

FUJITSU Software BS2000

HSMS V12.0A

Band 1: Funktionen, Verwaltung und Installation

Benutzerhandbuch

Ausgabe Juni 2020

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 bs2000services@ts.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 © 2020 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

Inhaltsverzeichnis

HSMS V12.0A 2020-06 Band 1: Funktionen, Verwaltung und Installation	14
1 Einleitung	15
1.1 Zielsetzung und Zielgruppen des Handbuchs	17
1.2 Konzept des Handbuchs	18
1.3 Änderungen gegenüber dem Vorgänger-Handbuch	19
1.4 Darstellungsmittel	20
2 Wesentliche Nutzungsmodelle von HSMS	21
2.1 Grundfunktionen	22
2.2 Speicherhierarchie	25
2.3 Archive	28
2.4 Miteigentümerschaft	30
2.5 Benutzerklassen	31
2.6 HSMS in einer Umgebung von SM-Pubsets	32
2.7 BS2000 Backup-Server in Shared-Pubset-Umgebung	34
2.8 Sicherung ferner Rechner mit HSMS	35
2.8.1 Sicherung von UNIX-Workstations	36
2.8.2 Speicherhierarchie „Knoten-S0“	37
2.8.3 Archive	38
2.8.4 Benutzerklassen	39
2.8.5 Unterstützung von Knoten-Pfadnamen	40
2.9 Anweisungseingabe über SDF	41
3 HSMS-Archive	42
3.1 Archivdefinition	44
3.1.1 Nutzung des Archivs (Archivtyp)	45
3.1.2 Zugriffsrechte	46
3.1.3 Monitoring für ein Archiv einstellen	49
3.1.4 Backup-Server für ein Archiv einstellen	50
3.2 Archivverzeichnis	51
3.3 Standard-Systemarchive	52
3.4 Zusammenhang zwischen SF-Pubsets und Archiven	55
3.4.1 Mögliche Organisationsformen	56
3.4.1.1 Zentrale Verwaltung	57
3.4.1.2 Dezentrale Verwaltung	58
3.4.1.3 Vorteile (+) und Nachteile (-) der beiden Organisationsformen	59
3.4.1.4 Empfohlene Organisationsform: Dezentrale Verwaltung	61
3.5 Zusammenhang zwischen SM-Pubsets und Archiven	63
3.6 Zusammenhang zwischen Knoten-S0 und Archiven	64

3.6.1 Zentrale und dezentrale Verwaltung	65
3.7 Sicherungsdatei	66
3.7.1 Sicherungsdateinamen	68
3.7.2 Schutzfrist und Freigabedatum	69
3.8 Sicherungsversion	72
3.8.1 Sicherungstyp	73
3.8.2 Komprimieren von Sicherungsversionen	74
3.8.3 Erweitern von Sicherungsversionen	75
3.9 Standard-Sicherungsdatei	76
3.9.1 Sicherungsdateien fortschreiben	77
3.10 Verwaltung der Archive	79
3.10.1 Datenträger-Pool verwalten	80
3.10.2 Sicherungsdateien freigeben	81
3.10.3 Auf Archivobjekte ohne Archivverzeichnis zugreifen	82
3.10.4 Archivdefinition löschen	83
3.11 Management-Klassen	84
3.11.1 Zuweisung einer Management-Klasse	85
3.11.2 Attribute einer Management-Klasse	86
3.11.3 Verwendung der Management-Klassen	88
3.11.4 Schutz der Management-Klassen	89
3.12 Directory-Dateien mit DIRCONV bearbeiten	90
3.12.1 DIRCONV-Funktionen	91
3.12.1.1 Mischen von Directory-Dateien	92
3.12.1.2 Konvertieren von Directory-Dateien	93
3.12.1.3 Umbenennen von Katalogkennungen	95
3.12.1.4 Entfernen einer Katalogkennung	96
3.12.1.5 Anzeigen von Katalogkennungen	97
3.12.1.6 Aktualisieren des Volume-Katalogs	98
3.12.1.7 Entfernen von nicht katalogisierten Dateien	99
3.12.1.8 Reorganisieren von Directory- und Repository-Dateien	100
3.12.1.9 Konvertieren von Repository-Dateien	101
3.12.2 Ein-/Ausgaben und Aufruf von DIRCONV	102
3.12.3 DIRCONV-Anweisungen	103
3.12.3.1 MERGE-DIRECTORIES Mischen von Directory-Dateien	104
3.12.3.2 REMOVE-CATID Entfernen von Katalogkennungen	105
3.12.3.3 REMOVE-UNCATALOGED-FILES Entfernen von nicht katalogisierten Dateien	107
3.12.3.4 RENAME-CATID Umbenennen von Katalogkennungen	108
3.12.3.5 REORGANIZE-DIRECTORY Reorganisieren einer Repository-Datei	110
3.12.3.6 SET-CATID Konvertieren von Directory-Dateien	112
3.12.3.7 SHOW-CATID Ausgeben von Katalogkennungen	113

3.12.3.8 UPDATE-VOLUME-CATALOG Aktualisieren des Volume-Katalogs . . .	114
3.12.4 Nutzungsmodelle	115
3.12.4.1 SM-Pubset erstellen, der aus Pubsets besteht, die bereits mit einer gemeinsamen Directory-Datei gesichert wurden	116
3.12.4.2 SM-Pubset erstellen, der aus Pubsets besteht, die bereits mit ihrer eigenen Directory-Datei gesichert wurden	117
3.12.4.3 Transfer einer Benutzerkennung auf einen anderen Pubset	118
4 Funktionen von HSMS	119
4.1 Datensicherung (Backup)	120
4.1.1 Allgemeine Bemerkungen über die beiden Sicherungstypen	121
4.1.2 Sicherung mit BACKUP-FILES	122
4.1.2.1 BS2000-Dateien und Jobvariablen sichern	123
4.1.3 Versions-Backup	129
4.1.3.1 Differenzsicherungen beim Versions-Backup	131
4.1.3.2 Versions-Backups reorganisieren	132
4.1.4 Sicherung mit der Funktion „Concurrent Copy“	133
4.1.5 Sicherungsoptionen	140
4.1.6 Automatisches Duplizieren von Sicherungsdateien	144
4.1.7 Sicherung auf Platte oder auf Net-Storage	146
4.1.7.1 Sicherungsdateien und Knotendateien mit der Anweisung MOVE-SAVE- FILES verlagern	147
4.1.7.2 Sicherungsdateien beim Reorganisieren des Versions-Backups verlagern . . 149	
4.1.8 Gesicherte BS2000-Dateien und Jobvariablen restaurieren	150
4.1.8.1 Restaurieren von großen Dateien (> 32 GB)	152
4.1.8.2 Sicherungsversionen auswählen	153
4.1.8.3 Plattenspeicher reorganisieren	157
4.1.8.4 Dateien und Jobvariablen umbenennen	161
4.1.8.5 Aus einem Schattenarchiv restaurieren	162
4.1.8.6 Elemente aus einer PLAM-Bibliothek restaurieren	163
4.1.8.7 Net-Storage-Dateien restaurieren	164
4.1.9 Knotendateien des lokalen BS2000-UFS und Knoten-S0 sichern	165
4.1.9.1 Umfang der Sicherung	166
4.1.9.2 System-Backup-Archiv für Knotendateien	167
4.1.9.3 Voll- und Differenzsicherung	168
4.1.9.4 Automatisches Duplizieren von Knotensicherungsdateien	169
4.1.9.5 Sicherung von LATEST-BACKUPS-OR-S0	170
4.1.10 Gesicherte Knotendateien restaurieren	171
4.1.10.1 Sicherungsversionen auswählen	172
4.1.10.2 Knotendateien umbenennen	173
4.1.10.3 Aus einem Schattenarchiv restaurieren	174
4.1.11 Beispiele zum Backup	175

4.2 Langzeitarchivierung (Archival)	191
4.2.1 BS2000-Dateien archivieren	192
4.2.1.1 System-Langzeitarchiv	194
4.2.1.2 Logisches Freigabedatum	195
4.2.1.3 Zusatzinformationen	196
4.2.1.4 Archivierungsoptionen	197
4.2.1.5 Automatisches Duplizieren in ein Schattenarchiv	198
4.2.1.6 Hinweise zur Verwendung von Standard-Sicherungsdateien	199
4.2.2 Archivierte BS2000-Dateien restaurieren	200
4.2.2.1 Restaurieren von großen Dateien (> 32 GB)	201
4.2.2.2 Sicherungsversionen auswählen	202
4.2.2.3 Aus einem Schattenarchiv restaurieren	203
4.2.2.4 Elemente aus einer PLAM-Bibliothek restaurieren	204
4.2.2.5 Net-Storage-Dateien restaurieren	205
4.2.3 Knotendateien des BS2000-UFS und ferner Knoten-S0 archivieren	206
4.2.3.1 System-Langzeitarchiv	207
4.2.3.2 Logisches Freigabedatum	208
4.2.3.3 Zusatzinformationen	209
4.2.3.4 Automatisches Duplizieren in ein Schattenarchiv	210
4.2.4 Archivierte Knotendateien restaurieren	211
4.2.4.1 Sicherungsversionen auswählen	212
4.2.4.2 Aus einem Schattenarchiv restaurieren	213
4.2.5 Beispiel zur Langzeitsicherung	214
4.3 Verdrängung (Migration) von BS2000-Dateien	216
4.3.1 Dateien verdrängen	217
4.3.1.1 Verdrängen bestimmter Mengen	218
4.3.1.2 Schnelles Verdrängen ohne Bandverarbeitung (Schnellmigration)	219
4.3.1.3 Migrationssperre	220
4.3.1.4 Standardmäßig von der Migration ausgeschlossene Dateien	221
4.3.2 Verdrängen durch den HSMS-Verwalter	222
4.3.2.1 Auskunftsmöglichkeiten	223
4.3.2.2 Dateien automatisch verdrängen	224
4.3.3 Verdrängte Dateien	227
4.3.4 Verdrängung und große Dateien (> 32 GB)	229
4.3.5 Verdrängung und Concurrent Copy	230
4.3.6 Verdrängte Dateien zurückholen	231
4.3.6.1 Explizites Zurückholen durch HSMS-Aufruf	232
4.3.6.2 Implizites Zurückholen bei Dateizugriff	233
4.3.7 RESTORE von verdrängten Dateien	235
4.3.8 Verdrängung nach S0	237
4.3.9 Alle Dateien eines Benutzers verlagern	239

4.3.10 Beispiele zur Verdrängung	240
4.3.11 Beispiele zur Remigration	251
4.4 Datentransfer von BS2000-Dateien und Jobvariablen	256
4.4.1 Dateien und Jobvariablen exportieren	257
4.4.1.1 Katalogeinträge exportieren	259
4.4.1.2 Verzeichnis	260
4.4.2 Exportierte Dateien und Jobvariablen importieren	261
4.4.2.1 Mit Verzeichnis importieren	262
4.4.2.2 Mit MAREN-Unterstützung importieren	263
4.4.2.3 Dateien und Jobvariablen umbenennen	264
4.4.2.4 Directory importieren	265
4.4.2.5 Dateien auf Net-Storage importieren	266
4.4.2.6 Dateien aus der Sicherungsdatei eines Pubsets oder des Net-Storage importieren	267
4.4.3 Beispiele zum Datentransfer	268
4.5 Duplizieren von Sicherungsdateien	274
4.5.1 Sicherungsdateien kopieren	275
4.5.1.1 Doppelführung von Sicherungsdateien	276
4.5.1.2 Innerhalb von Backup-Archiven kopieren	277
4.5.1.3 Innerhalb von Migrationsarchiven kopieren	278
4.5.1.4 Innerhalb von Langzeitarchiven kopieren	279
4.5.1.5 Von einem Archiv in ein anderes Archiv kopieren	280
4.5.1.6 Zulässige Zielspeichermedien bei der Anweisung COPY-SAVE-FILE ...	281
4.5.1.7 Verhalten der Anweisung COPY-SAVE-FILE in Verbindung mit von VSNs unabhängigen Sicherungsdateinamen	282
4.5.1.8 Verhalten der Anweisung COPY-SAVE-FILE ... TO-STORAGE=*S1- STORAGE-LEVEL abhängig von den Umgebungen, in denen ursprüngliches und Zielarchiv definiert sind	283
4.5.2 Beispiele	286
4.6 Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien	296
4.6.1 Auswahl von BS2000-Dateien	297
4.6.1.1 Auswahl durch Operanden	298
4.6.1.2 Auswahl durch die HSMS-Anweisung SELECT-FILE-NAMES	301
4.6.1.3 Auswahl im Dialog	302
4.6.2 Auswahl von Jobvariablen	306
4.6.3 Auswahl von Knotendateien des BS2000-UFS	307
4.6.3.1 Auswahl durch Operanden	308
4.6.3.2 Auswahl durch die HSMS-Anweisung SELECT-NODE-FILES	311
4.7 Auftragsbehandlung (Requests)	312
4.7.1 Aktionsanweisungen, Aufträge und HSMS-Servertasks	313
4.7.2 Steuerung der Auftragszeit	315
4.7.3 Steuerung der Auftragsbearbeitung (Operation-Control)	320

4.7.4 Auftragsverwaltung	324
4.7.4.1 Auskunft über die Aufträge	325
4.7.4.2 Aufträge löschen	327
4.7.4.3 Aufträge wiederanlaufen lassen (Restart)	328
4.7.5 Auftragsverwaltung bei Shared-Pubsets	330
4.7.6 Aufträge wiederherstellen nach einem Host-Ausfall (Recovery)	332
4.7.7 Prioritäten für die Auftragsbearbeitung vergeben	334
4.7.8 Sammelaufträge	335
4.7.9 Asynchrone und synchrone Verarbeitung	336
4.7.9.1 Asynchrone Verarbeitung	337
4.7.9.2 Synchrone Verarbeitung	338
4.7.10 Parallele und serielle Verarbeitung in ARCHIVE	339
4.7.10.1 Automatische Aufteilung in Pakete	341
4.7.10.2 Automatisches Erzeugen von ARCHIVE-Subtasks	342
4.7.10.3 Steuern der Paketerzeugung durch den Benutzer (Paketgenerierung) ..	343
4.7.10.4 Steuern der erzeugten Subtasks durch den Benutzer	347
4.8 Datenträgerverarbeitung	348
4.8.1 Datenträger auf S2	349
4.8.2 Zuweisung von Datenträgern	350
4.8.2.1 Datenträgerzuweisung durch den Benutzer oder den Operator	351
4.8.2.2 Datenträgerzuweisung aus dem Datenträger-Pool	352
4.8.2.3 Datenträgerzuweisung über MAREN	353
4.8.3 Datenträger- und Gerätereservierung	354
4.8.4 Lagerorte von Datenträgern und Gerätepools	355
4.8.5 Bandverarbeitungszeiten	357
4.8.5.1 Bandzugriffe nicht einschränken	358
4.8.5.2 Bandzugriffe aufschieben	359
4.8.5.3 Bandverarbeitungszeiten festlegen	360
4.8.5.4 Schreibzugriffe nach Menge steuern (nur für Archive für BS2000-Dateien) ..	361
4.8.6 Steuerung der Bandverarbeitung	362
4.8.6.1 Größe von Bandblöcken	363
4.8.6.2 Magnetbandkassette entladen	364
4.9 Behandlung von BS2000-Platten	365
4.9.1 Speicherplatzanforderung für Sicherungsdatei optimieren	366
4.9.2 Konvertierung von BS2000-Dateien	367
4.10 Ausgaben von HSMS	369
4.10.1 Ausgaben nach SHOW-Anweisungen	370
4.10.1.1 Bildschirmmasken	371
4.10.1.2 S-Variablen	373
4.10.2 Reporte	374

4.11 Jobvariable zur Auftragsüberwachung	380
4.11.1 Bearbeitung der Jobvariable durch HSMS	381
4.11.2 Inhalt der Jobvariable	382
4.11.3 Jobvariable bei Shared-Pubsets	386
4.12 Aliasnamen für BS2000-Dateien und Jobvariablen	387
5 Verwaltung von HSMS	388
5.1 Betriebsweisen von HSMS	389
5.1.1 Definitions-/Auskunfts-Betrieb	390
5.1.2 Simulations-Betrieb	391
5.1.3 Operationeller Betrieb	392
5.2 Speicherhierarchie in einer Umgebung mit SF-Pubsets verwalten	393
5.2.1 SF-Pubsets verwalten	394
5.2.1.1 Pubset außerhalb der HSMS-Kontrolle	395
5.2.1.2 Pubset den Speicherebenen S0 und S1 zuordnen	396
5.2.1.3 SM-Pubset der Speicherebene S1 zuordnen	397
5.2.1.4 Pubset unter Verwaltung nehmen und Parameter festlegen	399
5.2.1.5 Auskunft über Parameter und Nutzung der Pubsets	400
5.2.1.6 Pubset aus HSMS-Verwaltung entlassen	401
5.2.2 Maßnahmen bei besonderen Situationen	402
5.2.2.1 Restore nach einem S0-Crash	403
5.2.2.2 Restore nach einem Crash auf S1 oder S2	404
5.2.2.3 Reorganisation der Speicherebene S1	405
5.2.2.4 Reorganisation der Speicherebene S2	406
5.2.2.5 Benutzerkennung auf einen anderen Pubset verlegen	407
5.2.2.6 Benutzer- oder Katalogkennung reorganisieren	408
5.3 Speicherhierarchie in einer Umgebung mit SM-Pubsets verwalten	410
5.3.1 Bearbeitung von SM-Pubsets	411
5.3.2 Umfang der HSMS-Operationen	412
5.3.3 Konvertieren eines SM-Pubsets	413
5.3.3.1 Erstellen eines SM-Pubsets ohne verdrängte Dateien	414
5.3.3.2 Wiederverwenden eines vorhandenen Archivverzeichnisses, dessen Umfang auf einen SM-Pubset beschränkt ist	415
5.3.3.3 Vorhandene Archivverzeichnisse, die nicht auf den SM-Pubset beschränkt sind	421
5.3.3.4 Konvertieren eines SF-Pubsets zu einem Volume-Set eines SM-Pubsets ...	422
5.3.4 Ändern der SM-Pubset-Parameter	423
5.3.5 Anzeigen der SM-Pubset-Parameter	424
5.3.6 SM-Pubset importieren oder exportieren	425
5.3.7 Hinzufügen/Entfernen eines Volume-Sets zu/aus einem SM-Pubset	426
5.4 Arbeiten mit Shared-Pubsets	427

5.4.1 Arbeiten mit Shared-SF-Pubsets	428
5.4.1.1 Master-Modus	429
5.4.1.2 Lokaler Modus	431
5.4.2 Arbeiten mit Shared-SM-Pubsets	432
5.4.3 Backup-Server	433
5.4.4 Übersicht	435
5.5 Arbeiten mit SM-Pubsets und verschiedenen HSMS-Versionen	439
5.6 Dateiattribute in einer Umgebung mit SM-Pubsets verwalten	440
5.7 Verwaltung der Workstations	442
5.7.1 Notwendige Tätigkeiten an einer Workstation bei Zugriff über NFS	443
5.7.2 Notwendige Tätigkeiten im BS2000 (Server)	444
5.7.3 Zusammenhang zwischen Knoten-S0 und Archiven	447
5.7.3.1 Ein zentrales Archiv und zentrale Knoten-S0-Deklaration	448
5.7.3.2 Ein Archiv für einen einzigen Knoten-S0 und eine Knoten-S0-Deklaration ...	449
5.7.3.3 Multiplexbetrieb	450
5.7.4 Beispiele	452
5.8 HIPLEX und HSMS	462
5.8.1 Wiederherstellen von DMS-Dateien (Recovery)	463
5.8.2 HIPLEX-Fähigkeit für Knotendateien	470
5.8.2.1 HSMS und das UNIX-Dateisystem (UFS)	471
5.8.2.2 Allgemeine Voraussetzungen	472
5.8.2.3 Einschränkungen	473
5.8.2.4 Nutzungsmodelle	474
5.8.3 HIPLEX-Fähigkeit für UFS-Läufe von HSMS vorbereiten	488
5.9 Steuerung der Verdrängung und Verwaltung der Migrationsarchive	489
5.9.1 Verdrängung steuern	490
5.9.1.1 System-Migrationsarchiv	491
5.9.1.2 Verdrängung zulassen	492
5.9.1.3 Migrationssperre unterdrücken	493
5.9.1.4 Ausnahmedatei	494
5.9.1.5 Implizites Zurückholen steuern	495
5.9.1.6 Migration / Recall mit openSM2 überwachen	496
5.9.1.7 Verdrängung beschränken auf schnell migrierbare Dateien (Schnellmigration)	498
5.9.2 Migrationsarchiv reorganisieren	499
5.9.2.1 Verdrängung von S1 auf S1	500
5.9.2.2 Verdrängung von S1 auf S2	501
5.9.2.3 Reorganisation eines Migrationsarchivs in S2	502
5.9.2.4 Kopieren der Sicherungsdateien	503
5.9.3 Datensicherung und verdrängte Dateien	504

5.9.3.1 Inkonsistente Dateien	505
5.9.3.2 Behebung von Inkonsistenzen bei migrierten Dateien	506
5.9.4 Beispiele	507
5.10 Weitere Steuerparameter	519
5.10.1 Anzahl der Servertasks	520
5.10.2 Größe des Common-Memory-Pools	521
5.10.3 Wartezeiten für synchrone Aufträge und auf Reservierung	522
5.10.4 Verarbeitungsmodus für Sicherungsdateien	523
5.10.5 Zentrale Auftragsüberwachung am SE Server	524
5.10.6 Aufträge aufbewahren	526
5.11 Privilegien und Sicherheitsaspekte von HSMS	527
5.11.1 Privilegien	528
5.11.2 Datenschutz	529
5.11.3 Behandlung der Dateiattribute	531
5.11.4 Datensicherheit	532
5.11.5 Verschlüsselte Dateien	533
5.12 Einrichten einer HSMS-Konfiguration (Beispiel)	534
6 Aufruf und Ablauf von HSMS	539
6.1 Laden und Entladen von HSMS	540
6.2 Aufruf von HSMS	542
6.2.1 Starten und Beenden durch den Benutzer	543
6.2.2 Kommandoprozessor SDF	544
6.2.3 Dialog- und Stapelbetrieb	545
6.2.4 Fehlerbehandlung	546
6.3 Aufruf von HSMS aus Programmen	547
6.4 Arbeitsdateien	559
6.5 Möglichkeiten zur Performance-Verbesserung	564
6.5.1 Maximale Bandblockgröße	565
6.5.2 Beschleunigtes Kopieren von Sicherungsdateien	566
7 Installation von HSMS	567
7.1 Systemvoraussetzungen	568
7.1.1 Bearbeitung von Daten im BS2000	569
7.2 Lieferumfang von HSMS	570
7.3 Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen	574
7.3.1 SF-Steuerdatei (Home-Pubset)	575
7.3.2 SF-Auftragsdatei (Home-Pubset)	576
7.3.3 Shared-Pubsets und verschiedene HSMS-Versionen	577
7.3.3.1 Arbeiten mit S1-SM-Pubsets in verschiedenen HSMS-Versionen	578
7.3.4 Steuer- und Auftragsdateien auf SM-Pubsets	579
7.3.5 Arbeiten mit der erweiterten Speicherebene S1 in einer SM-Umgebung	580

7.3.5.1	Einrichten der erweiterten S1-Ebene	581
7.3.5.2	Sicherung, Archivierung, Versions-Backup und Verdrängung auf die erweiterte S1-Ebene	582
7.3.5.3	Erweiterte Speicherebene S1 wieder auf ein Volume-Set reduzieren	583
7.3.5.4	Kompatibilität mit kleineren Versionen oder bei inhomogener Umgebung	584
7.3.6	Versions-Backup in einer Shared-Umgebung	585
7.4	Benutzerkennung SYSHSMS	586
7.5	Umgebung zur Sicherung ferner Knoten-S0 erzeugen	587
7.6	Erster Start von HSMS	588
7.7	Aufgaben bei jedem Startup des BS2000-Systems	590
7.8	Hinweise zur Einführung der Verdrängung	591
8	Zusammenarbeit von HSMS mit ARCHIVE	592
8.1	ARCHIVE-Funktionen in HSMS	593
8.2	Besonderheiten bei Übergang von ARCHIVE- zu HSMS- Betrieb	597
8.3	Einschränkungen für den Rückstieg auf ARCHIVE	598
8.3.1	Kompatibilität der Directory-Dateien	599
8.3.2	Einschränkungen durch neue Funktionen	600
9	Meldungen	603
9.1	Meldungsklassen	604
9.2	Meldungen anderer Produkte	606
10	Abrechnungssätze	607
10.1	CPU-Zeit und Ein-/Ausgaben bei HSMS-Tätigkeiten	608
10.2	Belegter Speicherplatz (Speicherebene S1 und S2)	614
11	Anhang	615
11.1	Einführung von HSMS im Rechenzentrum	616
11.1.1	Motivation zum Einsatz der Migrationsfunktion	617
11.1.2	Verdrängte Dateien	618
11.1.3	Verdrängte Dateien zurückholen	620
11.1.3.1	Explizites Zurückholen durch HSMS-Aufruf	621
11.1.3.2	Implizites Zurückholen durch Zugriff	622
11.2	Auswirkungen auf Benutzer-Anwendungen	623
11.2.1	Zurückholen beim Dateizugriff	624
11.2.2	Zurückholen durch SECURE-RESOURCE-ALLOCATION	625
11.2.3	Schutz der Dateien vor Verdrängung	626
11.3	Hinweise zur Erstellung automatischer Verfahren	627
11.3.1	Empfohlene Umstellungen	628
11.3.2	Zwingend nachzuprüfendes Verhalten	629
11.3.3	Unkritische Behandlung verdrängter Dateien	630
11.4	Diagnosehilfen und Trace-Funktionen	631
11.4.1	HSMS-Trace	632
11.4.2	ARCHIVE-Trace	633

11.4.3 Performance-Statistik	635
11.4.4 Weitere Hinweise zur Diagnose	638
12 Fachwörter	639
13 Abkürzungen	657
14 Literatur	661

HSMS V12.0A 2020-06 Band 1: Funktionen, Verwaltung und Installation

1 Einleitung

HSMS (**H**ierarchisches **S**peicher **M**anagement **S**ystem) ist ein BS2000-Softwareprodukt zur Datensicherung und zur Unterstützung der Datenverwaltung auf externen Speichern in einem BS2000-System.

Seit BS2000/OSD-BC V9.0 kann der gemeinschaftliche Speicherplatz eines Pubsets erweitert werden um Speicherplatz, den ein Net-Server remote als einen sogenannten Net-Storage zur Verfügung stellt. Hier unterstützt HSMS auch im BS2000 katalogisierte Net-Storage-Dateien.

Ab HSMS V10.0 kann im Shared-Pubsets-Verbund ein BS2000-Backup-Server konfiguriert werden, der die Sicherungsaufgaben für alle anderen BS2000-Hosts dieses Verbunds übernimmt. Dies entlastet die produktiven Systeme.

Mit HSMS können auch Datenbestände von vernetzten, fernen UNIX-Systemen, Netzwerkservers oder Workstations gesichert und bei Bedarf restauriert werden. Dazu müssen diese Systeme mit dem lokalen BS2000-UFS (POSIX) über NFS verbunden sein.

Die Datenmengen, die in Rechenzentren zu verarbeiten sind, nehmen immer mehr zu. Entsprechend wächst der Bedarf an Personal und Datenträgern für das Erstellen und Verwalten von Sicherungskopien der bearbeiteten Daten.

HSMS bietet sowohl der Rechenzentrumsverwaltung als auch den Datenbankverwaltern eine Funktion zur **Datensicherung**, die mit Differenzsicherung und Teilsicherung sowie dem Fortschreiben von Sicherungsdatenträgern zeit- und platzsparende Möglichkeiten bietet. Mit Differenzsicherung ist der in der Fachliteratur anzutreffende englische Begriff „incremental backup“ gemeint.

Häufig werden ältere Daten in einem System nur noch in Ausnahmefällen benötigt. Sie müssen aber wegen rechtlicher Vorschriften oder als Vorsorge für den Katastrophenfall langfristig aufbewahrt werden. Hierzu bietet HSMS eine von der Datensicherung unabhängige **Langzeitarchivierung** an. Die archivierten Daten werden – getrennt von den gesicherten Daten – in eigenen Archiven verwaltet.

Der Bedarf gerade an online-verfügbaren Plattenspeichern ist enorm gewachsen. Dies führt zu erheblichen Kosten für Geräte, Stellplatz und Verwaltung. Viele dieser Plattenspeicher sind jedoch aus Vorsorge gegen Sättigungszustände nur zum Teil ausgelastet. Mit der **Verdrängung** bietet HSMS einen Mechanismus, der für eine bessere Ausnutzung der Plattenspeicher sorgt: während des laufenden Betriebes werden Daten, die seit längerer Zeit nicht mehr benötigt wurden, von der Verarbeitungsebene auf eine Hintergrundebene umgeschichtet. Die Verdrängung lässt sich beim Erreichen von Sättigungszuständen auch automatisch anstarten. Bei Bedarf werden die Daten auch ohne Aufruf von HSMS durch den Benutzer auf die Verarbeitungsebene zurückgeholt.

HSMS realisiert die Grundfunktionen Datensicherung, Langzeitarchivierung und Verdrängung im Rahmen einer dreistufigen Hierarchie von Externspeichern. Die Datenträger eines BS2000-Systems – Plattenspeicher und Magnetbandkassetten – werden entsprechend ihrer unterschiedlichen Leistungsmerkmale hinsichtlich Speicherkosten und Zugriffszeitverhalten verschiedenen Ebenen zugeordnet. Der normalen Verarbeitungsebene wird eine online-verfügbare Hintergrundebene aus Plattenspeichern und eine Offline-Hintergrundebene aus Magnetbandkassetten zugeordnet.

Neben den Single-Feature-Pubsets (SF-Pubsets) unterstützt HSMS in besonderer Weise die System-Managed-Pubsets (SM-Pubsets), da die Hintergrund- und Sicherungsebenen mit den Meta-Informationen des Pubsets in dieses integriert sind und Backup- und Migrationsstrategien über Managementklassen gesteuert werden können.

Ferner bietet HSMS die Möglichkeit des **Datentransfers**: Dateien, Jobvariablen und auch Katalogeinträge von BS2000-Dateien können mit Hilfe von Magnetbandkassetten auf andere BS2000-Systeme oder andere Benutzerkennungen übertragen werden. Der Datentransfer entspricht der EXPORT/IMPORT-Funktion von ARCHIVE; die von HSMS oder ARCHIVE erzeugten Datenträger sind dabei austauschbar.

HSMS umfasst das Datensicherungsprodukt ARCHIVE. Außerdem arbeitet HSMS optional mit den Softwareprodukten MAREN und ROBAR zusammen.

Für BS2000-Dateien ist ein gleitender Übergang vom Datensicherungsprodukt ARCHIVE zu HSMS möglich, da die Archivverzeichnis-Dateien und die Sicherungsdatenträger kompatibel sind.

HSMS wird über eine komfortable Benutzeroberfläche bedient. Die HSMS-Anweisungen werden über SDF eingegeben.

HSMS bietet vielfältige Steuerungs- und Auskunftsmöglichkeiten, wie etwa

- die Einschränkung der Bandverarbeitung auf festgelegte Zeiten
- eine Wiederanlaufmöglichkeit für unterbrochene Aufträge
- Informationen über die Belegung und Nutzung der Pubsets

Die Dateien, die bei den Grundfunktionen bearbeitet werden sollen, können durch komfortable Auswahlmöglichkeiten festgelegt werden.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Dieses Handbuch richtet sich sowohl an den nicht-privilegierten Benutzer als auch an den HSMS-Verwalter.

Der nicht-privilegierte Benutzer sollte über grundlegende BS2000-Kenntnisse verfügen. Er sollte mit den wichtigsten Kommandos vertraut sein und besonders Kenntnisse des Datenverwaltungssystems des BS2000 besitzen.

Der HSMS-Verwalter sollte darüber hinaus Kenntnisse über das UNIX-Dateisystem (UFS) besitzen und mit der Systemverwaltung und der Rechenzentrumsorganisation vertraut sein.

1.2 Konzept des Handbuchs

Dieses Handbuch dient zur Einarbeitung in die Funktionen und in die Verwaltung von HSMS. Es ist sowohl für nicht-privilegierte BS2000-Benutzer als auch für HSMS-Verwalter gedacht, wobei im Einzelnen darauf hingewiesen wird, für wen die Ausführungen von besonderem Interesse sind.

Nach einer Einführung in die Grundfunktionen von HSMS sind die Speicherhierarchie und die Benutzerklassen beschrieben. Danach werden die Archive als die grundlegenden Verwaltungseinheiten von HSMS vorgestellt. Anschließend werden die Funktionen von HSMS eingehend dargestellt.

Die Kapitel zur Verwaltung und Installation sind für den HSMS-Verwalter gedacht; hier finden Sie auch Hinweise zur Einführung der Verdrängung.

Das Kapitel „Zusammenarbeit von HSMS mit ARCHIVE“ enthält die wichtigsten Produkte, die HSMS unterstützt. Besondere Hinweise wurden für diejenigen Benutzer aufgenommen, die schon mit der Bedienung von ARCHIVE vertraut sind. Sie finden hier Hilfen beim Umstieg von ARCHIVE auf HSMS und Übersichten, auf welche HSMS-Anweisungen die bekannten ARCHIVE-Funktionen abgebildet wurden.

Im Anhang bilden die drei Abschnitte „Einführung von HSMS im Rechenzentrum“, „Auswirkungen auf Benutzer-Anwendungen“ und „Hinweise zur Erstellung automatischer Verfahren“ eine Zusammenfassung, die vor der Einführung von HSMS und besonders vor der Einführung der Verdrängung an die Benutzer eines Rechenzentrums zur Information verteilt werden sollte.

Dieser Teil des Anhangs ist somit vom Vervielfältigungsverbot ausgenommen.

Zusätzlich enthält der Anhang Hinweise zur Fehlerdiagnose.

Das Abkürzungs- und Fachwörterverzeichnis erläutert die wichtigsten Begriffe dieses Handbuchs.

Literaturhinweise sind im Text in Kurztiteln angegeben, die in Anführungszeichen stehen. Der vollständige Titel jeder Druckschrift, auf die verwiesen wird, ist im Literaturverzeichnis aufgeführt.

Am Ende des Handbuchs finden Sie ein Stichwortverzeichnis.

Für das Arbeiten mit dem Softwareprodukt HSMS V12.0A im Betriebssystem BS2000 steht Ihnen folgende weitere Dokumentation zur Verfügung:

HSMS V12.0A

Hierarchisches Speicher Management System

Band 2: Anweisungen

Benutzerhandbuch

Ergänzende Produkt-Informationen

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

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Die Handbücher „HSMS V12.0A, Band 1 und 2“ enthalten die Funktionserweiterungen von HSMS V12.0A.

Versions-Backup

In HSMS V12.0A wurde ein neuer Sicherungstyp eingeführt: der Versions-Backup.

Versions-Backup bietet folgende Funktionalität:

1. Sichern und Aufbewahren einer festgelegten Anzahl von Dateiversionen im Versions-Backup-Archiv. Die Anzahl der aufzubewahrenden Dateiversionen wird in den Dateieigenschaften festgelegt; dementsprechend wird das Versionsmanagement nur für im TSOSCAT des Pubsets katalogisierte Dateien durchgeführt.
2. Ein Mechanismus, der gewährleistet, dass gelöschte (bzw. versehentlich gelöschte) Dateien in den zuvor gesicherten Versionen im Archiv lange genug aufbewahrt werden, sodass der Benutzer Maßnahmen zum Restaurieren der Daten ergreifen kann.

Näheres hierzu siehe [Abschnitt "Versions-Backup"](#)

Auftragsverwaltung

HSMS bietet die Möglichkeit, beendete Aufträge auch nach einem Neustart oder Host-Ausfall aufzubewahren. Die Anzahl der Tage, um die beendete Aufträge aufbewahrt werden, kann mit der folgenden Anweisung definiert werden:

```
//MODIFY-HSMS-PARAMETERS OPERATION-CONTROL=*PARAMETERS(KEEP-REQUESTS=...)
```

Näheres hierzu siehe [Abschnitt "Aufträge aufbewahren"](#).

1.4 Darstellungsmittel

- Die Benutzereingaben in den Ablaufbeispielen sind in halbfetter Schrift wiedergegeben.
- Anweisungen im Text, die auf den Aufruf bestimmter Funktionen hinweisen, sind normalerweise so nicht gültig eingebbar, sondern zeigen die im Zusammenhang wichtigen Teile der Anweisung. In der Regel ist die Angabe weiterer Operanden notwendig; außerdem werden die SDF-Datentypen durch andere Angaben ersetzt.
- Literaturhinweise sind im Text durch Kurztitel angegeben, die in Anführungszeichen stehen. Die vollständigen Titel, auf die durch eine Nummer verwiesen wird, sind im Literaturverzeichnis hinter der entsprechenden Nummer zusammen mit einer Kurzbeschreibung aufgeführt.
- Verweise innerhalb dieses Handbuchs geben die betreffende Seite im Handbuch an und je nach Bedarf auch den Abschnitt oder das Kapitel. Verweise auf Themen, die in einem anderen Handbuch beschrieben sind, enthalten nur den Kurztitel dieses Handbuchs. Über das Stichwortverzeichnis können Sie in dem genannten Handbuch dann die entsprechende Stelle im Text finden.

2 Wesentliche Nutzungsmodelle von HSMS

Wegen der stetig wachsenden Datenmengen erfordert die Speicherung, Sicherung und Verwaltung der Daten in einem Rechenzentrum einen zunehmenden Aufwand an Geräten, Datenträgern und nicht zuletzt an Personal.

HSMS (**H**ierarchisches **S**peicher **M**anagement **S**ystem – Hierarchical Storage Management System) ist ein Softwareprodukt

- zur Organisation von Datensicherung und Archivierung
- zur Nutzungsoptimierung von externen Schnellspeichern in einem BS2000-System
- zum Datentransfer

HSMS erleichtert die Datensicherung und unterstützt damit in erster Linie Rechenzentrums- bzw. Systemverwalter. Aber auch dem nicht-privilegierten BS2000-Benutzer stehen Funktionen zum Zugriff auf Systemsicherungen, zur Archivierung von Daten und die Möglichkeit zur Speicherplatzoptimierung, wenn auch eingeschränkt, zur Verfügung.

Mit HSMS können im BS2000 katalogisierte Dateien (DVS) bearbeitet werden, d.h. Dateien von gemeinschaftlichen Datenträgern (SF- und SM-Pubsets), Privatplatten und Net-Storage (Dateien sowohl vom Typ BS2000 als auch NODE-FILE).

Zusätzlich ist das Sichern, Restaurieren und Archivieren von Dateien des lokalen BS2000-UFS (POSIX) und darin gemounteten fernen Knoten möglich. Diese Dateien sind nur über POSIX erreichbar, sie sind also nicht im BS2000 DVS katalogisiert.

Nicht-privilegierte Benutzer können solche Dateien und Jobvariablen anderer Benutzerkennungen sichern und zurückholen, wenn sie Miteigentümer sind (siehe Handbuch „SECOS“ [16]). Die Miteigentümerschaft betrifft in HSMS Dateien, Jobvariablen und Archive. Die Miteigentümerschaft auf ein Archiv ist in HSMS durch Miteigentümerschaft auf das zugehörige Archivverzeichnis realisiert.

HSMS unterstützt große Dateien und Volumes (≥ 32 GB). Dabei ist Folgendes zu beachten:

- Große Dateien können nur auf den dafür vorgesehenen Pubsets restauriert werden.
- Dateien < 32 GB, die sich auf Pubsets mit großen Volumes befinden, werden wie normale Dateien behandelt. Sie werden auf allen Pubsets und auch in allen BS2000-Versionen verarbeitet.

HSMS V12.0 kann auf Anlagen mit BS2000 OSD/BC ab V10.0 eingesetzt werden.

2.1 Grundfunktionen

HSMS bietet dem Benutzer die Grundfunktionen

- Datensicherung (Backup)
- Versions-Backup
- Langzeitarchivierung (Archival)
- Verdrängung (Migration)
- Datentransfer durch Magnetbandkassetten

Für jede Grundfunktion stellt HSMS komfortable Möglichkeiten zur Auswahl der zu verarbeitenden Dateien und zur Steuerung der Verarbeitung bereit. HSMS setzt dabei auf dem Softwareprodukt ARCHIVE auf. Die Funktionen, die bisher über ARCHIVE aufgerufen wurden, stehen auch über HSMS zur Verfügung.

Datensicherung (Backup)

Datensicherung ist das vorbeugende Erstellen, Aufbewahren und Verwalten von Kopien des Datenbestandes in einem Rechenzentrum. Mit diesen Kopien lassen sich Daten nach einem Datenverlust wiederherstellen. Datenverlust kann durch Bedienfehler – wie versehentliches Löschen – oder durch Hardware-Ausfall auftreten.

Datensicherung und anschließende Rekonstruktion kann zum Reorganisieren des Datenbestandes oder zum Umzug von Kennungen auf neue bzw. andere Pubsets verwendet werden. Beim Reorganisieren wird insbesondere die Aufsplitterung von Dateien auf viele Bereiche (Extents) wird behoben.

HSMS sichert ebenso wie ARCHIVE logisch: Die Dateien und Jobvariablen werden von einem oder mehreren Datenträgern gelesen und zusammenhängend, also in logischen Einheiten, auf andere Datenträger geschrieben (siehe dazu den Abschnitt „Datensicherung im BS2000“ im Handbuch „ARCHIVE“ [2]).

HSMS ermöglicht sowohl Systemsicherungen des gesamten Datenbestandes als auch die Sicherung einzelner Anwendungen (z.B. Datenbanken). Wahlweise sichert HSMS alle vom Benutzer angegebenen Dateien (Vollsicherung) oder nur Dateien, die sich seit der letzten Sicherung geändert haben bzw. die neu angelegt wurden (Differenzsicherung), und bei großen Dateien auch nur die geänderten Blöcke (Teilsicherung).

Alle Sicherungsarten sind auf Magnetbandkassetten oder auf Platten möglich. Beispielsweise können mit HSMS Vollsicherungen auf Magnetbandkassetten geschrieben werden, Differenzsicherungen zwischendurch auf Platten. Sicherungen können auch auf Platten zwischengelagert und anschließend zu passenden Zeiten auf Magnetbandkassetten verlagert werden.

Standardmäßig kann eine Datei während ihrer Sicherung nur im Lesemodus geöffnet werden, d.h. Dateiänderungen sind während der Sicherungsphase nicht möglich. Wird bei der Sicherung die Funktion „Concurrent Copy“ aktiviert, können Dateien auch während der Sicherung geändert werden. Diese Funktion ist besonders für Anwendungen gedacht, die überhaupt nicht oder möglichst wenig unterbrochen werden sollen, bei denen aber trotzdem Dateien gesichert werden sollen.

Nach Erstellen einer Sicherung auf Magnetbandkassette kann diese automatisch auf andere Magnetbandkassetten dupliziert werden. Dadurch lässt sich die Datensicherheit erhöhen: Wenn ein Datenträger beschädigt wird oder verloren geht, können mit der noch vorhandenen Kopie die zerstörten Daten rekonstruiert oder die verlorenen Dateien restauriert werden.

HSMS ermöglicht auch den Backup/Restore von Pubset-Kopien, die mit SHC-OSD-Spiegelungsfunktionen erstellt wurden. Dazu wird die Concurrent-Copy-Funktion verwendet.

Nicht-privilegierte Benutzer können Dateien und Jobvariablen einer anderen Benutzerkennung sichern und restaurieren, wenn sie Miteigentümer sind. Nähere Informationen zur Miteigentümerschaft finden Sie im Handbuch „SECOS“ [16]).

Bei der Sicherung von PLAM-Bibliotheken können zusätzliche Informationen über die Elementstruktur gespeichert werden (außer bei migrierten Bibliotheken). Mit Hilfe der gespeicherten Elementenstruktur kann jedes in der Bibliothek enthaltene Element mit der HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS auch einzeln restauriert werden.

Versions-Backup

Der Versions-Backup ist eine Variante des Backup. Hierbei sorgt HSMS dafür, dass eine Anzahl von Versionen einer Datei im Versions-Backup-Archiv vorgehalten werden. Die für eine Datei vorzuhaltende Anzahl Dateiversionen definiert der Benutzer mit dem Datei-Attribut NUM-OF-BACKUP-VERS (einstellbar mit dem BS2000-Kommando MODIFY-FILE-ATTRIBUTES). Im Rahmen des Versions-Backup werden nur Differenzsicherungen durchgeführt.

Langzeitarchivierung (Archival)

Archivierung bezeichnet die langfristige Auslagerung von Dateien, die online auf der Verarbeitungsebene nicht mehr benötigt werden. Für die archivierten Daten können Schutzfristen unabhängig von den Schutzfristen der Datenträger vereinbart werden.

Durch die Archivierung kann Speicherplatz auf der Verarbeitungsebene eingespart werden, indem die Dateien nach der Archivierung gelöscht werden. Dies führt auch zur Entlastung des Katalogs. Daneben dient die Langzeitarchivierung auch der Dokumentation, z.B. wenn Daten wegen rechtlicher Vorschriften über bestimmte Zeiträume hinweg aufbewahrt werden müssen.

Analog zur Sicherung können nach einer Archivierung die Daten automatisch auf andere Magnetbandkassetten dupliziert werden.

Nicht-privilegierte Benutzer können auch Dateien anderer Benutzerkennungen archivieren und restaurieren, wenn sie Miteigentümer sind (siehe Handbuch „SECOS“ [16]).

Verdrängung (Migration)

HSMS bietet mit der Verdrängung die Möglichkeit zu einer Nutzungsoptimierung für schnelle Plattenspeicher. Vor allem bei häufig wechselnden Anwendungen ist es sinnvoll, gerade nicht benötigte (inaktive) Daten von der Verarbeitungsebene auf eine Hintergrundebene zu verdrängen, wobei die Daten aber über ihren Katalogeintrag ansprechbar bleiben. Dadurch lassen sich Sättigungszustände auf gemeinschaftlichen Datenträgern (Pubspace-Saturation) oder das Erreichen des Pubspace-Limits für Benutzerkennungen vermeiden, ohne dass die Verfügbarkeit der Dateien merklich eingeschränkt wird, für Pubsets auch automatisierbar.

Für verdrängte Dateien erzeugt das Datenverwaltungssystem (DVS) bei Zugriffsversuchen HSMS-Aufträge: Die Dateien werden ohne Eingriff des Benutzers wieder auf die Verarbeitungsebene zurückgeholt.

Teure, schnelle Externspeicher lassen sich durch den Verdrängungsmechanismus wirtschaftlicher einsetzen, ohne den personellen Aufwand zu erhöhen. Da nur die tatsächlich belegten Seiten übertragen werden, wird auf der Hintergrundebene gegenüber der Verarbeitungsebene auch Speicherplatz eingespart.

Dateien, die auf den Platten eines Plattenspeichersystems liegen und mit SHC-OSD remote gespiegelt werden, sollten nicht verdrängt werden, wenn der ferne Rechner keinen Zugriff auf die Verarbeitungsebene S2 hat.

Mit der Anwendung SM2 können die Verdrängungs- und Rückhol-Tätigkeiten verfolgt sowie die Migrationsparameter optimal eingestellt werden.

Nicht-privilegierte Benutzer können auch solche Dateien migrieren und zurückholen, bei denen sie Miteigentümer sind (siehe Handbuch „SECOS“ [16]).

i Für Dateien auf Net-Storage, Privatplatte, für Dateien des BS2000-UFS (POSIX) und darin gemounteten fernen Knoten-S0 steht die Grundfunktion Verdrängung nicht zur Verfügung.

Datentransfer (Export/Import)

Mit HSMS können Dateien und Jobvariablen auf andere BS2000-Systeme oder andere Benutzerkennungen übertragen und Datenträger ausgetauscht werden. Dazu werden die Daten auf Magnetbandkassetten geschrieben (exportiert) und auf der gewünschten Anlage bzw. Benutzerkennung wieder eingelesen (importiert). Auf Wunsch kann dabei auch mit Archiv-Verzeichnissen gearbeitet werden.

Außerdem ist es möglich, nur die Katalogeinträge von Dateien zu übertragen.

Übersicht über die HSMS-Grundfunktionen

Datensicherung (BACKUP)	Archivierung (ARCHIVAL)	Verdrängung (MIGRATION)	Datentransfer (EXPORT)
Systemdienst und Benutzerkontrolle (Schutz vor Zerstörung oder Reorganisation)	Benutzerverfahren (Betriebs- und Rechtsvorschriften)	System- und Benutzerkontrolle	Benutzerverfahren
alle Dateien und Jobvariablen	ausgewählte Dateien und Jobvariablen	inaktive Dateien	ausgewählte Dateien und Jobvariablen
kurzfristige Lagerung einer Kopie (z.B. 6 Wochen) Versions-Backup: dauerhafte Aufbewahrung mehrerer Versionen einer Datei	langfristige Lagerung einer Kopie (mit Löschen des Originals)	mittelfristiges Auslagern von Dateien	Auslagern zur Übertragung
RESTAURIEREN (RESTORE)	RESTAURIEREN (RESTORE)	ZURÜCKHOLEN (RECALL)	IMPORTIEREN (IMPORT)
Bei Verlust des Originals oder Reorganisation	Nur unter Ausnahmebedingungen	Für Verarbeitung	Für Verarbeitung
mit HSMS-Anweisung	mit HSMS-Anweisung	mit HSMS-Anweisung und automatisch beim Zugriff	mit HSMS-Anweisung

2.2 Speicherhierarchie

HSMS realisiert die Verdrängung im Rahmen einer dreistufigen Speicherhierarchie, die Multiple-Public-Volume-Sets (MPVS, SF-Pubsets), Shared-SF-Pubsets, SM-Pubsets und Shared-SM-Pubsets unterstützt.

HSMS unterscheidet die drei Speicherebenen S0, S1 und S2, die unter diesem Kürzel auch in den HSMS-Anweisungen und BS2000-Kommandos angesprochen werden.

SF-Pubsets können entsprechend ihrer Eigenschaften, wie Verfügbarkeit, Zugriffszeit und Kosten entweder der Verarbeitungsebene S0 oder der Sicherungs- oder Hintergrundebene S1 zugeordnet werden. Die Sicherungs- und Hintergrundebene S2 wird durch Bandmedien realisiert.

SM-Pubsets enthalten aus HSMS-Sicht bereits intern eine Speicherhierarchie. Sie können aber auch in der SF-Umgebung als S1-Ebene zugeordnet werden (S1-SM-Pubsets).

Net-Storage und Privatplatten kann HSMS zwar sichern und für die Sicherung von Daten verwenden; sie werden aber keiner Speicherebene zugeordnet.

Speicherebene S0

Die Speicherebene S0 ist die normale Verarbeitungsebene. Sie besteht aus gemeinschaftlichen Datenträgern, Pubsets, auf die die Benutzer online zugreifen können.

S0 steht unter der Verwaltung des Datenverwaltungssystems (DVS). Auf S0 werden die Daten erzeugt und bearbeitet, für die HSMS die Grundfunktionen zur Verfügung stellt.

Wenn im System Plattenspeicher mit unterschiedlichen Charakteristika installiert sind, sollten der Ebene S0 vorzugsweise die Plattenspeicher mit schnellem Zugriff, d.h. kurzen Zugriffszeiten, zugeordnet werden.

Speicherebene S1

Die online-verfügbare Hintergrundebene S1 besteht ebenfalls aus Plattenspeichern, die in SF-Pubsets, als S1-SM-Pubsets oder als Volume-Set eines SM-Pubsets organisiert sind. Die Daten auf S1 werden von HSMS verwaltet (letztlich unterliegen aber auch die HSMS-Dateien der Verwaltung des DVS). Auf die Speicherebene S1 können Daten von S0 verdrängt werden.

Der Speicherebene S1 werden vorzugsweise Plattenspeicher mit hoher Kapazität zugeordnet, die längere Zugriffszeiten haben können als die auf S0, auf denen die Daten aber mit geringeren Kosten gehalten werden, wozu auch die komprimierte Ablage beiträgt.

Die Speicherebene S1 kann in der SF-Umgebung systemglobal oder auch spezifisch einzelnen S0-Pubsets zugeordnet werden (siehe [Bild 1](#)). Sie besteht aus einem oder mehreren SF-Pubsets oder S1-SM-Pubsets.

Dagegen kann in einem SM-Pubset ein Volume-Set als S1-Ebene definiert werden. Ab HSMS V11.0 kann die S1-Ebene auch auf alle Volume-Sets des SM-Pubsets, die unter HSMS-Verwaltung stehen, erweitert werden. Diese Erweiterung ist nur ab BS2000 OSD/BC V11.0 möglich. Unabhängig von der Anzahl der zur Verfügung stehenden Volume-Sets ist die S1-Ebene "pubset-lokal" nur in diesem SM-Pubset verfügbar.

Die Speicherebene S1 ist in einem unter HSMS-Verwaltung stehenden System nicht zwingend, da die Hintergrundebene S2 für einen normalen HSMS-Betrieb ausreicht.

Die Speicherebene S1 steht für die Bearbeitung von Knotendateien des BS2000-UFS (POSIX) und dort gemounteten fernen Knoten-S0 nicht zur Verfügung.

Speicherebene S2

Die Speicherebene S2 besteht aus Magnetbandkassetten. Die Aufbewahrungskosten sind erheblich geringer als auf S0 und S1. In der Regel sind aber Operator- oder zumindest Robotereingriffe erforderlich, bevor auf die Daten auf S2 zugegriffen werden kann. Die Zugriffszeiten liegen damit in einer anderen Größenordnung als bei den Speicherebenen S0 und S1.

Auf S2 können Daten verdrängt werden. Das Fortschreiben von Magnetbandkassetten führt zu einer optimalen Ausnutzung der Datenträger.

Die Daten auf S2 werden von HSMS verwaltet. Unter HSMS lassen sich die Zeiten, während denen Datenträger der Speicherebene S2 bedient werden, funktions- und archivspezifisch festlegen.

Übersicht über die HSMS-Speicherebenen

Speicherebene	Datenträger	Zugriff	Zugriffszeit
Verarbeitungsebene S0	Platten	online	sehr kurz
Hintergrundebene S1	Platten	online	kurz
Hintergrundebene S2	Magnetbandkassetten	offline	lang

Die Pubsets einer BS2000-Konfiguration können den unterschiedlichen Speicherebenen zugeordnet werden. Eine Pubset-Umgebung unter HSMS-Verwaltung könnte z.B. so aussehen:

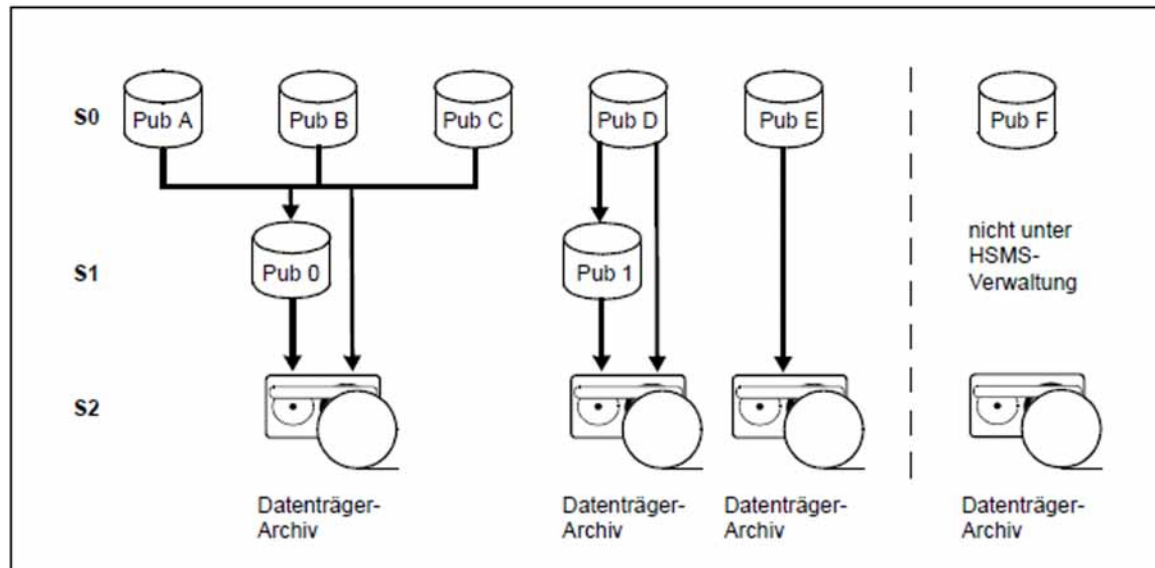


Bild 1: Beispiel eines Multiple-Public-Volume-Sets unter HSMS

In diesem Beispiel ist den Pubsets A, B und C der globale S1-Pubset 0 zugeordnet.

Pubset D ist abweichend davon der Pubset 1 als S1-Pubset zugeordnet. Pubset E hat keinen S1-Pubset; Sicherungen sind hier nur auf S2 möglich.

Jedem SM-Pubset in einer BS2000-Konfiguration kann bzw. können ein oder mehrere Volume-Sets als S1-Speicherebene zugewiesen werden. Wenn der SM-Pubset außerhalb des S1-Volume-Sets beschädigt wird und neu aufgebaut werden muss, können vorhandene volle Volume-Sets in die Rekonfiguration des SM-Pubsets integriert werden.

Daneben kann ein SM-Pubset als Speicherebene S1 innerhalb der SF-Umgebung eingerichtet werden. Dazu wird er als SM-Pubset unter HSMS-Kontrolle genommen und dann wie ein SF-Pubset entweder global oder pubsetspezifisch als Speicherebene S1 definiert. Details hierzu siehe [Abschnitt „SM-Pubset der Speicherebene S1 zuordnen“](#).

2.3 Archive

Archive sind die grundlegenden Verwaltungseinheiten für alle Daten, die von HSMS verwaltet werden. Alle gesicherten, archivierten und verdrängten Daten legt HSMS in Archiven ab, und zwar getrennt nach den Grundfunktionen in

- Backup-Archive für die Datensicherung
- Versions-Backup-Archive für Versions-Backup
- Langzeitarchive für die Langzeitarchivierung
- Migrationsarchive für die Verdrängung
- Schattenarchive für die Kopien von Sicherungsdateien. Die Kopien werden für Datensicherungen und Langzeitarchivierungen automatisch erstellt.

HSMS unterscheidet *öffentliche* Archive, die allen Benutzern zur Verfügung stehen, und *private* Archive, die nur der Archiveigentümer nutzen darf.

Der HSMS-Verwalter kann öffentliche Archive als Standard-Systemarchive zuweisen:

- systemweit
- für spezifische SF-Pubsets
- für SM-Pubsets.

Diese Systemarchive können alle Benutzer für eine der HSMS-Grundfunktionen nutzen. Sie können über symbolische Namen angesprochen werden, und zwar

- SYSBACKUP für das Standard-System-Backup-Archiv
- SYSVERSION für das Standard-System-Versions-Backup-Archiv
- SYSARCHIVE für das Standard-System-Langzeitarchiv
- SYSMIGRATE für das Standard-System-Migrationsarchiv

Die im Archiv verwalteten Daten werden in Sicherungsdateien aufbewahrt, die auf S1-Pubsets, Privatplatten, Magnetbandkassetten stehen können. Um Speicherplatz zu sparen, können die Daten vor dem Schreiben in die Sicherungsdatei komprimiert werden.

Zur Erhöhung der Datensicherheit bietet HSMS eine Funktion, mit der Sicherungsdateien – auch auszugsweise – innerhalb des Archivs und auch außerhalb kopiert werden können.

Zusammenfassung

Der unter HSMS-Verwaltung stehende Datenbestand und die Speichermedien lassen sich unter zwei Gesichtspunkten betrachten:

- Nach der *Verfügbarkeit* der Datenträger werden die Speichermedien in die Speicherebenen S0, S1 und S2 unterteilt.
- Nach der *Nutzung* ist der Datenbestand aufgeteilt in Verwaltungseinheiten, die durch die Standard-Systemarchive und private Archive gebildet werden.

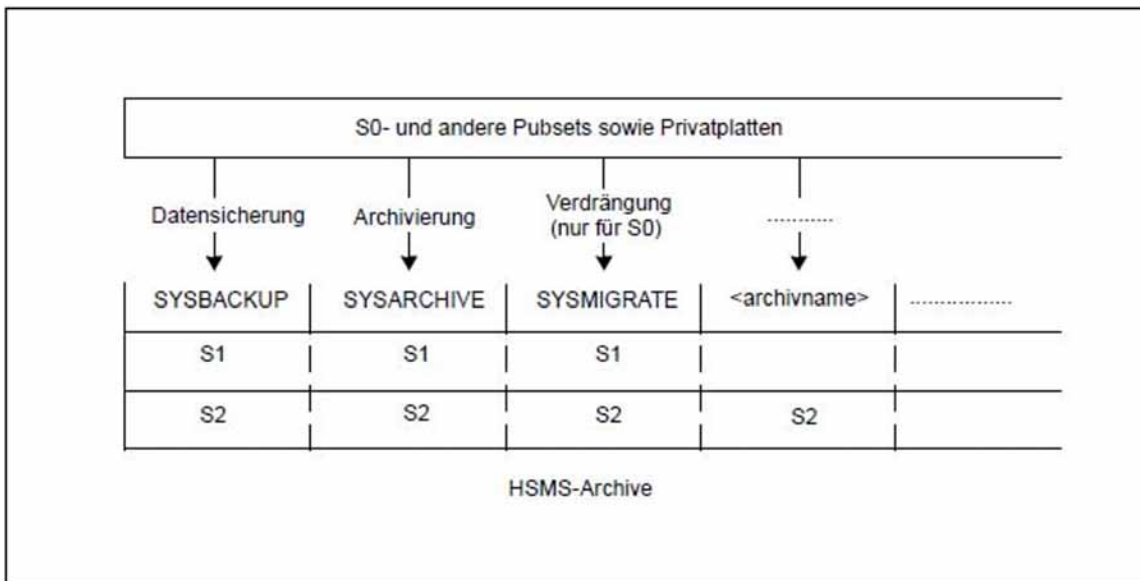


Bild 2: Struktur der Speicherverwaltung von HSMS

Zur Verwaltung der Datenbestände, Speichereinheiten, Archive etc. bietet HSMS eine Reihe von Auskunftsfunktionen. So lässt sich mit HSMS z.B. die Belegung der gemeinschaftlichen Datenträger ermitteln, um eventuellen Sättigungszuständen rechtzeitig vorbeugen zu können.

2.4 Miteigentümerschaft

Ein Objekteigentümer kann über den Miteigentümerschutz festlegen, für welche seiner Objekte er Miteigentümer (Co-Owner) benennen will und welche Zugriffsbedingungen die Miteigentümer bei Verwaltungszugriffen erfüllen müssen. Objekteigentümer ist die Benutzerkennung, die das Objekt einrichtet. Objekte können in HSMS Dateien, Jobvariablen und Archive sein. Die Miteigentümerschaft auf ein Archiv ist in HSMS durch Miteigentümerschaft auf das zugehörige Archivverzeichnis realisiert. Außerdem unterstützt HSMS die Miteigentümerschaft bei Ein- und Ausgabedateien und der überwachenden Jobvariablen.

Der Miteigentümer eines Archivs darf dieses wie sein eigenes benutzen. Ausgenommen sind lediglich Modifikationen der Archivattribute.

Ein Miteigentümer ist eine Benutzerkennung, die ungleich der des Objekteigentümers ist. Sie besitzt aber für ein bestimmtes Objekt dieselben Rechte wie der Objekteigentümer.

Allgemein gilt für Miteigentümer: Alle Lese-, Schreib- und Ausführungszugriffe auf Dateien werden nach den Regeln der traditionellen Dateischutz-Mechanismen kontrolliert:

- Wenn eine Datei über USER-ACCESS, ACCESS oder über BASIC-ACL geschützt ist, hat ein Miteigentümer dieselben Lese-, Schreib- und Ausführungsrechte wie der Dateieigentümer.
- Wenn eine Datei durch Guards geschützt ist, wird der Zugriff durch die Auswertung von Zugriffsbedingungen geregelt, die in STDAC-Guards festgelegt sind.

Miteigentümer können über die BS2000-Kommandos ADD-/MODIFY-COOWNER-PROTECTION-RULE und ADD-/MODIFY-/REMOVE-/SHOW-ACCESS-CONDITIONS festgelegt, geändert bzw. angezeigt werden.

Nähere Informationen zur Miteigentümerschaft finden Sie im Handbuch „SECOS“ [16].

2.5 Benutzerklassen

HSMS unterscheidet in der Nutzung und Steuerung seiner Funktionen drei Klassen von Benutzern: Nicht-privilegierte Benutzer, HSMS-Verwalter und Subsystem-Verwalter. In diesem Handbuch ist jeweils beschrieben, wieweit die einzelnen Funktionen den jeweiligen Benutzerklassen zur Verfügung stehen.

Nicht-privilegierte Benutzer (Privileg „STANDARD PROCESSING“)

Ein nicht-privilegiertes Benutzer darf Dateien und Jobvariablen, deren Eigentümer oder Miteigentümer er ist, in folgende Archive *sichern*:

- in ein öffentliches Systemarchiv (für welches ACCESS=*WRITE gilt)
- in sein eigenes Archiv
- in ein öffentliches Archiv eines anderen Benutzers, wenn die zu sichernden Daten diesem anderen Benutzer gehören
- in ein Archiv eines anderen Besitzers, für dessen zugehöriges Archivverzeichnis er die Miteigentümerschaft besitzt. Dieses Archiv muss nicht öffentlich sein.

Ein nicht-privilegiertes Benutzer darf Dateien *archivieren*, deren Eigentümer oder Miteigentümer er ist. Er darf diese Dateien entweder in ein Systemarchiv archivieren oder in ein Benutzerarchiv, falls er auf dieses Benutzerarchiv zugreifen darf.

Ein nicht-privilegiertes Benutzer darf auch Dateien, deren Eigentümer oder Miteigentümer er ist, in ein öffentliches Systemarchiv *migrieren*. Außerdem ist ihm der *Transfer* von Dateien und Jobvariablen gestattet, deren Eigentümer oder Miteigentümer er ist.

Letztendlich darf ein nicht-privilegiertes Benutzer auch eigene *Archive einrichten*, die privat oder öffentlich sein können.

HSMS-Verwalter (Privileg „HSMS-ADMINISTRATION“)

Der HSMS-Verwalter darf alle Funktionen und Möglichkeiten, die HSMS bietet, uneingeschränkt nutzen. Er ist dadurch gekennzeichnet und privilegiert, dass er unter einer Benutzererkennung arbeitet, die das Privileg „HSMS-ADMINISTRATION“ besitzt. Standardmäßig sind das die Kennungen SYSHSMS und TSOS.

Der HSMS-Verwalter ist zuständig für die Systemsicherung, d.h. die systemweite Sicherung der Daten aller Benutzer. Er hat nach dem Aufruf des Programms HSMS das Recht, auf die Dateien aller Benutzer auf allen Pubsets zuzugreifen. Er ist verantwortlich für die Speicherhierarchie-Verwaltung und die Einrichtung von Standard-Systemarchiven für die HSMS-Grundfunktionen und die Steuerung der Bandzugriffe für HSMS.

Außerdem ist der HSMS-Verwalter für das Einrichten von Schattenarchiven verantwortlich. In Schattenarchiven werden die Kopien von Sicherungsdateien verwaltet, die für Sicherungen und Langzeitarchivierungen automatisch auf Magnetbandkassetten erstellt werden.

Subsystem-Verwalter (Privileg „SUBSYSTEM-MANAGEMENT“)

Der Subsystem-Verwalter darf HSMS starten und beenden.

2.6 HSMS in einer Umgebung von SM-Pubsets

Im BS2000 werden zwei Pubset-Typen unterschieden, die Single-Feature-Pubsets (SF-Pubsets) und die System-Managed-Pubsets (SM-Pubsets), die beide über ihre Katalogkennung angesprochen werden.

Ein SF-Pubset besteht aus einer oder mehreren Platten, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit, Verfügbarkeit) übereinstimmen müssen. Ein SM-Pubset kann im Gegensatz dazu aus mehreren so genannten Volume-Sets mit unterschiedlichen Eigenschaften bestehen. Nur innerhalb eines Volume-Sets müssen die wesentlichen Eigenschaften der Platten übereinstimmen.

Wenn ein Benutzer volume-set-spezifische Eigenschaften für eine Datei auf einem SM-Pubset festlegt, ermittelt das System einen zu diesen Eigenschaften passenden Volume-Set des SM-Pubsets und legt die Datei dort ab. Dadurch ist es insbesondere möglich, eine Datei auf einen unterschiedlich performanten Datenträger innerhalb desselben SM-Pubsets zu verschieben, ohne den Namen der Datei ändern zu müssen.

Nähere Einzelheiten zu SM-Pubsets finden Sie im Handbuch „System Managed Storage“ [18].

Ein SM-Pubset unter HSMS-Kontrolle ist ein SM-Pubset, der von HSMS verwaltet wird. Die HSMS-Metadaten liegen auf dem SM-Pubset selbst. Ein SM-Pubset, der sich nicht unter HSMS-Kontrolle befindet, kann nur sehr eingeschränkt von HSMS bearbeitet werden.

Im Folgenden wird davon ausgegangen, dass sich der betreffende SM-Pubset unter HSMS-Kontrolle befindet. Deshalb wird er der Einfachheit halber nur als SM-Pubset bezeichnet. Wenn ein Missverständnis möglich ist, wird ausdrücklich erwähnt, ob sich der SM-Pubset unter HSMS-Kontrolle befindet oder nicht.

Ein SM-Pubset ist ein geschlossener Behälter, der Daten und Metadaten enthält. Die Dateien befinden sich entweder auf der Verarbeitungsebene S0 oder auf den Hintergrundebenen S1 oder S2, abhängig davon, ob sie aktive oder inaktive Daten enthalten.

Ein SM-Pubset hat folgende Eigenschaften:

- Jeder SM-Pubset, der sich unter HSMS-Kontrolle befindet, enthält genau ein SM-Pubset-Objekt des Datenverwaltungssystems DVS, das die Verarbeitungsebenen S0 und S1 repräsentiert. Ein SM-Pubset besteht aus mehreren Volume-Sets, die zusammen einen Pool bilden. Die obligatorische Verarbeitungsebene S0 wird von einem oder mehreren Volume-Sets gebildet, die nicht von HSMS reserviert sind; auf ihr müssen sich die Dateien befinden, die bearbeitet werden sollen.
- Für Sicherung und Verdrängung von Dateien der Verarbeitungsebene S0 innerhalb des SM-Pubsets werden Sicherungs- bzw. Hintergrundebenen benötigt:
 - S1 enthält auf Platte gesicherte und migrierte Dateien; die S1-Ebene wird entweder durch ein Volume-Set oder durch alle unter HSMS-Verwaltung stehenden Volume-Sets innerhalb des SM-Pubsets gebildet.
 - S2 enthält auf Band gesicherte und migrierte Dateien; die S2-Ebene besteht aus einem Tape-Pool, der durch ein Backup-Archivverzeichnis oder einen MAREN-Pool realisiert wird.
- Um die Dateien auf den verschiedenen Ebenen verwalten zu können, wird die SM-Pubset-Umgebung durch Metadaten beschrieben. Die Metadaten liegen auch innerhalb des Pools auf einem besonders gekennzeichneten Volume-Set, dem so genannten Control-Volume-Set. Solche Metadaten sind beispielsweise Daten zur Beschreibung der Volume-Set-Konfiguration, Benutzerkatalog, Guards-Katalog, Verweise auf Dateikataloge, Kataloge für spezielle Objekte wie Jobvariablen, migrierte Dateien, Dateien auf privaten Datenträgern usw. Die Metadaten werden – abhängig davon, ob sie anlagen- oder pubset-spezifisch sind – entweder in HSMS-Dateien auf dem SM-Pubset abgelegt oder in HSMS-Dateien auf dem Default-Pubset von SYSHSMS.

Um einen SM-Pubset als „geschlossenen Behälter“ bearbeiten zu können, wurde der Begriff „Umgebung“ in HSMS eingeführt. Eine Umgebung enthält Daten und Metadaten. Ein SM-Pubset unter HSMS-Kontrolle ist per Definition eine Umgebung. Ein anderer SM-Pubset ist auch eine andere Umgebung.

Die in einem SM-Pubset gespeicherten Daten und Metadaten betreffen nur Objekte dieses SM-Pubsets. Wenn beim Sichern/Restaurieren und Migrieren/Zurückholen auf ein Archiv Bezug genommen wird, das in einem SM-Pubset definiert ist, dann können nur Dateien und Jobvariablen bearbeitet werden, die zu diesem SM-Pubset gehören. Trotzdem ist es immer noch möglich, Dateien, die in einer bestimmten Umgebung gesichert sind, auf eine andere Umgebung zu restaurieren.

Eine Ausnahme stellen das Exportieren und Importieren dar. Exportieren und Importieren sind Funktionen, um Dateien von einer Konfiguration auf eine andere zu bringen. Deshalb handelt es sich hierbei um globale Funktionen, d.h. sie können nicht einer speziellen SM-Pubset-Umgebung zugeordnet werden. Diese Funktionen werden immer in einer SF-Umgebung bearbeitet, ohne Einschränkung für die zu bearbeitenden Daten. Die Metadaten (Aufträge, Steuerdatei, ...) werden physikalisch in der SF-Umgebung aufbewahrt, während die zugehörigen Daten in einer beliebigen Umgebung sein können.

Archivieren und Restaurieren sind Langzeitarchiv-Funktionen, bei denen sich in der Zwischenzeit die Konfiguration der Umgebung geändert haben kann. Archivieren und Restaurieren sind entweder einer Rechenzentrums-Konfiguration zugeordnet oder nicht. Deshalb kann der Umfang dieser Tätigkeiten global oder lokal sein, d.h. sie können einer SF- oder einer SM-Umgebung zugewiesen sein. Die Metadaten (Archive, Aufträge, Steuerdatei, ...) werden physikalisch in der entsprechenden Umgebung aufbewahrt, während die zugehörigen Daten in einer beliebigen Umgebung sein können.

Alle Tätigkeiten auf einem SM-Pubset können sogar nach dem Import des SM-Pubsets auf einem anderen Rechner oder auf demselben Rechner mit einem anderen Home-Pubset durchgeführt werden. Das erneute Starten von unterbrochenen Aufträgen ist ebenfalls möglich.

Die gesamte Menge der SF-Pubsets wird als eine einzige Umgebung betrachtet. Die Daten und Metadaten, die sich auf SF-Pubsets befinden, können von HSMS verwaltet werden.

2.7 BS2000 Backup-Server in Shared-Pubset-Umgebung

Für die Bearbeitung von HSMS-Aufträgen kann im SPVS-Verbund explizit ein Backup-Server vereinbart werden. Die Bearbeitung erfolgt auf dem Backup-Server, unabhängig davon, ob der Server Master- oder Slave-Host des Shared-Pubsets ist. Dies dient dem Ziel, Produktivsysteme zu entlasten.

Details zur Konfiguration und Arbeitsweise finden Sie im [Abschnitt „Backup-Server“](#).

2.8 Sicherung ferner Rechner mit HSMS

Durch den raschen Wandel von einer zentralen Datenverarbeitung hin zu einer unternehmensweit verteilten Informationstechnologie hat die Anzahl der fernen Rechner – vor allem Workstations und PCs – stark zugenommen. Diese fernen Rechner sind inzwischen sehr leistungsfähig. Auf ihren Festplatten können große Datenmengen gespeichert werden.

Mit HSMS können Dateien von vernetzten fernen UNIX-Rechnern ohne Zwischenablage im BS2000 gesichert, archiviert und bei Bedarf rekonstruiert werden. Die Migration und der Import/Export werden für ferne Rechner nicht unterstützt.

Ein Dateisystem einer UNIX-Workstation oder Teile davon können per NFS im BS2000-UFS (POSIX) gemountet werden, wenn auf dem UNIX-Client ein NFS-Server installiert ist und das entsprechende Verzeichnis exportiert wird. Über diesen Weg erfolgt der Zugriff von HSMS auf die zu sichernden bzw. zu restaurierenden Dateien des fernen UNIX-Systems. Das ferne UNIX ist aus Sicht von HSMS ein passiver Knoten, d.h. zur Durchführung der Sicherung wird keine eigene Sicherungssoftware auf dem UNIX benötigt.

2.8.1 Sicherung von UNIX-Workstations

Dateien, die sich auf solchen Workstations befinden, sind in UNIX-Dateisystemen (UFS) organisiert. Ein UNIX-Dateisystem ist hierarchisch aufgebaut und besteht aus Dateien und Dateiverzeichnissen. An der Spitze der Hierarchie steht das Dateiverzeichnis *root*, das durch einen Schrägstrich (/) gekennzeichnet ist. Von hier aus setzt sich die Verzeichnisstruktur weiter nach unten fort. Von Dateiverzeichnissen kann in weitere Dateiverzeichnisse oder in Dateien verzweigt werden. Eine Datei ist der tiefste Verzweigungspunkt. Von einer Datei aus ist keine Verzweigung mehr möglich.

Dateiverzeichnisse werden auch als Knotenpunkte eines Dateisystems bezeichnet, in denen Namen von Dateien oder weiteren Dateiverzeichnissen stehen. Die Namen für die Dateien und Dateiverzeichnisse kann der Benutzer vergeben, wobei bestimmte Konventionen einzuhalten sind.

Nähere Informationen zum UNIX-Dateisystem finden Sie im Handbuch „NFS“ [11].

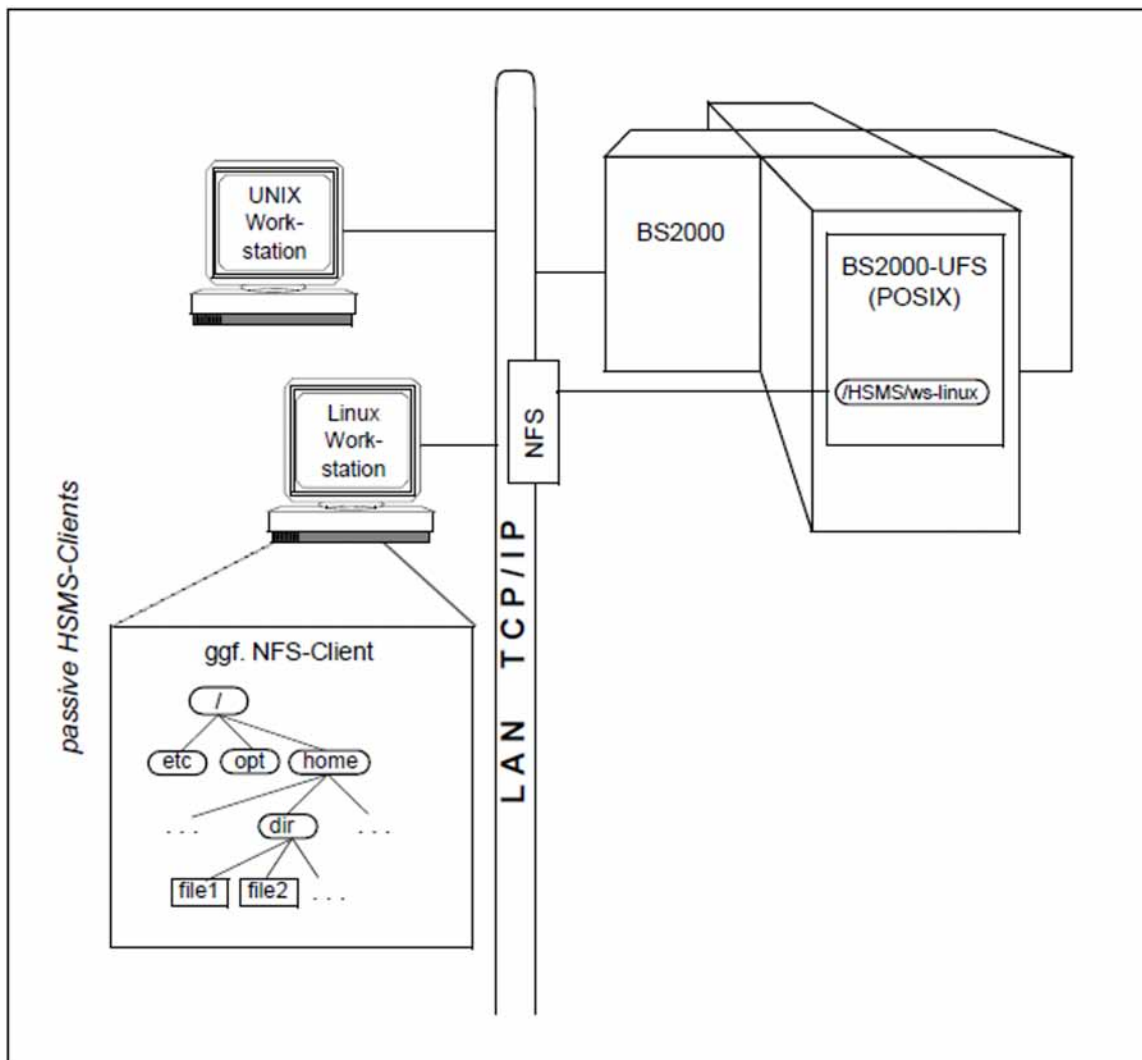


Bild 3: Sicherung ferner Rechner über POSIX-Mount

2.8.2 Speicherhierarchie „Knoten-S0“

Da HSMS die Migration von Knoten-Dateien über POSIX nicht unterstützt, gibt es nur eine einzige Speicherebene. Diese entspricht der Speicherebene S0 bei HSMS und wird im Folgenden „Knoten-S0“ genannt.

Die Ebene Knoten-S0 enthält eine Reihe von Knoten-S0. Jeder Knoten-S0 ist nach dem fernen Rechner benannt, an dem ein Dateisystem normalerweise bearbeitet wird. Nur der Knoten-S0, der dem lokalen BS2000-UFS entspricht, hat keinen Namen.

Jeder Knoten-S0 enthält ein oder mehrere Dateisysteme, entsprechend den Montierprozessen, die der BS2000-Systemverwalter aktiviert hat.

Der Zugriff auf den Knoten-S0 und seine Dateien, die so genannten Knotendateien, ist durch eine Definition innerhalb der Baumstruktur des lokalen BS2000-UFS möglich:

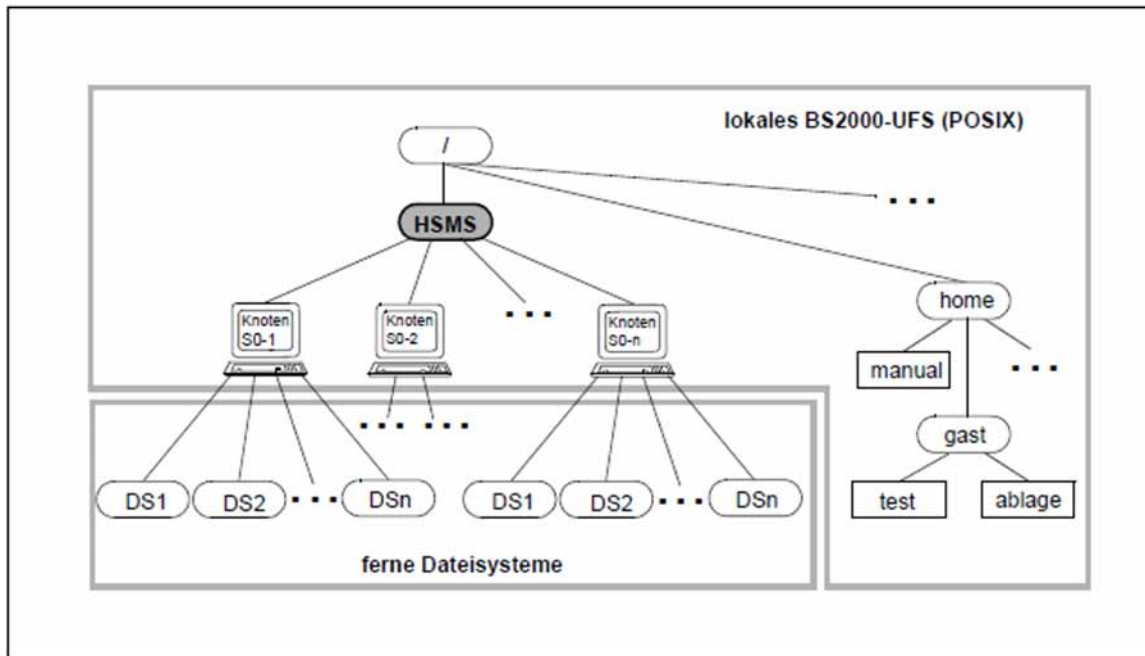


Bild 4: Passiver Knoten-S0 Hinweis

Hinweis

Mit „fernem Dateisystem“ wird bei HSMS ein Dateisystem mit folgenden Eigenschaften bezeichnet:

- Es ist unter dem Dateiverzeichnis „HSMS“ eingehängt.
- Es befindet sich unter einem festgelegten Knoten-S0.
- Es ist vom Root-Verzeichnis des BS2000 aus erreichbar ist.
- Es erlaubt den Zugriff durch das BS2000.
- Es ist im lokalen BS2000-UFS eingehängt.

2.8.3 Archive

Bei der Definition von Archiven sind zwei Fälle zu unterscheiden:

- **Bearbeitung von Dateien auf dem lokalen BS2000-UFS**

Jeder Benutzer kann Archive definieren, um seine eigenen Dateien bearbeiten zu können. Der HSMS-Verwalter muss öffentliche Archive definieren, auf die mit den symbolischen Namen SYSNODEBACKUP und SYSNODEARCHIVE zugegriffen werden kann.

- **Bearbeitung von Dateien auf fernen (passiven) Knoten-S0**

Der HSMS-Verwalter muss Archive entweder auf der Systemebene definieren (ein Archiv für alle Knoten-S0) oder auf der Ebene Knoten-S0 (ein Archiv für jeden Knoten-S0). Für welche der beiden Möglichkeiten sich der HSMS-Verwalter entscheidet, sollte davon abhängen, wie er seine Sicherung und Archivierung organisiert hat.

Der HSMS-Verwalter kann für jedes Archiv, das für die Sicherung und Langzeitarchivierung von Knotendateien verwendet wird, ein Schattenarchiv einrichten.

2.8.4 Benutzerklassen

HSMS unterscheidet in der Nutzung und Steuerung seiner Funktionen folgende Klassen von Benutzern. In diesem Handbuch ist jeweils beschrieben, wieweit die einzelnen Funktionen den jeweiligen Benutzerklassen zur Verfügung stehen.

HSMS-Verwalter (Privileg „HSMS-ADMINISTRATION“)

Der HSMS-Verwalter kann alle Dateien bearbeiten, die auf dem lokalen BS2000-UFS oder auf einem fernen Dateisystem liegen. Er muss die öffentlichen Archive für das lokale BS2000-UFS einrichten sowie seine Arbeitsarchive für die fernen Dateisysteme.

BS2000-Systemverwalter (Privileg „TSOS“ und „Root-Berechtigung“)

Der BS2000-Systemverwalter muss jedes ferne Dateisystem, das bearbeitet werden soll, am Knoten „HSMS“ des lokalen BS2000-UFS einhängen. Zum Einhängen benötigt er die Root-Berechtigung für den Rechner, an dem dieses ferne Dateisystem normalerweise bearbeitet wird. Die Root-Berechtigung wird im BS2000 durch die Vergabe der Benutzernummer 0 erteilt, die standardmäßig der Benutzerkennung SYSROOT zugeordnet ist.

Bei der Initialisierung muss der BS2000-Systemverwalter das Dateiverzeichnis „HSMS“ im lokalen BS2000-UFS anlegen.

Wenn das ferne Dateisystem eines neuen Rechners im lokalen BS2000-UFS eingehängt werden soll, muss der BS2000-Systemverwalter vorher für diesen Rechner den Knoten-S0 im Dateiverzeichnis „HSMS“ erzeugen (Kommando *mkdir*).

2.8.5 Unterstützung von Knoten-Pfadnamen

HSMS unterstützt für Knotendateien nur POSIX-Pfadnamen. Die Syntax der POSIX-Pfadnamen wird von HSMS überprüft. Dies gilt auch für Pfadnamen, die einer Liste (Operand PATH-NAMES=*SELECTED) oder einer Datei (Operand PATH-NAMES=*FROM-FILE) entnommen werden sowie für Aufträge von fernen Workstations.

Es muss immer auf die Groß-/Kleinschreibung geachtet werden, wenn POSIX-Dateinamen und POSIX-Pfadnamen eingegeben werden.

2.9 Anweisungseingabe über SDF

HSMS wird über eine Anweisungsschnittstelle bedient. Alle Eingaben an HSMS erfolgen über das Programm HSMS.

Die Anweisungen an HSMS verarbeitet der Kommandoprozessor SDF (System Dialog Facility). HSMS bietet damit verschiedene Formen des geführten oder ungeführten Dialogs mit der Möglichkeit, Hilfemenüs zu den HSMS-Anweisungen anzufordern und bei fehlerhaften HSMS-Anweisungen einen Korrekturdialog zu führen. Näheres dazu finden sie im Handbuch „Dialogschnittstelle (SDF)“ [5].

3 HSMS-Archive

Das Archiv ist die grundlegende Verwaltungseinheit in HSMS. In Archiven werden alle gesicherten, archivierten und verdrängten Daten (Dateien und Jobvariablen) von HSMS abgelegt und verwaltet.

Ein Archiv besteht aus:

- der Archivdefinition, die in einer Steuerdatei hinterlegt wird und in der die Attribute des Archivs festgelegt sind.
- dem Archivverzeichnis, in dem die im Archiv gesicherten Dateien, Jobvariablen und Datenträger verwaltet werden (realisiert durch eine ARCHIVE-Directory-Datei).
- den Datenträgern und Sicherungsdateien, die die gesicherten Daten enthalten.

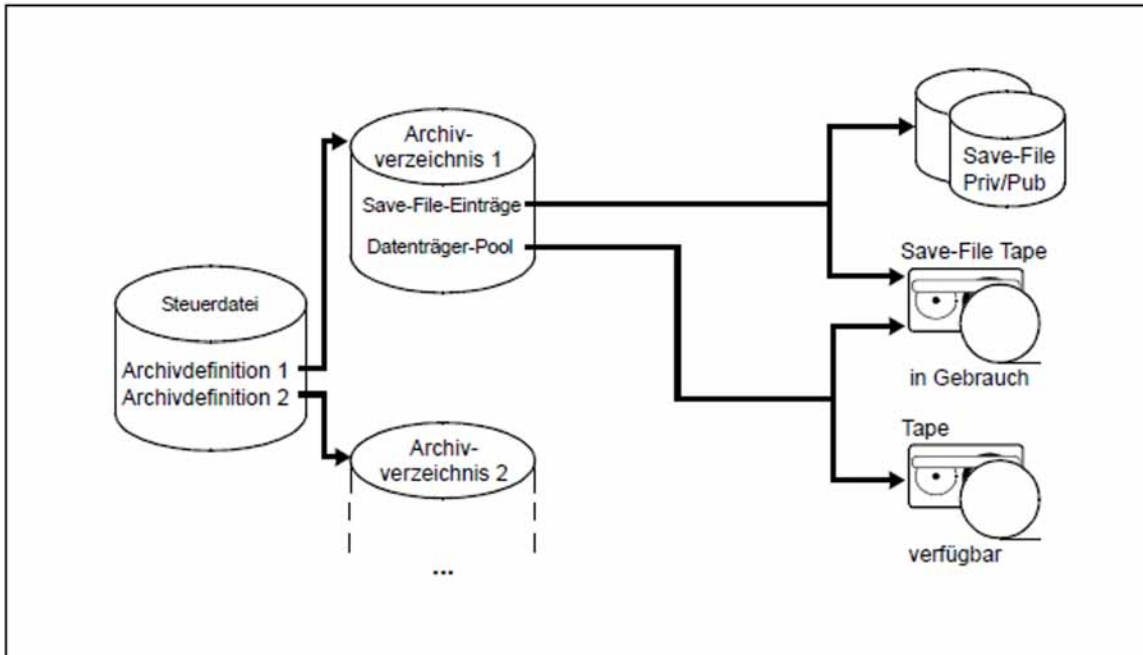


Bild 5: Aufbau eines HSMS-Archivs

Bei der Bearbeitung von *BS2000-Dateien* unterscheidet HSMS einen Archivtyp für jede der drei HSMS-Grundfunktionen Datensicherung, Langzeitarchivierung und Verdrängung sowie den Archivtyp „Schattenarchiv“. In einem Schattenarchiv werden die Kopien von Sicherungsdateien verwaltet, die für Sicherungen und Langzeitarchivierungen automatisch auf Magnetbandkassetten erstellt werden.

Bei der Bearbeitung von *Knotendateien* unterscheidet HSMS zwei weitere Archivtypen: ein Archivtyp wird für die Datensicherung und einer für die Langzeitarchivierung benötigt.

Bevor HSMS eine dieser Grundfunktionen ausführen kann, muss vorher ein Archiv für die entsprechende Funktion eingerichtet werden. Alle HSMS-Anweisungen für diese Grundfunktionen beziehen sich auf bereits eingerichtete Archive.

Ein nicht privilegierter Benutzer kann nur seine eigenen Archive verwalten. Er kann auf öffentliche Archive gemäß der eingestellten Zugriffsberechtigungen (USER-ACCESS) zugreifen, wobei öffentliche Backup-Archive unter SYSHSMS eingerichtet sein müssen.

Bei CREATE-ARCHIVE durch einen nicht-privilegierten Benutzer wird die Kennung des Aufrufers Eigentümer des Archivs.

Bei CREATE-ARCHIVE durch einen HSMS-Verwalter wird die Kennung SYSHSMS Eigentümer des Archivs, außer wenn im Archivnamen eine andere Kennung angegeben wurde, die dann Eigentümer wird.

Der Eigentümer ist für die Verwaltung des Archivs verantwortlich.

Schatten-, Migrations-, und Versions-Backup-Archive darf nur ein HSMS-Verwalter einrichten.

Ein nicht-privilegiertes Benutzer kann Auskunft über die in einem öffentlichen Archiv verwalteten Daten seiner Benutzerkennung erhalten. Wenn ein Archiveigentümer den Zugriff zu seinem Archiv/Directory durch Miteigentümerschaft (SECOS) erlaubt, dann kann jeder Miteigentümer dieses Archiv wie sein eigenes benutzen. Die Miteigentümer können das Archiv aber nicht verwalten, z.B. können sie keine Sicherungsdateien freigeben oder Retention-Periods ändern.

Datentransfer

Exportierte Daten verwaltet HSMS nicht in Archiven; schließlich sollen die exportierten Daten an anderer Stelle verwendet werden. Es ist aber möglich, Informationen über die exportierten Daten in einem Verzeichnis zu führen, das im Aufbau mit einem Archivverzeichnis identisch ist.

Die exportierten Daten werden in Sicherungsdateien geschrieben, die wie die Sicherungsdateien der anderen HSMS-Grundfunktionen behandelt werden können. Insofern gelten die Abschnitte über die Sicherungsdateien und -versionen in diesem Kapitel auch für den Datentransfer.

3.1 Archivdefinition

Ein Archiv wird mit der HSMS-Anweisung CREATE-ARCHIVE eingerichtet. Die Definition des Archivs mit all seinen Attributen wird in einer HSMS-Steuerdatei hinterlegt, darunter auch der Name des Archivverzeichnisses.

Die HSMS-Steuerdatei hängt von der Umgebung ab, in der das Archiv definiert ist:

- Für eine SF-Pubset-Umgebung ist es die zentrale HSMS-Steuerdatei.
- Für eine SM-Pubset-Umgebung ist es die Steuerdatei des SM-Pubsets.

Beim Einrichten muss dem Archiv ein Name gegeben werden. Der Archivname hat das Format eines Dateinamens; er darf aber nur bis zu 12 Zeichen lang sein. Zusätzlich zum Namen wird dem Archiv eine Eigentümerkennung (Owner-Id) zugeordnet.

Der HSMS-Verwalter kann eine Eigentümerkennung ausdrücklich angeben. Wenn er keine angibt, ist SYSHSMS die Eigentümerkennung, egal unter welcher Benutzerkennung er arbeitet.

Das Archiv kann in HSMS-Anweisungen über die Eigentümerkennung und den Archivnamen angesprochen werden in der Form:

`[$owner-id.]archivname`

Archive, in denen Shared-SF-Pubsets verwaltet werden, sollten so eingerichtet werden, dass ein eigenes Archiv pro Shared-Pubset angelegt wird. Das Archiv muss mit einer Directory-Datei auf dem jeweiligen Shared-Pubset und gleichen Archivdefinitionen auf allen Sharern eingerichtet werden. Aufträge für einen im Slave-Modus importierten Shared-Pubset können nämlich nur bearbeitet werden, wenn während dieses Auftrages nur Pubsets betroffen sind, denen derselbe Master-Rechner zugeordnet ist.

Bei Shared-SM-Pubsets werden die Directory-Dateien für Backup, Migration und Langzeitarchivierung in einer SM-Umgebung automatisch auf dem SM-Pubset selbst abgelegt. Bei SM-Pubsets gewährleistet HSMS dadurch automatisch die Kohärenz der Archivdefinitionen für Backup und Migration.

Mit der HSMS-Anweisung CREATE-ARCHIVE werden auch die Attribute des Archivs festgelegt. Sie lassen sich nach der Einrichtung noch mit der HSMS-Anweisung MODIFY-ARCHIVE-ATTRIBUTES ändern.

Auskunft über die Attribute eines Archivs erhält der Archiveigentümer mit der HSMS-Anweisung SHOW-ARCHIVE-ATTRIBUTES.

Ab HSMS V8.0 werden die Archivattribute bei CREATE-ARCHIVE bzw. MODIFY-ARCHIVE-ATTRIBUTES zusätzlich in der Directory-Datei gespeichert. Bei Verlust der Archivdefinition können die Attribute mit CREATE-ARCHIVE aus der Directory-Datei rekonstruiert werden (siehe „Versehentlich gelöschte Archivdefinition wiederherstellen“, „[Archivdefinition löschen](#)“).

Mit MODIFY-ARCHIVE-ATTRIBUTES kann die Katalogkennung des Archivverzeichnisses entsprechend geändert werden, wenn der Pubset umbenannt wurde.

3.1.1 Nutzung des Archivs (Archivtyp)

Wenn ein Archiv eingerichtet wird, muss mit dem Operanden ALLOWED-USAGE der HSMS-Anweisung CREATE-ARCHIVE der Archivtyp festgelegt werden, d.h. das Archiv muss einer der HSMS-Grundfunktionen zugeordnet werden. HSMS unterscheidet:

- Backup-Archive für die Datensicherung (ALLOWED-USAGE=*BACKUP)
- Versions-Backup-Archive für die Datensicherung im Rahmen des Versions-Backups (ALLOWED-USAGE=*VERSIONBACKUP)
- Langzeitarchive für die Langzeitarchivierung (ALLOWED-USAGE=*ARCHIVAL)
- Migrationsarchive für die Verdrängung (ALLOWED-USAGE=*MIGRATION)
- Backup-Archive für die Sicherung von Knotendateien des POSIX-Dateisystems oder ferner Knoten-S0 (ALLOWED-USAGE=*NODEBACKUP)
- Langzeitarchive für die Archivierung von Knotendateien des POSIX-Dateisystems oder ferner Knoten-S0 (ALLOWED-USAGE=*NODEARCHIVAL).
- Schattenarchive für die Kopien von Sicherungsdateien. Schattenarchive können nur für Backup- und Langzeitarchive eingerichtet werden. (ALLOWED-USAGE=*SHADOW).

Migrations- und Schattenarchive darf nur der HSMS-Verwalter einrichten. Alle anderen Archivtypen können auch nicht-privilegierte Benutzer einrichten.

Jedes Archiv kann grundsätzlich nur für die Funktion genutzt werden, für die es eingerichtet wurde. Beispielsweise ist es nicht möglich, Dateien in Backup-Archive zu verdrängen, da Verdrängung nur in Migrationsarchive erlaubt ist.

Schattenarchive können immer nur einem einzigen Backup- oder Langzeitarchiv mit SEVERAL-SVID-Struktur zugewiesen werden. Da diese Zuweisung bereits beim Einrichten des Schattenarchivs erfolgen muss, muss das zuzuordnende Backup- oder Langzeitarchiv bereits vorhanden sein.

Ein Schattenarchiv kann für folgende Zwecke verwendet werden:

- automatisches oder explizites Kopieren von Sicherungsdateien aus einem Backup- oder Langzeitarchiv in das zugehörige Schattenarchiv
- explizites Kopieren von Sicherungsdateien aus einem Schattenarchiv in das zugehörige Backup- oder Langzeitarchiv
- Restaurieren von Daten

Die Verbindung zwischen einem Archiv und dem zugehörigen Schattenarchiv wird aufgehoben, indem das Schattenarchiv gelöscht wird. Neben diesem vollständigen Löschen des Schattenarchivs werden auch folgende Funktionen angeboten:

- **Abtrennen des Schattenarchivs:** Die Verbindung zwischen einem Archiv und dem zugehörigen Schattenarchiv wird aufgehoben und das Schattenarchiv wird zu einem Backup- oder Langzeit-Archiv gemacht. Die Attribute des ursprünglich zugeordneten Archivs werden dabei für das neue Archiv übernommen.
- **Ersetzen des Archivverzeichnisses:** Die Definition des Schattenarchivs wird gelöscht und das Archivverzeichnis des Schattenarchivs wird für das zugeordnete Backup- oder Langzeit-Archiv übernommen. Das Original-Archivverzeichnis bleibt ohne Verknüpfung zu einem Archiv erhalten.

3.1.2 Zugriffsrechte

Die Zugriffsrechte auf ein Archiv kann der Archiveigentümer unterschiedlich regeln. Standardmäßig wird ein Archiv nicht mehrbenutzbar angelegt, d.h. nur der Archiveigentümer darf es zur Sicherung von Daten benutzen; es gilt als *privates* Archiv.

Alle Benutzer haben das Recht, ein Archiv einzurichten, das außer vom Eigentümer auch von anderen Benutzern für die vorgesehene Grundfunktion genutzt werden darf. Es gilt als *öffentliches* Archiv:

```
//CREATE-ARCHIVE USER-ACCESS=*ALL-USERS
```

Öffentliche Backup-Archive von nicht-privilegierten Benutzern können von anderen Benutzern aber nur im Rahmen der Miteigentümerschaft benutzt werden. Ein öffentliches Backup-Archiv steht nur dann allen Nutzern zur Verfügung, wenn SYSHSMS die Eigentümerkennung des Archivs ist (siehe [Abschnitt „Standard-Systemarchive“](#)). Für Langzeitarchive besteht diese Einschränkung dagegen nicht.

Mit ACCESS=*READ müssen öffentliche Archive eingerichtet werden, aus denen Benutzer Daten ihrer Benutzerkennung restaurieren dürfen. Sollen Benutzer Dateien in ein öffentliches Archiv sichern können, muss das Archiv mit ACCESS=*WRITE eingerichtet werden.

Eine Miteigentümerschaft für ein Archiv wird definiert über die Miteigentümerschaft für die entsprechende Verzeichnisdatei. Jeder Miteigentümer kann dieses Archiv wie sein eigenes benutzen, allerdings kann er die Eigenschaften des Archivs nicht verändern, Sicherungsdateien nicht freigeben und deren RETPD nicht ändern. Der Zugriff durch Miteigentümerschaft hängt nicht von den angegebenen Zugriffsattributen dieses Archivs ab.

Die Mehrbenutzbarkeit eines Archivs durch HSMS ist unabhängig von den Dateiattributen des Archivverzeichnisses. Sie ist vom Eintrag in der HSMS-Steuerdatei abhängig – mit Ausnahme der Miteigentümerschaft (siehe oben). Die Mehrbenutzbarkeit lässt sich durch BS2000-Kommandos nicht beeinflussen.

Die Zugriffsberechtigung auf ein Schattenarchiv beinhaltet nur das explizite Kopieren von Sicherungsdateien aus einem Backup- oder Langzeitarchiv ins zugehörige Schattenarchiv und umgekehrt und das Restaurieren von Daten aus dem Schattenarchiv. Die Berechtigung zum automatischen Duplizieren in ein Schattenarchiv hängt ganz allein von der Zugriffsberechtigung des zugehörigen Backup- oder Langzeitarchivs ab.

Wenn ein nicht-privilegiertes Benutzer eine Langzeitarchivierung in ein öffentliches Archiv durchführt, dem ein Schattenarchiv zugeordnet ist, dann werden die Daten automatisch in das Schattenarchiv dupliziert, selbst wenn das Schattenarchiv privat ist (ACCESS= *OWNER-ONLY).

Die folgende Übersicht verdeutlicht den Zugriff auf fremde Archive, wobei auch die Miteigentümerschaft einbezogen ist (in Verbindung mit SECOS).

Voraussetzung ist, dass die jeweiligen Zielarchive entsprechend eingerichtet sind (mit USER-ACCESS=*ALL-USERS und ACCESS=*WRITE oder Miteigentümer des Archivs/Directory), um den Zugriff zu ermöglichen.

Voraussetzung für die Sicherung eigener Dateien in fremde Archive ist die Miteigentümerschaft für das zugehörige Archivverzeichnis, es sei denn das Backup-Archiv ist unter der Kennung SYSHSMS angelegt.

Sicherung von Benutzer A:	in Archiv auf Kennung A	in Archiv auf Kennung B	in SYSBACKUP
Eigene Datei auf Kennung A	ja	ja	nein
Datei auf Kennung B, für die Benutzer A Miteigentümer ist	ja	ja	nein
Gemeinsam benutzbare Datei auf Kennung B (keine Miteigentümerschaft)	nein	nein	nein

ja: Zugriff erlaubt

nein: Zugriff nicht erlaubt

Archivierung von Benutzer A:	in Archiv auf Kennung A	in Archiv auf Kennung B	in SYSARCHIVE
Eigene Datei auf Kennung A	ja	ja	ja
Datei auf Kennung B, für die Benutzer A Miteigentümer ist	ja	ja	ja
Gemeinsam benutzbare Datei auf Kennung B (keine Miteigentümerschaft)	nein	nein	nein

ja: Zugriff erlaubt

nein: Zugriff nicht erlaubt

Migration von Benutzer A:	in Archiv auf Kennung A	in Archiv auf Kennung B	in SYSMIGRATE
Eigene Datei auf Kennung A	nein	nein	ja
Datei auf Kennung B, für die Benutzer A Miteigentümer ist	nein	nein	ja
Gemeinsam benutzbare Datei auf Kennung B (keine Miteigentümerschaft)	nein	nein	nein

ja: Zugriff erlaubt

nein: Zugriff nicht erlaubt

Versions-Backup von Benutzer A:	In Archiv auf Kennung A	in Archiv auf Kennung B	in SYSVERSION
Eigene Datei auf Kennung A	nein	nein	ja
Datei auf Kennung B, für die Benutzer A Miteigentümer ist	nein	nein	ja
Gemeinsam benutzbare Datei auf Kennung B (keine Miteigentümerschaft)	nein	nein	nein

ja: Zugriff erlaubt

nein: Zugriff nicht erlaubt

3.1.3 Monitoring für ein Archiv einstellen

Das Attribut MONITORING steuert archivspezifisch die Überwachung von Aufträgen im BS2000 Backup Monitor am SE Server:

- MONITORING=*NO legt fest, dass Aufträge zu diesem Archiv nicht im BS2000 Backup Monitor angezeigt werden. Dies ist die Default-Einstellung beim Einrichten des Archivs.
- MONITORING=*STD legt fest, dass für das Monitoring die aktuelle Einstellung des globalen HSMS-Parameters MONITORING ausgewertet wird.

