten.

19.2.1 Testprivilegien bei AID

Wurde aus einer lesegeschützten Datei (nach)geladen und das Lesekennwort nicht angegeben, so liegt eine „Execute-Only“-Situation vor. In einer „Execute-Only-Situation“ ist, unabhängig von den eingestellten Testprivilegien, kein Testen mit AID möglich.

Wert	Charakteristik
1	• Zugriff auf die Seiten im eigenen Benutzeradressraum[1)] und im Systemadressraum[2)], die keine Secret Pages[3)] und mit den Zugriffsrechten des normalen Benutzers lesbar sind • Zugriff auf die Benutzer-PCBs der eigenen Task
2	• zusätzlich Zugriff auf den TCB der eigenen Task
3	• zusätzlich Zugriff auf die Secret Pages im eigenen Benutzeradressraum, die mit den Zugriffsrechten des normalen Benutzers lesbar sind
4	nicht vergeben
5	nicht vergeben
6	• zusätzlich Zugriff auf die Seiten im eigenen Benutzeradressraum, die keine Secret Pages und nur mit den Zugriffsrechten des Betriebssystems lesbar sind • zusätzlich Zugriff auf die System-PCBs der eigenen Task und auf die XVT
7	nicht vergeben
8	• zusätzlich Zugriff auf alle Seiten im Systemadressraum, alle Secret Pages und alle Seiten anderer Tasks • zusätzlich Zugriff auf die Benutzer-PCBs, System-PCBs und TCBs anderer Tasks
9	nicht vergeben

Tabelle 51: Lese-Testprivilegien bei AID-Lesezugriffen

Wert	Charakteristik
1	• Zugriff auf die Seiten im eigenen Benutzeradressraum[1)], die keine Secret Pages[3)] und mit den Zugriffsrechten des normalen Benutzers beschreibbar sind • zusätzlich Zugriff auf die tasklokalen Seiten des Benutzers im eigenen Programmraum, die keine Secret Pages und mit den Zugriffsrechten des normalen Benutzers „nur lesbar“[4)] sind • Zugriff auf die Benutzer-PCBs der eigenen Task
2	nicht vergeben
3	• zusätzlich Zugriff auf die Secret Pages im eigenen Benutzeradressraum, die mit den Zugriffsrechten des normalen Benutzers beschreibbar sind • zusätzlich Zugriff auf die tasklokalen Secret Pages des Benutzers im eigenen Programmraum, die mit den Zugriffsrechten des normalen Benutzers „nur lesbar“ sind
4	nicht vergeben
5	nicht vergeben
6	• zusätzlich Zugriff auf die Seiten im eigenen Benutzeradressraum, die keine Secret Pages und nur mit den Zugriffsrechten des Betriebssystems beschreibbar sind
7	nicht vergeben
8	• zusätzlich Zugriff auf alle Seiten im Systemadressraum[2)], alle Secret Pages und alle Seiten anderer Tasks • zusätzlich Zugriff auf die XVT sowie auf die Benutzer-PCBs, System-PCBs und TCBs aller Tasks
9	nicht vergeben

Tabelle 52: Schreib-Testprivilegien bei AID-Schreibzugriffen

Legende

[1)] Programmraum- und Datenräume des Benutzers
[2)] Systemraum- und Datenräume des Systems
[3)] gegen Diagnosezugriffe geschützte Seiten
[4)] Das Seitenattribut „nur lesbar“ wird in diesem Fall von AID ignoriert.

Anmerkung zum OWN-UID-DEBUGGING

Über den Operanden OWN-UID-DEBUGGING beim Kommando MODIFY-TEST-OPTIONS kann der „statische AID-Test mit niedriger Testprivilegierung“ für andere Tasks, die unter der eigenen Benutzerkennung laufen, zugelassen werden. Falls dies der Fall ist, sind bei lesenden Zugriffen auf den Benutzeradressraum dieser Tasks die gleichen Testprivilegien ausreichend wie bei Zugriffen auf den Benutzeradressraum der eigenen Task.

Für schreibende Zugriffe auf den Benutzeradressraum dieser Tasks ist mindestens ein Schreibprivileg von 2 erforderlich, darüber hinaus gelten die gleichen Testprivilegien wie bei Schreibzugriffen auf den Benutzeradressraum der eigenen Task. Bei Schreibzugriffen auf Memory-Pool-Seiten dieser Tasks wird grundsätzlich das Schreibprivileg 8 verlangt.

19.2.2 Testprivilegien bei weiteren Software-Diagnoseprodukten

Wert	Produkt	Charakteristik
3	CDUMP	Ein Systemdump darf unter einer normalen Benutzerkennung gezogen werden.
8	ANITA	Das aktive System darf mit DAMP und anderen Benutzerprogrammen, die ANITA nutzen, diagnostiziert werden.