Zur Auswirkung der globalen Einstellungen in Kombination mit der archivspezifischen Einstellung siehe [Abschnitt „Zentrale Auftragsüberwachung am SE Server“](#).

3.1.4 Backup-Server für ein Archiv einstellen

Das Attribut BACKUP-SERVER-USAGE steuert archivspezifisch, in welchem System die Sicherungsaufträge des lokalen Systems durchgeführt werden:

- BACKUP-SERVER-USAGE=*NO legt fest, dass Sicherungsaufträge des lokalen Systems zu diesem Archiv an dem System ausgeführt werden, das aktuell Master-Share des Shared-Pubsets ist. Dies ist die Default-Einstellung beim Einrichten des Archivs.
- BACKUP-SERVER-USAGE=*STD legt fest, dass für die Durchführung von Sicherungsaufträgen des lokalen Systems zu diesem Archiv die aktuelle Einstellung im HSMS-Parameter BACKUP-SERVER ausgewertet werden soll.

Weitere Details zur Einstellung des Backup-Servers siehe [Abschnitt „Arbeiten mit Shared-Pubsets“](#).

3.2 Archivverzeichnis

Das Archivverzeichnis enthält Informationen über die im Archiv verwalteten Sicherungsbestände:

- BS2000-Dateien, Jobvariablen oder Knotendateien
- Sicherungsdateien
- Sicherungsversionen
- belegte und freie Sicherungsdatenträger

Der Name des Archivverzeichnisses wird beim Einrichten des Archivs mit der HSMS-Anweisung CREATE-ARCHIVE mit DIRECTORY-NAME festgelegt. Die Begriffe Archivverzeichnis und Directory-Datei werden synonym verwendet.

Standardmäßig wird das Archivverzeichnis bei der Einrichtung des Archivs von HSMS neu angelegt, wobei der Name frei wählbar ist. Es ist aber möglich, schon bestehende Archivverzeichnisse in ein neu einzurichtendes Archiv zu übernehmen, wenn es keinem anderen Archiv (mehr) zugeordnet ist. Allerdings können Archivverzeichnisse eines Migrations- oder Langzeitarchivs nach dem ersten Schreibauftrag nicht mehr einem Backup-Archiv zugeordnet werden.

Das Archivverzeichnis muss nicht der Benutzerkennung des Archiveigentümers zugewiesen werden. Aber nur der HSMS-Verwalter hat das Recht, Archivverzeichnisse unter einer fremden Benutzerkennung anzulegen.

Die Benutzerkennung des Archivverzeichnisses ist dieselbe Benutzerkennung, für die freie Bänder bei MAREN reserviert werden.

Öffentliche Archive können vor unberechtigtem Zugriff geschützt werden, indem das Archivverzeichnis mit einem Schutz (Kennwort, BACL oder GUARDS) versehen wird. Wenn ein derartiger Schutz festgelegt wurde, ist unbedingt *Schreibzugriff* für das Archivverzeichnis erforderlich, auch wenn für das Archiv USER-ACCESS=*ALL-USERS (ACCESS=*READ) vereinbart wurde.

In ähnlicher Weise können auch private Archive für den autorisierten Zugriff durch eine Gruppe von Benutzern durch die Miteigentümerschaft (SECOS) geöffnet werden. Dabei ist es gleichgültig, welcher Benutzerzugriff für das Archiv für alle Benutzer festgelegt wurde.

Ein Archivverzeichnis für BS2000-Dateien und Jobvariablen kann nicht an Stelle eines Archivverzeichnisses für Knotendateien verwendet werden und umgekehrt.

Hinweis für ARCHIVE-Benutzer

Realisiert wird das Archivverzeichnis durch eine ARCHIVE-Directory-Datei. Es ist möglich, ARCHIVE-Directory-Dateien unter HSMS-Verwaltung zu nehmen (siehe dazu den [Abschnitt „Besonderheiten bei Übergang von ARCHIVE- zu HSMS-Betrieb“](#)).

Hinweis zur Beschränkung der Satzlänge in Directory-Dateien

Zurzeit können die Daten einer BS2000-Datei maximal auf ungefähr 300 Datenträger verteilt werden, da die Satzlänge in Directory-Dateien begrenzt ist. Dieser Wert ist abhängig von den Sicherungsoptionen, z.B. kann er bei SAVE-PLAM-INFO = *YES auf etwa 294 sinken.

Falls beim Sichern der Datei mehr Datenträger benötigt werden, kann die vollständige Information nicht in das betreffende Directory geschrieben werden. In diesem Fall wird der Auftrag als "COMPLETED WITH ERRORS" beendet und die Meldung ARC0176 ausgegeben.

3.3 Standard-Systemarchive

HSMS sieht für die Grundfunktionen Standard-Systemarchive vor. Standard-Systemarchive richtet der HSMS-Verwalter als öffentliche Archive ein und ordnet ihnen die Grundfunktionen Datensicherung, Langzeitarchivierung oder Verdrängung zu. Sie können für diese Funktionen genutzt werden, ohne dass der Archivname bekannt ist, weil sie über symbolische Namen angesprochen werden können, und zwar:

- SYSBACKUP für das System-Backup-Archiv
- SYSVERSION für das System-Versions-Backup-Archiv
- SYSARCHIVE für das System-Langzeitarchiv
- SYSMIGRATE für das System-Migrationsarchiv
- SYSNODEBACKUP für das System-Backup-Archiv für Knotendateien des POSIX-Dateisystems oder ferner Knoten-S0
- SYSNODEARCHIVE für das System-Langzeitarchiv für Knotendateien des POSIX-Dateisystems oder ferner Knoten-S0

In einer Umgebung mit **SF-Pubsets** werden Standard-Systemarchive mehrbenutzbar eingerichtet und systemweit zugewiesen mit:

```
//MODIFY-HSMS-PARAMETERS DEFAULT-HSMS-STORAGE=*PAR( -
//  SYSBACKUP=<archivname>, -
//  SYSARCHIVE=<archivname>, -
//  SYSMIGRATE=<archivname>, -
//  SYSNODEBACKUP=<archivname>, -
//  SYSNODEARCHIVE=<archivname>)
```

Die Zuweisung der System-Backup-Archive gilt in diesem Fall nur für alle importierten SF-Pubsets und Knoten-S0.

Systemarchive für Langzeitarchivierung gelten ebenfalls für alle importierten SF-Pubsets und Knoten-S0, sind aber nicht auf diese beschränkt (siehe Zusammenhang mit SM-Pubsets).

Die Zuweisung des Migrationsarchivs gilt nur für solche BS2000 S0-Pubsets, für die Verdrängung zugelassen ist.

Von diesen systemweiten Einstellungen kann pubset-spezifisch bzw. Knoten-S0-spezifisch abgewichen werden:

- Einzelnen Pubsets können mit der HSMS-Anweisung MODIFY-PUBSET-PARAMETERS abweichende Standard-Systemarchive für SYSBACKUP, SYSARCHIVE, SYSMIGRATE und SYSVERSION zugewiesen werden.
- Einzelnen Knoten-S0 können mit der HSMS-Anweisung MODIFY-NODE-PARAMETERS abweichende Standard-Systemarchive für SYSNODEBACKUP und SYSNODEARCHIVE zugewiesen werden.

i Systemarchive für den Versions-Backup können nur einzelnen Pubsets mit der HSMS-Anweisung MODIFY-PUBSET-PARAMETERS zugeordnet werden. Außerdem kann ein Versions-Backup-Archiv nur einem Pubset zugewiesen sein.

In einer Umgebung mit **SM-Pubsets** werden Standard-Systemarchive mehrbenutzbar eingerichtet und dem SM-Pubset zugewiesen mit:

```
//MODIFY-SM-PUBSET-PARAMETERS -  
// SM-PUBSET-ID=<cat-id>, -  
// SYSBACKUP=<archivname>, -  
// SYSARCHIVE=<archivname>, -  
// SYSMIGRATE=<archivname>
```

Die Zuweisung der Systemarchive für Backup, Versions-Backup, Langzeitarchivierung und Migration gilt nur innerhalb dieses SM-Pubsets.

Die Archivierungstätigkeit ist nicht auf die SM-Pubset-Umgebung beschränkt, in der sie erteilt wird. Die Umgebung, auf die sich die Archivierungsfunktion bezieht, kann die Host-Umgebung sein. In diesem Fall ist die Archivdefinition in der zentralen HSMS-Steuerdatei hinterlegt.

Wenn ein Benutzer ein Standard-Systemarchiv über seinen symbolischen Namen anspricht, wird dieser Name aufgelöst. Dadurch kann der Name auf das Archiv abgebildet werden, das der HSMS-Verwalter einer festgelegten Umgebung zugewiesen hat.

Die Umgebung, von der die symbolischen Archivnamen aufgelöst werden, ist im Allgemeinen die Katalog- oder Knotenkennung, die in der HSMS-Anweisung angegeben ist. Alle betroffenen Katalog- oder Knotenkennungen werden abgesucht, um ihr zugewiesenes symbolisches Archiv zu finden:

- Bei HSMS-Anweisungen, die in einer SF-Umgebung arbeiten, handelt es sich um solche Katalogkennungen, die beim Operanden S0-PUBSET-ID (bei der HSMS-Anweisung REPAIR-CATALOG-BY-EXCHANGE und REPAIR-CATALOG-BY-RESTORE) angegeben sind oder die beim Operanden FILE-NAMES einer HSMS-Anweisung verwendet werden.
- Bei HSMS-Anweisungen, die in einer SM-Umgebung arbeiten, handelt es sich um die Katalogkennung, die beim Operanden ENVIRONMENT angegeben ist.
- Bei HSMS-Anweisungen, die auf Workstations arbeiten, werden die Knotenkennungen vom Operanden NODE-ID genommen.

Eine Ausnahme gibt es bei der HSMS-Anweisung SHOW-ARCHIVE in einer SF-Umgebung, wenn beim Operanden SELECT die Werte *SAVE-VERSIONS, *SAVE-FILES oder *VOLUMES verwendet werden. Bei diesen Operandenwerten kann nämlich keine Dateiliste, keine Katalogkennung und keine Knotenkennung angegeben werden:

- Bei einem nicht-privilegierten Benutzer wird die Standard-Katalogkennung des Benutzers verwendet (die Standard-Katalogkennung ist im Benutzerkatalog definiert) oder das BS2000-UFS, um die symbolischen Archivnamen aufzulösen.
- Bei einem privilegierten Benutzer werden alle Katalog- und Knotenkennungen abgesucht, die in HSMS definiert sind (mit den HSMS-Anweisungen MODIFY-PUBSET-PARAMETERS oder MODIFY-NODE-PARAMETERS), um die symbolischen Archivnamen aufzulösen.

Die Abbildung der symbolischen Namen muss eindeutig sein. Da die Abbildung pubsetweise/Knoten-S0-weise vorgenommen wird, gibt es folgende Einschränkung:

Wenn mit dem Operanden FILE-NAMES oder JV-NAMES Dateien bzw. Jobvariablen von verschiedenen Pubsets /Knoten-S0 bearbeitet werden sollen, muss bei Angabe eines symbolischen Archivs allen Pubsets/Knoten-S0 dasselbe Standard-Systemarchiv zugewiesen sein. Ist dies nicht der Fall, dann sollten pro Auftrag nur jeweils Dateien eines einzigen Pubsets/Knoten-S0 gesichert werden.

Bei Systemarchiven für den Versions-Backup ist es nicht erlaubt, ein und dasselbe Archiv mehreren Pubsets zuzuweisen. Hier muss sich somit der Operand FILE-NAMES auf die Dateien nur eines Pubsets beziehen.

Dennoch ist es aus Performancegründen sinnvoll, verschiedene Archive für die Pubsets/ Knoten-S0 einzurichten, da einige Archivtätigkeiten nur seriell durchgeführt werden können. Mehrere Systemarchive pro Grundfunktion verringern so die Wahrscheinlichkeit von Engpässen beim Zugriff auf Archive.

3.4 Zusammenhang zwischen SF-Pubsets und Archiven

In einem Rechenzentrum wird flexibel gearbeitet:

- Pubsets werden von einem Rechner getrennt und mit einem anderen Rechner verbunden.
- Benutzerkennungen werden von einem Pubset auf einen anderen Pubset verlegt.
- Migrierte Dateien werden gelöscht oder auf die Verarbeitungsebene S0 zurückgeholt. Deshalb müssen die Speicherebenen S1 und S2 regelmäßig reorganisiert werden.
- Bei einem Systemabsturz oder einem Bedienfehler sollte der Schaden eng begrenzt bleiben (z.B. auf einen einzigen Pubset).
- Nach einem Systemabsturz oder Bedienfehler müssen die verlorenen Daten so restauriert werden, dass die Arbeit auf den unbeschädigten Pubsets nicht behindert wird.
- Während auf einem Pubset Verwaltungsaufgaben durchgeführt werden (Backup, Migration, Reorganisation, ...), sollten die Benutzer auf den anderen Pubsets nicht gestört werden.

3.4.1 Mögliche Organisationsformen

Im Folgenden werden die zentrale und die dezentrale Verwaltung sowie deren Vor- und Nachteile dargestellt. Die Art der Verwaltung ist besonders für die Sicherung und Verdrängung von Bedeutung.

3.4.1.1 Zentrale Verwaltung

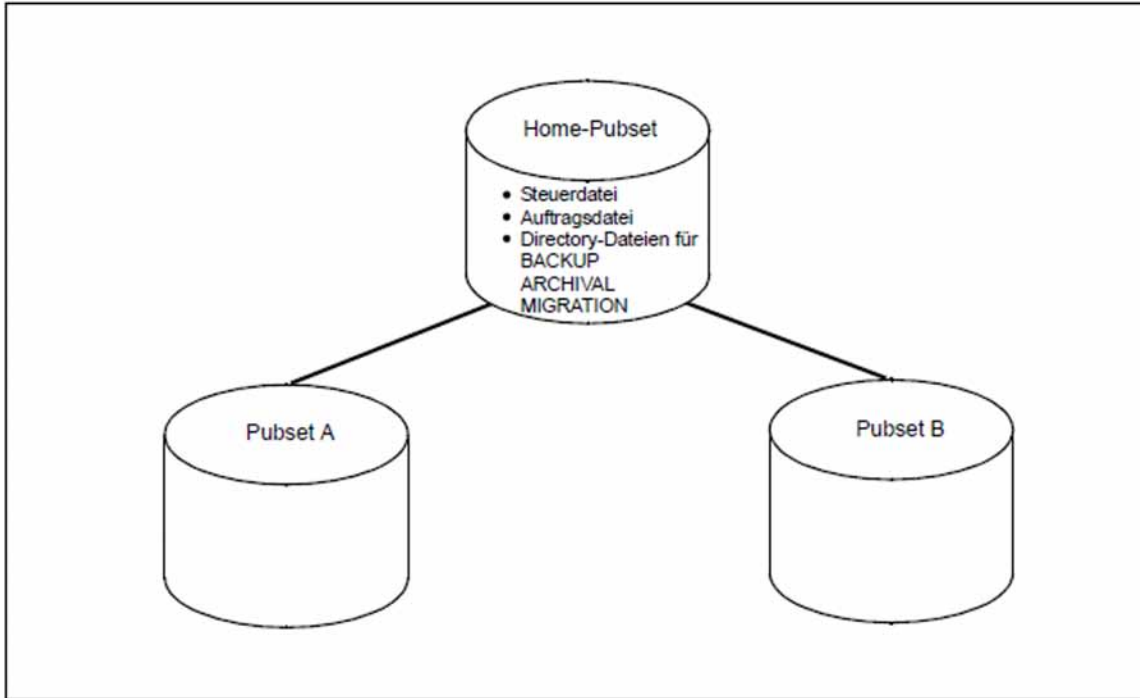


Bild 6: Zentrale Verwaltung

3.4.1.2 Dezentrale Verwaltung

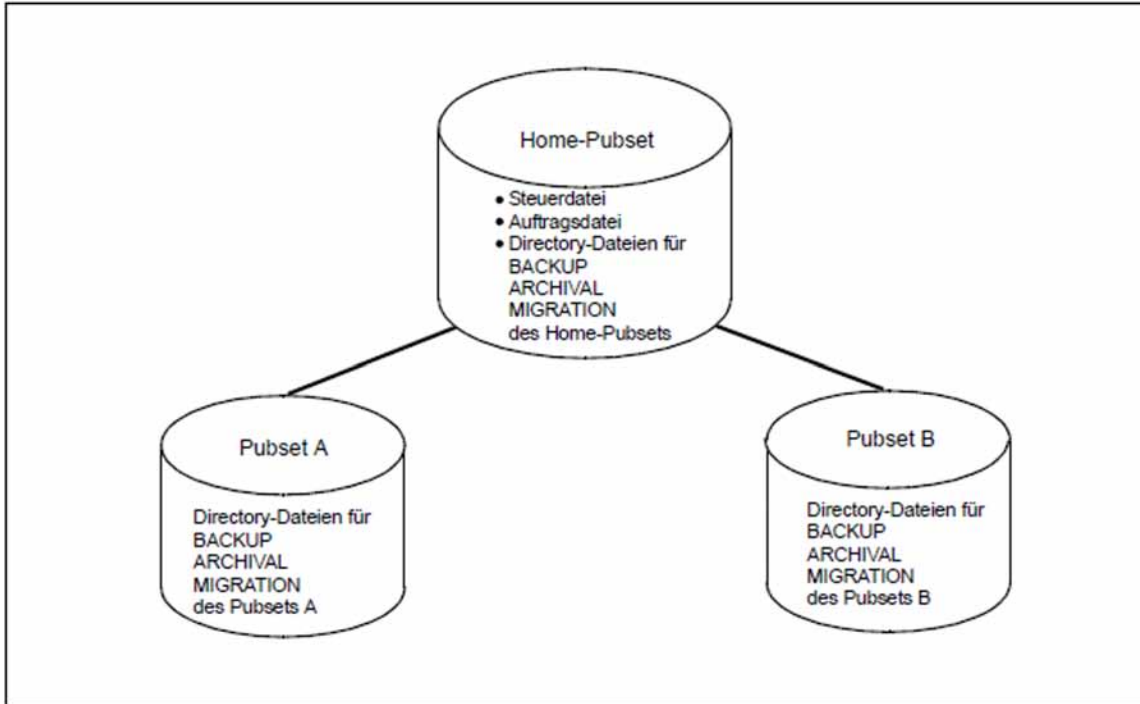


Bild 7: Dezentrale Verwaltung

3.4.1.3 Vorteile (+) und Nachteile (-) der beiden Organisationsformen

Zentrale Verwaltung	Dezentrale Verwaltung
<p>Alle Pubsets werden gemeinsam verwaltet: 1 Directory-Datei für Migration 1 Directory-Datei für Backup 1 Directory-Datei für Archival</p>	<p>Jeder Pubset wird getrennt verwaltet: 1 Directory-Datei für Migration pro Pubset 1 Directory-Datei für Sicherung pro Pubset 1 Directory-Datei für Archival pro Pubset 1 Directory-Datei für Versions-Backup pro Pubset</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #add8e6;"> <p>i Versions-Backup ist nur innerhalb der dezentralen Verwaltung möglich.</p> </div>
<p>+ Nur 1 HSMS-Anweisung für alle Pubsets für Sicherung und Migration</p>	<p>- 1 HSMS-Anweisung pro Pubset für Sicherung, Versions-Backup und Migration</p>
<p>+ ARCHIVE synchronisiert automatisch die Sicherung oder Migration von allen Pubsets.</p>	<p>- Die Synchronisation zwischen Tasks von verschiedenen Pubsets wird mit /SECURE-RESOURCE-ALLOCATION auf dem Ausgabegerät durchgeführt. Bei unterschiedlichen Pubsetgrößen kann ein Zeitverlust auftreten.</p>
<p>+ Optimales Vollschieben der Sicherungsdatenträger bei kleinen Pubsets (1 Datenträger pro Sicherungslauf für mehrere Pubsets).</p>	<p>- Schlechte Ausnutzung der Datenträger bei kleinen Pubsets (1 Datenträger pro Sicherungslauf und pro Pubset). Die Ausnutzung der Datenträger kann durch Fortschreiben bei nachfolgenden Sicherungen /Migrationen jedoch verbessert werden</p>
<p>- Große Directory-Dateien werden verwendet: Sehr negativer Einfluss auf die Performance</p>	<p>+ Kleinere Directory-Dateien verbessern die Laufzeit und verringern Zugriffskonflikte.</p>
<p>- Wenn ein Pubset auf ein anderes Home-Pubset verlegt wird, ist die Sicherung ohne zusätzliche Operation (COPY-SAVE-FILE) nicht mehr verfügbar.</p>	<p>+ Wenn das Home-Pubset wechselt, ist die Sicherung immer noch verfügbar. Wenn die Directory-Datei nicht auf dem Pubset liegt, muss sie kopiert werden.</p>
<p>- Wenn die Directory-Datei auf dem Home-Pubset liegt und auf diesem ein Crash auftritt, muss die Directory-Datei restauriert werden. Die Benutzer können erst wieder arbeiten, wenn die HSMS-Umgebung vollständig restauriert wurde.</p>	<p>+ Nach einem Crash des Home-Pubsets kann die Bearbeitung mit einem anderen Pubset fortgesetzt werden, sobald die Steuerdatei restauriert ist. Die Bearbeitung kann mit einem anderen Home-Pubset fortgesetzt werden, nachdem die Umgebung neu definiert wurde; die Directory-Dateien werden beibehalten.</p>

<ul style="list-style-type: none">- Die serielle Bearbeitung von Aufträgen bei jedem Archiv hat eine große Wirkung	<ul style="list-style-type: none">+ Die serielle Bearbeitung von Aufträgen hat eine geringe Wirkung.
<ul style="list-style-type: none">- Nach einem Wechsel des Home-Pubsets kann ein Auftrag nicht erneut gestartet werden.	
<ul style="list-style-type: none">- Nach einem Wechsel des Home-Pubsets gehen die Archivdefinitionen verloren.	

3.4.1.4 Empfohlene Organisationsform: Dezentrale Verwaltung

Um eine möglichst große Flexibilität im Rechenzentrum zu erreichen, sollte die dezentrale Verwaltung benutzt werden. Wenn allerdings sehr kleine Pubsets zu verwalten sind, ist die Bandbearbeitung mit einer zentralen Verwaltung kostengünstiger, da weniger Magnetbandkassetten benötigt werden.

Damit nach einem Systemabsturz ein Pubset leicht restauriert werden kann, sollten die HSMS-Anweisungen und Kommandos, die die Archive für Backup, Migration usw. definieren, als Prozedurdatei gesichert werden.

Hinweise für die Sicherung

1. Für jeden Pubset sollte eine Prozedurdatei erzeugt werden, in der ein Archiv für Verdrängung, ein Archiv für Backup sowie alle anderen gewünschten Archive definiert werden.

Beispiel

```
/BEGIN-PROCEDURE PARAMETERS=*YES( PROCEDURE-PARAMETERS=( &PVSID) -
/  ESCAPE-CHARACTER=C '&' )

  /START-HSMS
  //CREATE-ARCHIVE ARCHIVE-NAME=MIGRATE.&PVSID, -
  //  ALLOWED-USAGE=*MIGRATION, . . . , -
  //  DIRECTORY-NAME=:&PVSID:$SYSHSMS.MIGRATION.DIR
  //CREATE-ARCHIVE ARCHIVE-NAME=BACKUP.&PVSID, -
  //  ALLOWED-USAGE=*BACKUP( SAVE-FILE-STRUCTURE=*SEVERAL-SVID) , . . . , -
  //  DIRECTORY-NAME=:&PVSID:$SYSHSMS.BACKUP.DIR
  //CREATE-ARCHIVE ARCHIVE-NAME=ARCH.&PVSID, -
  //  ALLOWED-USAGE=*ARCHIVAL, . . . , -
  //  DIRECTORY-NAME=:&PVSID:$SYSHSMS.ARCHIVAL.DIR
  //MODIFY-PUBSET-PARAMETERS PUBSET-ID=&PVSID, -
  //  STORAGE-LEVEL=*S0( SYSMIGRATE=MIGRATE.&PVSID) , -
  //  SYSARCHIVE=ARCH.&PVSID, -
  //  SYSBACKUP=BACKUP.&PVSID
  //END

/END-PROCEDURE
```

2. Für die Systemsicherung muss für jeden Pubset folgende HSMS-Anweisung gegeben werden:

```
//BACKUP-FILES FILE-NAMES=:&PVSID:$*., -
//  JV-NAMES=:&PVSID:$*., . . .
```

3. Für die Migration aller Dateien des Systems ist für jeden Pubset folgende HSMS-Anweisung erforderlich:

```
//MIGRATE-FILES FROM-STORAGE=*S0-STORAGE-LEVEL -
//  ( FILE-NAMES=:&PVSID:$*., UNUSED= . . . , . . . ) . . .
```

4. Für Versions-Backup muss für jeden Pubset folgende HSMS-Anweisung gegeben werden:

```
//BACKUP-FILE-VERSIONS FILE-NAMES=:&PVSID:$*., . . .
```

5. Das Zurückholen oder das Restaurieren muss ebenfalls pubset-weise durchgeführt werden.

Hinweis für den Wechsel des Home-Pubsets

Wenn ein Pubset exportiert und anschließend importiert wird, sind die Archivdefinitionen verloren. Die Directory-Dateien sind aber immer noch auf dem Pubset vorhanden. Anstelle der oben aufgeführten Prozedur können ab HSMS V8.0 die Archivattribute mit der Anweisung CREATE-ARCHIVE (Operand RECONSTRUCT-ARCHIVE=*YES) aus der Directory-Datei rekonstruiert werden (siehe auch „Versehentlich gelöschte Archivdefinition wiederherstellen“, ["Archivdefinition löschen"](#)). Wenn die Verdrängung oder die Sicherung auf S1 durchgeführt wurde, muss auch der S1-Pubset importiert werden. Die S2-Speicherebene muss für den neuen Rechner ebenfalls verfügbar sein.

3.5 Zusammenhang zwischen SM-Pubsets und Archiven

Bei SM-Pubsets müssen die Archive für die Backup-, die Versions-Backup-, die Archivierungs- und die Migrationsfunktion dezentral verwaltet werden.

Darüber hinaus werden alle Funktionen, die zum Verwalten der Archive benötigt werden, im SM-Pubset selbst abgelegt. Dies bedeutet, dass eine Steuerdatei, die für einen SM-Pubset spezifisch ist, die Archivdefinitionen für diesen SM-Pubset enthält. Die Archivverzeichnisse, die den Archiven zugewiesen sind, werden ebenfalls im SM-Pubset abgelegt.

Hinweis

Das Standard-Systemarchiv eines SM-Pubsets für die Langzeitarchivierung von BS2000-Dateien kann auch in den globalen HSMS-Parametern festgelegt werden. Ein in der SF-Umgebung definiertes Langzeitarchiv für BS2000-Dateien kann sowohl für SF-Pubsets als auch für SM-Pubsets genutzt werden.

3.6 Zusammenhang zwischen Knoten-S0 und Archiven

Passive Knoten-S0 können – ebenso wie die Pubsets – flexibel benutzt werden:

- Die Sicherung/Langzeitarchivierung eines Knoten-S0 kann von einem zentralen BS2000-System verwaltet werden.
- Der Schaden bei einer zufälligen Zerstörung eines Archivs oder einer Directory-Datei kann auf einen einzigen Knoten-S0 begrenzt werden.
- Die Verwaltungsarbeit (Sicherung, Reorganisation der Directory-Datei usw.), die auf einem Knoten-S0 ausgeführt wird, sollte nicht andere Aufträge beeinträchtigen, die auf anderen Knoten-S0 bearbeitet werden.
- Dateien und Directory-Dateien können von einem Knoten-S0 zu einem anderen Knoten-S0 verlegt werden.

3.6.1 Zentrale und dezentrale Verwaltung

Im Folgenden werden die zentrale und die dezentrale Verwaltung sowie deren Vor- und Nachteile dargestellt.

Zentrale Verwaltung	Dezentrale Verwaltung
Alle Knoten-S0 werden gemeinsam verwaltet: 1 Directory-Datei für Backup 1 Directory-Datei für Archival	Jeder Knoten-S0 wird getrennt verwaltet: 1 Directory-Datei für Backup pro Knoten-S0 1 Directory-Datei für Archival pro Knoten-S0
+ Nur 1 HSMS-Anweisung für alle Knoten-S0 während der Sicherung erforderlich	- 1 HSMS-Anweisung pro Knoten-S0 für Sicherung erforderlich
- Große Directory-Dateien werden gebraucht.	+ Kleinere Directory-Dateien verbessern die Laufzeit und verringern Zugriffskonflikte.
- Wenn ein Knoten-S0 auf einen anderen BS2000-Server verlegt wird, ist die Sicherung ohne zusätzliche Operation (COPY-NODE-SAVE-FILE) nicht mehr verfügbar.	+ Wenn ein Knoten-S0 auf einen anderen BS2000-Server verlegt wird, ist die Sicherung immer noch verfügbar. Die Directory-Datei muss kopiert werden.
+ ARCHIVE synchronisiert automatisch die Sicherung oder Archivierung von allen Knoten-S0.	- /SECURE-RESOURCE-ALLOCATION synchronisiert die Tasks von verschiedenen Knoten-S0 auf dem Ausgabegerät. Größenunterschiede der Knoten-S0 können zu Zeitverlusten führen.
+ Optimales Vollschieben von Sicherungsdatenträgern bei kleinen Knoten-S0 (1 Datenträger pro Sicherungslauf für mehrere Knoten-S0)	- Schlechte Ausnutzung der Datenträger bei kleinen Knoten-S0 (1 Datenträger pro Sicherungslauf und pro Knoten-S0). Die Ausnutzung der Datenträger kann durch Fortschreiben bei nachfolgenden Sicherungen /Migrationen jedoch verbessert werden.

3.7 Sicherungsdatei

Eine Sicherungsdatei (Save-File) ist ein Behälter, in dem die vom Archiv verwalteten Daten aufbewahrt werden. Wenn von HSMS Daten auf Magnetbandkassette, Platte oder Net-Storage geschrieben werden, egal für welche Grundfunktion, werden diese Daten in eine Sicherungsdatei aufgenommen. HSMS legt also keine katalogisierten Kopien der gesicherten Dateien an.

Bei einem HSMS-Sicherungslauf wird immer nur in eine einzige Sicherungsdatei geschrieben, selbst wenn viele Dateien in dem Lauf gesichert werden.

Die Sicherungsdatei selbst besteht aus einer oder mehreren katalogisierten BS2000-Dateien auf Platte, auf einem oder mehreren Net-Storage-Volumes oder auf Magnetbandkassetten. Eine Sicherungsdatei auf Magnetbandkassetten besteht aus einer Reihe von Datenträgern mit demselben Eigentümer (dieselbe Benutzerkennung ist im Header des Magnetbandkassetten eingetragen) und derselben Schutzfrist.

Zu den Namen der von ARCHIVE für HSMS angelegten Sicherungsdateien siehe den [Abschnitt „Sicherungsdateinamen“](#).

Eine Sicherungsdatei auf Platte (Speicherebene S1) liegt unter der Benutzerkennung, die das Archivverzeichnis enthält, verbunden mit dem Archiv, in dem die Sicherungsdatei abgelegt ist. Zur besseren Portabilität werden Sicherungsdateien im neutralem NK4-Format angelegt. Dies ist unabhängig vom Format des Pubsets (K/NK) oder der Einstellung des Systemparameters BLKCTRL.

Jede Sicherungsdatei ist durch die Save-File-ID (SFID) eindeutig gekennzeichnet. Sie wird aus dem Datum und der Uhrzeit der Erzeugung gebildet:

`sfid = S.yymmdd.hhmmss`

HSMS bietet mit der Anweisung COPY-SAVE-FILE eine Funktion, mit der Sicherungsdateien – auch auszugsweise – innerhalb eines Archivs kopiert werden können. Dadurch ist z.B. eine effektive Reorganisation älterer Langzeitarchive möglich. In BS2000/OSD-BC ab V9.0 kann die Sicherungsdatei wahlweise auch auf einen verfügbaren Net-Storage kopiert werden. In HSMS ab V10.0 kann sie auch auf gemeinschaftliche Platte kopiert werden.

Mit der HSMS-Anweisung COPY-NODE-SAVE-FILE können Knoten-Sicherungsdateien des POSIX-Dateisystems oder ferner Knoten-S0 innerhalb eines Archivs kopiert werden.

Die HSMS-Anweisung COPY-EXPORT-SAVE-FILE dient zum Kopieren von Sicherungsdateien, die mit der HSMS-Anweisung EXPORT-FILES oder mit der EXPORT-Anweisung des Softwareprodukts ARCHIVE erzeugt wurden.

Eine Sicherungsdatei kann entweder nur eine einzige oder mehrere Sicherungsversionen enthalten. Dies hängt von der Speicherebene der Sicherungsdatei und von der Struktur des zugehörigen Archivs ab (SINGLE-SVID oder SEVERAL-SVID).

Migrationsarchive, Langzeitarchive, Versions-Backup-Archive und Archive für Knotendateien des POSIX-Dateisystems oder ferner Knoten-S0 sind immer Archive mit SEVERAL-SVID-Struktur. Backup-Archive können entweder eine SINGLE-SVID- oder eine SEVERAL-SVID-Struktur haben – abhängig vom Operanden SAVE-FILE-STRUCTURE in der Anweisung CREATE-ARCHIVE.

Sicherungsdateien von Archiven mit SINGLE-SVID-Struktur und Sicherungsdateien auf der Speicherebene S1, auf Privatplatte, auf Pubset-Platte und auf Net-Storage enthalten nur eine einzige Sicherungsversion.

Sicherungsdateien auf der Speicherebene S2 von Archiven mit SEVERAL-SVID-Struktur können mehrere Sicherungsversionen enthalten. Wenn eine Sicherungsdatei, die mehrere Sicherungsversionen enthält, auf die Speicherebene S1, auf Privatplatte, gemeinschaftliche Platte oder Net-Storage kopiert wird, dann wird für jede Sicherungsversion der Original-Sicherungsdatei auf dem Zielmedium eine eigene Sicherungsdatei erzeugt.

3.7.1 Sicherungsdateinamen

Die Namen der von ARCHIVE für HSMS angelegten Sicherungsdateien hängen vom Speichermedium und ggf. vom Verarbeitungsmodus für Sicherungsdateien (SAVE-FILE-PROCESSING) ab:

- Magnetband oder Magnetbandkassette

ARCHIVE.SAVE.FILE(*date-time-subsave#*-0)

- Private Platte

ARCHIVE.SAVE.FILE.*date.time.vsn*

- Gemeinschaftliche Platte / S1

V9:	ARCHIVE.SAVE.FILE. <i>date.time.subsave#</i>	oder
V10:	ARCHIVE.SAVE.FILE. <i>date.time.subsave#.seq#</i>	

- Net-Storage

V9:	ARCHIVE.SAVE.FILE. <i>date.time.vsn</i>	oder
V10:	ARCHIVE.SAVE.FILE. <i>date.time.subsave#.0</i>	

Legende:

date.time Erzeugungszeitpunkt im Format yymmdd.hhmmss

subsave# Subsave-Nummer des Laufs, der diese Sicherungsdatei erstellt hat

seq# Laufende Nummer der Sicherungsdatei (0..F)

V9: SAVE-FILE-PROCESSING=*HSMS-V9-COMPATIBLE

V10: SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE

3.7.2 Schutzfrist und Freigabedatum

Datensicherungen und Archivierungen werden erstellt, damit sie für einen bestimmten Zeitraum als möglicher Ersatz zur Verfügung stehen. Die Länge dieses Zeitraums kann angegeben werden:

Die Schutzfrist (Retention-Period) ist der Zeitraum, während dem ein Überschreiben des Datenträgers und damit ein Löschen der auf ihm enthaltenen Daten verboten ist.

Für die Sicherungsdateien des Archivs wird mit dem Operanden RETENTION-PERIOD festgelegt, wie viele Tage nach Abschluss der Sicherungsdatei der Datenträger mit der Sicherungsdatei nicht überschrieben werden darf.

Bei der Verdrängung und Versions-Backup wird die Schutzfrist nur über die Archivdefinition festgelegt; bei den anderen Grundfunktionen kann sie auch bei der jeweiligen HSMS-Anweisung bestimmt werden.

Aus der Schutzfrist ergibt sich das Freigabedatum (Expiration-Date). Es ist nach Ablauf der Schutzfrist erreicht:

Freigabedatum = Erzeugungsdatum + Schutzfrist (in Tagen)
(Expiration-Date = Creation-Date + Retention-Period)

Bei einer Standard-Sicherungsdatei mit NEW-STD-SAVE-FILE=*IN-PERIODS kommt noch die Fortsetzungsperiode hinzu (siehe [Abschnitt „Sicherungsdateien fortschreiben“](#)).

Das Freigabedatum wird auf dem Datenträger, im Archivverzeichnis und ggf. im MAREN-Katalog vermerkt. Das Freigabedatum kann später über HSMS im Archivverzeichnis erhöht werden; dabei wird automatisch auch das Freigabedatum des Datenträgers im MAREN-Katalog erhöht. Auf dem Datenträger selbst ist diese Änderung jedoch nicht möglich. Dieser ist deshalb nach Ablauf des ursprünglichen Freigabedatums nur dann geschützt, wenn MAREN geladen ist und der Datenträger nicht an ein anderes System gebracht wird.

Wenn die Schutzfrist einer Sicherungsdatei auf Platte erhöht wird, wird dadurch die Sicherungsdatei vor dem Löschen geschützt und das Freigabedatum ihres Katalogeintrags wird geändert.

Ab HSMS V8.0A kann die Schutzfrist einer Sicherungsdatei nicht nur erhöht, sondern auch reduziert werden (Operand SAVE-FILE-RETPD-UPDATE auch mit negativem Wert). Beim nachträglichen Ändern der Schutzfrist einer Sicherungsdatei muss in Backup-Archiven beachtet werden, dass die Einträge Cataloged-Not-Saved von nicht modifizierten Dateien aus Differenzsicherungen zum Restaurieren auch den Bezug zur letzten Vollsicherung in einer möglicherweise anderen Sicherungsdatei benötigen. Eine frühere Freigabe der Sicherungsdatei mit den Vollsicherungen vor Freigabe der Sicherungsdatei mit der Differenzsicherung ist nicht sinnvoll.

Neben dem Freigabedatum für Datenträger, das sich aus der physischen Schutzfrist ergibt, kennt HSMS bei der Langzeitarchivierung auch das logische Freigabedatum (File-Expiration-Date). Es kann einer Sicherungsversion (und den darin enthaltenen Dateien) unabhängig von der Schutzfrist des Datenträgers vergeben werden. Sinnvoll ist dies z.B. beim Umwälzen der Sicherungsbänder: Der Archiveigentümer kann beim Umwälzen der Magnetbandkassetten gezielt nur die Sicherungsversionen übernehmen, deren logisches Freigabedatum noch nicht erreicht ist.

Wenn die Magnetbandkassetten beispielsweise alle 2 Jahre umgewälzt werden und eine physische Schutzfrist von 2 Jahren vereinbart wird, können die Magnetbandkassetten nach dem Kopieren der Dateien, deren logisches Freigabedatum noch nicht erreicht ist, ohne neue Initialisierung wieder für Sicherungen verwendet werden (siehe dazu [Abschnitt „Innerhalb von Langzeitarchiven kopieren“](#)).

Für Langzeitarchive kann der HSMS-Verwalter mit dem Operanden FILE-EXPIRATION-DATE festlegen, ob bei Archivierungen das vom Benutzer angegebene logische Freigabedatum das physische Freigabedatum überschreiten darf (FILE-EXPIRATION-DATE= *UNRESTRICTED).

Da ein Datenträger nach Ablauf der Schutzfrist nicht mehr geschützt ist, muss der Archiveigentümer den Schutz des Datenträgers durch administrative Maßnahmen sicherstellen, z.B. durch die Verwendung des Operanden SAVE-FILE-RETPD-UPDATE. Mit diesem Operanden, der bei Archiven mit mehreren Sicherungsversionen empfehlenswert ist, kann die Schutzfrist des Datenträgers automatisch erhöht werden. Die neue Schutzfrist des Datenträgers hängt vom logischen Freigabedatum der neu erstellten Sicherungsversion ab. Diese Änderung betrifft auch den MAREN-Katalog.

Hinweis

Wenn für die Schutzfrist einer Sicherungsdatei der Wert 0 angegeben wird, kann dies unerwünschte Auswirkungen haben: Der Wert 0 bedeutet nämlich, dass Freigabedatum und Erzeugungsdatum gleich sind. Dadurch hat die Sicherungsdatei keine Schutzfrist und kann unmittelbar nach dem Erstellen gelöscht werden.

Dieser Zustand kann zu einem unerwarteten Löschen und zu Inkonsistenzen in der Directory-Datei führen: Wenn das Löschen eines Archivs, das als Schutzfrist den Wert 0 hat, initialisiert wird und gleichzeitig eine Aktionsanweisung für dieses Archiv gegeben wird (z.B. BACKUP-FILES), dann wird die Sicherungsdatei gelöscht. Da aber die Aktionsanweisung immer noch läuft, werden einige abhängige Datensätze in der Directory-Datei gesperrt; sie können deshalb nicht gelöscht werden. Zu einem späteren Zeitpunkt kann HSMS die Inkonsistenzen feststellen, da Datensätze in der Directory-Datei stehen bleiben.

Deshalb sollten Sie bei einer Schutzfrist mit dem Wert 0 sehr vorsichtig umgehen.

Schutzfrist und Freigabe

Wenn die Schutzfrist abgelaufen ist (Freigabedatum heute oder früher) und alle Sicherungsversionen einer Sicherungsdatei ihr Freigabedatum erreicht haben, gilt eine Sicherungsdatei als „obsolet“ und kann als solche mit der Anweisung MODIFY-ARCHIVE gelöscht werden mit MAREN-Freigabe des Datenträgers bei Bandsicherungen.

Ab HSMS V8.0A besteht auch die Möglichkeit zum automatischen Löschen von obsoleten Sicherungsdateien, implizit bei Sicherungen oder Kopiervorgängen in Backup- oder Langzeit-Archiven. Diese Funktion steht bei Migration und Versions-Backup nicht zur Verfügung. MAREN verzichtet ab V11.0 im Normalfall auch auf einen eigenen Freigabelauf, so dass mit dem Löschen das Band sofort freigegeben wird und bei der anschließenden Sicherung oder Kopie gleich wieder benutzt werden kann. Das automatische Löschen von obsoleten Sicherungsdateien wird eingeschaltet mit dem Archivattribut AUTOMATIC-DELETION = *OBSOLETE-SAVE-FILES.

Beim Several-SVID-Format kann es bei großen Magnetbandkassetten sinnvoll sein, eine Sicherungsdatei mit sehr vielen Sicherungsversionen fortzuschreiben. Irgendwann einmal werden dann vom Anfang der Sicherungsdatei aus viele Sicherungsversionen schon obsolet sein, dass dadurch das erste Bandvolumen der Sicherungsdatei schon freigegeben und wieder neu benutzt werden könnte. Die feinere Funktion auf Basis von Sicherungsversionen besteht für Backup- und Langzeitarchive mit dem Archivattribut AUTOMATIC-DELETION = *OBSOLETE-SAVE-VERSIONS: ein implizites Löschen von obsoleten Sicherungsversionen vom Anfang der Sicherungsdatei her im Archivverzeichnis verbunden mit einer Freigabe der nun nicht mehr gebrauchten Bänder innerhalb der Sicherungsdatei.

Bei beiden Varianten erfolgt das implizite Löschen nur bei Sichern oder Kopieren durch den Archiveigentümer.

*Skizze einer Sicherungsdatei auf Band mit Several-SVID-Format bei implizitem Löschen nach Variante
AUTOMATIC-DELETION = * OBSOLETE-SAVE-VERSIONS*

Unter der Annahme, dass die SVIDs A bis D jeweils ein abgelaufenes Freigabedatum haben, verteilen sich die SVIDs A bis H wie folgt auf die Sicherungsbänder:

Band-1:	SVID-A	SVID-B	SVID-C	
Band-2:	SVID-C-Rest	SVID-D	SVID-E	SVID-F
Band-3:	SVID-F-Rest	SVID-G	SVID-H	Continue-Position

Diese obsoleten Sicherungsversionen werden im Archivverzeichnis gelöscht und zusätzlich wird Band-1 freigegeben. Für Band-2 und Band-3 erfolgt keine Freigabe, weil diese Bänder auch noch nicht abgelaufene SVIDs enthalten. Band-3 wird außerdem auch gebraucht für den nächsten Sicherungslauf.

3.8 Sicherungsversion

Eine Sicherungsversion (Save-Version) entsteht bei jedem Schreibauftrag für ein Archiv. Sie enthält alle Dateien und Jobvariablen, die durch den entsprechenden Auftrag gesichert bzw. – bei einer Differenzsicherung von Dateien – als Cataloged-Not-Saved (CNS) vermerkt wurden.

Eine Sicherungsversion ist immer Teil einer Sicherungsdatei.

Eine Sicherungsversion ist intern gekennzeichnet durch die Save-Version-ID (SVID), die aus dem Datum und der Uhrzeit der Erzeugung gebildet wird. Die Sicherungsversion kann später in HSMS-Anweisungen zum einen über das Datum ihrer Erzeugung angesprochen werden, zum anderen über den Namen, der ihr bei der Aktionsanweisung gegeben wurde, die zur Erzeugung der Sicherungsversion führte. Dazu dient der Operand SAVE-VERSION-NAME. Eine Ausnahme stellen die Sicherungsversionen dar, die beim Versions-Backup erstellt wurden und nur über das Erzeugungsdatum angesprochen werden können. Es ist nicht möglich Sicherungsversionen von Versions-Backup-Archiven Namen zuzuordnen; die Angabe eines Wertes beim Operanden SAVE-VERSION-NAME bei der Anweisung MODIFY-ARCHIVE wird abgewiesen.

Hinweis für ARCHIVE-Benutzer

Die Unterscheidung von Sicherungsdatei und Sicherungsversion wurde bei HSMS eingeführt, da bei der Sicherung, Archivierung und Verdrängung im Gegensatz zu ARCHIVE mehrere Sicherungsversionen in einer einzigen Sicherungsdatei enthalten sein können.

Bei der Bearbeitung solcher Magnetbandkassetten mit ARCHIVE wird die Save-File-ID (SFID) im Report als Save-Version-ID (SVID) ausgegeben. Wenn der Benutzer einzelne Dateien importieren will, muss er deshalb die ausgegebene SVID beim IMPORT-Lauf als SFID angeben.

Die SVID kann über die HSMS-Anweisung LIST-VOLUMES ermittelt werden. Falls die SVIDs in geordneter Reihenfolge auf der Magnetbandkassette stehen, kann der Benutzer mit der LIST-Anweisung von ARCHIVE die SVIDs der einzelnen Dateien erfahren, wobei er die SFID angeben muss.

3.8.1 Sicherungstyp

Dateien werden in HSMS pro Sicherungsversion verwaltet. Bei jeder Sicherung erhalten die bearbeiteten Dateien einen Sicherungstyp (Save-Type), der bei den HSMS-Anweisungen SHOW-ARCHIVE und SELECT-FILE-NAMES und in den Reporten ausgegeben wird. In den Reporten wird allerdings nicht zwischen CNS und OPER unterschieden; es wird nur CNS ausgegeben.

Sicherungstyp	Bedeutung
FULL	Die Datei wurde komplett gesichert.
FULB	Die Datei wurde komplett gesichert. Meta-Informationen von Bibliothekselementen wurden mitgesichert.
PART	Nur die geänderten Teile der Datei wurden gesichert.
PARB	Nur die geänderten Teile der Datei wurden gesichert. Meta-Informationen von Bibliothekselementen wurden mitgesichert.
MIGF	Die Datei ist verdrängt; der Katalogeintrag wurde gesichert.
FGGI	Der Index einer Dateigenerationsgruppe wurde gesichert.
CATL	Nur der Katalogeintrag der Datei wurde gesichert.
FULJ	Es wurde eine Jobvariable gesichert.
CNS	Die Datei steht im Katalog, wurde aber nicht gesichert.
OPER	Untermenge von CNS. Die Datei wurde wegen eines Fehlers beim Öffnen nicht gesichert (in Reporten unter CNS aufgeführt).
MDS	Die Knotendatei wurde während des Sicherungsprozesses geändert. <i>Hinweis</i> Die Änderung kann nicht festgestellt werden, wenn die Datei bereits vor der Bearbeitung durch ARCHIVE geöffnet ist und erst nach dem ARCHIVE-Lauf geschlossen wird.
FNOD	FULL-NODE-FILE, SAM-Nodefiles die mit SAVE-SAM-STRUCT=*NO gesichert wurden (Standard, RAW-Modus). Diese SAM-Nodefiles können nur auf Net-Storage (FILE-TYPE=*NODE-FILE) rekonstruiert werden (ab HSMS 11.0A und BS2000 OSD/BC 11.0A).

3.8.2 Komprimieren von Sicherungsversionen

Um die Kapazität der Sicherungsdatenträger besser auszunutzen, bietet HSMS die Möglichkeit, die Daten vor dem Schreiben in eine Sicherungsdatei zu komprimieren, also zu verdichten. Mit dem Operanden COMPRESS-FILES der HSMS-Anweisungen ARCHIVE-FILES, BACKUP-FILES, BACKUP-FILE-VERSIONS, EXPORT-FILES und MIGRATE-FILES wird die softwareseitige Komprimierung durch HSMS veranlasst.

Durch die Archivvoreinstellung kann der Archiveigentümer den Standardwert für die entsprechenden HSMS-Anweisungen festlegen und darüber hinaus bestimmen, dass Sicherungsversionen nur beim Schreiben auf die Speicherebene S1 komprimiert werden, nicht jedoch beim Schreiben auf S2 (COMPRESS=*S1-ONLY).

Beim Lesen der Daten, z.B. beim Restaurieren, ist keine Angabe nötig. HSMS führt bei komprimierten Sicherungen automatisch eine Dekomprimierung durch.

Das Maß der Komprimierung ist abhängig von den gesicherten Dateien; z.B. werden Text-dateien stärker komprimiert als Phasen. Die Komprimierung erhöht zwar die CPU-Belastung, verringert aber die Ein- oder Ausgaben und die Zahl der Sicherungsdatenträger. Daher führt die Komprimierung nicht zu einer Zeiteinsparung bei der Sicherung, sondern verringert die Zahl der zur Sicherung benötigten Datenträger.

Moderne MBK-Geräte führen bereits Hardware-seitig eine Komprimierung durch. In diesem Fall führt HSMS keine Software-seitige Komprimierung durch und ignoriert entsprechende Angaben im Operand COMPRESS.

3.8.3 Erweitern von Sicherungsversionen

Das Erweitern von Sicherungsversionen ist nur bei Backup-Archiven mit SINGLE-SVID-Struktur und Versions-Backup-Archiven von Bedeutung, wenn die Sicherungsdatei, die eine Sicherungsversion enthält, auf dem Plattenspeicher liegt (Speicherebene S1, gemeinschaftliche Platte oder Net-Storage).

Eine Sicherungsversion kann mit dem Operanden `SAVE-FILE=*CONTINUE` erweitert werden

- in der Anweisung `BACKUP-FILES`, wenn das betroffene Archiv ein SINGLE-SVID-Archiv ist.
- in der Anweisung `BACKUP-FILE-VERSIONS`, wenn die Sicherungsdatei auf Platte oder Net-Storage liegt.

Eine Sicherungsversion kann aber nur mit einer disjunkten Menge von Dateien erweitert werden.

Im Falle eines Versions-Backup-Archivs kann nur die jüngste Sicherungsversion erweitert werden.

3.9 Standard-Sicherungsdatei

Die Standard-Sicherungsdatei ist nur bei Archiven mit SEVERAL-SVID-Struktur von Bedeutung.

Wenn in der Sicherungsanweisung keine andere Sicherungsdatei explizit angegeben ist, werden die Sicherungsversionen eines HSMS-Sicherungslaufs in die Standard-Sicherungsdatei eines Archivs geschrieben.

Wenn der Benutzer in der Sicherungsanweisung nichts anderes angibt, werden alle Daten, die in einem Langzeitarchiv oder Backup-Archiv mit SEVERAL-SVID-Struktur gesichert sind, in die Standard-Sicherungsdatei geschrieben. Die Schutzfrist dieser Standard-Sicherungsdatei wird durch die Archivdefinition festgelegt.

Bei der Verdrängung werden die Daten immer in die Standard-Sicherungsdatei geschrieben.

Der Name der Standard-Sicherungsdatei wird in die Archivdefinition des zugehörigen Archivs eingetragen und als Archivattribut angesehen.

3.9.1 Sicherungsdateien fortschreiben

Für Aufträge, die sich auf die Speicherebene S2 beziehen und SEVERAL-SVID-Archive betreffen, bietet HSMS die Möglichkeit, bereits bestehende Sicherungsdateien fortzuschreiben. Dabei wird hinter die schon enthaltenen Sicherungsversionen auf Magnetbandkassette die aktuelle Sicherungsversion geschrieben. Die in der Sicherungsdatei bereits enthaltenen Sicherungsversionen werden von dem Fortschreiben nicht berührt. Die neuen Daten werden hinter die schon beschriebenen Blöcke auf den Datenträger geschrieben. Durch das Fortschreiben werden die Datenträger besser ausgenutzt.

Ob und wie die Standard-Sicherungsdatei fortgeschrieben wird, hängt davon ab, wann die Standard-Sicherungsdatei des Archivs gewechselt wird. Gesteuert wird dies durch die Archivdefinition (Operand NEW-STD-SAVE-FILE):

- Wenn bei jedem Schreibauftrag die Standard-Sicherungsdatei gewechselt wird (*AT-EACH-REQUEST), findet Fortschreiben überhaupt nicht statt. Jede Sicherungsversion wird in einer eigenen Sicherungsdatei abgelegt.
- Wenn die Standard-Sicherungsdatei jeweils zu Beginn einer Bandverarbeitungszeit gewechselt wird (*EACH-TAPE-SESSION bei NEW-STD-SAVE-FILE), wird die Sicherungsdatei fortgeschrieben, bis die Bandverarbeitungszeit beendet ist. Während einer Bandverarbeitungszeit werden also alle Sicherungsversionen standardmäßig in dieselbe Sicherungsdatei geschrieben.
- Mit *IN-PERIODS und dem Operanden CONTINUATION-PERIOD kann eine Fortsetzungsperiode (Continuation-Period) in Tagen definiert werden. Die Standard-Sicherungsdatei wird jeweils zu Beginn der Fortsetzungsperiode gewechselt. Sie wird bis zum Ende der Fortsetzungsperiode fortgeschrieben. Alle Sicherungsversionen werden während dieser Zeit standardmäßig in diese Sicherungsdatei geschrieben.

Bei Festlegung einer Fortsetzungsperiode errechnet sich das Freigabedatum wie folgt:

$$\text{Freigabedatum} = \text{Erzeugungsdatum} + \text{Schutzfrist} + \text{Fortsetzungsperiode}$$

(Expiration-Date = Creation-Date + Retention-Period + Continuation-Period)

HSMS legt alle Sicherungsdateien mit SEVERAL-SVID-Struktur als fortsetzbar an – unabhängig von der Vorgabe durch die Archivdefinition. Dadurch können Sicherungsdateien auch ausdrücklich per HSMS-Anweisung durch den HSMS-Verwalter fortgesetzt werden. Dies ist bei verschiedenen HSMS-Anweisungen mit der Angabe SAVE-FILE=*CONTINUE möglich.

- Mit *PUBLIC-DISK(...) kann für ein Backup-Node-Archiv die Katalogkennung eines Pubsets vereinbart werden. Durch diesen Eintrag werden dann mit der Anweisung BACKUP-NODE-FILES und dem Verweis auf die Standard-Sicherungsdatei (Angabe SAVE-FILE=*STD) Backup-Aufträge für Knotendateien als Sicherung auf dem vereinbarten Pubset durchgeführt.
- Sicherungsdateien, die in HSMS-Versionen V9.0B und niedriger auf der Speicherebene S1, auf gemeinschaftlicher Platte oder Net-Storage erzeugt wurden, können mit HSMS ab V10.0A nur fortgeschrieben werden, wenn SAVE-FILE-PROCESSING=*HSMS-V9-COMPATIBLE (Default) eingestellt ist.
- Sicherungsdateien auf der Speicherebene S1, auf gemeinschaftlicher Platte oder Net-Storage, die mit dem Modus *HSMS-V10-COMPATIBLE erstellt wurden, können in HSMS der Version V9.0B oder niedriger nicht gelesen oder fortgesetzt werden.
- Sicherungsdateien, die auf der erweiterten Speicherebene S1 (gebildet von allen unter HSMS-Verwaltung stehenden Volume-Sets des SM-Pubsets) erstellt wurden, können nur mit SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE fortgesetzt werden. Das gilt auch, wenn die S1-Ebene wieder auf ein spezifisches Volume-Set reduziert wurde. In diesem Fall wird das zuletzt genutzte Volume-Set fortgesetzt.
- Sicherungsdateien können nur fortgeschrieben werden, wenn für den Operanden SAVE-SAM-STRUCTURE derselbe Wert angegeben wurde wie für die Sicherungsdatei, die fortgeschrieben werden soll.

- Falls die letzte SFID/SVID wegen eines DVS-Fehlers inkonsistent ist, kann die aktuelle Sicherungsdatei nicht fortgeschrieben werden. Es muss eine neue Sicherungsdatei angelegt werden.

3.10 Verwaltung der Archive

Die Verwaltung der Archive ist im Wesentlichen Verwaltung der Objekte in den Archiven, vor allem der Sicherungsdateien und Datenträger. Bei Migration und Versions-Backup kommt das Reorganisieren hinzu; es ist in einem eigenen Abschnitt beschrieben (siehe [Abschnitte „Migrationsarchiv reorganisieren“](#) und [„Versions-Backup reorganisieren“](#)).

Der *Archiveigentümer* erhält mit der HSMS-Anweisung SHOW-ARCHIVE Auskunft über die in einem Archiv verwalteten Objekte, wahlweise für Dateien, Jobvariablen, Sicherungsversionen und -dateien sowie über die Datenträger.

Nicht-privilegierte Benutzer erhalten mit dieser HSMS-Anweisung bei öffentlichen Archiven Informationen über Objekte, die ihnen entweder selbst gehören oder bei denen sie Miteigentümer sind.

Bei SHOW-ARCHIVE können über eine Markierungsfunktion beim Ausgeben der Archivübersichten Informationen über einzelne Sicherungsdateien und Datenträger angefordert werden.

3.10.1 Datenträger-Pool verwalten

Hinweis

Ist das Softwareprodukt MAREN im Einsatz, so werden die Datenträger durch MAREN verwaltet und insbesondere den Archivverzeichnissen zugeordnet (siehe das Handbuch zu „MAREN“ [10] sowie den [Abschnitt „Einschränkungen durch neue Funktionen“](#)). Die Datenträgerverwaltung von HSMS ist damit in die rechenzentrumsweite Datenträgerverwaltung integriert. In diesem Fall sollten Sie auf die in diesem Kapitel beschriebene Datenträgerverwaltung durch HSMS verzichten.

Einem Archiv kann eine Reihe von Magnetbandkassetten zugeordnet werden, die den Datenträger-Pool (Volume-Pool) des Archivs bilden.

Mit `//MODIFY-ARCHIVE VOLUMES=*ADD` werden Datenträger der Klasse „TAPE“ durch Angabe ihrer VSN und ihres Gerätetyps in das Archiv aufgenommen. Sie kommen in den Pool freier Datenträger. Aus diesem Pool heraus werden Sicherungsdaträger für Schreibaufträge für dieses Archiv standardmäßig angefordert.

Mit `//MODIFY-ARCHIVE VOLUMES=*REMOVE` werden Datenträger wieder aus dem Pool freier Datenträger entfernt.

Die ins Archiv aufgenommenen Datenträger werden bei `//SHOW-ARCHIVE SELECT= *VOLUMES` in der Rubrik OWNER mit POOL ausgegeben.

Datenträger, die bei Schreibaufträgen ins Archiv nicht aus dem Datenträger-Pool entnommen wurden, sondern vom Benutzer mit ihrer Archivnummer ausdrücklich angefordert oder vom Operator zugewiesen wurden, werden nur zeitweise ins Archivverzeichnis übernommen. Diese Datenträger werden in der Rubrik OWNER mit OPERATOR ausgegeben (siehe [Abschnitt „Zuweisung von Datenträgern“](#)).

Die Datenträger des Datenträger-Pools können verschiedene Zustände haben:

Zustand	Bedeutung
AVAILABLE	Die VSN wurde nach dem Einbringen ins Archiv noch nicht benutzt, oder die Sicherungsdatei wurde gelöscht und die Schutzfrist ist abgelaufen (Pool freier Datenträger).
IN-USE	Die VSN ist von einer Sicherungsdatei belegt, deren Schutzfrist noch nicht abgelaufen ist.
OBSOLETE	Die Schutzfrist der VSN ist abgelaufen. Die Sicherungsdatei wurde aber noch nicht gelöscht.
FORCED-DELETE	Die VSN wurde trotz noch nicht abgelaufener Schutzfrist freigegeben.
UNUSABLE	Die VSN kann wegen eines Fehlers nicht beschrieben werden.

Sollen Datenträger, die den Zustand FORCED-DELETE haben, vor Ablauf der Schutzfrist wieder beschrieben werden, so müssen sie aus dem Pool entfernt, neu initialisiert und wieder in den Pool eingebracht werden.

Eine Magnetbandkassette im Zustand UNUSABLE, bzw. dessen Archivnummer, kann erst nach Entfernen und erneutem Einbringen in den Pool wieder verwendet werden.

3.10.2 Sicherungsdateien freigeben

Sicherungsdateien, deren Schutzfrist abgelaufen ist, kann der Archiveigentümer mit dem Operanden SAVE-FILES der MODIFY-ARCHIVE-Anweisung aus dem Archiv freigeben (löschen). Auch wenn eine Sicherungsdatei mehrere Sicherungsversionen enthält, kann sie nur als Ganzes freigegeben werden.

Es ist möglich, einzelne Sicherungsdateien durch Angabe ihrer Save-File-Id freizugeben. Die Sicherungsdateien können aber auch durch ihre Attribute ausgewählt werden. So lassen sich alle Sicherungsdateien, deren Schutzfrist abgelaufen ist, mit folgender HSMS-Anweisung löschen:

```
//MODIFY-ARCHIVE SAVE-FILES=*DELETE -  
// (SAVE-FILE-ID=*BY-ATTR(SAVE-FILE-STATE=*OBSOLETE))
```

Gelöschte Sicherungsdateien werden aus dem Archiv entfernt. Bei Sicherungsdateien auf S2 werden die Datenträger wieder in den Pool freier Datenträger aufgenommen. Sicherungsdateien auf S1 oder Privatplatte werden gelöscht. Eine Sicherungsdatei wird jedoch nicht freigegeben,

- wenn sie im Fall eines Versions-Backup-Archivs eine Sicherungsversion mit zumindest einer gültigen Dateiversion enthält;
- wenn Sie im Fall eines Migrationsarchivs migrierte Dateien enthält.

Eine Sicherungsdatei kann auch dann freigegeben werden, wenn ihre Schutzfrist noch nicht abgelaufen ist oder im Falle von Migrations- oder Versions-Backup-Archiven, wenn sie gültige Dateien bzw. Dateiversionen enthält:

```
//MODIFY-ARCHIVE SAVE-FILES=*DELETE(...,FORCED-DELETE=*YES)
```

Die Sicherungsdatei wird ohne weitere Kontrollen gelöscht. Die dazugehörigen Datenträger im Datenträger-Pool des Archivs erhalten den Zustand FORCED-DELETE, bis die ursprünglich vereinbarte Schutzfrist abgelaufen ist. Wird mit einem Archiv ohne Datenträger-Pool gearbeitet, werden bei obiger Anweisung auch die Datenträger ohne weitere Kontrollen freigegeben.

3.10.3 Auf Archivobjekte ohne Archivverzeichnis zugreifen

HSMS unterstützt Vollzugriffe (Schreiben und Lesen) auf Magnetbandkassetten nur, wenn auf das zugehörige Archivverzeichnis zugegriffen werden kann. Wenn kein Archivverzeichnis mehr verfügbar ist, sollte auf die Datenträger nicht mehr schreibend zugegriffen werden. Lesezugriffe sind aber immer noch möglich:

- Beim Zurückholen von migrierten Dateien wird nicht auf das Archivverzeichnis zugegriffen, selbst wenn ein Archivverzeichnis verfügbar ist. Alle Informationen, die für das Zurückholen benötigt werden, werden im Katalogeintrag gefunden.
- Gesicherte Daten können folgendermaßen zurückgeholt werden:
 - mit der HSMS-Anweisung IMPORT-FILES:

```
//IMPORT-FILES ...,SAVE-FILE=*BY-VOLUME(SAVE-FILE-ID=svid)
```

oder

```
//IMPORT-FILES ...,SAVE-FILE=*BY-PUBLIC-DISK(  
                                SAVE-FILE-ID=svid,PUBSET-ID=cat_id)
```

falls auf Pubset oder Speicherebene S1 gesichert wurde

- mit den ARCHIVE-Anweisungen FILES und RESTORE:

```
FILES RESTORE ...,DIRECTORY=NONE,FROM=(vsn) bzw. FROM=svid,(vsn)
```

3.10.4 Archivdefinition löschen

Eine Archivdefinition kann der HSMS-Verwalter oder der Archiveigentümer mit der HSMS-Anweisung DELETE-ARCHIVE löschen. Es ist aber zu beachten, dass nur die Definition des Archivs aus der HSMS-Steuerdatei gelöscht wird.

Die Archivdefinition kann nur gelöscht werden, wenn keine Aufträge mehr für das Archiv in der Auftragsdatei warten.

Mit DELETE-ARCHIVE stehen außerdem folgende Funktionen zur Verfügung:

- **Abtrennen des Schattenarchivs:** Die Verbindung zwischen einem Archiv und dem zugehörigen Schattenarchiv wird aufgehoben und das Schattenarchiv wird zu einem Backup- oder Langzeit-Archiv gemacht. Die Attribute des ursprünglich zugeordneten Archivs werden dabei für das neue Archiv übernommen.
- **Ersetzen des Archivverzeichnisses:** Die Definition des Schattenarchivs wird gelöscht und das Archivverzeichnis des Schattenarchivs wird für das zugeordnete Backup- oder Langzeit-Archiv übernommen. Das Original-Archivverzeichnis bleibt ohne Verknüpfung zu einem Archiv erhalten.

DELETE-ARCHIVE löscht nicht das Archivverzeichnis sowie die darin verwalteten Dateien. Das Archivverzeichnis kann mit dem BS2000-Kommando DELETE-FILE gelöscht werden.

Versehentlich gelöschte Archivdefinition wiederherstellen

Die Attribute eines Archivs werden zusätzlich im Archivverzeichnis abgespeichert. Nach einem versehentlichen Löschen der Archivdefinition kann das Archiv mit der Anweisung CREATE-ARCHIVE aus dem Archivverzeichnis wiederhergestellt werden:

```
//CREATE-ARCHIVE ARCHIVE-NAME=... ,  
    DIRECTORY-NAME=<directory>(NEW-DIRECTORY=*NO(RECONSTRUCT-ARCHIVE=*YES))
```

3.11 Management-Klassen

Eine Management-Klasse ist eine benannte Sammlung von Attributen, die der HSMS-Verwalter festlegt. Sie wird verwendet, um die Backup-, Versions-Backup- und Migrations-Bearbeitung in einem SM-Pubset unter HSMS-Kontrolle zu steuern.

3.11.1 Zuweisung einer Management-Klasse

Einem einzelnen Objekt (Datei, Jobvariable oder Dateigenerationsgruppe) kann eine Reihe von Attributen zugeteilt werden, indem ihm der Name einer Management-Klasse zugewiesen wird. Die Management-Klasse legt für dieses Objekt eine Reihe von Attributen für Backup und Migration fest. Eine Management-Klasse kann nur Objekten zugewiesen werden, die sich auf SM-Pubsets befinden. Für Objekte auf einer Privatplatte, einem Band oder einem SF-Pubset ist dies nicht möglich.

Einer *Datei* kann mit den DVS-Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES eine Management-Klasse zugewiesen werden. Für eine *Jobvariable* stehen dazu die DVS-Kommandos CREATE-JV und MODIFY-JV-ATTRIBUTES zur Verfügung, für eine *Dateigenerationsgruppe* die DVS-Kommandos CREATE-FILE-GROUP und MODIFY-FILE-GROUP-ATTRIBUTES.

3.11.2 Attribute einer Management-Klasse

Alle Attribute einer Management-Klasse (für Backup, Versions-Backup und Migration) sind in jeder Management-Klasse enthalten. Wenn einem Objekt der Name einer Management-Klasse zugewiesen wird, wird diesem Objekt damit die gesamte HSMS-Strategie zugewiesen.

Man unterscheidet folgende Attribute:

1. Attribute, die Voraussetzungen festlegen, damit Objekte eines SM-Pubsets für Backup oder Migration infrage kommen (z.B. das Attribut UNUSED-DAYS für Migration).
2. Attribute, die das gewünschte Ergebnis eines Backups, eines Versions-Backups oder einer Migration festlegen (z.B. das Attribut TO-STORAGE für Migration).

Im Folgenden sind alle Attribute aufgeführt. Die möglichen Werte eines Attributs finden Sie im Handbuch „HSMS Bd. 2“ [1] bei der HSMS-Anweisung CREATE-MANAGEMENT-CLASS.

- **Attribut für Backup**

RETENTION-PERIOD (Typ 2)

Minimale Schutzfrist der Sicherungsdatei, die das Objekt enthält.

- **Attribute für die Migration von der S0-Ebene (nicht für Jobvariablen)**

UNUSED-DAYS (Typ 1)

Minimale Anzahl der Tage, die ein Objekt unbenutzt sein muss, damit es für die Migration auf eine Hintergrundebene infrage kommt.

MINIMUM-SIZE (Typ 1)

Minimale Anzahl der PAM-Seiten, die ein Objekt haben muss, damit es für die Migration auf eine Hintergrundebene infrage kommt.

MAXIMUM-SIZE (Typ 1)

Maximale Anzahl der PAM-Seiten, die ein Objekt haben darf, damit es für die Migration auf eine Hintergrundebene infrage kommt.

MINIMUM-EXTENTS (Typ1)

Minimale Anzahl der Extents, die eine Datei haben muss, damit sie für die Migration auf eine Hintergrundebene infrage kommt.

TO-STORAGE (Typ 2)

Hintergrundebene, auf die das Objekt migriert werden soll.

- **Attribute für die Migration von der S1-Ebene**

MINIMUM-DAYS-ON-S1 (Typ 1)

Minimale Anzahl der Tage, die ein Objekt auf S1 sein muss, damit es für eine weitere Migration nach S1 oder S2 infrage kommt.

MAXIMUM-DAYS-ON-S1 (Typ 1)

Maximale Anzahl der Tage, die ein Objekt auf S1 sein darf, damit es für eine weitere Migration nach S2 infrage kommt.

MINIMUM-SIZE (Typ 1)

Minimale Anzahl der PAM-Seiten, die ein Objekt haben muss, damit es für die Migration nach S1 oder S2 infrage kommt.

TO-STORAGE (Typ 2)

Hintergrundebene, auf die das Objekt migriert werden soll.

- **Attribute für Versions-Backup**

NUM-OF-BACKUP-VERS (Typ 2)

Die Anzahl der Backup-Versionen, die im Versions-Backup-Archiv aufbewahrt werden.

3.11.3 Verwendung der Management-Klassen

Eine Management-Klasse kann mit der HSMS-Anweisung CREATE-MANAGEMENT-CLASS definiert werden. Die Werte der Attribute können mit der HSMS-Anweisung SHOW-MANAGEMENT-CLASS-ATTRIBUTES angezeigt und mit MODIFY-MANAGEMENT-CLASS-ATTRIBUTES geändert werden.

Nicht-privilegierte Benutzer dürfen nur die HSMS-Anweisung SHOW-MANAGEMENT-CLASS-ATTRIBUTES benutzen.

Bei SM-Pubsets kann nur dann mit Management-Klassen gearbeitet werden, wenn HSMS auf dem Master-Host gestartet wird.

Um eine Sicherung mit Management-Klassen zu steuern, muss der HSMS-Verwalter den Namen einer Management-Klasse in der HSMS-Anweisung BACKUP-FILES angeben; bei einer Migration muss er den Namen einer Management-Klasse in der HSMS-Anweisung MIGRATE-FILES angeben; beim Versions-Backup muss der Systemverwalter den Namen der Management-Klasse bei der HSMS-Anweisung BACKUP-FILE-VERSIONS angeben. Nur die Objekte, denen der Name dieser Management-Klasse zugewiesen wurde, werden für die Bearbeitung ausgewählt. Durch dieses Verfahren lässt sich leicht ein Aggregat definieren. Unter einem Aggregat versteht man eine Reihe von Objekten, die HSMS alle auf dieselbe Art bearbeitet.

Die Operanden der HSMS-Anweisung, die einem Attribut entsprechen, müssen auf den Wert *STD gesetzt sein. Dadurch wird der Wert für diese Operanden der Management-Klasse entnommen.

Zu gegebener Zeit führt der HSMS-Verwalter die erforderlichen HSMS-Anweisungen aus. Die Operandenwerte dieser HSMS-Anweisungen werden dann durch die entsprechenden Attribute des Typs 1 oder 2 der betroffenen Management-Klasse ersetzt.

Beachten Sie, dass in HSMS V11.0A und niedriger Management-Klassen keine Attribute für Versions-Backup enthalten. Dateien, denen solche Management-Klassen zugewiesen wurden, werden von BACKUP-FILE-VERSIONS so bearbeitet, als wenn sie NUM-OF-BACKUP-VERS = 0 hätten, d.h. die Dateien werden ignoriert, falls sie nie zuvor in einem Versions-Backup-Archiv gesichert worden sind. Wenn sie aber bereits im Versions-Backup-Archiv gesichert wurden, wird im Archivverzeichnis für NUM-OF-BACKUP-VERS der Wert 0 eingetragen. Es wird daher empfohlen, Management-Klassen mit Versionsattributen zu versorgen, wenn diese in Zusammenhang mit Versions-Backup verwendet werden (siehe MODIFY-MANAGEMENT-CLASSES).

Die in HSMS V12.0A erstellten Management-Klassen können auch in niedrigeren Versionen ohne Einschränkungen ihrer Funktionalität verwendet werden.

3.11.4 Schutz der Management-Klassen

Wenn das Softwareprodukt GUARDS im Einsatz ist, können Management-Klassen durch Guard-Profile geschützt werden. Dadurch lässt sich die Benutzung auf bestimmte Management-Klassen einschränken.

Ein Guard-Profil muss zuerst mit GUARDS definiert werden, bevor es der HSMS-Verwalter einer Management-Klasse zuordnen kann. Für die Management-Klasse und das Guard-Profil muss dieselbe Umgebung verwendet werden.

Nähere Informationen zu GUARDS finden Sie im Handbuch „SECOS“ [\[16\]](#).

3.12 Directory-Dateien mit DIRCONV bearbeiten

Mit dem Konvertierungsprogramm DIRCONV (Task Unprivileged) können Sie Informationen, die in einer Directory-Datei enthalten sind, ohne SAVE-/RESTORE-Lauf konsistent ändern. Mit DIRCONV können Sie auch mehrere Directory-Dateien mischen.

3.12.1 DIRCONV-Funktionen

DIRCONV bietet Ihnen folgende Funktionen:

- Mischen von Directory-Dateien
- Konvertieren von Directory-Dateien
- Umbenennen von Katalogkennungen
- Entfernen von Katalogkennungen
- Ausgeben von Katalogkennungen
- Aktualisieren des Volume-Katalogs
- Entfernen von nicht katalogisierten Dateien
- Konvertieren von Repository-Dateien vom alten in das neue Format
- Reorganisieren von Directory-Dateien oder Repository-Dateien

DIRCONV läuft unter

- der Benutzerkennung TSOS der Systembetreuung.
- jeder anderen Benutzerkennung.

Es können nur Directory-Dateien bearbeitet werden, in denen keine Dateien oder Jobvariablen von fremden Benutzern verzeichnet sind. Wenn Dateien oder Jobvariablen von fremden Benutzern in einer Directory-Datei enthalten sind, wird der Lauf mit folgender Meldung abgebrochen:

```
RECORDS OF FOREIGN USER FOUND. RUN TERMINATED
```

Bei einer kennwortgeschützten Directory-Datei müssen Sie das Kennwort angeben (dies gilt auch für die Kennung TSOS).

3.12.1.1 Mischen von Directory-Dateien

Mit der DIRCONV-Anweisung MERGE-DIRECTORIES (siehe "[MERGE-DIRECTORIES Mischen von Directory-Dateien](#)") können Sie mehrere Directory-Dateien in eine einzige neue Directory-Datei mischen. Diese neue Directory-Datei enthält dann alle Informationen, die zum Wiederherstellen, Wiederauffinden und Zurückholen von Dateien, die mit einer der ursprünglichen Directory-Dateien gesichert waren, nötig sind. Mit dieser neuen Directory-Datei können Sie weiterhin Sicherung, Langzeitarchivierung und Migration durchführen. Directory-Dateien können jedoch nicht zusammengeführt werden.

Beim Mischvorgang kann es zu Namenskonflikten kommen:

- Wenn `SILENT-RUN=*NO` angegeben ist, wird bei einem Namenskonflikt die Bearbeitung unterbrochen und eine Fehlermeldung auf `SYSOUT` ausgegeben.
- Wenn `SILENT-RUN=*YES` angegeben ist, überprüft DIRCONV das komplette Eingabe-Directory auf Namenskonflikte und gibt diese in eine Liste aus.

Der MAREN-Katalog wird nicht automatisch aktualisiert. Der Name der Directory-Datei wird in den Einträgen der Datenträger, die zur ursprünglichen Directory-Datei gehören, nicht geändert. Deshalb müssen Sie bei einem erfolgreich verlaufenen Mischvorgang die Anweisung `UPDATE-VOLUME-CATALOG` (siehe "[UPDATE-VOLUME-CATALOG Aktualisieren des Volume-Katalogs](#)") verwenden, um einen konsistenten Zustand herzustellen.

Hinweis

Die umgekehrte Funktion, das Aufsplitten einer Directory-Datei in mehrere Directory-Dateien, wird nicht angeboten.

3.12.1.2 Konvertieren von Directory-Dateien

Mit der DIRCONV-Anweisung SET-CATID (siehe "[SET-CATID Konvertieren von Directory-Dateien](#)") können Sie Directory-Dateien konvertieren.

Für ARCHIVE-Läufe mit Directory-Datei gilt:

- Wenn der erste Sicherungslauf mit `PARAM CATID=YES` durchgeführt wurde (zur Unterstützung der MPVS-Funktion), müssen alle Folgeläufe ebenfalls mit `PARAM CATID=YES` durchgeführt werden.
- Wenn der erste Sicherungslauf mit `PARAM CATID=NO` durchgeführt wurde, müssen alle Folgeläufe ebenfalls mit `PARAM CATID=NO` durchgeführt werden.

Mit DIRCONV können Sie Directory-Dateien von `CATID=NO` nach `CATID=YES` konvertieren. Eine Konvertierung in umgekehrter Richtung ist nicht möglich.

Wenn die Directory-Datei bereits im Modus `CATID=YES` ist, wird der Lauf mit folgender Meldung abgebrochen:

```
DIRECTORY IS ALREADY CONVERTED
```

Dies ist auch der Fall, wenn die Directory-Datei mit der ARCHIVE-Anweisung POOL erzeugt wurde und mit ihr noch keine Sicherungsläufe durchgeführt wurden. Eine solche Directory-Datei können Sie sofort im Modus `CATID=YES` benutzen.

Mit einer konvertierten Directory-Datei können Sie Sicherungsversionen, die vor der Konvertierung erzeugt wurden, wegen des unterschiedlichen CATID-Modus nicht mit der ARCHIVE-Anweisung CONTINUE fortsetzen. Diese Sicherungsversionen erscheinen im Report eines INQUIRE-Laufs mit `SV=...` mit dem Zusatz

```
CREATED BEFORE DIRECTORY CONVERSION
```

Sonst verhalten sich konvertierte Directory-Dateien wie solche, die im Modus `CATID=YES` erstellt wurden; insbesondere muss jeder Fortsetzungslauf im Modus `CATID=YES` durchgeführt werden.

Zu Beginn des eigentlichen Konvertierungslaufs wird die Benutzerkennung, unter der DIRCONV gestartet wurde, und die dazugehörige Standard-Katalogkennung ausgegeben. DIRCONV gibt auch eine Meldung aus, wenn die Directory-Datei weder Dateien noch Jobvariablen enthält. In diesem Fall wird die Directory-Datei trotzdem konvertiert.

Bei einem Konvertierungslauf unter TSOS werden alle verzeichneten Benutzerkennungen zusammen mit der von DIRCONV bestimmten Katalogkennung ausgegeben, und zwar in der Reihenfolge, in der die Benutzerkennungen in der Directory-Datei vorkommen. Zusätzlich wird auf SYSLST jede Benutzerkennung protokolliert, der die Standard-Katalogkennung von TSOS zugeordnet wurde, weil sie zum Zeitpunkt des DIRCONV-Laufs gelöscht war.

Wenn ein Datei- oder JV-Name einschließlich der neuen Katalog- und Benutzerkennung länger als 54 Zeichen ist, wird folgende Meldung ausgegeben:

```
***FILENAME TOO LONG: $userid.filename
```

bzw.

```
***JVNAME TOO LONG: $userid.jvname
```

Der Datei- bzw. JV-Name wird in das neue Directory mit der Dummy-Katalogkennung S eingetragen. Somit hat der Anwender nachträglich diese Einträge mittels //RENAME-CATID zu bearbeiten.

Eine Meldung wird auch ausgegeben, wenn die Directory-Datei keine Sicherungsversion enthält.

Das Ende der Konvertierung wird durch die Meldung DIRECTORY CONVERSION FINISHED angezeigt.

Wenn ein DIRCONV-Lauf nicht unter TSOS abläuft, entsteht eine Datei mit dem Namen S.tsn.DIRCONV.@.VNOCRID, falls diese Datei nicht schon zufällig vorhanden ist. Sie können diese Datei gegebenenfalls löschen.

Arbeitsweise von DIRCONV beim Konvertieren in den Modus CATID=YES

DIRCONV fügt in jeden FILES- bzw. JOBVAR-Satz die Standard-Katalogkennung der zugehörigen Benutzerkennung ein.

Wenn die Konvertierung unter TSOS abläuft, werden die Benutzerkennungen, die nicht im Benutzerkatalog gefunden wurden, zusätzlich auf SYSLST ausgegeben.

Hinweise für die Systembetreuung

- Wenn eine Benutzerkennung zum Zeitpunkt des DIRCONV-Laufs nicht im Benutzerkatalog eingetragen ist (gelöscht), kann keine Standard-Katalogkennung ermittelt werden. Die FILES- bzw. JOBVAR-Sätze der Benutzerkennung werden dann mit der Standard-Katalogkennung von TSOS ergänzt.
- Während eines DIRCONV-Laufs sollten die Standard-Katalogkennungen der betroffenen Benutzer nicht geändert werden.
- Vorhandene SVID-Sätze werden als „vor der Konvertierung erzeugt“ gekennzeichnet.
- Die Directory-Datei erhält das Kennzeichen „nur im Modus CATID=YES zu benutzen“.
Beim Konvertieren von Directory-Dateien aus älteren BS2000-Versionen können Dateinamen zu lang werden, wenn ihnen eine Katalogkennung vorangestellt wird. Insgesamt darf der Dateiname einschließlich der Katalog- und Benutzerkennung nicht länger als 54 Zeichen sein.
Für die Länge der Dateinamen gilt dann Folgendes:
 - Wenn die Katalogkennung 1 Zeichen lang ist:
Der Dateiname darf ohne Katalog- und Benutzerkennung nur 41 Zeichen lang sein.
 - Wenn die Katalogkennung länger als 1 Zeichen (2 bis 4 Zeichen) ist:
Die maximal mögliche Länge von 41 Zeichen verringert sich, wenn bei einer mehrstelligen Katalogkennung die Summe aus Länge der Katalogkennung (einschließlich Doppelpunkte) und Länge der Benutzerkennung (einschließlich \$-Zeichen und Punkt) die Länge von 13 Zeichen überschreitet (*.catid:\$userid*).

Wenn DIRCONV beim Konvertieren einen zu langen Dateinamen findet, wird „S“ als Dummy-Katalogkennung eingesetzt. Wenn dadurch der Dateiname in der Directory-Datei nicht mehr eindeutig ist, bricht DIRCONV den Lauf ab. Auf SYSOUT wird dann der Text COPY ERROR, DMS051A ausgegeben.

Mit der Directory-Datei lassen sich keine Dateien mit einem zu langen Dateinamen einlesen.

3.12.1.3 Umbenennen von Katalogkennungen

Mit der DIRCONV-Anweisung RENAME-CATID (siehe "[RENAME-CATID Umbenennen von Katalogkennungen](#)") können Sie eine bestimmte Katalogkennung oder alle Katalogkennungen in einer Directory-Datei umbenennen. Wenn eine bestimmte Katalogkennung umbenannt werden soll, können Sie zusätzlich angeben, ob nur eine bestimmte Benutzerkennung oder alle Benutzerkennungen auf diesem Pubset geändert werden sollen.

Die Anweisung kann in folgenden Fällen verwendet werden:

- Es wird ein SM-Pubset erzeugt, der aus verschiedenen Pubsets besteht, die früher mit einer gemeinsamen Directory-Datei gesichert wurden. Alle Katalogkennungen in dieser Directory-Datei müssen dann in eine einzige gemeinsame SM-Pubset-Kennung umbenannt werden.
- Wenn die Standard-Katalogkennung einer Benutzerkennung geändert wird, müssen die zugehörigen Dateien auf die neue Katalogkennung übertragen werden – unabhängig davon, ob die Dateien migriert sind oder nicht. Die Information über diese Dateien, die in der Directory-Datei enthalten ist, muss konvertiert werden, damit der Wechsel der Katalogkennung berücksichtigt wird.
- Wenn ein Datei- oder JV-Name einschließlich der neuen Katalog- und Benutzerkennung länger als 54 Zeichen ist, wird folgende Meldung ausgegeben:

```
***FILENAME TOO LONG: $userid.filename
```

bzw.

```
***JVNAME TOO LONG: $userid.jvname
```

Der Datei- bzw. JV-Name wird in das neue Directory mit dem Dummy-Katalogkennung S eingetragen. Somit hat der Anwender nachträglich diese Einträge mittels //RENAME-CATID zu bearbeiten.

3.12.1.4 Entfernen einer Katalogkennung

Mit der DIRCONV-Anweisung REMOVE-CATID (siehe "[REMOVE-CATID Entfernen von Katalogkennungen](#)") können Sie aus einer Directory-Datei Informationen über alle Dateien, die auf einem bestimmten Pubset liegen, entfernen. Dateien, die auf anderen Pubsets liegen, sind davon nicht betroffen und können weiterhin rekonstruiert werden.

Durch diese Funktion lassen sich Namenskonflikte verhindern, die beim Mischen von Directory-Dateien oder beim Umbenennen von Katalogkennungen auftreten könnten.

Eine PURGE-Anweisung (ARCHIVE-Lauf) oder eine MODIFY-ARCHIVE-Anweisung (HSMS-Lauf) ist weiterhin möglich und muss auch erteilt werden, um die entsprechenden Datenträger freizugeben. Dies ist sogar dann erforderlich, wenn alle Dateien, die sich auf diesen Datenträgern befinden, aus der Directory-Datei entfernt wurden.

In der Regel sollte eine Katalogkennung erst dann entfernt werden, nachdem die gesicherten Dateien mit der HSMS-Anweisung COPY-SAVE-FILE in ein anderes Archiv kopiert wurden, wobei eine andere Directory-Datei verwendet werden muss.

3.12.1.5 Anzeigen von Katalogkennungen

Mit der DIRCONV-Anweisung SHOW-CATID (siehe "[SHOW-CATID Ausgeben von Katalogkennungen](#)") können Sie Katalogkennungen ausgeben, die in einer vorhandenen Directory-Datei enthalten sind. Diese Information kann wichtig sein, um eventuelle Namenskonflikte vor dem Mischen von Directory-Dateien feststellen zu können.

3.12.1.6 Aktualisieren des Volume-Katalogs

Mit der DIRCONV-Anweisung UPDATE-VOLUME-CATALOG (siehe "[UPDATE-VOLUME-CATALOG Aktualisieren des Volume-Katalogs](#)") können Sie Inkonsistenzen zwischen dem MAREN-Katalog und einer ARCHIVE-Directory-Datei in Bezug auf den Directory-Namen beheben: Alle Datenträger, die zu einer Directory-Datei gehören, erhalten in ihrem Katalogeintrag denselben Directory-Namen.

Für jeden Datenträger, der in der Directory-Datei eingetragen ist, die beim Operanden NEW-DIRECTORY-NAME angegeben wurde, wird in MAREN folgende MAREN-Anweisung abgesetzt:

```
//MODIFY-VOLUME-ATTRIBUTES VOL=... ,DIR-NAME=...
```

Außerdem wird für jede Directory-Datei, die der MAREN-Administrator beim Operanden DIRECTORY-NAMES angegeben hat, in MARENADM folgende MAREN-Anweisung abgesetzt:

```
//MODIFY-VOLUME-ATTRIBUTES VOLUME=*ALL,SELECT=FREE(  
    ARCHIVE-USAGE=alter_dateiname,NEW-ARCHIVE-USAGE=neuer_dateiname)
```

Dadurch wird der Pool der vorreservierten freien Datenträger dem neuen Directory-Namen zugewiesen.

Diese Funktion ist wichtig, um Diskrepanzen beseitigen zu können, die zum Beispiel durch das Mischen von mehreren Directory-Dateien in eine neue Directory-Datei entstanden sind.

Hinweise

- Wenn der MAREN-Administrator eine oder mehrere alte Directory-Namen vergessen hat, muss er die Vorreservierung ändern, indem er für die fehlenden Namen dieselbe MARENADM-Anweisung erteilt.
- Directory-Namen für Datenträger, die in der neuen Directory-Datei eingetragen sind, müssen nicht aktualisiert werden, wenn die alten Directory-Namen vergessen wurden.

3.12.1.7 Entfernen von nicht katalogisierten Dateien

Mit der DIRCONV-Anweisung REMOVE-UNCATALOGED-FILES (siehe "[REMOVE-UNCATALOGED-FILES Entfernen von nicht katalogisierten Dateien](#)") können Sie aus einer Directory-Datei die Dateien entfernen, die nicht mehr katalogisiert sind. Dadurch lassen sich beispielsweise Namenskonflikte verhindern, wenn mehrere Directory-Dateien gemischt werden. Für jede Datei, die entfernt wurde, wird eine Meldung ausgegeben.

Wenn auf eine der Katalogkennungen nicht zugegriffen werden kann, bleiben alle Dateien in der Directory-Datei erhalten. Die Bearbeitung wird dann mit der nächsten Katalogkennung fortgesetzt, die in der Directory-Datei eingetragen ist.

3.12.1.8 Reorganisieren von Directory- und Repository-Dateien

Mit der DIRCONV-Anweisung REORGANIZE-DIRECTORY können Directory- und Repository-Dateien reorganisiert werden.

Dabei werden Lücken in Datensätzen entfernt, die durch das Löschen von Sicherungsversionen entstanden sein können (mit //MODIFY-ARCHIVE ... SAVE-FILE=*DELETE). Dies ist nötig, um das Überschreiten der maximalen Anzahl von Sätzen für eine Datei und/oder JV zu verhindern. Falls die maximale Anzahl (256) für eine Datei oder JV erreicht wird, können für diese Datei und/oder Jobvariable keine weiteren Informationen in das Verzeichnis geschrieben werden, was mit der Meldung ARC0176 angezeigt wird.

Die maximale Anzahl kann insbesondere dann erreicht werden, wenn dieselben Dateien immer wieder im selben Archiv gesichert werden. Falls in diesem Fall regelmäßig alte Sicherungsversionen gelöscht werden, kann die Reorganisation der Directory-Datei die Anzahl der Datensätze erheblich reduzieren und somit einen Überlauf der Directory-Datei verhindern.

Daher wird empfohlen, regelmäßig eine Reorganisation mit der DIRCONV-Anweisung REORGANIZE-DIRECTORY durchzuführen.

Bei der Reorganisation wird eine Kopie der ursprünglichen Directory-Datei angelegt. Die Kopie kann nach erfolgreicher Reorganisation an Stelle der ursprüngliche Directory-Datei verwendet werden.

3.12.1.9 Konvertieren von Repository-Dateien

Mit der DIRCONV-Anweisung REORGANIZE-DIRECTORY können Repository-Dateien von dem alten Format (HSMS < V7.0) in das neue Format (ab HSMS V7.0) konvertiert werden. Das neue Format bietet deutlich verbesserte Performance für den Sicherungsbetrieb mit langen Pfadnamen von Knoten-Dateien (Unix). Alle Sicherungsdateien bleiben dabei unverändert. Im zugehörigen Node-Archiv muss nur das bisherige Verzeichnis ausgetauscht werden durch das neue Verzeichnis aus der DIRCONV-Konvertierung.

3.12.2 Ein-/Ausgaben und Aufruf von DIRCONV

Die Ausgaben von DIRCONV erfolgen nach SYSOUT.

DIRCONV kann nur Directory-Dateien bearbeiten. Andere Dateiarnten werden zurückgewiesen. Die Directory-Dateien für die Ein- und Ausgabe dürfen nicht identisch sein.

DIRCONV wird aufgerufen mit:

```
/START-DIRCONV
```

oder

```
/START-EXECUTABLE-PROGRAM $DIRCONV
```

DIRCONV überprüft, ob die angegebene Eingabedatei eine Directory-Datei ist. Wenn dies nicht der Fall ist, wird der Lauf beendet und folgende Meldung ausgegeben:

```
FILE EMPTY OR NOT IN CORRECT FORMAT
```

Wenn ein Fehler während eines Systemfunktionsaufrufs oder eines Datenzugriffs auftritt, beendet sich DIRCONV abnormal. Eine entsprechende Meldung wird ausgegeben.

Nach einem DIRCONV-Lauf sollten große Directory-Dateien immer reorganisiert werden.

3.12.3 DIRCONV-Anweisungen

Im folgenden Abschnitt sind die DIRCONV-Anweisungen in alphabetischer Reihenfolge beschrieben.

3.12.3.1 MERGE-DIRECTORIES Mischen von Directory-Dateien

Mit dieser Anweisung können Sie mehrere Directory-Dateien in eine einzige mischen.

MERGE-DIRECTORIES

DIRECTORY-NAMES = list-poss(10): <filename 1..54>
--

,NEW-DIRECTORY-NAME = <filename 1..54>

,SILENT-RUN = <u>*NO</u> / *YES
--

DIRECTORY-NAMES = list-poss(10): <filename 1..54>

Vollqualifizierte Pfadnamen der Directory-Dateien, die gemischt werden sollen.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der neuen Directory-Datei.

SILENT-RUN = *NO / *YES

Bei *NO wird eine Meldung ausgegeben, wenn ein Fehler auftritt. Der Lauf wird beendet.

Bei *YES wird der Lauf nur simuliert: die Eingabe-Directory wird vollständig gelesen, es wird jedoch keine Ausgabedatei erstellt. Jeder Fehler (z.B. Namenskonflikt), der während der Ausführung festgestellt wird, wird protokolliert.

3.12.3.2 REMOVE-CATID Entfernen von Katalogkennungen

Mit dieser Anweisung können Sie aus einer Directory-Datei Informationen über alle oder bestimmte Dateien entfernen, die zu einem beliebigen Sicherungszeitpunkt auf einem angegebenen Pubset waren.

REMOVE-CATID

DIRECTORY-NAME = <filename 1..54>

,**CATID** = <cat-id>

,**USER** = *ALL / <name 1..8>

,**FILE-NAME** = *ALL / *NONE / <filename 1..80 with-wild>

,**JV-NAME** = *ALL / *NONE / <filename 1..80 with-wild>

,**NEW-DIRECTORY-NAME** = <filename 1..54>

DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Directory-Datei, aus der eine Katalogkennung entfernt werden soll.

CATID = <cat-id>

Katalogkennung der Dateien, die entfernt werden sollen.

USER =

Benutzer, deren Dateien aus der Directory-Datei entfernt werden sollen.

USER = *ALL

Alle Benutzer, deren Dateien auf der angegebenen Katalogkennung liegen, sollen entfernt werden.

USER = <name 1..8>

Kennung des Benutzers, dessen Dateien entfernt werden sollen.

FILE-NAME =

Name der Dateien, die aus der Directory-Datei entfernt werden sollen.

FILE-NAME = *ALL

Alle Dateien der angegebenen Benutzer sollen entfernt werden.

FILE-NAME = *NONE

Es wird keine Datei entfernt.

FILE-NAME = <filename 1..80 with-wild>

Der Name der Dateien, die entfernt werden sollen, wird direkt angegeben. Er muss voll- oder teilqualifiziert sein, ohne Katalog- und Benutzerkennung. Dieser Name wird um die Katalog- und Benutzerkennung erweitert, die bei den anderen Operanden angegeben sind.

JV-NAME =

Name der Jobvariablen, die aus der Directory-Datei entfernt werden sollen.

JV-NAME = *ALL

Alle Jobvariablen der angegebenen Benutzer sollen entfernt werden.

JV-NAME = *NONE

Es wird keine Jobvariable entfernt.

JV-NAME = <filename 1..80 with-wild>

Der Name der Jobvariablen, die entfernt werden sollen, wird direkt angegeben. Er muss voll- oder teilqualifiziert sein, ohne Katalog- und Benutzerkennung. Dieser Name wird um die Katalog- und Benutzerkennung erweitert, die bei den anderen Operanden angegeben sind.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der neuen Directory-Datei.

3.12.3.3 REMOVE-UNCATALOGED-FILES Entfernen von nicht katalogisierten Dateien

Mit dieser Anweisung können Sie aus einer Directory-Datei die Dateien entfernen, die nicht mehr katalogisiert sind.

REMOVE-UNCATALOGED-FILES

DIRECTORY-NAME = <filename 1..54>
--

,NEW-DIRECTORY-NAME = <filename 1..54>

,SILENT-RUN = <u>*NO</u> / *YES
--

DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Directory-Datei, aus der die nicht katalogisierten Dateien entfernt werden sollen.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Ausgabe-Directory-Datei.

SILENT-RUN = *NO / *YES

Bei *NO wird eine Meldung ausgegeben, wenn ein Fehler auftritt. Der Lauf wird beendet.

Bei *YES wird der Lauf nur simuliert: die Eingabe-Directory wird vollständig gelesen, es wird jedoch keine Ausgabedatei erstellt. Jeder Fehler (z.B. Namenskonflikt), der während der Ausführung festgestellt wird, wird protokolliert.

3.12.3.4 RENAME-CATID Umbenennen von Katalogkennungen

Mit dieser Anweisung können Sie die Katalogkennung in den Datei- bzw. Jobvariablen-Namen verändern. Durch Angabe einer Katalog- und Benutzerkennung können Sie die Menge der umzubennenden Namen einschränken.

RENAME-CATID
DIRECTORY-NAME = <filename 1..54> ,OLD-CATID = <u>*ALL</u> / *BY-SPECIFICATION (...) *BY-SPECIFICATION (...) CATID = <cat-id> ,USER = <u>*ALL</u> / <name 1..8> ,NEW-CATID = <cat-id> ,NEW-DIRECTORY-NAME = <filename 1..54> ,SILENT-RUN = <u>*NO</u> / <u>*YES</u>

DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Directory-Datei, in der eine Katalogkennung umbenannt werden soll.

OLD-CATID =

Katalogkennung, die in den Datei- und Jobvariablen-Namen der angegebenen Directory-Datei umbenannt werden soll.

OLD-CATID = *ALL

Alle Katalogkennungen sollen in eine einzige gemeinsame Pubset-Kennung umbenannt werden.

OLD-CATID = *BY-SPECIFICATION(...)

Nur die nachfolgende angegebene Katalogkennung soll umbenannt werden.

CATID = <cat-id>

Katalogkennung, die umbenannt werden soll.

USER = *ALL

Die Katalogkennung soll für alle Benutzer umbenannt werden.

USER = <name 1..8>

Die Katalogkennung soll nur für Datei- und Jobvariablen-Namen dieser Benutzerkennung umbenannt werden.

NEW-CATID = <cat-id>

Neue Katalogkennung, die die bei OLD-CATID angegebene Katalogkennung ersetzen soll.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der neuen Directory-Datei.

SILENT-RUN = *NO / *YES

Bei *NO wird eine Meldung ausgegeben, wenn ein Fehler auftritt. Der Lauf wird beendet.

Bei *YES wird der Lauf nur simuliert: die Eingabe-Directory wird vollständig gelesen, es wird jedoch keine Ausgabedatei erstellt. Jeder Fehler (z.B. Namenskonflikt), der während der Ausführung festgestellt wird, wird protokolliert.

3.12.3.5 REORGANIZE-DIRECTORY Reorganisieren einer Repository-Datei

Mit dieser Anweisung können Sie Directory- und Repository-Dateien reorganisieren. Vor der Reorganisation sollten obsoletere Sicherungsversionen bzw. -dateien entfernt werden (ARCHIVE: PURGE bzw. HSMS: MODIFY-ARCHIVE SAVE-FILE=*DELETE).

Sie können mit dieser Anweisung auch eine Repository-Datei des alten Formats (vor HSMS V7.0) in das neue Format (ab HSMS V7.0) konvertieren. Das neue Format bietet deutlich verbesserte Performance für den Sicherungsbetrieb mit langen Pfadnamen von Knoten-Dateien.

REORGANIZE-DIRECTORY

DIRECTORY-NAME = <filename 1..54>

,NEW-DIRECTORY-NAME = <filename 1..54>

DIRECTORY-NAMES = <filename 1..54>

Vollqualifizierter Pfadname der Directory- oder Repository-Datei, die reorganisiert werden soll.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der neuen Directory- oder Repository-Datei. Der Name muss sich von dem im Operanden DIRECTORY-NAME angegebenen unterscheiden.

Das angegebene Directory sollte noch nicht existieren oder zumindest leer sein.

Nach der erfolgreichen Reorganisation einer Directory-Datei wird folgende Meldung ausgegeben:

```
THE DIRECTORY HAS BEEN REORGANIZED
```

Falls keine Reorganisation durchgeführt werden konnte, z.B. weil die Directory-Datei bereits reorganisiert war oder keine leeren Datensätze vorhanden waren, wird folgende Meldung ausgegeben:

```
THE DIRECTORY IS ALREADY REORGANIZED
```

Bei der Reorganisation von Repository-Dateien werden keine Meldungen ausgegeben.

3.12.3.6 SET-CATID Konvertieren von Directory-Dateien

Mit dieser Anweisung können Sie eine Directory-Datei, die mit dem CATID=*NO-Modus erstellt wurde, in den CATID=*YES-Modus konvertieren.

SET-CATID
DIRECTORY-NAME = <filename 1..54> ,NEW-DIRECTORY-NAME = <filename 1..54>

DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Directory-Datei, in der bei allen Dateinamen eine Katalogkennung hinzugefügt werden muss.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Name der neuen Directory-Datei.

3.12.3.7 SHOW-CATID Ausgeben von Katalogkennungen

Mit dieser Anweisung können Sie alle Katalogkennungen ausgeben, die in der angegebenen Directory-Datei enthalten sind.

SHOW-CATID

DIRECTORY-NAME = <filename 1..54>
--

DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der Directory-Datei.

3.12.3.8 UPDATE-VOLUME-CATALOG Aktualisieren des Volume-Katalogs

Mit dieser Anweisung können Sie einen konsistenten Zustand zwischen einem MAREN-Katalog und einer ARCHIVE-Directory-Datei herstellen. Ein MAREN-Administrator kann mit dieser Anweisung die Vorreservierung eines freien Datenträgerpools ändern.

UPDATE-VOLUME-CATALOG
DIRECTORY-NAMES = *NONE / list-poss(10): <filename 1..54> ,NEW-DIRECTORY-NAME = <filename 1..54>

DIRECTORY-NAMES

Gibt an, ob ein Pool mit vorreservierten freien Datenträgern eines alten Directory-Namens dem neuen Directory-Namen neu zugewiesen werden muss.

DIRECTORY-NAMES = *NONE

Es muss kein freier Datenträgerpool neu zugewiesen werden.

DIRECTORY-NAMES = list-poss(10): <filename 1..54>

Vollqualifizierte Pfadnamen der ursprünglichen Directory-Dateien, die zum Erstellen einer neuen Directory-Datei verwendet werden sollen und denen freie Datenträgerpools zugewiesen sind. Nur ein MAREN-Administrator darf hier einen Pfadnamen angeben.

NEW-DIRECTORY-NAME = <filename 1..54>

Vollqualifizierter Pfadname der neuen Directory-Datei.

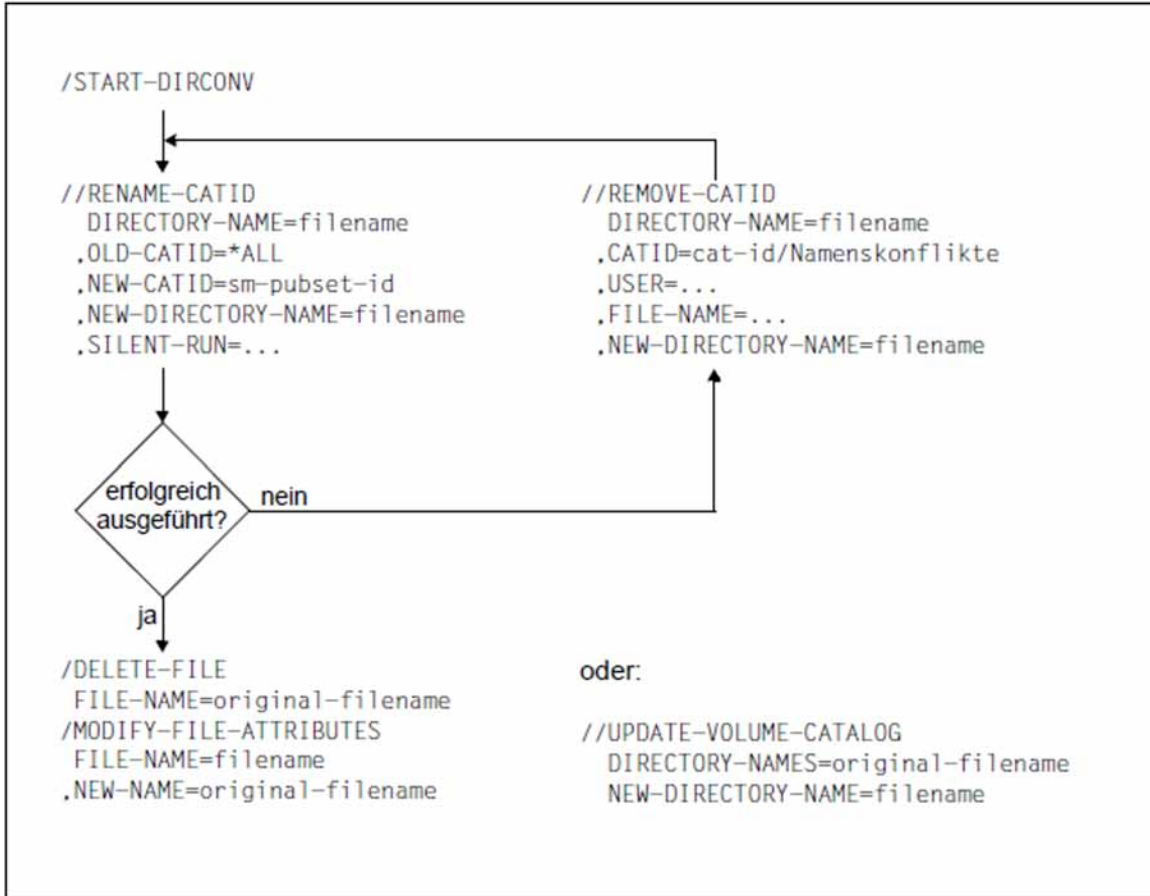
3.12.4 Nutzungsmodelle

In diesem Abschnitt sind drei wesentliche Nutzungsmodelle beschrieben.

3.12.4.1 SM-Pubset erstellen, der aus Pubsets besteht, die bereits mit einer gemeinsamen Directory-Datei gesichert wurden

Alle Katalogkennungen, die in der Directory-Datei enthalten sind, müssen in eine einzige SM-Pubset-Kennung konvertiert werden.

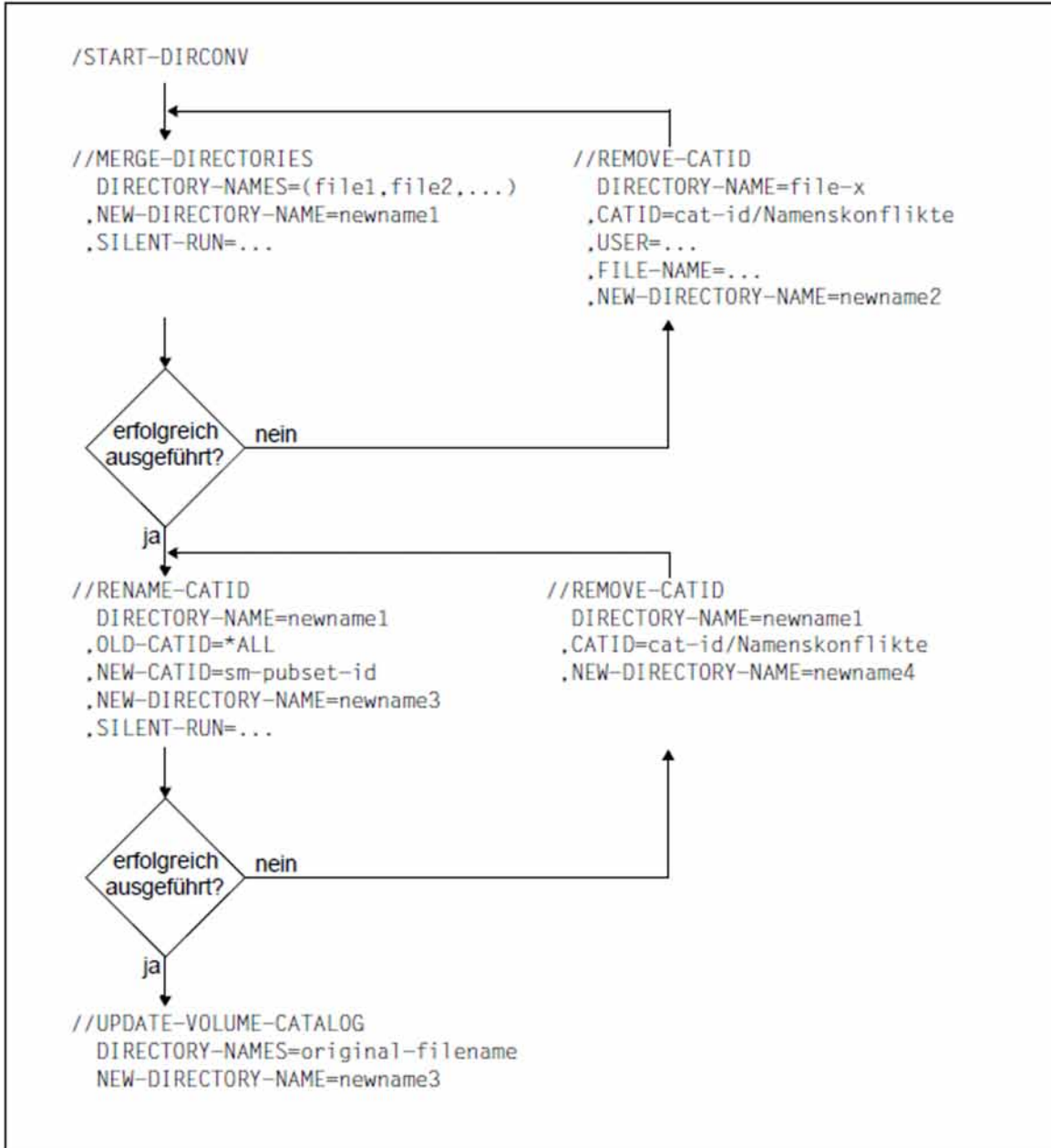
Durchzuführende Tätigkeiten:



3.12.4.2 SM-Pubset erstellen, der aus Pubsets besteht, die bereits mit ihrer eigenen Directory-Datei gesichert wurden

Die Directory-Dateien müssen in eine einzige neue Directory-Datei gemischt und alle Katalogkennungen in eine einzige SM-Pubset-Kennung konvertiert werden.

Durchzuführende Tätigkeiten:



3.12.4.3 Transfer einer Benutzererkennung auf einen anderen Pubset

Wenn eine Benutzererkennung auf einen anderen Pubset gebracht werden soll, müssen alle Dateien und Jobvariablen des Benutzers auf den Ziel-Pubset übertragen werden. Die migrierten Dateien des Benutzers müssen dabei nicht extra zurückgeholt werden.

Durchzuführende Tätigkeiten:

```
/START-DIRCONV
//RENAME-CATID DIRECTORY-NAME=backup.dir (Name des SYSBACKUP-Directory)
,OLD-CATID=*BY-SPECIFICATION(CATID=cat-id1,USER=user-id)
,NEW-CATID=cat-id2
,NEW-DIRECTORY-NAME=new.backup.dir
//RENAME-CATID DIRECTORY-NAME=migrate.dir
,OLD-CATID=*BY-SPECIFICATION(CATID=cat-id1,USER=user-id)
,NEW-CATID=cat-id2
,NEW-DIRECTORY-NAME=new.migrate.dir
//END
```

Nun müssen die ursprünglichen Directory-Dateien für SYSBACKUP (backup.dir) und SYSMIGRATE (migrate.dir) gelöscht werden. Die Directory-Datei *new.backup.dir* muss in *backup.dir* umbenannt werden, *new.migrate.dir* in *migrate.dir*, da diese Namen von den HSMS-Archiven SYSBACKUP bzw. SYSMIGRATE verwendet werden.

Anschließend müssen die Dateien der Benutzererkennung auf der neuen Katalogkennung rekonstruiert werden:

```
/START-HSMS
//RESTORE-FILES FILE-NAMES=:cat-id2:$user-id.
,MIGRATED-FILES=*CATALOG-ONLY
,JV-NAMES=:cat-id2:$userid.
[,ARCHIVE-NAME=*SYSBACKUP]
//END
```

Hinweis

Die Lokalisierungs-Information in den Katalogeinträgen der migrierten Dateien stimmt mit der in der SYSMIGRATE-Directory-Datei überein. Deshalb ist die HSMS-Anweisung REPAIR-CATALOG-BY-RESTORE nicht nötig.

4 Funktionen von HSMS

Im ersten Teil dieses Kapitels sind die Grundfunktionen (Anwendungen) von HSMS beschrieben:

Grundfunktion	DMS-Dateien	Knotendateien
Datensicherung (Backup)	X	X
Langzeitarchivierung (Archival)	X	X
Verdrängung (Migration)	X	-
Datentransfer	X	-
Versions-Backup	X	-

- X Die Grundfunktion wird unterstützt.
- Die Grundfunktion wird nicht unterstützt.

Zusätzlich ist die Möglichkeit zum Kopieren von Sicherungsdateien aufgeführt.

Mit Ausnahme der Verdrängung sind die Grundfunktionen vollständig beschrieben, d.h. sowohl die Anwendung als auch die Verwaltung dieser Funktionen. Die Steuerung und Verwaltung der Verdrängung sind separat im [Kapitel „Verwaltung von HSMS“](#) beschrieben, da sie erheblich umfangreicher als bei den anderen Grundfunktionen sind.

Im zweiten Teil dieses Kapitels sind die unterstützenden Funktionen im Rahmen der Grundfunktionen beschrieben, nämlich:

- die "[Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien](#)"
- die Aktionsanweisungen und deren Bearbeitung ([Abschnitt „Auftragsbehandlung \(Requests\)“](#))
- die Zuweisung der Datenträger und die Steuerung der Bandzugriffe ([Abschnitt „Datenträgerverarbeitung“](#))
- die Unterstützung der verschiedenen Plattenformate und die Konvertierung beim Einspielen ([Abschnitt „Behandlung von BS2000-Platten“](#))
- die Bildschirmmasken und Reporte von HSMS ([Abschnitt „Ausgaben von HSMS“](#))
- die Unterstützung von Aliasnamen ([Abschnitt „Aliasnamen für BS2000-Dateien und Jobvariablen“](#))

Für UNIX-Dateien wird im Folgenden der Begriff „Knotendateien“ verwendet.

Die Knotendateien können auf dem lokalen BS2000-UFS oder auf einem fernen Dateisystem liegen. Ein fernes UNIX-Dateisystem kann direkt über NFS gesichert werden. Für die NFS-Sicherung muss ein fernes UNIX-Dateisystem am speziellen Dateiverzeichnis „HSMS“ des lokalen BS2000-UFS eingehängt sein. Im Folgenden sind die Grundfunktionen für BS2000-Dateien und Jobvariablen einerseits und Knotendateien andererseits in getrennten Abschnitten beschrieben, falls Unterschiede bestehen.

HSMS bearbeitet auch Umgebungen mit SM-Pubsets. Die Funktionen Backup, Versions-Backup und Migration sind für eine Umgebung spezifisch, d.h. sie bearbeiten nur Objekte, die sich in dieser Umgebung befinden. Dagegen ist die Archivierungsfunktion nicht auf eine bestimmte Umgebung beschränkt; sie kann Objekte von verschiedenen Umgebungen (SF- und SM-Pubsets) in einem einzigen Lauf bearbeiten. Wenn aber das Standard-Langzeitarchiv für Langzeitarchivierung verwendet werden soll, dürfen in einer SM-Umgebung keine Langzeitarchive definiert werden, damit die Vermischung der Dateien aus verschiedenen Umgebungen möglich bleibt.

4.1 Datensicherung (Backup)

Unter Datensicherung versteht man das vorbeugende Erstellen und Mitführen von Kopien des Datenbestandes zur Wiederherstellung dieser Daten bei Verlust, sei es auf Grund von Bedienfehlern (z.B. versehentliches Löschen) oder wegen Hardwareausfalls.

In HSMS V12.0A wurde ein neuer Sicherungstyp eingeführt: Versions-Backup.

Je nach Bedarf kann jetzt parallel oder alternativ eine der beiden Sicherungsarten verwendet werden. Sie weisen unterschiedliche Vorteile und Besonderheiten auf:

- **Backup**
Ermöglicht die Sicherung von Dateien und Jobvariablen von verschiedenen Speichern; Voll-, Differenz- und Teilsicherung der Dateien; automatisches Duplizieren der gesicherten Dateien und automatisches Löschen der obsoleten Sicherungsdateien; Anweisungen BACKUP-FILES, BACKUP-NODE-FILES.
- **Versions-Backup**
Gewährleistet, dass immer eine bestimmte Anzahl von Dateiversionen aufbewahrt wird; es werden nur Differenzsicherungen durchgeführt; ein neuer Reorganisationsmechanismus sorgt dafür, dass benötigte Dateiversionen kopiert und obsoletere Dateiversionen gelöscht werden. Anweisungen: BACKUP-FILE-VERSIONS, REORGANIZE-VERSION-BACKUP, CHECK-CATALOGED-FILES

4.1.1 Allgemeine Bemerkungen über die beiden Sicherungstypen

BS2000-Dateien sind Dateien, die im BS2000 verwaltet und bearbeitet werden. Sofern nicht explizit unterschieden wird, umfasst der Begriff „BS2000-Datei“ bei Dateien auf Net-Storage sowohl den Dateityp BS2000 als auch den Dateityp Node-File, der ab BS2000 OSD/BC V10.0 zusätzlich unterstützt wird.

In diesem Abschnitt sind unter dem Begriff „Dateien“ immer BS2000-Dateien zu verstehen.

HSMS sichert Daten logisch.

HSMS kann auch große Dateien (Dateien > 32 GB) sichern. Dazu sind keine besonderen Vorkehrungen erforderlich.

Beim Restaurieren von großen Dateien sind aber Besonderheiten zu berücksichtigen (siehe [Abschnitt „Restaurieren von großen Dateien \(> 32 GB\)“](#)).

Verschlüsselte Dateien werden in verschlüsselter Form gesichert und in der ursprünglichen verschlüsselten Form wieder restauriert. Zum Sichern bzw. Restaurieren muss das zugehörige Crypto-Kennwort nicht angegeben werden. Für weitere Zugriffe auf die restaurierte Datei muss das Crypto-Kennwort wieder angegeben werden und im System müssen die technischen Voraussetzungen für die Dateiverschlüsselung vorhanden sein.

Wenn die HSMS-Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS in einer Umgebung mit SF-Pubsets verwendet werden, können sie nur SF-Pubsets bearbeiten; SM-Pubsets können dann nicht bearbeitet werden.

Wenn die HSMS-Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS in einer Umgebung mit SM-Pubsets verwendet werden, sind sie auf die Umgebung beschränkt, in der sie aufgerufen wurden; d.h. sie beziehen sich genau auf einen SM-Pubset.

Standardmäßig ist festgelegt (siehe //SHOW-HSMS-PARAMETERS, MIGRATION-CONTROL, BACKUP-MANDATORY=YES), dass aus Sicherheitsgründen eine Datei erst dann verdrängt werden kann, wenn der aktuelle Stand der Datei schon mindestens einmal in eine Sicherungsdatei geschrieben wurde.

Von migrierten Dateien können beim Backup mit BACKUP-FILES entweder nur die Katalogeinträge (Sicherungstyp MIGF) oder zusätzlich auch die Daten gesichert werden. Beim Versions-Backup werden grundsätzlich sowohl die Katalogeinträge als auch die Daten gesichert. Siehe dazu den [Abschnitt "Datensicherung und verdrängte Dateien"](#).

Wie eine Sicherungsdatei erweitert werden kann, ist abhängig von ihrer Struktur und von ihrem Ablageort:

- Bei Backup-Archiven mit Single-SVID-Struktur oder bei Sicherungsdateien auf Platte ist das Erweitern der Sicherungsdatei um eine disjunkte Menge beim Schreiben ins Archiv durch Angabe des Operanden SAVE-FILE=*CONTINUE möglich.
- Bei Archiven mit Several-SVID-Struktur kann die Sicherungsdatei mit einer neuen Sicherungsversion fortgesetzt werden (unabhängig davon, ob dieselben Dateien gesichert werden). Die Dateien sind durch ihre Sicherungsversion zu unterscheiden.

4.1.2 Sicherung mit **BACKUP-FILES**

In diesem Abschnitt sind spezifische Eigenschaften des ersten Sicherungstyps beschrieben. Anders gesagt, beschreibt dieser Abschnitt, wie sich Dateien und Jobvariablen mit der HSMS-Anweisung **BACKUP-FILES** sichern lassen.

4.1.2.1 BS2000-Dateien und Jobvariablen sichern

HSMS dient der System- und Anwendungssicherung und erlaubt dabei Voll-, Differenz- und Teilsicherung. HSMS kann auch zum Reorganisieren der Datenbestände auf Plattenspeichern benutzt werden.

Dieser Sicherungstyp verwendet Benutzer- sowie System-Backup-Archive.

Systemsicherungen kann nur der HSMS-Verwalter durchführen.

Eine Datensicherung ist allen Benutzern erlaubt.

Ein nicht-privilegierter Benutzer kann Dateien und Jobvariablen in jedes andere Benutzerarchiv sichern, wenn

- dieses Archiv öffentlich ist (und ACCESS=*WRITE eingestellt wurde)
- die Dateien dem Archiveigentümer gehören und der Auftragserteiler Miteigentümer der Dateien ist.
- für fremde Archive auf deren zugehörige Directories Miteigentümerschaft besteht. Diese dürfen wie eigene benutzt werden. Ausgenommen ist lediglich die Archivverwaltung, also das Ändern der Archiveigenschaften oder Löschen obsoleter Sicherungsdateien.

Ein nicht-privilegierter Benutzer kann Dateien und Jobvariablen seiner eigenen Benutzerkennung restaurieren sowie Dateien und Jobvariablen, bei denen er Miteigentümer ist:

- aus seinen eigenen Backup-Archiven oder aus fremden Archiven, wenn er durch die Miteigentümerschaft Zugriff auf dieses fremde Archiv/Directory hat.
- aus einem öffentlichen Backup-Archiv.

Die Maßnahmen, die der HSMS-Verwalter im Zusammenhang mit der Datensicherung treffen sollte, sind im [Abschnitt „Datensicherung und verdrängte Dateien“](#) beschrieben.

Der Archiveigentümer und der HSMS-Verwalter können mit dem Operanden SAVE-FILE der BACKUP-FILES-Anweisung festlegen, ob

- die zu sichernden Dateien in die Standard-Sicherungsdatei des Archivs geschrieben werden sollen (*STD), falls die Archivdefinition dies zulässt;
- eine neue Sicherungsdatei erstellt werden soll (*NEW);
- eine vorhandene Sicherungsdatei fortgesetzt werden soll (*CONTINUE).

Zum Sichern von Dateien und Jobvariablen dient die HSMS-Anweisung BACKUP-FILES.

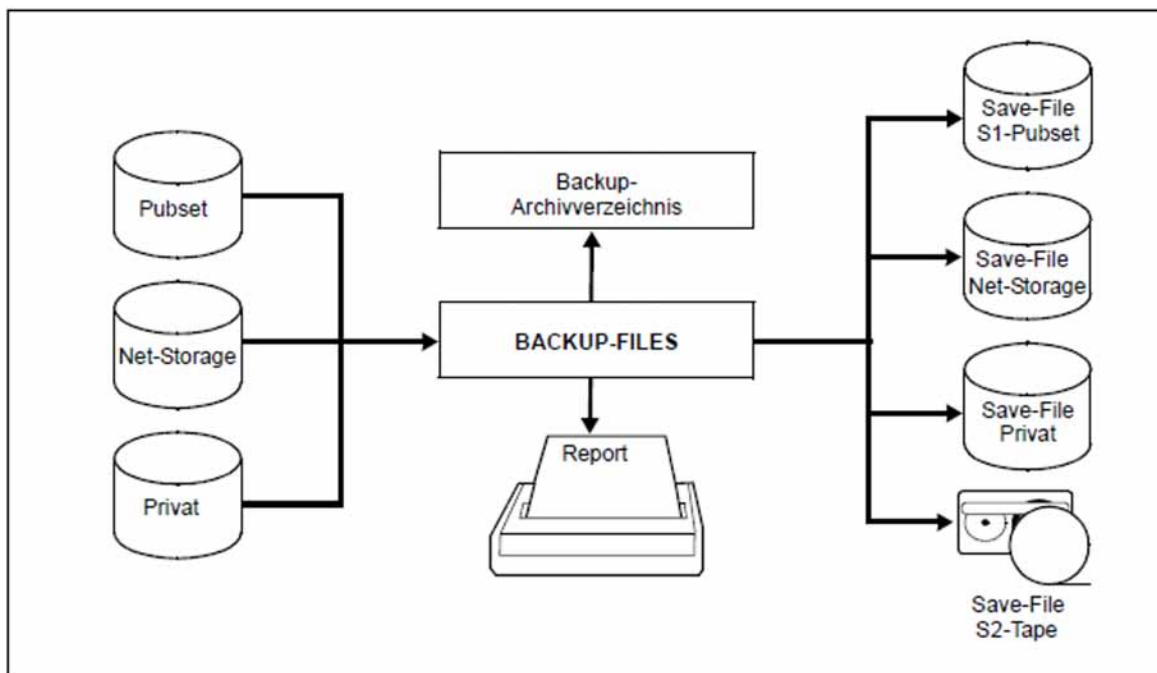


Bild 8: Sichern mit BACKUP-FILES

Die zu sichernden Dateien, die auf Pubsets, Net-Storage oder Privatplatten liegen müssen, können ausgewählt werden nach

- dem Datenträger, auf dem die Dateien liegen (SUPPORT)
- dem Dateityp für Net-Storage-Dateien (FILE-TYPE=*BS2000 / *NODE-FILE)
- der Sicherungsstufe, die den Dateien gegeben wurde (Operand MAXIMUM-BACKUP-CLASS).
- einer Dateinamensliste (siehe [Abschnitt „Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien“](#)).

System-Backup-Archiv

Die gesicherten Daten werden in Backup-Archiven verwaltet. System-Backup-Archive richtet der HSMS-Verwalter für die Systemsicherung von SF-Pubsets und von SM-Pubsets ein. Sie werden in der Regel über den symbolischen Namen SYSBACKUP angesprochen. SYSBACKUP ist das Standard-Backup-Archiv, auf das die HSMS-Anweisung RESTORE-FILES verweist, falls kein anderes Archiv angegeben ist.

Das System-Backup-Archiv muss als öffentliches Archiv mit Leseberechtigung eingerichtet werden (Operand USER-ACCESS=*ALL-USERS/ACCESS=*READ), damit auch nicht-privilegierte Benutzer Daten aus diesem Archiv restaurieren können.

Voll- und Differenzsicherung

Bei einer Vollsicherung werden alle angegebenen Dateien in vollem Umfang gesichert, unabhängig davon, ob sie sich seit der letzten Sicherung geändert haben oder nicht.

Bei einer Datensicherung durch einen HSMS-Verwalter werden die Daten von online-migrierten Dateien mitgesichert, wenn das Sicherungsverfahren entsprechend eingestellt wurde. Dabei werden die Katalogeinträge aus dem Katalog und die Daten von der Migrationsebene übernommen. Weiterhin ist es aber möglich, von migrierten Dateien nur die Katalogeinträge zu sichern.

Für Dateien auf Net-Storage kann ebenfalls gewählt werden, ob die gesamte Datei (Voreinstellung) oder nur der Katalogeintrag gesichert werden soll (Operand SAVE-OPTIONS= *PAR(SAVE-NET-STOR-DATA=*YES/NO)).

Bei einer Differenzsicherung werden nicht alle angegebenen Dateien gesichert. Wenn HSMS feststellt, dass der aktuelle Inhalt der Datei bereits im betreffenden Archiv enthalten ist, wird der Inhalt der Datei nicht gesichert. Die Datei wird allerdings als Cataloged-Not-Saved (CNS) im Archivverzeichnis vermerkt.

Es werden also nur Dateien gesichert, deren Inhalt sich seit der letzten Sicherung geändert hat oder die neu angelegt wurden. Dadurch wird der Zeitaufwand und der Speicherplatzbedarf für eine Systemsicherung erheblich vermindert. HSMS führt die Differenzsicherung standardmäßig durch. Node-Files auf Net-Storage werden – auch im Rahmen von Differenzsicherungen – stets voll gesichert.

Wenn eine Datei als Cataloged-Not-Saved (CNS) im Archivverzeichnis eingetragen ist, wird die Schutzfrist der letzten Vollsicherung dieser Datei überprüft. Wenn das Freigabedatum der letzten Vollsicherung älter als das Freigabedatum der aktuellen Differenzsicherung ist, wird die Schutzfrist der letzten Vollsicherung aktualisiert. Dadurch erhält die letzte Vollsicherung dasselbe Freigabedatum wie die aktuelle Differenzsicherung, d.h. die Vollsicherung verfällt nicht vor den zugehörigen Differenzsicherungen. Folglich kann die Vollsicherung nicht vor allen zugehörigen Differenzsicherungen gelöscht werden, es sei denn, der Benutzer erzwingt das Löschen mit der Angabe FORCE-DELETE in der MODIFY-ARCHIVE-Anweisung.

Falls bei der letzten Vollsicherung nur die Katalogeinträge der migrierten Dateien gesichert wurden (SAVE-DATA=*STD/*S0)), ist dafür Sorge zu tragen, dass auch die Daten der migrierten Dateien gesichert werden (SAVE-DATA=*S1-S0/*S2-S1-S0), damit bei Crash der Migrationsebene die Daten wieder hergestellt werden können. Details zur Behebung von S1-/S2-Crashes siehe [Abschnitt „RESTORE von verdrängten Dateien“](#).

Vollsicherung offline erstellen

Eine Vollsicherung kann entweder von der Speicherebene S0 durchgeführt werden (FROM=*S0), von früheren Sicherungen in einem Archiv (FROM=*ONLY-LATEST-BACKUPS) oder von beidem (FROM=*LATEST-BACKUPS-OR-S0).

Bei einer Sicherung mit FROM=*LATEST-BACKUPS-OR-S0 werden alle Dateien, die derzeit im zugehörigen Archiv gesichert sind, von ihrer Kopie in der Sicherungsdatei gesichert und nicht durch Lesen der Inhalte von der Speicherebene S0. Dadurch wird auf der Speicherebene S0 für eine Vollsicherung die gleiche Zugriffszeit benötigt wie für eine Differenzsicherung: Zuerst werden alle geänderten Dateien von der Speicherebene S0 gesichert. Nachdem dieser Vorgang beendet ist, ist die Speicherebene S0 wieder verfügbar. Anschließend wird die Sicherung mit dem Kopieren von Band zu Band fortgesetzt.

Bei einer Vollsicherung mit FROM=*ONLY-LATEST-BACKUPS wird die komplette Sicherung offline durchgeführt. Es sind keine vorbereitenden online-Schritte auf der Speicherebene S0 nötig, weil die letzte Differenzsicherung als Bezug genommen wird. Alle Dateien, die in dieser Differenzsicherung eingetragen sind (vollständig oder als CNS), werden von ihrer letzten Sicherung gesichert und in einer einzigen neuen Vollsicherung zusammengefasst. Wenn aber in dieser letzten Differenzsicherung Dateien enthalten sind, die nur partiell gesichert wurden (Teilsicherung), dann werden sie übersprungen, weil sie nicht offline rekonstruiert werden können.

Die Vollsicherung von migrierten Dateien mit FROM=*ONLY-LATEST-BACKUPS ist nicht erlaubt. Migrierte Dateien, die bei der letzten Sicherung als MIGF gekennzeichnet wurden, bleiben auch als MIGF bei einer reinen offline-Vollsicherung stehen.

Um eine komplette offline-Sicherung garantieren zu können, die die Speicherebenen S0, S1 und S2 einschließt, muss der vorhergehende letzte Sicherungslauf mit der Angabe SAVE-DATA=*S2-S1-S0 gestartet und ohne Teilsicherung durchgeführt worden sein.

Besonderheiten

Eine Vollsicherung mit dem Operanden LATEST-BACKUPS-OR-S0 läuft genauso wie jede andere Vollsicherung ab, außer der Behandlung der Metadaten (Katalogeintrag) von nicht geänderten Dateien: nicht geänderte Dateien werden nicht von der Speicherebene S0 genommen, sondern aus ihrer letzten Voll- oder Differenzsicherung (einschließlich ihrer Metadaten).

Basis für die zu sichernde Dateimenge sind die auf S0 katalogisierten Dateien; daher sollten auch während Offline-Sicherungen keine Daten auf S0 gelöscht werden.

Bei einer Vollsicherung von S0 oder von der letzten Sicherung im Archiv ist es nicht möglich, eine bestehende Sicherungsdatei fortzusetzen. In diesem Fall muss immer eine neue Sicherungsdatei erstellt werden.

Zwischen zwei Vollsicherungen einer Datei können maximal 255 partielle Sicherungen dieser Datei durchgeführt werden. Bei der 256. Sicherung führt HSMS automatisch eine Vollsicherung für diese Datei durch, auch wenn dies in der HSMS-Anweisung nicht vereinbart wurde.

Gesteuert wird die Art der Sicherung über den Operanden SELECT-FILES der BACKUP-FILES-Anweisung. Die Angabe SELECT-FILES=*MODIFIED-FILES führt zu einer Differenzsicherung, SELECT-FILES=*ALL-FILES zu einer Vollsicherung.

Die Art der Vollsicherung kann über den FROM-Operanden festgelegt werden. Beim Standardwert *S0 werden die Dateien von der Speicherebene S0 gesichert. Wenn der Wert *LATEST-BACKUPS-OR-S0 angegeben ist, wird die Sicherung kopiert, die im Archiv für CNS-Dateien gespeichert ist.

Die Vollsicherung aus Differenzsicherungen funktioniert folgendermaßen:

- Zur Bestandsaufnahme wird der Systemkatalog durchgegangen.
- Alle geänderten Dateien werden von der Speicherebene S0 gesichert.
- Alle unveränderten Dateien werden von Band gesichert.

Teilsicherung

Bei einer Differenzsicherung lässt sich der Sicherungsaufwand durch eine Teilsicherung (Partielle Sicherung) weiter verringern.

Für UPAM-Dateien mit BLOCK-CONTROL-INFO=*NO ist eine Teilsicherung nicht möglich. Eine Ausnahme bilden PLAM-Bibliotheken. Sie können partiell gesichert werden, weil die Zugriffsmethode PLAM über einen eigenen Mechanismus an Stelle des PAM-Schlüssels verfügt, mit dem sie geänderte Teile in Bibliotheken kennzeichnet.

Wie bei der Differenzsicherung werden die Dateien, die sich seit der letzten Sicherung nicht geändert haben, nicht gesichert. Bei den geänderten Dateien prüft HSMS, welche Teile (2-KByte-Blöcke, PAM-Seiten) der Datei sich seit der letzten Vollsicherung (nicht Teilsicherung) geändert haben. Nur diese Teile werden gesichert.

Zum Restaurieren einer teilgesicherten Datei ist also jeweils die letzte Teilsicherung und die letzte Vollsicherung erforderlich.

Wenn eine Datei als „Partially Saved“ im Archivverzeichnis eingetragen ist, wird die Schutzfrist der letzten Vollsicherung dieser Datei überprüft. Wenn das Freigabedatum der letzten Vollsicherung älter als das Freigabedatum der aktuellen Teilsicherung ist, wird die Schutzfrist der letzten Vollsicherung aktualisiert. Dadurch erhält die letzte Vollsicherung dasselbe Freigabedatum wie die aktuelle Teilsicherung, d.h. die Vollsicherung verfällt nicht vor den zugehörigen Teilsicherungen. Folglich kann die Vollsicherung nicht vor allen zugehörigen Teilsicherungen gelöscht werden, es sei denn, der Benutzer erzwingt das Löschen mit der Angabe FORCE-DELETE in der MODIFY-ARCHIVE-Anweisung.

Veranlasst wird eine Teilsicherung mit

```
//BACKUP-FILES SELECT-FILES=*MODIFIED-FILES -  
// (PARTIAL-FILE-SAVE=*LARGE-FILES/*YES)
```

Bei einer Teilsicherung werden entweder alle Dateien partiell gesichert, die im Katalogeintrag als LARGE=*YES bzw. mit SAVED-PAGES=*MODIFIED-PAGES gekennzeichnet sind, oder unabhängig davon Dateien, die eine bestimmte Größe haben. Die Größe wird mit *YES(MINIMUM-SIZE=<integer>) angegeben.

Hinweis

Dateigenerationsgruppen werden immer komplett in eine Sicherungsdatei mit mehreren Sicherungsversionen gesichert. Wenn eine Dateigeneration oder ein Dateigenerationsindex angegeben wird, wird eine Vollsicherung des Index und aller Dateigenerationen durchgeführt.

4.1.3 Versions-Backup

In HSMS V12.0A wurde der neue Sicherungstyp Versions-Backup eingeführt.

Dieser bietet folgende Funktionalität:

1. Sichern und Aufbewahren von mehreren Dateiversionen im Versions-Backup-Archiv für Dateien, die im jeweiligen Pubset katalogisiert sind. Die Anzahl der aufzubewahrenden Versionen legt der Benutzer in den Dateieigenschaften fest.
2. Mechanismus zur Sicherstellung, dass eine gelöschte (bzw. versehentlich gelöschte) Datei im Archiv lange genug in den zuvor gesicherten Versionen aufbewahrt wird, sodass der Benutzer Maßnahmen zum Restaurieren der Daten ergreifen kann.

Anzahl der Backup-Versionen

In BS2000 OSD V11.0B wurde ein neues Dateiattribut NUM-OF-BACK-VERS (0..32) für die Anzahl der Backup-Versionen eingeführt. Es legt fest, wie viele Kopien einer Datei gleichzeitig im Versions-Backup-Archiv aufbewahrt werden sollen.

NUM-OF-BACKUP-VERS = 0 heißt, dass die Datei am Versions-Backup-Verfahren nicht teilnimmt. In der SM-Umgebung kann die Anzahl der aufzubewahrenden Dateien außerdem mit Management-Klassen gesteuert werden. Der Wert in der Management-Klasse dominiert den abweichenden Wert im Katalogeintrag.

Das **Versions-Backup-Archiv** ist ein neuer Archivtyp, der speziell für Versions-Backup eingeführt wurde.

Es gibt nur ein Versions-Backup-Archiv pro Pubset, in dem die gewünschte Anzahl der Dateiversionen aufbewahrt wird. Dies stellt sicher, dass immer eine konsistente Anzahl von Dateiversionen verfügbar ist.

Es gibt somit keine Benutzerarchive für Versions-Backup. Versions-Backups können nur anhand der Systemarchive für Versions-Backup durchgeführt werden. Sie werden sowohl in der SF- als auch SM-Umgebung Pubset-spezifisch definiert und über den symbolischen Namen SYSVERSION angesprochen. Es gibt jedoch kein globales SYSVERSION-Archiv für die SF-Umgebung.

Das Systemarchiv für Versions-Backup muss als öffentliches Archiv mit Schreibzugriff (Operand USER-ACCESS=*ALL-USERS/ACCESS=*WRITE) erstellt werden, um nicht-privilegierten Benutzern Backup und Restore von Daten in diesem Archiv zu ermöglichen. Alternativ kann mit ACCESS=*READ nur der HSMS-Verwalter Sicherungsläufe ausführen. Nicht-privilegierte Benutzer können jedoch Daten aus dem Archiv restaurieren.

Die HSMS-Anweisung BACKUP-FILE-VERSIONS dient zur Sicherung von Dateien in einem Versions-Backup-Archiv.

Der Benutzer kann die Speichermedien angeben, von und auf die Daten geschrieben werden sollen.

Versions-Backup ist nicht möglich für:

- Dateien auf Privatplatten
- Dateigenerationen (FGGs)
- Temporäre Dateien
- Katalogeinträge von Banddateien
- Jobvariablen

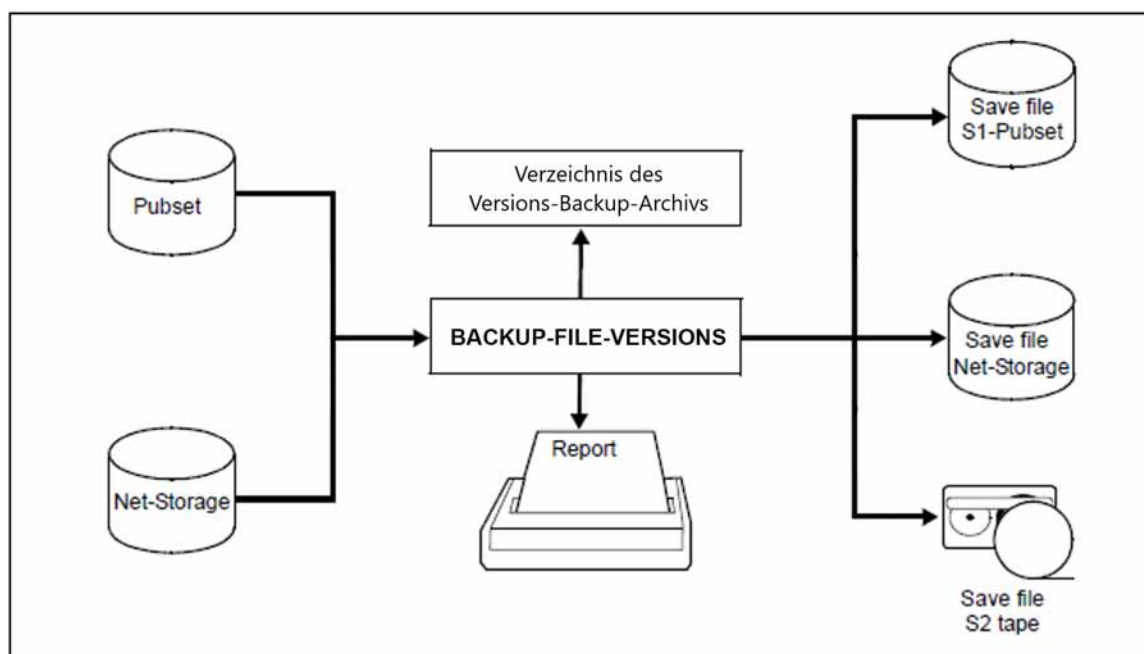


Bild: Versions-Backup mit BACKUP-FILE-VERSIONS

i Die Funktionalität von Versions-Backup steht nur nach der Einführung des Operanden NUMBER-OF-BACKUP-VERSIONS zur Verfügung (ab BS2000 OSD/BC V11.0B) und im Modus *HSMS-V10-COMPATIBLE. In einer inhomogenen BS2000/HSMS-Umgebung wird das Kommando nicht auf den Master-Rechner (Backup-Server) mit BS2000 < 11.0B oder HSMS < 12.0A übertragen. Der Kommandoauftrag wird mit einer entsprechenden Meldung auf dem Host abgebrochen, der den Auftrag initialisiert.

4.1.3.1 Differenzsicherungen beim Versions-Backup

Beim Versions-Backup ist nur die Differenzsicherung möglich: nur geänderte Dateien werden gesichert, die nicht geänderten Dateien werden ignoriert, d.h. es werden keine CNS-Einträge wie beim Backup erzeugt.

Während des Sicherungslaufs wird das Dateiattribut NUM-OF-BACKUP-VERS im entsprechenden Satz des Archivverzeichnisses gespeichert. Anhand dieser Anzahl kann HSMS festlegen, wie viele Dateiversionen aufbewahrt werden sollen sowie welche obsoleten Dateiversionen während des Reorganisationslaufs gelöscht werden sollen.

Falls das Dateiattribut NUM-OF-BACK-VERS geändert wurde, wird der entsprechende Wert im Archivverzeichnis bei der nächsten Sicherung aktualisiert, unabhängig davon, ob die betreffende Datei geändert wurde (also somit gesichert wurde) oder nicht. Das betrifft die beim BACKUP-FILES durch den Operanden FILE-NAMES angegebenen Dateien. Eine Aktualisierung des Wertes kann auch mit der Anweisung CHECK-CATALOGED-FILES durchgeführt werden.

Bei migrierten Dateien werden immer sowohl Katalogeinträge als auch Daten gesichert.

Die Backup-Klasse kann als Kriterium dienen, ob die Datei gesichert werden soll oder nicht.

4.1.3.2 Versions-Backups reorganisieren

Das Reorganisieren wird durchgeführt, um Speicherplatz der entsprechenden Speicherebene (S1 oder S2) zu sparen. Es gibt obsoletere Dateiversionen (Dateiversionen, die nach dem Dateiattribut Anzahl der Backup-Versionen (NUM-OF-BACKUP-VERS) nicht mehr aufbewahrt werden müssen) und die Dateien, die vor einer längeren Zeit aus dem Dateikatalog des Pubsets gelöscht wurden und somit aus dem Versions-Backup-Archiv zu entfernen sind. Das **Reorganisieren** eines Versions-Backup-Archivs wird mit der Anweisung REORGANIZE-VERSION-BACKUP durchgeführt. Die Schritte zum Reorganisieren eines Versions-Backup-Archivs lauten wie folgt:

1. HSMS-Anweisung CHECK-CATALOGED-FILES:

CHECK-CATALOGED-FILES hat zwei Zwecke: die Anweisung aktualisiert den Wert "Anzahl der Sicherungsversionen" (NUM-OF-BACKUP-VERS) in den Dateisätzen des Archivverzeichnisses aus den TSOSCAT-Katalogeinträgen. Außerdem überprüft CHECK-CATALOGED-FILES, ob die im Archivverzeichnis enthaltenen Dateien immer noch auf dem entsprechenden Pubset existieren. Wenn eine Datei aus dem Pubset gelöscht wurde, wird sie als gelöscht vermerkt und das aktuelle Datum als ihr Löschdatum in den Dateisatz des Archivverzeichnisses eingetragen. Es wird empfohlen, dass der HSMS-Verwalter während des Prüflaufs Reporte erstellt, um Benutzer über gelöschte Dateien zu informieren. Der Benutzer kann wiederum die Dateien mit der HSMS-Anweisung SHOW-ARCHIVE überprüfen. SHOW-ARCHIVE informiert den Benutzer, welche Dateien HSMS als von der S0-Ebene gelöscht erkannt hat, wann dies erkannt wurde und ob bereits eine Löschung der Dateien aus dem Archiv vorgemerkt wurde. Falls die Datei versehentlich gelöscht wurde, kann der Benutzer sie wiederherstellen. Falls CHECK-CATALOGED-FILES feststellt, dass die Datei ein Löschdatum hat und nun inzwischen wieder im TSOSCAT des entsprechenden Pubsets steht, wird das Löschdatum aus dem Datensatz entfernt.

2. MODIFY-ARCHIVE FILE = *MARK-FOR-DELETION(...)

Damit eine Datei während der Reorganisation aus dem Archiv gelöscht werden kann, muss sie erst zum Löschen "vorgemerkt" werden. Diese Vormerkung erfolgt mit MODIFY-ARCHIVE FILE = *MARK-FOR-DELETION(...) und ist erst möglich, nachdem eine gewisse Zeitspanne (SECURE-PERIOD) abgelaufen ist, seit die Datei vom Pubset gelöscht worden ist. Die SECURE-PERIOD ist eine Eigenschaft des Versions-Backup-Archivs. Sie beträgt standardmäßig 180 Tage. Die Datei ist jetzt „zum Löschen vorgemerkt“, aber das Löschen erfolgt noch nicht. Wenn der HSMS-Verwalter vorhat, die Datei aus dem Archiv zu entfernen, obwohl sie noch auf dem Pubset liegt, muss er die Löschvormerkung mit MODIFY-ARCHIVE FILE = *FORCE-DELETION(...) erzwingen.

3. Zum Schluss wird der eigentliche Reorganisationsprozess mit der Anweisung REORGANIZE-VERSION-BACKUP gestartet. Hier findet das eigentliche Löschen der obsoleten und der zum Löschen vorgemerkten Dateien aus dem Archiv statt.

Hinweis: Falls CHECK-CATALOGED-FILES bereits durchgeführt wurde und/oder einige Dateien zum Löschen mit MODIFY-ARCHIVE FILE=*MARK-FOR-DELETION(...)/*FORCE-DELETION(...) vorgemerkt waren und kein Reorganisieren erfolgte, werden bei einem folgenden BACKUP-FILE-VERSIONS oder CHECK-CATALOGED-FILES Löschdatum und Löschvormerkung der angegebenen Dateien zurückgesetzt, wenn die Dateien inzwischen restauriert worden sind. Dazugehörige Hinweise werden im Report ausgegeben.

i Der Benutzer kann die Anzahl der aufzubewahrenden Sicherungsversionen im Dateikatalog jederzeit ändern. Diese Änderung wird jedoch nicht sofort in das Archivverzeichnis übertragen, sondern erst, wenn ein Versions-Backup unter Angabe der betreffenden Datei stattfindet, oder die Anweisung CHECK-CATALOGED-FILES aufgerufen wird. D.h. erst anschließend können HSMS Informationsfunktionen oder Reorganisationsläufe die Aktualisierungen im Dateikatalog berücksichtigen.

4.1.4 Sicherung mit der Funktion „Concurrent Copy“

Die Funktion „Concurrent Copy“ (CCOPY) bietet eine Möglichkeit der Sicherung (einschließlich Versions-Backup), bei der eine Dateimenge konsistent gesichert und gleichzeitig die Dateien in beliebiger Weise bearbeitet werden können. Dadurch ergeben sich in einem Rechenzentrum gegenüber dem herkömmlichen Sicherungsverfahren erhebliche Zeiteinsparungen.

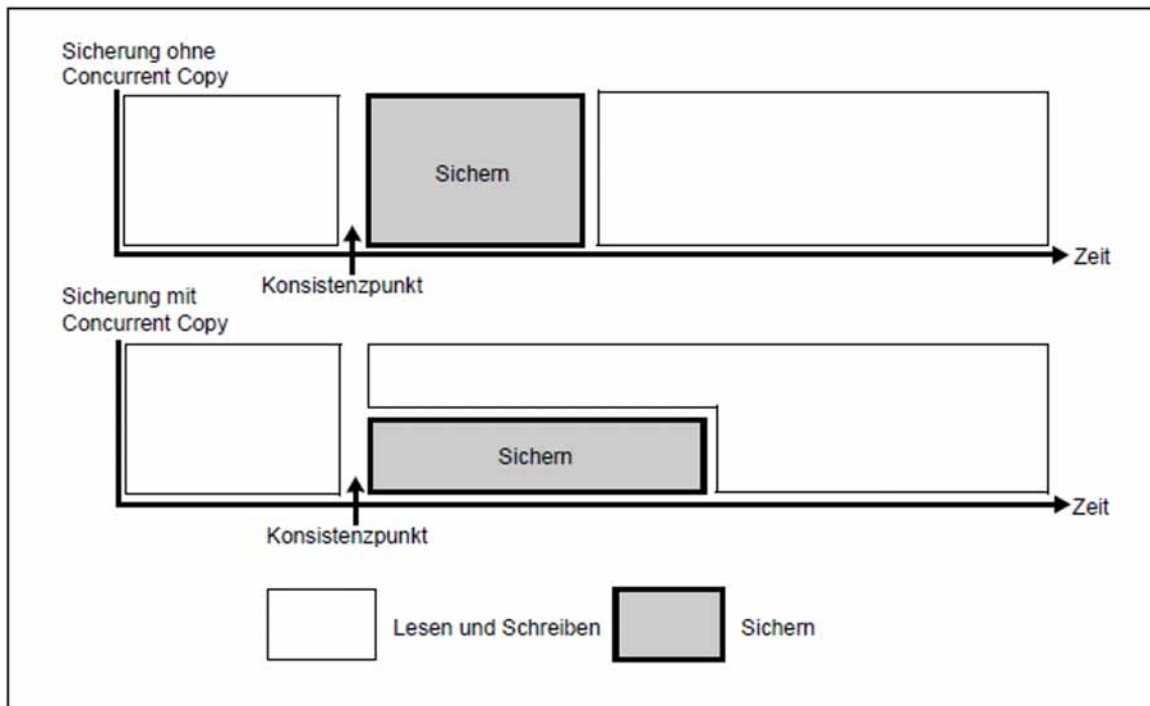


Bild 9: Verlauf einer Sicherung mit und ohne „Concurrent Copy“

Die CCOPY-Funktion ist besonders für Anwendungen gedacht, die nicht für längere Zeit geschlossen werden können, bei denen aber trotzdem Dateien gesichert werden sollen. Sie benötigt allerdings mehr Betriebsmittel als eine gewöhnliche Sicherung. Die CCOPY-Funktion wird in den Anweisungen BACKUP-FILES durch den Operanden CONCURRENT-COPY aktiviert.

Damit eine Sicherung mit der CCOPY-Funktion angefordert werden kann, muss bei Verwendung einer Arbeitsdatei (WORK-FILE-NAME=*STD bzw. <filename>) eine Dateimenge festgelegt werden, die auf diese Art gesichert werden soll. Diese Dateimenge kann z.B. alle von einer Anwendung benötigten Dateien enthalten. Beim Erstellen des Sicherungsauftrags dürfen diese Dateien höchstens lesend geöffnet sein (Ausnahmen sind die Online-Sicherung mit SHC-OSD auf "[Sicherung mit SHC-OSD](#)" sowie die Übernahme einer Snapshot-Sicherung auf "[Übernahme von Dateien und Jobvariablen aus Snapsets in ein Backup-Archiv](#)"), so dass sie auf einem konsistenten Stand sind. Der Sicherungsauftrag muss erstellt werden mit //BACKUP-FILES ..., CONCURRENT-COPY=*YES. bzw. im Falle des Versions-Backups mit //BACKUP-FILE-VERSIONS ..., CONCURRENT-COPY=*YES.

Nach dem Ende der Initialisierung erhält der HSMS-/Systemverwalter Meldungen, die das Ergebnis dieses ersten Schritts erläutern. Wenn auf seinem System der Jobvariablen-Service aktiv ist, kann der Verwalter über eine Jobvariable abfragen, ob ein Problem aufgetreten ist. Wenn die Sicherung für eine oder mehrere Dateien nicht initialisiert werden konnte, meldet HSMS dem Verwalter, welche Dateien nicht bearbeitet werden konnten. Ursache dafür kann beispielsweise sein, dass die Datei gesperrt war. Anschließend kann der Verwalter die Ursache beheben und die HSMS-Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS erneut starten.

Wenn während des Sicherungslaufs mit `CONCURRENT-COPY=*YES` eine oder mehrere Dateien nicht gesichert werden können oder ein Fehler auftritt, wird der Sicherungslauf für die übrigen Dateien nicht abgebrochen; es werden so viele Dateien wie möglich gesichert.

Dateien, die nicht gesichert werden konnten, können in eine spätere Sicherung einbezogen werden. Diese Sicherung kann mit oder ohne `CCOPY` durchgeführt werden. Die Konsistenz mit den zuerst gesicherten Dateien ist dann aber nicht gewährleistet.

Eine Sicherung mit `CONCURRENT-COPY=*YES` sichert die Dateien mit einem „gleichzeitigen“ Stand; sonst werden Dateien hintereinander gesichert.

Wenn eine der gewünschten Dateien nicht gefunden wird oder offen ist, wird die Sicherung in diesem Fall so fortgesetzt, als wenn sie ohne `CONCURRENT-COPY=*YES` gestartet worden wäre.

Bei einer Sicherung mit `CONCURRENT-COPY=*YES` ist keine Schreibpufferung mit DAB möglich.

Unabhängig von einer etwaigen Backup-Server-Definition werden Sicherungen mit Workfile stets an den Master im SPVS-Verbund geschickt und verarbeitet. Sicherungen von abgetrennten Spiegeln (`*BY-ADDITIONAL-UNIT`) werden stets lokal bearbeitet.

Sicherung mit Arbeitsdatei

Der standardmäßige Basismechanismus, mit dem für die Sicherung der Datenbestand des Konsistenzpunkts bereitgestellt wird, ist das Schreiben von sog. Before-Image-Dateien: Für die Dauer einer `CCOPY`-Sicherung wird, bevor ein Block einer in der Sicherungsmenge enthaltenen Datei zum ersten Mal geändert wird, eine Kopie des noch nicht geänderten Blocks in eine Arbeitsdatei geschrieben. Die `CCOPY`-Sicherung ermittelt aus internen Tabellen, die jeweils beim Schreiben einer Before-Image-Datei aktualisiert werden, ob ein Block aus der Originaldatei oder aus der Arbeitsdatei zu lesen ist.

Die Arbeitsdatei für die Before-Image-Dateien kann entweder standardmäßig von HSMS angelegt werden oder explizit in den HSMS-Anweisungen `BACKUP-FILES` oder `BACKUP-FILE-VERSIONS` angegeben werden:

```
//BACKUP-FILES ..., CONCURRENT-COPY=*YES(WORK-FILE-NAME= -  
    <filename 1..54 without-gen-vers>)
```

oder

```
//BACKUP-FILE-VERSIONS ..., CONCURRENT-COPY=*YES(WORK-FILE-NAME= -  
    <filename 1..54 without-gen-vers>)
```

Wenn Dateimengen auf Shared-Pubsets gesichert werden, muss bei expliziter Angabe einer Arbeitsdatei darauf geachtet werden, dass die Arbeitsdatei auf dem Master-Host oder Backup-Server des Shared-Pubset zugreifbar ist. Die Sicherung muss nämlich zwingend auf dem Master-Host oder Backup-Server durchgeführt werden (nicht aber die Erstellung des Sicherungsauftrags an HSMS).

Performance

Durch das Schreiben der Before-Image-Dateien in die Arbeitsdatei, das pro (Groß-)Block eine Lese- und eine Schreiboperation erfordert, sowie durch die interne Verwaltung der Before-Image-Dateien entsteht ein gewisser Mehraufwand an Betriebsmitteln: Ein-/Ausgaben, CPU-Zeit und Speicherplatz für die Arbeitsdatei. Der CPU-Mehrbedarf und die zusätzlichen Ein-/Ausgaben beeinflussen die Performance nachteilig. Sie sind abhängig vom Umfang und der Lokalität der Änderungen, die am Datenbestand parallel zur Sicherung durchgeführt werden. Bei geringer Last sind die Einflüsse entsprechend klein.

Sicherung mit SHC-OSD

Als weiterer Basismechanismus für CCOPY-Sicherungen stehen mit SHC-OSD die Spiegelungsfunktionen der Plattenspeichersysteme im BS2000 zur Verfügung. Dabei werden Platten eines Pubsets innerhalb eines Plattenspeichersystems auf zusätzlichen Platten (je nach Spiegelungsfunktion Additional-Units oder Clone-Units) gespiegelt. Diese zusätzlichen Spiegelplatten können bei Bedarf von den Originalplatten abgetrennt und als Pubset-Kopie unabhängig vom Original-Pubset verarbeitet werden. Das Abtrennen und das Wiederausammenführen der Plattenpaare steuert SHC-OSD. Diese Funktion lässt sich insbesondere auch zur Sicherung nutzen. Sie ist über das Subsystem CCOPY auch in HSMS integriert. Einzelheiten zur BS2000 Host-Komponente für Speichersysteme finden Sie im Handbuch „SHC-OSD“ [23].

Wenn remote in einem zweiten Plattenspeichersystem gespiegelt wird, können wahlweise auch abtrennbare Platten im Remote-Plattenspeichersystem benutzt werden.

Für HSMS-Sicherungen, die Spiegelungsfunktionen eines Plattenspeichersystems mit SHC-OSD nutzen, wird die Spiegelung auf dem Granulat-Pubset, d.h. für alle Volumes eines Pubsets, vorausgesetzt. Die Aufspaltung der Spiegelplatten-Paare wird bei der Initialisierung des Sicherungsauftrags für alle Volumes der beteiligten Pubsets vorgenommen. Während der Aufspaltung werden die Ein-/Ausgaben auf den Pubsets angehalten. Damit sind die folgenden Bedingungen erfüllt:

- Die Daten der zu sichernden Dateimenge sind in sich konsistent. Auch schreibend geöffnete Dateien führen nicht zum Abbruch der Sicherung oder können mit der Option SAVE-ONLINE-FILES=*YES wie bei Backup ohne CCOPY gesichert werden.
- Die Metadaten des Pubsets auf den abgetrennten Spiegeln (Additional-Units oder Clone-Units) sind konsistent in dem Sinn, dass sie die Rekonstruktion des Pubsets zum Zeitpunkt der Auftrennung erlauben.

Damit ist sichergestellt, dass während der gesamten Sicherung der konsistente Stand der Daten zu Sicherungsbeginn stets zur Verfügung steht. Nachdem die Sicherung erfolgreich beendet wurde, wird die Spiegelung automatisch wieder aufgenommen. Dies kann durch Angabe von DISCARD-COPY=*NO in den Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS verhindert werden. Auf die Behandlung von Fehlerfällen wird unten eingegangen.

Sicherung geöffneter Dateien

Bei der Sicherung von Pubset-Kopien unter CCOPY können auch geöffnete Dateien gesichert werden. Diese Funktion wurde speziell auf Datenbanken zugeschnitten (z.B. SESAM, UDS, ORACLE):

Wenn Datenbanken permanent zugreifbar sein sollen, können sie zum Sichern nicht heruntergefahren und geschlossen werden. Deshalb erfolgt die Sicherung in der Regel bei geöffneter Datenbank. Genauere Hinweise zum Sichern sind in den Handbüchern zu den jeweiligen Datenbanksystemen zu finden.

Mit dieser Variante der Online-Sicherung kann eine Datenbanksicherung wesentlich schneller beendet werden, da die Sperrzeiten für die Datenbankdateien kürzer werden und sich teilweise kleinere LOG-Dateien für die Rekonstruktion eines konsistenten Stands ergeben können.

Spiegelplatten umbenennen

Bei der Aufspaltung werden die Platten der Pubset-Kopie nach folgender Vorschrift umbenannt:

- Bei Volumes, deren Name einen Punkt enthält, wird der Punkt durch einen Doppelpunkt ersetzt, z.B. ABC.05 → ABC:05
- Volumes, deren Name mit dem Präfix PUB beginnt, erhalten das Präfix P:B, z.B. PUB500 → P:B500.

Das Umbenennen ist notwendig, damit nach der Aufspaltung die Volumes auf den Originalplatten und den Spiegelplatten nicht denselben Namen haben. Nur so können die Platten der Pubset-Kopie über einen eindeutigen Namen adressiert werden.

Pubset-Kopie erstellen

Beim Einleiten der Sicherung können Sie in den HSMS-Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS mit CONCURRENT-COPY=*YES(WORK-FILE-NAME=*BY-ADDITIONAL-UNIT) angeben, dass Sie eine Pubset-Kopie nutzen möchten. Dabei ist zu beachten, dass während der Durchführung eines derartigen Sicherungsauftrags die Spiegelplatten der Pubset-Kopie für weitere Sicherungen nicht zur Verfügung stehen. Daher müssen Sicherungsaufträge, die gleiche Spiegelplatten voraussetzen, koordiniert werden.

Nur eine mit //BACKUP-FILES gesicherte Pubsetkopie kann für die Rekonstruktion vollständiger SF- oder SM-Pubsets von Datenspeichern des Originalpubsets und gesicherten Dateien verwendet werden.

Wenn Sie auch geöffnete Dateien sichern wollen, müssen Sie zusätzlich SAVE-ONLINE-FILES=*YES angeben. Es werden aber nur die geöffneten Dateien gesichert, für die die Online-Sicherung mit dem Operanden OPNBACK des CATAL-Makroaufrufs ausdrücklich vereinbart wurde (siehe Handbuch „DVS-Makros“ [22]).

Außerdem ist es möglich, über die Angabe WRITE-CHECKPOINTS=*YES Sicherungsaufträge mit Spiegelplatten wiederanlauffähig und rücksetzbar zu machen (siehe unten). Dies wird empfohlen, da fast kein Zusatzaufwand damit verbunden ist. Andererseits ist damit die Verfügbarkeit des Datenbestands auf dem Stand zu Sicherungsbeginn während der gesamten Sicherungsdauer gewährleistet.

Berechtigung zur Sicherung mit SHC-OSD über Customer-Privilegien

Sicherungen auf Basis von SHC-OSD-Spiegelungsfunktionen binden in größerem Umfang Betriebsmittel, die nicht beliebig verfügbar und vermehrbar sind, sondern deren Einsatz genau geplant werden muss. Daher ist dieser Auftragstyp grundsätzlich Systembetreuern (Privileg TSOS) und HSMS-Verwaltern (Privileg HSMSADM) vorbehalten. Um ggf. auch Anwendungsverwaltern, die nicht über eines dieser Privilegien verfügen, eine Sicherung mit SHC-OSD-Spiegelungsfunktionen zu ermöglichen, kann hierfür zusätzlich ein Customer-Privileg angegeben werden. Die Steuerung über Customer-Privilegien setzt allerdings den Einsatz des Softwareprodukts SECOS (siehe Handbuch „SECOS“ [16]) voraus. Folgende Schritte sind erforderlich:

1. Es werden ein oder mehrere Customer-Privilegien festgelegt, die zu dieser Sicherung berechtigen. Diese Customer-Privilegien werden dem Subsystem CCOPY beim Start des Subsystems in einer Parameterdatei bekannt gemacht.

Standardmäßig dient dazu die Informationsdatei von CCOPY:

```
SYSSSI.CCOPY.<subsys-vers>
```

Alternativ kann im START-SUBSYSTEM-Kommando mit PARAMETER='<filename>' eine Parameterdatei mit frei wählbarem Namen angegeben werden; diese Angabe hat Vorrang.

Die Parameterdatei ist eine SAM-Datei mit variabler Satzlänge und enthält genau einen Satz. In diesem Satz werden die Customer-Privilegien für die Sicherung mit SHC-OSD-Spiegelungsfunktionen durch den String CU [STOMER]-PRIV[ILIGES]=(0,1,2,...,8) angegeben. Wenn nur ein einziges Customer-Privileg angegeben wird, können die runden Klammern entfallen. Der Wert 0 bedeutet, dass kein Customer-Privileg zu prüfen ist. Dieser Wert ist standardmäßig in der Informationsdatei von CCOPY festgelegt.

2. Den Customer-Privilegien, denen wie unter 1. beschrieben die Bedeutung „CCOPY-Sicherung mit SHC-OSD-Spiegelungsfunktionen zulässig“ zugeordnet wurde, werden über die entsprechende SECOS-Funktion eine oder mehrere Benutzerkennungen zugeordnet (siehe Handbuch „SECOS“ [16]).

Performance

CCOPY-Sicherungen mit SHC-OSD-Spiegelungsfunktionen unterscheiden sich in Bezug auf die Performance kaum von Sicherungen ohne CCOPY: Die zu sichernden Daten werden von den abgetrennten Spiegelplatten gelesen, während die parallel zur Sicherung weiter verarbeiteten Dateien auf den Originalplatten liegen. Es besteht also keine Wechselwirkung zwischen Sicherung und Anwendung.

Weil die Katalogzugriffe nicht von den laufenden Platten des Pubsets, sondern lokal von den abgetrennten Platten erfolgen, ist es bei Shared-Pubset-Betrieb nicht sinnvoll, den Auftrag für eine bessere Performance zum Pubset-Master zu verschicken. Deshalb werden CCOPY-Sicherungen mit SHC-OSD-Spiegelungsfunktionen lokal im System des Backup-Aufrufs ausgeführt.

Fehlerbehandlung

Bei CCOPY-Sicherungen mit SHC-OSD-Spiegelungsfunktionen stehen die zu sichernden Daten und alle Metadaten der beteiligten Pubsets auf den Pubset-Kopien mit dem Stand des Konsistenzpunkts zum Beginn der Sicherung zur Verfügung. Damit können im Fehlerfall weitgehende Recovery-Maßnahmen durchgeführt werden, wenn – was empfohlen wird – die Wiederanlauffähigkeit der Sicherung festgelegt wurde. Die Wiederanlauffähigkeit kann in den HSMS-Anweisungen BACKUP-FILES und BACKUP-FILE-VERSIONS oder bei der Definition des Archivs mit WRITE-CHECKPOINTS=*YES definiert werden.

Beim Arbeiten mit SHC-OSD-Spiegelungsfunktionen können durch diese Angabe auch anwendungsbezogene Recovery-Maßnahmen durchgeführt werden. Dagegen ist bei Angabe von WRITE-CHECKPOINTS=*NO das Verhalten wie bei CCOPY-Sicherungen mit Before-Image-Dateien: Es werden von HSMS/CCOPY keine Vorkehrungen zum Wiederaufsetzen getroffen.

Im Folgenden werden mögliche Fehlerfälle und die zugehörigen Möglichkeiten für den Wiederanlauf dargestellt, wenn mit WRITE-CHECKPOINTS=*YES gearbeitet wird:

1. Fehler im Sicherungslauf

Hier gilt dasselbe wie für Sicherungen ohne CCOPY: Wenn sich der HSMS-Auftrag im Zustand INTERRUPTED befindet, kann er mit der HSMS-Anweisung RESTART-REQUESTS fortgesetzt werden. Nähere Informationen dazu finden Sie im [Abschnitt „Aufträge wiederanlaufen lassen \(Restart\)“](#).

2. Systemausfall

- Sicherungslauf und restartfähige Anwendung

Nachdem das System wieder verfügbar ist, erfolgt ein Warmstart der Anwendung. Der Sicherungslauf kann mit der HSMS-Anweisung RESTART-REQUESTS fortgesetzt werden. Nähere Informationen dazu finden Sie im [Abschnitt „Aufträge wiederanlaufen lassen \(Restart\)“](#).

- Sicherungslauf und nicht restartfähige Anwendung

Nachdem das System wieder verfügbar ist, muss der betroffene Pubset auf den konsistenten Stand zu Sicherungsbeginn zurückgesetzt werden, um die abgebrochene Verarbeitung wiederholen zu können. Dazu sind folgende Schritte erforderlich:

- Die Originalplatten des Pubsets müssen mit den Spiegelplatten synchronisiert werden. Dies geschieht mit folgendem Kommando, wobei der Pubset nicht importiert sein darf:

```
/RESUME-MULTI-MIRRORING UNIT=*BY-PUBSET(PUBSET=<catid 1..4>), -  
/ RESTORE=*TO-ORIGINAL,UNLOCK-ADD-MIRROR=*YES
```

bzw.

```
/RESTART-CLONE-SESSION UNIT=*BY-PUBSET(PUBSET=<catid 1..4>), -  
/ RESTORED-SESSION=*ACCEPT
```

(siehe Handbuch „SHC-OSD“ [21]). Damit sind die Daten auf den Stand des Konsistenzpunkts zurückgesetzt.

- Nun müssen die Volumes des Pubsets, die jetzt die Archivnummern der Spiegelplatten haben, mit dem Dienstprogramm PVSREN zurückbenannt werden, sodass die Archivnummern wieder der Syntax für Pubsets genügen. Dies geschieht mit der PVSREN-Anweisung:

```
//RESTORE-LABELS-OF-PUBSET CATID=<catid 1..4>
```

(siehe Handbuch „Dienstprogramme“ [3])

- Der zurückgesetzte Pubset muss importiert werden.
- Die abgebrochene Verarbeitung muss wiederholt werden; ggf ist vorher der Sicherungsauftrag erneut zu stellen.

Einschränkungen

Für die Sicherung mit CCOPY gibt es folgende Einschränkungen:

- Die Funktionen WRITE-CHECKPOINTS und SAVE-ONLINE-FILES sind nur für Sicherungen von Pubset-Kopien verfügbar.
- Privatplatten werden nicht unterstützt.
- Eine CCOPY-Sicherung auf einem Shared-Pubset mit Arbeitsdatei (WORK-FILE-NAME=*STD bzw. <filename>) wird stets am Master bearbeitet. Eine CCOPY-Sicherung von Pubset-Kopien wird lokal auf der Seite des Backup-Aufrufs ausgeführt.
- Die Migration von Dateien wird in einer CCOPY-Sitzung nicht unterstützt.
- Der belegte Speicher einer Datei, die zur sichernden Dateimenge gehört, darf während der gesamten Sicherung nicht mit folgendem BS2000-Kommando reduziert werden:

```
/MODIFY-FILE-ATTRIBUTES . . . ,SUPPORT=*PUBLIC-DISK(SPACE=*RELEASE(. . .))
```

- Eine Datei kann zu einem bestimmten Zeitpunkt nur in einer einzigen **CCOPY-Sicherung** enthalten sein. Wenn für diese Datei ein Auftrag für eine weitere CCOPY-Sicherung erteilt wird, dann wird dieser Auftrag zurückgewiesen.
- CCOPY optimiert die Katalogzugriffe bei der Einleitung der Sicherung von einer Pubset-Kopie.

Übernahme von Dateien und Jobvariablen aus Snapsets in ein Backup-Archiv

Im BS2000 können Pubset-Kopien auf Basis von Snap-Platten (sogenannte Snapsets) erzeugt werden, von denen dann direkt per DMS-Kommando Dateien oder Jobvariablen im Original-Pubset restauriert werden können. Da die Gesamtzahl dieser Snapsets für einen Pubset durch das Plattensubsystem beschränkt ist, muss das Erzeugen und das Löschen von Snapsets im gleichen Turnus erfolgen. Vor dem Löschen eines Snapsets können die auf dem Snapset gesicherten Dateien und Jobvariablen mit der HSMS-Anweisung BACKUP-FILES in ein Backup-Archiv übernommen werden. Angegeben wird dabei die Katalogkennung des Pubsets und das Kennzeichen des zu sichernden Snapsets:

```
//BACKUP-FILES ..., CONCURRENT-COPY=*YES(WORK-FILE-NAME= -  
          *FROM-SNAPSET(PUBSET-ID=<cat id>, SNAPSET-ID=<snap-id>)
```

Gegenüber der Sicherung von Pubset-Kopien muss beim Sichern von Snapsets beachtet werden, dass die Dateien und Jobvariablen nicht dem aktuellen Stand zum Zeitpunkt der Bearbeitung von BACKUP-FILES oder BACKUP-FILE-VERSIONS entsprechen, sondern einen Sicherungsstand zum Zeitpunkt der Snapset-Erzeugung repräsentieren. Deshalb erhält die so erzeugte Sicherungsversion im Backup-Archiv oder Versions-Backup-Archiv das Datum der Snapset-Erzeugung.

Wenn im Backup-Archiv bereits neuere Sicherungsversionen vorhanden waren, muss zur Einhaltung der Monotonie der Sicherungsversionen innerhalb einer Sicherungsdatei eine neue Sicherungsdatei erstellt werden. Die Fortsetzung der Sicherungsdatei wird abgewiesen. Die Übernahme von Dateien bzw. Jobvariablen aus Snapsets sollte nicht mit direkten Sicherungen im gleichen Backup- oder Versions-Backup-Archiv kombiniert werden.

Daten von Snapsets können nicht in ein Versions-Backup-Archiv gesichert werden.

4.1.5 Sicherungsoptionen

Mit dem Operanden SAVE-OPTIONS können Optionen für den aktuellen Sicherungslauf eingestellt werden.

Online-Sicherung von Datenbanken (SAVE-ONLINE-FILES)

HSMS bietet die Möglichkeit, mit dem Operanden SAVE-ONLINE-FILES=*YES auch Dateien zu sichern, die während der Sicherung geöffnet sind. Allerdings werden nur die geöffneten Dateien gesichert, für die die Online-Sicherung mit dem Operanden OPNBACK des CATAL-Makroaufrufs ausdrücklich vereinbart wurde (siehe Handbuch „DVS-Makros“ [22]).

Diese Funktion ist nur für Datenbank-Dateien gedacht, da eine solche Sicherung bei normalen Dateien zu einem inkonsistenten Zustand der Datei führen kann.

Die Online-Sicherung von UDS-Datenbanken ist ab UDS-SQL V1.2 möglich. Voraussetzung dafür ist, dass mit einer AFTER-IMAGE-Datei gearbeitet wird. Die weiteren Bedingungen können Sie den Handbüchern für UDS-SQL entnehmen.

Sicherung des Archivverzeichnisses (SAVE-DIRECTORY)

Das Archivverzeichnis, das beim Sicherungsvorgang auf den neuesten Stand gebracht wurde, kann am Ende des Sicherungsauftrages mit in die Sicherungsdatei auf dem Sicherungsdatenträger oder Pubset geschrieben werden. Der Sicherungsdatei enthält dann außer den gesicherten Daten auch ein Verzeichnis der gesicherten Daten.

Das Archivverzeichnis wird durch die Angabe SAVE-DIRECTORY=*YES gesichert. Standardmäßig wird es nicht mitgesichert.

Die Sicherung des Archivverzeichnisses ist erlaubt für den HSMS-Verwalter, sowie für den Archiveigentümer und den Miteigentümer des Archivverzeichnisses.

Bei einer Sicherung des Archivverzeichnisses gilt Folgendes:

- Wenn bei dem Auftrag keinerlei Daten gesichert wurden, wird auch das Archivverzeichnis nicht gesichert.
- Das Archivverzeichnis wird in jedem Fall vollständig gesichert, auch bei Differenzsicherungen.
- Das Archivverzeichnis wird nicht auf verschiedene Datenträger verteilt. Die Sicherung des Archivverzeichnisses und der dafür verwendete Datenträger werden im Report angezeigt.
- Der Sicherungsdatenträger (Band oder MBK), der das Archivverzeichnis enthält, erhält bei Einsatz von MAREN ab V12.0 im MAREN-Katalog das Attribut „Volume Contains Directory“. Bei Rekonstruktion des Verzeichnisses mit IMPORT-FILES kann der entsprechende Datenträger wieder ermittelt werden.

Wenn das Archivverzeichnis gesichert wurde, kann es im Notfall (nicht mehr vorhanden oder nicht lesbar) von der Sicherungsdatei auf Band, Net-Storage, privater oder gemeinschaftlicher Platte restauriert werden. Dazu dient die Anweisung IMPORT-FILES mit der Angabe von *DIRECTORY und SAVE-FILE:

- Falls das Archivverzeichnis auf Band, Net-Storage (with SAVE-FILE-PROCESSING=*HSMS-V9-COMPATIBLE) oder eine Privatplatte gesichert worden ist, muss SAVE-FILE=*BY-VOLUME angegeben werden.
- Falls das Archivverzeichnis auf einen gemeinschaftlichen Speicher oder im Modus SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE auf ein Net-Storage gesichert worden ist, muss SAVE-FILE=*BY-PUBLIC-DISK angegeben werden.

Sicherungsdateien auf Band besitzen Several-SVID-Struktur. In diesem Fall müssen Sie eine SVID angeben, anderenfalls eine SFID. Die SFID, SVID und das Volume oder der Pubset sind im Report der Sicherung vermerkt. Bei Einsatz von MAREN ab V12.0 kann der Datenträger (Band oder MBK) auch aus dem MAREN-Katalog ermittelt werden.

Die Attribute eines Archivs werden zusätzlich im Archivverzeichnis abgespeichert. Nach einem Verlust der Archivdefinition kann das Archiv mit der Anweisung CREATE-ARCHIVE aus dem Archivverzeichnis wiederhergestellt werden (Operand RECONSTRUCT-ARCHIVE=*YES).

Beim Reorganisieren des Versions-Backup-Archivs ist es nicht möglich, das Archivverzeichnis zu sichern.

Sicherung von Informationen einer PLAM-Bibliothek (SAVE-PLAM-INFO)

Mit dem Operanden SAVE-PLAM-INFO der HSMS-Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS wird veranlasst, dass Meta-Informationen einer PLAM-Bibliothek gesichert werden. Mit der Voreinstellung *STD wird das gleichnamige Archiv-Attribut SAVE-PLAM-INFO ausgewertet. Bei der HSMS-Anweisung ARCHIVE-FILES werden die Meta-Informationen nur gesichert, wenn das Archiv-Attribut entsprechend gesetzt ist.

HSMS verwendet die gesicherten Informationen einer PLAM-Bibliothek, wenn ein oder mehrere Bibliothekselemente mit der HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS restauriert werden sollen. Mit RESTORE-LIBRARY-ELEMENTS können Sie

- eine alte Version eines Bibliothekselements in eine Bibliothek restaurieren, ohne die komplette Bibliothek restaurieren zu müssen.
- den letzten Sicherungsstand eines Bibliothekselements restaurieren, ohne die komplette Bibliothek restaurieren zu müssen.

Die HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS kann auch dazu verwendet werden, ausgewählte Elemente in eine andere Bibliothek (Zielbibliothek) zu kopieren.

Die mit RESTORE-LIBRARY-ELEMENTS adressierbaren Bibliothekselemente können im Dialog nicht selektiert werden. Mit SHOW-ARCHIVE können ebenfalls keine Elemente, sondern nur alle mit PLAM-INFO= *YES gesicherten PLAM-Bibliotheken angezeigt werden:

```
//SHOW-ARCHIVE <archiv-name>,SELECT=*FILES( . . . , -  
//          FILE-SAVE-STATE=*SAVED(TYPE=*WITH-PLAM-INFO) , . . . ) , . . .
```

Zum Restaurieren von Bibliothekselementen muss der Benutzer entweder die Elementnamen noch vom Zeitpunkt der Sicherung kennen oder er lässt sich die Elemente zuvor mit dem Operanden ELEMENTS=*LIST-ALL-TO-REPORT in der Anweisung RESTORE-LIBRARY-ELEMENTS auflisten.

Die zusätzlich gesicherten Informationen benötigen einigen Platz auf dem Sicherungsdaträger und verlängern die Sicherungsdauer. Der benötigte Platz und die Sicherungsdauer hängen von der Bibliotheksgröße und der Anzahl der Elemente ab, die in der gesicherten PLAM-Bibliothek enthalten sind.

Um beim Sichern eine optimale Performance zu erhalten, sollten Sie SAVE-PLAM-INFO

- nur verwenden, wenn Sie die HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS oft benutzen.
- bevorzugt für große Bibliotheken verwenden, die wenig Elemente enthalten, wenn Sie Probleme mit dem Speicherplatz haben.

Sicherung der Daten oder Katalogeinträge von BS2000-Net-Storage-Dateien



- Dieser Abschnitt bezieht sich nur auf die Sicherung mit BACKUP-FILES. Versions-Backup bietet diese Funktion nicht. Bei Versions-Backup werden bei Net-Storage-Dateien immer Daten und Katalog-Einträge mitgesichert.
- Dieser Abschnitt bezieht sich nur auf Net-Storage-Dateien des Typs BS2000-Datei. Net-Storage-Dateien des Typs Node-File werden immer vollständig – mit Daten und Katalog-Eintrag – gesichert.

Die Voreinstellung SAVE-NET-STOR-DATA=*YES in der HSMS-Anweisung BACKUP-FILES sichert von Net-Storage-Dateien Katalogeintrag und Daten. Mit SAVE-NET-STOR-DATA=*NO werden von Net-Storage-Dateien des Typs BS2000 nur die Katalogeinträge gesichert.

Mit //SHOW-ARCHIVE werden Dateien auf Net-Storage, von denen nur der Katalogeintrag gesichert wurde, mit dem Sicherungstyp CATL (bedeutet catalog only) ausgegeben. Die Kennzeichnung mit CATL wird auch in den Reports in der Spalte SAVE TYPE ausgegeben.

Wenn bei einer Net-Storage-Datei nur der Katalogeintrag gesichert wurde, wird die Sicherungsversion der Datei nicht hochgezählt.

Bei einer Sicherung mit SELECT-FILE=*MODIFIED-FILES (Differenzsicherung) werden Net-Storage-Dateien, die seit der letzten Sicherung nicht geändert wurden, nach folgender Regel gesichert:

Zuletzt gesichert mit	Aktuelle Differenzsicherung mit SAVE-NET-STOR-DATA=	
	*YES	*NO
nur FULL	CNS	CATL
nur CATL	FULL	CATL
FULL und CATL	CNS	CATL
weder FULL noch CATL	FULL	CATL

FULL, CATL, CNS bezeichnen den Sicherungstyp.

Bei //RESTORE-FILES kann der privilegierte Benutzer im Operanden NET-STORAGE-FILES wählen, ob bei Net-Storage-Dateien Daten und Katalogeintrag oder nur der Katalogeintrag restauriert wird:

- *DATA-AND-CATALOG restauriert Daten und Katologeintrag.
- *CATALOG-ONLY restauriert nur den Katalogeintrag.

Nicht-privilegierte Benutzer können bei Net-Storage-Dateien nur Daten und Katalogeintrag restaurieren.

Sicherungsversionen, die für eine Net-Storage-Datei den Sicherungstyp FULL enthalten, können auch verwendet werden, um den Katalogeintrag der Datei auf Net-Storage zu restaurieren.

Besonderheiten bei SAM-Node-Files

Dateien auf Net-Storage vom Typ NODE-FILE können sowohl von BS2000 als auch von offenen Systemen bearbeitet werden. Daten offener Systeme sind strukturlos abgelegt.

Auch PAM-Dateien vom Typ NODE-FILE werden vom BS2000 als strukturlose Dateien behandelt. Beim Sichern und Restaurieren braucht hier nichts weiter beachtet zu werden.

Bei SAM-Node-Files schreibt die BS2000-Zugriffsmethode die gewohnten SAM-Datenstrukturen zum Net-Storage. Auf dem Weg in das UFS werden die SAM-spezifischen Strukturinformationen vom Net-Client aus den zu übertragenden Datenblöcken entfernt und der Zeichensatz in einen ASCII-Zeichensatz umgesetzt. Somit wird eine Datei auf dem fernen Net-Storage abgelegt, die von Systemen der offenen Welt als Text-Datei aufgefasst werden kann.

Von BS2000 kann grundsätzlich auf zwei verschiedene Arten auf eine SAM-Node-File zugegriffen werden:

- so, dass die BS2000-Anwendung die SAM-Strukturen erhält, indem diese beim Lesen durch den Net-Client während der Übertragung vom Net-Server wieder hinzugefügt werden, oder
- ohne Konvertierung, d.h. ohne hinzufügen der SAM-Strukturen in die Datenblöcke und ohne Code-Konvertierung nach EBCDIC; dies ist der sogenannte RAW-Modus.

Der Zugriff im RAW-Modus auf SAM-Node-Files ist wesentlich performanter und für die schnelle Sicherung am besten geeignet. Allerdings können so gesicherte SAM-Node-Files auch nur wieder auf Net-Storage als Dateien vom Typ NODE-FILE wiederhergestellt werden.

Der RAW-Modus wird benutzt, wenn beim Sichern unter den SAVE-OPTIONS der Operand SAVE-SAM-STRUCTURE=*NO angegeben wird; dies ist wegen der besseren Performance der Standardwert.

Wenn SAVE-SAM-STRUCTURE=*YES angegeben ist, wird beim Sichern der SAM-Node-File die Konvertierung der Datei benutzt. Dadurch werden Datenblöcke im SAM-Format erzeugt und die Daten gemäß Vorgabe durch Coded-Character-Set und Net-Coded-Character-Set im Katalogeintrag der Datei in EBCDIC-Zeichen konvertiert. Das bedeutet, dass eine "echte" SAM-Datei in die Sicherungsdatei geschrieben wird. Diese Art der Sicherung ist weniger performant, hat jedoch den Vorteil, dass die gesicherten Daten sowohl auf Net-Storage (vom Typ BS2000 oder NODE-FILE) als auch auf Public-Space restauriert werden können (NEW-SUPPORT). Mit SAVE-SAM-STRUCTURE=*YES ist also die höchste Flexibilität beim Restore gegeben.



Beim Versions-Backup werden SAM-Knotendateien immer mit ihrer SAM-Struktur gesichert

4.1.6 Automatisches Duplizieren von Sicherungsdateien

Bei einer Datensicherung auf Magnetbandkassette wird die Sicherungsdatei automatisch dupliziert, wenn folgende Voraussetzungen erfüllt sind:

- Dem betroffenen Backup-Archiv wurde ein Schattenarchiv zugewiesen.
- Das automatische Duplizieren wurde nicht ausgeschaltet. D.h.: In der HSMS-Anweisung BACKUP-FILES ist beim Operanden OPERATION-CONTROL für SHADOW-COPY der Standardwert *ALLOWED wirksam.

Die Sicherungsdatei wird am Ende des Sicherungslaufs in das zugeordnete Schattenarchiv kopiert. Die Kopie im Schattenarchiv erhält dieselbe Save-Version-ID und Save-File-ID wie das Original, d.h. sie bekommt denselben Zeitstempel.

Wenn die Directory-Datei am Ende des Sicherungslaufs gesichert werden soll (SAVE-DIRECTORY=*YES), wird die Directory-Datei des zugeordneten Schattenarchivs am Ende des automatischen Duplizierens ebenfalls gesichert.

Die Directory-Datei des Schattenarchivs kann sich aber von der des zugehörigen Backup-Archivs unterscheiden, weil nicht alle Sicherungen in das Schattenarchiv kopiert werden (Sicherungen auf Platte, Sicherungen mit der Angabe SHADOW-COPY=*INHIBITED, ein neues Schattenarchiv kann mit einem nicht leeren Backup-Archiv verbunden werden).

Wenn eine Sicherungsdatei in einem Backup-Archiv erweitert werden soll, versucht der automatische Dupliziermechanismus, die Sicherungsdatei mit derselben SFID im zugeordneten Schattenarchiv ebenfalls zu erweitern. Wenn diese Sicherungsdatei nicht vorhanden ist, wird das Duplizieren (und das eventuelle Löschen der Dateien) nicht durchgeführt.

Beim impliziten Kopieren ins Schattenarchiv wird in diesem eine neue Sicherungsdatei erzeugt oder eine schon vorhandene Sicherungsdatei fortgesetzt in gleicher Weise wie bei der Sicherung im Hauptarchiv.

Im Rahmen eines Katastrophenschutzkonzeptes kann es sinnvoll sein, nach jedem Kopiervorgang sofort die Bandkopie in einen besonders geschützten Raum auszulagern. Dann können die Sicherungsdateien im Schattenarchiv aber nicht fortgesetzt werden wie im Hauptarchiv, sondern es muss mit jedem Kopiervorgang eine neue Sicherungsdatei angelegt werden. Dieser Modus wird über das Archiv-Attribut SHADOW-COPY=*ALLOWED-AND-NEW-SFID eingestellt.

Mit dem automatischen Dupliziermechanismus lassen sich die Vorsichtsmaßnahmen gegen Datenverlust an die Erfordernisse und die Speicherkapazität eines Rechenzentrums anpassen:

- Höchste Datensicherheit, sehr hohe Kosten in Bezug auf Bandverbrauch und hohe Kosten in Bezug auf Sicherungszeiten:
 - Jedem System-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle Sicherungen werden mit der Angabe SHADOW-COPY=*ALLOWED und dem Archiv-Attribut SHADOW-COPY=*ALLOWED-AND-NEW-SFID durchgeführt.

Ergebnis: Jede Sicherung auf S2 wird automatisch ins Schattenarchiv kopiert und diese Bandkopie sofort danach in ein sicheres Depot ausgelagert.

- Maximale Datensicherheit, hohe Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Jedem System-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle Sicherungen werden mit der Angabe SHADOW-COPY=*ALLOWED durchgeführt.

Ergebnis: Jede Sicherung auf S2 wird automatisch ins Schattenarchiv kopiert.

- Mittlere Datensicherheit, verbunden mit mittleren Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Jedem System-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle wöchentlichen Vollsicherungen werden mit der Angabe SHADOW-COPY= *ALLOWED durchgeführt.
 - Die dazwischenliegenden Differenzsicherungen werden mit der Angabe SHADOW-COPY=*INHIBITED durchgeführt.

Ergebnis: Nur die wöchentlichen Vollsicherungen auf S2 werden ins Schattenarchiv kopiert.

- Geringere Datensicherheit, verbunden mit geringeren Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Den System-Backup-Archiven werden keine Schattenarchive zugeordnet.

Ergebnis: Der automatische Dupliziermechanismus wird nicht genutzt.

4.1.7 Sicherung auf Platte oder auf Net-Storage

In Rechenzentren nimmt die Menge der zu sichernden Daten ständig zu. Da die Anzahl der verfügbaren Geräte und die Größe des Sicherungsfensters nicht ohne weiteres vergrößert werden können, bietet die Sicherung auf Platte einen Ausweg. Während Bandgeräte oft nur zu eingeschränkten Zeiten genutzt werden können, ist die Verfügbarkeit von Platten immer gewährleistet. Deshalb werden die Daten zuerst auf Platte oder auf Net-Storage gesichert. Später können sie dann auf Band verlagert werden. Mögliche Kriterien für den Verlagerungszeitpunkt sind z.B. die Verfügbarkeit von MBK-Geräten oder auch das Erreichen einer bestimmten Plattenspeicherbelegung.

Abhängig vom Archiv-Typ werden Sicherungsdateien mit folgenden Anweisungen verschoben:

- Backup-Archive: MOVE-SAVE-FILE
- Versions-Backup-Archiv: REORGANIZE-VERSION-BACKUP

4.1.7.1 Sicherungsdateien und Knotendateien mit der Anweisung MOVE-SAVE-FILES verlagern

Mit der Anweisung MOVE-SAVE-FILES können die von Sicherungsdateien sowie Knoten-Sicherungsdateien komfortabel von Platte oder Net-Storage auf Band verlagert werden. Außerdem können Sicherungsdateien ab HSMS V12.0A von Band auf eine gemeinschaftliche Platte oder Net-Storage verlagert werden. Die Anweisung fasst die Schritte Kopieren und Löschen für mehrere Sicherungsdateien bzw. Knoten-Sicherungsdateien auf dem Original-Speicher zusammen. Sie dient damit auch der Reorganisation einer Platte oder eines Bandes (nur ab HSMS V12.0A).

In einem Verlagerungsauftrag werden mehrere Sicherungsdateien von einem Speicher auf einen anderen übertragen.

- Sicherungsdateien können von Platte, Net-Storage oder Band auf eine gemeinschaftliche Platte, Net-Storage oder auf ein anderes Band übertragen werden.
- Knotensicherungsdateien können nur von Platte auf Band übertragen werden.

Gleichzeitig löscht MOVE-SAVE-FILES die nicht mehr benötigten Sicherungsdateien von dem Original-Speicher. Weil bei Sicherungen auf Platte oder Net-Storage keine Schattenarchive unterstützt werden, werden die Sicherungsdateien bei der Verlagerung (ebenso wie bei einer direkten Sicherung auf Band) implizit auf Band kopiert. Optional kann das Kopieren ins Schattenarchiv unterdrückt werden.

Nach der Verlagerung einer Sicherungs- bzw. Knoten-Sicherungsdatei ist derselbe Zustand erreicht wie bei einer Originalsicherung direkt auf einen neuen Speicher.

Eine Sicherungs- bzw. Knoten-Sicherungsdatei wird immer gesamt kopiert (eine Reduzierung wie beim expliziten Kopieren ist nicht möglich). Sie enthält deshalb auch alle Cataloged-Not-Saved-Einträge (CNS) der Original-Sicherungsdatei. Das Restaurieren aus dem Backup-Archiv liefert nach der Verlagerung dasselbe Ergebnis wie vor der Verlagerung.

Hinweis

Ein direkter Backup auf Band (für dieses Archiv) darf erst wieder nach Abschluss des Verlagerungsauftrags durchgeführt werden. Es entstehen sonst Probleme beim Verlagern, weil das Fortschreiben auf Band nicht mehr der zeitlichen Reihenfolge der Sicherungsversionen entspräche.

Dateien für die Verlagerung auswählen

Für die Auswahl der zu verlagernden Sicherungsdateien stehen folgende Auswahlkriterien zur Verfügung:

- Wenn Sicherungsdateien auf Plattenspeichern (Privatplatten, Pubsets und Net-Storage) oder Sicherungsdateien auf Band zu verlagern sind:
`FROM-STORAGE = *DISK(...)`
`FROM-STORAGE = *S2-STORAGE-LEVEL`
- Minimale Verweildauer der Sicherungs- bzw. Knotensicherungsdateien auf Platte: Es werden alle Sicherungs- bzw. Knotensicherungsdateien verlagert, die mindestens die angegebene Anzahl von Tagen oder schon länger auf Platte liegen.
`FROM-STORAGE=*DISK(MINIMUM-DAYS-ON-S1=tage, ...)`
- Größe des Speicherplatzes, der durch die Verlagerung frei werden soll: Es werden nur so viele Sicherungs- bzw. Knotensicherungsdateien von Platte auf einen anderen Speicher verlagert, bis die angegebene Zahl von 2-Kbyte-Blöcken durch die Verlagerung frei geworden ist.
`FROM-STORAGE=*DISK(... ,RELEASE-PAGES=pam-seiten)`

Verlagerte Sicherungsdateien neu anlegen oder fortschreiben

- Bei Backup-Archiven mit Single-SVID-Struktur werden bei der Verlagerung die Sicherungsdateien neu angelegt (entspricht der Voreinstellung). Pro verlagerter Sicherungsdatei wird eine neue Sicherungsdatei auf Band erzeugt.
- Mehrere Sicherungsdateien auf Platte oder Net-Storage werden zu einer Sicherungsdatei zusammengefasst. Bei Backup-Archiven mit Several-SVID-Struktur kann auch die jeweils zuletzt auf S2 erstellte Sicherungsdatei fortgeschrieben werden. Diese Möglichkeit besteht nicht, wenn nach der Sicherung auf Platte noch eine Sicherung auf Band erfolgt ist.
- Nach der Verlagerung haben die neuen Sicherungsversionen das Datum der entsprechenden alten, jeweils um 1 erhöht. Die Reihenfolge aller Sicherungsversionen bleibt dadurch gewahrt. Die Sicherungsdateien werden mit dem Datum des MOVE-SAVE-FILES-Zeitpunkts im Archivverzeichnis abgelegt.

Bei der Verlagerung **von Band auf Platte** wird jede Sicherungsversion in eine einzelne Sicherungsdatei auf Platte verlagert (die Save-File-ID der neuen Sicherungsdatei (SFID) ist die vorige Sicherungsversion (SVID) mit dem Zeitstempel plus eine Sekunde).

Bei der Verlagerung von der **Speicherebene S2 auf die Speicherebene S2** hängt das Ergebnis von dem Archivtyp ab:

- Für Archive mit SEVERAL-SVID-Struktur wird eine neue Sicherungsdatei angelegt. Alle Sicherungsversionen der Speicherebene S2 werden in diese verschoben.
- Für Archive mit SINGLE-SVID-Struktur wird jede Sicherungsversion in eine separate Sicherungsdatei geschrieben. Die Kopie der Sicherungsversion hat eine SVID, die gleich der Original-SVID plus eine Sekunde ist. Jede neue Sicherungsdatei hat eine SVID mit dem aktuellen Zeitstempel.

4.1.7.2 Sicherungsdateien beim Reorganisieren des Versions-Backups verlagern

Die Verlagerung der Dateien mit MOVE-SAVE-FILE ist bei Versions-Backup nicht möglich. Die Reorganisation des Versions-Backup-Archivs bietet jedoch die Möglichkeit, den Speicherort der zu reorganisierenden Sicherungsdateien zu ändern. Mit Hilfe der Anweisung REORGANIZE-VERSION-BACKUP ist es möglich, den Speicherort anzugeben, in den die Sicherungsdateien kopiert werden sollen.

```
TO-STORAGE = *S2-STORAGE-LEVEL(...)
```

Die zu reorganisierenden Dateien können über ihr Erzeugungsdatum ausgewählt werden:

```
SAVE-FILE-ID = *BY-ATTRIBUTES(CREATED-BEFORE=
```

Die beim Sicherungslauf angelegte Sicherungsversion erhält als SVID den aktuellen Zeitstempel. Die Funktionalität der Anweisung REORGANIZE-VERSION-BACKUP vereint in sich:

- Kopieren mehrerer Sicherungsdateien, wobei nur gültige Dateiversionen für den angegebenen Speicher ausgewählt werden (COPY-SAVE-FILE)
- Nachfolgende Bereinigung der ursprünglichen Sicherungsdateien aus dem Archiv.

Im Falle einer Reorganisation von Platte auf S2 wird jede Sicherungsdatei auf Platte als Sicherungsversion in einer neuen Sicherungsdatei mit dem aktuellen Datum als SVID auf Band dargestellt. Bei der Reorganisation auf Platte werden alle Sicherungsversionen und -Dateien in separate Dateien auf Platte geschrieben. Dabei werden neue Sicherungsdateien angelegt; bei der Reorganisation werden Sicherungsdateien nicht fortgesetzt.

Die Schutzfrist der durch die Reorganisation erzeugten Sicherungsdateien wird aus den Attributen des Versions-Backup-Archivs übernommen.

4.1.8 Gesicherte BS2000-Dateien und Jobvariablen restaurieren

Unter Restaurieren versteht man das Kopieren gesicherter Daten aus dem Backup-Archiv auf die Verarbeitungsebene. Hierzu steht die HSMS-Anweisung RESTORE-FILES zur Verfügung.

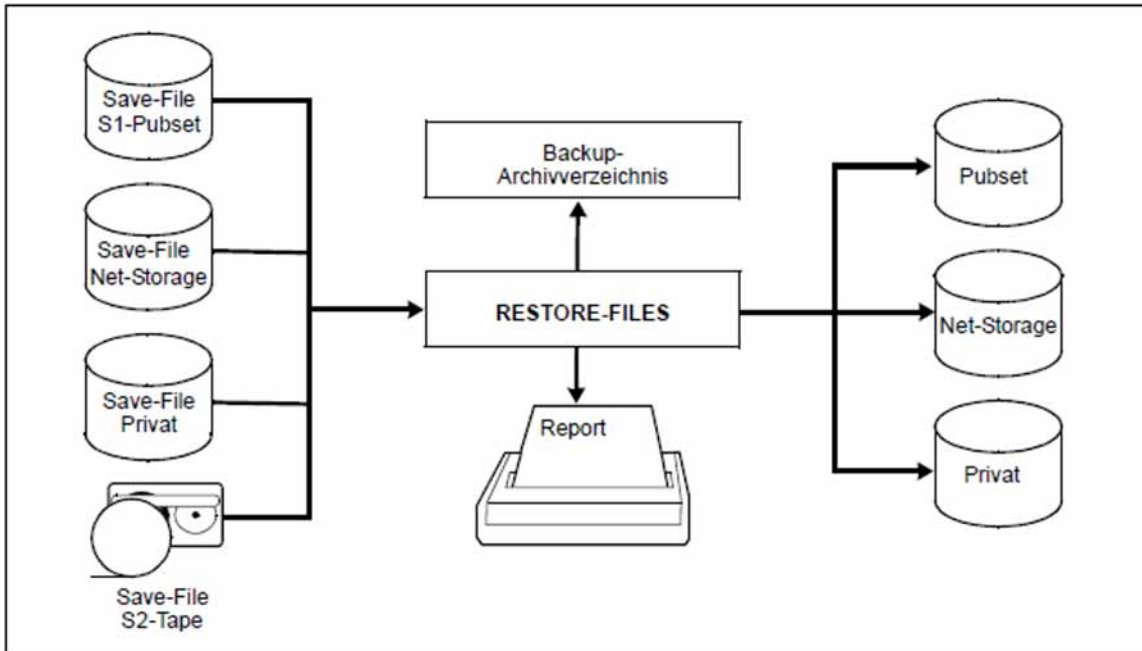


Bild 10 a: Restaurieren aus dem Backup-Archiv mit RESTORE-FILES

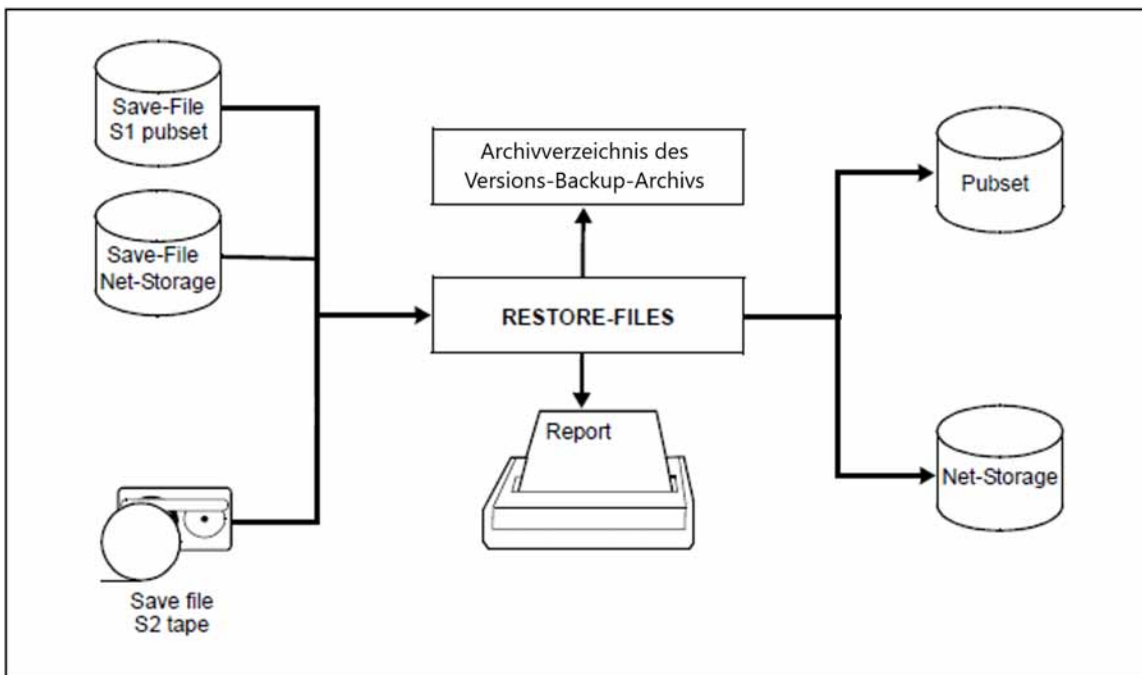


Bild 10 b: Restaurieren aus dem Versions-Backup-Archiv mit RESTORE-FILES

Eine RESTORE-FILES-Anweisung, die sich auf BACKUP-Anweisungen bezieht, ist auf die Umgebung beschränkt, in der sie gegeben wurde. Wenn sich eine RESTORE-FILES-Anweisung auf SF-Pubsets bezieht, kann sie keine SM-Pubsets in die Bearbeitung einschließen; wenn sie sich auf einen SM-Pubset bezieht, kann sie zu einem Zeitpunkt nur einen SM-Pubset bearbeiten.

Ein nicht-privilegierter Benutzer kann Dateien und Jobvariablen restaurieren, die ihm entweder selbst gehören oder bei denen er Miteigentümer ist (siehe Handbuch „SECOS“ [16]). Diese Objekte werden restauriert:

- entweder aus dem Standard-Systemarchiv (gilt sowohl für Backup- also auch Versions-Backup-Archive) oder
- aus Backup-Archiven, die in der Umgebung des Benutzers definiert sind (SF-Pubsets oder SM-Pubsets) oder
- aus Backup-Archiven, die in der Umgebung von anderen Benutzern definiert sind, vorausgesetzt, dass auf diese Archive zugegriffen werden darf.

Beim Restaurieren aus einem Backup-Archiv werden die Dateien mit den Katalogeigenschaften angelegt, die sie bei der Sicherung mit FULL oder MIGF hatten (falls nicht die Katalog- oder Benutzerkennung geändert werden).

Beim Restaurieren von migrierten (d.h. als „migriert“ gesicherten) Dateien kann der HSMS-Verwalter wählen, ob nur die Katalogeinträge oder auch die Daten der migrierten Dateien auf die Speicherebene S0 gebracht werden. Darüber hinaus kann er auch die Daten von migrierten Dateien auf eine Migrationsebene restaurieren.

Restaurieren aus einem Versions-Backup-Archiv ist nur ab BS2000 OSD V11.0B oder höher möglich.

4.1.8.1 Restaurieren von großen Dateien (> 32 GB)

Große Dateien (≥ 32 GB) können nur auf Pubsets mit den Attributen `LARGE-OBJECTS=*YES` und `LARGE-FILES-ALLOWED=*YES` restauriert werden.

Wenn eine Datei ≥ 32 GB auf einen Pubset mit dem Attribut `LARGE-FILES-ALLOWED=*NO` eingespielt werden soll, wird dies immer mit Fehler quittiert. Dies gilt auch, wenn die Datei z.B. über eine Mengenselektion wie Teilqualifizierung oder Wildcards implizit ausgewählt wurde.

Sicherungsbestände aus älteren BS2000-Versionen können auch in höheren BS2000-Versionen uneingeschränkt restauriert werden, also auf Standard-Pubsets und auf Pubsets mit dem Attribut `LARGE-OBJECTS=*YES`.

Bei einer partiellen Sicherung (Backup) ist Folgendes zu beachten:

Wenn die zu restaurierende partielle Sicherungsversion oder die zugehörige Vollsicherung eine Datei ≥ 32 GB ist, kann nur auf Pubsets mit den Attributen `LARGE-OBJECTS=*YES` und `LARGE-FILES-ALLOWED=*YES` restauriert werden.

4.1.8.2 Sicherungsversionen auswählen

Die zu restaurierenden Daten können typischerweise aus jeder im Archiv verwalteten Sicherungsversion entnommen werden.

Die Auswahlkriterien der Sicherungsversionen, nach denen die Daten restauriert werden sollen, unterscheiden sich wegen des unterschiedlichen Backup-Konzeptes zwischen dem Restaurieren aus einem Backup-Archiv und einem Versions-Backup-Archiv.

Sicherungsversionen aus dem Backup-Archiv auswählen

Die Dateien und Jobvariablen, die restauriert werden sollen, werden standardmäßig aus allen im Archiv verwalteten Sicherungsversionen geholt. Dies ist sinnvoll, weil in der Regel bestimmte Dateien oder Jobvariablen restauriert werden sollen, die mit FILE-NAMES, PATH-NAMES oder JV-NAMES angegeben sind. Diese werden dann in allen Sicherungsversionen gesucht. Standardmäßig werden sie aber der jeweils aktuellsten Sicherungsversion entnommen, in der sie enthalten sind.

Durch Angabe von `SELECT-SAVE-VERSIONS=*LATEST(...)` kann aber auch nur die zuletzt erzeugte Sicherungsversion zum Restore herangezogen werden. In diesem Fall können die Dateien und Jobvariablen, die als Cataloged-Not-Saved (CNS) vermerkt wurden, folgendermaßen restauriert werden:

- entweder von der zuletzt erzeugten Sicherungsversion durch die Angabe `SELECT-SAVE-VERSIONS=*LATEST(CREATED-AFTER=*EARLIEST-DATE)` oder
- von der zuletzt erzeugten Sicherungsversion, die nach einem angegebenen Datum erstellt wurde, durch die Angabe `SELECT-SAVE-VERSIONS=*LATEST(CREATED-AFTER=<date>)`.

i Bei der Auswahl von Sicherungsversionen für den Restore von Net-Storage-Dateien ist Folgendes zu beachten: Wenn eine Sicherungsversion ausgewählt wird, die als Cataloged-Only (CATL) vermerkt ist und jünger ist als die letzte Sicherungsversion mit einer vollständigen Sicherung (FULL), werden keine Net-Storage-Dateien restauriert. In diesem Fall sollte ein Intervall von Sicherungsversionen ausgewählt werden, das die CATL-Version nicht enthält, und der Restore wiederholt werden.

Katalogeinträge von BS2000-Net-Storage-Dateien restaurieren

i Dieser Abschnitt bezieht sich nur auf Net-Storage-Dateien des Typs BS2000-Datei. Es ist nicht möglich von Net-Storage-Dateien des Typs Node-File nur die Katalogeinträge zu restaurieren.

Die Möglichkeiten zur Auswahl von Sicherungsversionen gelten auch für den Restore des Katalogeintrags einer BS2000-Net-Storage-Datei. Die jüngste Sicherungsversion, die nur den Katalogeintrag enthält (Sicherungstyp CATL) wird zuerst gewählt. Wenn eine Sicherungsversion mit CATL vorhanden ist oder wenn die Sicherungsversion mit CATL nicht die jüngste Version der Datei enthält, kann der Katalogeintrag der Net-Storage-Datei auch aus der letzten Sicherungsversion restauriert werden, die auch Daten enthält (Sicherungstyp FULL oder PART).

Ob eine Sicherungsversion, die nur Katalogeintrag oder Katalogeintrag und Daten enthält, für den Restore des Katalogeintrags einer Net-Storage-Datei verwendet wird, entscheidet sich nach folgender Regel:

- Die Auswahl der Sicherungsversionen enthält sowohl Sicherungstypen CATL als auch FULL.
 - Wenn die Version und CFID der Datei gleich sind, wird der Katalogeintrag von der Sicherungsversion mit CATL restauriert.
 - Wenn die Version und CFID der Datei verschieden sind und die jüngste Version ist vom Typ FULL, wird der Katalogeintrag von dieser Sicherungsversion mit FULL restauriert.
 - Wenn die Auswahl Sicherungsversionen mit CNS enthält, wird der Katalogeintrag von der letzten Sicherungsversion, die die jüngste Version der Datei enthält (entweder CATL oder FULL), restauriert.
 - Wenn die Auswahl keine Sicherungsversion mit CATL aber mit FULL enthält, wird der Katalogeintrag von dieser Sicherungsversion mit FULL restauriert.
- Wenn nur Sicherungsversionen CNS vorhanden sind (kein CATL oder FULL), werden keine Katalogeinträge restauriert und die Meldung ARC0505 wird ausgegeben.