Tabelle 53: Lese-Testprivilegien der Software-Diagnoseprodukte

20 Literatur

Die Handbücher finden Sie im Internet unter *http://bs2manuals.ts.fujitsu.com*. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

[1] **BS2000**
Abrechnungssätze
Benutzerhandbuch

[2] **AID**
Basishandbuch

[3] **ARCHIVE**
Benutzerhandbuch

[4] **openNet Server**
BCAM
Benutzerhandbuch

[5] **BLSSERV**
Bindelader-Starter in BS2000
Benutzerhandbuch

[6] **BINDER**
Binder in BS2000
Benutzerhandbuch

[7] **CALENDAR**
Benutzerhandbuch

[8] **ETERNUS CS**
Inbetriebnahme und Administration des ETERNUS CS
Benutzerhandbuch

[9] **CRYPT**
Sicherheit mit Kryptographie
Bentzerhandbuch

[10] **DAB**
Disk Access Buffer
Benutzerhandbuch

[11] **DCAM**
COBOL-Aufrufe
Benutzerhandbuch

[12] **DCAM**
Makroaufrufe
Benutzerhandbuch

[13] **DCAM**)
Programmschnittstellen
Beschreibung

[14] **BS2000 OS DX**
Diagnosehandbuch
Benutzerhandbuch

[15] **BS2000 OS DX**
Dienstprogramme
Benutzerhandbuch

[16] **Distributed Print Services**
Drucken in Computernetzen
Benutzerhandbuch

[17] **-**

[18] **DSSM/SSCM**
Subsystem Management in BS2000
Benutzerhandbuch

[19] **BS2000 OS DX**
Einführung in das DVS
Benutzerhandbuch

[20] **BS2000 OS DX**
DVS-Makros
Benutzerhandbuch

[21] **ELSA**
Error Logging System Analysis
Benutzerhandbuch

[22] **FDDRL**
Physikalische Datensicherung in BS2000
Benutzerhandbuch

[23] **openFT (BS2000)**
Managed File Transfer in der offenen Welt
Benutzerhandbuch

[24] **HSMS**
Hierarchisches Speicher Management System
Benutzerhandbuch

[25] **IMON**
Installationsmonitor
Benutzerhandbuch

[26] **JV**
Jobvariablen
Benutzerhandbuch

[27] **BS2000 OS DX**
Kommandos
Benutzerhandbuch

[28] **BS2000**
System-Managed Storage
Benutzerhandbuch

[29] **LMS**
SDF-Format
Benutzerhandbuch

[30] **BS2000 OS DX**
Makroaufrufe an den Ablaufteil
Benutzerhandbuch

[31] **MAREN**
Bandverwaltung in BS2000

[32] **HIPLEX MSCF**
BS2000-Rechner im Verbund
Benutzerhandbuch

[33] **OMNIS/OMNIS-MENU**
Funktionen und Kommandos
Benutzerhandbuch

[34] **OMNIS/OMNIS-MENU**
Administration und Programmierung
Benutzerhandbuch

[35] **BS2000**
Operator-Kommandos (ISP-Format)
Benutzerhandbuch

[36] **PCS**
Performance Control Subsystem
Benutzerhandbuch

[37] **Performance Handbuch**
Benutzerhandbuch

[38] **POSIX**
Kommandos
Benutzerhandbuch

[39] **POSIX**
Grundlagen für Anwender und Systemverwalter
Benutzerhandbuch

[40] **PROP-XT**
Programmiertes Operating mit komfortablen Sprachmitteln von SDF-P

Produkthandbuch

[41] **ROBAR**
Steuerung von MBK-Archivsystemen
Benutzerhandbuch

[42] **PRM**
Benutzerhandbuch

[43] **SDF**
Dialogschnittstelle SDF
Benutzerhandbuch

[44] **SDF-A**
Benutzerhandbuch

[45] **SDF-P**
Programmieren in der Kommandosprache
Benutzerhandbuch

[46] **SECOS**
Security Control System - Zugangs- und Zugriffskontrolle
Benutzerhandbuch

[47] **SECOS**
Security Control System - Beweissicherung
Benutzerhandbuch

[48] **SHC-OSD**
Storage Management für BS2000
Benutzerhandbuch

[49] **openSM2**
Software Monitor
Benutzerhandbuch

[50] **SM2-PA**
SM2-Programmanalysator
Benutzerhandbuch

[51] **SNS**
SPOOL Notification Service
Benutzerhandbuch

[52] **SPACEOPT**
Optimierung und Reorganisation von Platten
Benutzerhandbuch

[53] **SPOOL**
Benutzerhandbuch

[54] **SPSERVE**
Benutzerhandbuch

[55] **BS2000 OS DX**
Systeminstallation
Benutzerhandbuch

[56] **BS2000 OS DX**
System-Exits
Benutzerhandbuch

[57] **FUJITSU Server BS2000 SE Serie**
Bedienen und Verwalten
Benutzerhandbuch

[58] **Unicode im BS2000**
Übersichtshandbuch

[59] **openUTM** (BS2000, UNIX, Windows)
Anwendungen generieren
Benutzerhandbuch

[60] **VM2000**
Virtuelles Maschinensystem
Benutzerhandbuch