In einigen Fällen können mehrere Sicherungsversionen am selben Tag erstellt werden. Der Begriff „letzte Sicherungsversion“ wird deshalb auf eine Reihe von Sicherungsversionen ausgeweitet, die alle am selben Tag erstellt wurden (Angabe `SELECT-SAVE-VERSIONS=*LATEST(DAY-INTERVAL=*YES)`). Dieser Operandenwert ist aber nur für gesicherte Dateien verfügbar.

Darüber hinaus können Sicherungsversionen mit einem bestimmten Namen oder einem bestimmten Sicherungszeitpunkt ausgewählt werden:

```
SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES( . . . )
```

Auch die Angabe eines Zeitintervalls zur Auswahl der Sicherungsversion ist möglich.

Beispiel für das Restaurieren von Sicherungsversionen

- Die Dateien wurden bei verschiedenen Sicherungsläufen folgendermaßen gesichert:

Save-Version-Name	BACKUP01	BACKUP02	BACKUP03	BACKUP04
FILE.1	FULL ¹	FULL	FULL	CNS
FILE.2	FULL	CNS	FULL	
FILE.3	FULL			FULL
FILE.4	FULL			

¹= Sicherungstyp

- Das Ergebnis unterschiedlicher Restore-Aufträge sieht dann so aus:

//RESTORE-FILES FILE-NAMES=FILE.,- SELECT-SAVE-VERSIONS=	restaurierte Dateien	aus Save-Version
*LATEST	FILE.1	BACKUP03
	FILE.3	BACKUP04
*ALL	FILE.1	BACKUP03
	FILE.2	BACKUP03
	FILE.3	BACKUP04
	FILE.4	BACKUP01
*BY-ATTRIBUTES(SAVE-VERSION-NAME=BACKUP02)	FILE.1	BACKUP02
	FILE.2	BACKUP01

Sicherungsversionen aus dem Versions-Backup-Archiv auswählen

Das Auswahlkriterium SELECT-SAVE-VERSION für ein Versions-Backup-Archiv hat seine eigene Bedeutung im Vergleich zu anderen Archivtypen.

Beim Restaurieren aus einem Versions-Backup-Archiv ist die Auswahl der Dateiversion nur über die Original-SVID möglich, d.h. über die Option *BY-ORIGINAL-DATE. Alle anderen Optionen werden abgelehnt. In Versions-Backup-Archiven ist dieses Auswahlkriterium "dateiversions-orientiert" (nicht "sicherungsversions-orientiert").

Beim Restaurieren der jüngsten Dateiversion muss der Operand SELECT-SAVE-VERSIONS den Wert *STD haben, d.h. SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES(SAVE-VERSION-NAME=*ANY, SAVE-VERSION-DATE=*BY-ORIGINAL-DATE(CREATED-BEFORE=*LATEST-DATE, CREATED-AFTER=*EARLIEST-DATE)).

Die Reihenfolge der gesicherten Versionen wird im Versions-Backup-Archiv nicht garantiert. Nach dem Reorganisieren kann unter Umständen die jüngste Version einer Datei in einer älteren Sicherungsversion (SVID) enthalten sein.

Wird der Restore der neuesten Dateiversionen benötigt, wird folgendes angegeben: SAVE-VERSION-DATE = *BY-ORIGINAL-DATE (CREATED-BEFORE = *LATEST-DATE, CREATED-AFTER = *SAME-AS-BEFORE). Über dieses Auswahlkriterium findet HSMS die jüngste Original-SVID innerhalb des ganzen Versions-Backup-Archivs und wählt nur die Dateien aus, die ursprünglich in dieser Sicherungsversion gesichert waren.

Alle möglichen Angaben zur Auswahl von Sicherungsversionen in Versions-Backup-Archiven mit Beispielen siehe im Handbuch "HSMS Band 2 [1]"

4.1.8.3 Plattenspeicher reorganisieren

Der gemeinschaftliche Plattenspeicher (Pubsets) kann durch eine Sicherung mit anschließendem Restore reorganisiert werden. Eine Reorganisation wird nötig, wenn die Dateien auf den Platten auf immer mehr Bereiche (Extents) verteilt sind und eventuell durch Erreichen der Höchstzahl zulässiger Extents nicht mehr erweitert werden können. Mit HSMS 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.

Beispiel für das Restaurieren von BS2000-Net-Storage-Dateien

- Die Dateien wurden bei verschiedenen Sicherungsläufen folgendermaßen gesichert:

Save-Version-Nameee	BACKUP01	BACKUP02	BACKUP03	BACKUP04
NET.FILE.1	FULL (1)	CNS (1)	CATL (1)	CNS (1)
NET.FILE.2	CATL (1)	CATL (2)	CNS (2)	-
NET.FILE.3	CATL (1)	FULL (1)	CATL (2)	CNS (2)
NET.FILE.4	FULL (1)	-	CNS (1)	-

FULL, CATL, CNS sind Sicherungstypen. Die verschiedenen Ziffern in Klammern zeigen verschiedene Versionen (oder CFIDs) der Dateien.

- Das Ergebnis unterschiedlicher Restore-Aufträge sieht dann so aus:

//RESTORE-FILES FILE-NAMES=NET.FILE.,- NET-STORAGE-FILES=*DATA-AND-CATALOG,- SELECT-SAVE-VERSIONS=	restaurierte Dateien	aus Save-Version
*ALL	NET.FILE.1	BACKUP01
	NET.FILE.2	ARC0081
	NET.FILE.3	ARC0507
	NET.FILE.4	BACKUP01
*LATEST	NET.FILE.1	BACKUP01
	NET.FILE.2	-
	NET.FILE.3	ARC0507
	NET.FILE.4	-
BACKUP03	NET.FILE.1	BACKUP01
	NET.FILE.2	ARC0081
	NET.FILE.3	ARC0507
	NET.FILE.4	BACKUP01
*INTERVAL(AFTER=BACKUP02,BEFORE=BACKUP03)	NET.FILE.1	BACKUP01
	NET.FILE.2	ARC0081
	NET.FILE.3	ARC0507
	NET.FILE.4	BACKUP01

- *Beispiel für das Restaurieren der Katalogeinträge von BS2000-Net-Storage-Dateien*
- Die Dateien wurden bei verschiedenen Sicherungsläufen folgendermaßen gesichert:

Save-Version-Name	BACKUP01	BACKUP02	BACKUP03	BACKUP04
NET.FILE.1	FULL (1)	CNS (1)	CATL (1)	CNS (1)
NET.FILE.2	CATL (1)	CATL (2)	CNS (2)	-
NET.FILE.3	CATL (1)	FULL (1)	CATL (2)	CNS (2)
NET.FILE.4	FULL (1)	-	CNS (1)	-

FULL, CATL, CNS sind Sicherungstypen. Die verschiedenen Ziffern in Klammern zeigen verschiedene Versionen (oder CFIDs) der Dateien.

- Das Ergebnis unterschiedlicher Restore-Aufträge sieht dann so aus:

//RESTORE-FILES FILE-NAMES=NET.FILE.,- NET-STORAGE-FILES=*CATALOG-ONLY,- SELECT-SAVE-VERSIONS=	restauriert e Dateien	aus Save-Version
*ALL	NET.FILE.1	BACKUP03
	NET.FILE.2	BACKUP02
	NET.FILE.3	BACKUP03
	NET.FILE.4	BACKUP01
*LATEST	NET.FILE.1	BACKUP03
	NET.FILE.2	-
	NET.FILE.3	BACKUP03
	NET.FILE.4	-
BACKUP02	NET.FILE.1	BACKUP01
	NET.FILE.2	BACKUP02
	NET.FILE.3	BACKUP01
	NET.FILE.4	-
*INTERVAL(AFTER=BACKUP01,BEFORE=BACKUP02)	NET.FILE.1	BACKUP01
	NET.FILE.2	BACKUP02
	NET.FILE.3	BACKUP02
	NET.FILE.4	BACKUP01

4.1.8.4 Dateien und Jobvariablen umbenennen

Beim Restaurieren von Dateien können die Dateinamen mit dem Operanden NEW-FILE-NAMES umbenannt werden. Dabei müssen aber die Konventionen für Dateinamen im BS2000 eingehalten werden. Das Umbenennen geschieht nicht einzeln pro Datei, sondern für alle Dateien, die in einem Auftrag restauriert werden.

Für das Umbenennen gibt es folgende Möglichkeiten:

- Die Dateien können mit einem Präfix oder Suffix versehen werden.
- Die Dateien können auf eine andere Katalog- und/oder Benutzerkennung gespielt werden, wenn der jeweilige Benutzer für diese Katalog- bzw. Benutzerkennung Zugriffserlaubnis hat.
- Der HSMS-Verwalter hat zusätzlich das Recht, die Dateien auf eine andere Benutzerkennung umzubenenen.

i Beim Restaurieren der Katalogeinträge von BS2000-Net-Storage-Dateien (//RESTORE-FILES mit dem Operanden NET-STORAGE-FILES=*CATALOG-ONLY) können die Dateien nur in einen anderen Katalog verschoben werden. Das Verschieben zu einer anderen Benutzerkennung und das Umbenennen von Dateien ist nicht erlaubt.

Entsprechend können beim Restaurieren von Jobvariablen aus dem Sicherungsarchiv die Jobvariablenamen mit dem Operanden NEW-JV-NAMES umbenannt werden.

4.1.8.5 Aus einem Schattenarchiv restaurieren

Wenn einem Backup-Archiv ein Schattenarchiv zugewiesen wurde, kann das Schattenarchiv für das Restaurieren der Dateien benutzt werden. Dies ist hilfreich, wenn Dateien im zugehörigen Archiv zerstört sind oder der entsprechende Datenträger nicht zugreifbar ist.

4.1.8.6 Elemente aus einer PLAM-Bibliothek restaurieren

Mit der HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS können Elemente einer PLAM-Bibliothek aus den Backup- und Versions-Backup-Archiven restauriert werden. Die vorangegangenen Sicherungen oder Versions-Backups der PLAM-Bibliothek müssen die Meta-Informationen über die Bibliothekselemente enthalten (SAVE-PLAM-INFO=*YES in den Anweisungen BACKUP-FILES oder BACKUP-FILE-VERSIONS bzw. bei den Sicherungsoptionen des Backup- oder Langzeit-Archivs).

Wenn eine PLAM-Bibliothek mit den Meta-Informationen über die Bibliothekselemente gesichert werden soll, muss sich diese Bibliothek auf der S0-Ebene befinden. Bei einer Migration von PLAM-Bibliotheken werden diese Meta-Informationen nicht mitgesichert.

Die Bibliothek muss auf S0 zugreifbar sein, darf also nicht verdrängt sein. Die Elemente können wahlweise überschrieben und umbenannt werden.

Die Sicherungsversion der Bibliothek, in der die angegebenen Elemente vorhanden sind, muss bekannt sein.

Die Bibliothekselemente können auch in eine andere Zielbibliothek als die Ausgangsbibliothek restauriert werden.

Die Elementnamen einer mit PLAM-INFO=*YES gesicherten PLAM-Bibliothek sind nur in der PLAM-Information auf der Sicherungsdatei aber nicht im Archivverzeichnis enthalten. Die Elementnamen einer so gesicherten PLAM-Bibliothek können mit dem Operanden ELEMENTS=*LIST-ALL-TO-REPORT in der Anweisung RESTORE-LIBRARY-ELEMENTS aufgelistet werden.

4.1.8.7 Net-Storage-Dateien restaurieren

Das Restaurieren kann auf Dateien auf Net-Storage beschränkt werden (Operand STORAGE-TYPE).

Net-Storage-Dateien vom Dateityp BS2000 oder Node-File, deren SAM-Struktur mitgesichert wurde, können auch auf andere Datenträgertypen restauriert werden. Dabei kann auch ihr Dateityp geändert werden. Im Gegensatz dazu können Net-Storage-Dateien vom Dateityp Node-File, deren SAM-Struktur nicht mitgesichert wurde, nur auf ein Net-Storage-Volume und nur mit dem ursprünglichen Dateityp Node-File restauriert werden.

4.1.9 Knotendateien des lokalen BS2000-UFS und Knoten-S0 sichern

HSMS bietet für Knotendateien des lokalen BS2000-UFS sowohl Voll- als auch Differenzsicherung an; Teilsicherung wird nicht unterstützt.

Alle Benutzer dürfen Sicherungen durchführen. Systemsicherungen sind aber dem HSMS-Verwalter vorbehalten.

Ein nicht-privilegierter Benutzer darf seine eigenen Knotendateien sichern und restaurieren, die auf dem lokalen BS2000-UFS liegen.

Der HSMS-Verwalter darf die Knotendateien aller Benutzer bearbeiten, die auf dem lokalen BS2000-UFS oder auf einem fernen Dateisystem liegen.

Knotendateien werden mit der HSMS-Anweisung `BACKUP-NODE-FILES` gesichert. Sie können durch die Angabe ihres Namens (mit Wildcard) und durch den Operanden `SELECTION-BOUNDARY` ausgewählt werden. Es werden entweder nur die angegebenen Dateien gesichert oder die angegebenen Dateien und – bei einem Dateiverzeichnis – alle untergeordneten Dateien und Dateiverzeichnisse bis zur Dateisystemgrenze oder bis zum Ende des Dateibaums (siehe [Abschnitt „Auswahl von Knotendateien des BS2000-UFS“](#)).

Beim Sichern von Knotendateien kann die Directory-Datei nicht im selben Sicherungslauf mitgesichert werden, da die Directory-Datei eine BS2000-Datei ist und deshalb in einem BS2000-Sicherungslauf gesichert werden muss.

4.1.9.1 Umfang der Sicherung

HSMS sichert, abhängig vom Dateityp der zu sichernden Knotendatei, entweder nur den Indexeintrag oder zusätzlich auch die Daten:

Knotendatei, die gesichert werden soll	UNIX
Normale Datei	ja
Dateiverzeichnis	ja
Symbolic Link	ja
Hard Link	ja
Semaphor	nein
FIFO-Datei	nein
Zeichenorientierte Gerätedatei	nein
Blockorientierte Gerätedatei	nein
Gemeinsam benutzbare Daten	nein
Dateisystem /proc	nein

Bei einem Symbolic Link bearbeitet HSMS nur den Indexeintrag und den Pfad der Knotendatei, auf die verwiesen wird.

Einen Hard Link bearbeitet HSMS folgendermaßen: Beim ersten Vorkommen eines Hard Links auf eine Datei werden der entsprechende Indexeintrag und die Daten wie eine normale Knotendatei bearbeitet. Wenn weitere Hard Links auf dieselbe Knotendatei verweisen, schreibt HSMS den Pfadnamen und einen Pointer auf die erste Verweisstelle in die Sicherungsdatei.

Nähere Informationen zu Symbolic Links und Hard Links finden Sie im Handbuch „POSIX Kommandos“ [\[13\]](#) beim Kommando `/n`.

4.1.9.2 System-Backup-Archiv für Knotendateien

Das System-Backup-Archiv muss der HSMS-Verwalter als öffentliches Archiv mit Leseberechtigung einrichten (USER-ACCESS=*ALL-USERS/ACCESS=*READ). Alle Benutzer können im Lesemodus darauf zugreifen, indem sie den symbolischen Namen SYSNODEBACKUP angeben.

Schreibzugriff für ein System-Backup-Archiv hat nur der HSMS-Verwalter. Deshalb wird die Angabe ACCESS=*WRITE ignoriert.

4.1.9.3 Voll- und Differenzsicherung

Für Knotendateien gelten dieselben Regeln wie für BS2000-Dateien.

Bei einer Differenzsicherung werden die Einträge der Directory-Datei, die mit CNS (cataloged-not-saved) gekennzeichnet sind, interpretiert als „immer noch im Knoten-Dateisystem vorhanden, aber seit der letzten Sicherung nicht geändert“.

Eine Vollsicherung von S0 oder von der letzten Sicherung im Archiv läuft wie jede andere Vollsicherung ab. Von den Metadaten (Eigentums- und Zugriffsrechte, Zeitstempel und andere Dateiattribute) sichert HSMS immer den aktuellen Stand.

Die Art der Sicherung wird durch den Operanden SELECT-FILES der HSMS-Anweisung BACKUP-NODE-FILES gesteuert. Die Angabe SELECT-FILES=*MODIFIED-FILES bzw. *BY-CATALOG-MODIFICATION veranlasst eine Differenzsicherung, die Angabe SELECT-FILES=*ALL-FILES eine Vollsicherung.

4.1.9.4 Automatisches Duplizieren von Knotensicherungsdateien

Bei einer Knotensicherung auf Magnetbandkassette wird die Sicherungsdatei automatisch dupliziert, wenn dem betroffenen Knoten-Backup-Archiv ein Schattenarchiv zugewiesen wurde. Der automatische Dupliziermechanismus ist standardmäßig eingeschaltet (siehe HSMS-Anweisung BACKUP-NODE-FILES, Operand OPERATION-CONTROL).

Die Sicherungsdatei wird am Ende des Sicherungslaufs in das zugeordnete Schattenarchiv kopiert. Die Kopie im Schattenarchiv erhält dieselbe Save-Version-ID und Save-File-ID wie das Original, d.h. sie bekommt denselben Zeitstempel.

Die Directory-Datei des Schattenarchivs kann sich von der des zugehörigen Knoten-Backup-Archivs unterscheiden, weil nicht alle Sicherungen in das Schattenarchiv kopiert werden (Sicherungen auf Platte, Sicherungen mit der Angabe SHADOW-COPY= *INHIBITED, ein neues Schattenarchiv kann mit einem nicht leeren Backup-Archiv verbunden werden).

Wenn eine Sicherungsdatei in einem Knoten-Backup-Archiv erweitert werden soll, versucht der automatische Dupliziermechanismus, die Sicherungsdatei mit derselben SFID im zugeordneten Schattenarchiv ebenfalls zu erweitern. Wenn diese Sicherungsdatei nicht vorhanden ist, wird das Duplizieren nicht durchgeführt.

Mit dem automatischen Dupliziermechanismus lassen sich die Vorsichtsmaßnahmen gegen Datenverlust an die Erfordernisse und die Speicherkapazität eines Rechenzentrums anpassen:

- Höchste Datensicherheit, sehr hohe Kosten in Bezug auf Bandverbrauch und hohe Kosten in Bezug auf Sicherungszeiten:
 - Jedem System-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle Sicherungen werden mit der Angabe SHADOW-COPY=*ALLOWED und dem Archiv-Attribut SHADOW-COPY=*ALLOWED-AND-NEW-SFID durchgeführt.

Ergebnis: Jede Sicherung auf S2 wird automatisch ins Schattenarchiv kopiert und diese Bandkopie sofort danach in ein sicheres Depot ausgelagert.

- Maximale Datensicherheit, aber hohe Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Jedem System-Knoten-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle Sicherungen werden mit der Angabe SHADOW-COPY=*ALLOWED durchgeführt.

Ergebnis: Jede Sicherung auf S2 wird automatisch ins Schattenarchiv kopiert.

- Mittlere Datensicherheit, verbunden mit mittleren Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Jedem System-Knoten-Backup-Archiv wird ein Schattenarchiv zugeordnet.
 - Alle wöchentlichen Vollsicherungen werden mit der Angabe SHADOW-COPY= *ALLOWED durchgeführt.
 - Die dazwischenliegenden Differenzsicherungen werden mit der Angabe SHADOW-COPY=*INHIBITED durchgeführt.

Ergebnis: Nur die wöchentlichen Vollsicherungen auf S2 werden ins Schattenarchiv kopiert.

- Geringere Datensicherheit, verbunden mit geringeren Kosten in Bezug auf Bandverbrauch und Sicherungszeiten:
 - Den System-Knoten-Backup-Archiven werden keine Schattenarchive zugeordnet.

Ergebnis: Der automatische Dupliziermechanismus wird nicht genutzt.

4.1.9.5 Sicherung von LATEST-BACKUPS-OR-S0

In einer Knoten-Umgebung können Sicherungsdateien im Multiplexbetrieb erstellt werden. Wenn eine der letzten Sicherungen, die während der Offline-Kopie als Eingabe verwendet wurde, im Multiplexbetrieb erstellt wurde, muss die neu erstellte Sicherungsdatei als Ganzes auch im Multiplexbetrieb erstellt werden. Dies bedeutet, dass alle nicht im Multiplexbetrieb erstellten Eingabe-Sicherungsversionen, die während der Offline-Kopie verwendet wurden, in der neuen Ausgabe-Sicherungsdatei auch gemultiplext sein müssen, damit die gesamte Struktur der neuen Sicherungsdatei konsistent bleibt. Wenn also eine Vollsicherung mit FROM=*LATEST-BACKUPS-OR-S0 durchgeführt wird und mindestens eine Sicherungsversion im betroffenen HSMS-Archiv gemultiplext ist, dann wird die Sicherung von S0 (also die geänderten Dateien) ebenfalls im Multiplexbetrieb erstellt, auch wenn dies der Benutzer nicht verlangt hat. Der Multiplexfaktor ist in diesem Fall 1.

4.1.10 Gesicherte Knotendateien restaurieren

Zum Restaurieren von gesicherten Knotendateien des BS2000-UFS oder ferner Knoten-S0 steht die HSMS-Anweisung RESTORE-NODE-FILES zur Verfügung.

Ein nicht-privilegierter Benutzer kann nur seine eigenen Knotendateien auf dem lokalen BS2000-UFS restaurieren. Er kann seine Knotendateien aus jedem Backup-Archiv restaurieren, zu dem er Zugriff hat (USER-ACCESS=*ALL-USERS). Zu diesem Zweck steht ihm die HSMS-Anweisung RESTORE-NODE-FILES in eingeschränktem Umfang zur Verfügung.

Der HSMS-Verwalter kann die Knotendateien aller Benutzer auf dem lokalen BS2000-UFS oder auf einem fernen Dateisystem restaurieren, und zwar aus jedem Backup-Archiv.

Wenn Hard Links auf eine Knotendatei restauriert werden sollen, werden die entsprechenden Indexeinträge und Daten beim ersten Durchgang restauriert. Als Nächstes wird nur die Verknüpfung zwischen dem Pfadnamen und dem Indexeintrag restauriert. Wenn die zu restaurierende Knotendatei bereits vor dem Restore-Vorgang auf dem Knoten-S0 vorhanden ist, werden die vorhandenen Hard Links zu diesem Zeitpunkt beibehalten.

4.1.10.1 Sicherungsversionen auswählen

Für Knotendateien gelten dieselben Auswahlregeln wie für BS2000-Dateien.

4.1.10.2 Knotendateien umbenennen

Beim Restaurieren können mit dem Operanden NEW-PATH-NAMES Knotendateien umbenannt werden. Es kann aber nicht eine einzelne Knotendatei umbenannt werden, sondern nur alle Knotendateien eines Restore-Auftrags. Folgende Möglichkeiten gibt es:

- Die Dateinamen können ein Präfix, ein Suffix oder einen neuen Pfadnamen bekommen.
- Die Knotendateien können an einen neuen Knoten-S0 verlagert werden.
- Ein Teil des Dateinamens kann geändert werden.

Wenn die ursprünglichen Dateinamen bei Symbolic Links oder Shortcuts benutzt wurden, so kann es beim Umbenennen zu inkonsistenten Zuständen kommen: Der Inhalt von Symbolic Links oder von Shortcuts wird beim Umbenennen nicht geändert.

4.1.10.3 Aus einem Schattenarchiv restaurieren

Wenn einem Backup-Archiv ein Schattenarchiv zugewiesen wurde, kann das Schattenarchiv in der RESTORE-NODE-FILES-Anweisung benutzt werden. Dies ist hilfreich, wenn gesicherte Dateien des ursprünglichen Archivs zerstört oder die zugehörigen Datenträger nicht zugreifbar sind.

4.1.11 Beispiele zum Backup

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Automatische Voll- und Teilsicherung durch Repeat-Jobs
- Sicherung auf S1 mit und ohne Komprimierung
- Teilsicherung nach Katalogmodifikation
- Versions-Backup in einer SM-Umgebung

Hinweise zum Restore sind jeweils mit angegeben.

Ein Beispiel zur Sicherung des Archivverzeichnisses und Restaurieren eines Archivs samt Archivverzeichnis finden Sie im [Abschnitt „Beispiel zur Langzeitsicherung“](#).

Automatische Voll- und Teilsicherung durch Repeat-Jobs

Wöchentliche Vollsicherungen und tägliche Teilsicherungen werden über automatisch gestartete Repeat-Jobs durchgeführt.

```

/.FULLSAVE SET-LOGON-PARAMETERS TSOS _____ (1)
/ START-HSMS
//MODIFY-ARCHIVE ARCHIVE-NAME=HSMS.BACKUP - _____ (2)
// SAVE-FILES=*DELETE(SAVE-FILE-ID=*BY-ATTR -
// (SAVE-FILE-STATE=*OBSOLETE))
//BACKUP-FILES FILE-NAMES=*ALL, - _____ (3)
// SELECT-FILES=*ALL-FILES, -
// ARCHIVE-NAME=*SYSBACKUP, -
// SAVE-FILE=*NEW(RETENTION-PERIOD=14, -
// USER-ACCESS=*ALL-USERS), -
// TO-STORAGE=*S2-STORAGE-LEVEL, -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=MANUAL.REPORT.FULL, -
// EXPRESS-REQUEST=*YES)
//END
/ SET-JOB-STEP
/ EXIT-JOB SYSTEM-OUTPUT=*DELETE
/.PARTSAVE SET-LOGON-PARAMETERS TSOS _____ (4)
/ START-HSMS
//BACKUP-FILES FILE-NAMES=*ALL, - _____ (5)
// SELECT-FILES=*MODIFIED-FILES(PARTIAL-FILE-SAVE=*YES -
// (MINIMUM-SIZE=200)), -
// ARCHIVE-NAME=*SYSBACKUP, -
// SAVE-FILE=*NEW(RETENTION-PERIOD=14, -
// USER-ACCESS=*ALL-USERS), -
// TO-STORAGE=*S1-STORAGE-LEVEL, -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=MANUAL.REPORT.PART, -
// EXPRESS-REQUEST=*YES)
//END
/ SET-JOB-STEP
/ EXIT-JOB SYSTEM-OUTPUT=*DELETE
/enter-job hsms.e.full-save,proc-admis=(user-id=tsos, - _____ (6)
/ account=adminstr),start=at(date=*today,time=21:00),repeat-job=*weekly
% JMS0066 JOB 'FULLSAVE' ACCEPTED ON 16-08-12 AT 14:27, TSN = 0ABW
/enter-job hsms.e.part-save,proc-admis=(user-id=tsos, - _____ (7)
/ account=adminstr),start=at(time=23:30),repeat-job=*daily
% JMS0066 JOB 'PARTSAVE' ACCEPTED ON 16-08-12 AT 14:27, TSN = 0ABY

```

(1) ENTER-Datei HSMS.E.FULL-SAVE

(2) Vor der Sicherung werden mit MODIFY-ARCHIVE die abgelaufenen Sicherungsdateien freigegeben. Damit stehen Magnetbandkassetten aus dem Datenträger-Pool für den folgenden Sicherungslauf zur Verfügung. Hier kann das Archiv nicht symbolisch angegeben werden.

(3) Die BACKUP-FILES-Anweisung zur wöchentlichen Vollsicherung wird erzeugt: Alle Dateien aller Pubsets werden ganz in das System-Backup-Archiv gesichert; die Sicherung darf 14 Tage nicht gelöscht werden.

(4) ENTER-Datei HSMS.E.PART-SAVE

- (5) Die BACKUP-FILES-Anweisung zur täglichen Differenz- und Teilsicherung wird erzeugt: Von allen Dateien aller Pubsets, die mehr als 200 Seiten groß sind, werden die seit der letzten Vollsicherung geänderten Seiten gesichert. Alle Dateien unter 200 Seiten werden ganz gesichert, wenn sie sich seit der letzten Differenzsicherung geändert haben.
- (6) Der Stapelprozess zur Vollsicherung wird gestartet; er wird wöchentlich wiederholt.
- (7) Der Stapelprozess zur Teilsicherung wird gestartet; er wird täglich wiederholt. Am Tag, an dem beide Prozesse gestartet werden, läuft die Vollsicherung zeitlich früher ab. Die folgende Teilsicherung wird also (fast) leer sein.

Aus dieser Systemsicherung kann sich ein Benutzer den letzten Stand einer Datei, die er z.B. versehentlich gelöscht hat, mit folgender HSMS-Anweisung holen:

```
//RESTORE-FILES FILE-NAMES=<dateiname>
```

Die Datei wird restauriert, ein Report wird, abhängig vom globalen HSMS-Parameter OUTPUT, auf Drucker ausgegeben oder als Anhang einer E-Mail verschickt. Aus diesem Archiv kann ein Benutzer auch jeden Stand einer Datei aus den letzten 14 Tagen restaurieren. Frühere Stände der Datei können durch den Operanden SELECT-SAVE-VERSIONS ausgewählt werden.

Sicherung auf S1 mit und ohne Komprimierung

Dateien werden in ein Backup-Archiv auf die Speicherebene S1 gesichert, einmal mit, einmal ohne Komprimierung der Sicherungsdatei. Die Sicherungsdateien werden verglichen.

Aus der komprimierten Sicherungsdatei werden Dateien restauriert.

```
//BACKUP-FILES FILE-NAMES=$MANUAL.FILE.*, - _____ (1)
// SELECT-FILES=*ALL-FILES, -
// TO-STORAGE=*S1-STORAGE-LEVEL, -
// ARCHIVE-NAME=*SYSBACKUP, -
// SAVE-FILE=*NEW, -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
//   OUTPUT=HSMS.MAN.R.BCF.1, -
//   WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
Report HSMS.MAN.R.BCF.1 (Ausschnitt): _____ (2)

***   BACKUP - FILES           HSMS V11.0           FULL           REPORT   *** 2016-08-12 15:
57:53   PAGE    2
        REQUEST-NAME=BCF#0AAK REQUEST-DATE=2016-08-12 15:57:37 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED WITHOUT ERROR

% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.155740', VERSION '11.0'
% ARC0033 ARCHIVE SUBTASK TSN '0AAL' GENERATED

        SAVE FILE IDENTIFIER - S.160812.155738           SAVE-VERSION-DATE=16-08-12  SAVE-
VERSION-TIME=15:57:38
        SUBSAVE
        NUMBER           VSNS

                0           0:2BC
```

```

/START-HSMS
//BACKUP-FILES FILE-NAMES=$MANUAL.FILE.*, - _____ (3)
// SELECT-FILES=*ALL-FILES, -
// TO-STORAGE=*S1-STORAGE-LEVEL, -
// ARCHIVE-NAME=*SYSBACKUP, -
// SAVE-FILE=*NEW, -
// COMPRESS-FILES=*YES, - _____ (4)
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=HSMS.MAN.R.BCF.2, -
// WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED

//END
% HSM0014 HSMS PROGRAM TERMINATED
Report HSMS.MAN.R.BCF.2 (Ausschnitt): _____ (5)

***   BACKUP - FILES                HSMS V11.0          FULL          REPORT   *** 2016-08-12
15:58:12   PAGE      3
          REQUEST-NAME=BCF#0AAK REQUEST-DATE=2016-08-12 15:57:53 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED WITHOUT ERROR

% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.155755', VERSION '11.0'
% ARC0033 ARCHIVE SUBTASK TSN '0AAM' GENERATED

          SAVE FILE IDENTIFIER - S.160812.155757          SAVE-VERSION-DATE=16-08-12  SAVE-
VERSION-TIME=15:57:57
          SUBSAVE
          NUMBER          VSNS

          0              0:2BC
    
```

```
/SHOW-FILE-ATT FILE-NAME=$MANUAL.FILE.*,INFORMATION=SPACE-SUMMARY      (6)
:2BY: PUBLIC:      21 FILES RES=      270 FRE=      45 REL=      27 PAGES

/SHOW-FILE-ATT FILE=:2BC:$*.ARCHIVE.SAVE.FILE.010812.155738., -
/ INFORMATION=SPACE-SUMMARY
:2BC: PUBLIC:      1 FILE RES=      255 FRE=      2 REL=      0 PAGES

/SHOW-FILE-ATT FILE=:2BC:$*.ARCHIVE.SAVE.FILE.010812.155757., -
/ INFORMATION=SPACE-SUMMARY
:2BC: PUBLIC:      1 FILE RES=      30 FRE=      2 REL=      0 PAGES
//RESTORE-FILES FILE-NAMES=$MANUAL.FILE.*, -      (7)
// REPLACE-FILES-AND-JV=*YES, -
// ARCHIVE-NAME=*SYSBACKUP, -
// SELECT-SAVE-VERSIONS=*LATEST, -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=HSMS.MAN.R.RSF.1, -
// WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED

//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Die Dateien der Benutzerkennung MANUAL mit dem Namen FILE. werden in das System-Backup-Archiv auf S1 gesichert.
- (2) Ausschnitt aus dem von HSMS erzeugten Report des Sicherungslaufs mit Ausgabe der SFID.
- (3) Dieselben Dateien werden noch einmal gesichert, diesmal aber ...
- (4) ... werden die Daten beim Schreiben komprimiert.
- (5) Ausschnitt aus dem von HSMS erzeugten Report des Sicherungslaufs mit Ausgabe der SFID.
- (6) Ein Vergleich der gesicherten Dateien mit den Sicherungsdateien, in die sie geschrieben wurden, zeigt den geringeren Platzbedarf der komprimierten Sicherungsdatei.
- (7) Aus der komprimierten Sicherungsdatei werden die Dateien wieder restauriert. Eine Angabe zur Dekomprimierung ist nicht notwendig; die Daten werden automatisch dekomprimiert.

Teilsicherung nach Katalogmodifikation

Diese Sicherung ähnelt einer Differenzsicherung. Es werden alle Dateien gesichert, die nach dem angegebenen Zeitpunkt geändert wurden.

Im vorliegenden Beispiel werden 5 Dateien (FILE1, FILE2, ..., FILE5) nach diesem Verfahren gesichert:

- Die Dateien werden nach einer Vollsicherung modifiziert bzw. gelöscht und zwischendurch gesichert.
- Nach dem 3.Tag werden die Dateien wieder restauriert.

1. Tag: Datum: 14.04.2016

```
/START-HSMS
//CREATE-ARCHIVE ARCHIVE-NAME=archive, -
//          ALLOWED-USAGE=*BACKUP(SAVE-FILE-STRUCTURE=*SEVERAL-SVID), -
//          DIRECTORY-NAME=archive.dir, -
//          RETENTION-PERIOD=100
//BACKUP-FILES FILE-NAMES=file*, -
//          SELECT-FILES=*ALL-FILES, -
//          ARCHIVE-NAME=archive, -
//          OPERATION-CONTROL=*PAR(REPORT=*FULL,OUTPUT=l.archive)
//END
```

Nach der Vollsicherung wird FILE2 modifiziert und FILE3 gelöscht.

2. Tag: Datum: 15.04.2016

```
/START-HSMS
//BACKUP-FILES FILE-NAMES=file*, -
//          SELECT-FILES=*BY-CATALOG-MODIFICATION -
//          (CHANGED-AFTER=*LATEST-SAVE-VERSION-DATE), -
//          ARCHIVE-NAME=archive, -
//          OPERATION-CONTROL=*PAR(REPORT=*FULL,OUTPUT=l.archive)
//END
```

Nach der Sicherung mit *BY-CATALOG-MODIFICATION (hier alle Dateien, die seit der letzten Sicherung am 14.04. geändert wurden) werden die Dateien FILE2 und FILE4 modifiziert und FILE5 gelöscht.

3. Tag: Datum: 16.04.2016

```
/START-HSMS
//BACKUP-FILES FILE-NAMES=file*, -
//          SELECT-FILES=*BY-CATALOG-MODIFICATION, -
//          ARCHIVE-NAME=archive, -
//          OPERATION-CONTROL=*PAR(REPORT=*FULL,OUTPUT=l.archive)
//END
```

Nach der Sicherung mit *BY-CATALOG-MODIFICATION (hier alle Dateien, die seit der letzten Sicherung am 15.04. geändert wurden) liegt das Archiv mit folgendem Inhalt vor:

```
//SHOW-ARCHIVE ARCHIVE-NAME=archive,SELECT=*SAVE-VERSIONS
```

... (*Ausschnitt der Ausgabe*)

M	SAV-DATE	SAV-TIME	EXP-DATE	SFID	SEL-F	BC	IND	USER-ID	SV-NAME
	16-04-14	10:20:02	16-07-23	S.160414.102001	ALL-F	D		UHU	
	16-04-15	10:12:44	16-07-24	S.160415.101243	CAT-F	D		UHU	
	16-04-16	09:26:12	16-07-25	S.160416.092611	CAT-F	D		UHU	

```
//SHOW-ARCHIVE ARCHIVE-NAME=archive
```

... (*Ausschnitt der Ausgabe*)

M	FILE-NAME	VERS	SAV-DATE	SAV-TIME	EXP-DATE	TYPE
	FILE1	1	16-04-14	10:20:02	16-07-23	FULL
	FILE2	1	16-04-14	10:20:02	16-07-23	FULL
	FILE2	1	16-04-15	10:12:44	16-07-24	FULL
	FILE2	1	16-04-14	10:20:02	16-07-23	FULL
	FILE4	1	16-04-14	10:20:02	16-07-23	FULL
	FILE4	1	16-04-16	09:26:12	16-07-25	FULL
	FILE5	1	16-04-14	10:20:02	16-07-23	FULL

Restore aller Dateien:

```
//RESTORE-FILES FILE-NAMES=*ALL, -
//          ARCHIVE-NAME=archive, -
//          OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=l.restore)
```

Hiermit werden alle Dateien FILE1 bis FILE5 restauriert, unabhängig, ob sie nun zwischenzeitlich gelöscht worden sind.

```

/SHOW-FILE-ATTRIBUTES FILE-NAME=file*,INFORMATION=*PAR(HISTORY=*YES)
%00000003 :SBZ3:$UHU.FILE1
% ----- HISTORY -----
% CRE-DATE   = 2016-02-17  ACC-DATE   = 2016-02-17  CHANG-DATE = 2016-02-17
% CRE-TIME   = 14:15:21   ACC-TIME   = 14:15:21   CHANG-TIME = 14:15:21
% ACC-COUNT  = 1          S-ALLO-NUM = 0
%00000003 :SBZ3:$UHU.FILE2
% ----- HISTORY -----
% CRE-DATE   = 2016-04-15  ACC-DATE   = 2016-04-15  CHANG-DATE = 2016-04-15
% CRE-TIME   = 10:18:32   ACC-TIME   = 10:18:32   CHANG-TIME = 10:18:32
% ACC-COUNT  = 6          S-ALLO-NUM = 0
%00000003 :SBZ3:$UHU.FILE3
% ----- HISTORY -----
% CRE-DATE   = 2016-01-22  ACC-DATE   = 2016-01-22  CHANG-DATE = 2016-01-22
% CRE-TIME   = 13:10:20   ACC-TIME   = 13:10:20   CHANG-TIME = 13:10:20
% ACC-COUNT  = 4          S-ALLO-NUM = 0
%00000003 :SBZ3:$UHU.FILE4
% ----- HISTORY -----
% CRE-DATE   = 2016-04-15  ACC-DATE   = 2016-04-15  CHANG-DATE = 2016-04-15
% CRE-TIME   = 10:18:52   ACC-TIME   = 10:18:52   CHANG-TIME = 10:18:52
% ACC-COUNT  = 5          S-ALLO-NUM = 0
%00000003 :SBZ3:$UHU.FILE5
% ----- HISTORY -----
% CRE-DATE   = 2016-02-17  ACC-DATE   = 2016-02-17  CHANG-DATE = 2016-02-17
% CRE-TIME   = 13:42:11   ACC-TIME   = 13:42:11   CHANG-TIME = 13:42:11
% ACC-COUNT  = 3          S-ALLO-NUM = 0
%:SBZ3: PUBLIC:      5 FILES RES=      15 FREE=      10 REL=      0 PAGES

```

Restore mit der letzten Save-Version:

```

//RESTORE-FILES FILE-NAMES=*ALL, -
//          ARCHIVE-NAME=archive, -
//          SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES, -
//          OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=l.restore)

```

Hiermit werden nur die Dateien restauriert, welche in der letzten Save-Version enthalten sind. Früher gesicherte Dateien werden nicht berücksichtigt.

```

/SHOW-FILE-ATTRIBUTES FILE-NAME=file*
%          3 :SBZ3:$UHU.FILE2
%          3 :SBZ3:$UHU.FILE4

```

Versions-Backup in SM-Umgebung

In diesem Beispiel wird das Versions-Backup in einer SM-Umgebung betrachtet.

```

/START-HSMS
//CRA ARCHIVE-NAME=SYSVERS.SMS,ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),- _____ (1)
//  ALLOWED-USAGE=*VERSIONBACKUP,-
//  USER-ACCESS=*ALL-USERS(ACCESS=*WRITE),-
//  DIRECTORY-NAME=:SMS:SYSVERS.SMS.DIR,RETENTION-PERIOD=365

//MSP SM-PUBSET-ID=SMS,SYSVERSION=SYSVERS.SMS _____ (2)

//CREATE-MANAGEMENT-CLASS ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),- _____ (3)
//  MANAGEMENT-CLASS=SOURCES,-
//  MIGRATION-ATTRIBUTES=*PARAMETERS(FROM-S0=*PARAMETERS(UNUSED-DAYS=0)), -
//  BACKUP-ATTRIBUTES=*PARAMETERS(RETENTION-PERIOD=365), -
//  VERSION-BACKUP-ATTR=*PARAMETERS(NUM-OF-BACKUP-VERS=20)
//CREATE-MANAGEMENT-CLASS ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),-
//  MANAGEMENT-CLASS=LISTINGS,-
//  MIGRATION-ATTRIBUTES=*PARAMETERS(FROM-S0=*PARAMETERS(UNUSED-DAYS=0)), -
//  BACKUP-ATTRIBUTES=*PARAMETERS(RETENTION-PERIOD=365), -
//  VERSION-BACKUP-ATTR=*PARAMETERS(NUM-OF-BACKUP-VERS=10)
//END

/MODIFY-FILE-ATTRIBUTES FILE-NAME=:SMS:*.SRC*,- _____ (4)
/  SUPPORT=*PUBLIC-DISK(MANAGEMENT-CLASS=SOURCES)
/MODIFY-FILE-ATTRIBUTES FILE-NAME=:SMS:*.LST*, -
/  SUPPORT=*PUBLIC-DISK(MANAGEMENT-CLASS=LISTINGS)

```

- (1) Der HSMS-Verwalter erstellt ein öffentliches Versions-Backup-Archiv in der SM-Umgebung.
- (2) Das Versions-Backup-Archiv wird als Systemarchiv für Versions-Backup in der SM-Umgebung definiert.
- (3) Es werden zwei Management-Klassen mit verschiedenen Werten des Operanden NUM-OF-BACKUP-VERS erstellt.
- (4) Einigen Dateien werden Management-Klassen zugewiesen.

Betrachten wir folgende Dateien und ihre Backup-Attribute. Einigen Dateien sind Management-Klassen zugewiesen, einigen nicht. All diese Dateien haben einen bestimmten Wert des Operanden NUM-OF-BACKUP-VERS in ihren Katalogeinträgen. Die Werte unterscheiden sich von den Werten in den Management-Klassen:

```
/sh :sms:*.lst*, back=*y
```



```
%0000000003 :SMS:$TSOS.NB.FILE1.LST
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = LISTINGS
% #BACK-VERS  = 20
%0000000003 :SMS:$TSOS.NB.FILE2.LST
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = LISTINGS
% #BACK-VERS  = 20
%0000000003 :SMS:$TSOS.NB.FILE3.LST
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = LISTINGS
% #BACK-VERS  = 20
```

```
/sh :sms:*.src*, back=*y
```

```
%0000000003 :SMS:$TSOS.NB.FILE1.SRC
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = SOURCES
% #BACK-VERS  = 15
%0000000003 :SMS:$TSOS.NB.FILE2.SRC
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = SOURCES
% #BACK-VERS  = 15
%0000000003 :SMS:$TSOS.NB.FILE3.SRC
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED  MAN-CLASS = SOURCES
% #BACK-VERS  = 15
%:SMS: PUBLIC:      8 FILES RES=      24 FRE=      16 REL=      0 PAGES
```

```
/sh :sms:*.rep*, back=*y
```

```
%0000000003 :SMS:$TSOS.NB.FILE1.REP
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG = COMPL-FILE  VERSION    = 1
% MIGRATE      = ALLOWED
% #BACK-VERS  =
5
```

Führen Sie Versions-Backup mit der Anweisung **BACKUP-FILE-VERSIONS** aus und geben **MANAGEMENT-CLASS** mit ***ANY** an.

```
/HSMS
//BFV FILE-NAMES=( :SMS:NB.*SRC*, :SMS:NB.*LST*, :SMS:NB.*REP* ), -
//ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS) , -
//MANAGEMENT-CLASS=*ANY, -
//TO-STORAGE=*S1-STORAGE-LEVEL, -
//OPERATION-CONTROL=*PARAMETERS( REPORT=*FULL, OUTPUT=:SMS:REP1)
```

Nach der Beendigung des Auftrags analysieren Sie den Inhalt des Versions-Backup-Archivs:

```
//SHA ARCHIVE-NAME=SYSVERS.SMS, -
//ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS) , -
//SELECT=*FILES( INFORMATION=*VERSION)
```

Es ist zu sehen, dass die Werte von NUM-OF-BACK-VERS den Management-Klassen entnommen sind, die den Dateien zugeordnet wurden. Die Dateien, denen wiederum keine Management-Klassen zugewiesen wurden, haben denselben Wert wie im Katalogeintrag.

```
SHOW-ARCHIVE (FILES)                                SHOW-FILE-VERSIONS = DIFFERENT
ENVIRONMENT      = SM(SMS)                          ARCHIVE-NAME = $SYSHSMS.SYSVERS.SMS
CATALOG-ID       = SMS                              SV-NAME      = ANY
USER-ID          = TSOS                             SV-DATE      = INTERVAL  EARLIEST LATEST
FILE-SAVE-STATE = ANY                               EXP-DATE     = ANY
INFORMATION      = VERSION

-----
M FILE-NAME      V#  O-DATE O-TIME NUM OBS M S D-DATE
NB.FILE1.LST    1   190301 150636 10  NO  N  Y
NB.FILE1.REP    1   190301 150636  5  NO  N  Y
NB.FILE1.SRC    1   190301 150636 20  NO  N  Y
NB.FILE2.LST    1   190301 150636 10  NO  N  Y
NB.FILE2.REP    1   190301 150636  5  NO  N  Y
NB.FILE2.SRC    1   190301 150636 20  NO  N  Y
NB.FILE3.LST    1   190301 150636 10  NO  N  Y
NB.FILE3.SRC    1   190301 150636 20  NO  N  Y
```

Hier ist der Report des Versions-Backup-Laufs:

```
A***  BACKUP-FILE-VERSIONS                HSMS V12.0                FULL                REPORT  *** 2019-03-
01 15:13:51    PAGE 1
      REQUEST-ENVIRONMENT=SM(SMS)
      REQUEST-NAME=BFV#2917 REQUEST-DATE=2019-03-01 15:13:46 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED

STATEMENT LISTING:

BFV      FILE-NAMES=( :SMS:NB.*SRC*, :SMS:NB.*LST*, :SMS:NB.*REP* ),
ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS) ,
      MANAGEMENT-CLASS = *ANY, TO-STORAGE = *S1-STORAGE-LEVEL, OPERATION-CONTROL =
*PARAMETERS( REPORT = *FULL,
      OUTPUT=:SMS:NB.REP2)

SAVE-FILE ATTRIBUTES
```

```

TO-STORAGE                : S1-STORAGE-LEVEL
RETENTION-PERIOD          : 365

SAVE-VERSION ATTRIBUTES
SAVE-VERSION-NAME        :

SELECT-FILES PARAMETER
MODIFIED-FILES PARTIAL-FILE-SAVE : NO
MAX-BACKUP-CLASS        : STD
% HSM0030 REQUEST 'BFV#2917' CREATED IN ENVIRONMENT 'SM(SMS)' WITH DATE '2019-03-01' AND
TIME '15:13:46'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.190301.151347', VERSION '12.0A10'
% ARC0802 COMPUTED BLOCK-SIZE VALUE IS '239'

% ARC0033 ARCHIVE SUBTASK TSN '6AG5' GENERATED
% ARC0815 SUBTASK '0' HAS TRANSFERRED '0' PAM PAGES FOR '0' FILES AND '0' JVS IN '0'
SECONDS
% ARC0037 SPECIFIED FILE NAME ':SMS:$TSOS.NB.*LST*' NOT PROCESSED
% ARC0037 SPECIFIED FILE NAME ':SMS:$TSOS.NB.*REP*' NOT PROCESSED
A*** BACKUP-FILE-VERSIONS          HSMS V12.0          FULL          REPORT *** 2019-03-
01 15:13:51          PAGE 2
      REQUEST-ENVIRONMENT=SM(SMS)
      REQUEST-NAME=BFV#2917 REQUEST-DATE=2019-03-01 15:13:46 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED WITH ERRORS
              *** CATALOG - SMS          USER - TSOS          ***
              FILE/JOB VARIABLE NAME          LASTPG/          SAVE VERSION          SAVE          INPUT
SUB OUTPUT
              VERS          SIZE          IDENTIFIER          TYPE          VSN
SAVE DISK(S)
      NB.FILE1.SRC          1          1          190301.151347          SMK.00
0 DIR-UPD: #BACK-VERS
      NB.FILE2.SRC          1          1          190301.151347          SMK.00
0 DIR-UPD: #BACK-VERS
      NB.FILE3.SRC          1          1          190301.151347          SMK.00
0 DIR-UPD: #BACK-VERS

      NUMBER OF FILES=          3 GLOBAL SIZE=          3 START= 2019-03-01 15:13:47
END= 2019-03-01 15:13:51

***          E N D          O F          HSMS V12.0          FULL          REPORT *** 2019-03-
01 15:13:51          ***

```

Ändern Sie die Versions-Attribute der Management-Klasse SOURCES.

```

//MODIFY-MANAGEMENT-CLASS ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),-
//MANAGEMENT-CLASS=SOURCES,-
//VERSION-BACKUP-ATTR=*PARAMETERS(NUM-OF-BACKUP-VERS=32)

```

Und Wiederholen den Lauf ohne vorher die Dateien geändert zu haben.

```
//BFV FILE-NAMES=( :SMS:NB.*SRC*, :SMS:NB.*LST*, :SMS:NB.*REP* ), -
//ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS) ,-
//MANAGEMENT-CLASS=*ANY, -
//TO-STORAGE=*S1-STORAGE-LEVEL, -
//OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=:SMS:NB.REP2)
```

Im Ergebnis wurden keine Dateien gesichert. Die Dateien, denen die aktualisierten Management-Klassen zugewiesen worden waren, sind dennoch betroffen: der neue Wert von NUM-OF-BACKUP-VERS wurde ins Archivverzeichnis geschrieben.

```
A*** BACKUP-FILE-VERSIONS          HSMS V12.0          FULL          REPORT *** 2019-03-
01 15:13:51      PAGE      1
      REQUEST-ENVIRONMENT=SM(SMS)
      REQUEST-NAME=BFV#2917 REQUEST-DATE=2019-03-01 15:13:46 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED
```

STATEMENT LISTING:

```
BFV          FILE-NAMES=( :SMS:NB.*SRC*, :SMS:NB.*LST*, :SMS:NB.*REP* ),
ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),
      MANAGEMENT-CLASS = *ANY, TO-STORAGE = *S1-STORAGE-LEVEL, OPERATION-CONTROL =
*PARAMETERS(REPORT = *FULL,
      OUTPUT=:SMS:NB.REP2)
```

SAVE-FILE ATTRIBUTES

```
TO-STORAGE          : S1-STORAGE-LEVEL
RETENTION-PERIOD    : 365
```

SAVE-VERSION ATTRIBUTES

```
SAVE-VERSION-NAME  :
```

SELECT-FILES PARAMETER

```
MODIFIED-FILES PARTIAL-FILE-SAVE : NO
MAX-BACKUP-CLASS : STD
```

```
% HSM0030 REQUEST 'BFV#2917' CREATED IN ENVIRONMENT 'SM(SMS)' WITH DATE '2019-03-01' AND
TIME '15:13:46'
```

```
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.190301.151347', VERSION '12.0A10'
```

```
% ARC0802 COMPUTED BLOCK-SIZE VALUE IS '239'
```

```
% ARC0033 ARCHIVE SUBTASK TSN '6AG5' GENERATED
```

```
% ARC0815 SUBTASK '0' HAS TRANSFERRED '0' PAM PAGES FOR '0' FILES AND '0' JVS IN '0'
SECONDS
```

```
% ARC0037 SPECIFIED FILE NAME ':SMS:$TSOS.NB.*LST*' NOT PROCESSED
```

```
% ARC0037 SPECIFIED FILE NAME ':SMS:$TSOS.NB.*REP*' NOT PROCESSED
```

```
A*** BACKUP-FILE-VERSIONS          HSMS V12.0          FULL          REPORT *** 2019-03-
01 15:13:51      PAGE      2
      REQUEST-ENVIRONMENT=SM(SMS)
      REQUEST-NAME=BFV#2917 REQUEST-DATE=2019-03-01 15:13:46 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED WITH ERRORS
```

```
*** CATALOG - SMS          USER - TSOS          ***
FILE/JOB VARIABLE NAME          LASTPG/    SAVE VERSION    SAVE    INPUT
SUB OUTPUT
```

```

                                VERS      SIZE      IDENTIFIER      TYPE      VSN
SAVE DISK(S)

  NB.FILE1.SRC                    1          1  190301.151347      SMK.00
0 DIR-UPD: #BACK-VERS
  NB.FILE2.SRC                    1          1  190301.151347      SMK.00
0 DIR-UPD: #BACK-VERS
  NB.FILE3.SRC                    1          1  190301.151347      SMK.00
0 DIR-UPD: #BACK-VERS

      NUMBER OF FILES=          3 GLOBAL SIZE=          3 START= 2019-03-01 15:13:47
END= 2019-03-01 15:13:51

***      E N D      O F          HSMS V12.0          FULL          REPORT *** 2019-03-
01 15:13:51      ***

```

Prüfen Sie den Inhalt des Version-Backup-Archivs:

```

//SHA ARCHIVE-NAME=SYSVERS.SMS,-
//ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS),-
//SELECT=*FILES(INFORMATION=*VERSION)

```

Die Dateien, denen die geänderten Management-Klassen zugewiesen worden waren, haben jetzt neue Werte von NUM-OF-BACK-VERS (siehe Spalte "NUM"):

```

SHOW-ARCHIVE (FILES)                                SHOW-FILE-VERSIONS = DIFFERENT
ENVIRONMENT      = SM(SMS)                          ARCHIVE-NAME = $SYSHSMS.SYSVERS.SMS
CATALOG-ID       = SMS                              SV-NAME      = ANY
USER-ID          = TSOS                             SV-DATE      = INTERVAL  EARLIEST LATEST
FILE-SAVE-STATE = ANY                              EXP-DATE     = ANY
INFORMATION      = VERSION
-----
M FILE-NAME                                         V#  O-DATE O-TIME NUM OBS M S D-DATE
NB.FILE1.LST                                       1   190301 150636 10 NO N Y
NB.FILE1.REP                                       1   190301 150636  5 NO N Y
NB.FILE1.SRC                                       1   190301 150636 32 NO N Y
NB.FILE2.LST                                       1   190301 150636 10 NO N Y
NB.FILE2.REP                                       1   190301 150636  5 NO N Y
NB.FILE2.SRC                                       1   190301 150636 32 NO N Y
NB.FILE3.LST                                       1   190301 150636 10 NO N Y
NB.FILE3.SRC                                       1   190301 150636 32 NO N Y
    
```

4.2 Langzeitarchivierung (Archival)

Archivierung ist die langfristige Auslagerung von Dateien und Jobvariablen, die online auf der Verarbeitungsebene nicht mehr benötigt werden, auf die Speicherebenen S1 oder S2, auf gemeinschaftliche Platten oder Net-Storage. Durch die Archivierung kann die Verarbeitungsebene entlastet werden, indem die Dateien und Jobvariablen nach der Archivierung gelöscht werden. Langzeitarchivierung dient auch der Dokumentation, z.B. wenn Daten wegen rechtlicher Vorschriften über bestimmte Zeiträume hinweg aufbewahrt werden müssen.

Für die archivierten Dateien lassen sich Verfallsdaten vereinbaren, die von der physischen Schutzfrist des Datenträgers unabhängig sind.

Die Langzeitarchivierung steht allen Benutzern zur Verfügung, wenn das Archiv entsprechend eingerichtet wird.

4.2.1 BS2000-Dateien archivieren

BS2000-Dateien können mit der HSMS-Anweisung ARCHIVE-FILES archiviert werden.

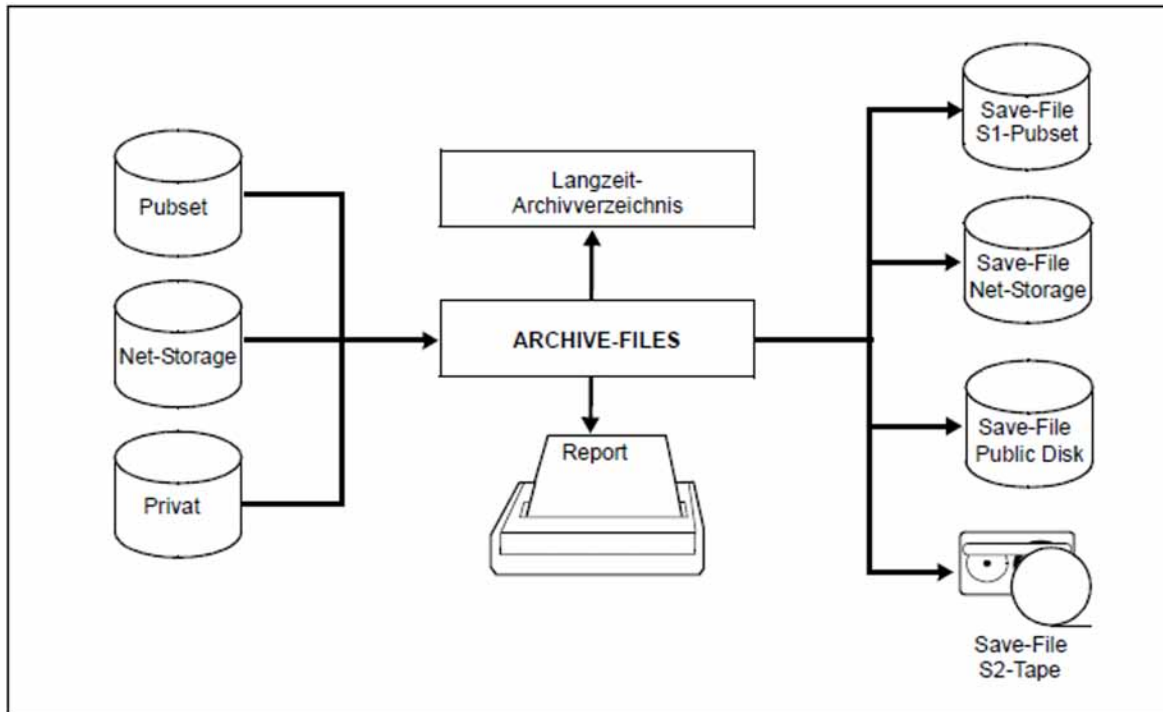


Bild 11: Archivieren mit ARCHIVE-FILES

Ein globales Langzeitarchiv, das in der SF-Umgebung definiert ist, kann entweder SF- oder SM-Dateien enthalten. Ein Langzeitarchiv, das für den SM-Pubset lokal ist, kann nur SM-Dateien enthalten.

Wenn sich die HSMS-Anweisung ARCHIVE-FILES auf ein Archiv bezieht, das in den zentralen HSMS-Parametern definiert ist, können die zu bearbeitenden Dateien entweder in der SF- oder in der SM-Umgebung liegen. Wenn sich die HSMS-Anweisung ARCHIVE-FILES auf ein lokales SM-Pubset-Archiv bezieht, können die zu bearbeitenden Dateien nur in dieser SM-Umgebung liegen.

Ein lokales SM-Pubset-Archiv kann nur auf den Speicherebenen S1 oder S2 abgelegt sein. Andere Angaben im Operanden TO-STORAGE werden abgewiesen.

Bei der ARCHIVE-FILES-Anweisung gibt es die Operanden FILE-NAMES, EXCEPT-FILE-NAMES und SELECT-FILES, um die zu archivierenden BS2000-Dateien auszuwählen.

Möglich ist selbstverständlich die Angabe einer Dateinamensliste und die Auswahl von BS2000-Dateien in Bildschirmmasken (siehe [Abschnitt „Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien“](#)).

Migrierte BS2000-Dateien können ebenfalls archiviert werden. Dabei werden die Katalogeinträge aus dem Katalog und die Daten von der Migrationsebene übernommen. Mit dem Operanden SELECT-FILES in der ARCHIVE-FILES-Anweisung können gezielt BS2000-Dateien archiviert werden, die bereits eine vorgebbare Zeitdauer migriert sind. Die migrierten BS2000-Dateien werden dabei direkt aus dem Migrationsarchiv ins Langzeitarchiv übernommen.

Mit der HSMS-Anweisung ARCHIVE-FILES können Sie auch Dateien archivieren, bei denen Sie Miteigentümer sind (siehe Handbuch „SECOS“ [\[16\]](#)).

Sie können bei der HSMS-Anweisung ARCHIVE-FILES mit dem Operanden JV-NAMES auch Jobvariablen angeben. Diese werden genauso wie bei der HSMS-Anweisung BACKUP-FILES gesichert.

Wenn der Archiveigentümer die Archivierung durchführt, kann er über den Operanden SAVE-FILE bestimmen, ob

- die zu archivierenden BS2000-Dateien in die Standard-Sicherungsdatei des Archivs geschrieben werden sollen (=*STD).
- eine Sicherungsdatei für diese Archivierung neu erzeugt werden soll (=*NEW).
- eine existierende Sicherungsdatei fortgeschrieben werden soll (=*CONTINUE).

Das Fortschreiben wird nur für Sicherungsdateien auf der Speicherebene S2 unterstützt.

Nicht-privilegierte Benutzer können, wenn sie nicht Archiveigentümer sind, nur mit dem Operanden SAVE-FILE=*STD die Archivierung durchführen.

Mit HSMS können auch große Dateien (≥ 32 GB) archiviert werden. Dazu sind keine besonderen Vorkehrungen erforderlich.

Beim Restaurieren von großen Dateien sind aber Besonderheiten zu berücksichtigen (siehe [Abschnitt „Restaurieren von großen Dateien \(> 32 GB\)“](#)).

Verschlüsselte Dateien werden in verschlüsselter Form gesichert und in der ursprünglichen verschlüsselten Form wieder restauriert. Zum Sichern bzw. Restaurieren muss das zugehörige Crypto-Kennwort nicht angegeben werden. Für weitere Zugriffe auf die restaurierte Datei muss das Crypto-Kennwort wieder angegeben werden.

4.2.1.1 System-Langzeitarchiv

Die archivierten Daten werden in Langzeitarchiven verwaltet, die der HSMS-Verwalter als öffentliches Archiv mit Schreibberechtigung eingerichtet hat (USER-ACCESS= *ALL-USERS und ACCESS=*WRITE).

System-Langzeitarchive für BS2000-Dateien werden in der Regel über den symbolischen Namen SYSARCHIVE angesprochen, der bei den HSMS-Anweisungen voreingestellt ist.

4.2.1.2 Logisches Freigabedatum

Bei einem Archivierungsauftrag kann für die archivierten Dateien mit dem Operanden FILE-EXPIRATION-DATE ein logisches Freigabedatum vereinbart werden. Darunter versteht man eine logische Schutzfrist, die von der physischen Schutzfrist des Datenträgers, der die Sicherungsversion enthält, abweichen kann. Die logische Schutzfrist kann länger sein als die physische Schutzfrist der Datenträger, wenn dies der HSMS-Verwalter für öffentliche Langzeitarchive so vereinbart hat. In diesem Fall muss der HSMS-Verwalter nach Ablauf der physischen Schutzfrist den Schutz der Datenträger vor Überschreiben durch administrative Maßnahmen gewährleisten:

- Dies kann z.B. durch Reorganisieren der Langzeitarchive und durch Umwälzen der Datenträger vor Ablauf der Schutzfrist geschehen. Unter Umwälzen versteht man das Kopieren von Sicherungsversionen auf Datenträger mit höherem Verfallsdatum. Beim Umwälzen können gezielt nur die Sicherungsversionen übernommen werden, deren logisches Freigabedatum noch nicht erreicht ist.
- Eine andere Möglichkeit besteht darin, den Operanden SAVE-FILE-RETPD-UPDATE zu verwenden. Mit diesem Operanden kann die Schutzfrist einer Sicherungsdatei automatisch erhöht werden, während eine neue Sicherungsversion erstellt oder das logische Freigabedatum geändert wird. Diese Änderung betrifft auch den MAREN-Katalog.

Mit der Funktion SAVE-FILES=*RETENTION-PERIOD(...) in der HSMS-Anweisung MODIFY-ARCHIVE kann die Schutzfrist einer Sicherungsdatei ebenfalls verlängert werden. Diese Verlängerung betrifft bei Sicherungsdateien auf Magnetbandkassette auch den MAREN-Katalog-Eintrag des Datenträgers und bei Sicherungsdateien auf Platte den Katalogeintrag.

Das logische Freigabedatum für eine Sicherungsversion kann auf zwei Arten festgelegt werden:

- Implizit ohne Angabe des Operanden FILE-EXPIRATION-DATE oder durch Angabe von FILE-EXPIRATION-DATE=*STD
In diesem Fall wird das Freigabedatum folgendermaßen berechnet:
Erstellungsdatum + Schutzfrist [+ Fortsetzungsperiode]
(Creation-Date + Retention-Period [+ Continuation-Period])
- Durch explizite Angabe eines Wertes (ungleich *STD) im Operanden FILE-EXPIRATION-DATE

Falls das Freigabedatum einer archivierten Datei/Jobvariable (siehe Katalogeintrag EXPIR-DATE der Datei) höher als das für die Sicherungsversion festgelegte Freigabedatum ist, wird eine Warnung ausgegeben und der Archivierungsauftrag erhält den Substatus COMPLETED WITH WARNINGS.

4.2.1.3 Zusatzinformationen

Da die archivierten Daten teilweise recht lange in den Archiven aufbewahrt werden, kann es sinnvoll sein, Informationen zu diesen Daten zusammen mit der Sicherung zu hinterlegen. Bei der Archivierung können ein Kurzbeschreibungstext (Operand DESCRIPTOR) sowie ein längerer Kommentar (Operand USER-INFORMATION) eingegeben werden, die die Sicherungsversion beschreiben und Informationen zu den archivierten Dateien enthalten. Diese Beschreibungen können mit SHOW-ARCHIVE ausgegeben werden. Dabei können Sicherungsversionen ausgewählt werden, die eine bestimmte Zeichenfolge im Kommentarfeld enthalten (Operand SEARCH-STRING).

Eine nachträgliche Änderung der Zusatzinformationen ist möglich mit:

```
//MODIFY-ARCHIVE . . . ,SAVE-VERSIONS=*MODIFY-USER-RECORD(. . .)
```

Das Datum der ursprünglichen Sicherung bleibt unverändert, wenn eine Sicherungsdatei kopiert wird. Dieses ursprüngliche Sicherungsdatum kann deshalb zur Auswahl einer Sicherungsversion in den Anweisungen COPY-SAVE-FILE und RESTORE-SAVE-FILE benutzt werden.

4.2.1.4 Archivierungsoptionen

Das Archivverzeichnis, das beim Archivieren auf den neuesten Stand gebracht wurde, kann am Ende der Archivierung mit auf den Archivierungsdatenträger geschrieben werden. Der Archivierungsdatenträger enthält dann außer den gesicherten Daten auch ein Verzeichnis der auf ihm enthaltenen Daten.

Das Archivverzeichnis wird durch die Angabe `SAVE-DIRECTORY=*YES` gesichert. Standardmäßig wird es nicht mitgesichert.

Die Sicherung des Archivverzeichnisses ist erlaubt für den HSMS-Verwalter, sowie für den Archiveigentümer und den Miteigentümer des Archivverzeichnisses.

Bei einer Sicherung des Archivverzeichnisses gilt Folgendes:

- Wenn bei dem Auftrag keinerlei Daten gesichert wurden, wird auch das Archivverzeichnis nicht gesichert.
- Das Archivverzeichnis wird in jedem Fall vollständig gesichert, auch bei Differenzsicherungen.
- Das Archivverzeichnis wird nicht auf verschiedene Datenträger verteilt. Die Sicherung des Archivverzeichnisses und der dafür verwendete Datenträger werden im Report angezeigt.

Wenn das Archivverzeichnis gesichert wurde, kann es im Notfall (nicht mehr vorhanden oder nicht lesbar auf S0) vom Datenträger restauriert werden. Dazu dient die Anweisung `IMPORT-FILES` mit der Angabe von `FILE-NAMES=*DIRECTORY` und des im Report der Sicherung vermerkten Datenträgers.

Außerdem werden die Attribute eines Archivs zusätzlich im Archivverzeichnis abgespeichert. Nach einem Verlust der Archivdefinition kann das Archiv mit der Anweisung `CREATE-ARCHIVE` aus dem Archivverzeichnis wiederhergestellt werden:

```
//CREATE-ARCHIVE ARCHIVE-NAME=... ,  
    DIRECTORY-NAME=<directory>(NEW-DIRECTORY=*NO(RECONSTRUCT-ARCHIVE=*YES))
```

Um aus der Langzeitsicherung von großen PLAM-Bibliotheken auch selektiv einzelne Elemente restaurieren zu können (mit `RESTORE-LIBRARY-ELEMENTS`), muss die Sicherung mit der Option `SAVE-PLAM-INFO=*YES` erfolgen. Dies ist der Fall, wenn das Langzeit-Archiv zum Zeitpunkt der Sicherung das Attribut `SAVE-PLAM-INFO=*YES` besitzt (siehe `CREATE-ARCHIVE` bzw. `MODIFY-ARCHIVE-ATTRIBUTES`).

Besonderheiten bei SAM-Node-Files

Von BS2000 kann grundsätzlich auf zwei verschiedene Arten auf eine SAM-Node-File zugegriffen werden:

- so, dass die BS2000-Anwendung die SAM-Strukturen erhält, indem diese beim Lesen durch den Net-Client während der Übertragung vom Net-Server wieder hinzugefügt werden, oder
- ohne Konvertierung, d.h. ohne hinzufügen der SAM-Strukturen in die Datenblöcke und ohne Code-Konvertierung nach EBCDIC; dies ist der sogenannte RAW-Modus.

Die Art des Zugriffs wird mit dem Operanden `SAVE-SAM-STRUCTURE=*NO` der Option `SAVE-OPTIONS` gesteuert. Beachten Sie hierzu den Abschnitt „Besonderheiten bei SAM-Node-Files“, „[Sicherungsoptionen](#)“.

4.2.1.5 Automatisches Duplizieren in ein Schattenarchiv

Die Sicherungsdatei eines Langzeitarchivs für BS2000-Dateien kann automatisch dupliziert werden, wenn sie auf der Speicherebene S2 abgelegt ist und dem betroffenen Langzeitarchiv ein Schattenarchiv zugewiesen wurde. Dabei muss der automatische Dupliziermechanismus in der HSMS-Anweisung ARCHIVE-FILES eingeschaltet sein. Dazu muss in der HSMS-Anweisung ARCHIVE-FILES beim Operanden OPERATION-CONTROL für SHADOW-COPY der Wert *ALLOWED angegeben sein (Default-Einstellung).

Die Sicherungsdatei wird am Ende des Archivierungslaufs in das zugeordnete Schattenarchiv kopiert. Die Kopie im Schattenarchiv erhält dieselbe Save-Version-ID und Save-File-ID wie das Original, d.h. sie bekommt denselben Zeitstempel.

Wenn die Dateien nach der Archivierung gelöscht werden sollen (DELETE-FILES=*YES), wird das Löschen erst durchgeführt, nachdem die Sicherungsdatei kopiert wurde.

Wenn eine Sicherungsdatei in einem Langzeitarchiv erweitert werden soll, versucht der automatische Dupliziermechanismus, die Sicherungsdatei mit derselben SFID im zugeordneten Schattenarchiv ebenfalls zu erweitern. Wenn diese Sicherungsdatei nicht vorhanden ist, wird das Duplizieren (und das eventuelle Löschen der Dateien) nicht durchgeführt und die Meldung HSM0318 ausgegeben.

4.2.1.6 Hinweise zur Verwendung von Standard-Sicherungsdateien

Standard-Sicherungsdateien werden nur auf der Speicherebene S2 abgelegt.

Wenn bei einer Archivierung mit `SAVE-FILE=*STD` im Operanden `TO-STORAGE` ein Wert ungleich `*S2-STORAGE-LEVEL` angegeben ist, wird eine neue Sicherungsdatei entsprechend der Angabe von `TO-STORAGE` angelegt.

Wenn eine Standard-Sicherungsdatei entsprechend den Archivattributen regelmäßig gesichert wird, wird sie bei nachfolgenden Archivierungen mit `TO-STORAGE=*S2-STORAGE-LEVEL` fortgeschrieben.

4.2.2 Archivierte BS2000-Dateien restaurieren

Mit der HSMS-Anweisung RESTORE-FILES können Sie archivierte BS2000-Dateien restaurieren.

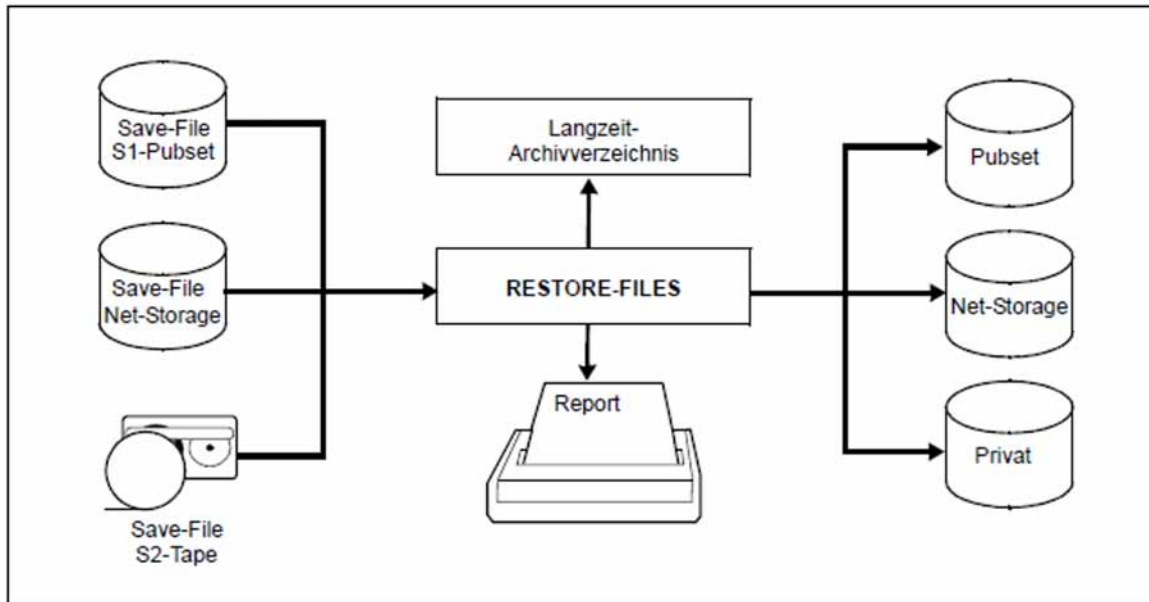


Bild 12: Restaurieren mit RESTORE-FILES

Beim Restaurieren aus Langzeitarchiven dürfen Objekte von SF- und SM-Pubsets im selben Lauf enthalten sein, da die Archivierungsfunktion nicht auf eine bestimmte Umgebung beschränkt ist.

Ein nicht-privilegierter Benutzer kann Daten seiner eigenen Benutzerkennung aus Standard-Systemarchiven oder anderen ihm zugänglichen Archiven restaurieren. Außerdem kann er auch Daten einer fremden Benutzerkennung aus dem Standard-Systemarchiv oder anderen ihm zugänglichen Archiven restaurieren, wenn er Miteigentümer dieser Daten ist. Die Daten können entweder auf seine eigene Benutzerkennung oder auf eine andere restauriert werden. Nähere Informationen zur Miteigentümerschaft finden Sie im Handbuch „SECOS“ [16].

Die HSMS-Anweisung RESTORE-FILES steht einem nicht-privilegierten Benutzer nur in eingeschränktem Umfang zur Verfügung.

Beim Restore aus einem Langzeitarchiv werden die Dateien mit den DVS-Standard-Katalogeigenschaften angelegt. Insbesondere erhalten Dateien und Jobvariablen ein neues Erstellungsdatum. Die Kennwörter werden allerdings aus dem Archiv übernommen. Optional können die Dateien durch Angabe von DATE-AND-PROTECTION=*ORIGINAL-ATTRIBUTES mit ihren Original-Katalogeigenschaften restauriert werden (insbesondere mit dem ursprünglichen Erstellungsdatum).

4.2.2.1 Restaurieren von großen Dateien (> 32 GB)

Große Dateien (≥ 32 GB) können nur auf Pubsets mit den Attributen `LARGE-OBJECTS= *YES` und `LARGE-FILES-ALLOWED=*YES` restauriert werden.

Wenn eine Datei ≥ 32 GB auf einen Pubset mit dem Attribut `LARGE-FILES-ALLOWED= *NO` eingespielt werden soll, wird dies immer mit Fehler quittiert. Dies gilt auch, wenn die Datei z.B. über eine Mengenselektion wie Teilqualifizierung oder Wildcards implizit ausgewählt wurde.

Sicherungsbestände aus älteren BS2000-Versionen können auch in höheren BS2000-Versionen uneingeschränkt restauriert werden, also auf Standard-Pubsets und auf Pubsets mit dem Attribut `LARGE-OBJECTS=*YES`.

4.2.2.2 Sicherungsversionen auswählen

Die Auswahl der Sicherungsversionen für *gesicherte* Dateien ist auf "[Sicherungsversionen auswählen](#)" (im "[Gesicherte BS2000-Dateien und Jobvariablen restaurieren](#)") beschrieben; für *archivierte* Dateien verläuft sie entsprechend. Zu beachten ist allerdings, dass bei Angabe von SELECT-SAVE-VERSIONS=*LATEST nur die Dateien restauriert werden, die in der letzten Sicherungsversion tatsächlich gesichert wurden. Wenn von einer Datei wegen eines Fehlers nur der Katalogeintrag gesichert wurde (CNS, OPER), wird diese Datei nicht in einer früheren Sicherungsversion gesucht.

4.2.2.3 Aus einem Schattenarchiv restaurieren

Wenn einem Langzeit-Archiv ein Schattenarchiv zugewiesen wurde, kann das Schattenarchiv für das Restaurieren benutzt werden. Dies ist hilfreich, wenn das ursprüngliche Langzeit-Archiv zerstört oder die zugehörigen Datenträger nicht zugreifbar sind.

4.2.2.4 Elemente aus einer PLAM-Bibliothek restaurieren

Werden bei der Langzeitsicherung neben den Bibliotheksdateien auch die Meta-Informationen über die Elemente mitgesichert, können aus der Langzeitsicherung (wie bei Backup-Archiven) selektiv Bibliothekselemente mit der Anweisung RESTORE-LIBRARY-ELEMENTS restauriert werden. Bei einer Langzeitsicherung werden die Meta-Informationen nur mitgesichert, wenn für das Langzeit-Archiv die Sicherungsoption SAVE-PLAM-INFO=*YES eingestellt ist.

Wenn eine PLAM-Bibliothek mit den Meta-Informationen über die Bibliothekselemente gesichert werden soll, muss sich diese Bibliothek auf der S0-Ebene befinden. Bei einer Migration von PLAM-Bibliotheken werden diese Meta-Informationen nicht mitgesichert.

Die Bibliothek muss auf S0 zugreifbar sein, darf also nicht verdrängt sein. Die Elemente können wahlweise überschrieben und umbenannt werden.

Die Sicherungsversion der Bibliothek, in der die angegebenen Elemente vorhanden sind, muss bekannt sein.

Die Bibliothekselemente können auch in eine andere Zielbibliothek als die Ausgangsbibliothek restauriert werden.

Die Elementnamen einer mit PLAM-INFO=*YES gesicherten PLAM-Bibliothek sind nur in der PLAM-Information auf der Sicherungsdatei aber nicht im Archivverzeichnis enthalten. Die Elementnamen einer so gesicherten PLAM-Bibliothek können mit dem Operanden ELEMENTS=*LIST-ALL-TO-REPORT in der Anweisung RESTORE-LIBRARY-ELEMENTS aufgelistet werden.

4.2.2.5 Net-Storage-Dateien restaurieren

Werden bei der Langzeitsicherung auch Dateien auf Net-Storage gesichert, kann das Restaurieren auf Dateien auf Net-Storage beschränkt werden (Operand STORAGE-TYPE).

Net-Storage-Dateien vom Dateityp BS2000 oder Node-File, deren SAM-Struktur mitgesichert wurde, können auch auf andere Datenträgertypen restauriert werden. Dabei kann auch ihr Dateityp geändert werden. Im Gegensatz dazu können Net-Storage-Dateien vom Dateityp Node-File, deren SAM-Struktur nicht mitgesichert wurde, nur auf ein Net-Storage-Volume und nur mit dem ursprünglichen Dateityp Node-File restauriert werden.

4.2.3 Knotendateien des BS2000-UFS und ferner Knoten-S0 archivieren

Knotendateien des BS2000-UFS und ferner Knoten-S0 können mit der HSMS-Anweisung ARCHIVE-NODE-FILES archiviert werden. Sie können durch Angabe ihres Namens (mit Wildcard) und durch den Operanden SELECTION-BOUNDARY ausgewählt werden. Es werden entweder nur die angegebenen Dateien archiviert oder die angegebenen Dateien und – bei einem Dateiverzeichnis – alle untergeordneten Dateien und Dateiverzeichnisse bis zur Dateisystemgrenze oder bis zum Ende des Dateibaums (siehe [Abschnitt „Auswahl von Knotendateien des BS2000-UFS“](#)).

Den Umfang der Archivierung zeigt die Tabelle auf "[Umfang der Sicherung](#)".

Ein nicht-privilegierter Benutzer kann nur Knotendateien archivieren, die auf dem lokalen BS2000-UFS liegen und auf die er Zugriffsrecht hat. Dies sind alle Knotendateien, für die der nicht-privilegierte Benutzer Leseberechtigung besitzt und für deren übergeordnete Dateiverzeichnisse er die Ausführungsberechtigung hat.

Der HSMS-Verwalter kann alle Knotendateien archivieren, die auf dem lokalen BS2000-UFS oder auf einem fernen Dateisystem liegen.

4.2.3.1 System-Langzeitarchiv

Die archivierten Daten werden in Langzeitarchiven verwaltet, die der HSMS-Verwalter als öffentliches Archiv mit Schreibberechtigung eingerichtet hat (USER-ACCESS=*ALL-USERS und ACCESS=*WRITE).

System-Langzeitarchive für Knotendateien werden in der Regel über den symbolischen Namen SYSNODEARCHIVE angesprochen.

4.2.3.2 Logisches Freigabedatum

Bei einem Archivierungsauftrag kann für die archivierten Dateien mit dem Operanden FILE-EXPIRATION-DATE ein logisches Freigabedatum vereinbart werden. Darunter versteht man eine logische Schutzfrist, die von der physischen Schutzfrist des Datenträgers, der die Sicherungsversion enthält, abweichen kann. Die logische Schutzfrist kann länger sein als die physische Schutzfrist der Datenträger, wenn dies der HSMS-Verwalter für öffentliche Langzeitarchive so vereinbart hat. In diesem Fall muss der HSMS-Verwalter nach Ablauf der physischen Schutzfrist den Schutz der Datenträger vor Überschreiben durch administrative Maßnahmen Gewähr leisten. Dies kann z.B. durch Umwälzen der Datenträger vor Ablauf der Schutzfrist geschehen. Unter Umwälzen versteht man das Kopieren von Sicherungsversionen auf Datenträger mit höherem Verfallsdatum. Beim Umwälzen können gezielt nur die Sicherungsversionen übernommen werden, deren logisches Freigabedatum noch nicht erreicht ist.

4.2.3.3 Zusatzinformationen

Da die archivierten Daten teilweise recht lange in den Archiven aufbewahrt werden, kann es sinnvoll sein, Informationen zu diesen Daten zusammen mit der Sicherung zu hinterlegen. Bei der Archivierung können ein Kurzbeschreibungstext (Operand DESCRIPTOR) sowie ein längerer Kommentar (Operand USER-INFORMATION) eingegeben werden, die die Sicherungsversion beschreiben und Informationen zu den archivierten Dateien enthalten. Diese Beschreibungen können mit SHOW-ARCHIVE ausgegeben werden.

Eine nachträgliche Änderung der Zusatzinformationen ist möglich mit:

```
//MODIFY-ARCHIVE . . . ,SAVE-VERSIONS=*MODIFY-USER-RECORD(. . .)
```

Das Datum der ursprünglichen Sicherung bleibt unverändert, wenn eine Sicherungsdatei kopiert wird. Dieses ursprüngliche Sicherungsdatum kann deshalb zur Auswahl einer Sicherungsversion in den Anweisungen COPY-SAVE-FILE und RESTORE-SAVE-FILE benutzt werden.

4.2.3.4 Automatisches Duplizieren in ein Schattenarchiv

Die Sicherungsdatei eines Langzeitarchivs für Knotendateien des BS2000-UFS kann automatisch dupliziert werden, wenn dem betroffenen Langzeitarchiv ein Schattenarchiv zugewiesen wurde. Dabei muss der automatische Dupliziermechanismus in der HSMS-Anweisung ARCHIVE-NODE-FILES eingeschaltet sein. Dazu muss in der HSMS-Anweisung ARCHIVE-NODE-FILES beim Operanden OPERATION-CONTROL für SHADOW-COPY der Wert *ALLOWED angegeben sein (Default-Einstellung).

Die Sicherungsdatei wird am Ende des Archivierungslaufs in das zugeordnete Schattenarchiv kopiert. Die Kopie im Schattenarchiv erhält dieselbe Save-Version-ID und Save-File-ID wie das Original, d.h. sie bekommt denselben Zeitstempel.

Wenn eine Sicherungsdatei in einem Langzeitarchiv erweitert werden soll, versucht der automatische Dupliziermechanismus, die Sicherungsdatei mit derselben SFID im zugeordneten Schattenarchiv ebenfalls zu erweitern.

Wenn die Bandkopie sofort nach der Sicherung vom Schattenarchiv in einen besonders geschützten Raum ausgelagert werden soll, muss im Schattenarchiv stets eine neue Sicherungsdatei erzeugt werden, auch wenn die Sicherungsdatei im Hauptarchiv fortgesetzt wird. Dieser Modus wird über das Archiv-Attribut SHADOW-COPY=*ALLOWED-AND-NEW-SFID eingestellt.

4.2.4 Archivierte Knotendateien restaurieren

Mit der HSMS-Anweisung RESTORE-NODE-FILES können Sie archivierte Knotendateien in das BS2000-UFS oder ferne Knoten-S0 restaurieren.

Ein nicht-privilegierter Benutzer darf Daten aus dem lokalen BS2000-UFS restaurieren. Dabei gelten folgende Regeln:

- Aus einem Systemarchiv darf er nur seine eigenen Knotendateien restaurieren.
- Aus einem privaten Archiv darf er alle Daten restaurieren, die er archiviert hat.

Die Knotendateien werden mit denselben Attributen restauriert, die sie zum Sicherungszeitpunkt hatten (Eigentums- und Zugriffsrechte, Zeitstempel, ...).

4.2.4.1 Sicherungsversionen auswählen

Die Auswahl der Sicherungsversionen für *gesicherte* Dateien ist auf "[Sicherungsversionen auswählen](#)" (im "[Gesicherte BS2000-Dateien und Jobvariablen restaurieren](#)") beschrieben; für *archivierte* Dateien verläuft sie entsprechend. Zu beachten ist allerdings, dass bei Angabe von SELECT-SAVE-VERSIONS=*LATEST nur die Dateien restauriert werden, die in der letzten Sicherungsversion tatsächlich gesichert wurden. Wenn von einer Datei wegen eines Fehlers nur der Katalogeintrag gesichert wurde (CNS, OPER), wird diese Datei nicht in einer früheren Sicherungsversion gesucht.

4.2.4.2 Aus einem Schattenarchiv restaurieren

Wenn einem Langzeit-Archiv ein Schattenarchiv zugewiesen wurde, kann das Schattenarchiv für das Restaurieren benutzt werden. Dies ist hilfreich, wenn das ursprüngliche Langzeitarchiv zerstört oder die zugehörigen Datenträger nicht zugreifbar sind.

4.2.5 Beispiel zur Langzeitsicherung

Ein Langzeitarchiv wird eingerichtet:

```
//CREATE-ARCHIVE ARCHIVE-NAME=ARC.ARF,ALLOWED-USAGE=*ARCHIVAL,
      DIRECTORY-NAME=DIR.ARF,RETENTION-PERIOD=9
```

Einige Dateien werden archiviert und das Archivverzeichnis wird als letztes auf den Sicherungsdatenträger geschrieben:

```
//ARCHIVE-FILES FILE-NAMES=U.,SAVE-DIRECTORY=*YES,ARCHIVE-NAME=ARC.ARF,
      OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=REPORT.FULL)
```

Auszug aus der druckaufbereiteten Protokolldatei:

```
SAVE FILE IDENTIFIER - S.160516.155430      SAVE-VERSION-DATE=16-05-16  SAVE-VERSION-TIME=15:
54:32

% ARC0820 DIRECTORY ':A:$USERA.DIR.ARF' SAVED ON VSN 'TAPE01'

      *** CATALOG - A          USER - USERA          ***
      FILE/JOB VARIABLE NAME          LASTPG/ SAVE  INPUT DEV  SUB  OUTPUT
      VERS          SIZE  TYPE          VSN  TYP  SAVE  VSN(S)

DIR.ARF          1          6  FULL  PUBA00  D    0  TAPE01
U.1              1          1  FULL  PUBA00  D    0  TAPE01
U.2              1          1  FULL  PUBA00  D    0  TAPE01
U.3              1          1  FULL  PUBA00  D    0  TAPE01

      *** DIRECTORY NAME IS :A:$USERA.DIR.ARF

NUMBER OF FILES=          4 GLOBAL SIZE=          9 START= 2016-05-16 15:54:18 END= 2016-05-
16 15:55:30
```

Danach werden die Dateien gelöscht.

Restaurieren der Dateien aus der Langzeitsicherung

Werden jetzt das Langzeitarchiv und das Archivverzeichnis gelöscht (oder gehen verloren), muss vor dem Restaurieren der Dateien zunächst das Archivverzeichnis importiert werden:

```
//IMPORT-FILES FILE-NAMES=*DIRECTORY,SAVE-FILE=*BY-VOLUME(VOLUMES=TAPE01),
OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=REPORT.FULL)
```

Auszug aus der druckaufbereiteten Protokolldatei:

SAVE FILE IDENTIFIER - S.160516.155430									
SUBSAVE									
NUMBER	VSNS								
0	TAPE01								
		*** CATALOG - A	USER - USERA		***				
FILE/JOB	VARIABLE	NAME	LASTPG/	SAVE	VERSION	SAVE	INPUT	SUB	
OUTPUT			VERS	SIZE	IDENTIFIER	TYPE	VSN	SAVE	
DISK(S)									
DIR.ARF			1	6	160516.155432	FULL	TAPE01	0	
PUBA00									

Nun kann das Archiv mit dem Archivverzeichnis wieder eingerichtet werden:

```
//CREATE-ARCHIVE ARCHIVE-NAME=ARC.ARF,ALLOWED-USAGE=*ARCHIVAL,
DIRECTORY-NAME=DIR.ARF(
NEW-DIRECTORY=*NO(RECONSTRUCT-ARCHIVE=*YES))
```

Danach können die Dateien wieder restauriert werden:

```
//RESTORE-FILES FILE-NAMES=*ALL,ARCHIVE-NAME=ARC.ARF,
OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=REPORT.FULL)
```

4.3 Verdrängung (Migration) von BS2000-Dateien

Verdrängung ist nur für BS2000-Dateien auf gemeinschaftlichen Datenträgern möglich. Verdrängung bezeichnet die Auslagerung inaktiver Daten von der Verarbeitungsebene auf eine Hintergrundebene. Die verdrängten Daten werden in ein Migrationsarchiv aufgenommen; der von ihnen belegte Speicherplatz auf der Verarbeitungsebene wird freigegeben.

Die Dateien behalten aber auf der Verarbeitungsebene ihren Katalogeintrag und können über diesen Eintrag weiterhin angesprochen werden. Insbesondere bei Zugriffsversuchen des BS2000-Datenverwaltungssystems werden die Daten implizit, d.h. automatisch wieder auf die Verarbeitungsebene zurückgeholt.

Die Verdrängung steht allen Benutzern zur Verfügung, wenn das Archiv entsprechend eingerichtet wird. Der HSMS-Verwalter kann aber die Möglichkeit zur Verdrängung einschränken.

Die Verdrängung ist nur auf Pubsets möglich, die unter HSMS-Verwaltung genommen wurden.

Dateien, die auf einem dem Pubset zugeordneten Net-Storage-Volume abgelegt sind, können nicht verdrängt werden.

Dateien, die auf den Platten eines Plattenspeichersystems liegen und remote in einem anderen Plattenspeichersystem gespiegelt werden (z.B. Symmetrix-Platten gespiegelt mit SRDF), sollten nicht verdrängt werden, wenn der ferne Rechner keinen Zugriff auf die Verarbeitungsebene S2 hat.

Mit der HSMS-Anweisung MIGRATE-FILES können Sie auch Dateien migrieren, bei denen Sie Miteigentümer sind (siehe Handbuch „SECOS“ [16]).

Wenn eine MIGRATE-FILES-Anweisung auf einem Host verwendet wird, der SF- und SM-Pubsets verwaltet, dann können zu einem Zeitpunkt nur Objekte migriert werden, die entweder auf einem SF- oder auf einem SM-Pubset liegen. Darüber hinaus können beim Migrieren von Objekten auf SM-Pubsets nur Objekte eines SM-Pubsets pro Lauf bearbeitet werden.

Die Steuerung der Verdrängung und die Verwaltung der Migrationsarchive durch den HSMS-Verwalter ist im [Abschnitt „Steuerung der Verdrängung und Verwaltung der Migrationsarchive“](#) beschrieben.

Der Zugriff auf eine verdrängte Datei führt zu einem DVS-Fehler, wenn

- HSMS im System nicht verfügbar ist.
- das implizite Zurückholen verboten ist.
- das implizite Zurückholen fehlerhaft verlief.

4.3.1 Dateien verdrängen

Dateien werden mit der HSMS-Anweisung MIGRATE-FILES auf eine Hintergrundebene verdrängt.

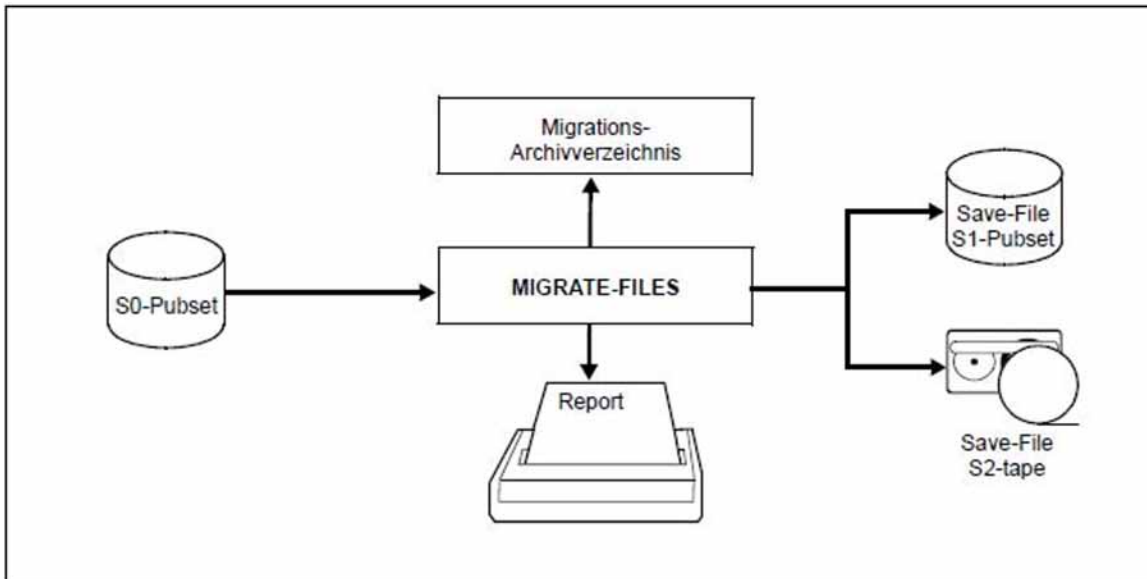


Bild 13: Verdrängen mit MIGRATE-FILES (nicht-privilegiert)

Die Verdrängung ist nur für Dateien möglich, die auf einem S0-Pubset katalogisiert sind, dem ein Migrationsarchiv zugeordnet ist.

Wenn die Verdrängung nur für Dateien erlaubt ist, die bereits vorher gesichert wurden (siehe Handbuch „HSMS Bd. 2“ [1], Operand BACKUP-MANDATORY in den HSMS-Anweisungen CREATE-SM-PUBSET-PARAMETERS und MODIFY-HSMS-PARAMETERS), sollte man sich vor der Verdrängung vergewissern, dass der aktuelle Stand der Datei schon in ein Backup-Archiv gesichert wurde.

Die Dateien können nur in die Standard-Sicherungsdatei eines Standard-Systemarchivs verdrängt werden. Deshalb ist die Angabe eines Archivs bei der Verdrängung weder sinnvoll noch möglich.

Der Benutzer kann über den Operanden TO-STORAGE bestimmen, ob die Dateien auf S1 oder auf S2 verdrängt werden sollen, falls der HSMS-Verwalter das Verdrängen auf S1 zugelassen hat.

Wenn für die verdrängten Dateien beim Löschen das Überschreiben mit binären Nullen vereinbart ist, wird auch bei der Migration der freigegebene Speicherplatz überschrieben.

Die zu verdrängenden Dateien können bei MIGRATE-FILES insbesondere nach dem Zeitraum ausgewählt werden, seit wann auf diese Dateien nicht mehr zugegriffen wurde (Operand UNUSED-DAYS) und nach ihrer Minimalgröße (Operand MINIMUM-SIZE) oder Maximalgröße (Operand MAXIMUM-SIZE).

Die zu verdrängenden Dateien können auch nach einer vorgebbaren Mindestanzahl von Plattenbereichen (extents) ausgewählt werden. Dazu dient der dem HSMS-Verwalter vorbehaltene Operand MINIMUM-EXTENTS.

Möglich ist selbstverständlich die Angabe einer Dateinamensliste und die Auswahl der Dateien im Dialog (siehe Abschnitt „Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien“).

4.3.1.1 Verdrängen bestimmter Mengen

HSMS-Verwalter und nicht-privilegierte Benutzer können mit einer Anweisung solange Dateien verdrängen, bis der dadurch frei gewordene Speicherplatz eine angebbare Zahl von PAM-Seiten erreicht hat:

```
//MIGRATE-FILES ... (RELEASE-PAGES=<zahl>)
```

HSMS verdrängt die Dateien nach absteigendem Inactive-Date, d.h. es verdrängt zuerst die Dateien, auf die am längsten nicht mehr zugegriffen wurde (LAST ACCESS DATE), und zwar solange, bis die angegebene Zahl freier Seiten erreicht ist. Dabei wird bei den Dateien die Zahl der belegten Seiten zu Grunde gelegt.

Auch in diesem Fall werden nur Dateien aus der Menge verdrängt, die durch FILE-NAMES und weitere Operanden ausgewählt wurde. Standardmäßig (RELEASE-PAGES=*MAXIMUM) werden alle Dateien dieser ausgewählten Menge verdrängt.

4.3.1.2 Schnelles Verdrängen ohne Bandverarbeitung (Schnellmigration)

HSMS-Verwalter und nicht-privilegierte Benutzer können mit einer Anweisung alle Dateien verdrängen, die sich schnell, d.h. ohne Bandverarbeitung, verdrängen lassen.

```
//MIGRATE-FILES ... (MIGRATION-INFO=*REMIGRATION)
```

HSMS verdrängt von der ausgewählten Dateimenge nur Dateien, die seit dem letzten Recall nicht verändert wurden und bei denen die Sicherungsdatei noch vorhanden ist. Diese Dateien erhalten sofort, d.h. ohne einen Bandzugriff, nur den Verweis auf die noch vorhandene Sicherungsdatei (S1 oder S2).

Bei dieser schnellen Verdrängung wird der belegte Speicherplatz sofort freigegeben und die Verdrängung erfordert keine zusätzliche Speicherkapazität auf S1 oder S2.

4.3.1.3 Migrationssperre

Der Eigentümer einer Datei kann über den Katalogeintrag selbst festlegen, ob diese Datei dem Verdrängungsmechanismus unterliegen soll oder nicht. Dies geschieht mit dem Operanden MIGRATE der Kommandos CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES:

- MIGRATE=*ALLOWED lässt die Verdrängung der Datei zu (Standard).
- MIGRATE=*INHIBITED verbietet sie (Ausnahme siehe "[Standardmäßig von der Migration ausgeschlossene Dateien](#)").

Der HSMS-Verwalter kann allerdings festlegen, dass auch Dateien mit MIGRATE= *INHIBITED verdrängt werden.

Für Benutzerkennungen mit dem Recht zur physikalischen Allokierung gibt es zusätzlich noch die Option MIGRATE=*FORBIDDEN. Dadurch wird die Migration absolut verboten. Dieses Kennzeichen wird im Katalogeintrag der Datei vermerkt und beim Kommando SHOW-FILE-ATTRIBUTES ausgegeben.

4.3.1.4 Standardmäßig von der Migration ausgeschlossene Dateien

Die folgenden Dateien können nicht auf eine Hintergrundebene verdrängt werden:

- Wenn der Operand BACKUP-MANDATORY (HSMS-Anweisung MODIFY-HSMS-PARAMETERS) den Wert *YES hat:
Dateien, die noch in keiner Sicherungsdatei enthalten sind (egal ob bei HSMS oder ARCHIVE), außer wenn die Datei die Sicherungsklasse „E“ hat.
- Dateien, die in die Ausnahmedatei aufgenommen wurden (siehe [Abschnitt „Ausnahmedatei“](#))
- geöffnete Dateien
- temporäre Dateien
- Dateien ohne Space-Allokierung
- Dateien unter der Benutzerkennung SYSHSMS
- Dateien auf Net-Storage
- Dateien auf privaten Datenträgern
- Dateigenerationsgruppen (FGG)
- Systemdateien wie der Katalog (aber nicht IPLR oder Startup-Datei)
- zu reparierende Dateien
- Dateien, die katalogisiert sind mit:
 - AVAILABILITY=*HIGH
 - MIGRATION=*INHIBITED und FILE-INHIBIT=*RESPECTED in der Anweisung MIGRATE-FILES oder in der letzten MODIFY-SM-PUBSET-PARAMETERS-Anweisung des betroffenen SM-Pubsets oder in der letzten MODIFY-HSMS-PARAMETERS-Anweisung.
 - MIGRATION=*FORBIDDEN

Wenn der Benutzer in einer HSMS-Anweisung Dateien angibt, die wegen der vorgenannten Gründe nicht bearbeitet werden, dann gibt das System für diese Dateien Warnungen aus. Wenn aber *OWN oder *ALL angegeben ist, werden keine Warnungen ausgegeben.

Beispiel:

```
% HSM0550 WARNING: SOME FILES NOT MIGRATED (REASON 'NET-STORAGE')
```

4.3.2 Verdrängen durch den HSMS-Verwalter

Verdrängung wird immer dann besonders akut, wenn es auf einem Pubset „eng“ wird, wenn also bestimmte Sättigungszustände drohen oder bereits erreicht sind. HSMS unterstützt dabei den HSMS-Verwalter durch folgende Funktionen:

- Umfangreiche Auskunftsmöglichkeiten informieren ihn über die Belegung der Pubsets (siehe den folgenden Abschnitt).
- Die Verdrängung von Dateien, bis eine bestimmte Menge an Speicherplatz frei geworden ist (siehe [Abschnitt „Verdrängen bestimmter Mengen“](#)), für den HSMS-Verwalter aber auch system- oder pubset-weise.
- Durch die Erstellung von Repeat-Jobs, die durch von HSMS versorgte Jobvariablen gesteuert werden, kann der HSMS-Verwalter automatisch auf das Erreichen von Sättigungszuständen reagieren (siehe den folgenden Abschnitt).

Sicherungsdateien lassen sich von S1 nach S2 verdrängen, also auf eine kostengünstigere Speicherebene.

4.3.2.1 Auskunftsmöglichkeiten

HSMS unterstützt den HSMS-Verwalter bei der Verdrängung durch verschiedene Auskunftsmöglichkeiten mit der HSMS-Anweisung SHOW-PUBSET-USAGE.

- Mit INFORMATION=*SUMMARY erhält er einen Überblick über die Gesamtkapazität der Pubsets unter HSMS-Verwaltung und den prozentualen Anteil der belegten, freien und verdrängten Seiten.
- Mit INFORMATION=*UNUSED-DAYS bekommt er pubset-weise einen Überblick über die inaktiven Dateien des Systems, also die Dateien, die sich zur Verdrängung anbieten, weil auf sie schon seit geraumer Zeit nicht mehr zugegriffen wurde.

Zu diesen Auskunftsmöglichkeiten siehe die HSMS-Anweisung SHOW-PUBSET-USAGE im Handbuch „HSMS Bd. 2“ [1].

4.3.2.2 Dateien automatisch verdrängen

HSMS überprüft in regelmäßigen Abständen, ob auf einem Pubset eine Sättigungsstufe erreicht wurde. Abhängig von dieser Überprüfung versorgt HSMS die Jobvariable \$SYSHSMS.SYS.HSM.MIGRATE.<pubset-id> pubset-spezifisch für alle in der Steuerdatei eingetragenen Pubsets.

Ähnliche Überprüfungen führt HSMS auch für alle Volume-Sets durch, die zu einem SM-Pubset unter HSMS-Kontrolle gehören. In diesem Fall liefert HSMS die Jobvariablen \$SYSHSMS.SYS.HSM.MIGRATE.<vol-id>, die auf dem entsprechenden SM-Pubset liegen.

Der HSMS-Verwalter kann durch Abfrage dieser Jobvariablen in ENTER-Jobs oder Programmen auf auftretende Sättigungszustände automatisch reagieren und so viele Dateien verdrängen lassen, dass die Sättigungsstufen auf den SF-Pubsets oder den Volume-Sets des SM-Pubsets nicht mehr bestehen. Die Anzahl der dazu notwendigen PAM-Seiten ist abhängig von den eingestellten Sättigungsstufen und wird in der Jobvariable ebenfalls übergeben. Die zu verdrängende Menge lässt sich dann bei der MIGRATE-FILES-Anweisung mit dem Operanden RELEASE-PAGES angeben.

HSMS legt die Jobvariablen beim Start automatisch an, falls sie nicht schon eingerichtet sind. Die Jobvariablen werden nicht mehrbenutzbar auf der Kennung SYSHSMS angelegt (auf dem Home-Pubset für ein SF-Pubset, auf dem SM-Pubset selbst für die eingeschlossenen Volume-Sets). Die Attribute der Jobvariablen dürfen nicht verändert werden.

Wenn HSMS nicht oder nur im Definitions-/Auskunfts-Betrieb geladen ist, enthalten die Jobvariablen auf den ersten 40 Byte x'FF...'. In den anderen Betriebsarten werden die Jobvariablen nach dem Start für SF-Pubsets oder beim Importieren eines jeden SM-Pubsets und dann alle 10 Minuten auf den aktuellen Stand gebracht. Die jeweils auf einem Pubset erreichte Sättigungsstufe wird in die Jobvariable des Pubsets eingetragen.

Die Inhalte der Jobvariablen haben folgende Bedeutung:

Position	Länge	Bedeutung
1	1	Zustand des Pubsets: A: zugreifbar N: Katalogkennung nicht verfügbar I: nicht zugreifbar S: Shared-Slave R: entfernt
2	1	Füllzeichen
3	1	Sättigungsstufe des Pubsets
4	2	Füllzeichen
6	10	Anzahl der belegten PAM-Seiten
16	5	Füllzeichen
21	10	Gesamtkapazität des Pubsets in PAM-Seiten
31	5	Füllzeichen
36	10	Anzahl der PAM-Seiten, um die die angezeigte Sättigungsstufe überschritten wird, falls eine Sättigungsstufe erreicht ist

46	3	Füllzeichen
----	---	-------------

49	12	Datum, an dem die Jobvariable auf den aktuellen Wert gesetzt wurde
61	5	Füllzeichen
66	8	Uhrzeit, an dem die Jobvariable auf den aktuellen Wert gesetzt wurde
74	2	Füllzeichen
76	181	Füllzeichen

Hinweis

Bei SM-Pubsets wird dieser Mechanismus nur auf Volume-Set-Ebene angeboten und nicht für den gesamten SM-Pubset.

Mit diesen Angaben lässt sich die Migration gezielt steuern.

Beispiele für einen ENTER-Job und ein Programm befinden sich am Ende des Abschnitts ab ["Beispiele zur Verdrängung"](#).

4.3.3 Verdrängte Dateien

Nach der Verdrängung besitzt eine Datei auf der Verarbeitungsebene nur noch einen Katalogeintrag; sie beansprucht aber dort keinen Speicherplatz mehr. Im Katalogeintrag wird vermerkt, wo die Daten dieser Datei stehen, d.h. auf welche Hintergrundebene (S1 oder S2) die Daten dieser Datei verdrängt wurden.

Als SUPPORT wird, je nach Hintergrundebene, PUB/S1 oder PUB/S2 ausgegeben. Bei EXTENTS ist NONE eingetragen.

Die Angaben für die Dateigröße und den Last-Page-Pointer ändern sich durch die Verdrängung nicht. Die Speicherebene, auf die die Datei verdrängt wurde, wird bei STOR-LEVEL ausgegeben.

Eine Übersicht über verdrängte Dateien lässt sich mit folgendem Kommando gewinnen:

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=*ALL, -
/ SELECT=*BY-ATTRIBUTES( STORAGE-LEVEL=(S1,S2))
```

Beispiel für Ausgaben von verdrängten Dateien

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=TEST.HSMS, INFORMATION=*ALL-ATTRIBUTES
%0000000543#:10SN:$MANUALE.TEST.HSMS
% ----- HISTORY -----
% CRE-DATE   = 2016-09-20  ACC-DATE   = 2016-09-29  CHANG-DATE = 2016-09-29
% CRE-TIME   = 15:15:13   ACC-TIME   = 10:34:05   CHANG-TIME = 10:34:05
% ACC-COUNT  = 4          S-ALLO-NUM = 0
% ----- SECURITY -----
% READ-PASS  = NONE       WRITE-PASS  = NONE       EXEC-PASS  = NONE
% USER-ACC   = OWNER-ONLY ACCESS        = WRITE        ACL         = NO
% AUDIT      = NONE       FREE-DEL-D  = *NONE       EXPIR-DATE = 2016-09-20
% DESTROY    = NO         FREE-DEL-T  = *NONE       EXPIR-TIME = 00:00:00
% SP-REL-LOCK= NO         ENCRYPTION = *NONE
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG  = COMPL-FILE  VERSION    = 2
% MIGRATE    = ALLOWED   STOR-LEVEL = S2
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM      BUF-LEN    = STD(1)     BLK-CONTR  = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)    = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)     REC-SIZE   = 512
% KEY-LEN    = 10        KEY-POS    = 5
% AVAIL      = *STD
% WORK-FILE  = *NO       F-PREFORM  = *K          S0-MIGR    = *ALLOWED
% ----- ALLOCATION -----
% SUPPORT    = PUB/S2    S-ALLOC    = 543        HIGH-US-PA = 521
% EXTENTS    VOLUME     DEVICE-TYPE
% NONE       NONE       NONE
%:2OSG: PUB/S2:      1 FILE RES=      543 FRE=      0 REL=      22 PAGES

/SHOW-FILE-ATTRIBUTES FILE-NAME=TEST.
% 543#:10SN:$MANUALE.TEST.HSMS
% 3 :10SN:$MANUALE.TEST.MSG
% 3#:10SN:$MANUALE.TEST.VOLX
%:1OSG: PUBLIC:      1 FILE RES=      3 FREE=      2 REL=      0 PAGES
%:1OSG: PUB/S2:      2 FILES RES=     546 FREE=      2 REL=      0 PAGES
```

Auch bei Übersichten sind verdrängte Dateien an dem Nummernzeichen (#) zwischen der Dateigrößenangabe und der Katalogkennung zu erkennen.

Solange eine Datei verdrängt ist, gilt für Zugriffe auf diese Datei und ihren Katalogeintrag:

- Veränderungen der Dateiattribute, wie z.B. der Schutzattribute, Kennwörter etc. sind möglich und bleiben auch nach dem Zurückholen wirksam.
- Die Datei darf allerdings nicht umbenannt werden; ein entsprechender Versuch wird mit einer Fehlermeldung zurückgewiesen.
- Die Veränderung der Dateigröße (Operand SPACE beim Kommando MODIFY-FILE-ATTRIBUTES) wird angenommen. Die Änderung wird allerdings erst nach dem Zurückholen wirksam.
- Die Datei kann jederzeit gelöscht werden. Eine verdrängte Datei, deren Katalogeintrag gelöscht wurde, ist für HSMS ungültig. Schließlich kann sie ohne Katalogeintrag nicht mehr zurückgeholt werden.
- Das Überschreiben der Datei (OPEN OUTPUT, OUTIN oder UPDATE) auf der Verarbeitungsebene ist zugelassen. Danach gilt die Datei nicht mehr als verdrängt und ist im Rahmen des Migrationsarchivs ebenfalls ungültig.

4.3.4 Verdrängung und große Dateien (> 32 GB)

BS2000 unterstützt auch Dateien ≥ 32 GB, aber nur auf Pubsets mit den Attributen `LARGE-OBJECTS=*YES` und `LARGE-FILES-ALLOWED=*YES`.

Mit HSMS können auch diese Dateien verdrängt und zurückgeholt werden.

Wenn große Dateien Pubsets mit den vorgenannten Attributen nicht verlassen, gibt es keine Einschränkungen gegenüber Dateien < 32 GB.

4.3.5 Verdrängung und Concurrent Copy

HSMS kann Dateien während einer Sitzung mit Concurrent Copy (CCOPY) nicht verdrängen. Daher empfehlen wir, die Migrationssperre aller Dateien, die für eine CCOPY-Sitzung vorgesehen sind, zu aktivieren. Der Systemverwalter sollte dazu folgende HSMS-Anweisung eingeben:

```
//MODIFY-HSMS-PARAMETERS MIGRATION-CONTROL=*PAR(EXCEPT-FILE=...)
```

Beim Operanden EXCEPT-FILE kann eine Liste der Dateien angegeben werden, die nicht verdrängt werden sollen.

4.3.6 Verdrängte Dateien zurückholen

Der zur Verdrängung umgekehrte Vorgang, das Kopieren der verdrängten Daten auf die Verarbeitungsebene, heißt Zurückholen. Es können nur so viele Dateien auf S0 eingespielt werden, wie auf den jeweiligen Benutzerkennungen Platz haben.

Es können nur Dateien zurückgeholt werden, die als verdrängt katalogisiert sind, und nur der Stand im Migrationsarchiv, der zum Katalogeintrag passt. Der aktuelle Stand einer Datei auf der Verarbeitungsebene kann also durch Zurückholen nicht verändert werden.

Der nicht-privilegierte Benutzer kann verdrängte Dateien einer fremden Benutzerkennung zurückholen, wenn Miteigentümerschaft besteht oder die Dateischutzattribute den Zugriff von einer anderen Benutzerkennung erlauben.

Achtung

Beim Zurückholen einer Datei ändern sich folgende Dateiattribute: interner Dateiname (CFID), DEVICE-TYPE, SUPPORT, VOLUME und EXTENTS. Anwendungen, die diese Dateiattribute auswerten, müssen das berücksichtigen.

Für das Zurückholen bietet HSMS zwei unterschiedliche Möglichkeiten, die in folgenden Abschnitten beschrieben sind.

4.3.6.1 Explizites Zurückholen durch HSMS-Aufruf

HSMS wird aufgerufen und die Datei ausdrücklich mit der HSMS-Anweisung RECALL-MIGRATED-FILES aus dem Migrationsarchiv zurückgeholt.

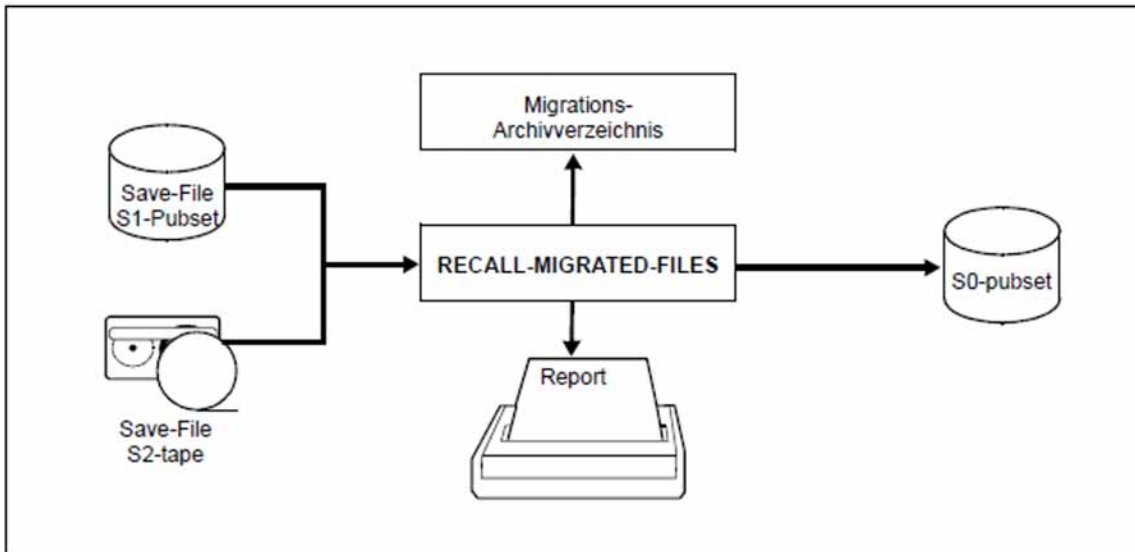


Bild 14: Explizites Zurückholen mit RECALL-MIGRATED-FILES

Der HSMS-Verwalter kann dabei mit dem Operanden FROM-STORAGE angeben, von welcher Speicherebene er die Dateien zurückholen will. Wenn er dazu keine Angabe macht, werden die Dateien aus der letzten Sicherungsdatei im Migrationsarchiv geholt, in der die Datei enthalten ist, egal auf welcher Speicherebene sie steht (auch wenn es eine Kopie ist).

Für nicht-privilegierte Benutzer ist letzteres Vorgehen Standard.

4.3.6.2 Implizites Zurückholen bei Dateizugriff

Der Benutzer kann auch ohne ausdrücklichen HSMS-Aufruf eine verdrängte Datei zurückholen. Wenn der Benutzer bzw. ein Benutzerprogramm auf eine verdrängte Datei zugreifen will, stellt das Datenverwaltungssystem (DVS) des BS2000 durch den Katalogeintrag fest, dass diese Datei nicht auf der Verarbeitungsebene verfügbar ist, sondern erst zurückgeholt werden muss. Beim Anfordern einer Datei mittels /SECURE-RESOURCE-ALLOCATION mit Wartezeit und bei Zugriffen wie OPEN INPUT, INOUT oder UPDATE holt das DVS die Datei dann automatisch (implizit) zurück.

Beim impliziten Zurückholen wird die Datei aus der letzten Sicherungsdatei (auch Kopie) auf der Speicherebene geholt, die im Katalog eingetragen ist.

Die Bearbeitung beim impliziten Zurückholen ist davon abhängig, ob der Zugriff auf die Datei durch das Kommando SECURE-RESOURCE-ALLOCATION oder durch ein OPEN erfolgte. Die Anforderung mit dem Kommando

```
/SECURE-RESOURCE-ALLOCATION FILE=<dateiname>,WAIT=...
```

mit Wartezeit vor Beginn der Verarbeitung hat folgende Vorteile:

- Alle für einen Job benötigten Dateien (bis zu 48) können mit einem Kommando angefordert und mit einem Auftrag zurückgeholt werden. Dadurch sind weniger HSMS-Aufrufe und evtl. weniger Bandanforderungen erforderlich.
- Es existiert ein definierter Wartepunkt, nachdem die Verarbeitung entweder starten kann oder abgebrochen werden muss. „Unliebsame Überraschungen“ während der Verarbeitung durch fehlende Dateien lassen sich so vermeiden.
- Für dermaßen angestoßene Zurückholaufträge gibt das HSMS einen zusammenfassenden Report aus, der sonst bei implizitem Zurückholen nur im Fehlerfall ausgegeben wird.

Die Wartezeit mit dem WAIT-Operanden sollte so lang gewählt werden, dass auch Zugriffe auf S2 in dieser Zeit möglich sind. Ohne Angabe einer Wartezeit meldet das DVS einen Fehler bei der Reservierung der Datei.

Wenn der Rückholauftrag nach einem OPEN gestartet wird, sind folgende Fälle zu unterscheiden:

- Zurückholen im Dialog von der Speicherebene S1:
Das Zurückholen wird immer ausgeführt; es wird keine Meldung ausgegeben.
- Zurückholen im Dialog von der Speicherebene S2:
HSMS gibt zuerst eine Meldung aus, dass ein Zugriff auf die Speicherebene S2 erforderlich ist und fragt den Benutzer, ob das Zurückholen gestartet werden soll.
Wenn der Benutzer einverstanden ist oder nicht innerhalb der voreingestellten Wartezeit antwortet, gibt HSMS eine zweite Meldung aus. Anschließend startet HSMS das Zurückholen, das dann nicht mehr unterbrochen werden kann.
Wenn der Benutzer mit dem Zurückholen nicht einverstanden ist, wird das Zurückholen nicht gestartet und der Zugriff auf die Datei wird abgewiesen.

Der HSMS-Verwalter kann das Verhalten von HSMS beeinflussen. Er kann den Unterbrechungsmechanismus mit dem Operanden CANCEL-AT-RECALL der HSMS-Anweisung MODIFY-HSMS-PARAMETERS ausschalten: HSMS gibt dann keine Meldung aus und das Zurückholen wird immer gestartet. Außerdem kann der HSMS-Verwalter die maximale Wartezeit mit dem Operanden MAXIMUM-WAIT-TIME zwischen 10 Sekunden und 60 Minuten einstellen. Der Unterbrechungsmechanismus sollte aber so eingestellt werden, dass er bereits nach einer kurzen Wartezeit ausgelöst wird.

- Zurückholen im Stapelbetrieb:
Das Zurückholen wird immer durchgeführt.

- Beim Zurückholen wird die Datei zuerst unter den Speicherplatz der Benutzerkennung SYSHSMS importiert, bevor die Zuweisung auf die endgültige Benutzerkennung geändert wird. Deshalb muss für die Benutzerkennung SYSHSMS genügend Speicherplatz vorgesehen werden – auch für sehr große Dateien. Beim Einrichten der Benutzerkennung SYSHSMS sollte aus diesem Grund die Speicherplatzerweiterung (PUBLIC-SPACE-EXCESS) zugelassen werden.

Wenn im Dialog gearbeitet wird, werden Dateien nur während der festgelegten Bandverarbeitungszeiten automatisch von der Speicherebene S2 zurückgeholt; ansonsten wird beim Zugriff auf eine verdrängte Datei eine Fehlermeldung ausgegeben.

Backup-Server nutzen

Ab HSMS V11.0 kann für das implizite Zurückholen von Dateien, die von Shared-Pubsets verdrängt wurden, der vereinbarte Backup-Server genutzt werden (siehe [Abschnitt „Backup-Server“](#)). Dazu muss für das Archiv-Attribut BACKUP-SERVER-USAGE des zugewiesenen Standard-Systemarchivs SYSMIGRATE in der entsprechenden Pubset-Umgebung ein entsprechender Wert vereinbart werden.

In einer SF-Pubset-Umgebung ist SYSMIGRATE ein Pubset-spezifisches, oder falls nicht definiert, ein global definiertes Migrations-Archiv. In einer SM-Pubset-Umgebung ist SYSMIGRATE immer ein Pubset-spezifisches Migrations-Archiv.

Mit der Einstellung BACKUP-SERVER-USAGE=*STD wird, falls definiert, der im lokalen System vereinbarte Backup-Server genutzt.

Mit der Einstellung BACKUP-SERVER-USAGE=*NO wird die Backup-Server-Funktionalität nicht genutzt, selbst wenn im lokalen System ein Backup-Server vereinbart ist. Die Verarbeitung erfolgt im Master- bzw. Slave-Modus.

Wenn kein Standard-Systemarchiv SYSMIGRATE definiert ist, erfolgt das Zurückholen ebenfalls im Master- bzw. Slave-Modus.

4.3.7 RESTORE von verdrängten Dateien

Bei verdrängten Dateien kann HSMS alle benötigten Sicherungsdaten nur dann geschlossen verwalten, wenn alle Daten der verdrängten Dateien mitgesichert werden. Nur in diesem Fall lassen sich bei Ausfallsituationen die Daten einfach wiederherstellen.

Beim Restaurieren behandelt HSMS verdrängte Dateien folgendermaßen:

- **Versionshaltung**

Beim RESTORE (Restore, Aktivierung, Import) durch den nicht-privilegierten Benutzer werden verdrängte Dateien immer auf die Speicherebene S0 gebracht.

- **Behebung eines S0-Crashes**

Der HSMS-Verwalter kann für verdrängte Dateien wahlweise nur den Katalogeintrag restaurieren. Dazu steht ihm der Operand MIGRATED-FILES=*CATALOG-ONLY in der RESTORE-FILES-Anweisung zur Verfügung. Dabei werden die Katalogeinträge so geändert, dass sie auf die jüngsten passenden Daten im Migrationsarchiv zeigen (impliziter EXCHANGE).

Wenn im Migrationsarchiv für eine Datei keine passenden Daten vorhanden sind, wird dies als Fehler gemeldet. Der Katalogeintrag bleibt dabei bestehen. Eine solche Fehlermeldung tritt für Dateien auf, die nach der letzten Sicherung zurückgeholt oder gelöscht wurden, wobei danach der Migrationsbereich reorganisiert wurde. Diese übrig bleibenden Inkonsistenzen können behoben werden durch:

- Restore der betroffenen Dateien nach S0 oder auf eine gewünschte Migrationsebene. Dazu steht die HSMS-Anweisung REPAIR-CATALOG-BY-RESTORE zur Verfügung.
- Löschen des Katalogeintrags

- **Behebung eines S1-Crashes**

Dazu können unterschiedliche Verfahren angewandt werden:

- Anweisung REPAIR-CATALOG-BY-EXCHANGE, falls Kopien der Daten auf der Speicherebene S2 in einem anderen Migrationsarchiv vorhanden sind.
- Anweisung REPAIR-CATALOG-BY-RESTORE, falls im Backup-Archiv Sicherungen der nach S1 migrierten Daten vorliegen.

Beide Verfahren können auch kombiniert werden. Zuerst werden für die Dateien, für die noch Daten im Migrationsarchiv vorhanden sind, mit REPAIR-CATALOG-BY-EXCHANGE die Lokalisierungsinformationen auf die entsprechenden Sicherungsdateien gerichtet. Anschließend werden die Dateien, für die keine Daten mehr im Migrationsarchiv vorhanden sind, die aber noch Daten im Backup-Archiv besitzen, mit REPAIR-CATALOG-BY-RESTORE aus dem Backup-Archiv restauriert. Beide Anweisungen wirken nur für migrierte Dateien, die in der ungültig gewordenen und der zur Reparatur verwendeten Sicherungsdatei denselben Dateistand (CFID) haben.

- **Behebung eines S2-Crashes**

Zur Behebung eines S2-Crashes bzw. bei Zerstörung einer Sicherungsdatei im Migrationsarchiv können unterschiedliche Verfahren angewandt werden:

- Anweisung REPLACE-SAVE-FILE-BY-EXCHANGE, falls im Migrationsarchiv eine Kopie der Sicherungsdatei vorliegt.
- Anweisung REPLACE-SAVE-FILE-BY-RESTORE, falls im Backup-Archiv Sicherungen der Daten der Sicherungsdatei vorliegen.
- Anweisung COPY-SAVE-FILE aus einem Langzeitarchiv in ein Migrationsarchiv mit anschließendem REPAIR-CATALOG-BY-EXCHANGE bzw. REPLACE-SAVE-FILE-BY-EXCHANGE.

- **Behebung eines Crashes auf S0 und den Migrationsebenen**

Dazu kann zuerst mit dem Operanden `MIGRATED-FILES=*CATALOG-ONLY` der `RESTORE-FILES`-Anweisung die Speicherebene S0 wiederhergestellt werden. Anschließend können dann mit den HSMS-Anweisungen `REPAIR-CATALOG-BY-EXCHANGE` und `REPAIR-CATALOG-BY-RESTORE` die Daten auf der Migrationsebene aktiviert bzw. restauriert werden.

i Zur Behebung von S1 bzw. S2-Crashes ist es grundsätzlich nötig, Kopien der migrierten Daten vorzuhalten. Zum Zeitpunkt der Migration wird dies gewährleistet, indem mit dem globalen bzw. SM-Pubset-spezifischen HSMS-Parameter `BACKUP-MANDATORY=*YES` nur Dateien migriert werden können, die zuvor gesichert worden sind. Die hier erstellten Sicherungen werden jedoch üblicherweise nach Ablauf der Retention-Period der Sicherung freigegeben (`//MODIFY-ARCHIVE ... SAVE-FILE=*DELETE(...)`) und stehen dann nicht mehr als Sicherheitskopie zur Verfügung. Es muss daher unbedingt darauf geachtet werden, dass von Zeit zu Zeit auch die migrierten Daten gesichert werden: dies geschieht mit `//BACKUP-FILES ... SAVE-OPTIONS=*PARAMETERS(SAVE-DATA=*S2-S1-S0)`

4.3.8 Verdrängung nach S0

Die Verdrängung nach S0 bringt eine Datei von einem Volume-Set eines SM-Pubsets auf einen anderen Volume-Set desselben SM-Pubsets. Dadurch kann man

- einen SM-Pubset reorganisieren.
- die Dateien auf einem SM-Pubset neu verteilen, wenn ein neuer, leerer Volume-Set hinzugefügt wurde.
- alle Dateien eines Volume-Sets, der aus einem SM-Pubset entfernt werden soll, auf einen anderen Volume-Set verlegen.

Die Verdrängung nach S0 wird in HSMS nicht durch eine neue Anweisung realisiert. Die bereits vorhandenen HSMS-Funktionen ermöglichen es dem HSMS-Verwalter, die Verdrängung nach S0 in drei aufeinander folgenden Schritten, die in einer SDF(-P)-Prozedur zusammengefasst werden können, durchzuführen:

1. Auswählen der Dateien, die verdrängt werden sollen, durch die HSMS-Anweisung SELECT-FILE-NAMES.

Nachfolgend zwei Beispiele:

- Zum Reorganisieren eines Pubsets werden alle Dateien ausgewählt, die in Bezug auf ihre Attribute nicht auf dem besten Volume-Set liegen:

```
//SELECT-FILE-NAMES FILE-NAMES=:<sm-pubset-id>:$*.*, -
SELECT-FROM=*CATALOG(STORAGE-LEVEL=*S0, -
SUPPORT=*SYSTEM-MANAGED-PUBSET -
(ALLOCATION-QUALITY=*NOT-BEST-VOLUME-SET)), -
OUTPUT=<files-to-move.list>
```

- Zum Entfernen eines Volume-Sets aus einem SM-Pubset werden alle Dateien ausgewählt, die auf diesem Volume-Set liegen:

```
//SELECT-FILE-NAMES FILE-NAMES=:<sm-pubset-id>:$*.*, -
SELECT-FROM=*CATALOG(STORAGE-LEVEL=*S0, -
SUPPORT=*SYSTEM-MANAGED-PUBSET(VOLUME-SET-ID=<volume-set-id>)), -
OUTPUT=<files-to-move.list>
```

2. Verdrängen der Dateien auf die S2-Ebene:

```
//MIGRATE-FILES ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=<sm-pubset-id>) , -
FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=*FROM-FILE( -
files-to-move.list>),TO-STORAGE=*S2-STORAGE-LEVEL)
```

3. Zurückholen der Dateien auf die S0-Ebene mit der HSMS-Anweisung RECALL-MIGRATED-FILES:

- Wenn die Dateien in Bezug auf ihre Attribute auf das beste Volume-Set zurückgeholt werden sollen, muss folgende Anweisung verwendet werden:

```
//RECALL-MIGRATED-FILES FILE-NAMES=*FROM-FILE(files.to.move.list) , -  
  ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=<sm-pubset-id>)
```

- Wenn die Dateien auf einem bestimmten Volume-Set des SM-Pubsets abgelegt werden sollen, muss folgende Anweisung verwendet werden:

```
//RECALL-MIGRATED-FILES FILE-NAMES=*FROM-FILE(files.to.move.list) , -  
  ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=<sm-pubset-id> , -  
  NEW-DATA-SUPPORT=<volume-set-id>)
```

4.3.9 Alle Dateien eines Benutzers verlagern

Der HSMS-Verwalter kann alle Dateien eines Benutzers auf einen anderen Pubset oder eine andere Benutzerkennung verlagern – unabhängig davon, ob die Dateien migriert sind oder nicht. Wenn Dateien auf eine andere Benutzerkennung verlagert werden sollen, muss diese Benutzerkennung auf der Ziel-Katalogkennung bereits vor dem Verlagern vorhanden sein.

Zum Verlagern von Dateien muss der HSMS-Verwalter die folgenden HSMS-Anweisungen in der angegebenen Reihenfolge ausführen:

1. Für alle betroffenen Sicherungsdateien, die Daten des zu verlagernden Benutzers enthalten, muss eine COPY-SAVE-FILE-Anweisung ausgeführt werden:

```
//COPY-SAVE-FILE SAVE-FILE-ID=<save file>,SELECT-FILES=$USER., -
NEW-FILE-NAMES=*BY-RULE(NEW-CATALOG-ID=<new cat-id>, -
NEW-USER-ID=<new user name>), -
MIGRATION-STATE=*MIGRATED-FILE, -
ARCHIVE-NAME=<name of the *SYSMIGRATE archive of the old cat-id>, -
TO-ARCHIVE-NAME=<name of the *SYSMIGRATE archive of the new cat-id>,
...
```

Diese HSMS-Anweisung erzeugt eine neue Sicherungsdatei, auf die die migrierten Dateien des neuen Benutzers verweisen.

2.


```
//BACKUP-FILES FILE-NAMES=$USER., -
NEW-FILE-NAMES=*BY-RULE(NEW-CATALOG-ID=<new cat-id>, -
NEW-USER-ID=<new user name>), -
SELECT-FILES=*ALL-FILES, -
SAVE-OPTIONS=*PARAMETERS(SAVE-DATA=*S0), -
ARCHIVE-NAME=<archive name>, ...
```

Das Archiv, das in dieser HSMS-Anweisung angegeben wird, kann das Standard-Sicherungsarchiv *SYSBACKUP der neuen Katalogkennung sein.

3.


```
//RESTORE-FILES FILE-NAMES=$<new user name>., -
MIGRATED-FILES=*CATALOG-ONLY, -
ARCHIVE-NAME=<archive name>, ...
```

Nach dem Restore zeigen alle Katalogeinträge auf die Sicherungsdatei, die durch die COPY-SAVE-FILE-Anweisung erzeugt wurde (siehe 1.).

4.3.10 Beispiele zur Verdrängung

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Verdrängung mit Komprimierung
- Verdrängung nach Selektionskriterien
- Verdrängung nach Menge
- Behebung von Speicherengpässen auf S0
- Pubset-weise Verdrängung mit Ausgabe der Pubset-Belegung
- Automatisches Starten von Migrationsaufträgen durch ENTER-Jobs
- Automatisches Starten von Migrationsaufträgen durch ein Programm

Die Beispiele beziehen sich auf die Beispielkonfiguration, die ab "[Einrichten einer HSMS-Konfiguration \(Beispiel\)](#)" beschrieben ist.

Verdrängung mit Komprimierung

Dateien werden auf die Speicherebene S1 verdrängt und dabei komprimiert. Die Sicherungsdateien werden mit den verdrängten Dateien verglichen.


```
//MIGRATE-FILES - _____ (1)
// FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=$MANUAL.FILE.0*, -
// COMPRESS-FILES=*YES, -
// TO-STORAGE=*S1-STORAGE-LEVEL), -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=HSMS.MAN.R.MGF.1, -
// WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
Report HSMS.MAN.R.MGF.1 (Ausschnitt): _____ (2)

*** MIGRATE - FILES                HSMS V11.0          FULL          REPORT    *** 2016-08-12
14:48:31    PAGE    2
      REQUEST-NAME=MGF#0AAK REQUEST-DATE=2016-08-12 14:48:09 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED WITHOUT ERROR

% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.144811', VERSION '11.0'
% ARC0033 ARCHIVE SUBTASK TSN 0AAO GENERATED
      SAVE FILE IDENTIFIER - S.160812.144813

      SUBSAVE
      NUMBER      VSNS
      0           0:2BC

      SAVE FILE IDENTIFIER - S.160812.144813
      *** CATALOG - 2BY          USER - MANUAL          ***
**  OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-08-12 SAVE-VERSION-TIME=14:48:13 *
      FILE/JOB VARIABLE NAME                LASTPG/  SAVE  INPUT DEV  SUB  OUTPUT
      VERS      SIZE  TYPE   VSN  TYP  SAVE  VSN(S)
FILE.01          1      3  FULL  2BY.01  D    0  0:2BC
FILE.02          1      8  FULL  2BY.01  D    0  0:2BC
FILE.03          1     13  FULL  2BY.01  D    0  0:2BC
FILE.04          1     18  FULL  2BY.01  D    0  0:2BC
FILE.05          1     16  FULL  2BY.00  D    0  0:2BC
FILE.06          1     11  FULL  2BY.00  D    0  0:2BC
FILE.07          1      6  FULL  2BY.00  D    0  0:2BC
***          E N D    O F                HSMS V11.0          FULL          REPORT    *** 2016-08-12
12:19:05          ***

/SHOW-FILE-ATT FILE-NAME=$MANUAL.FILE.0*, - _____ (3)
/ INFORMATION=SPACE-SUMMARY
%:2BY: PUB/S1:      7  FILES RES=      90  FRE=      15  REL=      9  PAGES
/SHOW-FILE-ATT FILE=:2BC:$*.ARCHIVE.SAVE.FILE.160812.144813., -
/ INFORMATION=SPACE-SUMMARY
%:2BC: PUBLIC:      1  FILE RES=      12  FRE=      2  REL=      0  PAGES
```

- (1) Die Dateien der Benutzererkennung MANUAL mit dem Namen FILE.0.. werden in das System-Migrationsarchiv verdrängt.
- (2) Ausschnitt aus dem von HSMS erzeugten Report des Migrationslaufs mit Ausgabe der SFID.
- (3) Ein Vergleich der verdrängten Dateien mit den Sicherungsdateien, in die sie geschrieben wurden, zeigt den geringeren Platzbedarf der komprimierten Sicherungsdatei. Da ja die verdrängten Dateien den Speicherplatz nicht mehr belegen, wurde die Differenz eingespart.

Verdrängung nach Selektionskriterien

Die Dateien werden über das Selektionskriterium Schutzfrist ausgewählt und verdrängt.

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=$MANUAL.FILE.1*, - _____ (1)
/ SELECT=*BY-ATTRIBUTES( STORAGE-LEVEL=*S0, -
/ EXPIRATION-DATE=*INTERVAL(TO=+365)), -
/ OUTPUT=#SELECT-LIST(FORM-NAME=*FILE-NAME)
%:2BY: PUBLIC: 7 FILES RES= 90 FREE= 15 REL= 9 PAGES
/START-HSMS
//MIGRATE-FILES - _____ (2)
// FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=*FROM-FILE -
// (#SELECT-LIST),COMPRESS-FILES=*YES, -
// TO-STORAGE=*S1-STORAGE-LEVEL), -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
// OUTPUT=HSMS.MAN.R.MGF.2, -
// WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Mit dem Kommando SHOW-FILE-ATTRIBUTES wird eine Liste aller Dateien der Benutzerkennung MANUAL mit Namen FILE.1., deren Schutzfrist innerhalb der nächsten 365 Tage ausläuft, in eine temporäre Datei geschrieben. Damit werden die Dateien ausgewählt, deren Schutzfrist innerhalb der Schutzfrist der Sicherungsdatei des Migrationsarchivs liegt.
- (2) Die Dateien, die in der per Kommando erzeugten Liste stehen, werden in das System-Migrationsarchiv verdrängt.

Verdrängung nach Menge

Es werden so viele Dateien verdrängt, bis eine bestimmte Anzahl PAM-Seiten frei geworden ist.

```
//MIGRATE-FILES - _____ (1)
// FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=$MANUAL.FILE.2*, -
//   RELEASE-PAGES=100, - _____ (2)
//   COMPRESS-FILES=*YES, TO-STORAGE=*S1-STORAGE-LEVEL), -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
//   OUTPUT=HSMS.MAN.R.MGF.3, -
//   WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Die Dateien der Benutzererkennung MANUAL mit dem Namen FILE.2... werden in das System-Migrationsarchiv verdrängt.
- (2) HSMS verdrängt allerdings aus der Menge der angegebenen Dateien nur so viele, bis durch die Verdrängung mindestens 100 PAM-Seiten frei geworden sind.

Behebung von Speicherengpässen auf S0

Um Speicherplatz auf gemeinschaftlichen Datenträgern zu schaffen, werden Dateien auf S1 migriert, auf die seit mindestens 28 Tagen nicht mehr zugegriffen wurde. Vorher könnte man sich mit SHOW-PUBSET-USAGE INFORMATION=*UNUSED-DAYS darüber informieren, ob der so frei gewordene Speicherplatz ausreicht.

```
//MIGRATE-FILES FROM-STORAGE=*S0-STORAGE-LEVEL - _____ (1)
// (FILE-NAMES=:2BY:$*, TO-STORAGE=*S1-STORAGE-LEVEL, -
//   UNUSED-DAYS=28), -
// OPERATION-CONTROL=*PAR(REPORT=*FULL, -
//   OUTPUT=HSMS.MAN.R.MGF.1, -
//   WAIT-FOR-COMPLETION=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Alle Dateien auf dem Pubset 2BY, die mindestens seit mehr als 28 Tagen nicht mehr benutzt wurden, werden auf die Speicherebene S1 migriert.
Die Angabe *ALL ist wegen der pubset-spezifischen Zuordnung der System-Migrationsarchive nicht sinnvoll; *ALL bezieht sich immer auf alle Dateien von allen Pubsets.

Pubset-weise Verdrängung mit Ausgabe der Pubset-Belegung

Einige große Dateien werden verdrängt, um die Auswirkung der Verdrängung auf die Belegung von Pubsets zu zeigen.

```
//SHOW-PUBSET-USAGE _____ (1)
```

```

SHOW-PUBSET-USAGE                                INFORMATION = SUMMARY
CATALOG-ID = ALL
-----
PUBSET  ST      CAPACITY  %USED  %AVAIL  %MIG  S1-MIG  S1-USED  S1-AVAIL
BVWC    S0      225657    87.0   13.0    .0     0       593178   70167
2BC     S1      663345    89.4   10.6    .0     .       593178   70167
2BY     S0      663345    77.5   22.5    .0     135     593178   70167
...
...
...
NEXT-PAGE: + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
    
```

```

% HSM0003 HSMS STATEMENT COMPLETED
//MIGRATE-FILES - _____ (2)
// FROM-STOR=*S0-STOR(F-NAMES=:2BY:$MANUAL.MAX-SIZE.*, -
//   MIN-SIZE=150,MAX-SIZE=2500,TO-STOR=*S1-STOR), -
// OPER-CONTROL=*PAR(REPORT=*NONE)
% HSM0003 HSMS STATEMENT COMPLETED
//MIGRATE-FILES - _____ (3)
// FROM-STOR=*S0-STOR(F-NAMES=:BVWC:$MANUAL.MAX-SIZE.*, -
//   MIN-SIZE=150,MAX-SIZE=2500,TO-STOR=*S1-STOR), -
// OPER-CONTROL=*PAR(REPORT=*NONE)
% HSM0003 HSMS STATEMENT COMPLETED
//MIGRATE-FILES - _____ (3)
// FROM-STOR=*S0-STOR(F-NAMES=:2BY:$MANUAL.MAX-SIZE.*, -
//   MIN-SIZE=2500,TO-STOR=*S2-STOR), -
// OPER-CONTROL=*PAR(REPORT=*NONE)
% HSM0003 HSMS STATEMENT COMPLETED
//MIGRATE-FILES - _____ (4)
// FROM-STOR=*S0-STOR(F-NAMES=:BVWC:$MANUAL.MAX-SIZE.*, -
//   MIN-SIZE=2500,TO-STOR=*S2-STOR), -
// OPER-CONTROL=*PAR(REPORT=*NONE)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
/SHOW-FILE-ATTR :*:$MANUAL.MAX-SIZE.* _____ (4)
%0000001302#:BVWC:$MANUAL.MAX-SIZE.2
%0000095004#:BVWC:$MANUAL.MAX-SIZE.4
%0000080004 :BVWC:$MANUAL.MAX-SIZE.6
%0000001302#:2BY:$MANUAL.MAX-SIZE.1
%0000095004#:2BY:$MANUAL.MAX-SIZE.3
%0000080004 :2BY:$MANUAL.MAX-SIZE.5
%SUM PUBLIC: 2 FILES RES= 160008 FRE= 160008 REL= 160008 PAGES
%SUM PUB/S1: 2 FILES RES= 2604 FRE= 2050 REL= 2046 PAGES
%SUM PUB/S2: 2 FILES RES= 190008 FRE= 160012 REL= 160008 PAGES
//SHOW-PUBSET-USAGE ----- (05)
    
```

```

SHOW-PUBSET-USAGE                                INFORMATION = SUMMARY
CATALOG-ID = ALL

-----
PUBSET ST CAPACITY %USED %AVAIL %MIG S1-MIG S1-USED S1-AVAIL
BVWC S0 225657 44.3 55.7 42.6 1302 593772 69573
2BC S1 663345 89.5 10.5 .0
2BY S0 663345 63.0 37.0 14.5 1437 593772 69573
...
...
...
NEXT-PAGE: + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
    
```

```

% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
    
```

- (1) Ausgabe der Pubset-Belegung. Es sind keine Dateien migriert. Die S0-Pubsets sind mit über 70% ausgelastet.
- (2) Alle Dateien der Benutzerkennung MANUAL, die mit „MAX-SIZE.“ beginnen und zwischen 150 und 2500 Seiten groß sind, werden auf S1 migriert. Diese Dateien werden also auf Platte geschrieben. Es wird kein Report erzeugt. Eine Information über die Dateien ist weiterhin über den Katalog möglich.
- (3) Alle Dateien der Benutzerkennung MANUAL die mit „MAX-SIZE.“ beginnen und über 2500 Seiten groß sind, werden auf S2 migriert. Diese Dateien werden also auf Magnetbandkassette geschrieben.
- (4) Ein SHOW-FILE-ATTRIBUTES-Kommando zeigt die migrierten Dateien und die Speicherebene, auf die sie geschrieben wurden.
MANUAL.MAX-SIZE.5 und .6 wurden nicht verdrängt; sie sind durch die Migrationssperre geschützt.
- (5) Die Belegung der Pubsets wird erneut ausgegeben: Der Prozentsatz der freien Seiten auf den S0-Pubsets ist gestiegen. (%USED und %AVAIL ergeben weiterhin 100%, obwohl auch %MIG > 0 ist. Die beiden Werte beziehen sich lediglich auf die Daten, die aktuell auf dem Pubset stehen, nicht auf die migrierten Daten.)

Automatisches Starten von Migrationsaufträgen durch einen ENTER-Job

Auf dem S0-Pubset 2BY sollen Dateien automatisch verdrängt werden, wenn die Sättigungsstufe 4 oder 5 erreicht wird. Erreicht wird dies durch Starten dieses ENTER-Jobs nach dem Laden von HSMS. Die Erläuterungen zum Ablauf sind im ENTER-Job dokumentiert.

```

/.AUTOMIG SET-LOGON-PARAMETERS
/
/ REMARK =====
/ REMARK === CATID und STORAGE-LEVEL des zu ueberwachen= ===
/ REMARK === den Pubsets sowie, evtl. daraus generiert, ===
/ REMARK === der Name der MIGRATIONS-JV kann als QUASI= ===
/ REMARK === PARAMETER in temp. JV uebergeben!! ===
/ REMARK =====
/ SET-JV-LINK JV-NAME=#CATID
/ MODIFY-JV JV-NAME(#CATID),VALUE='2BY'
/ REMARK >>>>> CATID anpassen >>>>>^<<<<<< CATID anpassen <<
/ REMARK
    
```

```

/          SET-JV-LINK JV-NAME=#ST-LEVEL
/          MODIFY-JV   JV-NAME(#ST-LEVEL),VALUE='S0'
/          REMARK >>>>> STORAGE-LEVEL anpassen >^<<<<<<< ST-LEVEL <<<<
/          REMARK
/          SET-JV-LINK JV-NAME=#MIGJV
/          MODIFY-JV   JV-NAME(#MIGJV),           -
/                   VALUE=' $SYSHSMS.SYS.HSM.MIGRATE.&(#CATID) '
/          REMARK
/          REMARK
/          REMARK =====
/          REMARK === Synchron warten, bis Subsystem HSMS aktiv ist ==
/          REMARK === Spin-Off, wenn nach Wartezeit nicht geladen! ==
/          REMARK =====
/          REMARK
/          WAIT-EVENT UNTIL=                               -
/                   JV((#MIGJV),1,4) NE X'FFFFFFFF')
/          REMARK =====
/          REMARK Wenn Saturation-Level 4 oder 5 erreicht wurde, kann
/          REMARK durch MIGRATION Platz geschaffen werden.
/          REMARK
/          REMARK Freigabe von soviel Seiten, wie in der MIGRATIONS-JV
/          REMARK eingetragen sind, genuegt, um das naechst-niedrigere
/          REMARK Saturation-Level zu erreichen
/          REMARK =====

/.WAIT      WAIT-EVENT UNTIL=                               -
/                   JV((#MIGJV),3,1) GE C'4'),           -
/                   TIMEOUT-LABEL=TIMEOUT
/          REMARK =====
/          REMARK == Wenn HSMS beendet ist, wird dieser Job auch
/          REMARK == beendet
/          REMARK =====
/          SKIP-COMMANDS TO-LABEL=LOGOFF,                   -
/                   IF=JV((#MIGJV),3,1) EQ X'FF')
/          REMARK =====
/          REMARK ==== Anzahl freizugebender Seiten extrahieren ====
/          REMARK ==== und Migrations-Auftrag absetzen               ====
/          REMARK =====
/          SET-JV-LINK JV-NAME=#PAGES
/          MODIFY-JV   JV-NAME(#PAGES),                   -
/                   VALUE=JV(JV-NAME=&(#MIGJV),          -
/                   POSITION=36, LENGTH=10)
/          REMARK
/          REMARK
/          SKIP-COMMANDS TO-LABEL=MIGS1,                   -
/                   IF=JV(#ST-LEVEL EQ 'S1')
/.MIGS0     REMARK =====
/          REMARK ==== Fuer S0-Pubset: Normale Migration ....      ====
/          REMARK =====
/          ASSIGN-SYSDTA TO-FILE=*SYSCMD
/          START-HSMS
//MIGRATE-FILES FROM-STORAGE=*S0-STORAGE-LEVEL           -
//                   (FILE-NAMES=:#CATID):$*.,           -
//                   TO-STORAGE=*S1-STORAGE-LEVEL),       -
//                   RELEASE-PAGES=&(#PAGES),             -
//                   OPERATION-CONTROL=*PARAMETERS(EXPRESS-REQUEST=*YES)
//END
/          SKIP-COMMANDS TO-LABEL=CONT
/.MIGS1     REMARK =====

```

```

/          REMARK ==== Fuer S1-Pubset: ... Reorganisation      ====
/          REMARK =====
/          ASSIGN-SYSDTA TO-FILE=*SYSCMD
/          START-HSMS
//MIGRATE-FILES FROM-STORAGE=*S1-STORAGE-LEVEL              -
//          (S1-PUBSET-ID=&(#CATID),                          -
//          TO-STORAGE=*S1-STORAGE-LEVEL),                    -
//          RELEASE-PAGES=&(#PAGES),                          -
//          OPERATION-CONTROL=*PARAMETERS(EXPRESS-REQUEST=*YES)
//END
/.CONT          REMARK =====
/          REMARK ==== Warten, bis JV von HSMS modifiziert wurde ====
/          REMARK =====
/          SET-JV-LINK JV-NAME=#LAST-TIME
/          MODIFY-JV   JV-NAME(#LAST-TIME),                  -
/                   VALUE=JV(JV-NAME=&(#MIGJV),              -
/                   POSITION=50, LENGTH=25)
/          WAIT-EVENT UNTIL=JV                                -
/                   ((&(#MIGJV),50,25) NE (#LAST-TIME,1,25))
/          REMARK
/          REMARK =====
/          REMARK ==== Solange HSMS noch aktiv ist, wird der     ===
/          REMARK ==== Pubset weiter ueberwacht                 ===
/          REMARK =====
/          SKIP-COMMANDS TO-LABEL=WAIT,                       -
/                   IF=JV(&(#MIGJV),3,1) NE X'FF')
/          SKIP-COMMANDS TO-LABEL=LOGOFF
/          REMARK
/.TIMEOUT      REMARK =====
/          REMARK ==== Wartezeit abgelaufen, doch Ereignis nicht  ===
/          REMARK ==== eingetreten; Toleranz max 10 Wartezeiten!  ===
/          REMARK =====
/          SKIP-COMANDS TO-LABEL=WAIT,                         -
/                   IF=JV($SYSJV.COUNTER LE '0010')
/          SET-JOB-STEP
/          REMARK =====
/          REMARK ==== SPIN-OFF wegen /WAIT-EVENT Time-out oder  ===
/          REMARK ==== echtem Fehler; Protokoll wird gedruckt    ===
/          REMARK =====
/          LOGOFF SYSTEM-OUTPUT=*PRINT
/.LOGOFF      REMARK =====
/          REMARK ==== Normale Terminierung nach Entladen von HSMS ===
/          REMARK =====
/          LOGOFF SYSTEM-OUTPUT=*DELETE

```

Automatisches Starten von Migrationsaufträgen durch ein Programm

Im Programm werden zwei Ereignisse definiert, einmal das Erreichen der Sättigungsstufe 1 und zum anderen die Beendigung der HSMS-Session. Beim Erreichen der Sättigungsstufe werden über den HSMS-Makro Dateien verdrängt. (Zur Verwendung des HSMS-Makros siehe den [Abschnitt „Aufruf von HSMS aus Programmen“](#).)

Die Erläuterungen zum Ablauf sind im Programm dokumentiert.

```

JVTEST  START
        PRINT NOGEN
        AMODE ANY

```

```

RMODE ANY
GPARMOD 31
BASR R10,0
USING *,R10
*
* Definition der Ereigniskennungen fuer zwei Faelle:
* 1. Die Pubsetbelegung ueberschreitet das vorgesehene
*   Saturationlevel 1 und es sollen Dateien migriert werden.
* 2. Die HSMS-Session wird beendet und damit soll auch dieses
*   Programm beendet werden.
*
ENAEI EINAME=MIGRATE,EIIDRET=IDMIGR
ENAEI EINAME=TERMINAT,EIIDRET=IDTERM
*
* Das Ereignis, dass das Saturationlevel 1 ueberschritten wird,
* soll asynchron behandelt werden. Dafuer wird eine Contingency
* definiert.
*
ENACO CONAME=COMIGR,COADAD=CONTADD,COIDRET=IDCONT
*
* Die Bedingungen fuer die oben definierten Ereignisse werden
* festgelegt:
* 1. Im Pubset mit der Katalogkennung '2BC' wird das Saturation-
*   level 1 ueberschritten, d.h., die zugehoerige Jobvariable
*   erhaelt an der Stelle 4 den Wert '1'.
* 2. HSMS wird entladen und das Pubset nicht laenger von HSMS
*   unterstuetzt, d.h., die Jobvariable fuer das Pubset
*   bekommt in den ersten 40 Stellen den Wert x'ff...ff'.
*
ONEVT '($SYSHSMS.SYS.HSM.MIGRATE.2BC ,4,1)=C''1'',           -
      EIID=IDMIGR,POST='MIGR'
*
ONEVT '($SYSHSMS.SYS.HSM.MIGRATE.2BC ,4,1)=X''FF'',           -
      EIID=IDTERM,POST='TERM'
*
XC     RESPONSE,RESPONSE
*
* Fuer die definierten Ereignisse werden die Signale angefordert.
*
* Asynchrones Warten auf die Ueberschreitung des Saturation-
* levels 1 (Wartezeit 10 Minuten):
*
SOLSIG EIID=IDMIGR,                                           -
      COID=IDCONT,LIFETIM=600
*
* synchrones Warten auf die Beendigung von HSMS
* (Wartezeit 12 Minuten):
*
SOLSIG EIID=IDTERM,COND=UNCOND,                               -
      LIFETIM=720,RPOSTAD=RESPONSE
*
DISEI  DS    0H
*
* Deaktivieren der angemeldeten Ereigniskennungen:
*

```



```

        DISEI EIID=IDMIGR
        DISEI EIID=IDTERM
*
TERM      DS      0H
*
*      Beendigung des Programms
*
        TERM
*
        EJECT ,
CONTA     DS      0H
*****
*
*      Ablaufcode fuer die Contingency zur Behandlung der
*      Saturationlevel-Ueberschreitung
*
*****
        BALR  10,0
        USING *,10
*
*
*      Startmeldung: 'MIGRATIONS-CONTINGENCY GESTARTET'
*
*
        WROUT TEXT1,RETCO
*
*      Aufruf der vorgesehenen Migrationsanweisung ueber den
*      HSMS-Makro
*
        PRINT GEN
        HSMS  MF=S,ADDR=HSMSSTRG,LENGTH=255
        PRINT NOGEN
*
*
*      BEENDIGUNG DER CONTINGENCY
*
*
RETCO     DS      0H
        RETCO
*
*****
*
*      Definition der auszufuehrenden Migrationsanweisung:
*      Dateien im Umfang von mindestens 5000 PAM-Seiten sollen
*      nach S1 migriert werden.
*
*****
HSMSSTRG DC      CL255 'MIGRATE-FILES FROM-STORAGE=*S0-STORAGE-LEVEL/
                (FILE-NAMES=:2BC:$*.,RELEASE-PAGES=5000,TO-STORAGE-LEVEL=/
                *S1-STORAGE-LEVEL) '
*
*
*      Definition des Meldungstextes
*
*
TEXT1     DC      Y(TEXT1E-TEXT1)
        DS      XL3
        DC      C'MIGRATIONS-CONTINGENCY GESTARTET'
TEXT1E   EQU     *
*

```

```
IDMIGR  DS    F
IDTERM  DS    F
IDCONT  DS    F
RESPONSE DS   2F
*
CONTADD DC    A(CONTA)
R10     EQU   10
        END
```

4.3.11 Beispiele zur Remigration

Beispiel 1

Der Benutzer TSOS besitzt im SM-Pubset SMS1 die Dateien Y2 bis Y12, die alle 3 PAM-Seiten groß sind. Die Dateien Y10, Y11 und Y12 wurden am 08.05.2016 auf die Ebene S2 verdrängt und später wieder auf die Ebene S0 zurück geholt.

Es wurde seitdem nur lesend auf sie zugegriffen.

Durch die folgende Migrations-Anweisung sollen durch Verdrängen von Dateien auf die Ebene S2 5 PAM-Seiten freier Platz im Pubset SMS1 geschaffen werden:

```
//MIGRATE-FILES ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1) ,  
  FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y* ,  
  RELEASE-PAGES=5) , OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)
```

```

***  MIGRATE - FILES                HSMS V11.0          FULL          REPORT
**  2016-05-21  12:44:17    PAGE    1
    REQUEST-ENVIRONMENT=SM(SMS1)
    REQUEST-NAME=MGF#0015 REQUEST-DATE=2016-05-21 12:44:15 USER-ID=SYSHSMS  REQ
EST-STATE=COMPLETED  WITHOUT ERROR

STATEMENT LISTING:
MIGRATE-FILES  ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1),  FROM-STORAGE=
*SO-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y*,
    RELEASE-PAGES=5), OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)

    ENVIRONMENT                : SM(SMS1)
    ARCHIVE-NAME                : $SYSHSMS.MAX

    SAVE-FILE ATTRIBUTES
    TO-STORAGE                  : S2-STORAGE-LEVEL
    DEVICE-TYPE                 : TAPE-C4
    RETENTION-PERIOD            : 66

    SAVE-VERSION ATTRIBUTES
    SAVE-VERSION-NAME          : MIGRATE

***  CATALOG - SMS1          USER - TSOS          ***
**  OUTPUT SAVE VERSION:  SAVE-VERSION-DATE=16-05-08  SAVE-VERSION-TIME=12:12:03
*
FILE/JOB VARIABLE NAME          LASTPG/          SAVE          INPUT  DEV    SUB    OUTPUT
                                VRS             SIZE          TYPE
VSN      TYP    SAVE
VSN(S)
Y11                                1              0          FULL
*REMIG   D      0
TAPE01
Y12                                1              0          FULL
*REMIG   D      0
TAPE01
    NUMBER OF FILES=          0 GLOBAL SIZE=          0 START= 2016-05-21
12:44:15 END= 2016-05-21 12:44:17

***  E N D   O F                HSMS V11.0          FULL          REPORT
***  2016-05-21  12:44:17    ***

```

Die remigrierbaren Dateien Y11 und Y12 werden bevorzugt verdrängt. Die Remigration geschieht bereits im HSMS-Servertask, und zwar ohne neue Speicherung der Daten auf S2.

Beispiel 2

Durch eine weitere Migrations-Anweisung werden nun die restlichen Dateien Y2 bis Y10 nach S2 verdrängt, wobei Y10 remigriert wird. Dabei erfolgt die Verdrängung in 2 Schritten.

- Im ersten Schritt werden die remigrierbaren Dateien verdrängt (im HSMS-Servertask).
- Im zweiten Schritt werden die verbliebenen Dateien durch einen Archive-Subtask nach dem alten Verfahren verdrängt.

```
//MIGRATE-FILES ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1),
FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y*),
OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)
```

```
A*** MIGRATE - FILES          HSMS V11.0          FULL          REPORT
*** 2016-05-21 12:45:45      PAGE      1
    REQUEST-ENVIRONMENT=SM(SMS1)
    REQUEST-NAME=MGF#0015 REQUEST-DATE=2016-05-21 12:45:17 USER-ID=SYSHSMS REQ
UEST-STATE=COMPLETED WITHOUT ERROR
```

STATEMENT LISTING:

```
MIGRATE-FILES ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1), FROM-STORAGE=*
S0-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y*),
OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)
```

```
ENVIRONMENT          : SM(SMS1)
ARCHIVE-NAME         : $SYSHSMS.MAX
```

```
SAVE-FILE ATTRIBUTES
TO-STORAGE           : S2-STORAGE-LEVEL
DEVICE-TYPE          : TAPE-C4
RETENTION-PERIOD     : 66
```

```
SAVE-VERSION ATTRIBUTES
SAVE-VERSION-NAME    : MIGRATE
```

```
% ARC0033 ARCHIVE SUBTASK TSN '0AAG' GENERATED
% ARC0815 SUBTASK '0' HAS TRANSFERRED '9' PAM PAGES FOR '8' FILES AND '0' JVS
IN '0' SECONDS
```

SAVE FILE IDENTIFIER - S.160521.124519

```
SUBSAVE
NUMBER      VSNS
           0      TAPE02
```

SAVE FILE IDENTIFIER - S.160521.124519

*** CATALOG - SMS1 USER - TSOS ***

```
** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-05-08 SAVE-VERSION-TIME=12:12:03
```

FILE/JOB	VARIABLE	NAME	LASTPG/	SAVE	INPUT	DEV	SUB	OUTPUT
				VERS	SIZE	TYPE		
VSN	TYP	SAVE						
VSN(S)								
Y10				1		0	FULL	
*REMIG	D	0						
TAPE01								

SAVE FILE IDENTIFIER - S.160521.124519

*** CATALOG - SMS1 USER - TSOS ***

```
** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-05-21 SAVE-VERSION-TIME=12:45:20
```

FILE/JOB	VARIABLE	NAME	LASTPG/	SAVE	INPUT	DEV	SUB	OUTPUT
				VERS	SIZE	TYPE		
VSN	TYP	SAVE						
VSN(S)								
Y2				1		1	FULL	

```

PUBK00    D        0
TAPE02
  Y3                      1          1    FULL
PUBK00    D        0
TAPE02
  Y4                      1          1    FULL
PUBK00    D        0
TAPE02
  Y5                      1          1    FULL
PUBK00    D        0
TAPE02
  Y6                      1          1    FULL
PUBK00    D        0
TAPE02
  Y7                      1          1    FULL
PUBK00    D        0
TAPE02
  Y8                      1          1    FULL
PUBK00    D        0
TAPE02
  Y9                      1          1    FULL
PUBK00    D        0
TAPE02
  NUMBER OF FILES=          8 GLOBAL SIZE=          8 START= 2016-05-21
12:45:18 END= 2016-05-21 12:45:45

***      E N D    O F          HSMS V11.0          FULL          REPORT
*** 2016-05-21 12:45:45      ***

```

Beispiel 3

Ausgangslage ist wieder die Dateimenge aus Beispiel 1. Die folgende MIGRATE-Anweisung schafft freien Platz im Pubset SMS1 geschaffen, indem nur remigrierbare Dateien ohne Datenspeicherung nach S2 verdrängt werden.

```

//MIGRATE-FILES ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1),
FROM-STORAGE=*S0-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y*,
MIGRATION-INFO=*REMIGRATION),OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)

```

```

A*** MIGRATE - FILES                HSMS V11.0          FULL          REPORT
*** 2016-05-21 13:04:35          PAGE    1
      REQUEST-ENVIRONMENT=SM(SMS1)
      REQUEST-NAME=MGF#0015 REQUEST-DATE=2016-05-21 13:04:32 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED   WITHOUT ERROR

STATEMENT LISTING:

MIGRATE-FILES  ENVIRONMENT=*SYSTEM-MANAGED(CATALOG-ID=SMS1) ,      FROM-STORAGE=
*SO-STORAGE-LEVEL(FILE-NAMES=:SMS1:Y*,
      MIGRATION-INFO=*REMIGRATION) , OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL)
      ENVIRONMENT                : SM(SMS1)
      ARCHIVE-NAME                : $SYSHSMS.MAX

      SAVE-FILE ATTRIBUTES
      TO-STORAGE                  : S2-STORAGE-LEVEL
      DEVICE-TYPE                 : TAPE-C4
      RETENTION-PERIOD           : 66

      SAVE-VERSION ATTRIBUTES
      SAVE-VERSION-NAME          : MIGRATE

*** CATALOG - SMS1          USER - TSOS          ***
** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-05-08 SAVE-VERSION-TIME=12:12:03
*
FILE/JOB VARIABLE NAME          LASTPG/          SAVE          INPUT          DEV          SUB          OUTPUT
                                VERS          SIZE          TYPE
VSN          TYP          SAVE
VSN(S)
  Y10                                1              0              FULL
*REMIG          D          0
TAPE01
  Y11                                1              0              FULL
*REMIG          D          0
TAPE01
  Y12                                1              0              FULL
*REMIG          D          0
TAPE01
      NUMBER OF FILES=          0 GLOBAL SIZE=          0 START= 2016-05-21
13:04:33 END= 2016-05-21 13:04:35
***          E N D          O F          HSMS V11.0          FULL          REPORT
*** 2016-05-21 13:04:35          ***

```

4.4 Datentransfer von BS2000-Dateien und Jobvariablen

Mit HSMS können Dateien, Jobvariablen und Katalogeinträge (von Dateien auf privaten Datenträgern) auf andere BS2000-Systeme oder auf andere Benutzerkennungen übertragen werden. Dazu werden diese Daten auf Magnetbandkassetten, Privatplatte, gemeinschaftliche Platte oder Net-Storage geschrieben (exportiert), die am Ziel der Übertragung, dem gewünschten System oder der gewünschten Benutzerkennung wieder eingelesen (importiert) werden.

Die erzeugten Magnetbandkassetten sind ARCHIVE-kompatibel, sodass auf dem anderen System entweder HSMS oder ARCHIVE installiert sein muss. Ebenso möglich ist das Importieren von Magnetbandkassetten, die mit der EXPORT-Funktion von ARCHIVE beschrieben wurden. Dazu besteht die Möglichkeit, die Daten ohne die Katalogkennung auf den Datenträger zu schreiben bzw. Daten ohne Katalogkennung zu importieren.

Beim Datentransfer können Verzeichnisse (Directory-Dateien) verwendet werden, die mit den ARCHIVE-Directory-Dateien kompatibel sind. Die Verzeichnisse können hinter die exportierten Daten auf den Datenträger geschrieben werden.

4.4.1 Dateien und Jobvariablen exportieren

Dateien und Jobvariablen werden mit der HSMS-Anweisung EXPORT-FILES exportiert.

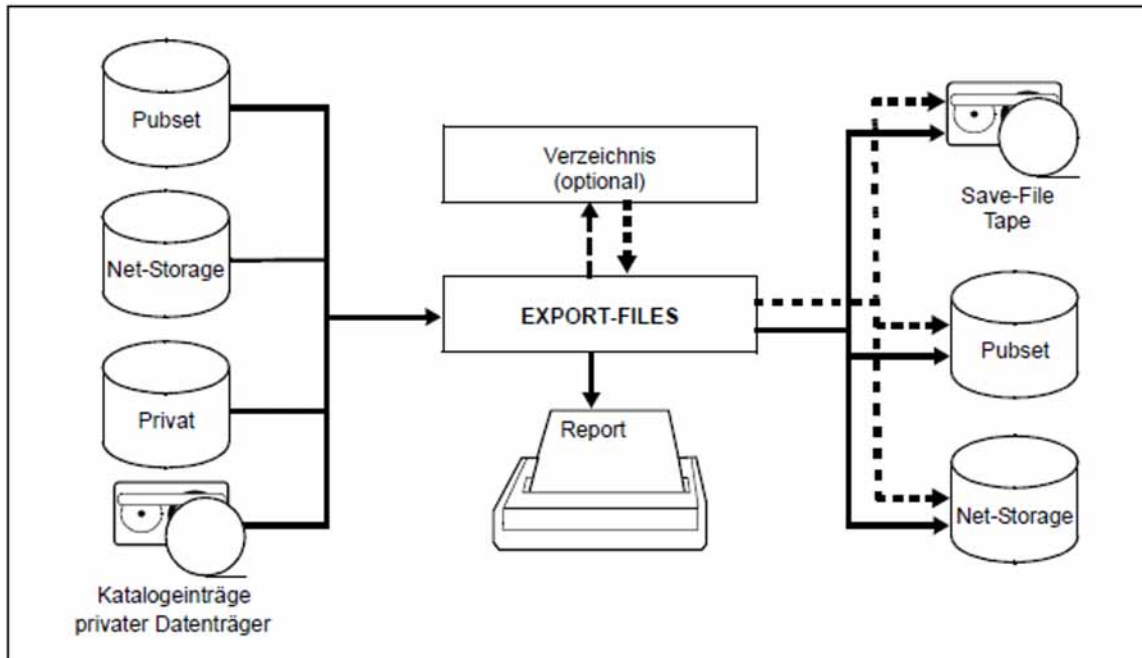


Bild 15: Exportieren mit EXPORT-FILES

Die zu exportierenden Dateien und Jobvariablen von Pubsets oder Privatplatten können nach dem Datenträgertyp ausgewählt werden, auf dem sie liegen (Operand SUPPORT). Bei Auswahl von Dateien auf Net-Storage (Operand STORAGE-TYPE) können die Net-Storage-Dateien zusätzlich auf den Dateityp BS2000 oder Node-File beschränkt werden.

Möglich ist selbstverständlich die Angabe einer Dateinamensliste (siehe [Abschnitt „Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien“](#)).

Migrierte Dateien können ebenfalls exportiert werden (ohne Recall). Dabei werden die Katalogeinträge aus dem Katalog und die Daten von der Migrationsebene übernommen.

Dateien oder Jobvariablen, die durch ein Kennwort geschützt sind, können nur nach Angabe des Kennworts exportiert werden. Dazu dient der Operand PASSWORDS.

Beim Exportieren von Dateien ohne Verzeichnis können auch mehrbenutzbare Dateien anderer Benutzer mitgesichert werden.

Beim Exportieren von Dateien/Jobvariablen mit EXPORT-FILES ohne Verzeichnis kann mit den Operanden NEW-FILE-NAMES/NEW-JV-NAMES ein neuer Name angegeben werden.

Beim Exportieren kann über den Operanden SAVE-FILE bestimmt werden, ob

- eine Sicherungsdatei für diesen Exportlauf neu erzeugt wird (=NEW)
- eine existierende Sicherungsdatei fortgeschrieben wird (=CONTINUE)

Wenn die Daten auf einer anderen Benutzerkennung importiert werden sollen, muss die Sicherungsdatei mehrbenutzbar angelegt werden:

```
SAVE-FILE=*NEW(USER-ACCESS=*ALL-USERS)
```

Außerdem kann die Sicherungsdatei durch ein eigenes Kennwort geschützt werden, ohne dessen Angabe die Sicherungsdatei nicht importiert werden kann.

4.4.1.1 Katalogeinträge exportieren

HSMS sichert und archiviert von einer Datei grundsätzlich sowohl die Daten als auch den Katalogeintrag (falls die Daten verfügbar sind). Beim Exportieren ist es dagegen möglich, von Dateien, die sich auf auf Magnetbandkassetten oder auf Privatplatte befinden, nur die Katalogeinträge auf den EXPORT-Datenträger zu schreiben (Sicherungstyp CATL).

Bei Dateien auf Privatplatte geschieht dies durch die Angabe

```
SUPPORT=*PRIVATE-DISK( . . . , CATALOG-ENTRIES-ONLY=*YES )
```

Bei Banddateien geschieht dies durch die Angabe

```
SUPPORT=*TAPE
```

Um den Import von Dateien in eine andere Systemumgebung zu erleichtern, kann bereits beim Export dieser Dateien angegeben werden, ob

- gewisse Dateiattribute vom Katalogeintrag der Datei übernommen werden oder ob diese Dateiattribute auf den Standardwert gesetzt werden (Operand FILE-ATTRIBUTES=*KEEP/*RESET-TO-STD).
- die Katalog- und Benutzerkennung der Datei in die Sicherungsdatei übernommen werden (Operand CAT-AND-USER-ID=*KEEP/*IGNORE).

Wenn die Katalog- und Benutzerkennung nicht in die Sicherungsdatei übernommen werden, können die Dateien ohne Kenntnis der exportierenden Katalog- und Benutzerkennung importiert werden.

4.4.1.2 Verzeichnis

Beim Exportieren kann auf Wunsch ein Verzeichnis der Daten, die auf den Export-Datenträger geschrieben wurden, erzeugt werden. Dieses Verzeichnis entspricht dem Archivverzeichnis der anderen HSMS-Grundfunktionen und ist zu den ARCHIVE-Directory-Dateien kompatibel.

Veranlasst wird die Erzeugung durch den Operanden DIRECTORY-NAME. Dabei kann außerdem angegeben werden, ob ein neues Verzeichnis angelegt werden soll und ob das Verzeichnis zusätzlich als letzte Datei auf den Datenträger geschrieben werden soll (SAVE-DIRECTORY=*NO/*YES).

Das Verzeichnis kann dann beim Importieren als erste Datei vom Datenträger gelesen werden. Es ermöglicht auf dem anderen BS2000-System einen Überblick über die importierbaren Daten. Allerdings kann sich nur der HSMS-Verwalter Dateien anderer Benutzerkennungen mit der HSMS-Anweisung SHOW-ARCHIVE ausgeben lassen.

4.4.2 Exportierte Dateien und Jobvariablen importieren

Der umgekehrte Vorgang, das Einlesen exportierter Daten von Magnetbandkassetten, gemeinschaftlicher Platte oder Net-Storage (gleichgültig ob sie mit HSMS oder mit ARCHIVE erzeugt wurden) auf ein anderes BS2000-System bzw. eine andere Benutzerkennung heißt Importieren.

Dazu steht die HSMS-Anweisung IMPORT-FILES zur Verfügung.

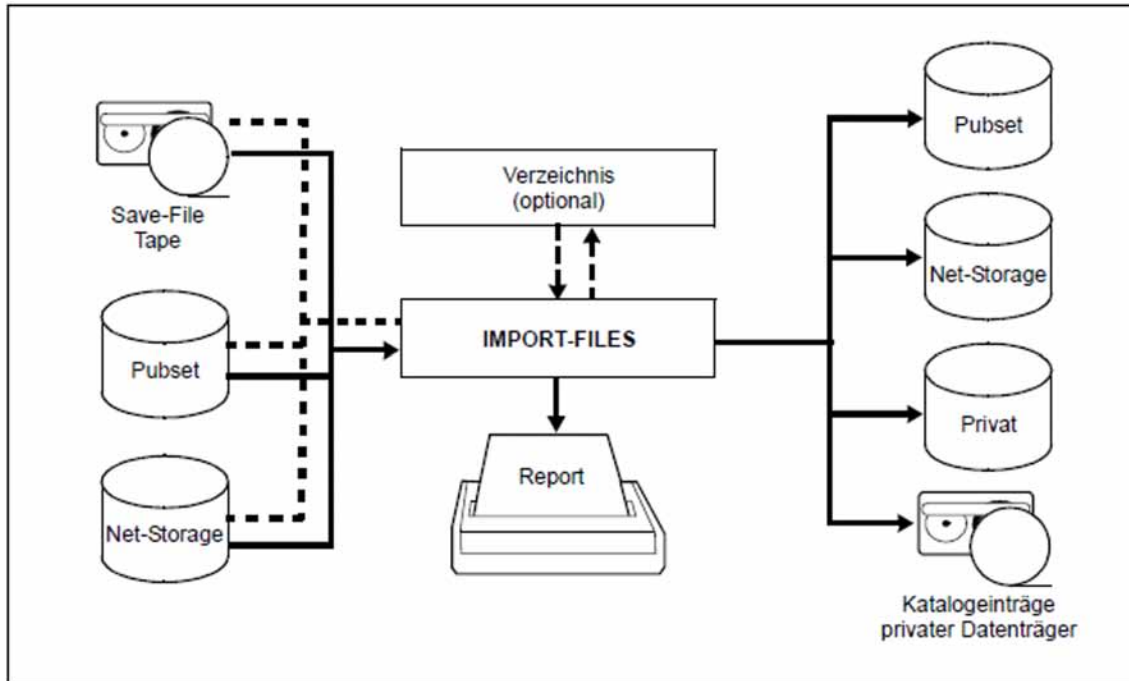


Bild 16: Importieren mit IMPORT-FILES

Beim Importieren müssen die Kennwörter für die Daten und gegebenenfalls für die Sicherungsdatei angegeben werden (Operand PASSWORDS).

Daten, die ohne Katalogkennung exportiert wurden, werden beim Import mit der Standard-Katalogkennung ergänzt.

Beim Importieren von Sicherungsdateien, die in mehreren Parallelläufen erstellt wurden, muss für jeden Parallellauf eine eigene *GROUPED-BY-RUN-Angabe gemacht werden (siehe Beispiel auf "[Beispiele zum Datentransfer](#)").

4.4.2.1 Mit Verzeichnis importieren

Wenn beim Exportieren ein Verzeichnis erstellt und auf den Datenträger geschrieben wurde, kann das Verzeichnis beim Importieren benutzt werden. Dazu muss es in einem eigenen Importlauf importiert werden. Dieser Lauf wird selbst ohne Verzeichnis durchgeführt.

In einem zweiten Importlauf können die anderen Daten unter Angabe des Verzeichnisses importiert werden:

```
//IMPORT-FILES SAVE-FILE=*FROM-DIRECTORY -  
// (DIRECTORY-NAME=<verzeichnisname>)
```

4.4.2.2 Mit MAREN-Unterstützung importieren

Wenn beim Exportieren kein Verzeichnis erstellt wurde, müssten beim Importieren alle VSNs und der Volumetyp der Sicherungsdatenträger angegeben werden. Bei Einsatz von MAREN ab V12.0 kann die Eingabe-Sicherungsdatei auch nur mit ihrer Save-File-ID angegeben werden und MAREN ermittelt zugehörige VSNs und Volumetyp:

```
//IMPORT-FILES SAVE-FILE=*BY-VOLUME-CATALOG -  
// (SAVE-FILE-ID=S.yymmdd.hhmmss)
```

4.4.2.3 Dateien und Jobvariablen umbenennen

Beim Importieren der Dateien besteht mit dem Operanden NEW-FILE-NAMES die Möglichkeit – bei abweichender Benutzerkennung und Konfiguration auch die Notwendigkeit –Dateien umzubennenen. Dies geschieht nicht einzeln pro Datei, sondern für alle Dateien, die in einem Auftrag importiert werden, nach festgelegten Regeln:

- Die Dateien können mit einem Präfix oder Suffix versehen werden.
- Die Dateien können auf eine andere Katalogkennung gespielt werden, wenn der jeweilige Benutzer auf diesem Pubset einen Eintrag im Benutzerkatalog hat.
- Wenn die Sicherungsdatei mit USER-ACCESS=*ALL-USERS angelegt wurde, können die Dateien auf eine andere Benutzerkennung umbenannt werden.
Ohne Mehrbenutzbarkeit hat nur der HSMS-Verwalter diese Möglichkeit.

Entsprechend geschieht dies mit dem Operanden NEW-JV-NAMES für Jobvariablen. Selbstverständlich müssen beim Umbenennen die Konventionen für Dateinamen im BS2000 eingehalten werden.

4.4.2.4 Directory importieren

Wenn das Directory gesichert wurde, kann es im Notfall (nicht mehr vorhanden oder nicht lesbar) von der Sicherungsdatei auf Band, Net-Storage, privater oder gemeinschaftlicher Platte restauriert werden. Dazu dient die Anweisung IMPORT-FILES mit der Angabe von *DIRECTORY und SAVE-FILE. Dabei sind folgende Fälle zu unterscheiden:

- Die Sicherungsdatei befindet sich auf gemeinschaftlicher Platte oder sie wurde ab HSMS V10.0A im Modus *HSMS-V10-COMPATIBLE auf Net-Storage erzeugt. In diesen Fällen muss SAVE-FILE=*BY-PUBLIC-DISK mit der Pubset-ID und der SVID angegeben werden:

```
//IMPORT-FILES FILE-NAMES=*DIRECTORY, SAVE-FILE=*BY-PUBLIC-DISK
// (SAVE-FILE-ID = S.yymmdd.hhmmss, PUBSET-ID = <cat-id>)
```

- Die Sicherungsdatei befindet sich auf Band oder Privatplatte oder sie wurde in HSMS V9.0B oder in HSMS ab V10.0A im Modus *HSMS-V9-COMPATIBLE auf Net-Storage erzeugt. Dann muss SAVE-FILE=*BY-VOLUME angegeben werden.

```
//IMPORT-FILES FILE-NAMES=*DIRECTORY, SAVE-FILE=*BY-VOLUME(
  (SAVE-FILE-ID=S.yymmdd.hhmmss,
   VOLUMES=<vsn 1..6>, DEVICE-TYPE=<c-string 1..8>)
```

Sicherungsdateien auf Band besitzen Several-SVID-Struktur. In diesem Fall müssen Sie eine SVID angeben, anderenfalls eine SFID. Die SFID, SVID und das Volume oder der Pubset sind im Report der Sicherung vermerkt.

4.4.2.5 Dateien auf Net-Storage importieren

Ab BS2000/OSD-BC V9.0 kann einem Pubset zusätzlicher Speicherplatz, den ein Net-Server bereitstellt, als Net-Storage zugeordnet werden. Damit können Dateien von Pubsets entweder auf lokalen Pubset-Platten oder remote auf einem Net-Storage liegen.

BS2000 OSD/BC ab V10.0 unterstützt auf Net-Storage neben dem Dateityp BS2000 auch den Dateityp Node-File.

Im Operanden ORIGINAL-SUPPORT kann angegeben werden, dass Dateien von einem Net-Storage (Operand STORAGE-TYPE) importiert werden. Das Importieren von Net-Storage-Dateien kann zusätzlich auf den Dateityp BS2000 oder Node-File beschränkt werden (Operand FILE-TYPE). Voreingestellt ist das Importieren aller Net-Storage-Dateien unabhängig vom Dateityp.

Im Operanden NEW-SUPPORT kann angegeben werden, dass Dateien beim Importieren auf einem Net-Storage abgelegt werden.

BS2000-Dateien von gemeinschaftlichem Datenträger oder Net-Storage-Dateien vom Dateityp BS2000 können auf einem Net-Storage als Node-File importiert werden. Net-Storage-Dateien vom Dateityp Node-File, deren SAM-Struktur mitgesichert wurde, können auch auf Net-Storage oder auf gemeinschaftlichen Datenträger importiert werden. Dabei kann auch ihr Dateityp geändert werden. Im Gegensatz dazu können Net-Storage-Dateien vom Dateityp Node-File, deren SAM-Struktur nicht mitgesichert wurde, nur auf ein Net-Storage-Volume und nur mit dem ursprünglichen Dateityp Node-File restauriert werden.

4.4.2.6 Dateien aus der Sicherungsdatei eines Pubsets oder des Net-Storage importieren

Auch wenn das Archivverzeichnis einer Sicherung auf gemeinschaftlicher Platte oder Net-Storage nicht mehr existiert oder die Dateien ohne Directory exportiert wurden, können die Dateien wieder importiert werden. In diesen Fällen muss SAVE-FILE=*BY-PUBLIC-DISK mit der Pubset-ID und der SVID angegeben werden:

```
//IMPORT-FILES FILE-NAMES=..., SAVE-FILE=*BY-PUBLIC-DISK  
// (SAVE-FILE-ID = S.yymmdd.hhmmss, PUBSET-ID = <cat-id>)
```

Im Falle von Net-Storage ist der Import nur möglich, wenn die Sicherungsdatei ab HSMS V10.0A im Modus *HSMS-V10-COMPATIBLE auf Net-Storage erzeugt wurde. Anderenfalls können Dateien noch mit SAVE-FILE=*BY-VOLUME() importiert werden:

```
//IMPORT-FILES FILE-NAMES=..., SAVE-FILE=*BY-VOLUME-  
// (SAVE-FILE-ID = S.yymmdd.hhmmss, VOLUMES = <vsn 1..6>, DEVICE-TYPE = NETSTOR)
```

4.4.3 Beispiele zum Datentransfer

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Übertragen von Dateien mit mehreren Parallelläufen
- Übertragen von Katalogeinträgen privater Platten
- Übertragen an einen anderen Rechner mit Verzeichnis
- Übertragen auf eine andere Standard-Katalogkennung
- Übertragen von Dateien auf gemeinschaftlichen Datenträger und Net-Storage

Übertragen von Dateien mit mehreren Parallelläufen

Die Dateien von drei Benutzern sollen übertragen werden; die Dateien sollen parallel bearbeitet werden.

```

/START-HSMS
//EXPORT-FILES - _____ (1)
// F-NAMES=( $MANUAL.FILE.0*, $USEROLD.FILE.1*, -
// $USERNEW.FILE.2* ), -
// DIR-NAME=HSMS.MAN.EXF.DIR.1 (NEW-DIR=*YES), -
// TO-STOR=*TAPE (VOL=( HSMS11, HSMS22, HSMS33 ) ), -
// SAVE-F=*NEW (RET-PER=0), -
// OPER-CONTROL=*PAR (REPORT=*FULL, PAR-RUNS=3, - _____ (2)
// OUT=HSMS.MAN.R.EXF.1, WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//SHOW-ARCHIVE - _____ (3)
// ARCH-NAME=*BY-DIR (DIR-NAME=HSMS.MAN.EXF.DIR.1), -
// SELECT=*SAVE-F

```

```

SHOW-ARCHIVE (SAVE-FILES)                INFORMATION          = SUMMARY
ARCHIVE-NAME      = BY-DIR (:2BY:$TSOS.HSMS.MAN.EXF.DIR.1)
SAVE-FILE-STATE  = ANY                    SAVE-FILE-STORAGE = ANY
CREATED-BEFORE   = LATEST                 EXPIRATION-BEFORE = LATEST
-----
M SFID           CREA-DATE  EXP-DATE  OBS  ACCESS  ST  DEVICE      #VOL #SV #R
UNS
S.160812.155059  16-08-12   16-08-12  YES  ALL     TAP TAPE-C4    3   1   3
...
...
...
NEXT-PAGE: +  (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

```

% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED

```

```
/START-HSMS
//IMPORT-FILES F-NAMES=*ALL, - _____ (4)
//  SAVE-F=*BY-VOL(VOL= -
//    (*GROUPED-BY-RUN(HSMS11), - _____ (5)
//      *GROUPED-BY-RUN(HSMS22), -
//        *GROUPED-BY-RUN(HSMS33))), -
//  REPL-F-AND-JV=*YES, -
//  OPER-CONTROL=*PAR(REPORT=*FULL, PAR-RUNS=3, - _____ (6)
//    OUT=HSMS.MAN.R.IMF.1, -
//    WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Die Dateien von drei verschiedenen Benutzerkennungen werden in einem Exportlauf auf Magnetbandkassetten geschrieben. (Der Report dieses Laufs dient als Beispiel zur Erläuterung des Report-Aufbaus; siehe dazu den [Abschnitt „Reporte“](#).)
- (2) Drei Parallelläufe werden angegeben, auf die HSMS die bei VOLUMES angegebenen Magnetbandkassetten verteilt.
- (3) Da der Exportlauf mit Verzeichnis durchgeführt wurde, können die Attribute der erzeugten Sicherungsdatei mit der SHOW-ARCHIVE-Anweisung ausgegeben werden. Die Dateien wurden in drei Parallelläufen (#RUNS) auf drei Datenträger (#VOL) geschrieben.
- (4) Die Dateien werden in einem Importlauf wieder eingelesen, wobei die Bestimmung der Sicherungsdatei durch die Datenträger geschieht.
- (5) Für jede Magnetbandkassette eines Parallellaufs wird eine GROUPED-BY-RUN-Angabe gemacht. Die Angabe VOLUME=(HSMS11,HSMS22,HSMS33) wäre auch möglich. *GROUPED-BY-RUN führt aber zu einer Laufzeitverbesserung.
- (6) Außerdem müssen drei Parallelläufe vereinbart werden.

Übertragen von Katalogeinträgen privater Platten

Die Katalogeinträge von privaten Platten werden an einen anderen BS2000-Rechner übertragen.

```
/START-HSMS
//EXPORT-FILES F-NAMES=$MANUAL.FILE.*, - _____ (1)
// SUP=*PRIV-DISK(VOL=*ALL,CAT-ENTRIES-ONLY=*YES), - _____ (2)
// DIR-NAME=*NONE, -
// TO-STOR=*TAPE(VOL=HSMS33), -
// SAVE-F=*NEW(RET-PER=0), -
// OPER-CONTROL=*PAR(REPORT=*FULL)
% HSM0002 HSMS STATEMENT ACCEPTED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

An einem anderen Rechner:

```
/START-HSMS
//IMPORT-FILES F-NAMES=*ALL, - _____ (3)
// SAVE-F=*BY-VOL(VOL=HSMS33), -
// ORIG-SUP=*PRIV-DISK(VOL=WORKKB), -
// REPL-F-AND-JV=*YES, -
// OPER-CONTROL=*PAR(REPORT=*FULL)
% HSM0002 HSMS STATEMENT ACCEPTED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) In einem Exportlauf werden die Katalogeinträge aller Dateien auf Privatplatten, deren Namen mit „\$manual.file.“ beginnen, auf Magnetbandkassette geschrieben. Ein Verzeichnis wird nicht angelegt.
- (2) Die Angabe CATALOG-ENTRIES-ONLY bewirkt, dass nicht die Daten, sondern nur die Katalogeinträge in die Sicherungsdatei aufgenommen werden.
- (3) Die Katalogeinträge der dem Suchmuster entsprechenden Dateien, die sich auf der Privatplatte WORKKB befanden, werden in einem Importlauf an einem BS2000-Rechner eingelesen. Die Sicherungsdatei wird über die Magnetbandkassette bestimmt.

Übertragen an anderen Rechner mit Verzeichnis

Eine Reihe von Dateien soll an einen anderen BS2000-Rechner unter eine andere Benutzerkennung übertragen werden. Dabei wird ein Verzeichnis verwendet.

```

/START-HSMS
//EXPORT-FILES F-NAMES=$MANUAL.FILE.*, - _____ (1)
//  DIR-NAME=$MANUAL.HSMS.MAN.EXF.DIR.4 -
//  (NEW-DIR=*YES,SAVE-DIR=*YES), -
//  TO-STOR=*TAPE(VOL=HSMS33), -
//  SAVE-F=*NEW(RET-PER=0), -
//  OPER-CONTROL=*PAR(REPORT=*FULL)
% HSM0002 HSMS STATEMENT ACCEPTED
//END
% HSM0014 HSMS PROGRAM TERMINATED

```

An einem anderen Rechner:

```

/START-HSMS
//IMPORT-FILES F-NAMES=*DIRECTORY, - _____ (2)
//  NEW-F-NAMES=*BY-RULE(NEW-USER-ID=USERNEW2), -
//  SAVE-F=*BY-VOL(VOL=HSMS33), -
//  REPL-F-AND-JV=*YES, -
//  OPER-CONTROL=*PAR(REPORT=*FULL)
% HSM0002 HSMS STATEMENT ACCEPTED
//IMPORT-FILES F-NAMES=$MANUAL.FILE., - _____ (3)
//  NEW-F-NAMES=*BY-RULE(NEW-USER-ID=USERNEW2), -
//  SAVE-F=*FROM-DIR(DIR-NAME=$USERNEW2.HSMS.MAN.EXF.DIR.4), -
//  REPL-F-AND-JV=*YES, -
//  OPER-CONTROL=*PAR(REPORT=*FULL)
% HSM0002 HSMS STATEMENT ACCEPTED
//END
% HSM0014 HSMS PROGRAM TERMINATED

```

- (1) Alle Dateien der Benutzerkennung MANUAL, deren Name mit FILE. beginnt, werden mit EXPORT-FILES auf eine Magnetbandkassetten geschrieben. Für den Lauf wird ein neues Verzeichnis angelegt und mit auf den Export-Datenträger geschrieben. Der Report wird nach SYSLST geschrieben.
- (2) In einem ersten Importlauf wird vorab das Verzeichnis eingelesen. Die Sicherungsdatei wird über den Datenträger bestimmt.
- (3) Anschließend wird für die anderen Dateien ein Importlauf unter Verwendung des Verzeichnisses gestartet. Mit Hilfe des Verzeichnisses wäre auch die Auswahl der Dateien mit SELECT-FILE-NAMES vor dem Lauf möglich.

Übertragen auf andere Standard-Katalogkennung

Ein Benutzer soll in Zukunft seine Dateien standardmäßig auf einem anderen Pubset führen. Dazu müssen die existierenden Dateien auf den neuen Standard-Pubset übertragen werden.

```

/START-HSMS
//EXPORT-FILES F-NAMES=$MANUAL.*, - _____ (1)
// TO-STOR=*TAPE(VOL=HSMS33), -
// SAVE-F=*NEW(RET-PER=0), -
// OPER-CONTROL=*PAR(REPORT=*FULL, -
// OUT=HSMS.MAN.R.EXF.6, WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
/MODIFY-USER-ATTRIBUTES MANUAL, DEFAULT-PUBSET=2BC, - _____ (2)
/ PUBLIC-VOLUME-SET=2BY
/START-HSMS
//IMPORT-FILES F-NAMES=:2BY:$MANUAL.*, - _____ (3)
// NEW-F-NAMES=*BY-RULE(NEW-CAT-ID=2BC), -
// SAVE-F=*BY-VOL(VOL=HSMS33), -
// OPER-CONTROL=*PAR(REPORT=*FULL, -
// OUT=HSMS.MAN.R.IMF.6, WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED

```

- (1) Alle Dateien der Benutzerkennung MANUAL werden mit der HSMS-Anweisung EXPORT-FILES auf eine Magnetbandkassetten geschrieben.
- (2) Die Standard-Katalogkennung der Benutzerkennung MANUAL wird geändert.
- (3) Die Dateien werden importiert und dabei mit dem Operanden NEW-FILE-NAMES auf die Standard-Katalogkennung geschrieben. Die alte Katalogkennung muss bekannt sein, da sie nicht mehr Standard dieser Kennung ist.

Anmerkung

Wenn der Exportlauf mit CATALOG-ID-MODE=*NO durchgeführt wird, ist das Umbenennen beim Importlauf nicht erforderlich. Dann nämlich schreibt HSMS automatisch in die Standard-Katalogkennung.

Übertragen von Dateien auf gemeinschaftlichen Datenträger und Net-Storage

Die Dateien werden auf gemeinschaftlichen Datenträger übertragen:

```
//EXPORT-FILES FILE-NAMES=LM.TEST.* , -  
// TO-STORAGE=*PUBLIC-DISK(PUBSET-ID=IBA6) , -  
// OPERATION-CONTROL=*PAR (REPORT=*FULL , -  
// OUTPUT=LM.EXP.PUB.REP)  
  
% HSM0003 HSMS STATEMENT COMPLETED
```

Die Dateien werden auf Net-Storage übertragen:

```
//EXPORT-FILES FILE-NAMES=LM.TEST.* , -  
// TO-STORAGE=*NET-STORAGE(VOLUMES=NATB00) , -  
// OPERATION-CONTROL=*PAR (REPORT=*FULL , -  
// OUTPUT=LM.EXP.NET.REP)  
  
% HSM0003 HSMS STATEMENT COMPLETED
```

An einem anderen Server:

```
//IMPORT-FILES FILE-NAMES=LM.TEST ,  
// SAVE-FILE=*BY-PUBLIC-DISK( -  
// SAVE-FILE-ID=S.161228.144335 , PUBSET-ID=IBA6) , _____ (1)  
// OPERATION-CONTROL=*PARAMETERS( -  
// REPORT=*FULL , OUTPUT=LM.EXP.REP)  
  
% HSM0003 HSMS STATEMENT COMPLETED
```

- (1) Das Net-Storage Volume NATB00 ist dem Pubset IBA6 zugewiesen. Dieser Pubset kann für die Dateiübertragung verwendet werden.

4.5 Duplizieren von Sicherungsdateien

HSMS ermöglicht das Duplizieren (Kopieren) von Sicherungsdateien innerhalb eines Archivs sowie das Duplizieren von einem Archiv in ein anderes. Das Duplizieren von Sicherungsdateien steht nur dem jeweiligen Archiveigentümer und dem HSMS-Verwalter zur Verfügung. Die kopierte Sicherungsdatei und die darin enthaltenen BS2000-Dateien, Jobvariablen, Knotendateien und Sicherungsversionen werden auch im Archivverzeichnis eingetragen.

Das Duplizieren kann für folgende Aufgaben eingesetzt werden:

Aufgabe	DMS-Dateien	Knotendateien
Verlagern von Sicherungsdateien von Platte auf Band	X	X
Duplizieren von Sicherungsdateien als Vorsorge gegen Datenverlust	X	X
Einsparen von Datenträgern durch Zusammenfassung von Sicherungsdateien (siehe Beispiel im "Beispiele")	X	X
Umwälzen von Datenträgern innerhalb von Langzeitarchiven	X	X
Reorganisieren eines Migrationsarchivs	X	-

X Die Aufgabe wird unterstützt.

- Die Aufgabe wird nicht unterstützt.

4.5.1 Sicherungsdateien kopieren

Sicherungsdateien können mit den HSMS-Anweisungen COPY-SAVE-FILE, COPY-NODE-SAVE-FILE und COPY-EXPORT-SAVE-FILE kopiert werden.

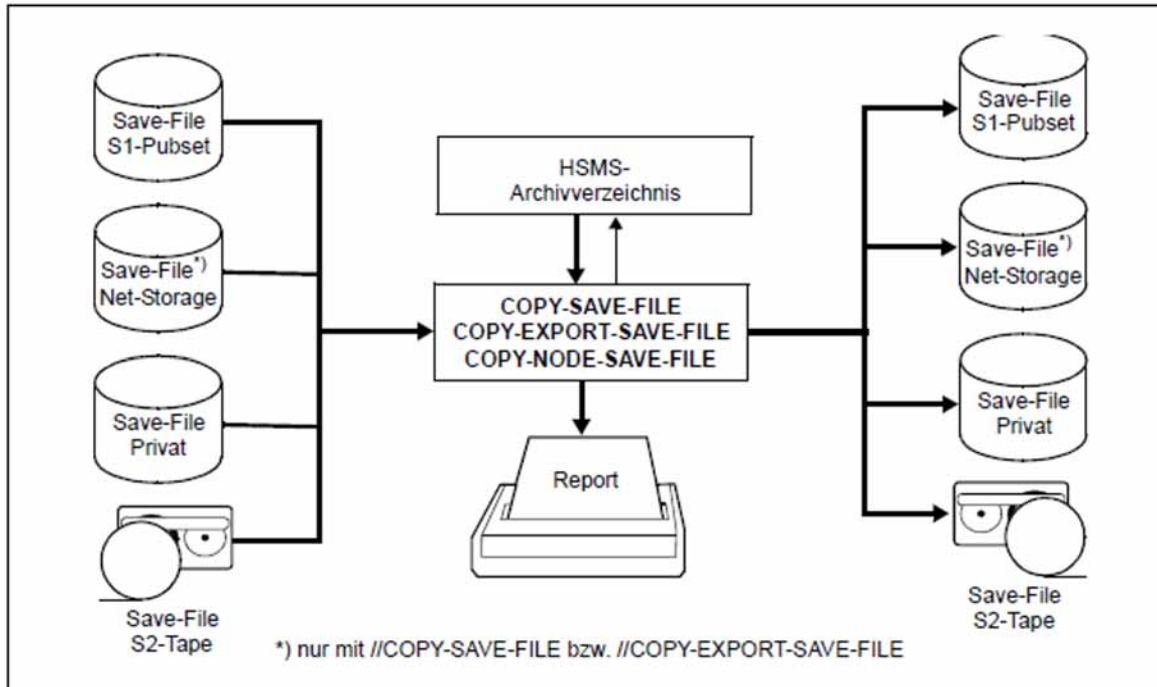


Bild 17: Sicherungsdateien kopieren

Beim Kopieren kann aus der Menge der in der Sicherungsdatei enthaltenen Sicherungsversionen, BS2000-Dateien, Jobvariablen und Knotendateien eine Auswahl getroffen werden.

Die neue Sicherungsdatei und die darin verwalteten Sicherungsversionen erhalten – abhängig von den Ein- und Ausgabearchiven – entweder denselben Zeitstempel oder einen neuen.

Beim Kopieren von Sicherungsversionen eines Migrations- oder Langzeitarchivs wird das Ursprungsdatum (ORIGINAL-DATE) mitgeführt. Das Ursprungsdatum einer Sicherungsversion eines Migrations- oder Langzeitarchivs enthält Datum und Uhrzeit (sekundengenau) der Original-Migration bzw. Original-Archivierung.

Das Ursprungsdatum kann mit folgender Anweisung ausgegeben:

```
//SHOW-ARCHIVE SELECT=*SAVE-VERSIONS( INFORMATION=*USER-INFORMATION)
```

Wenn eine Sicherungsversion durch Archivierung erzeugt wird, erhält sie das ihrer SVID entsprechende Ursprungsdatum.

Wenn eine Sicherungsdatei von der S1-Ebene, von gemeinschaftlicher Platte, Privatplatte oder Net-Storage in die S2-Ebene kopiert wird, werden alle Dateien und Jobvariablen in eine Sicherungsversion kopiert.

4.5.1.1 Doppelführung von Sicherungsdateien

Mit //COPY-SAVE-FILE und //COPY-NODE-SAVE-FILE können Sicherungsdateien dupliziert werden, um so durch doppelte Haltung die Datensicherheit zu erhöhen. Standardmäßig wird dabei die letzte Sicherungsdatei mit allen enthaltenen Daten auf die Speicherebene S2 kopiert. Nur der Archivname muss angegeben werden.

Wenn Sicherungsdateien über einen längeren Zeitraum fortgeschrieben werden, können auch während der Fortsetzungsperiode die Sicherungsdateien sukzessive dupliziert werden:

```
SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES (
    SAVE-VERSION-DATE=*INTERVAL(CREATED-AFTER=<datum> ) )
```

Die am betreffenden Tag jeweils dazugekommenen Sicherungsversionen werden in die Sicherungsdatei der Kopie übernommen.

Sicherungsdateien für den Datentransfer

Mit //EXPORT-FILES erstellte Sicherungsdateien können mit //COPY-EXPORT-SAVE-FILES dupliziert werden. Standardmäßig wird die Kopie auf Magnetbandkassette erstellt. Die Kopie kann auch auf gemeinschaftlicher Platte oder Net-Storage erstellt werden.

Ein Aktualisierung der enthaltenen Daten ist mit //UPDATE-EXPORT-SAVE-FILE möglich.

4.5.1.2 Innerhalb von Backup-Archiven kopieren

Eine Sicherung, die außerhalb der Bandverarbeitungszeiten auf ein S1-Pubset oder eine Privatplatte geschrieben wurde, kann zu einem späteren Zeitpunkt auf die Speicherebene S2, d.h. auf Magnetbandkassetten übertragen werden.

Aus Archiven für Datensicherung (Backup-Archiven) können nur die aktuellen Sicherungsdateien kopiert werden (SAVE-FILE-ID=*LATEST). Auch wenn die Sicherungsversionen ein neues Datum erhalten, bleibt so die zeitliche Reihenfolge der Sicherungen im Archiv mit den Kopien erhalten.

Beim Kopieren von Differenzsicherungen werden die CNS-Einträge mit in die neue Sicherungsdatei übernommen, sodass die Wirkung von //RESTORE-FILES mit SAVE-VERSIONS= *LATEST erhalten bleibt.

4.5.1.3 Innerhalb von Migrationsarchiven kopieren

Das Kopieren von Sicherungsdateien eines Migrationsarchivs kann der Verlagerung auf eine kostengünstigere Speicherebene und dem Reorganisieren des Migrationsarchivs dienen (siehe [Abschnitt „Kopieren der Sicherungsdateien“](#)).

Sicherungsdateien aus Migrationsarchiven können nicht auf Privatplatte oder auf S1 kopiert werden.

4.5.1.4 Innerhalb von Langzeitarchiven kopieren

Archivbänder können bei längerer Lagerzeit umgewälzt werden, um eine Überalterung der Datenträger zu verhindern. Beim Kopieren dieser Archivierungsbänder lassen sich gezielt nur die Sicherungsversionen übernehmen, deren logische Schutzfrist noch nicht abgelaufen ist:

```
//COPY-SAVE-FILE SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES -  
// (EXPIRATION-AFTER=<datum>)
```

Beim Kopieren einer Sicherungsversion erhält die Kopie das Ursprungsdatum des Originals.

Sicherungsdateien aus Langzeitarchiven können nicht auf Privatplatte kopiert werden.

4.5.1.5 Von einem Archiv in ein anderes Archiv kopieren

Sicherungsdateien können von einem Archiv in ein anderes Archiv kopiert werden. Mit dem Operanden TO-ARCHIVE-NAME der HSMS-Anweisungen COPY-SAVE-FILE und COPY-NODE-SAVE-FILE kann ein Zielarchiv für die Kopie angegeben werden. Folgende Möglichkeiten gibt es:

- **Kopie von einem Backup-Archiv in ein anderes Backup-Archiv**

Es wird eine identische Kopie erzeugt, d.h. der Zeitstempel der kopierten Sicherungsversion ist mit dem Original identisch; die CNS-Einträge werden mitkopiert.

Hinweis

Der Zeitstempel der kopierten Sicherungsversion muss größer als der Letzte des Zielarchivs sein.

- **Kopie von einem Langzeitarchiv in ein anderes Langzeitarchiv**

Die Kopie bekommt einen neuen Zeitstempel. Beim Kopieren einer Sicherungsversion erhält die Kopie das Ursprungsdatum des Originals.

- **Kopie von einem Migrationsarchiv in ein anderes Migrationsarchiv**

Die Kopie bekommt einen neuen Zeitstempel. Wenn eine durch Migration erzeugte Sicherungsversion kopiert wird, erhält die Kopie das Ursprungsdatum, das der SVID des Originals entspricht. Wenn eine Sicherungsversion kopiert wird, die ein Ursprungsdatum hat, erhält die Kopie das Ursprungsdatum des Originals.

- **Kopie von einem Migrationsarchiv in ein Langzeitarchiv**

Die Kopie bekommt einen neuen Zeitstempel. Wenn eine durch Migration erzeugte Sicherungsversion kopiert wird, erhält die Kopie das Ursprungsdatum, das der SVID des Originals entspricht. Wenn eine Sicherungsversion kopiert wird, die ein Ursprungsdatum hat, erhält die Kopie das Ursprungsdatum des Originals.

- **Kopie von einem Langzeitarchiv in ein Migrationsarchiv**

Bei einem Ausfall der Migrations-Sicherungsdatei kann die Kopie im Langzeitarchiv als Backup genutzt werden, indem die Sicherungsdatei aus dem Langzeitarchiv in das Migrationsarchiv kopiert wird. Beim Kopieren einer Sicherungsversion erhält die Kopie das Ursprungsdatum des Originals.

- **Kopie von einem Langzeit- oder Migrationsarchiv in ein Backup-Archiv, in dem mehrere Sicherungsversionen pro Sicherungsdatei erlaubt sind**

Die Kopie der Sicherungsversion erhält denselben Zeitstempel wie das Original. Der Zeitstempel der kopierten Sicherungsversion muss größer sein als der letzte Zeitstempel des Zielarchivs.

- **Kopie von einem Backup- oder Langzeitarchiv in das zugeordnete Schattenarchiv oder von einem Schattenarchiv in das zugehörige Backup- oder Langzeitarchiv**

Es wird – genauso wie beim Kopieren zwischen Backup-Archiven – eine identische Kopie mit demselben Zeitstempel erstellt. Allerdings muss hier der Zeitstempel nicht der Letzte des Zielarchivs sein. Das Kopieren wird aber zurückgewiesen, wenn im Zielarchiv bereits eine Sicherungsversion mit demselben Zeitstempel vorhanden ist.

- **Kopie von einem Versions-Backup-Archiv in ein anderes Versions-Backup-Archiv**

Das Kopieren von einem Versions-Backup-Archiv in ein Archiv eines anderen Typs ist nicht möglich und umgekehrt wird das Kopieren von einem Archiv eines anderen Typs in ein Versions-Backup-Archiv zurückgewiesen. Außerdem wird das Kopieren aus dem Versions-Backup-Archiv nicht unterstützt.

4.5.1.6 Zulässige Zielspeichermedien bei der Anweisung COPY-SAVE-FILE

Das Zielspeichermedium beim Kopieren von Sicherungsdateien ist abhängig vom Typ des Archivs, in dem zu kopierenden Dateien abgelegt sind:

- **Kopieren aus einem Backup-Archiv**

In einer SM-Umgebung kann nur auf die Speicherebenen S1 und S2 kopiert werden.

In einer SF-Umgebung kann auf die Speicherebenen S1 und S2, auf Privatplatte und Net-Storage kopiert werden. Der HSMS-Administrator kann außerdem auf gemeinschaftliche Platte kopieren.

Wenn nur die letzte Sicherungsversion ausgewählt ist, kann aus einem Backup-Archiv, das mehrere Sicherungsversionen enthält, genauso wie aus einem Backup-Archiv mit SINGLE-SVID-Struktur auf die Speicherebene S1 kopiert werden.

Wenn aus einem Backup-Archiv, das mehrere Sicherungsversionen enthält, auf die Speicherebene S1, auf gemeinschaftliche Platte oder Privatplatte und nicht innerhalb desselben Archivs kopiert wird, dann wird für jede Sicherungsversion der Original-Sicherungsdatei auf dem Zielmedium eine eigene Sicherungsdatei erzeugt.

- **Kopieren aus einem Langzeitarchiv**

In einer SM-Umgebung kann nur auf die Speicherebenen S1 und S2 kopiert werden.

In einer SF-Umgebung kann auf die Speicherebenen S1 und S2 und auf Net-Storage kopiert werden. Der HSMS-Administrator kann außerdem auf gemeinschaftliche Platte kopieren.

Aus einem Langzeitarchiv kann nicht auf Privatplatte kopiert werden.

Wenn aus einem Langzeitarchiv, das mehrere Sicherungsversionen enthält, auf die Speicherebene S1, auf Privatplatte, gemeinschaftliche Platte oder Net-Storage kopiert wird, dann wird für jede Sicherungsversion der Original-Sicherungsdatei auf dem Zielmedium eine eigene Sicherungsdatei erzeugt.

Mit der Anweisung COPY-SAVE-FILE können nur Sicherungsdateien auf Magnetbandkassette fortgeschrieben werden.

- **Kopieren aus einem Migrationsarchiv**

Aus Migrationsarchiven kann ausschließlich auf die Speicherebene S2 kopiert werden.

4.5.1.7 Verhalten der Anweisung COPY-SAVE-FILE in Verbindung mit von VSNs unabhängigen Sicherungsdateinamen

Das Kopieren auf die Speicherebene S1, auf gemeinschaftliche Platte oder Net-Storage wird abgewiesen, wenn dort bereits eine Sicherungsdatei mit derselben SFID existiert.

4.5.1.8 Verhalten der Anweisung COPY-SAVE-FILE ... TO-STORAGE=*S1-STORAGE-LEVEL abhängig von den Umgebungen, in denen ursprüngliches und Zielarchiv definiert sind

Das Ergebnis der Anweisung COPY-SAVE-FILE mit der Angabe TO-STORAGE=*S1-STORAGE-LEVEL hängt davon ab, in welcher Umgebung ursprüngliches und Zielarchiv definiert sind.

- Falls das Zielarchiv in der SF-Umgebung definiert ist, wird die Definition der S1-Ebene den globalen HSMS-Parametern entnommen. Ein globaler S1-Pubset ist entweder ein normaler SF-Pubset oder ein S1-SM-Pubset. Bei einem S1-SM-Pubset sollten all seine Volume-Sets mit USAGE=*STD definiert sein und die Sicherungsdateien können auf einem beliebigen Volume-Set dieses S1-SM-Pubsets abgelegt sein. Nur der Control-Volume-Set sollte nicht verwendet werden. Dies kann verhindert werden, indem mit der Anweisung MODIFY-PUBSET-RESTRICTIONS eine entsprechende Einschränkung festgelegt wird.
- Falls das Zielarchiv in der SM-Umgebung definiert ist, wird die Definition der S1-Ebene den lokalen SM-Pubset-Parametern der Umgebung entnommen. Ab HSMS V11.0A kann diese S1-Ebene sich auf einen Volume-Set beschränken oder über alle Volume-Sets unter HSMS-Kontrolle verteilt sein.

Die folgenden Tabellen zeigen die möglichen Ergebnisse der Anweisung COPY-SAVE-FILE ... TO-STORAGE=*S1-STORAGE-LEVEL in Abhängigkeit von der Umgebung der betroffenen Archive:

- Ursprüngliches Archiv *SINGLE-FEATURE

Umgebung des Zielarchivs	S1-Ebene in der Ziel-Umgebung (in SF-Umgebung globale S1-Ebene)	Ergebnis der Anweisung COPY-SAVE-FILE ... TO-STORAGE=*S1-STORAGE-LEVEL
*SINGLE-FEATURE	NOT-DEFINED	Die Anweisung wird mit Meldung HSM0214 abgewiesen.
*SINGLE-FEATURE	SF1	Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf dem SF-Pubset SF1 angelegt. SF1 kann auch ein S1-SM-Pubset sein. In diesem Fall wird die Kopie der Sicherungsdatei auf einem Volume-Sets mit USAGE=*STD abgelegt. Datei- und Jobvariablenamen bleiben unverändert, falls nicht anders angegeben.
*SYSTEM-MANAGED (CATALOG-ID = SM1)	NOT-DEFINED	Die Anweisung wird mit Meldung HSM0213 abgewiesen.
*SYSTEM-MANAGED (CATALOG-ID = SM1)	HSM1	Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf dem Volume-Set HSM1 des SM-Pubsets SM1 angelegt. Dateien und Jobvariablen werden automatisch umbenannt. Die Katalogkennung wird in SM1 geändert.

*SYSTEM-MANAGED (CATALOG-ID = SM1)	*ALL-HSMS- CONTROLLED	Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf einem oder mehreren Volume-Sets unter HSMS-Kontrolle des SM-Pubsets SM1 angelegt (abhängig von der Kapazität). Dateien und Jobvariablen werden automatisch umbenannt. Die Katalogkennung wird in SM1 geändert.
---------------------------------------	--------------------------	---

- Ursprüngliches Archiv *SYSTEM-MANAGED(CATALOG-ID = SM1)

Umgebung des Zielarchivs	S1-Ebene in der Ziel-Umgebung (in SF-Umgebung globale S1-Ebene)	Ergebnis der Anweisung COPY-SAVE-FILE ... TO-STORAGE=*S1-STORAGE-LEVEL
*SINGLE-FEATURE	NOT-DEFINED	Die Anweisung wird mit Meldung HSM0214 abgewiesen.
*SINGLE-FEATURE	SF1	Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf dem SF-Pubset SF1 angelegt. SF1 kann auch ein S1-SM-Pubset sein. In diesem Fall wird die Kopie der Sicherungsdatei auf einem Volume-Set mit USAGE=*STD abgelegt. Falls es sich bei der Zieldatei nicht um ein Langzeitarchiv handelt, muss explizit die Umbenennung von Dateien /Jobvariablen festgelegt sein: Die Katalogkennung muss geändert werden. Andernfalls wird die Anweisung mit der Meldung HSM02A3 abgewiesen.
*SYSTEM-MANAGED (CATALOG-ID = SM1)	NOT-DEFINED	Die Anweisung wird mit Meldung HSM0213 abgewiesen.
*SYSTEM-MANAGED (CATALOG-ID = SM1)	HSM1/*ALL-HSMS- CONTROLLED	Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf HSM1 bzw. auf einem oder mehreren Volume-Sets unter HSMS-Kontrolle des SM-Pubsets SM1 angelegt (abhängig von der Kapazität). Beachten Sie: Falls die ursprüngliche Sicherungsdatei ebenfalls auf der S1-Ebene abgelegt ist und //COPY-SAVE-FILE von einem Backup-Archiv in ein anderes kopieren soll, wird keine Kopie erzeugt. Grund: Die Zeitstempel der ursprünglichen und der Zieldatei und damit ihre Dateinamen sind identisch.
*SYSTEM-MANAGED (CATALOG-ID = SM2)	NOT-DEFINED	Die Anweisung wird mit Meldung HSM0213 abgewiesen.

<p>*SYSTEM-MANAGED (CATALOG-ID = SM2)</p>	<p>HSM2/*ALL-HSMS- CONTROLLED</p>	<p>Die Anweisung wird akzeptiert. Die Kopie der Sicherungsdatei wird auf HSM2 bzw. auf einem oder mehreren Volume-Sets unter HSMS-Kontrolle des SM-Pubsets SM2 angelegt (abhängig von der Kapazität). Dateien und Jobvariablen werden automatisch umbenannt. Die Katalogkennung wird in SM2 geändert.</p>
---	---------------------------------------	---

4.5.2 Beispiele

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Zusammenfassen von Sicherungsdateien
- Verlagern von Sicherungsdateien von S1 auf S2
- Verlagern von Knoten-Sicherungsdateien von S1 auf S2

Zusammenfassen von Sicherungsdateien

Verschiedene Archivierungen wurden in mehrere Sicherungsdateien auf mehreren Magnetbandkassetten durchgeführt. Diese Sicherungsdateien sollen zusammengefasst werden. Die Zusammenfassung ist nur für Migrations- und Langzeitarchive möglich und sinnvoll.

In der neu erzeugten Sicherungsdatei sind die ursprünglichen Sicherungsversionen noch erhalten.

```
/START-HSMS
//COPY-SAVE-FILE ARCH-NAME=$SYSHSMS.HSMS.AR.2BY, - _____ (1)
// S-F-ID=S.160812.142005,SAVE-F=*NEW, -
// TO-STOR=*S2-STOR(VOL=*FROM-POOL), -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.CSF.13, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//COPY-SAVE-FILE ARCH-NAME=$SYSHSMS.HSMS.AR.2BY, - _____ (2)
// S-F-ID=S.160812.142018,SAVE-F=*CONT(S-F-ID=*LATEST), -
// TO-STOR=*S2-STOR, -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.CSF.14, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

Report HSMS.MAN.R.CSF.13 (Ausschnitt): _____ (3)

```

*** COPY-SAVE-FILE          HSMS V11.0          FULL          REPORT *** 2016-08-12  14:
21:53          PAGE          3
  REQUEST-NAME=CSF#0AAK REQUEST-DATE=2016-08-12 14:21:40 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED   WITHOUT ERROR

                                SAVE FILE IDENTIFIER - S.160812.142142

SUBSAVE
  NUMBER          VSNS

                                SAVE FILE IDENTIFIER - S.160812.142142
*** CATALOG - 2BY          USER - MANUAL          ***
** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-08-12 SAVE-VERSION-TIME=14:21:43 *
  FILE/JOB VARIABLE NAME          LASTPG/ SAVE INPUT DEV SUB OUTPUT
                                VERS    SIZE TYPE   VSN TYP SAVE VSNS(S)

FILE.01          1          3 FULL HSMS11 T    0 HSMS33
FILE.02          1          8 FULL HSMS11 T    0 HSMS33
FILE.03          1         13 FULL HSMS11 T    0 HSMS33
FILE.11          1          3 FULL HSMS11 T    0 HSMS33
FILE.12          1          8 FULL HSMS11 T    0 HSMS33
FILE.13          1         13 FULL HSMS11 T    0 HSMS33
FILE.21          1          3 FULL HSMS11 T    0 HSMS33
FILE.22          1          8 FULL HSMS11 T    0 HSMS33
FILE.23          1         13 FULL HSMS11 T    0 HSMS33

***      E N D    O F          HSMS V11.0          FULL          REPORT *** 2016-08-12  14:
21:53          ***

```

Report HSMS.MAN.R.CSF.14 (Ausschnitt): _____ (4)

```

*** COPY-SAVE-FILE          HSMS V11.0          FULL          REPORT *** 2016-08-12
14:23:51    PAGE      3
      REQUEST-NAME=CSF#0AAK REQUEST-DATE=2016-08-12 14:21:54 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED WITHOUT ERROR

                        SAVE FILE IDENTIFIER - S.160812.142142

SUBSAVE
NUMBER      VSNS

      0      HSMS33

                        SAVE FILE IDENTIFIER - S.160812.142142
*** CATALOG - 2BY          USER - MANUAL          ***
** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-08-12 SAVE-VERSION-TIME=14:21:57 *
      FILE/JOB VARIABLE NAME          LASTPG/ SAVE  INPUT DEV SUB  OUTPUT
                                VERS    SIZE  TYPE   VSN  TYP SAVE  VSN(S)

FILE.04          1          18  FULL  HSMS22  T    0  HSMS33
FILE.05          1          16  FULL  HSMS22  T    0  HSMS33
FILE.06          1          11  FULL  HSMS22  T    0  HSMS33
FILE.14          1          18  FULL  HSMS22  T    0  HSMS33
FILE.15          1          16  FULL  HSMS22  T    0  HSMS33
FILE.16          1          11  FULL  HSMS22  T    0  HSMS33
FILE.24          1          18  FULL  HSMS22  T    0  HSMS33
FILE.25          1          16  FULL  HSMS22  T    0  HSMS33
FILE.26          1          11  FULL  HSMS22  T    0  HSMS33

** OUTPUT SAVE VERSION: SAVE-VERSION-DATE=16-08-12 SAVE-VERSION-TIME=14:21:58 *
MAX-SIZE.1          1          277  FULL  HSMS22  T    0  HSMS33
MAX-SIZE.3          1         14998  FULL  HSMS22  T    0  HSMS33

***      E N D    O F          HSMS V11.0          FULL          REPORT *** 2016-08-12
14:23:51          ***

```



```
//MODIFY-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.AR.2BY, - _____ (5)
// SAVE-F=*DEL(S-F-ID=(S.160812.142005,S.160812.142018))
% HSM0003 HSMS STATEMENT COMPLETED
//SHOW-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.AR.2BY,SEL=*VOLUMES _____ (6)
```

```
SHOW-ARCHIVE (VOLUMES)
ARCHIVE-NAME = $SYSHSMS.HSMS.AR.2BY
VOLUME-STATE = ANY
-----
VSN          SFID                VOLUME-STATE    EXP-DATE    DEVICE    OWNER
HSMS11                          AVAILABLE                          TAPE-C4    POOL
HSMS22                          AVAILABLE                          TAPE-C4    POOL
HSMS33    S.160812.142142    OBSOLETE        16-08-12    TAPE-C4    POOL
...
NEXT-PAGE: + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
```

```
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Die durch die SFID bestimmte Sicherungsdatei wird in eine neue Sicherungsdatei kopiert. Der Datenträger wird dem Pool des Archivs entnommen.
- (2) Eine weitere Sicherungsdatei wird kopiert. Durch die Angabe CONTINUE= *LATEST wird die mit der vorherigen COPY-SAVE-FILE erzeugte Sicherungsdatei fortgesetzt. Die beiden bisherigen Sicherungsdateien befinden sich jetzt auf einer Magnetbandkassette in einer Sicherungsdatei, was auch die Reporte zeigen.
- (3) Der Report des ersten Kopierauftrags wird ausgegeben.
- (4) Der Report des zweiten Kopierauftrags wird ausgegeben.

In beiden Reporten werden die Sicherungsdateien und die darin enthaltenen Dateien aufgelistet. In beiden Reporten ist die SFID dieselbe.

- (5) Die beiden ursprünglichen Sicherungsdateien werden freigegeben.
- (6) Die Magnetbandkassetten HSMS11 und HSMS22 wurden freigegeben. Nur noch die neu beschriebene Magnetbandkassette HSMS33 ist in Gebrauch.

Verlagern von Sicherungsdateien von S1 nach S2

In diesem Beispiel sollen die täglichen Sicherungen der DMS-Dateien statt auf S2-Ebene zuerst auf S1-Ebene erfolgen. Diese werden regelmässig von der S1- auf die S2-Ebene verlagert, sofern sich die Sicherungsversionen bereits länger als drei Tage auf der S1-Ebene befinden.

Tägliches Sichern mit der Anweisung BACKUP-FILES:

```
//BCF FILE-NAMES=DMS. -
//      ,SELECT-FILES=*ALL-FILES -
//      ,ARCHIVE-NAME=MANUAL.DMS -
//      ,TO-STORAGE=*S1-STORAGE-LEVEL -
//      ,OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=#2)
```

Übersicht der Sicherungsversionen nach drei Sicherungen, vor MOVE-SAVE-FILE

```
SHOW-ARCHIVE (SAVE-VERSIONS)          INFORMATION = SUMMARY
ENVIRONMENT   = SF                    ARCHIVE-NAME = $SYSHSMS.MANUAL.DMS
SV-NAME       = ANY                  SV-DATE      = INTERVAL EARLIEST LATEST
USER-ID       = OWN                  EXP-DATE     = ANY
```

M	SAV-DATE	SAV-TIME	EXP-DATE	SFID	SEL-F	BC	IND	USER-ID	SV-NAME
	16-05-12	16:41:02	16-05-14	S.160512.164102	ALL-F	D		SYSHSMS	
	16-05-13	08:59:33	16-05-13	S.160513.085933	MOD-F	D		SYSHSMS	
	16-05-14	15:50:13	16-05-14	S.160514.155013	MOD-F	D			

```
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
```

Verlagern aller Sicherungsversionen, die sich länger als drei Tage auf S1 befinden

Die Anweisung MOVE-SAVE-FILES wurde am 16.05.16 ausgeführt:

```
//MSF ARCHIVE-NAME=MANUAL.DMS -
//      ,FROM-STORAGE=*DISK(MINIMUM-DAYS-ON-DISK=3) -
//      ,OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=#1)
```

Übersicht der Sicherungsversionen nach drei Sicherungen, nach MOVE-SAVE-FILE

```

SHOW-ARCHIVE (SAVE-VERSIONS)          INFORMATION = SUMMARY
ENVIRONMENT = SF                      ARCHIVE-NAME = $SYSHSMS.MANUAL.DMS
SV-NAME = ANY                        SV-DATE = INTERVAL EARLIEST LATEST
USER-ID = OWN                         EXP-DATE = ANY
-----
M SAV-DATE SAV-TIME EXP-DATE SFID          SEL-F BC IND USER-ID SV-NAME
  16-05-12 16:41:03 16-05-12 S.160516.110025 ALL-F D      SYSHSMS
  16-05-13 08:59:34 16-05-13 S.160516.110025 MOD-F D      SYSHSMS
  16-05-14 15:50:13 16-05-14 S.160514.155013 MOD-F D
-----
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

Übersicht der Sicherungsdateien nach MOVE-SAVE-FILE

```

HOW-ARCHIVE (SAVE-FILES)             INFORMATION = SUMMARY
ENVIRONMENT = SF                     ARCHIVE-NAME = $SYSHSMS.MANUAL.DMS
SAVE-FILE-STATE = ANY                SAVE-FILE-STORAGE = ANY
CREATED-BEFORE = LATEST               EXPIRATION-BEFORE = LATEST
-----
M SFID          CREA-DATE EXP-DATE OBS ACCESS ST DEVICE      #VOL #SV #RUNS
  S.160514.155013 16-05-14 16-05-14 YES OWNER PUB          1    1    1
  S.160516.110025 16-05-16 16-05-16 YES OWNER TAP TAPE-C4  1    2    1
-----
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

Verlagern von Knoten-Sicherungsdateien von S1 auf S2

Das Beispiel zeigt die Verlagerung aller S1-Sicherungsdateien auf S2-Ebene für Knotendateien mit Schattenarchiv.

Statt auf S2-Ebene sollen alle Knotendateien zunächst auf S1-Ebene gesichert werden (z.B. wegen Hardwarewartung der Bandgeräte). Anschliessend werden alle S1-Sicherungen auf die S2-Ebene verlagert. Dem Archiv ist ein Schattenarchiv zugeordnet.

Mit der Anweisung BACKUP-NODE-FILES werden drei Sicherungen auf Public-Platte (S1-Ebene) durchgeführt:

```
//BNF PATH-NAMES=*PATH-NAME (PATH=/manual) -
//      ,ARCHIVE-NAME=MANUAL.NF -
//      ,SELECT-FILES=*MODIFIED-FILES -
//      ,TO-STORAGE=*PUBLIC-DISK (PUBSET-ID=DUB) -
//      ,OPERATION-CONTROL=*PARAMETERS (REPORT=*FULL, OUTPUT=#1)
```

Sicherungsversionen und Sicherungsdateien nach drei Sicherungen

- Hauptarchiv

```
SHOW-ARCHIVE (SAVE-VERSIONS)          INFORMATION = SUMMARY
ENVIRONMENT   = SF                     ARCHIVE-NAME = $SYSHSMS.MANUAL.NF
SV-NAME       = ANY                    SV-DATE      = INTERVAL EARLIEST LATEST
USER-ID       = OWN                    EXP-DATE     = ANY
```

M	SAV-DATE	SAV-TIME	EXP-DATE	SFID	SEL-F	BC	IND	USER-ID	SV-NAME
	16-05-19	13:13:57	16-05-19	S.160519.131355	MOD-F			SYSHSMS	
	16-05-19	13:14:43	16-05-19	S.160519.131441	MOD-F			SYSHSMS	
	16-05-19	13:15:24	16-05-19	S.160519.131522	MOD-F			SYSHSMS	

```
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
```

```

SHOW-ARCHIVE (SAVE-FILES)          INFORMATION          = SUMMARY
ENVIRONMENT          = SF          ARCHIVE-NAME          = $SYSHSMS.MANUAL.NF
SAVE-FILE-STATE = ANY          SAVE-FILE-STORAGE = ANY
CREATED-BEFORE = LATEST          EXPIRATION-BEFORE = LATEST
-----
M SFID          CREA-DATE  EXP-DATE  OBS  ACCESS ST  DEVICE          #VOL #SV #RUNS
S.160519.131355 16-05-19  16-05-19  YES  OWNER  PUB          1 1
S.160519.131441 16-05-19  16-05-19  YES  OWNER  PUB          1 1
S.160519.131522 16-05-19  16-05-19  YES  OWNER  PUB          1 1
-----
NEXT-PAGE : +      (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

- Schattenarchiv

Das Schattenarchiv ist noch leer. Sicherungen auf S1-Ebene/Public wurden noch nicht in das Schattenarchiv übernommen!

Mit der Anweisung MOVE-SAVE-FILES werden die Sicherungen auf die S2-Ebene verlagert:

```

//MSF ARCHIVE-NAME=MANUAL.NF -
//      ,ENVIRONMENT=*NODE-STD -
//      ,OPERATION-CONTROL=*PARAMETERS (REPORT=*SUMMARY ,OUTPUT=#1)

```

Sicherungsversionen und Sicherungsdateien nach der Verlagerung

- Hauptarchiv

```

SHOW-ARCHIVE (SAVE-VERSIONS)          INFORMATION = SUMMARY
ENVIRONMENT = SF                      ARCHIVE-NAME = $SYSHSMS.MANUAL.NF
SV-NAME = ANY                         SV-DATE = INTERVAL EARLIEST LATEST
USER-ID = OWN                          EXP-DATE = ANY
-----
M SAV-DATE SAV-TIME EXP-DATE SFID          SEL-F BC IND USER-ID SV-NAME
  16-05-19 13:13:58 16-05-19 S.160519.131806 MOD-F          UNDEF
  16-05-19 13:14:44 16-05-19 S.160519.131806 MOD-F          UNDEF
  16-05-19 13:15:25 16-05-19 S.160519.131806 MOD-F          UNDEF
-----
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

```

HOW-ARCHIVE (SAVE-FILES)              INFORMATION = SUMMARY
ENVIRONMENT = SF                      ARCHIVE-NAME = $SYSHSMS.MANUAL.NF
SAVE-FILE-STATE = ANY                 SAVE-FILE-STORAGE = ANY
CREATED-BEFORE = LATEST                EXPIRATION-BEFORE = LATEST
-----
M SFID          CREA-DATE EXP-DATE OBS ACCESS ST DEVICE #VOL #SV #RUNS
  S.160519.131806 16-05-19 16-05-19 YES OWNER TAP TAPE-C4 1 3 1
-----
NEXT-PAGE : + (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

- Schattenarchiv

```

SHOW-ARCHIVE (SAVE-VERSIONS)          INFORMATION = SUMMARY
ENVIRONMENT      = SF                 ARCHIVE-NAME = $SYSHSMS.MANUAL.NF.SH
SV-NAME         = ANY                 SV-DATE     = INTERVAL EARLIEST LATEST
USER-ID        = OWN                 EXP-DATE    = ANY
-----
M SAV-DATE SAV-TIME  EXP-DATE  SFID          SEL-F  BC IND  USER-ID  SV-NAME
  16-05-19 13:13:58  16-05-19  S.160519.131806  MOD-F          UNDEF
  16-05-19 13:14:44  16-05-19  S.160519.131806  MOD-F          UNDEF
  16-05-19 13:15:25  16-05-19  S.160519.131806  MOD-F          UNDEF
-----
NEXT-PAGE : +   (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

```

SHOW-ARCHIVE (SAVE-FILES)            INFORMATION      = SUMMARY
ENVIRONMENT      = SF                 ARCHIVE-NAME    = $SYSHSMS.MANUAL.NF.SH
SAVE-FILE-STATE = ANY                 SAVE-FILE-STORAGE = ANY
CREATED-BEFORE  = LATEST              EXPIRATION-BEFORE = LATEST
-----
M SFID          CREA-DATE  EXP-DATE  OBS  ACCESS ST  DEVICE      #VOL #SV #RUNS
  S.160519.131806 16-05-19  16-05-19  YES  OWNER  TAP TAPE-C4  1   3   1
-----
NEXT-PAGE : +   (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED

```

4.6 Auswahl von BS2000-Dateien, Jobvariablen und Knotendateien

Bei den Grundfunktionen verarbeitet HSMS in der Regel eine Vielzahl von Dateien und Jobvariablen. Zur Auswahl der jeweils zu verarbeitenden Dateien und Jobvariablen bietet HSMS verschiedene Möglichkeiten.

4.6.1 Auswahl von BS2000-Dateien

BS2000-Dateien sind Dateien, die im BS2000 verwaltet und bearbeitet werden. Sofern im Folgenden nicht explizit unterschieden wird, umfasst „BS2000-Datei“ bei Dateien auf Net-Storage sowohl den Dateityp BS2000 als auch den Dateityp Node-File, der ab BS2000 OSD/BC V10.0 zusätzlich unterstützt wird.

BS2000-Dateien können ausgewählt werden durch:

- verschiedene Operanden bei den einzelnen HSMS-Anweisungen
- die HSMS-Anweisung SELECT-FILE-NAMES
- eine Eingabe im Dialog

4.6.1.1 Auswahl durch Operanden

Die zu verarbeitenden Dateien werden in der entsprechenden HSMS-Anweisung durch die Operanden FILE-NAMES und EXCEPT-FILE-NAMES ausgewählt. In der HSMS-Anweisung COPY-SAVE-FILE heißen die entsprechenden Operanden SELECT-FILES und EXCEPT-FILES. Die Auswahl kann durch weitere Operanden noch spezifisch eingeschränkt werden. Bei einigen HSMS-Anweisungen können die Dateien auf Wunsch noch einmal endgültig in einer Bildschirmmaske ausgewählt werden.

Die einfachste Art der Auswahl ist es, alle Dateien eines Benutzers oder – für den HSMS-Verwalter – alle Dateien des Systems mit einer einzigen HSMS-Anweisung zu bearbeiten:

```
FILE-NAMES/SELECT-FILES=*OWN/*ALL
```

Dies ist insbesondere bei Gesamtsicherungen sinnvoll, aber auch dann, wenn die Auswahl der Dateien nicht über den Namen erfolgen soll, sondern wenn die Auswahl aus der Menge aller Dateien nach bestimmten Kriterien erfolgen soll (siehe dazu die folgenden Abschnitte).

Bei der Angabe *OWN/*ALL werden Shared-Pubsets in vielen Fällen übergangen, wenn der eigene Rechner Slave ist. Näheres ist bei den jeweiligen HSMS-Anweisungen beschrieben.

Direkte Eingabe der Dateinamen

Bei allen HSMS-Anweisungen zur Dateibearbeitung können die Dateinamen direkt eingegeben werden. Dabei ist pro HSMS-Anweisung die Angabe von bis zu 20 Dateien möglich. Falls die Anzahl der Dateien größer ist, sollte die Eingabe über eine Datei erfolgen (siehe dazu den folgenden Abschnitt „Eingabe durch Dateien“).

Die Dateien können voll- und teilqualifiziert eingegeben werden, mit oder ohne Katalog- und Benutzerkennung.

Die Verwendung der Wildcard-Syntax ist möglich; diese ist im Handbuch „HSMS Bd. 2“ [1] im Abschnitt „Metasyntax“ beschrieben.

Bei der Auflösung von Wildcards in der Katalogkennung werden Shared-Pubsets, für die der eigene Rechner Slave ist, in vielen Fällen komplett übergangen. Näheres siehe in der Beschreibung der jeweiligen HSMS-Anweisung.

Durch die teilqualifizierte Eingabe von Katalog- oder Benutzerkennung lassen sich komplette Pubsets und Benutzerkennungen einfach bearbeiten.

Ein nicht-privilegiertes Benutzer kann auch diejenigen Dateien einer fremden Benutzerkennung bearbeiten, bei denen er Miteigentümer ist (siehe Handbuch „SECOS“ [16]).

Eingabe durch Dateien oder Bibliothekselemente

Die Dateinamen können durch eine Datei an HSMS übergeben werden, die satzweise die Namen der zu verarbeitenden Dateien enthält. Diese Datei muss eine SAM-Datei mit variabler Satzlänge sein. Alternativ kann auch ein Bibliothekselement vom Typ S verwendet werden. Eine Eingabedatei kann z.B. mit dem BS2000-Kommando

```
/SHOW-FILE-ATTRIBUTES . . . ,OUTPUT=<dateiname>(FORM-NAME=*FILE-NAME)
```

oder mit der HSMS-Anweisung SELECT-FILE-NAMES erstellt werden. Das Erstellen einer Datei mit der HSMS-Anweisung SELECT-FILE-NAMES ist auf "[Auswahl durch die HSMS-Anweisung SELECT-FILE-NAMES](#)" beschrieben.

Die Dateinamen können in der Datei voll- und teilqualifiziert angegeben werden; sie können mit oder ohne Katalog- und Benutzererkennung angegeben werden. Es sind nur Großbuchstaben erlaubt. Die Wildcard-Syntax (siehe Handbuch „HSMS Bd. 2“ [1], Abschnitt „Metasyntax“) darf verwendet werden, wobei aber folgende Einschränkung gilt:

Die Dateinamen in der Datei dürfen nicht mit einem Bindestrich beginnen, um sie von der Bearbeitung auszunehmen.

Der Name dieser Datei oder des Bibliothekselements muss beim Operanden FILE-NAMES bzw. EXCEPT-FILE-NAMES oder SELECT-FILES bzw. EXCEPT-FILES der betreffenden HSMS-Anweisung mit *FROM-FILE(...) bzw. *FROM-LIBRARY-ELEMENT(...) angegeben werden, z.B.:

```
FILE-NAMES/SELECT-FILES=*FROM-FILE(LIST-FILE-NAME=<dateiname>) bzw.  
*FROM-LIBRARY-ELEMENT(LIBRARY=<bibliothekname>,ELEMENT=<elementname>)
```

Die Datei oder die Bibliothek müssen bei einem nicht-privilegierten Benutzer in dessen Kennung liegen oder der Zugriff muss durch Miteigentümerschaft erlaubt sein.

Einschränkung durch Anweisungsoperanden

Bei vielen HSMS-Anweisungen bietet HSMS die Möglichkeit, die bei FILE-NAMES bestimmte Dateimenge noch durch Angabe weiterer Operanden gezielt einzuschränken: Dateien, die mit FILE-NAMES ausgewählt wurden, werden nur verarbeitet, wenn sie auch den durch andere Operanden festgelegten Bedingungen genügen.

Beispiel

```
//BACKUP-FILES FILE-NAMES=*ALL,SUPPORT=*PRIVATE-DISK(...)
```

Es werden nicht alle Dateien gesichert, sondern nur die Dateien, die auf einer (näher zu bestimmenden) Privatplatte liegen.

Die Standardwerte dieser einschränkenden Operanden sind allerdings stets so gewählt, dass durch sie die bei FILE-NAMES getroffene Auswahl nicht eingeschränkt wird.

Dateien auf Net-Storage auswählen oder ausschließen

Ab BS2000/OSD-BC V9.0 kann einem Pubset zusätzlicher Speicherplatz, den ein Net-Server bereitstellt, als Net-Storage zugeordnet werden. Damit können Dateien von Pubsets entweder auf lokalen Pubset-Platten oder remote auf einem Net-Storage liegen.

BS2000 OSD/BC ab V10.0 unterstützt auf Net-Storage neben dem Dateityp BS2000 auch den Dateityp Node-File.

Beim Sichern mit BACKUP-FILES kann in den Sicherungsoptionen (SAVE-OPTIONS) eingestellt werden, dass Daten vom Net-Storage-Dateien des Typs BS2000 nicht mitgesichert werden sollen. Die Katalogeinträge der Dateien werden unabhängig davon gesichert.

Net-Storage-Dateien vom Typ Node-File werden unabhängig von der Angabe im Operanden SAVE-NET-STOR-DATA immer mit Daten gesichert.

Beispiel

```
//BACKUP-FILES FILE-NAMES=*ALL, . . . ,
      SAVE-OPTIONS=*PAR (SAVE-NET-STOR-DATA=*NO)
```

Wenn die Sicherung im Operanden SUPPORT auf Dateien von gemeinschaftlichen Platten beschränkt wird, kann die Sicherung zusätzlich auf Dateien, die auf Net-Storage liegen, oder auf Dateien, die auf lokalen Pubset-Platten liegen, beschränkt werden.

Bei Net-Storage kann die Sicherung im Operanden FILE-TYPE zusätzlich auf den Dateityp BS2000 oder Node-File beschränkt werden. Voreingestellt ist die Sicherung aller Net-Storage-Dateien unabhängig vom Dateityp.

Beispiele

```
//BACKUP-FILES FILE-NAMES=*ALL, . . . ,
      SUPPORT=*PUBLIC-DISK (STORAGE-TYPE=*PUBLIC-SPACE) , . . .
```

Es werden nur Dateien von lokalen Pubset-Platten gesichert.

```
//BACKUP-FILES FILE-NAMES=*ALL, . . . ,
      SUPPORT=*PUBLIC-DISK (
        STORAGE-TYPE=*NET-STORAGE (VOLUME=(1#VBIG, 2#VBIG)) )
```

Es werden nur Dateien, die auf Net-Storage-Volumes 1#VBIG und 2#VBIG liegen, gesichert.

```
//BACKUP-FILES FILE-NAMES=*ALL, . . . ,
      SUPPORT=*PUBLIC-DISK (
        STORAGE-TYPE=*NET-STORAGE (VOLUME=(1#VBIG, 2#VBIG) ,
        FILE-TYPE=*BS2000) )
```

Es werden nur Dateien vom Typ BS2000-Datei, die auf Net-Storage-Volumes 1#VBIG und 2#VBIG liegen, gesichert. Net-Storage-Dateien vom Typ Node-File, die ggf. in benutzerspezifischen Verzeichnissen dieser Volumes existieren, werden nicht mitgesichert.

Diese Auswahloption ist auch verfügbar in den Anweisungen EXPORT-FILES, IMPORT-FILES und RESTORE-FILES (hier Operand ORIGINAL-SUPPORT bzw. NEW-SUPPORT).

Dateigenerationen auswählen

HSMS sichert sowohl Dateigenerationen als auch Dateigenerationsgruppen. Dateigenerationsgruppen können wie Dateien angegeben werden; es werden dann alle Generationen der Gruppe gesichert. Wenn eine bestimmte Generation bearbeitet werden soll, muss der Dateiname vollqualifiziert (mit Generationsnummer) angegeben werden.

4.6.1.2 Auswahl durch die HSMS-Anweisung SELECT-FILE-NAMES

Die HSMS-Anweisung SELECT-FILE-NAMES erzeugt eine Liste von Pfadnamen, die in den nachfolgenden Operanden zur Auswahl von BS2000-Dateien benutzt werden kann:

```
FILE-NAMES/SELECT-FILES=*SELECTED
```

Diese Liste wird standardmäßig in eine temporäre Datei (Tempfile) geschrieben, die nach der Task-Beendigung gelöscht wird.

Wahlweise kann diese Liste auch in eine katalogisierte Datei geschrieben und als Listdatei angegeben werden:

```
FILE-NAMES/SELECT-FILES=*FROM-FILE(LIST-FILE-NAME=<dateiname>)
```

Auf diese Weise kann die erzeugte Liste nicht nur während eines HSMS-Laufs, sondern beliebig oft verwendet werden.

Die Dateien können bei SELECT-FILE-NAMES ausgewählt werden:

- aus dem Katalog eines Pubsets bei der Vorbereitung einer Datensicherung, Archivierung, Verdrängung oder beim Exportieren
- aus einem Archivverzeichnis beim Restaurieren oder Zurückholen bzw. aus einem Verzeichnis beim Importieren.

Zusätzlich zum Operanden FILE-NAMES kann eine ganze Reihe von Selektionskriterien verwendet werden, um Dateien aus einem Katalog oder einem Archivverzeichnis auszuwählen.

Auswahl aus einem Katalog

Dateien aus einem Pubset-Katalog werden mit dem Operanden SELECT-FROM= *CATALOG ausgewählt. Auswahlkriterien sind:

- der Datenträger, auf dem sie stehen
- der Speichertyp (*PUBLIC-SPACE oder *NET-STORAGE)
- der Dateityp von Net-Storage-Dateien (*BS2000 oder *NODE-FILE)
- die Speicherebene, in der sie verwaltet werden
- der Zeitraum, seit dem auf diese Dateien nicht mehr zugegriffen wurde
- die Größe der Dateien
- die Sicherungsstufe (Backup-Klasse)

Auswahl aus einem Archiv

Dateien aus einem Archiv werden mit dem Operanden SELECT-FROM=*ARCHIVE ausgewählt. Auswahlkriterien sind:

- das Archiv, in dem sie verwaltet werden
- die Sicherungsversion, in der sie gesichert wurden
- das Freigabedatum der Dateien
- der Sicherungstyp der Dateien

Die Auswahl der Dateien im Dialog ist auch bei SELECT-FILE-NAMES möglich.

4.6.1.3 Auswahl im Dialog

Die Auswahl im Dialog ist nur für BS2000-Dateien und nur bei blockorientierten Datensichtstationen (z.B. 9750) möglich.

Sowohl bei SELECT-FILE-NAMES als auch bei einigen Aktionsanweisungen kann aus der Liste, die nach den angegebenen Kriterien erzeugt wurde, eine weitere Auswahl getroffen werden: Die Dateien der Liste werden bei DIALOG-FILE-SELECT=*YES in Bildschirmmasken präsentiert. Sie können durch Markieren einzeln oder schirmweise akzeptiert oder ausgeschlossen werden.

Beispiel

Ausgabe einer Bildschirmmaske für DIALOG-FILE-SELECT

```

ELECT-FILE-NAME (FROM-CATALOG)                               #FILES      = 27
STORAGE-LEVEL = ANY           UNUSED-DAYS = 7           MINIMUM-SIZE = NONE
SUPPORT       = ANY           BACKUP-CLASS = ANY         MAXIMUM-SIZE = NONE
-----
M  FILE-NAME                                     UNUSED      #PAGES  ST BC
X  :T:$TESTUSER.FILE.TEST.1                      8           31     S0 A
X  :T:$TESTUSER.FILE.TEST.3                      12          43     S0 A
X  :T:$TESTUSER.LIB.TEST.2                       13          523    S0 A
X  :T:$TESTUSER.LIB.TEST.5                       7           267    S0 A
X  :T:$TESTUSER.LIB.TEST.7                      23          1290   S0 A
X  :T:$TESTUSER.LIB.TEST.10                      9           101    S0 A
X  :T:$TESTUSER.PROC.TEST.3                      7            7     S0 A
X  :T:$TESTUSER.PROC.TEST.6                     21           3     S0 A
X  :T:$TESTUSER.PROG.TEST.1                     17           198    S0 A
X  :T:$TESTUSER.PROG.TEST.2                     11           353    S0 A
X  :T:$TESTUSER.PROG.TEST.2                     11           353    S0 A
X  :T:$TESTUSER.SYNTAX.TEST.GROUP               15           333    S0 A
X  :T:$TESTUSER.SYNTAX.TEST.USER               15           145    S0 A
X  :T:$TESTUSER.TEST.1                          12           232    S0 A
X  :T:$TESTUSER.TEST.12                        37           87     S0 A
X  :T:$TESTUSER.TEST.14                        34           102    S0 A
-----
NEXT-PAGE : __  (+, -, ++, --, E)   MARK :      (A: ALL, N: NONE)
    
```

Legende:

Rubrik – Werte	Bedeutung
M	Markierungsspalte (Zeichen = Datei ausgewählt, Leerzeichen = Datei wird nicht ausgewählt)
FILE-NAME	Pfadname der Datei
UNUSED	Anzahl der Tage seit dem letzten Zugriff auf die Datei
#PAGES	Größe der Datei in PAM-Seiten (FILE-SIZE)
ST – S0, S1, S2	Speicherebene (Storage-Level), auf dem die Datei steht – mögliche Speicherebenen

BC – A, B, C, D, E	Backup-Klasse der Datei – mögliche Backup-Klassen
-----------------------	--

Auswahl durch Markieren

Eine Datei gilt als ausgewählt, wenn sie in der Maske mit einem Zeichen in der Markierungsspalte (M-Spalte) markiert ist. Sie wird aus der Menge ausgeschlossen, wenn sie nicht markiert ist (Leerzeichen in der M-Spalte).

Bei einem Aufruf von DIALOG-FILE-SELECT werden die Dateien in FHS-Masken aufgelistet, sind jedoch nicht ausgewählt.

Durch Eingabe von „A“ in das MARK-Feld in der Fußzeile werden alle Dateien ausgewählt, auch die nicht auf dem Bildschirm ausgegebenen.

Es kann auch ein „N“ in das MARK-Feld eingegeben werden. Dann werden alle Dateien aus der Menge ausgeschlossen, d.h. alle Dateien werden in der Maske mit einem Leerzeichen in der Markierungsspalte ausgegeben. Anschließend können die gewünschten Dateien durch Markieren mit einem beliebigen Zeichen (mit Ausnahme des Leerzeichens) ausgewählt werden (Positivauslese). Durch Eingabe von „N“ oder „A“ werden alle bis dahin gemachten Eingaben hinfällig.

Entscheidend für die Auswahl ist der Stand bei Beendigung der Funktion durch Eingabe von „E“ in das NEXT-PAGE-Feld.

Positionieren mit Suchstring

In der Maske kann durch Eingabe von „+“, „-“ usw. geblättert werden. Es kann aber auch durch Eingabe eines in Hochkommata eingeschlossenen Suchstrings (bis zu 10 Zeichen) in das NEXT-PAGE-Feld auf bestimmte Dateien positioniert werden.

Als Positionierungskriterium kann ein voll- oder teilqualifizierter Pfadname ohne Wildcards oder ein Teil eines Pfadnamens (nur Katalog- und/oder Benutzerkennung und/oder Dateiname) angegeben werden. Nicht angegebene führende Pfadnamensteile werden mit den Teilen des Pfadnamens der ersten in der jeweiligen Maske ausgegebenen Datei ergänzt. Der Suchstring wird intern durch eine Wildcard ergänzt, sodass auch längere Pfadnamen gefunden werden. Wenn der angegebene (evtl. ergänzte) Pfadname nicht existiert, wird auf den Nächsten in der Sortierreihenfolge positioniert.

Wenn eine Katalogkennung eingegeben wird, wird beispielsweise auf die erste Datei mit dieser Katalogkennung bzw. mit der nächstgrößeren existierenden Katalogkennung positioniert.

Wenn eine Benutzerkennung eingegeben wird, wird auf eine Datei mit der gewünschten Benutzerkennung und mit der Katalogkennung der ersten in der aktuellen Maske gezeigten Datei positioniert. Die Benutzerkennung muss mit „\$“ und abschließendem Punkt eingegeben werden. Sonst macht HSMS einen secondary read auf die Standard-Benutzerkennung.

Auf einzelne Dateien kann positioniert werden, indem die ersten Zeichen des Namens ohne Punkt eingegeben werden. Gesucht wird dann eine Datei mit der Benutzer- und Katalogkennung der ersten angezeigten Datei und dem gewünschten Dateinamen.

Beispiel

Es soll auf eine Datei, die mit dem String 'FILED' beginnt, des Benutzers USERP auf dem Pubset M positioniert werden. Dies geschieht in mehreren Stufen:

SELECT-FILE-NAME (FROM-CATALOG)	#FILES - 1000	
STORAGE-LEVEL - ANY	UNUSED-DAYS - 0	MINIMUM-SIZE - NONE
SUPPORT - ANY	BACKUP-CLASS - ANY	MAXIMUM-SIZE - NONE

M	FILE-NAME	UNUSED	#PAGES	ST	BC
X	:A:\$USERA.FILEA.VERS1	0	3	SO	A
X	:A:\$USERA.FILEA.VERS2	0	3	SO	A

Position cursor to files of Pubset M:

NEXT-PAGE : ':m:' (+,-,++,--, E, 'STRING') MARK : (A: ALL, N: NONE)

SELECT-FILE-NAME (FROM-CATALOG)	#FILES - 1000	
STORAGE-LEVEL - ANY	UNUSED-DAYS - 0	MINIMUM-SIZE - NONE
SUPPORT - ANY	BACKUP-CLASS - ANY	MAXIMUM-SIZE - NONE

M	FILE-NAME	UNUSED	#PAGES	ST	BC
X	:M:\$USERF.FILEA.VERS1	0	3	SO	A
X	:M:\$USERF.FILEA.VERS2	0	3	SO	A

Position cursor to files of user ID USERP of Pubset M:

NEXT-PAGE : '\$userp.' (+,-,++,--, E, 'STRING') MARK : (A: ALL, N: NONE)


```

SELECT-FILE-NAME (FROM-CATALOG)          #FILES - 1000
STORAGE-LEVEL - ANY                      UNUSED-DAYS - 0    MINIMUM-SIZE - NONE
SUPPORT - ANY                            BACKUP-CLASS - ANY MAXIMUM-SIZE - NONE
-----
M FILE-NAME                               UNUSED #PAGES  ST BC
X :M:$USERP.FILEC.VERS1                   0      3  SO A
X :M:$USERP.FILEC.VERS2                   0      3  SO A
X :M:$USERP.FILED.VERS2                   0      3  SO A
X :M:$USERP.FILED.VERS3                   0      3  SO A

Positionieren auf Datei des Benutzers USERP, die FILED enthalten:

-----
NEXT-PAGE : 'FILED'      (+,-,++,--, E, 'STRING')  MARK : (A: ALL, N: NONE)

-----
SELECT-FILE-NAME (FROM-CATALOG)          #FILES - 1000
STORAGE-LEVEL - ANY                      UNUSED-DAYS - 0    MINIMUM-SIZE - NONE
SUPPORT - ANY                            BACKUP-CLASS - ANY MAXIMUM-SIZE - NONE
-----
M FILE-NAME                               UNUSED #PAGES  ST BC
X :M:$USERP.FILED.VERS2                   0      3  SO A
X :M:$USERP.FILED.VERS3                   0      3  SO A

```

Es kann also trotz der Kürze des Suchstrings durch sukzessive Eingabe von Katalog- und Benutzerkennung und Namensteilen auf die gewünschte Datei positioniert werden.

4.6.2 Auswahl von Jobvariablen

Für die Auswahl von Jobvariablen gibt es folgende Möglichkeiten:

- Beim Sichern und Exportieren von Sicherungsdateien gibt es die Anweisungsoperanden JV-NAMES und EXCEPT-JV-NAMES.
- Beim Kopieren von Sicherungsdateien gibt es die Anweisungsoperanden SELECT-JV und EXCEPT-JV.

Dabei können entweder alle Jobvariablen (*OWN/*ALL), keine Jobvariablen (*NONE) oder eine Liste von Jobvariablen (*FROM-FILE(...) bzw. *FROM-LIBRARY-ELEMENT(...)) angegeben werden.

Wenn mit *FROM-FILE(...) oder mit *FROM-LIBRARY-ELEMENT(...) eine Liste von Jobvariablen angegeben werden soll, muss der Benutzer eine SAM-Datei oder ein Bibliothekselement vom Typ S erstellen, die die Namen der zu bearbeitenden Jobvariablen enthalten:

- Der Benutzer kann diese Datei oder das Bibliothekselement außerhalb von HSMS selbst erstellen, z.B. mit dem EDT.
Pro Satz ist eine Jobvariable anzugeben. Die Wildcard-Syntax darf verwendet werden, wobei aber folgende Einschränkung gilt: Die Jobvariablen in der Datei dürfen nicht mit einem Bindestrich beginnen, um sie von der Bearbeitung auszunehmen.
- Der Benutzer kann mit der HSMS-Anweisung SELECT-JV-NAMES eine SAM-Datei mit den gewünschten Jobvariablenamen erstellen.

4.6.3 Auswahl von Knotendateien des BS2000-UFS

Knotendateien des BS2000-UFS oder Knoten-S0 können ausgewählt werden durch:

- verschiedene Operanden bei den einzelnen HSMS-Anweisungen
- die HSMS-Anweisung SELECT-NODE-FILES

4.6.3.1 Auswahl durch Operanden

Die zu verarbeitenden Dateien werden in der entsprechenden HSMS-Anweisung durch die Operanden PATH-NAMES und EXCEPT-PATH-NAMES ausgewählt. In der HSMS-Anweisung COPY-NODE-SAVE-FILE heißen die entsprechenden Operanden SELECT-PATHS und EXCEPT-PATHS. Die Auswahl kann durch weitere Operanden noch spezifisch eingeschränkt werden. Bei einigen HSMS-Anweisungen können die Dateien auf Wunsch noch einmal endgültig in einer Bildschirmmaske ausgewählt werden.

Direkte Eingabe der Dateinamen

Der Benutzer kann bei allen HSMS-Anweisungen, die Knotendateien bearbeiten, den Pfadnamen der Knotendateien angeben, die ausgewählt werden sollen. Der Pfadname darf Wildcards enthalten. In einer HSMS-Anweisung darf aber nur ein einziger Pfadname vorkommen. Wenn mehrere Knotendateien angegeben werden müssen, sollte die Eingabe über eine Datei erfolgen (siehe dazu den Abschnitt „Eingabe durch Dateien oder Bibliothekselemente“).

Auf die gleiche Weise kann der Benutzer die Pfadnamen von Knotendateien angeben, die in der Auswahl von der Bearbeitung ausgeschlossen werden sollen.

Ein nicht-privilegiertes Benutzer darf nur auf Knotendateien verweisen, die auf dem lokalen BS2000-UFS liegen. Der HSMS-Verwalter darf auf alle Knotendateien verweisen, die auf dem lokalen BS2000-UFS oder auf einer fernen Workstation liegen.

Wenn Dateien und Dateiverzeichnisse von der angegebenen Hierarchieebene ausgewählt werden sollen, kann der Benutzer außerdem festlegen, ob die Auswahl bis zur Grenze des aktuellen Dateisystems gehen soll oder bis zum Ende des Dateibaums.

Damit der Benutzer eines der nummerierten Ergebnisse in der folgenden Grafik erhält, muss er für den Pfadnamen und den Operanden SELECTION-BOUNDARY die Angaben machen, die in der Tabelle nach der Grafik eingetragen sind:

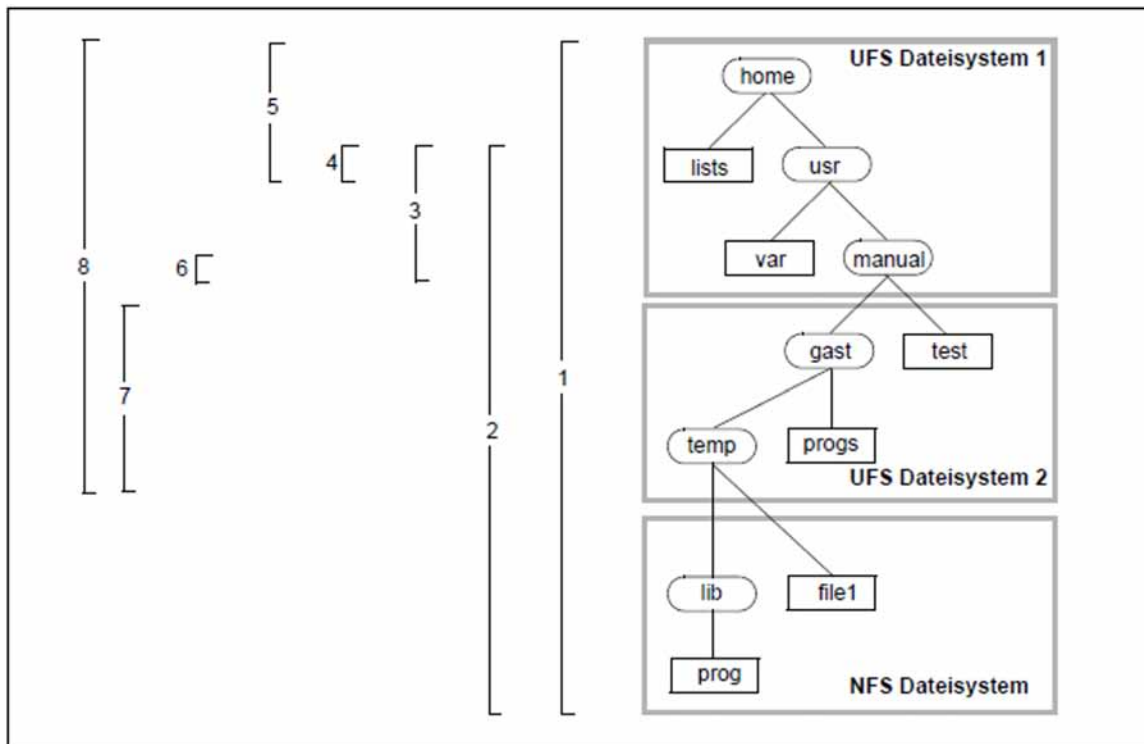


Bild 18: Umfang der Sicherung von Knotendateien

Nummer	PATH-NAMES=	SELECTION-BOUNDARY=
1	/home oder	*ALL-FILE-SYSTEMS
	/home*	*ALL-FILE-SYSTEMS
2	/home/*	*ALL-FILE-SYSTEMS
3	/home/*	*CURRENT-FILE-SYSTEM
4	/home/*	*SPECIFIED-PATHS
5	/home oder	*ALL-FILE-SYSTEMS
	/home*	Zusätzliche Angabe: EXCEPT-PATH-NAMES=/home/usr/*
6	/home/usr/*	*CURRENT-FILE-SYSTEM
7	/home/usr/manual/*	*ALL-FILE-SYSTEMS
8	/home	*ALL-LOCAL-FILE-SYSTEMS

Bei der oben dargestellten Auswahlstrategie wird der Umfang „*ALL-FILE-SYSTEMS“ zum Sichern aller fernen Rechner empfohlen, besonders wenn mehrere Dateisysteme unter der „root“ („/“) des Rechners eingehängt sind.

Auf die Knotendateien einer passiven Workstation wird mit dem Softwareprodukt NFS zugegriffen. Deshalb werden die Namen von fernen Dateien intern folgendermaßen erweitert:

```
/HSMS/<node-id>/<pathname>
```

Die Gesamtlänge des so erweiterten Dateinamens darf 1023 Zeichen nicht überschreiten. Diese Begrenzung ist besonders bei Restore-Vorgängen, bei denen Dateinamen umbenannt werden, zu beachten. Sie verhindert auch den Zugriff auf ferne Knotendateien, deren Namen mit dem Präfix „/HSMS/<node-id>“ die Länge von 1023 Zeichen überschreiten, sogar wenn der Benutzer einen kürzeren Dateinamen angibt (Dateiverzeichnis-Ebene und rekursive Auswahl, Wildcards, ...).

Eingabe durch Dateien oder Bibliothekselemente

Die Dateinamen können durch eine BS2000-Datei oder ein Bibliothekselement vom Typ S an HSMS übergeben werden. Diese enthalten satzweise die Pfadnamen der zu verarbeitenden Knotendateien oder Knotendateigruppen.

Die BS2000-Datei muss eine SAM-Datei mit variabler Satzlänge sein, die z.B. mit dem BS2000-Editor EDT oder mit der HSMS-Anweisung SELECT-NODE-FILES erstellt werden kann. Das Erstellen einer Datei mit der HSMS-Anweisung SELECT-NODE-FILES ist auf "[Auswahl durch die HSMS-Anweisung SELECT-NODE-FILES](#)" beschrieben.

Die Pfadnamen können in der Datei folgendermaßen angegeben werden:

Nr.	Pfad	verweist auf	Zugriffsberechtigung	
			HSMS-Verwalter	Nicht-privil. Benutzer
1.	/<dir>/... wobei <dir> != „HSMS“	lokales BS2000-UFS	X	X
2.	<node-id>:/...	angegebene Workstation	X	-
3.	/ * ...	lokales BS2000-UFS	X	X
4.	* : /...	alle Workstations	X	-

Die Pfadnamen, die in der mit FROM-FILE angegebenen Datei eingetragen sind, dürfen nicht mit einem Bindestrich beginnen, um sie von der Bearbeitung auszunehmen.

Der Name dieser Datei bzw. des Bibliothekselements muss beim Operanden PATH-NAMES oder SELECT-PATHS der betreffenden HSMS-Anweisung mit *FROM-FILE(...) bzw. *FROM-LIBRARY-ELEMENT(...) angegeben werden:

```
PATH-NAMES/SELECT-PATHS=*FROM-FILE(LIST-FILE-NAME=<dateiname>) bzw.
*FROM-LIBRARY-ELEMENT(LIBRARY=<bibliotheksname>,ELEMENT=<elementname>)
```

Die Pfade werden in der Reihenfolge bearbeitet, wie sie in der bei *FROM-FILE angegebenen Datei eingetragen sind. Deshalb kann der Benutzer die Performance beim Sichern von Knotendateien verbessern, wenn er parallele Verarbeitung einsetzt (entweder durch Multiplexbetrieb oder durch Verwendung mehrerer Laufwerke).

Dazu ein Beispiel:

Wenn ein Netzwerk aus mehreren getrennten, unabhängigen Teilnetzen besteht, kann es zweckmäßig sein, Workstations von verschiedenen Teilnetzen zu mischen, um die Datenpfade zu trennen, die von den ARCHIVE-Subtasks benutzt werden.

Konkret könnte folgende Situation vorliegen: Die Workstations W1 und W2 gehören zu einem Teilnetz S1, das über HNC1 mit dem Server S verbunden ist. Weiter gehören die Workstations W3 und W4 zu einem Teilnetz S2, das über HNC2 ebenfalls mit dem Server S verbunden ist. Die vier Workstations sollen mit zwei ARCHIVE-Subtasks gesichert werden (mit oder ohne Multiplexbetrieb).

Wenn die Workstations in der Reihenfolge W1, W2, W3, W4 in der Datei eingetragen sind, arbeiten die beiden ARCHIVE-Subtasks gleichzeitig auf demselben Teilnetz, was nicht optimal ist. Die bessere Alternative ist – mit Hinblick auf die Trennung von Datenpfaden – die Reihenfolge W1, W3, W2, W4. Dieselben Überlegungen sind auch auf Pfadenebene gültig (mehrere Pfade pro Workstation, mehrere Workstations).

Die Reihenfolge, in der die Workstations in der bei *FROM-FILE angegebenen Datei eingetragen sind, ist also sehr wichtig. Deshalb sollte der Aufbau dieser Datei sorgfältig überlegt werden. Außerdem sollte zwischen den definierten Pfaden ein ausgewogenes Verhältnis in Bezug auf das Datenvolumen bestehen.

Siehe auch den Abschnitt „Multiplexbetrieb“ im ["Parallele und serielle Verarbeitung in ARCHIVE"](#).

4.6.3.2 Auswahl durch die HSMS-Anweisung SELECT-NODE-FILES

HSMS stellt die Anweisung SELECT-NODE-FILES zur Verfügung, um die Pfadnamen von Knoten auswählen zu können. Diese HSMS-Anweisung erzeugt eine Liste von Pfadnamen, die in den nachfolgenden Operanden zur Auswahl von Knotendateien benutzt werden kann:

```
PATH-NAMES/SELECT-PATHS=*SELECTED
```

Diese Liste wird standardmäßig in eine temporäre Datei geschrieben, die nach der Task-Beendigung gelöscht wird. Wahlweise kann diese Liste auch in eine katalogisierte Datei geschrieben und als Listdatei angegeben werden:

```
PATH-NAMES/SELECT-PATHS=*FROM-FILE(LIST-FILE-NAME=<dateiname>)
```

Auf diese Weise kann die erzeugte Liste nicht nur während eines HSMS-Laufs, sondern beliebig oft verwendet werden.

Mit SELECT-NODE-FILES können die Knotendateien beim Restaurieren aus einem Archivverzeichnis ausgewählt werden. Zusätzlich zum Operanden PATH-NAMES kann eine ganze Reihe von Selektionskriterien verwendet werden, um Knotendateien aus einem Archivverzeichnis auszuwählen.

Auswahl aus einem Archiv

Dateien aus einem Archiv werden mit dem Operanden SELECT-FROM=*ARCHIVE ausgewählt. Auswahlkriterien sind:

- das Archiv, in dem sie verwaltet werden
- die Sicherungsversion, in der sie gesichert wurden
- das Freigabedatum der Dateien
- der Sicherungstyp der Dateien

4.7 Auftragsbehandlung (Requests)

Der folgende Abschnitt enthält Informationen über:

- Aktionsanweisungen, Aufträge und HSMS-Servertasks
- Konsistenzprüfung
- Steuerung der Auftragszeit
- Steuerung der Auftragsbearbeitung
- Auftragsverwaltung
- Auftragsverwaltung bei Shared-Pubsets
- Aufträge wiederherstellen nach einem Host-Ausfall (Recovery)
- Prioritäten für die Auftragsverarbeitung vergeben
- Sammelaufträge
- Asynchrone und synchrone Verarbeitung
- Gerätereservierung
- Parallele und serielle Verarbeitung

4.7.1 Aktionsanweisungen, Aufträge und HSMS-Servertasks

Eine Reihe von HSMS-Anweisungen führen zu Ein-/Ausgaben auf Benutzer- und Sicherungsdateien, werden in einer HSMS-Subtask (nicht in der Aufrufer-Task) ausgeführt und erfordern eine Aktion des Softwareprodukts ARCHIVE. Diese HSMS-Anweisungen sind die Aktionsanweisungen:

ARCHIVE-FILES (ARF)	Dateien archivieren
ARCHIVE-NODE-FILES (ANF)	Knotendateien archivieren
BACKUP-FILES (BCF)	Dateien und Jobvariablen sichern
BACKUP-FILE-VERSIONS (BFV)	Dateien sichern
BACKUP-NODE-FILES (BNF)	Knotendateien sichern
COPY-EXPORT-SAVE-FILE (CES)	Sicherungsdatei kopieren
COPY-NODE-SAVE-FILE (CNF)	Knoten-Sicherungsdateien kopieren
COPY-SAVE-FILE (CSF)	Sicherungsdateien kopieren
EXPORT-FILES (EXF)	Dateien und Jobvariablen exportieren
IMPORT-FILES (IMF)	Dateien und Jobvariablen importieren
MIGRATE-FILES (MGF)	Dateien verdrängen
MOVE-SAVE-FILES (MSF)	Sicherungsdateien verschieben
RECALL-MIGRATED-FILES (RMF)	Dateien zurückholen
REORGANIZE-VERSION-BACKUP (RVB)	Versions-Backup reorganisieren
REPAIR-CATALOG-BY-RESTORE (RCR)	Inkonsistenzen bei migrierten Dateien durch RESTORE aufheben
REPLACE-SAVE-FILE-BY-RESTORE (RFR)	Sicherungsdatei im Migrationsarchiv durch RESTORE ersetzen
RESTORE-FILES (RSF)	Dateien und Jobvariablen restaurieren
RESTORE-LIBRARY-ELEMENTS (RLE)	Elemente einer Bibliotheksdatei restaurieren
RESTORE-NODE-FILES (RNF)	Knotendateien restaurieren
UPDATE-EXPORT-SAVE-FILE (UES)	Sicherung aktualisieren

Bei jeder dieser Aktionsanweisungen lässt sich über den Operanden OPERATION-CONTROL die Bearbeitung der Aufträge steuern.

Für alle Ein- und Ausgaben, die durch Aktionsanweisungen angestoßen wurden, erzeugt HSMS einen Auftrag (Request). Diese Aufträge werden durch eigens dazu dienende Benutzeraufträge (Tasks) abgearbeitet, die sog. HSMS-Servertasks. Außer beim Datentransfer laufen die HSMS-Servertasks unter der Kennung TSOS mit dem Jobnamen „HSMSSERV“; sie sind ständig während der HSMS-Session vorhanden.

Diese HSMS-Servertasks rufen ARCHIVE auf; ARCHIVE wiederum erzeugt zur Bearbeitung der Ein- und Ausgaben ARCHIVE-Subtasks. Die ARCHIVE-Subtasks erhalten den Jobnamen gemäß der Angabe im Operanden REQUEST-NAME (siehe [Abschnitt „Steuerung der Auftragsbearbeitung \(Operation-Control\)“](#)).

4.7.2 Steuerung der Auftragszeit

Unter Steuerung der Auftragszeit versteht man das Verfahren, wie ein Auftrag, der von einem Benutzertask erzeugt wird, an einen HSMS-Servertask weitergereicht wird. Bei der Steuerung der Auftragszeit müssen – außer dem Weiterreichen eines Auftrags von einem Task zu einem anderen Task – beispielsweise folgende Gegebenheiten berücksichtigt werden:

- Serielle Verarbeitung von Aufträgen, die für dasselbe Archiv erteilt wurden:
Einige Aufträge erfordern alleinigen Zugriff auf das angegebene Archiv.
- Serielle Verarbeitung von Aufträgen, die in dieselbe Sicherungsdatei eines Archivs schreiben:
Da hierbei dieselben Datenträger betroffen sind, ist unbedingt serielle Verarbeitung erforderlich.
- Optimierung wie z.B. Bandbehandlung durch Sammelaufträge.
- Parameter für eine Bandverarbeitungszeit, die den Bandzugriff (Speicherebene S2) steuern.

Es gibt Aufträge, die nicht seriell bearbeitet werden müssen. Dies sind Aufträge zum Restaurieren und Zurückholen von Dateien sowie Aufträge, die eine Sicherungsdatei auf S1 erstellen. Solche Aufträge werden vollständig parallel bearbeitet, wenn sie keine Dateien von S2 oder von derselben SFID derselben Magnetbandkassette restaurieren oder zurückholen. Der erste freie HSMS-Servertask erhält den Auftrag und bearbeitet ihn.

Aufträge, die Sicherungen oder Knotensicherungen betreffen, können nur die letzte Sicherungsdatei des betreffenden Archivs fortschreiben oder eine neue Sicherungsdatei erzeugen, da die Reihenfolge der Sicherungsdateien (SFID's) und Sicherungsversionen (SVID's) gewährleistet werden muss. Beim Versuch eine andere als die letzte Sicherungsdatei fortzuschreiben wird eine neue Sicherungsdatei angelegt. Solche Aufträge werden vollständig seriell bearbeitet. Wenn ein Servertask den Auftrag eines seriell zu bearbeitenden Archivs aufnimmt, bekommt der Servertask alle Aufträge, die dieses Archiv betreffen. Der Servertask ruft für all diese Aufträge ARCHIVE auf. ARCHIVE stellt die serielle Bearbeitung dieser Aufträge sicher.

Aufträge, die die anderen Archivarten betreffen, werden folgendermaßen bearbeitet:

- parallel, wenn sie in verschiedene Sicherungsdateien schreiben.
- seriell, wenn sie in dieselbe Sicherungsdatei schreiben.

Wenn ein Servertask den Auftrag eines solchen „SFID-seriellen“ Archivs aufnimmt, sperrt der Servertask die betroffene Ausgabe-Sicherungsdatei. Alle anderen Aufträge, die dasselbe Archiv und dieselbe Sicherungsdatei betreffen, werden vom selben Servertask bearbeitet; dieser stellt die korrekte serielle Bearbeitung sicher.

Die Anzahl der Ausgabe-Sicherungsdateien, die parallel bearbeitet werden können, ist physikalisch auf acht verschiedene Sicherungsdateien pro Archiv beschränkt.

Wenn ein Servertask mehrere Aufträge aufnimmt, kann er einen Sammelauftrag erstellen, um die Bandbearbeitung zu optimieren (siehe [Abschnitt „Sammelaufträge“](#)).

Die folgenden Tabellen enthalten die Bearbeitung für jede mögliche Auftragskombination, die für dasselbe Archiv erteilt wird:

	BCF	RSF / RLE	ARF	BFV	MGF	RMF	CSF/MSF	RVB
BCF	parallel	parallel	nicht relevant	nicht relevant	nicht relevant	nicht relevant	parallel	nicht relevant
RSF / RLE		parallel	parallel	parallel	parallel	parallel	parallel	parallel
ARF			parallel	nicht relevant	nicht relevant	nicht relevant	parallel	nicht relevant
BFV				seriell (1)	nicht relevant	nicht relevant	nicht relevant	seriell (2)
MGF					parallel	parallel	parallel	nicht relevant
RMF						parallel	parallel	nicht relevant
CSF /MSF							parallel	nicht relevant
RVB								seriell (2)

Tabelle 1: Bearbeitung von DVS-Aufträgen auf S1

BCF BACKUP-FILES

RSF RESTORE-FILES

RLE RESTORE-LIBRARY-ELEMENTS

BFV BACKUP-FILE-VERSIONS

ARF ARCHIVE-FILES

MGF MIGRATE-FILES

RMF RECALL-MIGRATED-FILES

CSF COPY-SAVE-FILE

MSF MOVE-SAVE-FILES

RVB REORGANIZE-VERSION-BACKUP

- (1) Bei der Anweisung BACKUP-FILE-VERSIONS muss die Reihenfolge der Dateiversionen garantiert werden. Daher ist keine parallele Verarbeitung möglich.
- (2) Während der Reorganisation eines Versions-Backup-Archivs sind andere Schreibaufträge verboten. Daher ist keine parallele Verarbeitung möglich.

	BCF	RSF / RLE	ARF	BFV	MGF	RMF	CSF/MSF	RVB
BCF	seriell (1)	parallel	nicht relevant	nicht relevant	nicht relevant	nicht relevant	seriell (3)	nicht relevant
RSF / RLE		parallel	parallel	parallel	parallel	parallel	durch SFID (2)	parallel
ARF			durch SFID (2)	nicht relevant	nicht relevant	nicht relevant	durch SFID (2)	nicht relevant
BFV				seriell (4)	nicht relevant	nicht relevant	nicht relevant	seriell (5)
MGF					durch SFID (2)	parallel	durch SFID (2)	nicht relevant
RMF						parallel	parallel	nicht relevant
CSF / MSF							durch SFID (2)	nicht relevant
RVB								seriell (5)

Tabelle 2: Bearbeitung von DVS-Aufträgen auf S2

BCF BACKUP-FILES

RSF RESTORE-FILES

RLE RESTORE-LIBRARY-ELEMENTS

BFV BACKUP-FILE-VERSIONS

ARF ARCHIVE-FILES

MGF MIGRATE-FILES

RMF RECALL-MIGRATED-FILES

CSF COPY-SAVE-FILE

MSF MOVE-SAVE-FILES

RVB REORGANIZE-VERSION-BACKUP

- (1) Bei einer Sicherung oder Knotensicherung muss die Reihenfolge der SFIDs und SVIDs für Teilsicherungen sichergestellt werden. Deshalb ist keine parallele Bearbeitung möglich.
- (2) Aufträge, die dieselbe Sicherungsdatei betreffen, werden seriell bearbeitet. Aufträge, die verschiedene Sicherungsdateien betreffen, werden parallel bearbeitet.
- (3) Bei einer COPY-SAVE-FILE-Anweisung für ein Backup- oder Knoten-Backup-Archiv muss die Reihenfolge der SFIDs und SVIDs sichergestellt werden. Deshalb ist keine parallele Bearbeitung möglich.

- (4) Bei der Anweisung `BACKUP-FILE-VERSIONS` muss die Reihenfolge der Dateiversionen garantiert werden. Deshalb ist keine parallele Bearbeitung möglich.
- (5) Während der Reorganisation eines Versions-Backup-Archivs sind andere Schreibaufträge verboten, sodass die Konsistenz des Archivverzeichnisses garantiert wird. Daher ist keine parallele Verarbeitung möglich.

	BNF	RNF	ANF	CNF/MSF
BNF	seriell (1)	parallel	nicht relevant	seriell (3)
RNF		parallel	parallel	parallel
ANF			durch SFID (2)	durch SFID (2)
CNF/MSF				durch SFID (2)

Tabelle 3: Steuerung von Knoten-Aufträgen

BNF `BACKUP-NODE-FILES`

RNF `RESTORE-NODE-FILES`

ANF `ARCHIVE-NODE-FILES`

CNF `COPY-NODE-SAVE-FILE`

MSF `MOVE-SAVE-FILES`

- (1) Bei einer Sicherung oder Knotensicherung muss die Reihenfolge der SFIDs und SVIDs für Teilsicherungen sichergestellt werden. Deshalb ist keine parallele Bearbeitung möglich.
- (2) Aufträge, die dieselbe Sicherungsdatei betreffen, werden seriell bearbeitet. Aufträge, die verschiedene Sicherungsdateien betreffen, werden parallel bearbeitet.
- (3) Bei einer `COPY-NODE-FILE`-Anweisung für ein Backup- oder Knoten-Backup-Archiv muss die Reihenfolge der SFIDs und SVIDs sichergestellt werden. Deshalb ist keine parallele Bearbeitung möglich.

Bei einem `COPY-SAVE-FILE`-Auftrag sind zwei Archive beteiligt; die Serialisierung findet nach dem Zielarchiv statt. Das Verzeichnis des Quellarchivs bleibt während der Ausführung des Kopierauftrags geöffnet und kann während dieser Zeit andere Aufträge (wie z.B. Restore- oder Backup-Aufträge) für dasselbe Archiv blockieren.

Achtung

Parallel zu bearbeitende Aufträge können unter Umständen unerwartete Ergebnisse zur Folge haben. Deshalb ist Folgendes besonders zu beachten:

- Ein Restore-Auftrag für Dateien, die gerade von einem Backup-Auftrag bearbeitet werden, kann entweder den vorherigen Stand der Dateien von der vorangegangenen Sicherung restaurieren oder aber den aktuellen Stand der Dateien von der Sicherung, die gerade läuft.
- Nach dem Import eines SM-Pubsets oder dem Start des HSMS-Subsystems kann es vorkommen, dass beim Reaktivieren von früher angenommenen Aufträgen die ursprüngliche Reihenfolge der Auftragsannahme nicht berücksichtigt wird.
- Die parallele Bearbeitung von Aufträgen – hauptsächlich beim Restaurieren – gestattet verschiedenen Servertasks, dieselbe Sicherungsdatei zu benutzen. Wegen der Reservierung der Datenträger werden die Aufträge aber in Wirklichkeit seriell bearbeitet.

- Wenn Sicherungsdateien mit der MODIFY-ARCHIVE-Anweisung bereinigt werden, ist ein exklusiver Zugriff auf das Archivverzeichnis notwendig. Deshalb wird eine solche Bereinigung zurückgewiesen, wenn andere Aufträge auf das Archiv zugreifen. Umgekehrt wird die Ausführung von Aufträgen zurückgestellt, solange ein Archiv bereinigt wird.

4.7.3 Steuerung der Auftragsbearbeitung (Operation-Control)

Zur Steuerung der Auftragsbearbeitung dient der Operand OPERATION-CONTROL. Die Werte sind teilweise durch die Archivdefinition voreingestellt (außer beim Datentransfer). Von diesen Werten kann aber bei der Aktionsanweisung, die den Auftrag erzeugt, abgewichen werden.

PARALLEL-RUNS

Anzahl der gleichzeitig ablaufenden Sicherungstasks (ARCHIVE-Subtasks) für einen Sicherungsauftrag. Damit können, vor allem bei Systemsicherungen, die Sicherungszeiten erheblich verkürzt werden, indem HSMS die Dateien verschiedener Benutzerkennungen auf verschiedene Bandgeräte sichert.

Wenn bei Operationen mit den Anweisungen BACKUP-NODE-FILES und RESTORE-NODE-FILES die Vor-/Nachbearbeitung aktiviert ist (Operand PRE-POST-PROCESSING), warten die ARCHIVE-Subtasks das Ergebnis der Vorbearbeitung ab, bevor sie den Datenaustausch mit der Workstation beginnen. Dadurch verschlechtert sich allerdings die Performance.

Wenn kein Multiplexbetrieb eingeschaltet ist (siehe [Abschnitt „Multiplexbetrieb“](#)), muss pro Task beim Schreiben auf S2 oder Lesen von S2 ein Bandgerät zur Verfügung stehen, damit die Verarbeitung beschleunigt wird (zu den Bedingungen siehe den [Abschnitt „Parallele und serielle Verarbeitung in ARCHIVE“](#) und im Handbuch „ARCHIVE“ [2]).

Wenn Multiplexbetrieb eingeschaltet ist, berechnet das System die Anzahl der ARCHIVE-Subtasks. Die Anzahl der Bandgeräte wird der Benutzeranweisung entnommen (siehe [Abschnitt „Multiplexbetrieb“](#)).

Wenn Sicherungsdateien von S2 nach S2 oder automatisch in ein Schattenarchiv dupliziert werden, müssen zwei Bandgeräte pro Task zur Verfügung stehen.

Wenn Sicherungsdateien auf Net-Storage geschrieben werden, muss eine entsprechende Anzahl von Net-Storage-Volumes angegeben werden. Diese Angabe ist abhängig von der Anzahl der gleichzeitig ablaufenden Sicherungstasks.

Bei jedem Lesen einer Sicherungsdatei (Restaurieren, Zurückholen, Duplizieren) können nicht mehr ARCHIVE-Subtasks ablaufen, als beim Schreiben in diese Sicherungsdatei verwendet wurden.

Wenn beim Lesen einer Sicherungsdatei weniger Parallelläufe verwendet werden als beim Schreiben in diese Sicherungsdatei, kann es vorkommen, dass die Bänder zur Bandanfangsmarke zurückgespult werden.

WRITE-CHECKPOINTS

Bestimmt, ob während der Verarbeitung durch ARCHIVE Wiederaufsetzpunkte in die ARCHIVE-Checkpointdatei geschrieben werden, die bei einem Abbruch (Status INTERRUPTED) einen späteren Wiederanlauf (Restart) ermöglichen (siehe [Abschnitt „Aufträge wiederanlaufen lassen \(Restart\)“](#)).

OPERATOR-INTERACTION

Regelt die Behandlung von Meldungen, die eine Antwort des Operators erfordern. Wenn die Meldungen nicht am Bedienplatz ausgegeben werden sollen, führt HSMS eine Standardbehandlung durch (siehe Handbuch „ARCHIVE“ [2], PARAM-Anweisung).

REQUEST-NAME

Mit dem Operanden REQUEST-NAME kann einem Auftrag ein symbolischer Name gegeben werden. Unter diesem Namen kann der Auftrag in den nachfolgenden HSMS-Anweisungen angesprochen werden. HSMS versieht den Namen intern mit der Benutzerkennung des Aufrufers und einem Zeitstempel. Der Name muss also nicht eindeutig sein. In den HSMS-Anweisungen zur Auftragsverwaltung können gegebenenfalls mehrere Aufträge unter demselben Namen angesprochen werden.

Der Auftragsname kann als HSMS-Jobname angesehen werden. Er wird auch als Jobname für ARCHIVE-Subtasks verwendet.

Wenn der Benutzer keinen Auftragsnamen vergibt, erhalten die Aufträge Standardnamen, die durch ein festes Kürzel und die TSN (Task-Sequence-Number) des Benutzerauftrags gebildet werden.

Aktionsanweisung	Standardname des Auftrags
ARCHIVE-FILES	ARF#yyyy
ARCHIVE-NODE-FILES	ANF#yyyy
BACKUP-FILES	BCF#yyyy
BACKUP-FILE-VERSIONS	BFV#yyyy
BACKUP-NODE-FILES	BNF#yyyy
COPY-EXPORT-SAVE-FILE	CES#yyyy
COPY-NODE-SAVE-FILE	CNF#yyyy
COPY-SAVE-FILE	CSF#yyyy
EXPORT-FILES	EXF#yyyy
IMPORT-FILES	IMF#yyyy
MIGRATE-FILES	MGF#yyyy
MOVE-SAVE-FILES	MSF#yyyy
RECALL-MIGRATED-FILES	RMF#yyyy
REPAIR-CATALOG-BY-RESTORE	RCR#yyyy
REORGANIZE-VERSION-BACKUP	RVB#yyyy
REPLACE-SAVE-FILE-BY-RESTORE	RFR#yyyy
RESTORE-FILES	RSF#yyyy
RESTORE-LIBRARY-ELEMENTS	RLE#yyyy

RESTORE-NODE-FILES	RNF#yyyy
UPDATE-EXPORT-SAVE-FILE	UES#yyyy

Tabelle 4: Aktionsanweisungen und Standardnamen von Aufträgen

REQUEST-DESCRIPTOR

Ein Auftragserteiler kann mit diesem Operanden einen beliebigen Text angeben, der einen Auftrag näher beschreibt. Dieser Text wird auf der Systemkonsole ausgegeben, wenn der Auftrag gestartet wird. Er kann mit der HSMS-Anweisung SHOW-REQUESTS angezeigt werden.

Mit dem Operanden REQUEST-DESCRIPTOR kann der Auftragserteiler dem Systemverwalter auf einfache Weise Informationen über einen Auftrag senden. Der Systemverwalter kann dann für besondere Aufträge geeignete Maßnahmen ergreifen.

Allerdings ist es nicht möglich, auf diesen Operanden als Auswahlkriterium in nachfolgenden Anweisungen Bezug zu nehmen, z.B. in den HSMS-Anweisungen DELETE-REQUESTS, RESTART-REQUESTS oder SHOW-REQUESTS.

EXPRESS-REQUEST

Der HSMS-Verwalter kann seine Aufträge als Expressaufträge starten, die unabhängig von den Aufträgen nicht-privilegierter Benutzer gesteuert werden können. Expressaufträge werden während der für sie gesondert festgelegten Bandverarbeitungszeiten reaktiviert. Die Bandverarbeitungszeiten für Expressaufträge sind von den normalen Bandverarbeitungszeiten für Schreib-/Leseaufträge getrennt. Deshalb sollten alle dringenden Aufträge als Expressaufträge gestartet und Bandverarbeitungszeiten für Expressaufträge festgelegt werden, die früher oder öfter als normale Bandverarbeitungszeiten öffnen.

Zusätzlich werden Expressaufträge mit einer höheren Priorität behandelt als normale Aufträge, die für dasselbe Archiv erteilt werden (siehe [Abschnitt „Prioritäten für die Auftragsbearbeitung vergeben“](#)).

WAIT-FOR-COMPLETION

Der Benutzer kann festlegen, ob er warten will, bis HSMS die eingegebene Anweisung im Hintergrund bearbeitet hat, oder ob er sofort die nächste Eingabe machen kann (siehe [Abschnitt „Asynchrone und synchrone Verarbeitung“](#)).

4.7.4 Auftragsverwaltung

Folgende HSMS-Anweisungen stehen für die Auftragsverwaltung zur Verfügung:

- DELETE-REQUESTS: Löschen von Aufträgen aus der Auftragsdatei
- RECOVER-REQUESTS: Wiederherstellen von Aufträgen in SM-Umgebung
- RESTART-REQUESTS: Wiederanlaufen lassen von unterbrochenen Aufträgen
- SHOW-REQUESTS: Ausgeben von Aufträgen in die Auftragsdatei

Die Verwaltungsfunktionen für SF- und SM-Umgebungen sind gleich. Für SM-Umgebungen stehen zusätzlich Funktionen für den Wiederanlauf und das Wiederherstellen von Aufträgen zur Verfügung. Die Verwaltungsfunktionen sind nachstehend beschrieben.

- Aufträge für SF-Umgebungen werden in der globalen HSMS-Auftragsdatei ausschließlich von dem Host verwaltet, auf dem der Home-Pubset läuft.

Für Aufträge, die nur SF-Pubsets betreffen, ist die Auftragsverwaltung für den Host rein lokal. Für Aufträge, die an einem Shared-SF-Pubset erteilt werden, ist die Auftragsverwaltung komplexer: Aufträge, die an einem Slave-Shareer erteilt werden, werden zur Bearbeitung in der Regel an den Master-Shareer geschickt. Nähere Informationen dazu finden Sie im [Abschnitt „Auftragsverwaltung bei Shared-Pubsets“](#).

- Aufträge für SM-Pubsets werden in der SM-Pubset-spezifischen Auftragsdatei verwaltet. Ein SM-Pubset kann, da er ein geschlossener Container ist, von Host zu Host transportiert und zwischen mehreren Hosts geteilt werden. Deshalb hat die Auftragsdatei eines SM-Pubsets einen eigenen Auftrags-Sperrmechanismus, der die Auftragskohärenz während eines Host-Wechsels oder einer Konfigurationsänderung gewährleistet.

Grundsätzlich gelten für Aufträge von SM-Pubsets die beiden folgenden Regeln:

1. Ein Auftrag, der bereits bearbeitet wird, wird durch den Sperrmechanismus an den Host gekoppelt, der den Auftrag bearbeitet. Ein solcher Auftrag kann von keinem anderen Host geändert, gestartet oder gelöscht werden.
2. Ein noch nicht bearbeiteter Auftrag ist an keinen Host gekoppelt. Auf einen solchen Auftrag kann von jedem Host oder Shareer, der den Pubset importiert hat, zugegriffen werden.

Der [Abschnitt „Auftragsverwaltung bei Shared-Pubsets“](#) informiert Sie über die Auswirkungen von Shared-Pubsets auf die Auftragsverwaltung. Der [Abschnitt „Aufträge wiederherstellen nach einem Host-Ausfall \(Recovery\)“](#) beschreibt Probleme, die sich aus einem Host-Wechsel oder einer Konfigurationsänderung ergeben. Die Kompatibilität von HSMS-Definitionen in einer heterogenen Versionsumgebung ist ab ["Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen"](#) beschrieben.

4.7.4.1 Auskunft über die Aufträge

Bei der Bearbeitung durch HSMS durchlaufen die Aufträge verschiedene Zustände (Request-States):

Status	Substatus	Bedeutung
INCOMPLETE		Auftrag nicht angenommen (Fehler bei der Annahme)
ACCEPTED		Auftrag von HSMS angenommen und in Auftrags datei geschrieben. Auftrag wartet auf seinen Start.
STARTED		Auftrag in Bearbeitung, und zwar:
	COLLECTED	– durch Sammelauftrag
	START-ARCHIVE	– an ARCHIVE übergeben
	ARCHIVE-COMPLETED	– von ARCHIVE beendet
	START-REPORT	– der Report wird erstellt
	SENT-TO-MASTER	– an den Master gesendet (Shared-Pubset)
	SENT-TO-BACK-SERV	– an den Backup-Server gesendet (Shared-Pubset)
	BACK-SERV-REPLIED	– Antwort vom Backup-Server
	BACK-SERV-NO-CONN	– keine Verbindung zum Backup-Server
	BACK-SERV-TIMEOUT	– Antwortzeit überschritten (Shared-Pubset, Backup-Server)
	MASTER-REPLIED	– Antwort vom Master (Shared-Pubset)
	MASTER-NO-CONNECT	– keine Verbindung zum Master (Shared-Pubset)
	MASTER-TIMEOUT	– Antwortzeit überschritten (Shared-Pubset)
	IN-TRANSMIT	– Report ist fertig; er muss noch zum aktiven Knoten gesendet werden
IN-DELETE	– Pseudo-Substatus: Das Löschen des gestarteten Auftrags wird gerade angestoßen.	
INTERRUPTED		Auftrag wurde während der Bearbeitung unterbrochen (eventuell Wiederanlauf)
	MASTER-NO-CONNECT	– keine Verbindung zum Master (Shared-Pubset)
	MASTER-TIMEOUT	– Antwortzeit überschritten (Shared-Pubset)
	MASTER-REPLIED	– Antwort vom Master (Shared-Pubset)
	BACK-SERV-REPLIED	– Antwort vom Backup-Server (Shared-Pubset)

	BACK-SERV-NO-CONN	– keine Verbindung zum Backup-Server (Shared-Pubset)
	BACK-SERV-TIMEOUT	– Antwortzeit überschritten (Shared-Pubset)
	START-ARCHIVE	– Auftrag an ARCHIVE übergeben
COMPLETED		Auftrag wurde von HSMS bearbeitet. Je nach Verlauf werden folgende Zusätze angehängt:
	WITH-WARNINGS	– Auftrag wurde mit Warnungen beendet.
	WITH-ERRORS	– Auftrag wurde mit Fehlern beendet.
CANCELLED		Auftrag wurde gelöscht, bevor er gestartet wurde. Kein Report verfügbar. Dies kommt bei Aufträgen mit Concurrent Copy vor (die keine SHC-OSD-Spiegelungsfunktionen verwenden), wenn ein /EXPORT-PUBSET gegeben wird, während der Auftrag den Status ACCEPTED hat.

Tabelle 5: Auftragsstatus

Die HSMS-Anweisung SHOW-REQUESTS informiert über die in der Auftragsdatei stehenden Aufträge und deren Status. Voreingestellt informiert die Anweisung über alle Aufträge der SF-Umgebung und alle vorhandenen SM-Umgebungen. Für SM-Umgebungen werden zusätzlich Informationen über den Auftrags-Sperrmechanismus ausgegeben. Das Feld „HOST/TSN“ zeigt an, welcher Host gerade den Auftrag sperrt. Nur aktive Aufträge (Status ACCEPTED oder STARTED) werden von ihrem bearbeitenden Host gesperrt.

Für die gezielte Auskunft über einzelne Aufträge ist die Auswahl ist möglich nach:

- einer HSMS-Umgebung (Auftragsdatei der SF- oder einer SM-Umgebung)
- dem Auftragsnamen, der bei der Erzeugung vergeben wurde
- dem Auftragsstatus
- dem Zeitpunkt der Erzeugung
- Benutzerkennungen (nur für den HSMS-Verwalter)
- dem Archiv, für das die Aufträge erzeugt wurden (nur für Archiveigentümer und HSMS-Verwalter)

4.7.4.2 Aufträge löschen

Mit der HSMS-Anweisung DELETE-REQUESTS können Aufträge mit folgendem Status aus der Auftragsdatei gelöscht werden:

- COMPLETED (Aufträge, die bereits vollständig bearbeitet wurden)
- INTERRUPTED (Aufträge, die zwar gestartet, aber unterbrochen wurden)
- ACCEPTED (Aufträge, die auf das Reaktivieren („Scheduling“) warten)
- CANCELLED (Aufträge, die zwar akzeptiert, aber vor dem Start abgebrochen wurden)
- STARTED (Aufträge, die gestartet wurden und gerade laufen)

Aufträge, die gelöscht werden sollen, können wie bei der Anweisung SHOW-REQUESTS durch Attribute ausgewählt werden. *ANY wählt alle Aufträge aus, die entweder den Status COMPLETED, INTERRUPTED, ACCEPTED oder CANCELLED haben. Das Löschen ist außerdem möglich über die Maskenausgabe der Anweisung SHOW-REQUEST. Dabei werden die zu löschenden Aufträge mit einer speziellen Delete-Markierung ausgewählt.

Zum Löschen von Aufträgen auf Shared-Pubsets siehe [Abschnitt „Auftragsverwaltung bei Shared-Pubsets“](#).

Aufträge mit dem Status STARTED werden nicht sofort gelöscht. Zum Löschen eines gestarteten Auftrags wird zuerst eine Löschmeldung an den HSMS-Servertask geschickt, der den Auftrag bearbeitet. Der HSMS-Servertask überprüft die Löschmeldung und bricht ab, nachdem er sie gelesen hat. Da aber die Löschmeldung keine Priorität hat, kann sie nicht gelesen werden, wenn der HSMS-Servertask beispielsweise in einer Sicherheits-Warteschlange gesperrt ist. Deshalb werden die gestarteten Aufträge nicht sofort gelöscht sondern erst nach einigen Minuten.

Aufträge mit dem Status COMPLETED, die älter als der mit KEEP-REQUESTS angegebene Wert sind, werden zu Beginn jeder HSMS-Session automatisch durch ein implizites Recovery gelöscht. Nähere Einzelheiten darüber finden Sie im [Abschnitt „Aufträge wiederherstellen nach einem Host-Ausfall \(Recovery\)“](#).

4.7.4.3 Aufträge wiederanlaufen lassen (Restart)

Aufträge, die während der Bearbeitung unterbrochen wurden (Status INTERRUPTED), können mit der HSMS-Anweisung RESTART-REQUESTS fortgesetzt werden. Sie müssen nicht komplett von vorne gestartet werden. Dabei lassen sich zwei Fälle unterscheiden:

- Der Auftrag wurde unterbrochen, bevor ein ARCHIVE-Lauf zur Bearbeitung erzeugt wurde. Der Auftrag kann fortgesetzt werden.
- Der Auftrag wurde unterbrochen, während er von ARCHIVE bearbeitet wurde. In diesem Fall lässt sich der Auftrag nur dann mit RESTART-REQUESTS fortsetzen, wenn Wiederaufsetzpunkte für den Auftrag in die ARCHIVE-Checkpointdatei geschrieben wurden.

Bei HSMS-Anweisungen für BS2000-Dateien (BACKUP-FILES, ARCHIVE-FILES, ...) sowie bei der HSMS-Anweisung BACKUP-NODE-FILES wird das Schreiben von Wiederaufsetzpunkten durch die Angabe WRITE-CHECKPOINTS=*YES beim Operanden OPERATION-CONTROL veranlasst oder durch WRITE-CHECKPOINTS=*STD, wenn die Voreinstellung für das betreffende Archiv *YES ist.

Die Aufträge können nur in der Betriebsweise, in der sie von HSMS angenommen wurden (SIMULATION oder OPERATION), fortgesetzt werden.

Bei einigen zufälligen Bedingungen kann es vorkommen, dass HSMS unter einem Auftrag mehrere ARCHIVE-Läufe erzeugt.

Wenn der Auftrag unterbrochen wird, läuft bei einem RESTART-REQUESTS nur der zuletzt gestartete ARCHIVE-Lauf wieder an, und der Auftrag wird danach beendet. Eine entsprechende Fehlermeldung wird im Report des Auftrags ausgegeben.

Beim Wiederanlauf von Aufträgen für Knotendateien müssen die entsprechenden UNIX-Dateisysteme eingehängt sein; andernfalls können längere Wartezeiten entstehen, bis die UNIX-Dateisysteme wieder aktiv sind.

Sicherungsaufträge mit Concurrent Copy, bei denen SHC-OSD-Spiegelungsfunktionen verwendet werden, können bei Problemen während der Bearbeitung den Status INTERRUPTED erreichen. In diesem Fall können sie fortgesetzt werden.

Alle anderen Sicherungsaufträge mit Concurrent Copy erhalten bei Problemen den Status COMPLETED-WITH-ERRORS und nicht den Status INTERRUPTED. Deshalb können sie nicht fortgesetzt werden.

Bei SM-Umgebungen werden Aufträge mit dem Status INTERRUPTED nicht mit dem bearbeitenden Host gekoppelt. Deshalb kann der Restart von jedem beliebigen Sharer des SM-Pubsets erteilt werden, unabhängig davon, auf welchem Host der Auftrag ursprünglich erzeugt wurde. Allerdings ist der Restart eines Auftrags nur mit derselben HSMS-Version möglich, mit der der Auftrag vor der Unterbrechung bearbeitet wurde.

Normalerweise kann ein Restart nur für Aufträge, die wegen eines Geräte- oder Datenträgerfehlers unterbrochen wurden, erteilt werden.

Ein Restart für Aufträge, die wegen Systemausfall an beliebiger Stelle im Ablauf unterbrochen wurden, kann in den meisten Fällen nicht erfolgreich ausgeführt werden.

i Es wird empfohlen, sobald in einer inhomogenen SM-Umgebung auf einem Sharer HSMS V12.0A installiert wurde, unterbrochene Aufträge in derselben Konfiguration wie bei ihrem Start fortzusetzen: das Verarbeitungssystem (Master oder Backup-Server) sollte dieselbe Version haben, wie der Auftrag bei seinem Start.

Beachten Sie, dass der RESTART von Aufträgen, die sich auf die erweiterte S1-Ebene beziehen, nur innerhalb von HSMS V12.0A möglich ist. Außerdem werden solche Aufträge in niedrigeren Versionen von HSMS abgebrochen und können danach innerhalb von HSMS V12.0A nicht gestartet werden.

4.7.5 Auftragsverwaltung bei Shared-Pubsets

Die Verwaltung von Aufträgen, die Shared-Pubsets betreffen, ist ziemlich umfangreich. Aufträge, die an einem Slave-Sharer erteilt werden, werden nämlich zur Bearbeitung in der Regel an den Master-Sharer gesendet. Nähere Informationen über die Bearbeitungsarten „Master“ und „lokal“ finden Sie im [Abschnitt „Arbeiten mit Shared-Pubsets“](#).

Ein Auftrag, der an den Master-Sharer gesendet wird, wird vom Master kopiert. Deshalb wird ein Auftrag, der von einem Benutzer erteilt wird, von zwei separaten Aufträgen erledigt, die sich dieselbe Auftragsidentifikation teilen. Der Originalauftrag, der am Slave-Host erstellt wurde, wird als „Primärkopie“ des Auftrags bezeichnet. Der duplizierte Auftrag, der vom Master-Host erstellt wurde, wird als „Masterkopie“ des Auftrags bezeichnet.

Aus Benutzersicht können Auftragsverwaltungs-Funktionen nur für die Primärkopie eines Auftrags erteilt werden, d. h. für den Auftrag, den der Benutzer tatsächlich erteilt hat. Die eventuelle vorhandene Masterkopie eines Auftrags wird von den Auftragsverwaltungs-Funktionen automatisch verwaltet.

Einige Unterschiede zwischen SF- und SM- Pubsets wirken sich auch auf die Auftragsverwaltung bei Shared-Pubsets aus:

- Bei SF-Umgebungen schreibt der Slave die Primärkopie des Auftrags in die HSMS-globale Auftragsdatei des Slave. Die Masterkopie des Auftrags schreibt der Master in die HSMS-globale Auftragsdatei des Masters.
- Bei SM-Umgebungen werden sowohl die Primär- als auch die Masterkopie eines Auftrags in dieselbe Auftragsdatei des SM-Pubsets geschrieben.

Im Folgenden werden die Besonderheiten der Auftragsverwaltung bei Shared-Pubsets für die SHOW-, DELETE- und RESTART-Funktion beschrieben.

Anzeigen von Aufträgen auf Shared-Pubsets

Bei der SHOW-Funktion erkennt man Masterkopien von Aufträgen am Wert YES im Ausgabefeld FROM-REMOTE.

Bei SF-Umgebungen kann nur die lokale Kopie sofort angezeigt werden, abhängig davon, ob die Anzeige auf dem Slave- oder Master-Host ausgeführt wird.

Bei SM-Umgebungen können die Primär- und die Masterkopie desselben Auftrags gleichzeitig angezeigt werden, wenn beide den Auswahlkriterien für den Auftragsstatus entsprechen. Die Masterkopie des Auftrags wird unmittelbar nach der Primärkopie angezeigt. Das Feld „HOST/TSN“ zeigt, welcher Auftrag auf welchem Host läuft.

Löschen von Aufträgen auf Shared-Pubsets

Bei der DELETE-Funktion können nur die Primärkopien von Aufträgen ausgewählt werden. Die zugehörigen Masterkopien werden von HSMS implizit gelöscht. Das Löschen der Primärkopie eines Auftrags wird nur nach dem erfolgreichen Löschen der zugehörigen Masterkopie veranlasst.

Ein Auftrag kann nicht gelöscht werden, während seine Masterkopie den Status STARTED hat.

Aufträge, die in einer SF-Umgebungen definiert wurden, können nur von dem Host aus gelöscht werden, an dem sie erteilt wurden. Zum impliziten Löschen der Masterkopie wird der Master-Host aufgerufen.

Aufträge, die in einer SM-Umgebungen definiert wurden, können prinzipiell von jedem Host aus gelöscht werden, der den SM-Pubset importiert hat. Solange ein Auftrag von einem Host gesperrt ist, kann er aber nur von diesem aus gelöscht werden.

Restart von Aufträgen auf Shared-Pubsets

Die RESTART-Funktion kann nur für die Primärkopie von Aufträgen erteilt werden, die den Status INTERRUPTED haben. Es wird aber die zugehörige Masterkopie angefordert, da nur die Masterkopie die Wiederaufsetzpunkte für HSMS enthält, die zum Restart benötigt werden. Wenn die Masterkopie nicht verfügbar ist oder nicht den Status INTERRUPTED hat, misslingt der Restart.

Bei SF-Umgebungen muss der Auftrag an den Host geschickt werden, der die Masterkopie des Auftrags gespeichert hat. Deshalb sendet der Restart den Auftrag immer an den Host, der zu dem Zeitpunkt Master-Sharer war, als der Auftrag zuerst bearbeitet wurde, auch wenn dieser Host inzwischen nicht mehr der Master-Sharer ist. Wenn dieser Host nicht mehr verfügbar ist, misslingt der Restart.

Bei SM-Umgebungen kann der Restart von jedem Host oder Sharer erteilt werden, der den SM-Pubset importiert hat. Wenn ein Masterkopie-Auftrag festgestellt wird, wird die Primärkopie des Auftrags geändert; dabei werden die Wiederaufsetzpunkt-Informationen der Masterkopie verwendet. Die Masterkopie wird anschließend aus der Auftragsdatei entfernt. Wenn der Auftrag bearbeitet ist, wird die Master-/Slave-Konfiguration des SM-Pubsets wieder aufgelöst.

Wenn der lokale Host Slave-Zugriff auf den SM-Pubset hat, wird der Auftrag zur Bearbeitung an den aktuellen Master geschickt.

4.7.6 Aufträge wiederherstellen nach einem Host-Ausfall (Recovery)

HSMS unterstützt das implizite und explizite Wiederherstellen von Aufträgen.

Implizites Wiederherstellen

Beim impliziten Wiederherstellen wird versucht, die Auftragsdatei zu bereinigen. Grundsätzlich werden folgende Tätigkeiten an den Aufträgen durchgeführt:

Das implizite Wiederherstellen von Aufträgen erfolgt zu Beginn einer HSMS-Session.

Bei einer SF-Umgebung geschieht das implizite Wiederherstellen bei jedem Erzeugen des Subsystems. Es werden alle Aufträge bearbeitet, die sich in der globalen HSMS-Auftragsdatei befinden.

Aufträge mit dem Status COMPLETED werden entsprechend dem Wert von KEEP-REQUESTS gelöscht. Der Wert von KEEP-REQUESTS kann mit dem Kommando MODIFY-HSMS-PARAMETERS geändert werden.

Standardmäßig ist der Wert *YES (40 Tage).

Bei einer SM-Umgebung geschieht das implizite Wiederherstellen bei jedem Startup des SM-Pubsets, d.h. beim Import des Pubsets oder beim Erzeugen des Subsystems, wenn auf den SM-Pubset zugegriffen werden konnte. Beim impliziten Wiederherstellen wird der Sperrmechanismus berücksichtigt: Ein fremder Host kann keinen Auftrag aktualisieren, der durch einen anderen Host gesperrt ist. Dies bedeutet, dass ein implizites Wiederherstellen eines SM-Pubsets lediglich ein „host-lokales“ Wiederherstellen ist.

Explizites Wiederherstellen

Im Rahmen des HIPLEX-Konzepts (siehe [Abschnitt „HIPLEX und HSMS“](#)) unterstützt HSMS einen expliziten Mechanismus zum erweiterten Wiederherstellen von Aufträgen. Dieser Mechanismus steht aber nur für Umgebungen mit SM-Pubsets zur Verfügung. Durch das explizite Wiederherstellen kann HSMS Host-Ausfällen und Host-Rekonfigurationen innerhalb von SM-Pubsets entgegenwirken.

Beim expliziten Wiederherstellen wird auf Aufträge zugegriffen, die durch einen anderen Host gesperrt sind. In der Regel befindet sich dieser ferne Host nicht mehr in der SM-Pubset-Konfiguration. Ursache dafür kann ein normaler Export oder ein System-Ausfall sein. Es besteht die Möglichkeit, dass in der Auftragsdatei noch Aufträge enthalten sind, die durch den fernen Host gesperrt sind.

Das explizite Wiederherstellen steht nur dem HSMS-Administrator auf dem Master-Sharer des SM-Pubsets zur Verfügung. Der HSMS-Administrator leitet es auf der Basis eines SM-Pubsets und eines Host-Namens ein, wenn er die Notwendigkeit dafür sieht. Ein ganz gewöhnlicher Grund dafür könnte sein, dass der ferne Host die SM-Pubset-Konfiguration nicht innerhalb einer angemessenen Zeitspanne wiederherstellen kann.

Beim expliziten Wiederherstellen wird überprüft, ob der wiederherzustellende Host ein aktiver Sharer des SM-Pubsets ist. Wenn dies der Fall ist, wird das Wiederherstellen zurückgewiesen. Bei einem exklusiven SM-Pubset wird dies dadurch erreicht, dass der Host selbst vom Wiederherstellen ausgeschlossen wird. Bei Shared-SM-Pubsets wird dies durch Aufruf der MSCF-Überwachungstask erreicht, die alle aktiven Sharern aller Shared-SM-Pubsets laufend überwacht. Wenn die Verbindung zur Überwachungstask abbricht, wird das Wiederherstellen ebenfalls zurückgewiesen.

Mit dem Operanden FORCE der Anweisung RECOVER-REQUESTS kann die Überprüfung unterdrückt werden. Dies kann sinnvoll sein, wenn der wiederherzustellende Host immer noch aktiv ist, aber keine HSMS-Tätigkeiten für den angegebenen SM-Pubset zeigt. Generell ist es nicht empfehlenswert, das Wiederherstellen eines Sharers zu erzwingen, weil das Wiederherstellen

- Bearbeitungssperren beseitigt

- einige Aufträge zweimal starten könnte
- Inkonsistenzen im Archivverzeichnis zur Folge haben könnte.

Durch das Wiederherstellen wird jeder Auftrag, der durch einen Host gesperrt ist, normalisiert. Die Sperre wird aufgehoben und der Auftrag wird in Abhängigkeit von seinem Status geändert, ähnlich wie beim impliziten Wiederherstellen:

- Aufträge mit dem Status ACCEPTED werden am lokalen Host reaktiviert, wobei eine neue Sperre gesetzt wird.
- Aufträge mit Concurrent Copy, die den Status ACCEPTED haben und keine SHC-OSD-Spiegelungsfunktionen benutzen, erhalten den Status CANCELLED.
- Aufträge mit dem Status STARTED erhalten den Status INTERRUPTED.
- Aufträge mit dem Status STARTED, die nicht erneut gestartet werden können, erhalten den Status COMPLETED.

Nach dem Wiederherstellen kann auf die entsperrten Aufträge von jedem Sharer aus zugegriffen werden. Aufträge mit dem Status INTERRUPTED können von jedem Sharer erneut gestartet werden. Aufträge mit dem Status INTERRUPTED, COMPLETED oder CANCELLED können von jedem Sharer gelöscht werden.

Beim Ausfall von zwei Hosts – beispielsweise Master und ein Slave – kann für jeden Host ein explizites Wiederherstellen auf dem neuen Master-Host angefordert werden. Dabei hat die Reihenfolge des Wiederherstellens keinen Einfluss auf das Ergebnis.

4.7.7 Prioritäten für die Auftragsbearbeitung vergeben

Der HSMS-Verwalter kann für jeden HSMS-Auftrag festlegen, mit welcher Priorität er bearbeitet werden soll. Beispielsweise können wichtige Jobs wie Systemsicherungen eine hohe Priorität bekommen und bereits starten, selbst wenn das System noch mit vielen Jobs von nicht-privilegierten Benutzern belastet ist.

Dazu kann der HSMS-Verwalter mit dem Operanden REQUEST-PRIORITIES der HSMS-Anweisung MODIFY-HSMS-PARAMETERS eine Reihe von Standard-Prioritäten in den globalen HSMS-Steuerparametern festlegen. Jedem Archivtyp kann eine Standard-Priorität zugeteilt werden:

- Backup-Archive: Standard-Priorität für Lesen und Schreiben
- Langzeitarchive: Standard-Priorität für Lesen und Schreiben
- Migrationsarchive: Standard-Priorität für Lesen und Schreiben
- Archive für die Sicherung von Knoten: Standard-Priorität für Lesen und Schreiben
- Archive für die Langzeitarchivierung von Knoten: Standard-Priorität für Lesen und Schreiben
- Schattenarchive: Standard-Priorität für Lesen und Schreiben
- Standard-Priorität für Import und Export
- Standard-Priorität für implizites Zurückholen (Recall)

Die vergebenen Standard-Prioritäten können mit der HSMS-Anweisung SHOW-HSMS-PARAMETERS angezeigt werden (im Informationsblock *REQUEST-PRIORITIES*).

Anschließend kann der HSMS-Verwalter für jedes einzelne Archiv die Lese- und Schreibpriorität definieren. Er kann entweder eine spezifische Priorität festlegen oder auf die Standard-Priorität für den angegebenen Archivtyp verweisen. Dazu steht der Operand REQUEST-PRIORITIES in den HSMS-Anweisungen CREATE-ARCHIVE und MODIFY-ARCHIVE-ATTRIBUTES zur Verfügung.

Archive von nicht-privilegierten Benutzern haben standardmäßig einen Verweis auf die Standard-Prioritäten.

Prioritäten für Schattenarchive werden nur für HSMS-Anweisungen berücksichtigt, die ausschließlich im Schattenarchiv ausgeführt werden – beispielsweise eine RESTORE-FILES-Anweisung für ein Schattenarchiv.

Wenn ein Auftrag für ein Archiv erteilt wird, wird die Bearbeitungspriorität anhand der individuellen Lese- bzw. Schreibpriorität für dieses Archiv ermittelt. Falls für das betreffende Archiv die Standard-Priorität festgelegt wurde (REQUEST-PRIORITIES=*STD), wird die Standard-Priorität für diesen Archivtyp verwendet. Dabei wird zwischen Lese- und Schreibaufträgen unterschieden.

Ausnahmen

Zu dem allgemeinen Verfahren existieren die beiden folgenden Ausnahmen:

- Die serielle Verarbeitung von Aufträgen (siehe [Abschnitt „Steuerung der Auftragszeit“](#)) wird berücksichtigt. Dadurch kann sich die Priorität eines Auftrags ändern. Bei serieller Verarbeitung werden die Aufträge als „Sammlung von Aufträgen“ abgewickelt. Deshalb erhält jeder Auftrag die Priorität der Sammlung. Neue Aufträge erhalten dieselbe Priorität wie die vorher gesammelten Aufträge.
- Aufträge, die der HSMS-Verwalter als Expressaufträge erteilt, werden immer mit der höchsten Priorität bearbeitet. Bei Expressaufträgen, die seriell bearbeitet werden, kann sich allerdings die Priorität – wie vorstehend beschrieben – verringern.

4.7.8 Sammelaufträge

Mehrere Aufträge, die sich auf ein Archiv beziehen, arbeitet HSMS normalerweise einzeln und unabhängig voneinander ab. Unter bestimmten Voraussetzungen fasst HSMS aber Aufträge für ein Archiv vor der Verarbeitung zu einem Sammelauftrag (Collector-Request) zusammen. Dadurch kommt es bei der Bearbeitung zu weniger Bandzugriffen und Positionierbewegungen.

Aufträge werden zu einem Sammelauftrag zusammengefasst (collected), wenn folgende Bedingungen erfüllt sind:

- Die Aufträge können nicht sofort bearbeitet werden.
- Die Aufträge werden an dem Rechner bearbeitet, an dem sie erteilt wurden.
- Es sind Archivierungsaufträge oder Verdrängungsaufträge auf der Ebene S2.
- Die Aufträge wurden nicht vom HSMS-Verwalter erzeugt.
- Die Aufträge beziehen sich auf ein Systemarchiv (Eigentümer SYSHSMS).
- Die Standard-Sicherungsdatei dieses Archivs wird benutzt und fortgeschrieben.

Für die zusammengefassten Aufträge werden einzelne Reporte erzeugt und ausgegeben. Die Information über den erzeugten Auftrag ist weiterhin unter dem normalen Auftragsnamen möglich. Der Substatus ist COLLECTED, solange er in Bearbeitung (STARTED) ist.

Bei Abbruch eines Sammelauftrags muss der Wiederanlauf (Restart) des Sammelauftrags durch den HSMS-Verwalter erfolgen.

Wenn bei der Migration ein bestimmter Auftrag mit SM2 überwacht werden soll, dann werden alle Aufträge des Sammelauftrags überwacht.

4.7.9 Asynchrone und synchrone Verarbeitung

Bei der Eingabe einer Aktionsanweisung kann der Benutzer bestimmen, ob er auf die Abarbeitung der Anweisung durch HSMS warten oder sofort die Möglichkeit erhalten will, weitere HSMS-Anweisungen einzugeben.

4.7.9.1 Asynchrone Verarbeitung

Wenn der Benutzer nicht ausdrücklich etwas anderes vereinbart, prüft HSMS bei der Eingabe einer Aktionsanweisung lediglich die Syntax der HSMS-Anweisung und die Gültigkeit bestimmter Rahmenbedingungen. Danach erhält der Benutzer sofort wieder die Möglichkeit zur nächsten Eingabe.

Zur Abarbeitung der erforderlichen Aktionen wird ein Auftrag erzeugt und in die Auftragsdatei geschrieben; die Verarbeitung läuft im Hintergrund ab.

Auskunft über den Fortschritt seiner Aufträge erhält der Benutzer mit der HSMS-Anweisung SHOW-REQUESTS (siehe [Abschnitt „Auskunft über die Aufträge“](#)).

4.7.9.2 Synchrone Verarbeitung

Auf Wunsch kann der Benutzer nach der Eingabe der HSMS-Anweisung auch warten, bis HSMS die erforderlichen Aktionen beendet hat. Dazu gibt er unter OPERATION-CONTROL=*PARAMETERS(...) den Operanden WAIT-FOR-COMPLETION=*YES an.

Wenn HSMS feststellt, dass für die Bearbeitung der HSMS-Anweisung Bandzugriffe erforderlich sind, Bandverarbeitung zurzeit aber nicht zugelassen ist, dann wird die HSMS-Anweisung im Dialogbetrieb abgewiesen. Im Stapelbetrieb wird die Anweisung zu Beginn einer Bandverarbeitungszeit an einen Servertask übergeben.

Für die Annahme und den Beginn der Bearbeitung der Aufträge kann der HSMS-Verwalter maximale Wartezeiten vereinbaren (MODIFY-HSMS-PARAMETERS). Wenn der Auftrag innerhalb der festgelegten Wartezeit nicht angenommen wurde, wird der Auftrag zurückgewiesen.

Wenn die Bearbeitung eines Auftrags durch einen Servertask nicht innerhalb der festgelegten Wartezeit abgeschlossen wurde, bearbeitet HSMS den Auftrag automatisch asynchron weiter.

Die maximalen Wartezeiten kann der HSMS-Verwalter getrennt für Dialog- und Stapelaufträge festlegen.

Schema der Verarbeitung (vereinfacht):

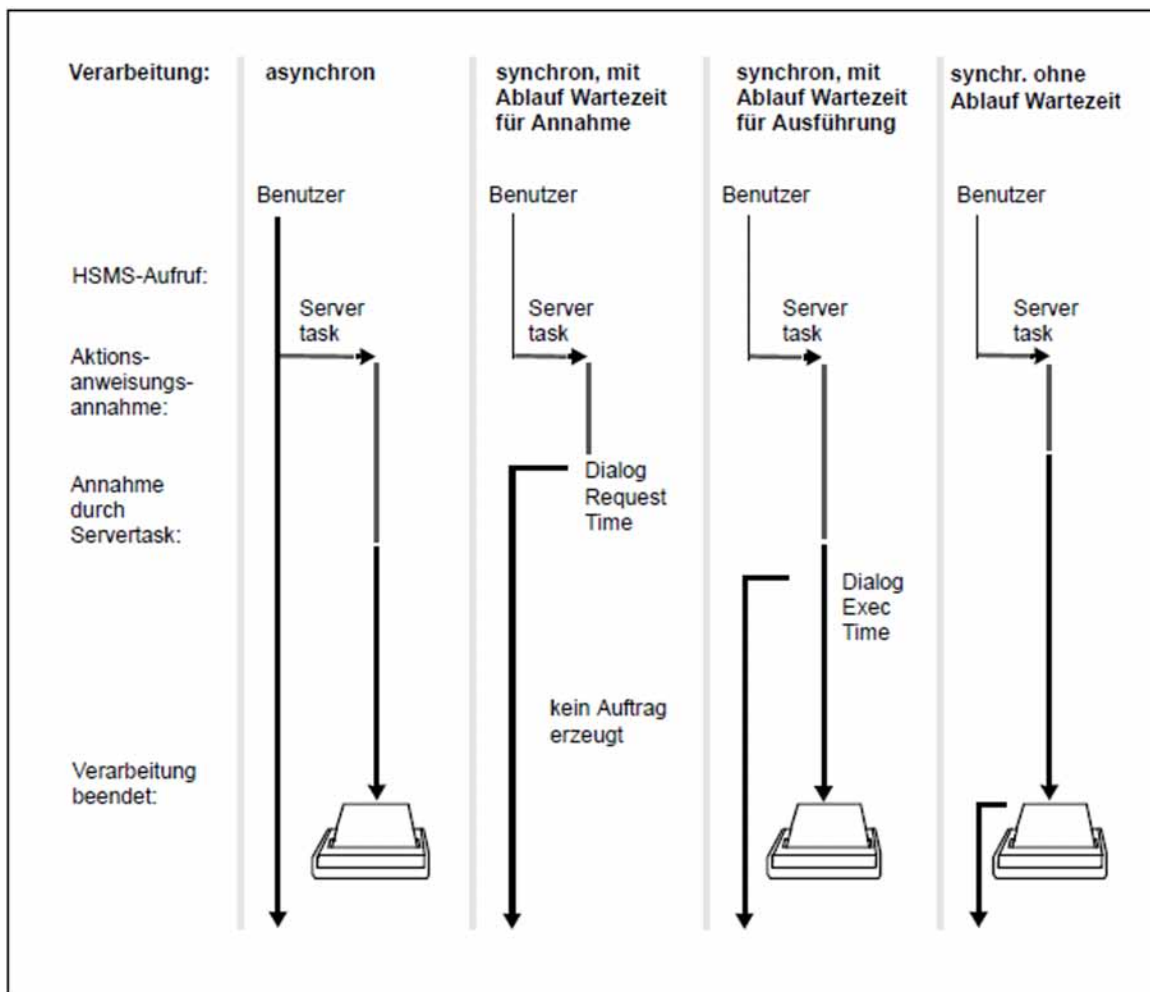


Bild 19: Asynchrone und synchrone Verarbeitung

4.7.10 Parallele und serielle Verarbeitung in ARCHIVE

Der folgende Abschnitt beschreibt, wie Sie beim Arbeiten auf der Speicherebene S2 eine bessere Performance erreichen können.

Jeder SAVE-, RESTORE- und COPY-Auftrag wird von einem einzigen HSMS-Servertask bearbeitet. Dieser HSMS-Servertask unterteilt den Auftrag in so genannte Pakete. Jedes Paket kann von einem ARCHIVE-Subtask individuell bearbeitet werden. ARCHIVE-Subtasks können parallel ablaufen. Am Anfang erhält jeder ARCHIVE-Subtask ein Paket. Wenn ein Subtask ein Paket abgearbeitet hat, erhält er ein neues Paket.

Für gute Performance sollte die Anzahl der erzeugten ARCHIVE-Subtasks gleich oder kleiner gleich der Anzahl der erstellten Pakete sein. Wenn kein Multiplexbetrieb eingestellt ist, benötigt jeder ARCHIVE-Subtask für SAVE-/RESTORE-Tätigkeiten ein Bandgerät und für COPY-Tätigkeiten zwei Bandgeräte. Die Anzahl der ARCHIVE-Subtasks, die parallel erfolgreich ablaufen können, ist so durch die Anzahl der verfügbaren Bandgeräte begrenzt.

Wenn Multiplexbetrieb eingestellt ist, können sich mehrere ARCHIVE-Subtasks dieselben Geräte für SAVE-/RESTORE-Tätigkeiten teilen, um eine bessere Performance und Auslastung der Geräte zu erreichen. Für COPY-Tätigkeiten werden doppelt so viele Geräte wie für Sicherungen benötigt.

In HSMS wird die Aufteilung in Pakete implizit durchgeführt. Die Anzahl der ARCHIVE-Subtasks, die für RESTORE- und COPY-Tätigkeiten benötigt werden, wird automatisch berechnet. Der Benutzer hat aber die Möglichkeit, die Anzahl der für einen BACKUP-FILES-Auftrag erstellten Pakete ebenso wie die Anzahl der für einen BACKUP-FILES- oder BACKUP-NODE-FILES erzeugten ARCHIVE-Subtasks zu steuern (siehe "[Steuern der Paketerzeugung durch den Benutzer \(Paketgenerierung\)](#)").

Wenn bei Operationen mit den Anweisungen BACKUP-NODE-FILES und RESTORE-NODE-FILES die Vor-/Nachbearbeitung aktiviert ist (Operand PRE-POST-PROCESSING), warten die ARCHIVE-Subtasks das Ergebnis der Vorbearbeitung ab, bevor sie den Datentransfer mit der Workstation beginnen.

Multiplexbetrieb

Im Multiplexbetrieb können sich mehrere ARCHIVE-Subtasks gleichzeitig parallel dieselben MBK-Geräte und Magnetbandkassetten teilen. Dadurch wird eine größere Performance beim Sichern und Restaurieren, eine optimale Ausnutzung der Bandgeräte sowie Streamen erreicht. Außerdem werden die Magnetbandkassetten optimal gefüllt, was automatisch die Speicherkosten senkt.

Vorgegeben wird beim Multiplexbetrieb die Zahl der parallel für einen Auftrag genutzten Bandgeräte (bzw. Bandgerätepaare beim Kopieren) sowie der Multiplex-Faktor, der die Zahl der ARCHIVE-Subtasks bestimmt, die sich ein Bandgerät teilen.

Im BS2000-Sicherungsbetrieb ist Multiplexbetrieb gut geeignet für Konfigurationen mit hochperformanten Magnetbandkassettengeräten (ab TAPE-C5/-C6).

Multiplexbetrieb wird in HSMS dadurch ermöglicht, dass mehrere ARCHIVE-Subtasks mit einem ARCHIVE-Bandlaufwerk verbunden werden. Das ARCHIVE-Bandlaufwerk ist für alle Zugriffe auf das damit verbundene Gerät verantwortlich. Die Anzahl der Subtasks für einen Gerätetreibertask legt den Faktor für seinen Multiplexbetrieb fest. Dadurch beziehen sich die Subtasks auf den Gerätetreibertask für das Schreiben auf Band bzw. das Lesen von Band.

Beim Restaurieren wird automatisch paralleles Demultiplexen durchgeführt. Die Anzahl der benötigten Subtasks hängt von der Anzahl der Geräte, der Multiplex-Konfiguration bei der Sicherung und den zu restaurierenden Dateien ab (siehe auch [Abschnitt „Multiplexbetrieb“](#)).

Eingeleitet wird der Multiplexbetrieb durch die Angabe `PARALLEL-RUNS= *MULTIPLEXING(...)` der HSMS-Anweisung `BACKUP-NODE-FILES`, wobei die Unteroperanden `NUMBER-OF-DEVICES` und `MULTIPLEXING-FACTOR` die Geräteanzahl und den Multiplex-Faktor bestimmen.

Die Einstellungen für den Multiplexbetrieb können auch auf Archivebene mit der HSMS-Anweisung `CREATE-ARCHIVE` bzw. `MODIFY-ARCHIVE-ATTRIBUTES` festgelegt werden. Diese werden bei der Sicherung durch die Angabe von `PARALLEL-RUNS=*STD` ausgewertet.

Wenn Daten auf Platte gesichert werden, wird kein Multiplexbetrieb durchgeführt. Aber die Anzahl der `ARCHIVE-Subtasks` ist genauso groß wie die vorgesehene Anzahl der Geräte (`NUMBER-OF-DEVICES`).

Mit `MULTIPLEXING-FACTOR=*AUTOMATIC` (Voreinstellung) bestimmt `ARCHIVE` selbst einen günstigen Multiplexfaktor. Das Ergebnis der Aufteilung ergibt den Multiplexfaktor pro Laufwerk. Weitere Informationen finden Sie im [Abschnitt „Multiplexbetrieb“](#).

Angabe von Knoten-Dateien über `PATH-NAMES` im Multiplexbetrieb

Wenn die Pfadnamen der zu bearbeitenden Knoten-Dateien aus einer Datei oder einem Bibliothekselement übernommen werden (Operanden `PATH-NAMES=*FROM-FILE` bzw. `*FROM-LIBRARY-ELEMENT`), werden sie in der Reihenfolge bearbeitet, wie sie der Benutzer in der angegebenen Datei bzw. dem Bibliothekselement eingetragen hat. Deshalb lassen sich `BACKUP-NODE-FILES`-Läufe optimieren, wenn die Reihenfolge der Dateien vorher gut überlegt wird. Siehe auch [Abschnitt „Auswahl von Knotendateien des BS2000-UFS“](#).

Format von gemultiplexten Sicherungsdateien

Sicherungsdateien, die im Multiplexbetrieb erstellt wurden, haben ein etwas anderes Format als Sicherungsdateien, die nicht im Multiplexbetrieb erstellt wurden. Unter anderem werden den Bandsicherungsblöcken zusätzliche Kontrollinformationen hinzugefügt.

Fortsetzung von Sicherungsdateien im Multiplexbetrieb

Eine nicht gemultiplexte Sicherungsdatei kann nicht mit einer gemultiplexten Sicherungsdatei fortgesetzt werden, da die Bandformate unterschiedlich sind. Die gesamte Sicherungsdatei muss nämlich ein einheitliches Format besitzen, damit Dateien korrekt restauriert werden können.

Allerdings ist es möglich, eine gemultiplexte Sicherungsdatei mit einer nicht gemultiplexten Sicherungsdatei fortzusetzen. Dazu erstellt `ARCHIVE` automatisch eine Multiplex-Umgebung. Der verwendete Multiplexfaktor ist in diesem Fall 1.

Auswahl von Dateien über `*LATEST-BACKUPS-OR-S0` im Multiplexbetrieb

Wenn die zu sichernden Dateien über die Angabe `FROM=*LATEST-BACKUPS-OR-S0` ausgewählt werden, können während der offline-Kopierphase verschiedene Sicherungsdateien als Eingabe verwendet werden. Dies bedeutet, dass sowohl gemultiplexte als auch nichtgemultiplexte Sicherungsdateien kopiert werden können. Wenn eine gemultiplexte Sicherungsdatei kopiert wird, muss die gesamte neue Ausgabe-Sicherungsdatei als gemultiplext erstellt werden. Deshalb werden nicht gemultiplexte Eingabe-Sicherungsdateien in der neuen Ausgabe-Sicherungsdatei gemultiplext.

4.7.10.1 Automatische Aufteilung in Pakete

Bei Sicherungsläufen hängt das Erstellen der Pakete vom Dateityp ab:

BS2000-Dateien

Bei Sicherungsläufen erfolgt die Aufteilung in Pakete nach der Katalog- und Benutzerkennung.

Bei PARALLEL-RUNS=1 gibt es ein Paket pro Katalog- und Benutzerkennung. Sobald aber PARALLEL-RUNS>1 angegeben ist, teilt HSMS die Dateien einer Katalog- und Benutzerkennung nach ihrer Lage auf den Pubset-Platten jeweils in vier Pakete auf.

Wenn die Anzahl Pubset-Platten größer drei ist, enthalten alle vier Pakete Dateien. Wenn die Anzahl der Pubset-Platten kleiner drei ist, entspricht die Anzahl der gefüllten Pakete der Anzahl der Pubset-Platten. Das bedeutet, dass die restlichen der vier Pakete leer bleiben.

Die Paketaufteilung ermöglicht so die Parallelverarbeitung auch beim Sichern von Dateien aus nur einer Katalog- und Benutzerkennung. Die Aufteilung in kleinere Pakete bewirkt außerdem eine gleichmäßigere Auslastung der parallelen Subtasks und dadurch auch verkürzte Sicherungszeiten.

Eine Aufteilung wie im Abschnitt „[Steuern der Paketerzeugung durch den Benutzer \(Paketgenerierung\)](#)“ beschrieben, ist nur noch in Ausnahmefällen z.B. bei der Sicherung sehr weniger und gleichzeitig sehr großer Dateien anzuraten.

Dateien auf Net-Storage

Für die Dateien auf Net-Storage erzeugt HSMS unabhängig von der Angabe im Operanden PARALLEL-RUNS pro betroffenen Net-Storage-Volume ein Paket pro Katalog- und Benutzerkennung.

Knotendateien passiver Knoten-S0

Bei Sicherungsläufen werden Knotendateien von verschiedenen passiven Knoten in getrennte Pakete eingeordnet. Grundsätzlich wird erst die angegebene Anzahl von ARCHIVE-Subtasks gestartet und anschließend die Paketbildung vorgenommen. In einem Knoten wird zunächst aus den angegebenen Knotendateien die niedrigste gemeinsame Verzeichnisebene bestimmt und dann werden auf dieser Ebene die voneinander unabhängigen Teilbäume bestimmt. Wenn weniger freie Subtasks als Teilbäume vorhanden sind, werden mehrere Teilbäume zusammengefasst. In jedem Teilbaum wird für jede Verzeichnisebene ein Paket gebildet, dabei werden alle Pakete zu einem Teilbaum und damit alle Dateien von einem Dateibaum von der gleichen Subtask bearbeitet.

Diese Aufteilung gilt sowohl bei PARALLEL-RUNS wie auch bei MULTIPLEXING.

Wenn bei einem Sicherungslauf für Knotendateien im BS2000 mit File-Angabe *ALL gearbeitet wird, damit die Include/Exclude-Dateien auf dem zugehörigen Client ausgewertet werden, erfolgt vorab eine interne Aufteilung in Teilbäume, als wenn die Include/Exclude-Angaben im BS2000 erfolgt wären.

Bei RESTORE- und COPY-Läufen hängt die Aufteilung in Pakete davon ab, wie die Daten auf die Datenträger gesichert wurden.

4.7.10.2 Automatisches Erzeugen von ARCHIVE-Subtasks

Bei *Sicherungsläufen* wird die Anzahl der benötigten ARCHIVE-Subtasks wie folgt berechnet:

- Ohne Multiplexbetrieb als Zahl der parallel betriebenen Geräte, angegeben durch PARALLEL-RUNS mit Standardwert 1.
- Bei Multiplexbetrieb legt der Multiplexfaktor die Anzahl der ARCHIVE-Subtasks fest, die sich dieselben Geräte teilen. Die Gerätezahl multipliziert mit dem zahlenmässig vorgegebenem Multiplexfaktor ergibt so die Zahl der parallel arbeitenden Subtasks.
- Bei Vorgabe von MULTIPLEXING-FACTOR=*AUTOMATIC wird der tatsächliche Multiplexfaktor und damit auch die Zahl der Subtasks intern errechnet nach der Zahl der unabhängigen Teilbäume aus der Vorgabe der zu sichernden Dateien.

Bei *COPY*-Läufen ergibt sich die Anzahl der benötigten Subtasks aus der Anzahl der parallel betriebenen Geräte bei der Sicherung, unabhängig von Multiplexing.

Bei *RESTORE*-Läufen geht außerdem noch der verwendete Multiplexing-Faktor von der Sicherung ein.

4.7.10.3 Steuern der Paketerzeugung durch den Benutzer (Paketgenerierung)

Beim Sichern von BS2000-Dateien kann der Benutzer eine weitere Aufteilung der automatisch erzeugten Pakete durch die Option `SELECT=*FROM-FILE(...)` einleiten.

i Die automatische Paketaufteilung durch HSMS (siehe [Abschnitt „Automatische Aufteilung in Pakete“](#)) ermöglicht den Parallelbetrieb auch in den Fällen, für die in früheren HSMS-Versionen der Benutzer – wie nachfolgend beschrieben – die Paketerzeugung steuern musste. Außerdem ist die automatische Paketaufteilung durch HSMS für den Parallelbetrieb günstiger als die benutzergesteuerte Paketerzeugung, weil intern die für den Parallelbetrieb wichtige Lage der Dateien auf unabhängigen Platten beachtet wird.

Wenn ein Benutzer viele große Dateien in einem System besitzt, kann es vorteilhaft sein, die Sicherung dieser Dateien durch mehrere parallele Subtasks durchzuführen. Das Verfahren dazu ist nachfolgend beschrieben.

Beim Sichern wird in der Eingabe- bzw. Ausnahmedatei angegeben, welche Benutzerdateien gesichert bzw. von der Sicherung ausgenommen werden sollen und in welcher Reihenfolge sie bearbeitet werden sollen.

Dateien mit verschiedener Benutzerkennung oder verschiedener Katalogkennung werden auf verschiedene Pakete aufgeteilt. Die Pakete werden auf parallele Subtasks verteilt und parallel bearbeitet.

Dateien mit gleicher Benutzer- und Katalogkennung wurden bis HSMS V7.0 in ein Paket eingeordnet und daher auch in einem Subtask bearbeitet. Nur mit der im Folgenden beschriebenen Sondersyntax konnten solche Pakete für eine parallele Verarbeitung bei `BACKUP-FILES` aufgeteilt werden.

Aktuell teilt HSMS diese Dateien in Abhängigkeit von ihrer Plattenlage auf maximal

4 Pakete auf und ermöglicht so die parallele Verarbeitung. Die interne und automatische Feinaufteilung ist für den Benutzer bequemer und durch ihre Plattenorientierung günstiger für die Parallelverarbeitung und die gleichmäßige Subtask-Auslastung als die bisherige Methode mit Sondersyntax.

Die nachfolgende Hilfskonstruktion ist in Ausnahmefällen z.B. bei der Sicherung von wenigen aber sehr großen Dateien sinnvoll, um eine gleichmäßige Auslastung der Laufwerke zu erreichen:

Jeder Satz einer Eingabe- bzw. Ausnahmedatei muss einen Pfadnamen enthalten. Außerdem kann er wahlweise eine Gruppen-Identifikation (Gruppen-ID) enthalten. Durch die Gruppen-ID wird die Aufteilung in getrennte Pakete für Parallelverarbeitung durch verschiedene Subtasks erreicht. Mit Angabe der Gruppen-ID kann aber nicht das Zusammenfassen von Dateien verschiedener Benutzer- oder Katalogkennung im gleichen Paket erzwungen werden!

```
[<gruppen-id>;] <pfadname>
```

Der Pfadname kann voll- oder teilqualifiziert angegeben werden, mit oder ohne Katalog- und Benutzerkennung. Wildcards sind erlaubt. Wenn eine Gruppen-ID angegeben wird, muss sie aus mindestens einem, maximal acht alphanumerischen Zeichen bestehen. Wenn keine Gruppen-ID angegeben wird, wird ein Leerzeichen angenommen.

Wenn Gruppen-IDs zusammen mit dem Ausschluss von Dateien (Operand `EXCEPT-FILE-NAMES`) verwendet werden, müssen die in der Ausschlussdatei angegebenen Gruppen-IDs mit den in der Auswahldatei angegebenen Gruppen-IDs zusammenpassen.

Beispiele

Die folgenden Beispiele zeigen die Umsetzung einer HSMS-Dateiangabe in eine ARCHIVE-Angabe beim NAME-Operanden.

Beispiel 1

Eingabedatei:	:a:file1 :a:file2 :a:file3 :b:file1	
ARCHIVE-Angaben:	NAME 1 :a:file1 :a:file2 :a:file3	NAME 2 :b:file1

Beispiel 2

Eingabedatei:	pkt1;;a:file1 pkt2;;a:file2 pkt1;;a:file3 pkt1;;b:file1		
ARCHIVE-Angaben:	NAME 1 :a:file1 :a:file3	NAME 2 :a:file2	NAME 3 :b:file1

Beispiel 3

Eingabedatei:	:a:file1 2;;a:file2 :a:file3 1;;b:file1		
ARCHIVE-Angaben:	NAME 1 :a:file1 :a:file3	NAME 2 :a:file2	NAME 3 :b:file1

Beispiel 4

Eingabedatei:	PKT1;;a*:\$u*.file1 PKT2;;a*:\$u*.file2	Pubset auf dem System: a1, a2
ARCHIVE-Angaben:	NAME 1 :a1:\$u*.file1	NAME 2 :a1:\$u*.file2

	NAME 3 :a2:\$u*.file1	NAME 4 :a2:\$u*.file2
--	--------------------------	--------------------------

Beispiel 5

Eingabedatei:	pkt1;;a:file1* pkt2;;a:file2* pkt1;;b:file1*
Ausnahmedatei:	pkt1;;a:file13 pkt2;;a:file22 pkt1;;b:file11
ARCHIVE-Angaben:	NAME1 :a:file1* außer :a:file13
	NAME2 :a:file2* außer :a:file22
	NAME 3 :b:file1* außer :b:file11

4.7.10.4 Steuern der erzeugten Subtasks durch den Benutzer

Der Benutzer kann die Anzahl der erzeugten Subtasks mit dem Operanden PARALLEL-RUNS steuern. Die Auswertung des bei PARALLEL-RUNS angegebenen Wertes ist aber folgenden Regeln unterworfen:

- Die Anzahl der Subtasks, die zum Fortsetzen, Lesen oder Kopieren einer vorhandenen Sicherungsdatei verwendet werden kann, darf nicht die Anzahl der Subtasks überschreiten, die beim Erstellen der Sicherungsdatei tatsächlich benutzt wurde. Dies gilt auch für das Fortsetzen einer vorhandenen Sicherungsdatei ohne Multiplexbetrieb.

Wenn Multiplexbetrieb eingestellt ist, kann eine vorhandene Sicherungsdatei mit einem höheren Multiplexfaktor fortgesetzt werden als die vorangegangenen Sicherungsversionen. Daher kann zwar die Anzahl der ARCHIVE-Subtasks höher sein, nicht aber die Anzahl der Geräte.

- Wenn bei COPY-Tätigkeiten auf einer Sicherungsdatei die Anzahl der verwendeten Subtasks kleiner ist als die Anzahl der Subtasks, die beim Erstellen der Sicherungsdatei tatsächlich benutzt wurde, kann es (bei SEVERAL-SVID) vorkommen, dass der Datenträger während der Verarbeitung nochmals zur Bandanfangsmarke zurückgespult wird.

Hinweise

- Wenn die Anzahl der erzeugten Pakete geringer ist als die Anzahl der angeforderten ARCHIVE-Subtasks, werden einige ARCHIVE-Subtasks erzeugt, die nicht tätig werden.
- Die Anzahl der ARCHIVE-Subtasks, die zum Erstellen von Sicherungsdateien verwendet wird, sollte so berechnet werden, dass dieselbe Anzahl von Subtasks für COPY-Tätigkeiten von diesen Sicherungsdateien benutzt werden kann, wobei die zum Kopierzeitpunkt verfügbaren Bandgeräte zu berücksichtigen sind.

4.8 Datenträgerverarbeitung

Im folgenden Abschnitt sind beschrieben:

- die Datenträger, die auf der Speicherebene S2 unterstützt werden
- die Zuweisung von Datenträgern
- Datenträgerverwaltung über Maren
- die Datenträger- und Gerätereservierung
- die Lagerorte von Datenträgern und Gerätepools
- die Bandverarbeitungszeiten
- die Steuerung der Bandverarbeitung

4.8.1 Datenträger auf S2

HSMS und ARCHIVE unterstützen auf der Speicherebene S2 alle Datenträger der Klasse TAPE, die von den BS2000-Versionen unterstützt werden, auf der die aktuelle HSMS-Version ablaufen kann.

Bei den Aktionsanweisungen, die Schreibaufträge auf S2 erzeugen, muss der gewünschte Datenträgertyp nicht angegeben werden. Eine Ausnahme besteht, wenn Datenträger nicht aus dem Archiv entnommen werden (siehe [Abschnitt „Zuweisung von Datenträgern“](#)).

Der Datenträgertyp, der angefordert wird, ist durch globale Steuerparameter festgelegt oder durch die Definition des Archivs, in das geschrieben wird.

Der HSMS-Verwalter legt den Datenträgertyp fest, der HSMS-global für alle Schreibaufträge auf S2 angefordert wird, sofern der Archiveigentümer keinen anderen Gerätetyp ausdrücklich vereinbart hat:

```
//MODIFY-HSMS-PARAMETERS -  
//  DEFAULT-HSMS-STORAGE=*PARAMETERS(S2-DEVICE-TYPE=<c-string>)
```

Archivspezifisch kann ein abweichender Standard-Datenträgertyp für Sicherungen auf S2 in dieses Archiv vereinbart werden:

```
//CREATE-ARCHIVE S2-DEVICE-TYPE=<c-string>
```

oder

```
//MODIFY-ARCHIVE-ATTRIBUTES S2-DEVICE-TYPE=<c-string>
```

Mit der Festlegung eines Standard-Datenträgertyps kann systemweit oder archivspezifisch ein neuer Datenträgertyp eingeführt werden, ohne dass Änderungen in Prozeduren erforderlich sind.

4.8.2 Zuweisung von Datenträgern

Für Schreibaufträge auf S2 werden Sicherungsdaträger benötigt. Schreibaufträge beziehen sich immer auf bestimmte Archive. Standardmäßig werden in diesem Fall die Datenträger aus dem Pool freier Datenträger des Archivs (oder aus dem zugeordneten MAREN-Pool) angefordert. Die Einrichtung und Verwaltung dieses Datenträger-Pools ist im [Abschnitt „Datenträger-Pool verwalten“](#) beschrieben.

Bei Sicherungsaufträgen und beim Kopieren von Sicherungsdateien haben der HSMS-Verwalter bzw. der jeweilige Archiveigentümer verschiedene Möglichkeiten die Zuweisung der Datenträger zu steuern.

4.8.2.1 Datenträgerzuweisung durch den Benutzer oder den Operator

Der Benutzer kann durch Angabe einer Liste von Archivnummern (VOLUMES=<vsn>) die Datenträger vorzugeben.

Der Operator erhält in folgenden Fällen am Bedienplatz eine Aufforderung zur Zuweisung von Datenträgern:

- Es wurde VOLUMES=*FROM-OPERATOR angegeben.
- Der Datenträger-Pool des Archives ist erschöpft ist und es können aus ihm keine weiteren Datenträger entnommen werden.

Sowohl die Datenträger, die der Operator zugewiesen hat, als auch die, die der Benutzer ausdrücklich angegeben hat, aber im Pool nicht enthalten sind, werden in das Archivverzeichnis aufgenommen. Sie werden bei //SHOW-ARCHIVE SELECT=*VOLUMES mit OWNER= OPERATOR ausgegeben.

Die Archiveinträge für diese Datenträger werden nach dem Löschen der Sicherungsdatei, die sie enthalten, wieder aus dem Archiv entfernt (auch bei FORCED-DELETE). Sie werden also nicht in den Datenträger-Pool aufgenommen.

Bei Einsatz von MAREN müssen diese Datenträger bereits der Benutzererkennung zugewiesen sein (STATUS=RESERVED). Sie werden dann von MAREN verwaltet.

4.8.2.2 Datenträgerzuweisung aus dem Datenträger-Pool

Bei `VOLUMES=*FROM-POOL` werden die Datenträger aus dem Datenträger-Pool des Archivs entnommen, der im Directory angelegt ist. Wurde kein Datenträger-Pool angelegt oder ist dieser Pool erschöpft, so werden die Bänder über MAREN aus einem zugeordneten MAREN-Pool zugewiesen.

Sind beide Pools erschöpft, so werden die Datenträger wie bei `VOLUMES=*FROM-OPERATOR` über den Operator angefordert.

Bei Freiwerden des Datenträgers bleiben diese dem ARCHIVE zugeordnet und stehen zur wiederholten Verwendung zur Verfügung.

Die Verwaltung des Datenträger-Pools ist im [Abschnitt „Datenträger-Pool verwalten“](#) beschrieben.

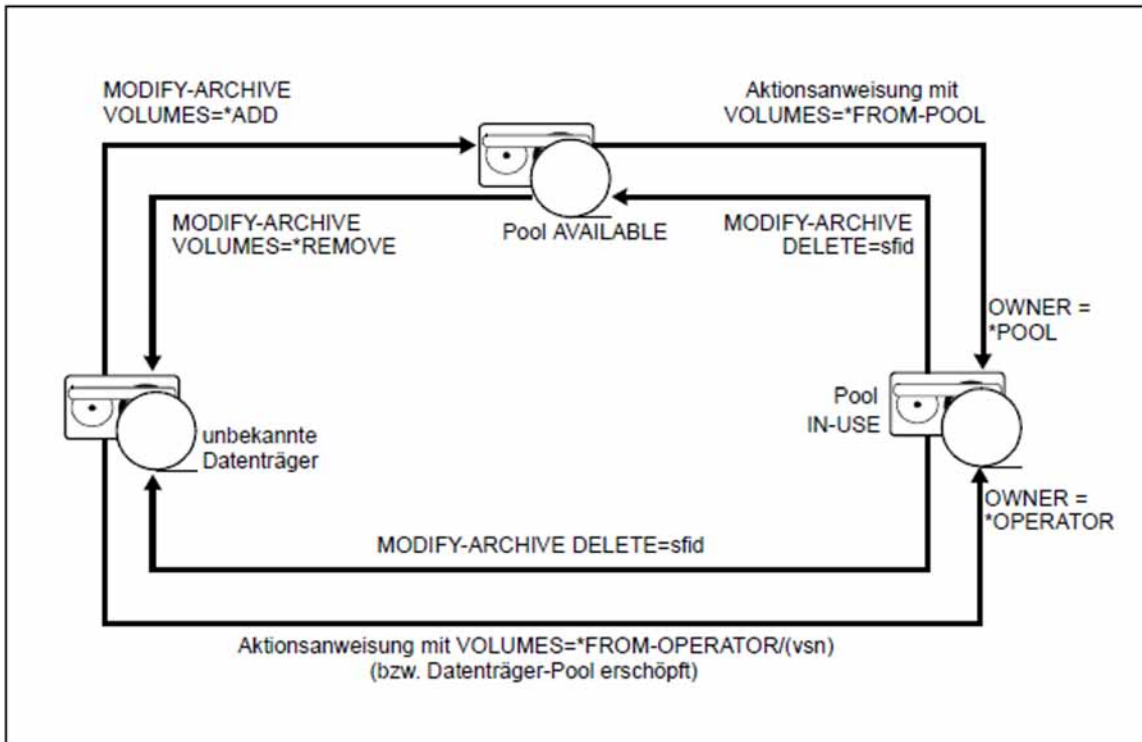


Bild 20: Zuweisung der Datenträger in HSMS

4.8.2.3 Datenträgerzuweisung über MAREN

Das MAREN-System (**M**agnetdatenträger-**A**rchivierungssystem im **R**echnernetz) verwaltet Datenträgerbestände in einem BS2000-Rechenzentrum. MAREN speichert alle Informationen über die Datenträger in einem eigenen MAREN-Katalog, der zentral für mehrere Anlagen eingerichtet werden kann. Durch eine enge Kopplung von MAREN an andere BS2000-Produkte, wie z.B. ROBAR, lässt sich der Rechenzentrumsbetrieb optimal organisieren.

Die freien Datenträger werden in MAREN in Freiband-Pools verwaltet.

Das Zuweisen von Datenträgern in MAREN bedeutet die Reservierung eines Datenträgers entweder explizit über MAREN-Anweisung oder implizit über die Freibandzuweisung. Dabei wird den Datenträgern insbesondere die Benutzerkennung und das Archivverzeichnis zugeordnet.

Dem HSMS-Archivverzeichnis wird ein MAREN-Pool auf dieselbe Art zugeordnet, wie dies bei der ARCHIVE-Directory-Datei geschieht: Der MAREN-Pool wird also nicht dem Archivnamen zugeordnet, sondern dem Archivverzeichnis. Die Ausführungen im Handbuch zu „MAREN“ [10] über ARCHIVE-Directory gelten also sinngemäß für HSMS-Archivverzeichnisse.

Datenträger aus MAREN-Freiband-Pools können in HSMS auf verschiedene Arten zugewiesen werden:

- Bei der expliziten Zuweisung reserviert sich der Benutzer vor dem HSMS-Aufruf selbst einen freien Datenträger mit der MAREN-Anweisung //RESERVE-FREE-VOLUMES. Er kann diesen Datenträger dann bei der Datensicherung direkt als Sicherungsdaträger angeben (VOLUMES=<vsn>).
- Die implizite Reservierung wird von der automatischen Freibandzuweisung durchgeführt. Die automatische Freibandzuweisung wird in Anspruch genommen bei Angabe von VOLUMES=*FROM-OPERATOR. Bei Angabe von VOLUMES=*FROM-POOL wird er in Anspruch genommen, wenn der Datenträger-Pool des Archivs erschöpft ist oder nicht angelegt wurde. Der Datenträger wird in diesem Fall für die Benutzerkennung des Archivverzeichnisses reserviert.
- Wird kein Band über die automatische Freibandzuweisung zugewiesen, so wird ein Datenträger am Bedienplatz angefordert. Der Operator weist dann einen Datenträger zu, der von MAREN verwaltet wird.

Zur Verwaltung von Datenträgern über MAREN siehe das Handbuch zu „MAREN“ [10].

4.8.3 Datenträger- und Gerätereservierung

Wenn unter HSMS ein Auftrag von ARCHIVE bearbeitet wird, setzt ARCHIVE ein SECURE-RESOURCE-ALLOCATION-Kommando ab, entweder für den Datenträger, der beschrieben oder gelesen werden soll, oder für das Gerät, das benutzt werden soll.

Ein SECURE-RESOURCE-ALLOCATION-Kommando wird mit der Wartezeit ausgegeben, die der HSMS-Verwalter für die Annahme und den Start von Aufträgen definiert hat (HSMS-Parameter DIALOG-REQUEST-TIME bzw. BATCH-REQUEST-TIME).

Die minimale Wartezeit beträgt 200 Sekunden. Bei Angabe einer Wartezeit von 99999 Sekunden wird die Secure-Wartezeit auf 7 Tage erhöht.

Wenn die Gerätereservierung nicht innerhalb der Wartezeit ausgeführt wird, wird der Vorgang entweder unterbrochen (bei Ausgaben) oder fortgesetzt und versucht, den Eingabedatenträger zu öffnen (bei Eingaben).

Hinweis

In der Benutzertask sollten keine SECURE-RESOURCE-ALLOCATION-Kommandos für Geräte (Operand DEVICE) abgesetzt werden, da diese in der Benutzertask reserviert bleiben und HSMS nicht zur Verfügung gestellt werden, wodurch die Ausführung des Auftrags behindert wird.

Beim Duplizieren von Sicherungsversionen kann ein gegenseitiges Sperren (Deadlock) eintreten, wenn zu wenig Bandgeräte verfügbar sind. Dies kann auch während BACKUP, ARCHIVAL oder EXPORT von verdrängten Dateien vorkommen. Das gegenseitige Sperren hört automatisch am Ende der Wartezeit auf, die bei REQUEST-WAIT-LIMITS in den HSMS-Operanden festgelegt wurde.

Der Benutzer kann diese Situation vermeiden, indem er

- die Wartezeit bei DIALOG-REQUEST-TIME bzw. BATCH-REQUEST-TIME kurz genug festlegt, um zu langes gegenseitiges Sperren zu verhindern, aber auch lang genug, um die Beendigung eines normalen Laufs zu ermöglichen.
- dafür sorgt, dass genügend Bandgeräte für alle Parallelläufe zur Verfügung stehen.

Zur Behebung von Deadlock-Situationen wird folgendes Verfahren bereitgestellt:

Wenn eine Deadlock-Situation festgestellt wird, wird die Meldung ARC0840 bzw. ARC0861 ausgegeben und automatisch eine Behebung gestartet. Dabei werden die Bänder der Task, die die Deadlock-Situation festgestellt hat, geschlossen und ihre aktuell geöffneten Datenträger werden freigegeben und erneut angefordert. Wenn die Deadlock-Situation ordnungsgemäß beendet und der Lauf normal fortgesetzt wurde, wird die Meldung ARC0862 ausgegeben.

Andernfalls wird die Meldung ARC0707 ausgegeben und die Task abgebrochen.

Hinweis

Bei RESTORE, IMPORT und beim Restart von Jobs kann keine Deadlock-Situation auftreten.

4.8.4 Lagerorte von Datenträgern und Gerätepools

Die Lagerorte von Datenträgern werden in MAREN verwaltet (siehe das Handbuch zu „MAREN“ [10]). MAREN protokolliert alle Lagerorte der Datenträger. Immer wenn ein Datenträger an einen anderen Lagerort gebracht wird, muss der zugeordnete Eintrag im MAREN-Katalog aktualisiert werden. In diesem Fall sind folgende Eingabefelder betroffen:

- TEMPORARY-LOCATION: Momentaner Lagerort des Datenträgers
- HOME-LOCATION: Dauer-Lagerort des Datenträgers

Besonders zur Unterstützung robotergesteuerter Archive im BS2000 (Softwareprodukt ROBAR) wurde im Nucleus Device Management (NDM) eine Verwaltung der Gerätepools entwickelt. Diese Gerätepoolverwaltung ermöglicht im Zusammenwirken mit MAREN dem Benutzer, gezielt bestimmte Geräte für die Bearbeitung von Band-Datenträgern auszuwählen. Die Zuordnung von Bandgeräten zu den Lagerorten erfolgt mit dem Kommando ADD-DEVICE-DEPOT. Die Lagerorte müssen allerdings vorher vom Systemverwalter in MAREN bekannt gegeben werden und mit den Angaben des Operators übereinstimmen.

Mit dem Kommando SECURE-RESOURCE-ALLOCATION ist es möglich, ein Bandgerät für einen angegebenen Lagerort zu reservieren. Wenn HSMS eine Magnetbandkassette zum Lesen oder Schreiben benötigt, wird intern ein /SECURE-RESOURCE-ALLOCATION mit Lagerortangabe abgesetzt. Dabei sind folgende Einschränkungen zu berücksichtigen:

- Die Geräte werden mit dem angegebenen oder im Archiv festgelegten Lagerort angefordert. Sind auch die Datenträger angegeben, müssen diese dem angegebenen Lagerort zugeordnet sein.
- Ist HSMS kein Lagerort, sondern nur der Datenträger bekannt, so wird der Lagerort der Datenträger über MAREN ermittelt.
- Bei den HSMS-Anweisungen RESTORE-FILES, IMPORT-FILES und RECALL-MIGRATED-FILES wird jeder Eingabedatenträger vor der Benutzung mit /SECURE-RESOURCE-ALLOCATION reserviert. Die Wartezeit dafür entspricht der Angabe, die im HSMS-Parameter DIALOG-REQUEST-TIME bzw. BATCH-REQUEST-TIME festgelegt wurde.

- Für andere Tätigkeiten (BACKUP-FILES, BACKUP-NODE-FILES, ARCHIVE-FILES, ARCHIVE-NODE-FILES, MIGRATE-FILES, EXPORT-FILES, COPY-SAVE-FILE, COPY-NODE-SAVE-FILE, COPY-EXPORT-SAVE-FILE, UPDATE-EXPORT-SAVE-FILE) gilt:
 - Ein /SECURE-RESOURCE-ALLOCATION wird einmal für das Ausgabegerät mit dem Lagerort des ersten Ausgabedatenträgers abgesetzt. Es wird dabei vorausgesetzt, dass jeder Folgedatenträger vom selben Lagerort kommt.
 - Ein /SECURE-RESOURCE-ALLOCATION wird zu Beginn des Duplizierens jeder Sicherungsversion abgesetzt. Es wird dabei vorausgesetzt, dass alle Datenträger, die zu einer Sicherungsversion gehören, vom selben Lagerort kommen (d.h., dass der Benutzer den Lagerort im MAREN-Katalog nicht geändert hat).
 - Wenn im System kein Lagerort festgelegt ist, wird /SECURE-RESOURCE-ALLOCATION ohne Angabe eines Lagerortes abgesetzt. Ist aber mindestens ein Lagerort angegeben, dann geht HSMS davon aus, dass alle Lagerorte, die in den Datenträgerfeldern des MAREN-Katalogs eingetragen sind, mit einem gültigen Lagerort übereinstimmen. Wenn dies nicht der Fall ist, führt dies zu einem Fehler und in einigen Fällen sogar zur Unterbrechung des Laufs.
 - Wenn einem Task bereits ein Ausgabegerät zugewiesen wurde und dieser Task von MAREN ein neues Ausgabeband anfordert, kann MAREN über dessen Ausgabegerät verfügen. Dadurch kann MAREN – falls nötig – das neu angeforderte Ausgabeband initialisieren. Nach dem Initialisieren wird das Ausgabegerät an den betroffenen Task zurückgegeben, damit dieser seine Bearbeitung fortsetzen kann.
- Bei //COPY-SAVE-FILE bzw. //COPY-NODE-SAVE-FILE mit SAVE-FILE= *NEW(...) unterstützt ARCHIVE keine Bandinitialisierung. In diesem Fall versucht ARCHIVE, das Band zu benutzen, ohne es zu initialisieren. Wenn trotzdem die Initialisierung des Bandes verlangt wird, weist MAREN das Band zurück.

4.8.5 Bandverarbeitungszeiten

Unter HSMS lässt sich die Auftragsbearbeitung so steuern, dass Eingaben von S2 und Ausgaben auf S2 nur zu bestimmten Zeiten stattfinden. Nur während einer Bandverarbeitungszeit (Tape-Session) werden Aufträge bearbeitet, für die Datenträger der Klasse „TAPE“ montiert werden müssen.

Die synchrone Bearbeitung von Aktionsanweisungen, die Datenträger der Klasse „TAPE“ benötigen, ist nur während der Bandverarbeitungszeiten zugelassen.

Zu Beginn einer Bandverarbeitungszeit stellt HSMS fest, welche Aufträge zu diesem Zeitpunkt in der Auftragsdatei warten, die Zugriffe auf S2 erfordern (für das betroffene Archiv, für eine bestimmte Zugriffsart, siehe unten). All diese Zugriffe werden während dieser Bandverarbeitungszeit bearbeitet. Gegebenenfalls werden bis zu diesem Zeitpunkt eingegangene Aufträge zu Sammelaufträgen zusammengefasst, was die Bearbeitung der Aufträge beschleunigt (siehe [Abschnitt „Sammelaufträge“](#)).

Alle Aufträge, die nach dem Beginn einer Bandverarbeitungszeit eingehen, werden der nächsten Bandverarbeitungszeit zugewiesen. Die Länge einer Bandverarbeitungszeit ist nicht steuerbar. Sie bemisst sich allein nach der Zeit, die die bis dahin aufgelaufenen Zugriffe zur Bearbeitung erfordern.

Ist die vorherige Bandverarbeitungszeit noch nicht beendet, wenn die Nächste anfängt, so schließt sie sich nach der Beendigung an.

Bandverarbeitungszeiten lassen sich systemweit und archivspezifisch festlegen. Die Steuerung der Bandverarbeitungszeiten liegt allein beim HSMS-Verwalter, auch für private Archive anderer Archiveigentümer. Mit dem Operanden DEFAULT-TAPE-CONTROL der HSMS-Anweisung MODIFY-HSMS-PARAMETERS nimmt er systemweite Voreinstellungen vor.

Für den Datentransfer gelten diese Einstellungen grundsätzlich; bei den anderen Grundfunktionen kann der HSMS-Verwalter für einzelne Archive mit der HSMS-Anweisung MODIFY-TAPE-CONTROL abweichende Vereinbarungen treffen.

Die Bandverarbeitungszeiten werden getrennt nach den verschiedenen Zugriffsarten und Auftragstypen festgelegt:

Auftragsart	Operand	Zugriffe
Leseaufträge	READ-CONTROL	Eingaben
Schreibaufträge	WRITE-CONTROL	Ausgaben
Expressaufträge	EXPRESS-CONTROL	Ein-/Ausgaben des HSMS-Verwalters

Als Leseaufträge gelten alle Aufträge, die erzeugt wurden durch:

- IMPORT-FILES
- RECALL-MIGRATED-FILES mit FROM-STORAGE=*ANY/*S2-STORAGE-LEVEL
- RESTORE-FILES
- RESTORE-NODE-FILES
- implizites Zurückholen von migrierten Dateien

4.8.5.1 Bandzugriffe nicht einschränken

Bei ...-CONTROL=*PROCESS-REQUESTS sind alle Aufträge, die sich auf das betreffende Archiv beziehen und zur gewünschten Zugriffsart gehören, jederzeit zugelassen.

Alle während *PROCESS-REQUESTS eingehenden Aufträge werden abgearbeitet, auch nach dem Wechsel auf eine andere Steuerungsart.

4.8.5.2 Bandzugriffe aufschieben

Bei ...-CONTROL=*HOLD-REQUESTS müssen alle betroffenen Zugriffe warten. Für das jeweilige Archiv und die angegebene Zugriffsart sind Zugriffe vorläufig nicht zugelassen.

Alle betroffenen Aufträge werden in der Auftragsdatei protokolliert, bis die Bandverarbeitung für diesen Auftragstyp durch einen anderen Wert geändert wird.

Hinweis

Wenn bei der ersten MODIFY-TAPE-CONTROL-Anweisung VALID-PERIOD= *SESSION vereinbart war, ändert der nächste Start von HSMS die Bandverarbeitung und startet in den meisten Fällen die wartenden Aufträge.

4.8.5.3 Bandverarbeitungszeiten festlegen

Bandverarbeitungszeiten werden festgelegt, indem bei ...-CONTROL=*BY-TAPE-SESSIONS(START-TIME=<zeit>) eine Uhrzeit festgelegt wird, zu der an jedem Tag aufs Neue die erste Bandverarbeitungszeit beginnt.

Bei START-TIME=*IMMEDIATELY sind Bandzugriffe nach der Eingabe der HSMS-Anweisung bzw. nach dem Start der HSMS-Session zugelassen, bei PERIOD=0 vom Beginn der Bandverarbeitungszeit bis zum Tagesende.

Mit *BY-TAPE-SESSIONS(PERIOD=<minutes>) wird angegeben, wie viele Minuten jeweils zwischen dem Beginn zweier Bandverarbeitungszeiten vergehen sollen.

4.8.5.4 Schreibzugriffe nach Menge steuern (nur für Archive für BS2000-Dateien)

Bei Schreibzugriffen (Operand WRITE-CONTROL) gibt es eine weitere Möglichkeit der Steuerung. Mit dem Operanden MINIMUM-PAGES kann der Beginn einer Bandverarbeitungszeit von der zu schreibenden Menge abhängig gemacht werden: Bandzugriffe werden erst zugelassen, wenn der Gesamtumfang aller Schreibaufträge in 2-KByte-Blöcken (PAM-Seiten) die angegebene Seitenzahl erreicht. Wenn die Mindestzahl bis zum Ende der HSMS-Session nicht erreicht wird, werden die Aufträge für die nächste HSMS-Session vorgemerkt.

Gezählt werden bei den MINIMUM-PAGES nur:

- Schreibaufträge von S0 auf S2
- Schreibaufträge von nicht-privilegierten Benutzern.

Nicht mitgezählt werden also:

- Schreibaufträge des HSMS-Verwalters
- Aufträge zum Kopieren von Sicherungsdateien
- Aufträge zum Verdrängen von S1 nach S2.

Durch die Angabe von START-TIME=*IMMEDIATELY, PERIOD=0 und MINIMUM-PAGES >= 0 kann eingestellt werden, dass Schreibzugriffe zeitlich ungesteuert jeweils erst dann erfolgen, wenn eine Mindestanzahl von PAM-Seiten geschrieben werden kann.

Wird zusätzlich zur Angabe von MINIMUM-PAGES für das betroffene Archiv mit NEW-STD-SAVE-FILE=*EACH-TAPE-SESSION ein Wechsel der Standard-Sicherungsdatei bei jeder Bandverarbeitungszeit vereinbart, so kann der ungefähre Füllgrad der beim Ausschreiben erzeugten Bandsicherungsdatei gesteuert werden. Dies gilt für Langzeit- und Migrationsarchive und ebenso für Sicherungsarchive, wenn ihre Definition diese Bearbeitung erlaubt.

Da bei der Zahl der zu schreibenden Seiten die Schreibaufträge des HSMS-Verwalters nicht berücksichtigt werden, sie also zusätzlich auf den Datenträger geschrieben würden, empfiehlt es sich in diesem Fall, Schreibaufträge des HSMS-Verwalters grundsätzlich als Expressaufträge zu starten.

4.8.6 Steuerung der Bandverarbeitung

Die Steuerung der Bandverarbeitung wird durch die Archivdefinitionen vorbestimmt (Operanden OPERATION-CONTROL und TAPE-CONTROL). Durch die gleichnamigen Operanden bei den Aktionsanweisungen kann die Bandverarbeitung aber auch für den einzelnen Auftrag beeinflusst werden.

4.8.6.1 Größe von Bandblöcken

Große Bandblöcke haben zwei Vorteile:

- Sie erhöhen den Füllgrad der Bänder wegen der geringeren Anzahl von Zwischenräumen (Gaps).
- Sie verbessern den Durchsatz beim Sichern auf Band und beschleunigen damit die Bandverarbeitung.

Die Bandblockgröße kann über den Operanden BLOCKING-FACTOR pro Anweisung oder als Archiv-Attribut eingestellt werden. Darüber hinaus kann die Bandblockgröße global über die ARCHIVE-Parameter BLOCK-SIZE-TAPE (für Langband) und BLOCK-SIZE-T-C (für MBK-Geräte) eingestellt werden.

*BLOCKING-FACTOR=*STD*

Die Angabe von *STD (ist auch Voreinstellung) wählt den Blockungsfaktor wie folgt:

- Für Anweisungen, bei denen das Archiv angegeben wird, gilt der Blockungsfaktor aus der Archivdefinition. Ist dort ebenfalls *STD eingestellt (Voreinstellung bei Anlegen des Archivs), gelten die Einstellungen der globalen ARCHIVE-Parameter.
- Für Anweisungen, bei denen kein Archiv angegeben wird, gelten die Einstellungen der globalen ARCHIVE-Parameter.

Für MBK-Geräte ist in der Parameterdatei SYSPAR.ARCHIVE.<vers> standardmäßig eine Blockgröße von 256 Kbyte eingestellt (BLOCK-SIZE-T-C=BIG). Für Geräte, die diese Blockgröße nicht unterstützen (emulierte Bandgeräte, T9G) nutzt ARCHIVE automatisch eine Blockgröße von 32 Kbyte. Sicherungsdateien werden mit der bisher genutzten Bandblockgröße fortgesetzt.

*BLOCKING-FACTOR=*MAX*

Die Angabe *MAX wählt den Blockungsfaktor, der in der aktuellen BS2000-Version maximal möglich ist. Derzeit ist der maximale Wert 128 (Blockgröße 256 Kbyte). Eine Blockgröße von 256 Kbyte entspricht umgekehrt auch der maximal möglichen Blockgröße für die aktuell freigegebenen Bandgeräte.

Sicherungsdateien, die mit dem maximalen Wert einer BS2000-Version geschrieben werden, können in einer niedrigeren BS2000-Version, die diesen Blockungsfaktor nicht unterstützt, nicht gelesen werden.

4.8.6.2 Magnetbandkassette entladen

Mit dem Operanden UNLOAD-TAPE kann gesteuert werden, ob eine Magnetbandkassette nach dem Beschreiben ganz zurückgespult und entladen oder auf die Bandanfangsmarke positioniert werden soll. Das Entladen des Bandes ist auch abhängig von der Systemvorgabe im BS2000, die der Operator mit dem Parameter UNLOAD-RELEASED-TAPE im Kommando MODIFY-MOUNT-PARAMETER einstellen kann.

4.9 Behandlung von BS2000-Platten

Im BS2000 werden sowohl Platten mit PAM-Schlüssel (K-Platten) als auch Platten ohne PAM-Schlüssel (NK-Platten) unterstützt. HSMS bedient ebenfalls beide Plattentypen. HSMS schreibt und liest Dateien mit allen PAM-Schlüssel-Formaten (BLOCK-CONTROL-INFO=*NO/*WITHIN-DATA-BLOCK/*PAMKEY).

Dateien können auf NK4-Platten gesichert bzw. von NK4-Platten restauriert werden. Zu den Randbedingungen bei Restore-Läufen auf NK4-Platten siehe das Handbuch „ARCHIVE“ [2], Abschnitt „NK4-Platten“.

Beim Lesen werden Dateien standardmäßig mit dem Format wieder eingespielt, mit dem sie gesichert wurden, sofern das Format des Datenträgers, auf das sie geschrieben werden, dies erlaubt.

Mit Ausnahme der PLAM-Bibliotheken ist die partielle Sicherung für PAM-Dateien ohne PAM-Schlüssel (BLOCK-CONTROL-INFO=*NO) nicht möglich.

In BS2000/OSD-BC ab V9.0 kann einem Pubset zusätzlicher Speicherplatz, den ein Net-Server bereitstellt, als Net-Storage-Volume zugeordnet werden. Aus BS2000-Sicht ist ein Net-Storage-Volume eine Platte mit dem Volumetyp NETSTOR.

Dateien auf Net-Storage werden wie Dateien auf lokalen Pubset-Platten gesichert und restauriert. Beim Restaurieren können Dateien auf Net-Storage geschrieben werden. Sicherungsdateien können auf Net-Storage erstellt bzw. auch dorthin kopiert werden.

4.9.1 Speicherplatzanforderung für Sicherungsdatei optimieren

Um die Performance beim Sichern auf Platte zu optimieren, sollten im Katalogeintrag der Sicherungsdatei die Werte für Primär- und Sekundär-Zuweisung an die erwartete Menge der zu sichernden Daten angepasst werden. Bei zu kleinen Werten entstehen entsprechend viele Datei-Extends, die sich ungünstig auf die Zugriffszeiten auswirken können.

Dasselbe gilt auch für beim Sichern auf die erweiterte S1-Ebene in einer SM-Umgebung und auf S1-SM-Pubsets in einer SF-Umgebung. In diesen Fällen wirken sich zu kleine Werte so aus, dass Sicherungsdateien mit der nächsten laufenden Nummer auf bereits benutzten Volume-Sets angelegt werden. Dies kann zu Problemen bei nachfolgenden Restore- oder Recall-Aufträgen führen.

Bei der Sicherung auf Net-Storage erhält die Primär-Zuweisung automatisch einen entsprechend hohen Wert.

4.9.2 Konvertierung von BS2000-Dateien

Für die Fälle, in denen der Ausgabedatenträger das Schreiben von PAM-Schlüsseln nicht erlaubt, die gesicherten Dateien aber einen PAM-Schlüssel besitzen, bietet HSMS die Möglichkeit der Konvertierung an.

Gerade beim Importieren von Dateien, die von anderen BS2000-Systemen stammen, oder beim Restaurieren von Dateien aus Langzeitarchiven kann eine Konvertierung erforderlich sein.

Die Konvertierung von Dateien kann mit dem Operanden FILE-CONVERSION der HSMS-Anweisungen RESTORE-FILES und IMPORT-FILES gesteuert werden. Für Jobvariablen wird dieser Operand ignoriert.

Standardmäßig konvertiert HSMS Dateien beim Restaurieren und Importieren. Bei Angabe von FILE-CONVERSION=*NO werden Dateien, die mit PAM-Schlüssel in der Sicherungsdatei stehen, beim Restaurieren bzw. Importieren auf eine NK-Platte nicht bearbeitet. SAM-Dateien mit PAM-Key werden ebenfalls nicht bearbeitet, falls sie als Datei des Typs Node-File auf Net-Storage restauriert oder importiert werden sollen. Für jede aus diesen Gründen nicht bearbeitete Datei gibt HSMS im Report eine Warnung aus.

Die Dateikonvertierung erfolgt durch das Subsystem PAMINT.

Beim Konvertieren gibt es zwei Möglichkeiten:

Standard-Format

PAMINT konvertiert Dateien mit PAM-Schlüssel beim Einlesen auf eine NK-Platte nach folgenden Regeln:

- K-ISAM Dateien in NK-ISAM-Dateien
(BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK)
- K-SAM-Dateien in NK-SAM-Dateien
(BLOCK-CONTROL-INFO=*WITHIN-DATA-BLOCK)
- K-UPAM-Dateien in NK-UPAM-Dateien
(BLOCK-CONTROL-INFO=*NO)

PAM-Dateien mit bekanntem Aufbau (Phasen, Bibliotheken) werden entsprechend umgesetzt. Bei PAM-Dateien mit unbekanntem Aufbau geht die PAM-Schlüssel-Information verloren. Eine Meldung zeigt an, wenn der PAM-Schlüssel Informationen enthielt, die verloren gegangen sind. LMR-Bibliotheken werden ins PLAM-Format umgesetzt

CONV-Format

PAMINT setzt Dateien mit PAM-Schlüssel beim Einlesen auf eine NK-Platte ins CONV-Format um. Die restaurierte bzw. importierte Datei enthält alle PAM-Schlüssel am Dateiende in separaten Blöcken.

Bei partiell gesicherten Dateien ist nur die Angabe *CONV möglich. Wenn eine zu restaurierende Datei konvertiert werden muss, wird der Operand RELEASE-UNUSED-SPACE ignoriert.

Konvertierung von großen Dateien (>= 32 GB)

Dateikonvertierungen mit PAMCONV unter HSMS funktionieren in gewohnter Weise auch für große Dateien. Zu beachten ist, dass große Dateien nur auf Pubsets mit den Attributen LARGE-OBJECTS=*YES und LARGE-FILES-ALLOWED=*YES konvertiert werden können.

Sonderfall

Der Platzbedarf für eine Datei im K-Format kann durch die Konvertierung nach NK steigen, so dass bei entsprechender Ausgangsgröße und Struktur der K-Datei die 32 GB Grenze überschritten wird. Eine Konvertierung ist nur dann möglich, wenn die Zielfeile auf einem Pubset mit den Attributen LARGE-OBJECTS=*YES und LARGE-FILES-ALLOWED=*YES liegt.

4.10 Ausgaben von HSMS

HSMS gibt Informationen für den Benutzer nach SHOW-Anweisungen in Bildschirmmasken und S-Variablen und nach der Bearbeitung von Aufträgen in Reporten aus.

4.10.1 Ausgaben nach SHOW-Anweisungen

- Bildschirmmasken
- S-Variablen

4.10.1.1 Bildschirmmasken

HSMS verwendet Bildschirmmasken, damit die bei den SHOW-Anweisungen angeforderten Informationen ausgegeben und Dateien im Dialog ausgewählt werden können. Alle HSMS-Bildschirmmasken haben

- einen Kopfteil, der neben der aktuellen HSMS-Anweisung die gewählten Anweisungsoptionen enthält. Bei der HSMS-Anweisung SHOW-HSMS-PARAMETERS enthält der Kopfteil zusätzlich die Angabe, ob die in der HSMS-Steuerdatei hinterlegten permanenten Parameter oder die temporär für die HSMS-Session gültigen Parameter ausgegeben werden.
- einen Datenteil zur Ausgabe der gewünschten Objekte.
- eine Fußzeile zur Eingabe von Aktionen wie Blättern oder Beenden der Ausgabe.
- Platz für die Ausgabe einer Meldung in der letzten Zeile.

Beim Datenteil unterscheidet HSMS Einzelbildschirme und Tabellenschirme:

- Einzelbildschirme werden verwendet, um alle Attribute eines Objekts (Archiv, Pubset etc.) auszugeben.
- Tabellenschirme enthalten Informationen über eine Vielzahl gleichartiger Objekte, jeweils ein Objekt pro Zeile.

Die Bildschirmmasken dienen nur zur Information. Die ausgegebenen Objekte bzw. Attribute können nicht verändert werden.

Die Maskenausgabe kann folgendermaßen erfolgen:

- Ausgabe in die logische Systemdatei SYSOUT und damit auf den Bildschirm
- Ausgabe in die logische Systemdatei SYSLST oder in eine Datei, die SYSLST zugewiesen wurde. Damit ist ein Ausdruck der Bildschirmmaske oder eine Bearbeitung in einer eigenen Datei möglich. Um Papier zu sparen, werden bei Ausgaben nach SYSLST Bildschirmmasken mit 43 Zeilen verwendet.

Alle Bildschirmmasken bieten in der Fußzeile hinter NEXT-PAGE die Möglichkeit zum Verändern der Ausgabe auf vorherige oder folgende Bildschirme (Blättern):

Eingabe	Wirkung
+	Ausgabe eines Bildschirms mit den nächsten Objekten
++	Ausgabe eines Bildschirms mit den letzten Objekten
–	Ausgabe eines Bildschirms mit den vorherigen Objekten
--	Ausgabe eines Bildschirms mit den ersten Objekten
E	Ende der Ausgabe

Hinweis

Nur die signifikanten, führenden Zeichen werden berücksichtigt.

Die Bildschirmmasken sind im Einzelnen im Handbuch „HSMS Bd. 2“ [1] im Anschluss an die HSMS-Anweisung, durch deren Eingabe die Ausgabe der Bildschirmmaske veranlasst wird, beschrieben.

Bei Bildschirmmasken in Tabellenform ist die Bedeutung der Rubriken jeweils in einer Legende im Anschluss an die Bildschirmmaske erklärt.

Hinweis

Die Anzeige von FHS-Masken wird nur von blockorientierten Datensichtstationen (z.B. 9750) unterstützt. Bei zeilenorientierten Datensichtstationen (z.B. Verbindung über VTSU) können keine FHS-Masken angezeigt werden. Deshalb sollte bei zeilenorientierten Datensichtstationen die Ausgabe der SHOW-...-Anweisungen nach *SYSLST gesendet werden.

4.10.1.2 S-Variablen

Bei folgenden Anweisungen kann der Inhalt der Bildschirmmasken auch in S-Variablen ausgegeben werden:

SHOW-ARCHIVE
SHOW-ARCHIVE-ATTRIBUTES
SHOW-HSMS-PARAMETERS
SHOW-NODE-PARAMETERS
SHOW-PUBSET-PARAMETERS
SHOW-REQUESTS
SHOW-SM-PUBSET-PARAMETERS
SHOW-TAPE-CONTROL

Die Struktur der S-Variablen entspricht der zeilenweise Ausgabe der Bildschirmmasken. Die S-Variablen sind für eine maschinelle Auswertung besser geeignet als der Inhalt der SYSLST-Datei.

4.10.2 Reporte

Nachdem eine Aktionsanweisung oder eine der Anweisungen MODIFY-ARCHIVE oder CHECK-CATALOGED-FILES bearbeitet ist, erzeugt HSMS Reporte, die über das Ergebnis des Auftrags informieren.

Bei den Aktionsanweisungen lässt sich die Ausgabe der Reporte beim Operanden OPERATION-CONTROL durch Unteroperanden steuern. Für den Unteroperanden OUTPUT kann mit dem globalen HSMS-Parameter OUTPUT ein Standardwert festgelegt werden. Dieser kann *PRINTER oder *MAIL sein.

Der Unteroperand REPORT legt den Umfang der Ausgabe fest:

- *SUMMARY führt zur Ausgabe einer Übersicht mit Informationen über die HSMS-Anweisung, den erzeugten Auftrag und die erzeugte Sicherungsdatei.
- *FULL gibt zusätzlich Informationen zu den einzelnen bearbeiteten Dateien, Elementen der PLAM-Bibliothek und Jobvariablen aus.
- *SAVED-FILES/*RESTORED-FILES gibt beim Sichern bzw. Restaurieren nur die tatsächlich bearbeiteten Dateien aus.
- *NONE unterdrückt die Ausgabe eines Reports.

Hinweis

Bei Migration von S1 oder von S2 (Reorganisation) werden intern mehrere Kopier- und Löschaufträge ausgeführt. Dabei erfolgt die Protokollierung wie im Operanden REPORT angegeben. Nur Löschaufträge protokollieren auch bei Angabe von REPORT=*SUMMARY vollständig (*FULL), damit der Report in jedem Fall die frei gewordenen Datenträger enthält.

Die Ausgaben zur Sicherungsdatei und zu den einzelnen Dateien ähneln den ARCHIVE-Listen. Sie enthalten jedoch einen veränderten Seitenkopf und zusätzliche HSMS-Meldungen.

Das Ziel der Ausgabe wird durch den Unteroperanden OUTPUT bestimmt:

- OUTPUT=*STD entspricht abhängig vom Wert des globalen HSMS-Parameters OUTPUT entweder OUTPUT=*PRINTER oder OUTPUT=*MAIL.
- OUTPUT=*PRINTER führt zur Ausgabe auf Drucker. Die Ausgabe geht immer auf Drucker, nicht in die Systemdatei *SYSLST.
- OUTPUT=<dateiname> führt zur Ausgabe in eine Datei. Wenn eine bereits katalogisierte, nicht leere SAM-Datei angegeben wird, wird diese nicht überschrieben, sondern erweitert (OPEN EXTEND). Falls diese Datei keine SAM-Datei ist, wird eine Fehlermeldung ausgegeben und die Ausgabe auf *PRINTER umgeleitet.
- OUTPUT=*LIBRARY-ELEMENT(LIBRARY=<bibliothek>,ELEMENT=<element>) führt zur Ausgabe in ein PLAM-Bibliothekselement. Erzeugt wird ein Element vom Typ P mit einer Version, die die Benutzerkennung des Aufrufers sowie Datum und Zeit enthält. Falls die Ausgabe in das Bibliothekselement nicht möglich ist, wird eine Fehlermeldung ausgegeben und die Ausgabe auf *PRINTER umgeleitet.
- OUTPUT=*MAIL führt zum Versenden des Reports als Anhang einer E-Mail. Verschickt wird die E-Mail an die Adresse, die im Benutzereintrag des Aufrufers eingetragen ist (siehe Kommando MAIL-FILE). Falls das Versenden per E-Mail nicht möglich ist, wird eine Fehlermeldung ausgegeben und die Ausgabe auf *PRINTER umgeleitet.

Die Reporte, die auf der Server-Seite ausgedruckt werden, haben ein unterschiedliches Layout. Das Layout hängt vom Typ der Workstation ab, auf dem die Dateien bearbeitet werden (BS2000- oder UFS-Report).

Reporte für Aktionsanweisungen, die an einem aktiven Knoten eingegeben werden, werden in vordefinierte Strukturen formatiert und in Antwortdateien geschrieben. Die Inhalte dieser Antwortdateien sendet der Kommunikations-Task über das Netz an den Eingabeknoten. Dieser Report wird vom Client übersetzt, damit er das passende Layout hat.

Bei HSMS-Aufträgen, die auch im BS2000 Backup Monitor am SE Server überwacht werden, werden die Reporte auch als PDF-Datei unter der Benutzerkennung SYSHSMS abgelegt. Auf diese Datei kann nur das System zugreifen.

Nachfolgend wird an einem Sicherungslauf der Aufbau eines HSMS-Reports erläutert.

1. Seite des Reports

Der *Kopfst* auf allen Seiten des Reports gleich. Er enthält in der Titelzeile:

- den Namen der Aktionsanweisung, die zum Auftrag führte
- die verwendete HSMS-Version
- Datum und Uhrzeit
- die laufende Seitennummer des Reports.

In der zweiten Zeile stehen Informationen zum Auftrag:

- Umgebung, in der die Anweisung bearbeitet wurde (SF oder SM)
- Auftragsname und -datum
- Benutzerkennung, unter der die Anweisung ausgeführt wurde
- Auftragsstatus (hier COMPLETED WITHOUT ERROR).

Anschließend wird die eingegebene HSMS-Anweisung protokolliert. Unter STATEMENT LISTING wird die komplette HSMS-Anweisung ausgegeben, mit der der Auftrag erzeugt wurde.

Danach sind weitere Operanden und ihre Werte aufgelistet (ENVIRONMENT, ARCHIVE-NAME, SAVE-FILE ATTRIBUTES, ...).

```

A***   BACKUP - FILES                HSMS V11.0          FULL          REPORT   ***2016-09-17
12:05:38   PAGE      1
        REQUEST-ENVIRONMENT=SF
        REQUEST-NAME=BCF#4362   REQUEST-DATE=2016-09-17 12:04:52   USER-ID=SYSHSM
        REQUEST-STATE=COMPLETED WITHOUT ERROR
        STATEMENT LISTING:
        BACKUP-FILES   FILE-NAMES=( ρMANUAL.FILE.0* , ρUSEROLD.FILE.1* , ρUSERNEW.FILE.2* ) , SELECT-
FILES=*ALL-FILES ,
                ARCHIVE-NAME=MANUAL.BAC , SAVE-FILE=*NEW(RETENTION-PERIOD=0) ,
                TO-STORAGE=*S2-STORAGE-LEVEL( VOLUMES=( WORK12 , WORK13 , WORK14 ) ) ,
                OPERATION-CONTROL=*PARAMETERS( PARALLEL-RUNS=3 , REPORT=*FULL , OUTPUT=HSMS.MAN.R.
BCF.1 )

        ENVIRONMENT                : SF
        ARCHIVE-NAME                : ρSYSHSMS.MANUAL.BAC
        SAVE-FILE ATTRIBUTES        : NEW
        TO-STORAGE                  : S2-STORAGE-LEVEL
        DEVICE-TYPE                 : TAPE-C4
        RETENTION-PERIOD            : 0
        SELECT-FILES PARAMETER
        ALL-FILES
        MAX-BACKUP-CLASS            : STD
    
```


2. Seite des Reports

Auf der zweiten Seite werden die Meldungen von ARCHIVE über die Annahme des Auftrags (ARC0002) und die Erzeugung von Subtasks (ARC0033) aufgelistet. Bei einigen HSMS-Anweisungen (z.B. //MODIFY-ARCHIVE) befinden sich diese Meldungen bereits auf der ersten Seite unten.

Anschließend wird der Name der bearbeiteten Sicherungsdatei ausgegeben und welcher Parallelauf welchen Datenträger bearbeitet hat:

Spalte	Bedeutung
SAVE FILE IDENTIFIER	Nummer der Sicherungsdatei (SFID), die geschrieben oder gelesen wurde
SUBSAVE NUMBER	Nummer des Parallellaufs
VSNS	Archivnummern der Datenträger mit der Sicherungsdatei. Bei Sicherungsdateien auf S1 wird nur die Pubset-Id ausgegeben, eingeleitet mit „0:“

Danach werden die bearbeiteten Dateien und Jobvariablen in einer Tabelle aufgelistet, deren Spalten die folgende Bedeutung haben:

Spalte	Bedeutung
FILE/JOB VARIABLE NAME	Namen der bearbeiteten Dateien und Jobvariablen
VERS	Versionsnummer der gesicherten Datei
LASTPG/SIZE	Größe der Datei in PAM-Seiten (Last-Page-Pointer), bei Jobvariablen die Größe in Bytes
SAVE TYPE	Sicherungstyp (FULL, PART, MIGF, FGCI, CNS, OPER, FNOD)
INPUT VSN	Archivnummer des ersten Eingabedatenträgers
DEV TYP	Gerätetyp des Eingabedatenträgers (D=Disk, T=Tape, C=Catalog)
SUBSAVE	Nummer des Parallellaufs, in dem der Datenträger geschrieben/gelesen wurde
OUTPUT VSN(S)	Archivnummern der Ausgabedatenträger

Die zu den aufgelisteten Dateien gehörigen Kennungen werden in einer Überschriftszeile aufgeführt, z.B.:

```

***   CATALOG - RD3A   USER - MANUAL   ***
A***  BACKUP - FILES   HSMS V11.0   FULL   REPORT   *** 2016-09-17 12:05:38   PAGE   2
      REQUEST-ENVIRONMENT=SF
      REQUEST-NAME=BCF#4362   REQUEST-DATE=2016-09-17 12:04:52   USER-ID=SYSHSMS
      REQUEST-STATE=COMPLETED WITHOUT ERROR
%   ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160917.120456', VERSION '11.0'
%   ARC0033 ARCHIVE SUBTASK TSN '4487' GENERATED
%   ARC0033 ARCHIVE SUBTASK TSN '4488' GENERATED
%   ARC0033 ARCHIVE SUBTASK TSN '4489' GENERATED
      SAVE FILE IDENTIFIER - S.160917.120456   SAVE-VERSION-DATE=16-09-17   SAVE-VERSION-
TIME=12:04:57
      SUBSAVE
      NUMBER           VSNS
      0                WORK14
      1                WORK13
      2                WORK12
      SAVE FILE IDENTIFIER - S.160917.120456   SAVE-VERSION-DATE=16-09-17   SAVE-
VERSION-TIME=12:04:57
***   CATALOG - RD3A   USER - MANUAL   ***
      FILE/JOB VARIABLE NAME
                                LASTPG/  SAVE  INPUT DEV  SUB  OUTPUT
                                VERS    SIZE TYPE   VSN  TYP  SAVE  VSN(S)
FILE.01                        1      5  FULL  RD3A.0  D    2  WORK12
FILE.02                        1      4  FULL  RD3A.0  D    2  WORK12
FILE.03                        1      4  FULL  RD3A.0  D    2  WORK12
FILE.04                        1     14  FULL  RD3A.0  D    2  WORK12
FILE.05                        1      9  FULL  RD3A.0  D    2  WORK12
FILE.06                        1      1  FULL  RD3A.0  D    2  WORK12
FILE.07                        1     21  FULL  RD3A.0  D    2  WORK12

```

3. und weitere Seiten des Reports

Wenn die zweite Seite zur Auflistung der bearbeiteten Dateien und Jobvariablen nicht ausreicht, folgen weitere Seiten, z.B.:

```

A***  BACKUP - FILES                HSMS V11.0          FULL          REPORT  *** 2016-09-17
12:05:38  PAGE      3
        REQUEST-ENVIRONMENT=SF
        REQUEST-NAME=BCF#4362  REQUEST-DATE=2016-09-17 12:04:52  USER-ID=SYSHSMS
        REQUEST-STATE=COMPLETED WITHOUT ERROR
        SAVE FILE IDENTIFIER - S.160917.120456  SAVE-VERSION-DATE=16-09-17  SAVE-VERSION-
TIME=12:04:57

                ***  CATALOG - RD3A          USER - USERNEW          ***
        FILE/JOB VARIABLE NAME                LASTPG/  SAVE  INPUT DEV  SUB  OUTPUT
                                                VERS    SIZE  TYPE    VSN  TYP  SAVE  VSN(S)
FILE.21                1         5  FULL  RD3A.0  D    1  WORK13
FILE.22                1         4  FULL  RD3A.0  D    1  WORK13
FILE.23                1         4  FULL  RD3A.0  D    1  WORK13
FILE.24                1        14  FULL  RD3A.0  D    1  WORK13
FILE.25                1         9  FULL  RD3A.0  D    1  WORK13
FILE.26                1         1  FULL  RD3A.0  D    1  WORK13
FILE.27                1        21  FULL  RD3A.0  D    1  WORK13
        SAVE FILE IDENTIFIER - S.160917.120456  SAVE-VERSION-DATE=16-09-17  SAVE-VERSION-
TIME=12:04:57

                ***  CATALOG - RD3A          USER - USEROLD          ***
        FILE/JOB VARIABLE NAME                LASTPG/  SAVE  INPUT DEV  SUB  OUTPUT
                                                VERS    SIZE  TYPE    VSN  TYP  SAVE  VSN(S)
FILE.11                1         5  FULL  RD3A.0  D    0  WORK14
FILE.12                1         4  FULL  RD3A.0  D    0  WORK14
FILE.13                1         4  FULL  RD3A.0  D    0  WORK14
FILE.14                1        14  FULL  RD3A.0  D    0  WORK14
FILE.15                1         9  FULL  RD3A.0  D    0  WORK14
FILE.16                1         1  FULL  RD3A.0  D    0  WORK14
FILE.17                1        21  FULL  RD3A.0  D    0  WORK14
        NUMBER OF FILES=                21 GLOBAL SIZE=                174 START=2016-09-17 12:04:53
END= 2016-09-17 12:05:38
    
```

Die vorletzte Zeile der letzten Report-Seite enthält folgende Informationen:

NUMBER OF FILES=	Anzahl der bearbeiteten Dateien und Jobvariablen
GLOBAL SIZE=	Summe der Größe aller Dateien und Jobvariablen
START=	Datum und Uhrzeit, an dem der Auftrag erstellt wurde
END=	Datum und Uhrzeit, an dem der Auftrag beendet wurde

Die letzte Seite des HSMS-Reports wird mit folgender Zeile abgeschlossen:

```

***      E N D   O F          HSMS V11.0          FULL  REPORT  *** 2016-09-17 12:05:38      ***
    
```

4.11 Jobvariable zur Auftragsüberwachung

Bei folgenden HSMS-Aktionsanweisungen kann eine Jobvariable zur Auftragsüberwachung angegeben werden (Operand CONTROL-JV innerhalb der Ablaufsteuerung unter OPERATION-CONTROL=*PARAMETERS(...)):

- ARCHIVE-FILES
- ARCHIVE-NODE-FILES
- BACKUP-FILES
- BACKUP-FILE-VERSIONS
- BACKUP-NODE-FILES
- COPY-EXPORT-SAVE-FILE
- COPY-NODE-SAVE-FILE
- COPY-SAVE-FILE
- EXPORT-FILES
- IMPORT-FILES
- MIGRATE-FILES
- MOVE-SAVE-FILES
- RECALL-MIGRATED-FILES
- REORGANIZE-VERSION-BACKUP
- REPAIR-CATALOG-BY-RESTORE
- REPLACE-SAVE-FILE-BY-RESTORE
- RESTORE-FILES
- RESTORE-LIBRARY-ELEMENTS
- RESTORE-NODE-FILES
- UPDATE-EXPORT-SAVE-FILE

Die Jobvariable ist keine Monitor-Jobvariable und wird deshalb ausschließlich von HSMS bearbeitet. Diese Jobvariable liefert dem Benutzer bei asynchroner Auftragsbearbeitung das Ergebnis wichtiger Bearbeitungsschritte in HSMS/ARCHIVE. Unter anderem liefert die Jobvariable Informationen über die Initialisierung einer CCOPY-Sitzung und über das automatische Duplizieren in ein Schattenarchiv.

4.11.1 Bearbeitung der Jobvariable durch HSMS

HSMS löscht den Inhalt der Jobvariable, während der Auftrag überprüft und angenommen wird. Wenn die Jobvariable noch nicht im Katalog eingetragen ist, legt sie HSMS neu an. Tritt zu diesem Zeitpunkt ein Fehler beim Zugriff auf die Jobvariable auf, dann wird der Auftrag zurückgewiesen.

Wenn HSMS zu einem späteren Zeitpunkt die Jobvariable aktualisieren will und dabei ein Zugriffsfehler auftritt, wird der Fehler protokolliert. Der Auftrag wird aber normal bearbeitet.

4.11.2 Inhalt der Jobvariable

Die Jobvariable setzt sich aus Feldern fester Länge zusammen. Während der Auftrag überprüft wird, wird jedes Feld mit Leerzeichen initialisiert.

Die Jobvariable zur Überwachung von HSMS-Aufträgen orientiert sich nicht an den Konventionen für Monitor-Jobvariablen. Insbesondere beginnt das Feld mit dem Auftragsstatus nicht auf Position 1 (siehe auch das Handbuch „Jobvariablen“ [23])

Feld	Pos.	Länge	Wert	Bedeutung
CCS INIT STATUS	1	1	Leerzeichen	Auftrag ohne aktivierte CCOPY-Funktion oder (noch) nicht komplett
			T	erfolgreich beendet
			A	abnormal beendet
AUTOMATIC DUPLICATION	2	1	Leerzeichen	Auftrag ohne automatisches Duplizieren in ein Schattenarchiv oder (noch) nicht gestartet
			S	gestartet
TSN SERVERTASK	3	4	txt	TSN des Servertasks
REQUEST DATE and TIME	7	14	dig	Datum und Uhrzeit des Auftrags Format: YYYYMMDDHHMMSS
SFID	21	14	dig	Sicherungsdatei-ID Format: YYYYMMDDHHMMSS
SVID	35	14	dig	Sicherungsversion-ID Format: YYYYMMDDHHMMSS
(LOCAL) STATUS	49	11	txt	siehe STATUS in nachfolgender Tabelle
(LOCAL) SUB-STATUS	60	17	txt	siehe SUBSTATUS in nachfolgender Tabelle
FULL-FROM-FULL-STATE	77	1	Leerzeichen	Sichern von Backup noch nicht gestartet
			2	Sichern von Backup läuft
(REMOTE) STATUS	78	11	txt	siehe folgende Tabelle
(REMOTE) SUB-STATUS	89	17	txt	siehe folgende Tabelle
SERVER-NAME	106	8	txt	BCAM-Name des Hosts, der momentan den Auftrag bearbeitet
CCOPY SESSION-ID	114	8	txt	Identifikator des CCOPY-Sicherungslaufs

Die folgende Tabelle zeigt die möglichen Werte für STATUS und SUBSTATUS:

STATUS	SUBSTATUS
INCOMPLETE	---
ACCEPTED	---
STARTED	COLLECTED START-ARCHIVE ARCHIVE-COMPLETED START-REPORT SENT-TO-MASTER MASTER-REPLIED MASTER-TIMEOUT MASTER-NO-CONNECT IN-TRANSMIT SENT-TO-BACK-SERV BACK-SERV-REPLIED BACK-SERV-NO-CONN BACK-SERV-TIMEOUT
COMPLETED	--- WITH-WARNINGS WITH-ERRORS
INTERRUPTED	MASTER-TIMEOUT MASTER-NO-CONNECT START-ARCHIVE START-REPORT MASTER-REPLIED BACK-SERV-REPLIED BACK-SERV-NO-CONN BACK-SERV-TIMEOUT
CANCELLED	---

Wenn HSMS die Bearbeitung eines Auftrags anhält, ist der Status des Auftrags entweder COMPLETED, INTERRUPTED oder CANCELLED (nähere Informationen über die möglichen Auftragszustände finden Sie im [Abschnitt „Auskunft über die Aufträge“](#)). Der Status, der an eine Jobvariable geschickt wird, kann dann von einer Prozedur abgefragt werden.

Wenn ein Auftrag, der sich im Status ACCEPTED oder STARTED befindet, von einer anderen Task gelöscht wird, dann wird die Jobvariable zur Auftragsüberwachung aktualisiert. Sie wird auf den Status COMPLETED und den Substatus WITH-ERRORS gesetzt.

Wenn Aufträge gelöscht werden, die sich in einem anderen Status befinden, wird die Jobvariable nicht aktualisiert.

1. CCS INIT STATUS

Informiert über den Status der Initialisierung einer CCOPY-Sitzung.

Dadurch lässt sich die CCOPY-Bearbeitung optimieren: Die Anwendung wird vor Eingabe der BACKUP-FILES-Anweisung geschlossen und sofort wieder geöffnet, nachdem die Initialisierung der CCOPY-Sitzung erfolgreich beendet wurde (noch vor der Auftragsannahme!).

2. AUTOMATIC DUPLICATION

Zeigt bei einer Sicherung oder Langzeitarchivierung mit automatischem Duplizieren an, dass die Sicherungsdatei des Originalarchivs vollständig ist und das automatische Duplizieren beginnt. Dies bedeutet, dass HSMS/ARCHIVE nicht mehr auf die katalogisierten Dateien zugreift. Deshalb können die Anwendungen, die auf diese Dateien zugreifen, erneut gestartet werden.

3. TSN SERVERTASK

TSN des Servertasks, der den Auftrag behandelt.

4. REQUEST DATE and TIME

Diese Information kann für HSMS-Anweisungen hilfreich sein, bei denen über das Datum und die Uhrzeit ein Auftrag ausgewählt werden kann, z.B. SHOW-REQUESTS, DELETE-REQUESTS.

5. SFID

ID der Sicherungsdatei, die vom Auftrag vergeben wird. Sie ist nur bei folgenden HSMS-Anweisungen vorhanden: ARCHIVE-FILES, ARCHIVE-NODE-FILES, BACKUP-FILES, BACKUP-FILE-VERSIONS und BACKUP-NODE-FILES.

6. SVID

ID der Sicherungsversion, die vom Auftrag vergeben wird. Sie ist nur bei folgenden HSMS-Anweisungen vorhanden: ARCHIVE-FILES, ARCHIVE-NODE-FILES, BACKUP-FILES, BACKUP-FILE-VERSIONS und BACKUP-NODE-FILES.

7. (LOCAL) STATUS

Aktueller Zustand des Auftrags. Dieses Feld kann beispielsweise für S-Prozeduren verwendet werden, die einen Auftrag überwachen.

Siehe auch HSMS-Anweisung SHOW-REQUESTS: STATUS („HSMS Bd. 2“ [\[1\]](#))

8. (LOCAL) SUB-STATUS

Liefert zusätzliche Informationen über den Zustand des Auftrags.

Siehe auch HSMS-Anweisung SHOW-REQUESTS: SUBSTATUS („HSMS Bd. 2“ [\[1\]](#))

9. FULL-FROM-FULL-STATE

Dieser Status betrifft das Sichern mit FROM=*LATEST-BACKUP-OR-S0. Das Feld enthält eine 2, wenn das Sichern von S0 abgeschlossen ist und das Sichern von früheren Sicherungen gestartet wurde.

10. (REMOTE) STATUS

Aktueller Zustand des Auftrags auf dem fernen Server.

Für lokale Aufträge hat dieses Feld keine Bedeutung.

11. (REMOTE) SUB-STATUS

Liefert zusätzliche Informationen über den Zustand des Auftrags auf dem fernen Server.

Für lokale Aufträge hat dieses Feld keine Bedeutung.

12. SERVER-NAME

BCAM-Name des fernen Servers, der den Auftrag bearbeitet.

Für lokale Aufträge hat dieses Feld keine Bedeutung.

13. CCOPY-SESSION-ID

Dieses Feld ist der Identifikator für einen CCOPY-Sicherungslauf mit BACKUP-FILES CURRENT-COPY=YES oder BACKUP-FILE-VERSIONS BACKUP-FILES CURRENT-COPY=YES.

Die folgende Tabelle gibt wichtige Informationen darüber, wo und wann der Inhalt der Jobvariable verfügbar ist.

Feld	Wo befindet sich die Information?	Wann ist die Information verfügbar?
CCS INIT STATUS	Benutzerseite	während der Annahme des Auftrags
AUTOMATIC DUPLICATION	Serverseite	nach dem 1. Sichern
TSN SERVERTASK	Serverseite	nach dem Starten des Auftrags
REQUEST DATE and TIME	Benutzerseite	nach der Annahme des Auftrags
SFID	Serverseite	nach dem Sichern
SVID	Serverseite	nach dem Sichern
(LOCAL) STATUS	Benutzerseite	ändert sich laufend
(LOCAL) SUB-STATUS	Benutzerseite	ändert sich laufend
FULL-FROM-FULL-STATE	Serverseite	nach Differenzsicherung
(REMOTE) STATUS	Serverseite	ändert sich laufend
(REMOTE) SUB-STATUS	Serverseite	ändert sich laufend
SERVER-NAME	Serverseite	nach dem Starten des Auftrags
CCOPY-SESSION-ID	Benutzerseite	ab Annahme des Auftrags

4.11.3 Jobvariable bei Shared-Pubsets

Einige Aktionsanweisungen können auch im Master-Modus oder am Backup-Server bearbeitet werden (siehe [Abschnitt „Arbeiten mit Shared-Pubsets“](#)). Dies bedeutet, dass der Auftrag derzeit von einem anderen Host bearbeitet wird, nämlich vom Master des Shared-Pubsets oder vom Backup-Server, der von diesem Auftrag betroffen ist.

Wenn für einen solchen Auftrag im Master-Modus eine Jobvariable zur Auftragsüberwachung verwendet wird, werden einige Teile der Jobvariable vom Slave-Share gesetzt, einige vom Master-Share. Wenn der Backup-Server einen solchen Auftrag ausführt, werden einige Teile der Jobvariable vom lokalen System gesetzt, einige vom Backup-Server. In einem solchen Fall sollte man eine Jobvariable verwenden, die auf beiden Seiten verfügbar ist. Am Besten verwendet man eine Jobvariable auf dem Pubset, der bereits von diesem Auftrag betroffen ist (der Pubset, auf dem die Dateien gesichert, restauriert, zurückgeholt, ... werden sollen).

4.12 Aliasnamen für BS2000-Dateien und Jobvariablen

Dieser Abschnitt beschreibt die Auswirkungen des Softwareprodukts ACS (**A**lias **C**atalog **S**ervice) auf HSMS. ACS ist ein Subsystem des BS2000, mit dessen Hilfe Aliasnamen für BS2000-Dateien und Jobvariablen verwaltet werden. Ein Aliasname ist ein fast beliebiger Name, den der Benutzer an Stelle des Datei- bzw. JV-Namens verwenden kann.

Um mit Aliasnamen arbeiten zu können, muss der Systemverwalter das ACS-Subsystem gestartet haben. Außerdem muss für den gerade aktuellen Task ein Aliaskatalog angelegt sein, aus dem eindeutig die Zuordnung von Aliasnamen zu Datei- bzw. JV-Namen hervorgeht.

HSMS akzeptiert Aliasnamen im Operanden FILE-NAMES bzw. JV-NAMES der betreffenden HSMS-Anweisungen. Aliasnamen werden auch für die Namen von Directory-Dateien akzeptiert.

Intern verwendet HSMS nur Datei- bzw. JV-Namen vom Katalog und keine Aliasnamen. Deshalb werden nur Dateinamen in die HSMS-Auftragsdatei oder in die ARCHIVE-Checkpointdatei geschrieben.

Die Änderung eines Aliasnamens wird nach dem Wirksamwerden einer HSMS-Anweisung ignoriert. Dies gilt ebenso für Änderungen, die vor dem Wiederaanlauf eines unterbrochenen Auftrags durchgeführt wurden.

In Reporten wird der Aliasname im Anweisungsteil des Benutzers, der Datei- bzw. JV-Name im Ergebnisteil ausgegeben.

Nähere Informationen zum Arbeiten mit Aliasnamen und zur Verwaltung des Aliaskatalogs finden Sie im Handbuch „Einführung in das DVS“ [4].

5 Verwaltung von HSMS

Im laufenden Betrieb hat der HSMS-Verwalter eine Reihe von Aufgaben zu erledigen. Ein Teil dieser Aufgaben ist bereits in den Kapiteln "[HSMS-Archive](#)" und "[Funktionen von HSMS](#)" beschrieben, wie z.B. das Einrichten von Standard-Systemarchiven und die Festlegung der Bandverarbeitungszeiten.

In diesem Kapitel sind noch einige zusätzliche Aufgaben beschrieben. Dazu gehört die Einteilung der Pubsets in eine Speicherhierarchie – entsprechend den Anforderungen des jeweiligen Systems.

Außerdem ist in diesem Kapitel die Steuerung der Verdrängung beschrieben, weil hier die Steuerungsmöglichkeiten vielfältig sind. Besonders die Verwaltung der Migrationsarchive erfordert Aufwand durch den HSMS-Verwalter, der bei den anderen Grundfunktionen nicht gegeben ist. Zusätzlich wird der Zusammenhang zwischen Datensicherung und Verdrängung angesprochen.

Aspekte des Datenschutzes werden in einem eigenen Abschnitt behandelt.

Dieses Kapitel wird durch ein Beispiel abgeschlossen. Es erläutert, wie eine Speicherhierarchie eingerichtet wird und die Systemarchive für eine BS2000-Konfiguration erzeugt und zugewiesen werden.

5.1 Betriebsweisen von HSMS

HSMS kann in verschiedenen Betriebsweisen gefahren werden, die der Systemverwalter bei der HSMS-Anweisung START-HSMS bzw. der HSMS-Verwalter mit der HSMS-Anweisung MODIFY-HSMS-PARAMETERS festlegt.

5.1.1 Definitions-/Auskunfts-Betrieb

Im Definitions-/Auskunfts-Betrieb (OPERATIONAL-MODUS=*DEFINE-SHOW) sind alle AuskunftsFunktionen (SHOW-Anweisungen) zugelassen. Außerdem können die HSMS-Steuer- und Pubsetparameter geändert sowie Archive eingerichtet und geändert werden. Aktionsanweisungen werden abgewiesen. In diesem Modus kann man sich mit HSMS vertraut machen.

5.1.2 Simulations-Betrieb

Im Simulations-Betrieb (OPERATIONAL-MODUS=*SIMULATION) sind alle HSMS-Anweisungen zugelassen. Die Aktionsanweisungen werden aber nur simuliert, d.h. die Syntax der HSMS-Anweisungen wird geprüft, die Aufträge werden erzeugt und in die Auftragsdatei geschrieben.

Durch die Servertasks erfolgt aber nur eine Art Pseudo-Bearbeitung. Die Aufträge erhalten zwar den Status COMPLETED; ARCHIVE stößt aber keine Aktionen an: Datenmengen werden nicht manipuliert. In diesem Modus kann man sich mit den Möglichkeiten der Aktionsanweisungen vertraut machen, ohne auf Benutzerdateien tatsächlich zuzugreifen.

Mit der HSMS-Anweisung MODIFY-ARCHIVE sollte man im Simulations-Betrieb sehr vorsichtig umgehen, da einige Tätigkeiten nicht nur simuliert, sondern tatsächlich ausgeführt werden (z.B. die Freigabe von Sicherungen mit //MODIFY-ARCHIVE SAVE-FILES= *DELETE).

Aktionsanweisungen von aktiven Knoten werden zurückgewiesen, wenn HSMS im Simulations-Betrieb läuft.

5.1.3 Operationeller Betrieb

Im operationellen Betrieb (OPERATIONAL-MODUS=*OPERATION) wird HSMS in vollem Funktionsumfang betrieben. Dies ist die normale Betriebsweise von HSMS nach der Einführungsphase.

5.2 Speicherhierarchie in einer Umgebung mit SF-Pubsets verwalten

HSMS realisiert seine Funktionen im Rahmen einer dreistufigen Speicherhierarchie. In diese Speicherhierarchie werden die Platten- und Bandspeicher entsprechend ihrer Verfügbarkeit, Zugriffszeit und den Kosten eingeteilt (siehe [Abschnitt „Speicherhierarchie“](#)).

Speicherebene S2

Die Speicherebene S2 muss in HSMS nicht ausdrücklich eingerichtet werden. Sie ist standardmäßig in einem HSMS-System definiert.

Die Voreinstellung für Datenträger, die für Schreibaufträge auf S2 angefordert werden, kann festgelegt werden mit:

```
//MODIFY-HSMS-PARAMETERS DEFAULT-HSMS-STORAGE=*PAR -  
// (S2-DEVICE-TYPE=<c-string>)
```

HSMS unterstützt Datenträger der Klasse „TAPE“, LTO-Magnetbandkassetten, sowie die vom Archivsystem Eternus CS emulierten Magnetbänder (siehe [Abschnitt „Datenträger auf S2“](#)).

5.2.1 SF-Pubsets verwalten

Eine wichtige Verwaltungseinheit für HSMS ist der Pubset, ein Satz gemeinschaftlicher Plattenspeicher mit einem gemeinsamen Katalog, definiert durch seine Pubset-Kennung. MPVS-Systeme (Multiple-Public-Volume-Sets) unterstützt HSMS in vollem Umfang.

HSMS unterstützt auch Shared-Pubset-Systeme: Datensicherung, Versions-Backup, Archivierung und Verdrängung sind in vollem Umfang möglich; allerdings sind Einschränkungen bei der Auswahl von Dateien zu beachten (siehe [Abschnitt „Arbeiten mit Shared-SF-Pubsets“](#)).

Nachdem HSMS installiert wurde, ist nur der Home-Pubset unter der Verwaltung von HSMS. Alle anderen Pubsets sind im Zustand UNDEFINED und haben keinen Eintrag in der HSMS-Steuerdatei.

Ein Pubset in einem MPVS-System kann entweder außerhalb der HSMS-Kontrolle in einem undefinierten Zustand bleiben oder vom HSMS-Verwalter einer der beiden Online-Speicherebenen S0 und S1 zugeordnet werden (siehe nächste Seite).

5.2.1.1 Pubset außerhalb der HSMS-Kontrolle

Auch für einen Pubset, der nicht unter HSMS-Kontrolle genommen wurde, d.h. keiner Speicherebene zugeordnet wurde, sind HSMS-Funktionen anwendbar, wenn der Pubset im System verfügbar ist.

Die Voreinstellung erlaubt Datensicherung und Langzeitarchivierung in die globalen, in der HSMS-Steuerdatei festgelegten Standard-Systemarchive. Die Verdrängung und Versions-Backup von [Dateien](#) ist aber für diesen Pubset grundsätzlich nicht möglich.

Dateien, die auf einem Pubset außerhalb von HSMS liegen, können mit der Funktion Concurrent Copy nur gesichert werden, wenn auf die Kennung SYSHSMS vom Pubset aus zugegriffen werden kann. Auf der Kennung SYSHSMS werden nämlich die Arbeitsdatei für Concurrent Copy und die Metadateien erstellt.

5.2.1.2 Pubset den Speicherebenen S0 und S1 zuordnen

Ein Pubset wird der Speicherebene S0 zugeordnet, wenn für die Dateien diesen Pubsets Verdrängung möglich sein soll oder wenn pubset-spezifische, von der systemweiten Festlegung abweichende Standard-Systemarchive vereinbart werden sollen.

Wenn Versions-Backups der Dateien auf dem Pubset durchzuführen sind, muss der Pubset als S0 definiert werden. Bitte beachten Sie, dass nur pubset-spezifische Standard-Systemarchive für Versions-Backup angegeben werden können. Es gibt keine globalen Archive für Versions-Backup.

Einem Pubset der Speicherebene S0 kann als Hintergrundebene ein S1-Pubset zugeordnet werden, auf den Dateien gesichert oder verdrängt werden können. Ein Pubset wird als globaler S1-Pubset vereinbart mit:

```
//MODIFY-HSMS-PARAMETERS DEFAULT-HSMS-STORAGE=*PAR -  
// (S1-PUBSET-ID=<S1-pubset-id>)
```

Dieser globale Pubset dient allen S0-Pubsets als Hintergrundebene, falls nicht ein abweichender S1-Pubset festgelegt wird:

```
//MODIFY-PUBSET-PARAMETERS PUBSET-ID=<S0-pubset-id>, -  
// STORAGE-LEVEL=*S0(S1-PUBSET-ID=<S1-pubset-id>)
```

Ein S1-Pubset ist in einem System, das unter HSMS-Verwaltung steht, nicht zwingend nötig: Bei allen Grundfunktionen können auch die Speicherebene S2 oder Privatplatten verwendet werden.

5.2.1.3 SM-Pubset der Speicherebene S1 zuordnen

Da die Größe eines SF-Pubsets auf 4TB beschränkt ist, kann es vorkommen, dass ein SF-Pubset nicht für die Sicherung und Verdrängung auf die Speicherebene S1 ausreicht. In diesem Fall kann ab HSMS V10.0 (Parameter SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE) auch ein SM-Pubset als Speicherebene S1 zugeordnet werden. Dabei werden mit Ausnahme von Control-Volume-Set und S1-Volume-Set alle Volume-Sets des SM-Pubsets genutzt. Ein solcher SM-Pubset wird als S1-SM-Pubset bezeichnet. Die maximal mögliche Größe eines SM-Pubsets liegt bei nahezu 1000TB.

Damit der Control-Volume-Set eines S1-SM-Pubsets nicht mit Sicherungsdateien belegt werden kann, muss die Nutzung des Control-Volume-Sets zuvor folgendermaßen eingeschränkt werden:

```
/MODIFY-PUBSET-RESTRICTIONS PUBSET=<cat_id of S1-SM-pubset> -
/      ,PUBSET-TYPE=*SYSTEM-MANAGED(VOLUME-SET=<control-volume-set> -
/      ,RESTRICTION=*NEW-FILE-ALLOCATION(MODE=*PHYSICAL-ONLY))
```

i Es wird dringend empfohlen, dass ein SM-Pubset, der als S1-SM-Pubset verwendet werden soll, ausschließlich aus Volume-Sets im NK2-Format besteht.

S1-SM-Pubsets können nicht in einer SM-Umgebung verwendet werden, d.h. ein S1-SM-Pubset kann keinem anderen SM-Pubset als Speicherebene S1 zugewiesen werden.

Sicherung auf ein S1-SM-Pubset

Die Sicherung auf einen S1-SM-Pubset erfolgt mit den Anweisungen BACKUP-FILES, BACKUP-FILE-VERSIONS, ARCHIVE-FILES, COPY-SAVE-FILE, MOVE-SAVE-FILE, REORGANIZE-VERSION-BACKUP oder MIGRATE-FILES mit der Angabe TO-STORAGE=*S1-STORAGE-LEVEL, wenn ein SM-Pubset systemglobal oder für einzelne Pubsets als Speicherebene S1 zugeordnet ist.

Die Sicherungsdatei für einen solchen Sicherungslauf wird mit folgendem Namen eingerichtet:

```
ARCHIVE.SAVE.FILE.date.time.subsave#.seq#
```

Legende:

date.time Erzeugungszeitpunkt im Format yymmdd.hhmmss

subsave# Subsave-Nummer des Laufs, der diese Sicherungsdatei erstellt hat

seq# Laufende Nummer der Sicherungsdatei mit demselben Zeitstempel (0..F)

Falls die Kapazität des Volume-Sets, auf dem die Sicherungsdatei angelegt wird, für die Sicherung nicht ausreicht, wird eine weitere Sicherungsdatei auf einem anderen Volume-Set des S1-SM-Pubset angelegt.

Deren Name unterscheidet sich von dem der ursprünglichen Sicherungsdatei nur durch die um 1 erhöhte laufende Nummer *seq#*.

Eine Subtask kann maximal 16 derartige Sicherungsdateien anlegen.

SM-Pubset als Speicherebene S1 einrichten

Ein SM-Pubset kann nur dann als Speicherebene S1 eines SF-Pubsets verwendet werden, wenn der globale HSMS-Parameter SAVE-FILE-PROCESSING auf *HSMS-V10-COMPATIBLE eingestellt ist (Anweisung MODIFY-HSMS-PARAMETERS). Falls SAVE-FILE-PROCESSING=*HSMS-V9-COMPATIBLE eingestellt ist, wird der Versuch, auf einen S1-SM-Pubset zu sichern, abgewiesen.

i Das Ändern dieser Einstellung hat Auswirkungen auf die Funktionalität von HSMS. Näheres siehe [Abschnitt „Verarbeitungsmodus für Sicherungsdateien“](#).

Um einen SM-Pubset als Speicherebene S1 einzurichten, sind folgende Schritte erforderlich:

- Den SM-Pubset unter HSMS-Kontrolle bringen (Anweisung CREATE-SM-PUBSET-PARAMETERS).
- Den SM-Pubset als Speicherebene S1 definieren (Anweisung MODIFY-PUBSET-PARAMETERS).
- Den SM-Pubset global oder pubset-spezifisch als Speicherebene S1 hinzufügen (Anweisung MODIFY-HSMS-PARAMETERS oder MODIFY-PUBSET-PARAMETERS).

5.2.1.4 Pubset unter Verwaltung nehmen und Parameter festlegen

Dem HSMS-Verwalter steht zum Einbringen eines Pubsets unter HSMS-Verwaltung und zum Einstellen der Pubset-Parameter die HSMS-Anweisung MODIFY-PUBSET-PARAMETERS zur Verfügung. Alle vorgenommenen Änderungen gelten vom Zeitpunkt der Anweisungseingabe in der laufenden HSMS-Session sowie in den folgenden.

Ein Pubset kann nur unter HSMS-Verwaltung genommen werden, wenn folgende Bedingungen erfüllt sind:

- Der Pubset muss lokal verfügbar sein, d.h. er muss bereits mit IMPORT-PUBSET importiert worden sein.
- Auf dem Pubset muss die Benutzerkennung SYSHSMS eingerichtet sein, bevor er einer Speicherebene zugewiesen werden kann.
- Ein Pubset muss als S1-Pubset eingerichtet sein, bevor er einem S0-Pubset als S1-Pubset zugeordnet werden kann.

Wenn ein Pubset unter HSMS-Verwaltung genommen werden soll, müssen einige Besonderheiten des Pubsets berücksichtigt werden. Deshalb sollte diese Tätigkeit nur zusammen mit dem Systemverwalter durchgeführt werden.

Ein Pubset erhält nur dann einen Eintrag in der HSMS-Steuerdatei, wenn wenigstens bei einem Operanden vom Standardwert abgewichen wird.

Wenn ein bisher nicht unter HSMS-Verwaltung stehender Pubset unter HSMS-Verwaltung genommen wird, gelten folgende voreingestellten Werte:

Operandenwert	Voreingestellter Wert für Pubsets unter HSMS-Verwaltung
STORAGE-LEVEL	*UNDEFINED
SYSARCHIVE	globale Einstellung aus der HSMS-Steuerdatei
SYSBACKUP	globale Einstellung aus der HSMS-Steuerdatei
SYSVERSION	*UNDEFINED
falls S0:	
– S1-PUBSET-ID	globale Einstellung aus der HSMS-Steuerdatei
– SYSMIGRATE	globale Einstellung aus der HSMS-Steuerdatei
– MIGRATION	*ALLOWED

Ein Pubset kann einer Speicherebene zugeordnet werden, und es können dem Pubset Standard-Systemarchive für Datensicherung, Versions-Backup und Langzeitarchivierung zugeordnet werden.

Für einen S0-Pubset lassen sich darüber hinaus Parameter für die Verdrängung vereinbaren (siehe [Abschnitt „Steuerung der Verdrängung und Verwaltung der Migrationsarchive“](#)).

Hinweise

- Sicherungsdateien auf einem S1-Pubset werden unter derselben Benutzerkennung angelegt wie das Directory des Archivs, das die Sicherungsdatei verwaltet. Deshalb muss diese Benutzerkennung auch auf dem Pubset, das der Speicherebene S1 zugeordnet ist, existieren.
- Das Überschreiten des Speicherplatzlimits sollte erlaubt werden (PUBLIC-SPACE-EXCESS).

5.2.1.5 Auskunft über Parameter und Nutzung der Pubsets

Mit der HSMS-Anweisung SHOW-PUBSET-PARAMETERS kann sich der HSMS-Verwalter die Parameter aller unter HSMS-Verwaltung stehenden Pubsets ausgeben lassen (siehe Handbuch „HSMS Bd. 2“ [1], HSMS-Anweisung SHOW-PUBSET-PARAMETERS).

Mit der HSMS-Anweisung SHOW-PUBSET-USAGE kann sich der HSMS-Verwalter über die Belegung aller verfügbaren Pubsets informieren, also nicht nur der unter HSMS-Verwaltung stehenden (siehe Handbuch „HSMS Bd. 2“ [1], HSMS-Anweisung SHOW-PUBSET-USAGE).

Diese Übersichten sind besonders hilfreich bei der Verwaltung der Pubsets und bei der rechtzeitigen Verdrängung oder Archivierung, bevor es zur Speicherplatzsättigung auf einem Pubset kommt (siehe auch [Abschnitt „Steuerung der Verdrängung und Verwaltung der Migrationsarchive“](#)).

5.2.1.6 Pubset aus HSMS-Verwaltung entlassen

Nur ein Pubset, für den von den Voreinstellungen abweichende Parameter vereinbart wurden, erhält einen Eintrag in der HSMS-Steuerdatei. Der Eintrag für einen Pubset in der HSMS-Steuerdatei kann wieder gelöscht werden, indem alle Parameter auf die ursprünglichen Voreinstellungen zurückgesetzt werden. Der Eintrag für den Pubset wird zwar sofort aus der Steuerdatei gelöscht, aber während der laufenden HSMS-Session bei den entsprechenden HSMS-Anweisungen noch ausgegeben.

5.2.2 Maßnahmen bei besonderen Situationen

- Restore nach einem S0-Crash
- Restore nach einem Crash auf S1 oder S2
- Reorganisation der Speicherebene S1
- Reorganisation der Speicherebene S2
- Benutzererkennung auf einen anderen Pubset verlegen
- Benutzer- oder Katalogkennung reorganisieren

5.2.2.1 Restore nach einem S0-Crash

Dabei sind zwei Fälle zu unterscheiden:

1. Die Speicherebenen S1 und S2 sind unbeschädigt

Für jeden beschädigten Pubset, z.B. Pubset A, sind folgende Anweisungen erforderlich:

```
//RESTORE-FILES FILE-NAMES=:A:$*., -  
//  MIGRATED-FILES=*CATALOG-ONLY, -  
//  JV-NAMES=:A:$*., -  
//  ...
```

Die normale Bearbeitung kann auf den Pubsets fortgesetzt werden, die nicht beschädigt sind.

Wenn der Home-Pubset beschädigt ist, muss zuerst die Prozedur, die die Archive definiert, restauriert und aufgerufen werden. Die Directory-Datei für Sicherung muss ebenfalls restauriert werden.

2. Die Speicherebenen S1 und S2 sind auch beschädigt

Zuerst muss die Verarbeitungsebene S0 restauriert werden. Anschließend müssen die Speicherebenen S1 und S2 restauriert werden (siehe dazu den folgenden Abschnitt).

5.2.2.2 Restore nach einem Crash auf S1 oder S2

Für jeden S0-Pubset, der vom Crash betroffen ist (z.B. A), sind folgende Anweisungen erforderlich:

```
//REPAIR-CATALOG-BY-RESTORE S0-PUBSET-ID=A, -  
// TO-STORAGE=..., -  
// ARCHIVE-NAME=*SYSBACKUP, -  
// ...
```

5.2.2.3 Reorganisation der Speicherebene S1

Nach dem Löschen oder Zurückholen verdrängter Dateien ist der Speicherplatz in der Sicherungsdatei von ARCHIVE nicht freigegeben. Dadurch wird nach einiger Zeit viel Speicherplatz auf S1 verschwendet. Mit der Anweisung //SHOW-PUBSET-USAGE ... INFORMATION=*REUSABLE-S1-SPACE kann sich der HSMS-Verwalter über den auf der S1-Ebene freizugebenden Speicherplatz informieren. Wenn eine dezentrale Verwaltung benutzt wird, sind für jeden S1-Pubset folgende Anweisungen erforderlich:

```
//MIGRATE-FILES FROM-STORAGE=*S1-STORAGE-LEVEL -  
// (S1-PUBSET-ID=..., -  
//   ..., -  
// TO-STORAGE=S1-STORAGE-LEVEL, -  
// ARCHIVE-NAME=<Name des in Beziehung stehenden Archivs>)
```

5.2.2.4 Reorganisation der Speicherebene S2

Nach dem Zurückholen einer verdrängten Datei wird der Speicherplatz auf den von ARCHIVE erstellten Datenträgern nicht freigegeben. Dadurch wird nach einiger Zeit viel Speicherplatz auf S2 verschwendet. Mit der nachfolgenden Anweisung werden die Dateien von der Speicherebene S2 auf S2 verdrängt und dabei reorganisiert. Realisiert wird dies durch Kopieren in eine neue Sicherungsdatei, in der nur die noch migrierten Dateien enthalten sind. Die ursprünglichen Sicherungsdateien werden dabei automatisch gelöscht, unabhängig von ihrer Schutzfrist. Für jedes verwendete Migrations-Archiv sind dabei folgende Anweisungen erforderlich:

```
//MIGRATE-FILES FROM-STORAGE=*S2-STORAGE-LEVEL -  
// (SAVE-FILE-ID=..., -  
//   UNUSED-SPACE=..., -  
//   ..., -  
//   ARCHIVE-NAME=<Name des in Beziehung stehenden Archivs>)
```

Die Standard-Sicherungsdatei wird dabei nicht berücksichtigt.

5.2.2.5 Benutzererkennung auf einen anderen Pubset verlegen

Wenn eine Benutzererkennung von einem Pubset auf einen anderen Pubset verlegt werden soll – unter Berücksichtigung von verdrängten Dateien – empfehlen wir folgendes Vorgehen:

- Backup in ein temporäres Archiv (mit den Daten der verdrängten Dateien);
- Der Benutzererkennung auf dem Ziel-Pubset ausreichend Speicherplatz vor dem Transfer zuweisen;
- RESTORE mit Umbenennen der Katalogkennung (verdrängte Dateien mit Daten); die Dateien erhalten ein neues CREATION-DATE, LAST-ACCESS-DATE usw.
- Nach dem Backup der Dateien die ursprünglich verdrängten Dateien erneut verdrängen;
- Löschen des temporären Archivs;
- Zurücksetzen des Speicherplatz-Limits.

5.2.2.6 Benutzer- oder Katalogkennung reorganisieren

Wechsel der Benutzerkennung bei gleicher Katalogkennung:

Unterschieden werden drei Varianten:

1. Restore mit Umbenennung der Benutzerkennung aus Archiven mit unverändertem Directory
Diese Variante hat den Nachteil, dass bei Restore aus Backup-Archiven unterschiedlich gearbeitet werden muss für Sicherungsdateien: vor dem Wechsel der Benutzerkennung mit und nach dem Wechsel ohne Umbenennung. Auch bei SHOW-ARCHIVE bleiben die vor dem Wechsel gesicherten Dateien der alten Benutzerkennung zugeordnet.
2. Einrichten von neuen Archiven mit neuem Directory für die neue Kennung und Umkopieren von Sicherungsdateien aus dem altem Archiv (vor Wechsel der Kennung) in ein neues Archiv mit Umbenennung der Benutzerkennung
Diese Variante ist nur möglich für den HSMS-Administrator. Bei einem solchen Umkopieren von Sicherungsversionen aus Backup-Differenzsicherungen muss darauf geachtet werden, dass auch die zugrunde liegende Vollsicherung so kopiert wird.
3. In dem unveränderten Backup-Archiv nach Wechsel eine aktuelle Vollsicherung erstellen durch Backup mit FROM-LATEST-OR-S0 Umbenennung der Benutzerkennung Diese Variante ist ebenfalls nur möglich für den HSMS-Administrator.

Allgemein gilt

Migrierte Dateien müssen vor dem Wechsel der Benutzerkennung nach S0 zurückgeholt werden.

Wechsel der Katalogkennung (ohne Änderung des Pubset-Typs)

Unterschieden werden vier Varianten:

1. Restore mit Umbenennung der Katalogkennung aus Archiven mit unverändertem Directory
Diese Variante hat den Nachteil, dass bei Restore aus Backup-Archiven unterschiedlich gearbeitet werden muss für Sicherungsdateien: vor dem Wechsel der Katalogkennung mit und nach dem Wechsel ohne Umbenennung. Auch bei SHOW-ARCHIVE bleiben die vor dem Wechsel gesicherten Dateien der alten Katalogkennung zugeordnet.
2. Einrichten von neuen Archiven mit neuem Directory für die neue Katalogkennung und Umkopieren von Sicherungsdateien aus dem altem Archiv (vor Wechsel der Katalogkennung) in ein neues Archiv mit Umbenennung der Katalogkennung
Diese Variante ist nur möglich für den HSMS-Administrator. Bei einem solchen Umkopieren von Sicherungsversionen aus Backup-Differenzsicherungen muss darauf geachtet werden, dass auch die zugrunde liegende Vollsicherung so kopiert wird.
3. In dem unverändertem Backup-Archiv nach Wechsel eine aktuelle Vollsicherung erstellen durch Backup mit FROM-LATEST-OR-S0 Umbenennung der Katalogkennung Diese Variante ist ebenfalls nur möglich für den HSMS-Administrator.
4. Umstellen des Directory eines Archivs auf neue Katalogkennung mit DIRCONV Diese Variante ist möglich, wenn das Directory nur Einträge mit der alten Katalogkennung enthält (für SM-Pubset oder für dezentrale Organisation bei SF-Pubsets).

Der Wechsel von alter auf neue Katalogkennung ist bei den Sicherungsdateien auf Platte ohne Problem möglich.

Verfahren für migrierte Dateien

Bei den Varianten 1 bis 3 müssen vor dem Wechsel der Katalogkennung alle migrierten Dateien nach S0 zurückgeholt werden.

Bei Variante 4 reicht es (neben der DIRCONV-Umstellung auf neue Katalogkennung für das Migrations-Directory) von den migrierten Dateien nur den Katalogeintrag zu restaurieren – nicht die Dateien.

5.3 Speicherhierarchie in einer Umgebung mit SM-Subsets verwalten

HSMS realisiert seine Funktionen im Rahmen einer dreistufigen Speicherhierarchie. In diese Speicherhierarchie werden die Platten- und Bandspeicher entsprechend ihrer Verfügbarkeit, Zugriffszeit und den Kosten eingeteilt (siehe [Abschnitt „Speicherhierarchie“](#)).

5.3.1 Bearbeitung von SM-Pubsets

Zur Bearbeitung von SM-Pubsets bietet HSMS eine komfortable Benutzerschnittstelle an. Da für die Sicherung, Archivierung und Verdrängung einige Metadaten auf dem SM-Pubset benötigt werden, muss zum Durchführen dieser Tätigkeiten der SM-Pubset unter HSMS-Kontrolle sein. SM-Pubsets, die sich noch nicht unter HSMS-Kontrolle befinden, müssen zuerst mit der Anweisung CREATE-SM-PUBSET-PARAMETERS unter HSMS-Kontrolle gebracht werden.

Die folgende Tabelle gibt einen Überblick, wie SM-Pubsets bearbeitet werden.

	SM-Pubsets außer HSMS-Kontrolle	SM-Pubsets unter HSMS-Kontrolle
HSMS	Keine Verdrängung: keine Verdrängung nach S1/S2; keine Verdrängung nach S0; kein Zurückholen	Volle Bearbeitung einschließlich: Verdrängung nach S1/S2; Verdrängung nach S0; Zurückholen
	keine Sicherung; kein Versions-Backup	Versions-Backup
	EXPORT/IMPORT möglich	
	ARCHIVE-FILES/RESTORE-FILES mit einem Langzeitarchiv des Hosts möglich	
	Ein Restore mit Umbenennen der Katalogkennung ist aus jedem Archiv möglich	Ein Restore mit Umbenennen der Katalogkennung ist aus jedem Archiv möglich
ARCHIVE	möglich	möglich

5.3.2 Umfang der HSMS-Operationen

Die folgende Tabelle zeigt für jede Funktion:

- in der ersten und zweiten Spalte den Ort, wo sich die Metadaten (Archivdefinitionen und Aufträge) befinden.
- in der dritten Spalte, ob nach einem Host-Wechsel die Aufträge erneut gestartet werden können.
- in der letzten Spalte die Umgebung, die für Dateien und Jobvariablen angegeben werden kann.

	Archivdefinition in der Steuerdatei	Auftrag in der Auftragsdatei	Restart nach Host- Wechsel?	Umfang der Dateien/JV
Sicherung von SF-Pubsets	auf dem Host	auf dem Host	nicht möglich	alle SF-Pubsets
Sicherung eines SM-Pubsets	auf dem SM- Pubset	auf dem SM- Pubset	möglich	dieser SM- Pubset
Versions-Backup eines SF- Pubsets	auf dem Host	auf dem Host	nicht möglich	dieser SF- Pubset
Versions-Backup eines SM- Pubsets	auf dem SM- Pubset	auf dem SM- Pubset	möglich	dieser SM- Pubset
Verdrängung von SF-Pubsets	auf dem Host	auf dem Host	nicht möglich	alle SF-Pubsets
Verdrängung eines SM-Pubsets	auf dem SM- Pubset	auf dem SM- Pubset	möglich	dieser SM- Pubset
Langzeitarchivierung des Hosts	auf dem Host	auf dem Host	nicht möglich	alle Pubsets
Langzeitarchivierung von SM-Pubsets	auf dem SM- Pubset	auf dem SM- Pubset	möglich	dieser SM- Pubset
Export/Import	nicht relevant	immer auf dem Host	nicht möglich	alle Pubsets

5.3.3 Konvertieren eines SM-Pubsets

Damit HSMS einen SM-Pubset bearbeiten kann, muss er in zwei Schritten konvertiert werden:

1. SM-Pubset unter HSMS-Verwaltung bringen. Dies ist bei einem neuen, leeren SM-Pubset genauso notwendig, wie bei einem SM-Pubset, welcher durch Konvertierung aus anderen Pubsets erstellt wurde. Dazu steht die HSMS-Anweisung `CREATE-SM-PUBSET-PARAMETERS` zur Verfügung. Diese Anweisung überprüft den Status des SM-Pubsets und schaltet das Merkmal „unter HSMS-Kontrolle“ für diesen SM-Pubset ein. Die Anweisung `CREATE-SM-PUBSET-PARAMETERS` schließt die folgenden Aktionen ein:
 - Anlegen der Steuer- und Auftragsdateien des SM-Pubsets
 - Wenn eine S1-Speicherebene benötigt wird:
 - Explizit ein Volume-Set des SM-Pubset als S1-Volume-Set festlegen.
 - Oder mit `*ALL-HSMS-CONTROLLED` festlegen, dass alle Volume-Sets des SM-Pubsets unter HSMS-Kontrolle als erweiterte S1-Speicherebene genutzt werden können (ab HSMS V11.0 mit Parameter `SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE` und BS2000 OSD/BC V11.0A).
 - Bereitstellen der Generic Catalog Facility des SM-Pubsets
 - Einrichten der Systemarchive des SM-Pubsets für SM-Sicherung, SM-Verdrängung und SF-globale oder SM-Pubset-Langzeitarchivierung
 - Überprüfen und Löschen von alten HSMS-SF-Definitionen des neuen SM-Pubsets (falls ein SF-Pubset unter derselben Katalogennung in einen SM-Pubset konvertiert wird)

Hinweise

- Wenn der bereitgestellte S1-Volume-Set nicht geeignet ist, wird die Anweisung `CREATE-SM-PUBSET-PARAMETERS` zurückgewiesen.
- Das Einrichten der Systemarchive ist nicht obligatorisch.
- Es kann lediglich eine Teilmenge der Archivparameter angegeben werden. Die Standardwerte sind so eingerichtet, wie es bei der Anweisung `CREATE-SM-PUBSET-PARAMETERS` beschrieben ist. Wenn diese Standardwerte für Sie nicht passend sind, können die Systemarchive später mit der Anweisung `CREATE-ARCHIVE` eingerichtet und mit der Anweisung `MODIFY-SM-PUBSET-PARAMETERS` geändert werden.
- Wenn ein Fehler während des Einrichtens des Archivs auftritt, wird ein Warnhinweis ausgegeben, der Vorgang ansonsten aber fortgesetzt.

Wenn der SM-Pubset keine verdrängten Dateien enthält, ist das Konvertieren damit abgeschlossen.

2. Falls der SM-Pubset durch Konvertierung aus anderen Pubsets erstellt wurde und der SM-Pubset verdrängte Dateien enthält oder wenn einige vorhandene Archive von diesem SM-Pubset verwendet werden sollen, müssen diese Archivverzeichnisse mit `DIRCONV` zusammengefasst und konvertiert werden.

Im Folgenden werden einige mögliche Situationen erklärt, um das Konvertieren von SM-Pubsets in verschiedenen Konfigurationen zu veranschaulichen.

5.3.3.1 Erstellen eines SM-Pubsets ohne verdrängte Dateien

Ein SM-Pubset, der keine verdrängten Dateien enthält, soll konvertiert werden. In diesem Fall können neue Standardarchive erstellt werden.

Wenn alte Archivverzeichnisse, die vor dem Konvertieren erstellt wurden, wieder verwendet werden sollen, muss ein weiterer Fall betrachtet werden. Dabei gibt es folgende Möglichkeiten:

- Die Verdrängung auf eine Hintergrundebene wurde vorher nicht benutzt.
- Alle verdrängten Dateien wurden vor dem Erstellen des SM-Pubsets zurückgeholt.

Beispiele

```
//CREATE-SM-PUBSET-PARAMETERS SM-PUBSET-ID=S, SYSBACKUP=*PAR, SYSMIGRATE=*PAR
```

Der SM-Pubset S wird konvertiert. Die Standardarchive für Sicherung und Verdrängung sowie die zugehörigen Archivverzeichnisse werden mit einem Standardnamen erstellt. Die Archivverzeichnisse müssen neu sein.

```
//CREATE-SM-PUBSET-PARAMETERS SM-PUBSET-ID=S, SYSMIGRATE=*PAR, -  
//    SYSBACKUP=*PAR ( ARCHIVE-NAME=A, DIRECTORY-NAME=D )
```

Der SM-Pubset S wird konvertiert. Das Standardarchiv für Verdrängung sowie das zugehörige Archivverzeichnis werden mit einem Standardnamen erstellt. Für das Standardarchiv für Sicherung sowie für das zugehörige Archivverzeichnis wird ein eigener Name angegeben. Die Archivverzeichnisse müssen neu sein.

5.3.3.2 Wiederverwenden eines vorhandenen Archivverzeichnisses, dessen Umfang auf einen SM-Pubset beschränkt ist

Bereits vorhandene Archive sollen als Standardarchive für einen SM-Pubset unter HSMS-Kontrolle verwendet werden. Alle Pubsets, die diese Archive verwenden, sollen in diesem SM-Pubset eingeschlossen werden.

HSMS überprüft beim Erstellen von Archiven (öffentliche oder private) für die Sicherung oder Verdrängung in einer SM-Pubset-Umgebung, ob die angegebenen Archivverzeichnisse nur Datei- oder Jobvariablensätze für diesen SM-Pubset enthalten. Langzeitarchive werden nicht überprüft, da sie nicht lokal auf einem SM-Pubset sind.

Das folgende Bild zeigt eine typische Standardnutzung von Archiven durch Pubsets. Dabei wird ein einziges Archiv ausschließlich von den Pubsets M, N und O benutzt:

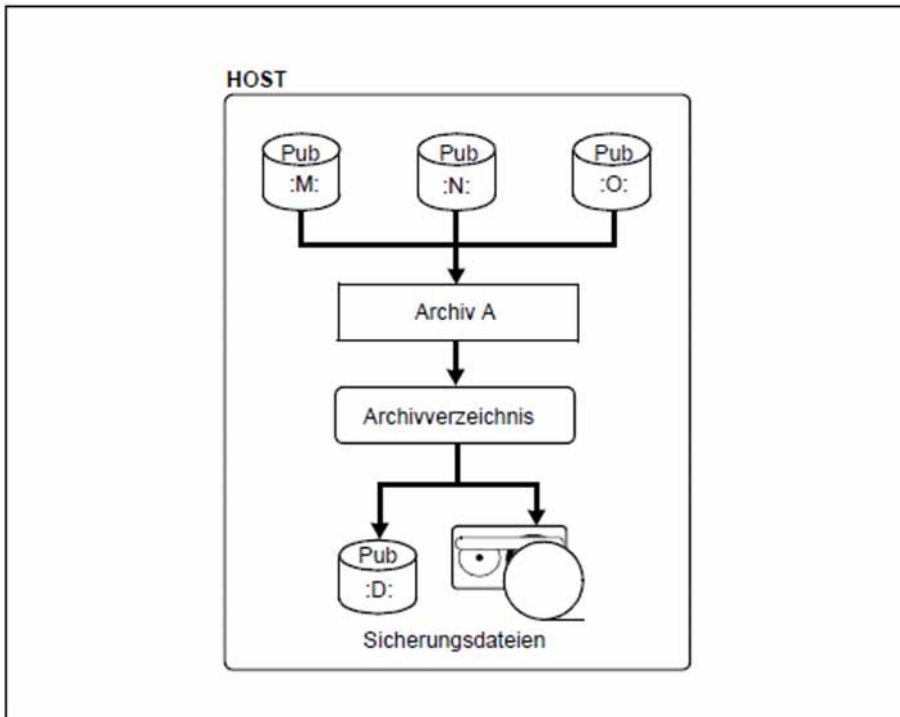


Bild 21: Archiv mit SF-Pubsets

Nach dem Konvertieren der Pubsets M, N und O in einen SM-Pubset (mit DVS-Anweisungen) können die Archivdefinitionen nicht mehr verwendet werden, da HSMS nicht mit einem SM-Pubset benutzt werden kann, der sich außer HSMS-Kontrolle befindet.

Die Archivdefinition befindet sich immer noch in der Host-Umgebung. Sie muss gelöscht werden, bevor der SM-Pubset für HSMS deklariert wird. Dies zeigt das folgende Bild.

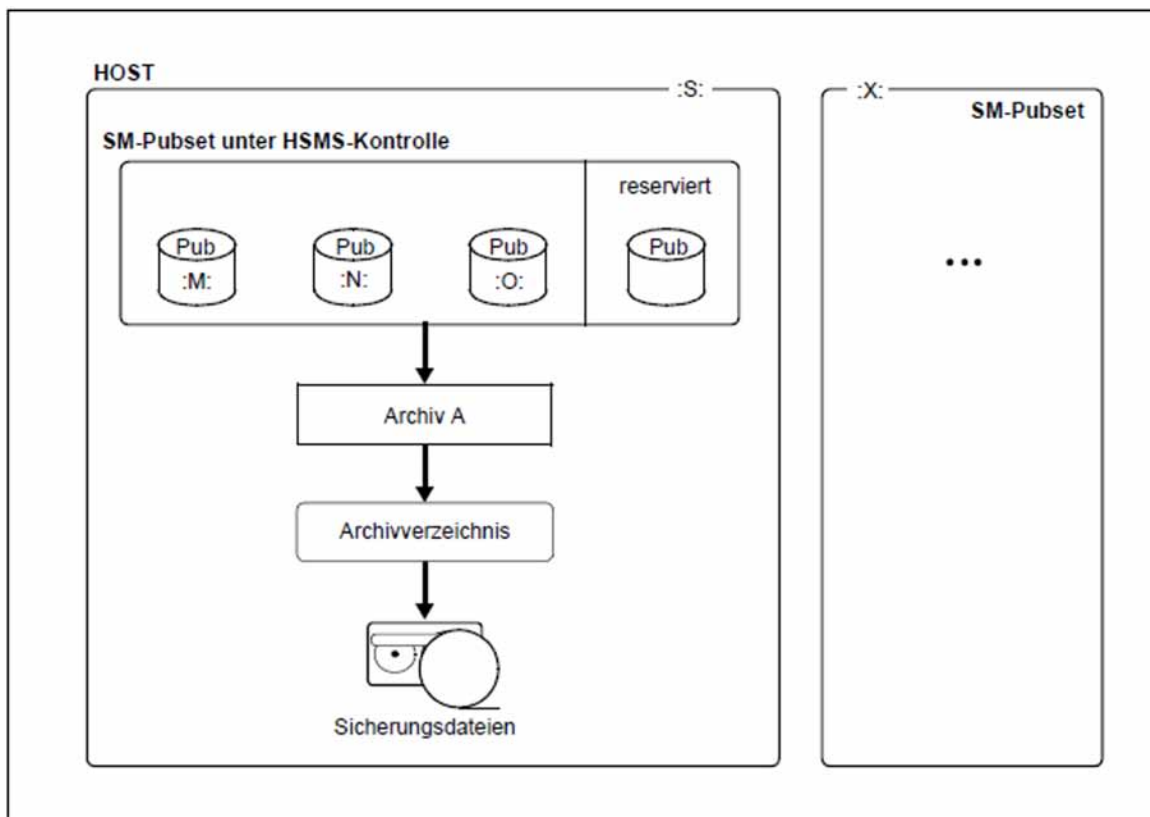


Bild 22: SM-Pubset außer HSMS-Kontrolle

Ein Pubset muss leer sein, damit er als S1-Volume-Set eines SM-Pubsets deklariert werden kann. Deshalb ist es nicht möglich, Sicherungsdateien zu benutzen, die auf Platte liegen (S1-Ebene). Damit ein bestehendes Standard-Migrationsarchiv genutzt werden kann, muss der HSMS-Verwalter die Sicherungsdateien vor dem Erstellen des SM-Pubsets von S1 nach S2 verdrängen. Dazu muss er folgende Anweisung eingeben:

```
//MIGRATE-FILES FROM-STORAGE=*S1-STORAGE-LEVEL(..., -
TO-STORAGE=*S2-STORAGE-LEVEL)
```

Für Backup-Archive muss die HSMS-Anweisung COPY-SAVE-FILE vor dem Konvertieren verwendet werden.

Der SM-Pubset wird mit folgender Anweisung unter HSMS-Kontrolle gebracht:

```
//CREATE-SM-PUBSET-PARAMETERS SM-PUBSET-ID=S, SYSMIGRATE=*UNDEFINED
```

Mit der HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS können bereits vorhandene Archivverzeichnisse angegeben werden, damit sie innerhalb des konvertierten SM-Pubsets benutzt werden können. Dabei müssen folgende Voraussetzungen berücksichtigt werden:

- Das Archivverzeichnis muss innerhalb des SM-Pubsets liegen. Es darf nur auf Dateien und Jobvariablen verweisen, die auf dem SM-Pubset liegen.
- Die Sicherungsdateien müssen im SM-Pubset auf der S2-Ebene liegen. Alle Datenträger (Magnetbandkassetten) müssen im Pool des Archivverzeichnisses sein.
Die Verdrängung nach S1 ist nur nach dem Konvertieren möglich.
- Das Archivverzeichnis darf nicht in einem anderen Archiv einer anderen Umgebung verwendet werden.

Das folgende Bild zeigt, wie ein vorhandenes Archiv nach dem Konvertieren des SM-Pubsets benutzt wird.

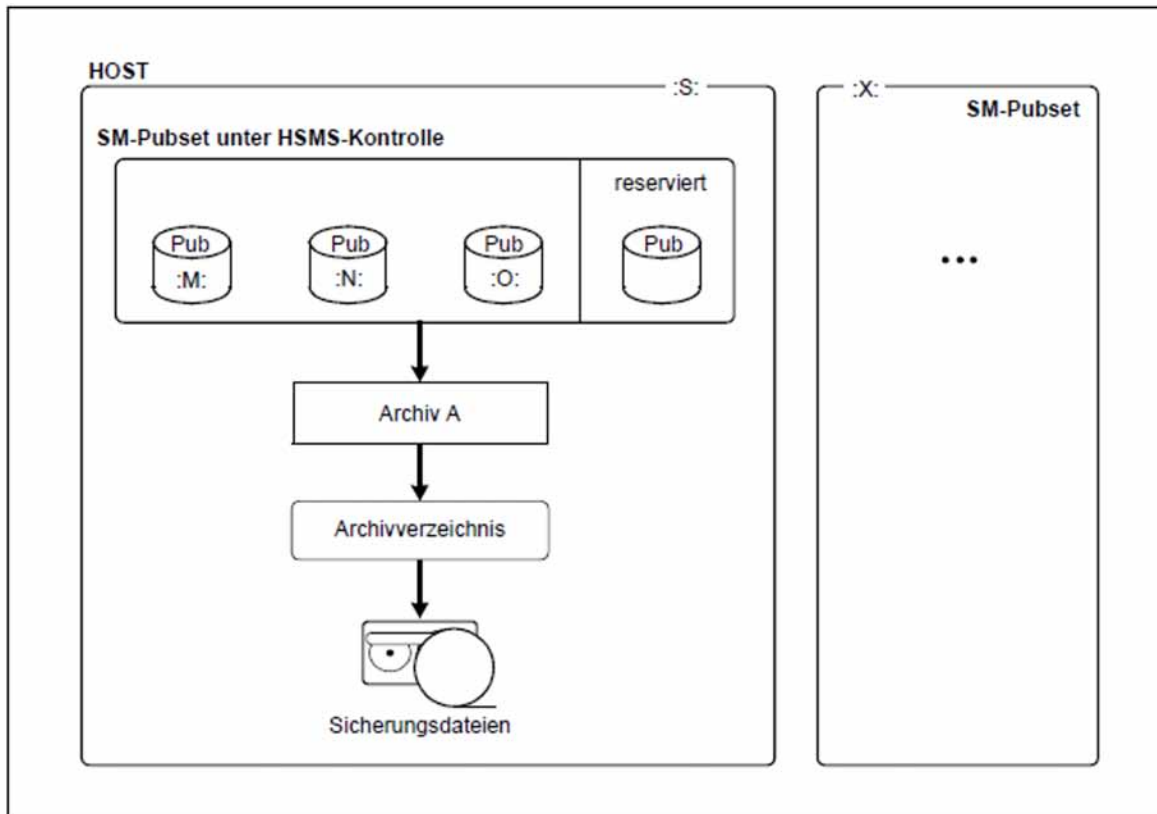


Bild 23: SM-Pubset unter HSMS-Kontrolle

Beispiel

```
//START-DIRCONV ...
//CREATE-ARCHIVE ARCHIVE-NAME=A,DIRECTORY-NAME=D(NEW-DIR=*NO), -
//
//          ENVIRONMENT=*SYSTEM-MANAGED(CAT-ID=*NO)
//MODIFY-SM-PUBSET-PARAMETERS SM-PUBSET-ID=S, SYSMIGRATE=A
```

Wenn das Archivverzeichnis D vorher schon von einem Archiv, das in der Steuerdatei des Hosts definiert ist, benutzt wurde, muss dieses Archiv zuerst gelöscht werden.

Für Backup- und Langzeitarchive kann dieselbe Operation durchgeführt werden.

Konvertieren/Einbringen von Archivverzeichnissen

Wenn ein Archivverzeichnis für einen SM-Pubset verwendet wird, überprüft HSMS, ob alle Datei- und Jobvariablensätze nur auf die Katalogkennung diesen SM-Pubsets verweisen. Wenn dies nicht der Fall ist, muss das Archivverzeichnis konvertiert werden, indem die Katalogkennung in allen Datei- und Jobvariablensätzen des Archivverzeichnisses umbenannt wird.

Im folgenden Bild werden zwei Pubsets in einen einzigen SM-Pubset eingebracht; dabei können diese Pubsets kein Systemarchiv oder ihr eigenes Systemarchiv (nicht mehrbenutzbar oder mehrbenutzbar mit Pubsets, die in denselben SM-Pubset eingebracht werden) für Sicherung und Verdrängung haben. Der HSMS-Verwalter ist dafür verantwortlich, dass die Archivdefinitionen und die alten Archivverzeichnisse gelöscht werden. Der MAREN-Katalog wird von HSMS automatisch aktualisiert.

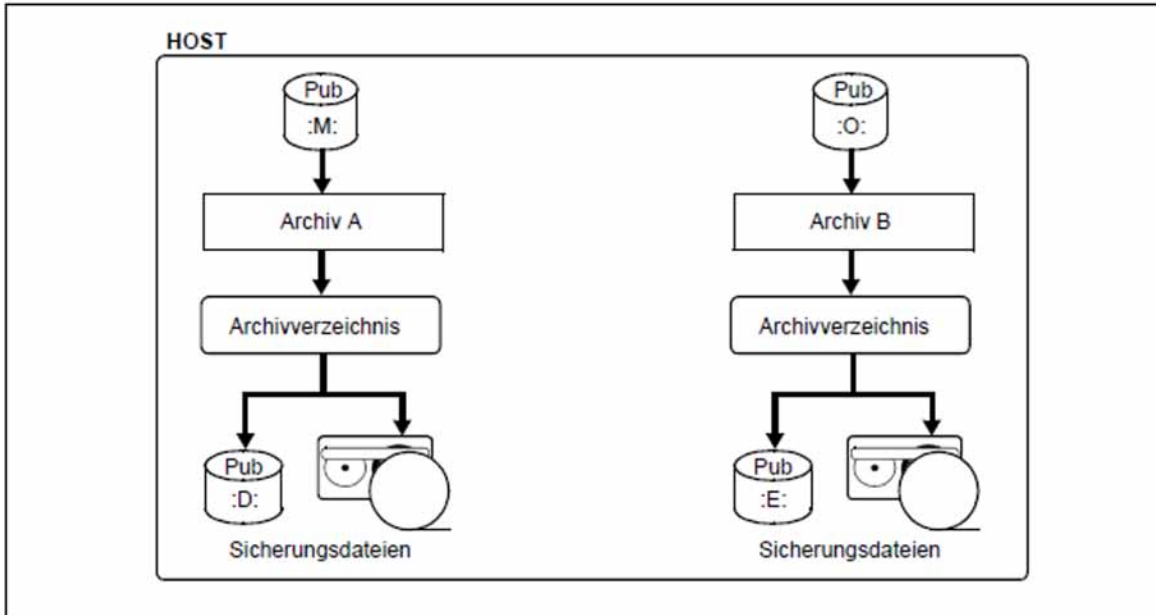


Bild 24: Verschiedene Archive mit getrennten SF-Pubsets

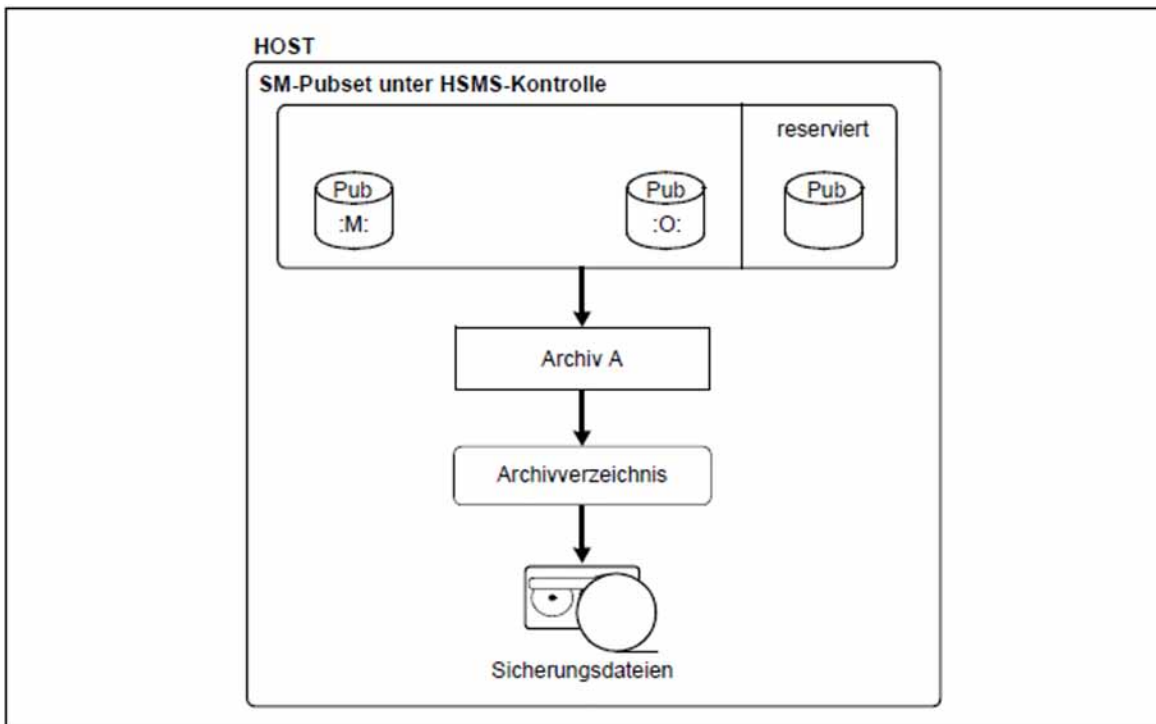


Bild 25: SM-Pubset unter HSMS-Kontrolle

Das Tool DIRCONV ermöglicht es, eine oder mehrere Archivverzeichnisse in ein neues Archivverzeichnis mit einer angegebenen Katalogkennung in alle Datei- und Jobvariablensätze zu konvertieren und einzubringen. Wenn ein Namenskonflikt auftritt, wird das Einbringen mit einer Fehlermeldung gestoppt. Das Problem muss dann individuell gelöst werden.

Falls der SM-Pubset aus mehreren SF-Pubsets erstellt wurde, die bereits ein gemeinsames Archivverzeichnis besitzen, müssen in DIRCONV die Anweisungen RENAME-CATID und UPDATE-VOLUME-CATALOG verwendet werden. Dies ist nur dann notwendig, wenn das neue Archivverzeichnis einen anderen Namen als das bereits vorhandene Archivverzeichnis hat.

Wenn der SM-Pubset aus verschiedenen SF-Pubsets erstellt wurde, die jeweils verschiedene Archivverzeichnisse besitzen, müssen in DIRCONV die Anweisungen MERGE-DIRECTORIES, RENAME-CATID und UPDATE-VOLUME-CATALOG verwendet werden.

5.3.3.3 Vorhandene Archivverzeichnisse, die nicht auf den SM-Pubset beschränkt sind

Wenn vorhandene Archivverzeichnisse für den zukünftigen SM-Pubset nicht lokal sind, sind mehrere Operationen nötig, um vor dem Konvertieren des SM-Pubsets die Umgebung dieses SM-Pubsets umzuwandeln.

Wenn ein Teil des zukünftigen SM-Pubsets ein gemeinsames Backup- oder Migrationsarchiv benutzt, kann dieses Archiv für diesen Teil verwendet werden; für den Rest des SM-Pubsets muss dann das folgende Verfahren angewandt werden.

1. Schritt: Konvertieren des Migrationsarchivs

Vor dem Erstellen des SM-Pubsets müssen alle betroffenen verdrängten Dateien gesichert werden:

```
//SELECT-FILE-NAMES FILE-NAMES=... , -
//  SELECT-FROM=*CATALOG(SUPPORT=*PUBLIC-DISK( -
//  STORAGE-TYPE=*PUBLIC-SPACE),STORAGE-LEVEL=( *S1,*S2)), -
//  OUTPUT=*SELECT-LIST
```

Anschließend muss ein neues Backup-Archiv mit einem neuen Archivverzeichnis erstellt werden:

```
//BACKUP-FILES FILE-NAMES=*SELECTED, -
//  SELECT-FILES=*ALL-FILES, -
//  NEW-FILE-NAMES=*BY-RULE(NEW-CATALOG-ID=...), -
//  SAVE-OPTIONS=*PARAMETERS(SAVE-DATA=*S2-S1-S0), -
//  ARCHIVE-NAME=...
```

Jetzt kann der SM-Pubset mit DVS-Mitteln erstellt werden. Danach muss das verwendete Archivverzeichnis in den SM-Pubset übertragen werden.

Der SM-Pubset wird dann konvertiert, wobei das oben genannte Archivverzeichnis für das System-Backup-Archiv benutzt wird. Das Migrationsarchiv wird mit einem neuen Archivverzeichnis festgelegt. Das neue Migrationsarchiv wird mit der HSMS-Anweisung REPAIR-CATALOG-BY-RESTORE aktualisiert.

2. Schritt: Kopieren der Sicherungsdateien in die SM-Pubset-Umgebung

Der Benutzer kann nach dem Festlegen der SM-Pubset-Umgebung (mit den neuen Archivdefinitionen) die Sicherungsdateien von der SF-Umgebung in die SM-Pubset-Umgebung kopieren; dazu steht die HSMS-Anweisung COPY-SAVE-FILE zur Verfügung.

Bei Sicherungs- und Langzeitarchiven ist es nicht immer nötig, alle Sicherungsdateien vom Host auf das SM-Pubset zu kopieren. Der Benutzer kann immer noch einen Restore vom „alten“ Archiv, das am Host definiert ist, zum SM-Pubset durchführen, wobei die Katalogkennung umbenannt wird.

5.3.3.4 Konvertieren eines SF-Pubsets zu einem Volume-Set eines SM-Pubsets

Bevor ein SF-Pubset als Volume-Set in einen SM-Pubset integriert werden kann, muss er dazu erst vorbereitet werden (siehe auch Handbuch „System Managed Storage“ [18]).

Wenn sich der betroffene SM-Pubset unter HSMS-Kontrolle befindet, muss der zu integrierende SF-Pubset vorher außer HSMS-Kontrolle gebracht werden. Dazu müssen in der HSMS-Anweisung MODIFY-PUBSET-PARAMETERS alle Operanden auf die Standardwerte gesetzt werden.

In HSMS ist es nicht vorgesehen, dass ein Pubset zwei Umgebungen angehört. Deshalb ist es empfehlenswert, die zentrale Steuerdatei zu sichern, bevor die SF-Umgebung in die SM-Umgebung migriert wird. Dadurch ist es später möglich, zur SF-Umgebung zurückzukehren und mit der gesicherten Steuerdatei die alten SF-Metadaten wiederherzustellen.

5.3.4 Ändern der SM-Pubset-Parameter

Nachdem ein SM-Pubset unter HSMS-Kontrolle genommen wurde, kann die Umgebung mit den verschiedenen, SM-Pubset-spezifischen Parametern mit der HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS geändert werden.

5.3.5 Anzeigen der SM-Pubset-Parameter

Die HSMS-Parameter, die für einen SM-Pubset spezifisch sind, können mit der HSMS-Anweisung SHOW-SM-PUBSET-PARAMETERS angezeigt werden.

5.3.6 SM-Pubset importieren oder exportieren

Für SM-Pubsets ist das Importieren und Exportieren von Pubsets wesentlich komplexer als für SF-Pubsets, da SM-Pubsets spezifische Metadaten enthalten.

Wenn das Kommando IMPORT-PUBSET für einen SM-Pubset gegeben wird, ruft die IMPORT-Task HSMS auf. HSMS überprüft die Namen der Steuer- und Auftragsdatei des SM-Pubsets, liest die Steuerdatei in Speichertabellen und reaktiviert die Aufträge der Auftragsdatei. Die General Catalog Facility wird geöffnet und die Sättigungsstufe des Pubsets wird überprüft. Eine Überprüfung wird durchgeführt, um aus den HSMS-Tabellen alle SF-Definitionen des Pubsets zu entfernen. SF-Definitionen können vorkommen, wenn der SM-Pubset unter derselben Katalogkennung konvertiert wurde oder wenn der SM-Pubset an einen anderen Host importiert wird. (Das Entfernen der SF-Definitionen ist nicht endgültig, da es nicht in der Steuerdatei, sondern im Common Memory Pool durchgeführt wird.)

Wenn die Startup-Verarbeitung abgeschlossen ist, steht der SM-Pubset für HSMS zur Verfügung. Die IMPORT-Task wartet maximal ca. 2 Minuten auf den Abschluss der Startup-Verarbeitung; dann gibt sie die Kontrolle zurück.

Wenn während der HSMS-Startup-Verarbeitung ein Fehler auftritt, wird die Verarbeitung abgebrochen, obwohl der Import des Pubsets fortgesetzt wird. Dies führt zu einem SM-Pubset, auf den das DVS zugreifen kann, der aber nicht „HSMS-gestartet“ ist. HSMS erkennt solche Situationen: ein neuer Startup wird automatisch ausgelöst, sobald eine Aktion in der Umgebung durchgeführt wird. Beispielsweise wird bei der Anweisung SHOW-SM-PUBSET-PARAMETERS versucht, die Umgebung erneut zu starten. Dadurch hat der HSMS-Verwalter die Möglichkeit, ein Problem in der Startup-Verarbeitung zu korrigieren, ohne den vollständigen Export/Import des SM-Pubsets durchzuführen. Erneut wartet die Task, die den Start der Umgebung anfordert, maximal ca. 2 Minuten auf den Abschluss der Startup-Verarbeitung. Die HSMS-Anweisung, die den Startup angefordert hat, misslingt, wenn ein Fehler oder eine Zeitlimitüberschreitung auftritt (obwohl in diesem Fall der Startup asynchron fortgesetzt wird).

Diese HSMS-Import-Verarbeitung wird ebenfalls automatisch beim Erstellen des Subsystems von der HSMS-Maintask aufgerufen, für jeden aktuell zugreifbare SM-Pubset, der unter HSMS-Kontrolle steht.

Beim Kommando EXPORT-PUBSET für einen SM-Pubset wird implizit die gesamte HSMS-Stop-Verarbeitung für eine SM-Umgebung durchführt. Die Stop-Verarbeitung bricht namentlich alle Aufträge mit dem Status ACCEPTED ab. Außerdem wartet sie auf den Abschluss aller bereits gestarteten Aufträge. Für jede HSMS-Servertask, die gerade in der Umgebung arbeitet, wird eine Meldung am Bedienplatz ausgegeben. Wenn der Synchronisierungspunkt erreicht ist, löscht die EXPORT-Task die Archivdefinitionen der SM-Umgebung aus dem Speicher und gibt die Kontrolle zurück.

Aufträge mit dem Status ACCEPTED, die von der Export-Verarbeitung abgebrochen wurden, werden nicht gelöscht. Sie bleiben in der Auftragsdatei und werden beim nächsten Import des SM-Pubsets reaktiviert. Dies gilt nicht für Concurrent-Copy-Aufträge, die keine SHC-OSD-Spiegelungsfunktionen verwenden und die abgebrochen werden und deren zugehörige Session vom EXPORT-Task deinitialisiert wird. Solche Concurrent-Copy-Aufträge werden beim nächsten Import des SM-Pubsets in den Status CANCELLED gebracht, da sie nicht wiederanlauffähig sind. Es findet keine Verarbeitung statt und es ist kein Report verfügbar. Wenn ein solcher Auftrag durch einen Export vorzeitig deinitialisiert wird, sendet der EXPORT-Task eine Meldung an den Bedienplatz.

5.3.7 Hinzufügen/Entfernen eines Volume-Sets zu/aus einem SM-Pubset

Wenn ein Volume-Set zu einem SM-Pubset hinzugefügt werden soll, ruft der entsprechende Benutzertask die HSMS-Schnittstelle DHSEMPV auf. DHSEMPV überprüft, ob der hinzuzufügende Volume-Set als SF-Pubset für HSMS deklariert ist:

- Wenn der Volume-Set als SF-Pubset für HSMS deklariert ist, kann er keinem SM-Pubset hinzugefügt werden (siehe auch [Abschnitt „Konvertieren eines SF-Pubsets zu einem Volume-Set eines SM-Pubsets“](#)).
- Wenn der Volume-Set nicht als SF-Pubset für HSMS deklariert ist, kann es einem SM-Pubset hinzugefügt werden. Asynchron dazu wird die Sättigungsstufe des neuen Volume-Sets überprüft.

Wenn ein Volume-Set aus einem SM-Pubset entfernt werden soll, ruft der entsprechende Benutzertask ebenfalls die HSMS-Schnittstelle DHSEMPV auf. DHSEMPV überprüft, ob der zu entfernende Volume-Set zur S1-Ebene des SM-Pubsets gehört. Folgende Fälle sind möglich:

- Der zu entfernende Volume-Set ist explizit als S1-Ebene eingestellt. In diesem Fall führt HSMS implizit ein MODIFY-SM-PUBSET-PARAMETERS mit S1-VOLUME-SET= *UNDEFINED aus. Damit ist für den SM-Pubset keine S1-Ebene mehr definiert.
- Der zu entfernende Volume-Set ist unter HSMS-Kontrolle und gehört wegen der Einstellung *ALL-HSMS-CONTROLLED zu der erweiterten S1-Ebene. In diesem Fall prüft HSMS, ob noch weitere Volume-Sets unter HSMS-Kontrolle für die S1-Ebene zur Verfügung stehen. Nur falls kein weiterer Volume-Set zur Verfügung steht, führt HSMS implizit ein MODIFY-SM-PUBSET-PARAMETERS mit S1-VOLUME-SET= *UNDEFINED aus. Damit ist für den SM-Pubset keine S1-Ebene mehr definiert.

5.4 Arbeiten mit Shared-Pubsets

Mit den HSMS-Anweisungen können Dateien und Jobvariablen bearbeitet werden, die sich auf einem Shared-Pubset befinden.

Wenn kein Backup-Server eingestellt ist und wenn der Shared-Pubset auf dem Slave-Host importiert ist, müssen folgende zwei Modi unterschieden werden:

- „Master-Modus“
Die HSMS-Anweisungen werden auf dem Master-Sharer bearbeitet. Dies ist der normale Modus.
- „Lokaler Modus“
Die HSMS-Anweisungen werden auf dem Sharer bearbeitet, an dem sie eingegeben wurden.

Wenn ein Backup-Server eingestellt ist, werden die HSMS-Anweisungen auf diesem Server bearbeitet. Dabei kann der Backup-Server Master- oder auch Slave-Host des Shared-Pubsets sein (siehe [Abschnitt „Backup-Server“](#)).

Eine Ausnahme bilden Sicherungsaufträge, die unter Verwendung der Funktion „Concurrent Copy“ erzeugt wurden. Solche Aufträge werden in der gewohnten Slave-Master-Umgebung abgewickelt. Eventuelle Backup-Server-Einstellungen werden ignoriert. Näheres hierzu siehe [Abschnitt „Sicherung mit der Funktion „Concurrent Copy““](#).

Wenn in Ihrem Rechenzentrum mehrere Rechner, auf denen verschiedene HSMS-Versionen im Einsatz sind, Pubsets gemeinsam nutzen, sollten Sie auch den [Abschnitt „Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen“](#) beachten.

Wird von mehreren Systemen und deren SF-Umgebungen gleichzeitig auf eine gemeinsame S1-Ebene bzw. Shared-SF-Pubsets gesichert, wird die notwendige Serialisierung der Läufe und damit der Zeitstempel mittels der Datei ARCHIVE.ASN erreicht. Diese Datei wird auf dem jeweiligen Shared-Pubset bzw. der Shared S1-Ebene verwaltet - siehe [Abschnitt "Arbeitsdateien"](#). In SM-Umgebung wird die Serialisierung der Läufe durch die SM-Pubset lokale Auftragsdatei und ARCHIVE Checkpoint-Datei gewährleistet.

5.4.1 Arbeiten mit Shared-SF-Pubsets

- Master-Modus
- Lokaler Modus

5.4.1.1 Master-Modus

- **Auflösung von Wildcards**

In HSMS-Anweisungen können Objekte, die sich auf mehreren Pubsets befinden, über Angaben wie *ALL, *OWN oder durch Wildcards in Verbindung mit der Pubsetkennung angegeben werden. Grundsätzlich werden solche Angaben über alle importierten Pubsets aufgelöst. Wenn unter diesen importierten Pubsets Shared-Pubsets sind, die im Slave-Modus importiert sind, werden diese Shared-Pubsets bei der Auflösung nicht in die Auswertung aufgenommen. Das System gibt dann eine entsprechende Meldung aus.

Beispiele

- FILE-NAMES = *ALL/*OWN/Wildcards über Katalogkennung
- JV-NAMES = *ALL/*OWN/Wildcards über Katalogkennung
- ADD-FILE-NAMES = *ALL/*OWN/Wildcards über Katalogkennung

bei den HSMS-Anweisungen

ARCHIVE-FILES, BACKUP-FILES, EXPORT-FILES, MIGRATE-FILES, RECALL-MIGRATED-FILES, SELECT-FILE-NAMES und UPDATE-EXPORT-SAVE-FILE.

- Angabe von PUBSET-ID = *ALL bei der HSMS-Anweisung SHOW-PUBSET-USAGE

- **Bearbeitung von HSMS-Anweisungen**

Wenn in HSMS-Anweisungen auf Shared-Pubsets Bezug genommen wird, die im Slave-Modus importiert sind, dann werden diese Anweisungen so abgearbeitet, wie es in der Übersichtstabelle auf "[Übersicht](#)" beschrieben ist.

- **Empfehlungen für die HSMS-Unterstützung für einen Shared-Pubset**

- Der HSMS-Verwalter sollte für jeden Shared-Pubset ein eigenes Systemarchiv für Sicherung, Versions-Backup, Langzeitarchivierung und Verdrängung anlegen.
- Auf allen Sharern eines Shared-Pubsets sollte das Sicherungs-Systemarchiv dieses Pubsets mit denselben Operanden angelegt werden, d.h. mit demselben Namen, demselben Archivverzeichnis, denselben Operanden für die Band- bzw. Plattenverarbeitung usw.
Das Gleiche gilt für die Systemarchive für Langzeitarchivierung, Versions-Backup und Verdrängung.
- Auf allen Sharern eines Shared-Pubsets sollten dem Sicherungs-Systemarchiv dieses Pubsets dieselben Operanden für TAPE-CONTROL (d.h. dieselben Bandverarbeitungszeiten usw.) zugewiesen werden.
Das Gleiche gilt für das Systemarchiv für Langzeitarchivierung, Versions-Backup und Verdrängung.
- Bei einem Shared-Pubset sollte das Archivverzeichnis des Systemarchivs für Sicherung, Langzeitarchivierung, Versions-Backup und Verdrängung auf diesem Pubset liegen.
- Der S1-Pubset, der einem gemeinsam benutzbaren S0-Pubset zugewiesen ist, muss für alle Sharer des S0-Pubsets derselbe sein und gemeinsam benutzbar sein.
- Der Band-Pool, der dem Systemarchiv für Sicherung, Langzeitarchivierung, Versions-Backup und Verdrängung eines Shared-Pubsets zugewiesen ist, muss mindestens für den Master dieses Pubsets zugreifbar sein (über MAREN).
- Der Master eines Shared-Pubsets muss für die HSMS-Bearbeitung auf genügend Bandgeräte zugreifen können.
- Bei Verwendung eines BACKUP-Servers (ab HSMS-V10.0 mit Parameter SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE), der nicht dem Master entspricht, muss der BACKUP-Server auf genügend Bandgeräte zugreifen können. Der Master benötigt nur wenige Bandgeräte (für //RESTORE-FILES, //RECALL-MIGRATED-FILES, ...)

- **Einschränkungen und Nachteile**

- Die Standard-Sicherungsdatei eines Archivs kann nur fortgesetzt werden, wenn zwischen zwei Läufen der Master nicht gewechselt wird. Die Standard-Sicherungsdatei ist nämlich Teil der Archivdefinition und die Archivdefinition ist für jeden Host lokal.
Diese Einschränkung gilt nicht für Shared-SM-Pubsets, bei denen die Archivdefinition lokal zum SM-Pubset ist.
- Private Plattendateien, die auf einem Shared-Pubset katalogisiert sind, können nur dann in das Sicherungs-Systemarchiv gesichert werden, wenn die Dateien einer gemeinsam benutzbaren, privaten Platte zugeordnet sind, die für den Master zugreifbar sein muss. Private Plattendateien, die diese Bedingung nicht erfüllen, müssen in ein besonderes Sicherungsarchiv gesichert werden, das von betroffenen Sharern bearbeitet werden kann.
- Die HSMS-Anweisungen ARCHIVE-FILES, BACKUP-FILES, BACKUP-FILE-VERSIONS und MIGRATE-FILES müssen nach einem Wechsel des Masters nicht erneut gestartet werden.
- Die Anweisungen BACKUP-FILES and BACKUP-FILE-VERSIONS für Dateien auf einem Shared-Pubset kann nicht an einem Slave erteilt werden, wenn eine Concurrent-Copy-Sitzung für einige dieser Dateien abläuft.
- Wenn zahlreiche Shared-Pubsets denselben Master/Backup-Server haben, kann dieser Hostrechner durch die HSMS-Bearbeitung überlastet sein.
- Wenn ein Master/Backup-Server während der Bearbeitung einer HSMS-Anweisung abstürzt oder während eines impliziten Recalls, der durch ein /SECURE-RESOURCE-ALLOCATION für verdrängte Dateien ausgelöst wurde, dann muss die HSMS-Anweisung bzw. das SECURE-RESOURCE-ALLOCATION-Kommando nochmals gegeben werden.
Nur ein impliziter Recall, der durch einen OPEN verursacht wurde, wird auf dem Backup-Master automatisch nach einem Wechsel des Masters wiederholt.
- Wenn auf dem Master/Backup-Server eine höhere HSMS-Version als auf anderen Sharern läuft, können einige Meldungen undefiniert sein, die in die Ausgabedatei geschrieben werden. In diesem Fall können Sie nur die Meldungsnummer und die Inserts in der Ausgabedatei lesen. Die Bedeutungs- und Hilfetexte können Sie mit dem Kommando /HELP-MSG entweder auf dem Master ansehen oder auf jedem Sharer, auf dem eine HSMS-Version läuft, in der diese Meldung definiert ist.

- **Vorteile des Master-Modus**

Bessere Performance als im Lokalen Modus, da nur geringe Kommunikation zwischen dem Slave-Sharer und dem Master erforderlich ist.

5.4.1.2 Lokaler Modus

- **Auflösung von Wildcards**

Shared-Pubsets werden wie exklusiv importierte Pubsets behandelt, unabhängig vom Master-/Slave-Modus.

- **Bearbeitung von HSMS-Anweisungen**

Alle HSMS-Anweisungen, die Shared-Pubsets betreffen, werden auf dem Hostrechner bearbeitet, auf dem HSMS-Anweisungen erteilt wurden.

- **Empfehlungen für die HSMS-Unterstützung für einen Shared-Pubset**

- Der HSMS-Verwalter sollte für jeden Shared-Pubset ein eigenes Systemarchiv für Sicherung, Langzeitarchivierung und Verdrängung anlegen.
- Auf allen Sharern eines Shared-Pubsets sollte das Sicherungs-Systemarchiv dieses Pubsets mit demselben Namen und demselben Archivverzeichnis erstellt werden. Die anderen Archiv-Operanden können von einander abweichen.
Das Gleiche gilt für die Systemarchive für Langzeitarchivierung, Versions-Backup und Verdrängung.
- Auf allen Sharern eines Shared-Pubsets können dem Sicherungs-Systemarchiv dieses Pubsets entweder dieselben oder andere Operanden für TAPE-CONTROL zugewiesen werden, wobei eine Ausnahme gilt: Die Intervalle der Bandverarbeitungszeiten der Pubset-Sharer dürfen sich nicht überschneiden.
Das Gleiche gilt für das Systemarchiv für Langzeitarchivierung, Versions-Backup und Verdrängung sowie für private Sicherungs- und Langzeitarchive mit Archivverzeichnissen, die von mehreren Pubset-Sharern gemeinsam benutzt werden.
- Bei einem Shared-Pubset sollte das Archivverzeichnis des Systemarchivs für Sicherung, Langzeitarchivierung, Versions-Backup und Verdrängung auf diesem Pubset liegen.
- Der S1-Pubset, der einem gemeinsam benutzbaren S0-Pubset zugewiesen ist, muss für alle Sharer des S0-Pubsets derselbe sein und gemeinsam benutzbar sein.
- Der Band-Pool, der dem Systemarchiv für Sicherung, Langzeitarchivierung, Versions-Backup und Verdrängung eines Shared-Pubsets zugewiesen ist, muss für alle Sharer des Pubsets zugreifbar sein (über MAREN).
- Alle Sharer eines Shared-Pubsets müssen für die HSMS-Bearbeitung auf genügend Bandgeräte zugreifen können.

- **Einschränkungen und Nachteile**

- Es ist darauf zu achten, dass bei überlappenden Bandverarbeitungszeiten verschiedene Sharer nicht mit demselben Archivverzeichnis arbeiten.
- Zwischen dem Slave und dem Master ist eine umfangreiche Kommunikation erforderlich, um die Metadaten des Pubsets zu bearbeiten.

- **Vorteile des lokalen Modus**

- Die Einschränkungen und Nachteile des Master-Modus gelten nicht für den lokalen Modus.
- Wenn ein Master während der Bearbeitung einer HSMS-Anweisung abstürzt, wird die Bearbeitung nach dem Wechsel des Masters automatisch fortgesetzt.

5.4.2 Arbeiten mit Shared-SM-Pubsets

Bei Shared-SM-Pubsets gewährleistet HSMS selbst – im Gegensatz zu Shared-SF-Pubsets – die Kohärenz der Migrations-, Backup- und Versions-Backup-Archivdefinitionen auf allen Sharern, d.h. der Verwalter muss sich darum nicht mehr kümmern.

Bei Shared-SM-Pubsets ist die Auftragsverwaltung einfacher als bei Shared-SF-Pubsets, da nur eine einzige Auftragsdatei – nämlich die lokale Auftragsdatei des SM-Pubsets – beteiligt ist. Das Anzeigen, Wiederanlaufen lassen (Restart) und Löschen von Aufträgen ist von jedem Sharer aus möglich. Beim Wiederanlaufenlassen von Aufträgen wird der neue Master-Sharer bei jedem Restart dynamisch getrennt. Beim Löschen von Aufträgen muss der Master-Sharer nicht aufgerufen werden.

Außerdem ist in einer SM-Umgebung nur eine einzige Katalogkennung beteiligt; deshalb müssen keine Wildcards aufgelöst werden.

5.4.3 Backup-Server

Für die Bearbeitung von HSMS-Anweisungen kann explizit ein Backup-Server vereinbart werden (ab HSMS V10.0 mit dem Parameter SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE). Die Bearbeitung erfolgt auf dem Backup-Server, unabhängig davon, ob der Server Master- oder Slave-Host des Shared-Pubsets ist.

Eine Ausnahme bilden Sicherungs- und Versions-Backup-Aufträge, die unter Verwendung der Funktion "Concurrent Copy" erzeugt wurden. Solche Aufträge werden in der gewohnten Slave-Master-Umgebung abgewickelt. Eventuelle Backup-Server-Einstellungen werden ignoriert. Näheres hierzu siehe [Abschnitt „Sicherung mit der Funktion „Concurrent Copy“](#).

Das Archiv-Attribut BACKUP-SERVER-USAGE bestimmt, wo HSMS-Anweisungen, die dieses Archiv betreffen, bearbeitet werden:

- Mit *NO werden die Anweisungen, wie bisher in HSMS < V10.0, in der Regel am Pubset-Master bearbeitet (Details siehe [Abschnitt „Master-Modus“](#) und [Abschnitt „Übersicht“](#)).
- Mit *STD werden die Anweisungen entsprechend der aktuellen Einstellung im HSMS-Parameter BACKUP-SERVER bearbeitet:
 - Mit *NONE werden die Anweisungen, wie bisher in HSMS < V10.0, am Pubset-Master bearbeitet. Wenn ein Backup-Server verwendet wird, muss an diesem Server *LOCALHOST eingetragen sein.
 - Wenn ein Backup-Server eingetragen ist, werden die Anweisungen an diesem Server bearbeitet.
 - Wenn *LOCALHOST oder der Name des lokalen Systems eingetragen ist, werden die Anweisungen im lokalen System bearbeitet. Diese Einstellung ist obligatorisch an einem System, das als Backup-Server eingesetzt werden soll.

Wirkungsweise der verschiedenen Einstellungen

Archiv-Definition BACKUP-SERVER- USAGE=	Wert des Operanden BACKUP-SERVER am lokalen System:		
	*NONE	<alphanum-name 1..8> <bcam-name>	*LOCALHOST
*STD	Bearbeitung ohne Nutzung der Backup-Server-Funktionalität ¹	Bearbeitung am Backup-Server ^{2 3}	Bearbeitung im lokalen System Aktuelles System ist der Backup-Server
*NO	Bearbeitung ohne Nutzung der Backup-Server-Funktionalität ¹		

¹ Wenn das lokale System Master-Share eines Shared-Pubsets ist, wird der Auftrag lokal ausgeführt. Wenn das lokale System Slave-Share eines Shared-Pubsets ist, wird der Auftrag an den Master gesendet.

² Am Backup-Server wird geprüft, ob BACKUP-SERVER=*LOCALHOST eingestellt ist. Ist dies nicht der Fall, wird der Auftrag abgewiesen und die Meldung HSM0329 in die Protokolldatei ausgegeben.

³ Die Angabe des Namens des lokalen Systems hat dieselbe Wirkung wie *LOCALHOST.

Unabhängig von der Einstellung BACKUP-SERVER werden die folgenden Aufträge immer lokal ausgeführt:

- Aufträge unter CCOPY
- Anweisungen für Datentransfer
IMPORT-FILES
EXPORT-FILES
COPY-EXPORT-SAVE-FILE
UPDATE-EXPORT-SAVE-FILE
- Anweisungen für Nodes
- COPY-SAVE-FILE
- MOVE-SAVE-FILE

Fehlerbehandlung

Wenn eine Anweisung zur Bearbeitung an einen Backup-Server geschickt wird, wird der HSMS-Parameter BACKUP-SERVER geprüft, um festzustellen, ob dieses System aktuell ein Backup-Server ist (also ob *LOCALHOST oder der Hostname des Systems eingetragen ist). Wenn das System aktuell kein Backup-Server ist, wird der Auftrag abgewiesen und erhält den Status INTERRUPTED mit dem Substatus BACK-SERV-REPLIED.

Im HSMS Report zeigt die Meldung HSM0329 zusätzlich die Fehlerursache an.

5.4.4 Übersicht

Die folgende Übersicht zeigt, wo Anweisungen bearbeitet werden, wenn Shared-Pubsets betroffen sind, abhängig davon, ob ein Backup-Server eingestellt ist oder nicht:

Anweisung	Bearbeitung	
	kein Backup-Server eingestellt	Backup-Server eingestellt
ARCHIVE-FILES	(1)	(2)
ARCHIVE-NODE-FILES	lokal (Knoten)	lokal (Knoten)
BACKUP-FILES	(1)	(2)
BACKUP-FILE-VERSIONS	(1)	(2)
BACKUP-NODE-FILES	lokal (Knoten)	lokal (Knoten)
CHECK-CATALOGED-FILES	lokal + SM-dynamisch	lokal + SM-dynamisch
COPY-EXPORT-SAVE-FILE	lokal (Datentransfer)	lokal (Datentransfer)
COPY-NODE-SAVE-FILE	lokal (Knoten)	lokal (Knoten)
COPY-SAVE-FILE	lokal (Anzahl Master?)	lokal
CREATE-ARCHIVE	lokal + SM-dynamisch	lokal
CREATE-MANAGEMENT-CLASS	lokal + SM-dynamisch	lokal + SM-dynamisch
CREATE-SM-PUBSET-PARAMETERS	lokal + SM-dynamisch	lokal + SM-dynamisch
DELETE-ARCHIVE	lokal + SM-dynamisch	lokal + SM-dynamisch
DELETE-MANAGEMENT-CLASS	lokal + SM-dynamisch	lokal + SM-dynamisch
DELETE-REQUESTS	(3)	(3)
EXPORT-FILES	lokal (Datentransfer)	lokal (Datentransfer)
IMPORT-FILES	lokal (Datentransfer)	lokal (Datentransfer)
LIST-VOLUMES	lokal	lokal
MIGRATE-FILES	(1)	(2)
MODIFY-ARCHIVE	lokal + SM-dynamisch	lokal + SM-dynamisch
MODIFY-ARCHIVE-ATTRIBUTES	lokal + SM-dynamisch	lokal + SM-dynamisch
MODIFY-HSMS-PARAMETERS	lokal	lokal

MODIFY-PUBSET-PARAMETERS	lokal	lokal
MODIFY-SM-PUBSET-PARAMETERS	lokal + SM-dynamisch	lokal + SM-dynamisch
MODIFY-TAPE-CONTROL	lokal + SM-dynamisch	lokal + SM-dynamisch
MOVE-SAVE-FILES	lokal	lokal
RECALL-MIGRATED-FILES	(1)	(2)
RECALL implizit (durch /SECURE-RESOURCE-ALLOCATION oder OPEN)	(1)	(2)
RECOVER-REQUESTS	lokal	lokal
REORGANIZE-VERSION-BACKUP	lokal (Anzahl Master?)	(2)
REPAIR-CATALOG-BY-EXCHANGE	lokal	lokal
REPAIR-CATALOG-BY-RESTORE	(1)	(2)
REPLACE-SAVE-FILE-BY-EXCHANGE	lokal	lokal
REPLACE-SAVE-FILE-BY-RESTORE	lokal (Anzahl Master?)	lokal
RESTART-REQUESTS	(3)	(3)
RESTORE-FILES	(1) und (4)	(2) und (4)
RESTORE-LIBRARY-ELEMENTS	(1) und (4)	(2) und (4)
RESTORE-NODE-FILES	lokal (Knoten)	lokal (Knoten)
SELECT-FILE-NAMES	lokal	lokal
SELECT-JV-NAMES	lokal	lokal
SELECT-NODE-FILES	lokal (Knoten)	lokal (Knoten)
SHOW-ARCHIVE	lokal	lokal
SHOW-ARCHIVE-ATTRIBUTES	lokal	lokal
SHOW-HSMS-PARAMETERS	lokal	lokal
SHOW-MANAGEMENT-CLASS	lokal	lokal
SHOW-NODE-PARAMETERS	lokal (Knoten)	lokal (Knoten)
SHOW-PUBSET-PARAMETERS	lokal	lokal
SHOW-PUBSET-USAGE	lokal	lokal
SHOW-REQUESTS	lokal	lokal

SHOW-SM-PUBSET-PARAMETERS	lokal	lokal
SHOW-TAPE-CONTROL	lokal	lokal
UPDATE-EXPORT-SAVE-FILE	lokal (Datentransfer)	lokal (Datentransfer)

Legende

(1)

Die nachstehenden Regeln werden eingehalten:

- Shared-Pubsets, die im Master-Modus importiert sind, werden genauso wie lokale Pubsets behandelt.
- Wenn lokale Pubsets zusammen mit Shared-Pubsets angegeben sind, dann werden die lokalen Pubsets lokal bearbeitet und die im Slave-Modus importierten Shared-Pubsets werden mit einer Warnung zurückgewiesen.
- Anweisungen, die im Slave-Modus importierte Shared-Pubsets behandeln, werden am Master bearbeitet, wenn der Master eindeutig ist; ansonsten wird die Anweisung zurückgewiesen.

(2)

Die nachstehenden Regeln werden eingehalten:

- Aufträge werden auf dem Backup-Server bearbeitet, wenn die Archiv-Definition BACKUP-SERVER-USAGE den Wert *STD hat und gleichzeitig der globale HSMS-Parameter BACKUP-SERVER einen Wert ungleich *NONE hat.
 - a. Wenn für BACKUP-SERVER ein Host-Name angegeben ist, wird am Backup-Server geprüft, ob für BACKUP-SERVER der Wert *LOCALHOST oder der Name des lokalen Systems angegeben ist. Ist dies nicht der Fall, wird der Auftrag abgewiesen und die Meldung HSM0329 in die Protokolldatei ausgegeben.
 - b. Wenn BACKUP-SERVER den Wert *LOCALHOST hat, wird der Auftrag lokal ausgeführt, da das lokale System der Backup-Server ist.
 - c. Wenn für BACKUP-SERVER der Host-Name des lokalen Systems angegeben ist, werden die Aufträge so behandelt, als wenn *LOCALHOST angegeben wäre.
- Wenn lokale Pubsets zusammen mit Shared-Pubsets angegeben sind, werden die lokalen Pubsets ignoriert, da die Backup-Server-Funktionalität nur für Shared-Pubsets unterstützt wird.

(3)

Auftragsverwaltung.

Siehe Abschnitt „[Auftragsverwaltung](#)“ zum Löschen/Wiederanlaufen lassen von Aufträgen.

(4)

Restore-Aufträge werden lokal bearbeitet, wenn die Katalogkennung umbenannt wird.

lokal (Anzahl Master?)

Aufträge, für die die Wahrscheinlichkeit groß ist, dass sie mehr als einen Master haben, werden immer lokal bearbeitet. Dies ist besonders der Fall, wenn eine Anweisung die gesamte Sicherungsdatei bearbeitet.

lokal (Datentransfer)

Anweisungen für Datenübertragung werden immer lokal bearbeitet.

lokal (Knoten)

Anweisungen für Knoten werden immer lokal bearbeitet.

lokal + SM-dynamisch

Anweisungen werden lokal bearbeitet. Allerdings sind in einer SM-Umgebung die Ergebnisse der Anweisungsbearbeitung auf anderen Sharern des betroffenen SM-Pubsets sichtbar.

5.5 Arbeiten mit SM-Pubsets und verschiedenen HSMS-Versionen

SM-Pubsets können von System zu System exportiert und importiert werden. Auf den Systemen können verschiedene HSMS-Versionen im Einsatz sein. Um Übereinstimmung zwischen den jeweiligen HSMS-Steuer- und Auftragsdateien zu erreichen, wurden für die HSMS-Definitionen auf den SM-Pubsets neue Kompatibilitätsmerkmale festgelegt: Jedes Archiv oder jeder Auftrag innerhalb eines SM-Pubsets besitzt ein Attribut, das so genannte Versionsattribut. Dieses Attribut beschreibt die HSMS-Version, die zur Bearbeitung des Archivs oder des Auftrags benötigt wird. Dabei gelten folgende allgemeine Regeln:

- Wenn eine Definition zum ersten Mal erstellt wird, ermittelt HSMS die älteste HSMS-Version, die die Definition bearbeiten kann.
- Bevor eine Definition benutzt oder geändert wird, überprüft HSMS, ob die geforderte HSMS-Version für die Definition ausreichend ist.
- Wenn eine Definition geändert wird, ermittelt HSMS erneut die älteste HSMS-Version, die die Definition bearbeiten kann.

Wenn eine Definition mit einer SHOW-Anweisung angezeigt wird, versucht HSMS immer, möglichst viele richtige Operanden anzuzeigen. Wenn ein Operand außerhalb des zulässigen Bereichs liegt, wird sein Wert durch das Zeichen „?“ dargestellt.

Wenn in einer HSMS-Anweisung ein Operand verboten wird, der ein Archiv betrifft, dann wird das Versionsattribut des betroffenen Auftrags nicht herabgesetzt (Beispiel: Einem Archiv ist ein Schattenarchiv zugeordnet und der Auftrag wurde mit SHADOW-COPY= *INHIBITED erteilt).

Bei Shared-SM-Pubsets senden die Slave-Sharer ihre Aufträge meistens an den Master-Sharer zur Bearbeitung. Wenn ein Slave-Sharer einen Auftrag erstellt, ermittelt er die funktionelle HSMS-Version des Auftrags, wobei die oben beschriebenen Bedingungen herangezogen werden.

Wenn der Master-Sharer den Auftrag erhält, überprüft er zuerst anhand der funktionellen HSMS-Version, ob er den Auftrag überhaupt bearbeiten kann. Ist die funktionelle HSMS-Version des Auftrags größer als die HSMS-Version auf dem Master-Sharer, dann weist der Master-Sharer den Auftrag zurück.

Dieser funktionelle Versions-Mechanismus ermöglicht Sharern, auf denen höhere HSMS-Versionen laufen, Aufträge, die keine neuen HSMS-Funktionen enthalten, an Sharer mit niedrigeren HSMS-Versionen zu übergeben.

5.6 Dateiattribute in einer Umgebung mit SM-Pubsets verwalten

Die Dateiattribute lassen sich in zwei Gruppen einteilen:

- Schutzattribute
 - Kennwörter
 - Standardzugriffskontrolle mit USER-ACCESS und ACCESS
 - GUARDS, BASIC-ACL

Diese Schutzattribute sind nicht für einen SF- oder SM-Pubset spezifisch. Sie sind im Handbuch „ARCHIVE“ [2] beschrieben.

- Zuordnungsattribute
 - STORAGE-CLASS
 - AVAILABILITY
 - WORKFILE
 - IOPERF
 - IOUSAGE
 - BLOCK-CONTROL-INFO

Diese Attribute werden in einem SM-Pubset speziell verwendet. Die folgende Tabelle zeigt, wie die Attribute bei RESTORE bzw. IMPORT bearbeitet werden. Wenn eine Datei, die von einem SM-Pubset gesichert wurde, auf einen SF-Pubset restauriert wird, dann werden diese Attribute zurückgesetzt.

Attribut	Zurückholen	Restaurieren ohne Umgebungswechsel	Restaurieren mit Umgebungswechsel
STORAGE-CLASS	von Platte	von Band	Zurücksetzen auf *NONE
AVAILABILITY	von Platte	von Band	von Band
WORKFILE	von Platte	von Band	von Band
IOPERF	von Platte	von Band	von Band
IOUSAGE	von Platte	von Band	von Band

Im Gegensatz zu SF-Pubsets, bei denen der komplette Pubset entweder Key-Format oder Non-Key-Format hat, kann ein SM-Pubset Volume-Sets mit Key-Platten und Volume-Sets mit Non-Key-Platten enthalten. Deshalb wird in einer Umgebung mit SM-Pubsets nicht automatisch vom Key-Format in das Non-Key-Format konvertiert. Eine Datei wird mit denselben BLKCTRL- und PAMKEY-Werten wie bei der Sicherung restauriert/zurückgeholt.

Eine Datei wird in Bezug auf ihre Zuordnungsattribute und Benutzerstandards auf den bestmöglichen Volume-Set restauriert/zurückgeholt, wobei es aber folgende Ausnahmen gibt:

- Eine Datei mit dem Attribut S0-MIGRATION-FORBIDDEN wird immer auf dem Volume-Set restauriert /zurückgeholt, auf dem sie ursprünglich lag.
- Bei der RESTORE-/RECALL-Anweisung wurde ein Volume-Set angegeben (dies gilt nicht für eine Datei mit dem Attribut S0-MIGRATION-FORBIDDEN).

Wenn eine Dateizuordnung nicht möglich ist, weil der angegebene Volume-Set nicht existiert, dann wird die Datei auf den bestmöglichen Volume-Set restauriert/zurückgeholt (dies gilt auch für eine Datei mit dem Attribut SO-MIGRATION-FORBIDDEN).

5.7 Verwaltung der Workstations

Dieser Abschnitt ist nur für den HSMS-Verwalter. Er beschreibt, wie die Verbindung zwischen HSMS im BS2000 und einer Workstation hergestellt und verwaltet wird, wenn diese über das BS2000-UFS (POSIX) gemountet wird und deren Dateien gesichert oder archiviert werden sollen. Dazu werden dann die Anweisungen BACKUP-NODE-FILES bzw. ARCHIVE-NODE-FILES verwendet.

Bei UNIX-Workstations und PCs kann die Verbindung nur durch NFS hergestellt werden.

Da es für NFS ein eigenes Handbuch gibt [\[11\]](#), sind im Folgenden nicht alle Details der durchzuführenden Tätigkeiten beschrieben.

5.7.1 Notwendige Tätigkeiten an einer Workstation bei Zugriff über NFS

Wenn der Zugriff auf eine UNIX-Workstation über NFS erfolgt, muss der HSMS-Verwalter vorher einige Tätigkeiten an der Workstation durchführen. Im Folgenden ist beschrieben, was bei einer UNIX-Workstation zu tun ist.

Der Verwalter einer Workstation muss all seine Dateisysteme, die HSMS bearbeiten soll, folgendermaßen konfigurieren, damit sie gemeinsam benutzbar (shareable) sind:

- Dateisysteme in die Datei `/etc/dfs/sharetab` eintragen
- Dateisysteme mit dem UNIX-Kommando `share` gemeinsam benutzbar machen

Durch den Einsatz von NFS können aber nur solche Dateisysteme als gemeinsam benutzbar (share) erklärt werden, bei denen nicht bereits ein Teil gemeinsam benutzbar ist.

Für die betroffenen Dateisysteme auf der UNIX-Workstation muss die Root-Berechtigung für den BS2000-Rechner erteilt werden, auf dem HSMS läuft.

Der Administrator muss die Liste der gemeinsam benutzbaren Dateisysteme, die er sichern will, im entsprechenden Sicherungskommando für Knotendateien angeben. Eine Liste der gemeinsam benutzbaren Dateisysteme kann mit dem UNIX-Kommando `dfshares` ermittelt werden.

5.7.2 Notwendige Tätigkeiten im BS2000 (Server)

Damit Workstations bearbeitet werden können, muss der HSMS-Verwalter im BS2000 folgende Tätigkeiten durchführen:

Archive einrichten

Damit Dateien des lokalen BS2000-UFS bearbeitet werden können, darf jeder Benutzer Archive einrichten, um seine eigenen Dateien bearbeiten zu können. Der HSMS-Verwalter kann Standard-Systemarchive für die Datensicherung und die Langzeitarchivierung einrichten. Die öffentlichen Archive für die Datensicherung können mit dem symbolischen Namen *SYSNODEBACKUP angesprochen werden, die für die Langzeitarchivierung mit *SYSNODEARCHIVE.

Pfad definieren und Dateisysteme einhängen

Wenn HSMS als Backup-Server für Knotendateien verwendet wird, muss der Systemverwalter diese Knotendateien verfügbar machen:

- POSIX muss für die Verarbeitung durch NFS vorhanden sein. Das lokale BS2000-UFS ist das POSIX-Dateisystem. Die Dateien des POSIX-Dateisystems sind automatisch verfügbar.
- Um Daten unter Verwendung von NFS von einer Workstation sichern bzw. auf eine Workstation restaurieren zu können, muss jedes ferne Dateisystem, das HSMS bearbeiten soll, mit NFS am entsprechenden Knotenpunkt „HSMS/<node-id>“ des lokalen BS2000-UFS eingehängt werden. Wenn es sich um ein untergeordnetes Dateisystem innerhalb einer Workstation handelt, muss es an einer niedrigeren Dateiverzeichnisebene eingehängt werden.
- Das Dateiverzeichnis „HSMS“ sollte die Lese-, Schreib- und Ausführungsberechtigung nur für root besitzen. Alle anderen Schutzattribute sollten auf die höchste Sicherheitsstufe gesetzt werden. Sonst können nämlich die Dateien auf einer Workstation auch von anderen UFS-Benutzern gelesen werden.

- Wenn bei Verwendung von NFS ein Dateisystem noch weitere untergeordnete Dateisysteme enthält, muss der BS2000-Systemverwalter jedes von ihnen explizit am richtigen Knoten in derselben Reihenfolge einhängen, in der sie an der Workstation eingehängt waren. Das folgende Bild soll dies verdeutlichen:

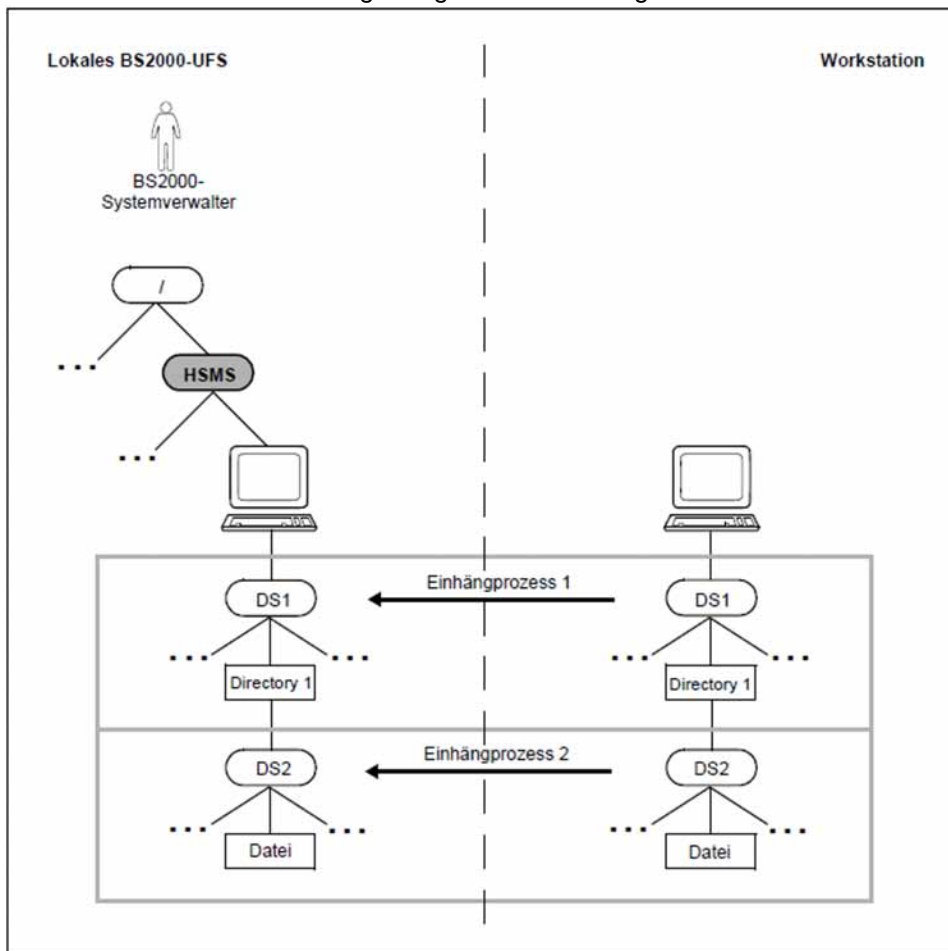


Bild 26: Richtiges Einhängen von Dateisystemen

Wir empfehlen Ihnen, eine Standardprozedur zu erstellen, damit Sie möglichst einfach Dateisysteme im lokalen BS2000-UFS in derselben Reihenfolge wie an der Workstation einhängen können.

Beispiel

NFS-Kommando `share <WS-address>`

Mögliche Angaben:

RESOURCE	SERVER	ACCESS	TRANSPORT
<WS-address>:/	<WS-address>	---	---
<WS-address>:/opt	<WS-address>	---	---
<WS-address>:/usr	<WS-address>	---	---
<WS-address>:/var	<WS-address>	---	---
<WS-address>:/home	<WS-address>	---	---

Das Einhängen muss von der Spitze des Dateibaums aus nach unten begonnen werden. Im Beispiel oben muss zuerst das Root-Verzeichnis „/“ unter einem eindeutigen Einhängepunkt eingehängt werden.

Es müssen nicht alle Dateisysteme eines Dateibaums aufgeführt werden; aber alle Einträge müssen verschiedene Dateisysteme betreffen. Das Kommando *share* erlaubt für ein Dateisystem keine Mehrfach-Deklarationen. Das BS2000 kann das Vorhandensein aller Dateisysteme nicht kontrollieren.

Für das obige Beispiel verläuft das Einhängen folgendermaßen:

<code>mount -F nfs <WS-address>/</code>	<code>/HSMS/<WS-subdir>/</code>
<code>mount -F nfs <WS-address>/opt</code>	<code>/HSMS/<WS-subdir>/opt</code>
<code>mount -F nfs <WS-address>/usr</code>	<code>/HSMS/<WS-subdir>/usr</code>
<code>mount -F nfs <WS-address>/var</code>	<code>/HSMS/<WS-subdir>/var</code>
<code>mount -F nfs <WS-address>/home</code>	<code>/HSMS/<WS-subdir>/home</code>

5.7.3 Zusammenhang zwischen Knoten-S0 und Archiven

HSMS-Archive können entweder einem, mehreren oder allen Knoten-S0 zugeordnet werden. Die Entscheidung für eine zentrale oder dezentrale Archivverwaltung hängt von folgenden Kriterien ab:

- Gleichzeitiges Ablaufen der Verarbeitungsaufträge:
Mehrere Archive ermöglichen das gleichzeitige Ablaufen von verschiedenen Aufträgen.
- Anzahl der zu bearbeitenden Dateien:
Viele Dateien in einem einzigen Archiv beeinträchtigen die Performance.

5.7.3.1 Ein zentrales Archiv und zentrale Knoten-S0-Deklaration

1. Im folgenden Beispiel wird ein zentrales Archiv für die Sicherung und ein zentrales Archiv für die Langzeitarchivierung eingerichtet.

```
/START-HSMS
//CREATE-ARCHIVE ARCHIVE-NAME=CENTRAL.BAC, -
//  ALLOWED-USAGE=*NODEBACKUP,..., -
//  DIRECTORY-NAME=$SYSHSMS.CENTRAL.BAC.DIR, -
//  USER-ACCESS=*ALL-USERS (ACCESS=*WRITE)
//CREATE-ARCHIVE ARCHIVE-NAME=CENTRAL.ARC, -
//  ALLOWED-USAGE=*NODEARCHIVAL,..., -
//  DIRECTORY-NAME=$SYSHSMS.CENTRAL.ARC.DIR, -
//  USER-ACCESS=*ALL-USERS (ACCESS=*WRITE)
```

Jetzt können diese Archive für alle angegebenen Knoten-S0 benutzt werden.

2. Wenn ein einziges Archiv für alle Knoten-S0 eingerichtet wird, können alle passiven Knoten mit einer Anweisung gesichert werden:

```
//BACKUP-NODE-FILES PATH-NAMES=*PATH-NAME( -
//                      NODE-ID=*,PATH=*), ...
```


5.7.3.2 Ein Archiv für einen einzigen Knoten-S0 und eine Knoten-S0-Deklaration

1. Das folgende Beispiel zeigt eine Prozedur, die für einen angegebenen Knoten-S0 ein Archiv für die Sicherung und ein Archiv für die Langzeitarchivierung einrichtet. Anschließend wird HSMS der Knoten-S0 mitgeteilt.

```

/BEGIN-PROCEDURE PARAMETERS=*YES (PROCEDURE-PARAMETERS=( &NODEID ), -
/  ESCAPE-CHARACTER=C'&' )
/
/START-HSMS
//CREATE-ARCHIVE ARCHIVE-NAME=BAC.&NODEID, -
//  ALLOWED-USAGE=*NODEBACKUP, . . . , -
//  DIRECTORY-NAME=$SYSHSMS.BAC.&NODEID..DIR, -
//  USER-ACCESS=*ALL-USERS (ACCESS=*WRITE)
//CREATE-ARCHIVE ARCHIVE-NAME=ARC.&NODEID, -
//  ALLOWED-USAGE=*NODEARCHIVAL, . . . , -
//  DIRECTORY-NAME=$SYSHSMS.ARC.&NODEID..DIR, -
//  USER-ACCESS=*ALL-USERS (ACCESS=*WRITE)
//END
/
/END-PROCEDURE

```

2. Wenn für jeden Knoten-S0 ein Archiv eingerichtet wird, muss jeder Knoten durch eine eigene HSMS-Anweisung gesichert werden:

```

//BACKUP-NODE-FILES PATH-NAMES=*PATH-NAME( -
//  NNODE-ID=<node-id>,PATH=*) , . . .

```

3. Das Restaurieren von Dateien muss ebenso Knoten-S0 für Knoten-S0 durchgeführt werden.

5.7.3.3 Multiplexbetrieb

Im Multiplexbetrieb können sich mehrere ARCHIVE-Subtasks gleichzeitig parallel dieselben MBK-Geräte und Magnetbandkassetten teilen. Dadurch wird eine größere Performance und eine optimale Ausnutzung der Bandgeräte erreicht. Außerdem werden die Magnetbandkassetten optimal gefüllt.

Der Multiplexbetrieb wird durch die Angabe `PARALLEL-RUNS=*MULTIPLEXING(..)` in der HSMS-Anweisung `BACKUP-NODE-FILES` eingestellt. Multiplexbetrieb kann auch mit den HSMS-Anweisungen `CREATE-ARCHIVE` und `MODIFY-ARCHIVE-ATTRIBUTES` auf Archivebene definiert werden.

Beim Restaurieren von Knotendateien (`//RESTORE-NODE-FILES`) müssen Sie keinen Operanden angeben, um den Multiplexbetrieb zu aktivieren. ARCHIVE berechnet automatisch die Anzahl der erforderlichen Subtasks.

Näheres zum Multiplexbetrieb siehe "[Parallele und serielle Verarbeitung in ARCHIVE](#)".

Beispiele

Im Folgenden wird gezeigt, welche verschiedenen Möglichkeiten der Benutzer hat, um den Multiplexbetrieb auszulösen und wie HSMS seine Multiplex-Konfiguration darauf aufbaut:

1. Beispiel: Sichern eines einzelnen Pfades

```
// BACKUP-NODE-FILES PATH-NAMES=*PATH-NAME(
  PATH=/<dir>,NODE-ID=<node-id>)
  OPERATION-CONTROL=*PAR(PARALLEL-RUNS=*MULTIPLEXING(NUMBER-OF-DEVICES=
    <integer 1..16>,MULTIPLEXING-FACTOR=<integer 2..14> oder *AUTOMATIC))
```

Multiplexbetrieb findet in diesem Fall nicht statt, da keine parallele Bearbeitung möglich ist. Es wird aber eine Multiplexumgebung erzeugt (ein Subtask für ein Bandlaufwerk), da die Sicherungsdatei immer das Format „gemultiplext“ haben muss. Dadurch ist später das Fortsetzen eines gemultiplexten Bandes möglich.

2. Beispiel: Restaurieren einer einzelnen Datei oder mehrerer Dateien, die vom selben Subtask gesichert wurden

Wenn nur eine einzelne Datei restauriert werden soll, kann nur ein einziger Subtask arbeiten. Deshalb braucht keine Multiplexumgebung erstellt werden. Es wird nur ein Subtask erzeugt, der direkt mit dem Gerät zusammenarbeitet (wie bei Läufen ohne Multiplexbetrieb).

Dasselbe trifft auch zu, wenn alle zu restaurierenden Dateien von derselben Subtask gesichert wurden. Solche Dateien können nicht parallel restauriert werden, da sie sequenziell auf dem Band abgelegt wurden.

3. Beispiel: Restaurieren mehrerer gemultiplexter Dateien (d.h. Dateien, die von mehreren Subtasks parallel gesichert wurden)

Wenn mehrere Dateien, die alle zur selben Sicherungsversion gehören, von verschiedenen Subtasks gesichert wurden, können alle Dateien parallel restauriert werden (Die Dateien sind alle auf dem Band gemischt und während des Restaurierens wird das Band nicht zurückgespult).

ARCHIVE berechnet automatisch die Anzahl der benötigten Subtasks und erstellt die Umgebung, die einen parallelen Restore ermöglicht. Wenn 2 Dateien restauriert werden müssen, werden maximal nur 2 Subtasks benötigt, auch wenn die Sicherungsversion mit 14 Subtasks erstellt wurde. ARCHIVE verwendet immer möglichst wenig Subtasks.

Veranschaulichung

- Die Datei F1 wird vom Subtask T1 gesichert, F2 von T2, F3 von T3 und F4 von T4 auf dem Gerät G1 in der Sicherungsdatei SF1, Sicherungsversion SV1.
- Die Datei F5 wird von T1 gesichert, F6 von T2 auf dem Gerät G1 in der Sicherungsdatei SF2, Sicherungsversion SV2.

Zum Restaurieren aller Dateien werden 4 Subtasks für die Sicherungsversion SV1 und 2 Subtasks für SV2 benötigt.

Wenn PARALLEL-RUNS=1 ist, erzeugt ARCHIVE 4 Subtasks ($=\max(4,2)$) und 1 Bandlaufwerk-Task.

Wenn PARALLEL-RUNS=2 ist, erzeugt ARCHIVE 8 Subtasks und 2 Bandlaufwerk-Tasks; für beide Geräte wird derselbe Multiplexfaktor verwendet, da ARCHIVE nicht weiß, welches Bandlaufwerk die Sicherungsdatei SF1 bzw. SF2 bearbeitet. Zum Restaurieren von F1, F2, F3 und F4 werden 2 Subtasks benötigt ($=\max(2,2)$). Dieselbe Regel gilt für das Berechnen der Konfiguration; erzeugt werden: PARALLEL-RUNS x (2 Subtasks + 1 Bandlaufwerk).

5.7.4 Beispiele

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Zentrale Vollsicherung des lokalen BS2000-UFS und mehrerer passiver Workstations durch den HSMS-Verwalter
- Restaurieren einer Knotendatei auf eine passive Workstation durch den HSMS-Verwalter
- Restaurieren von ausgewählten Dateibäumen in das lokale BS2000-UFS durch einen nicht-privilegierten Benutzer
- Archivieren des Dateibaums einer passiven Workstation durch den HSMS-Verwalter

Zentrale Vollsicherung des lokalen BS2000-UFS und mehrerer passiver Workstations durch den HSMS-Verwalter

Vorbereitende Tätigkeiten durchführen:

```

/START-HSMS
//MODIFY-NODE-PARAMETERS - _____ (1)
//  NODE-ID=*BS2000-UFS, -
//  HSMS-CONTROL=*YES(SYSNODEBACKUP=*STD)
//MODIFY-NODE-PARAMETERS - _____ (2)
//  NODE-ID=WORKSTA1, -
//  HSMS-CONTROL=*YES(SYSNODEBACKUP=*STD)
//MODIFY-NODE-PARAMETERS -
//  NODE-ID=WORKSTA2, -
//  HSMS-CONTROL=*YES(SYSNODEBACKUP=*STD)
//MODIFY-NODE-PARAMETERS -
//  NODE-ID=WORKSTA3, -
//  HSMS-CONTROL=*YES(SYSNODEBACKUP=*STD)
//CREATE-ARCHIVE - _____ (3)
//  ARCHIVE-NAME=CENTRAL, -
//  ALLOWED-USAGE=*NODEBACKUP, -
//  USER-ACCESS=*ALL-USERS, -
//  DIRECTORY-NAME=NODEBACKUP.CENTRAL.DIR, -
//  RETENTION-PERIOD=150, -
//  S2-DEVICE-TYPE='TAPE-C4', -
//  OPERATION-CONTROL=*PARAMETERS(PARALLEL-RUNS=2)
//MODIFY-ARCHIVE - _____ (4)
//  ARCHIVE-NAME=CENTRAL, -
//  VOLUMES=*ADD(VOLUMES=(VOLUM1,VOLUM2,...))
//MODIFY-HSMS-PARAMETERS - _____ (5)
//  VALID-PERIOD=*PERMANENT, -
//  DEFAULT-HSMS-STORAGE=*PARAMETERS(SYSNODEBACKUP=CENTRAL)
//END

```

- (1) Das lokale BS2000-UFS wird unter HSMS-Verwaltung genommen. Ihm wird das Standard-Systemarchiv „SYSNODEBACKUP“ zugewiesen.
- (2) Die Workstations „WORKSTA1“ , „WORKSTA2“ und „WORKSTA3“ werden unter HSMS-Verwaltung genommen. Ihnen wird das Standard-Systemarchiv „SYSNODEBACKUP“ zugewiesen. Die Sicherung erfolgt über NFS.
- (3) Das Archiv „CENTRAL“ wird als öffentliches Archiv für die Sicherung von Knotendateien mit Leseberechtigung eingerichtet.
Die gesicherten Knotendateien können erst nach 150 Tagen gelöscht werden. Als Datenträger werden Magnetbandkassetten vereinbart, um ein angeschlossenes Roboterarchiv nutzen zu können.
Zwei Sicherungstasks sollen gleichzeitig ablaufen.

Hinweis

Damit die Directory-Datei auch von nicht-privilegierten Benutzern verwendet werden kann, muss sie mehrbenutzbar (USER-ACCESS=*ALL-USERS bzw. *SPECIAL) sein.

- (4) Magnetbandkassetten werden in den Pool freier Datenträger des Archivs „CENTRAL“ aufgenommen.
- (5) Das Archiv „CENTRAL“ wird als globales Standard-Systemarchiv für die Sicherung von Knotendateien definiert.

Liste der Pfadnamen erstellen, die gesichert werden sollen:

Inhalt der SAM-Datei „PATH.LIST“:

```
_____ (6)
<bs2000-ufs path-name1>
<bs2000-ufs path-name2>
<bs2000-ufs ...>
WORKSTA1: /<path-name2>
WORKSTA1: /...
WORKSTA2: /<path-name1>
WORKSTA2: /<path-name2>
WORKSTA2: /...
WORKSTA3: /<path-name1>
WORKSTA3: /<path-name2>
WORKSTA3: /...
```

- (6) Der HSMS-Verwalter erstellt die SAM-Datei „PATH.LIST“ mit den Pfadnamen der zu sichernden Knotendateien, da in der BACKUP-NODE-FILES-Anweisung nur ein einziger Pfadname angegeben werden kann.

Vollsicherung starten:

```
/START-HSMS
//BACKUP-NODE-FILES - _____ (7)
//  PATH-NAMES=*FROM-FILE(LIST-FILE-NAME=PATH.LIST), -
//  ARCHIVE-NAME=*SYSNODEBACKUP, -
//  SELECT-FILES=*ALL-FILES(FROM=LATEST-BACKUPS-OR-S0), - _____ (8)
//  OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=FULL.BACKUP)
//END
```

- (7) Die Pfadnamen der zu sichernden Knotendateien werden der Datei „PATH.LIST“ entnommen. Es wird eine Vollsicherung in das Standard-Systemarchiv „*SYSNODEBACKUP“ durchgeführt.
- (8) Aus Performancegründen wird mit dem Operanden FROM=*LATEST-BACKUPS-OR-S0 gesichert: Knotendateien, die sich seit der letzten Sicherung geändert haben, werden vom Knoten-S0 gesichert; nicht geänderte Knotendateien werden von ihrer letzten Sicherung im selben Archiv kopiert. Bei der allerersten Vollsicherung darf der Operand FROM=*LATEST-BACKUPS-OR-S0 nicht angegeben werden, da zunächst die Sicherungsumgebung hergestellt werden muss.

Dezentrale Vollsicherung einer Workstation durch den HSMS-Verwalter

Die Workstation ist im POSIX unter /HSMS/WORKSTA3 eingehängt und über NFS zugreifbar.

Vorbereitende Tätigkeiten durchführen:

```

/START-HSMS
//MODIFY-NODE-PARAMETERS - _____ (1)
//  NODE-ID=WORKSTA3
//CREATE-ARCHIVE - _____ (2)
//  ARCHIVE-NAME=WORKSTA3, -
//  ALLOWED-USAGE=*NODEBACKUP, -
//  USER-ACCESS=*ALL-USERS, -
//  DIRECTORY-NAME=NODEBACKUP.WORKSTA3.DIR, -
//  RETENTION-PERIOD=150, -
//  S2-DEVICE-TYPE=`TAPE-C4`
//CREATE-ARCHIVE - _____ (3)
//  ARCHIVE-NAME=WS3DIR, -
//  ALLOWED-USAGE=*BACKUP(SAVE-FILE-STRUCTURE=*SEVERAL-SVID), -
//  DIRECTORY-NAME=BACKUP.WS3DIR.DIR, -
//  RETENTION-PERIOD=150, -
//  S2-DEVICE-TYPE=`TAPE-C4`
//MODIFY-ARCHIVE - _____ (4)
//  ARCHIVE-NAME=WORKSTA3, -
//  VOLUMES=*ADD(VOLUMES=(VOLUM1,VOLUM2,...))
//END
    
```

- (1) Die Workstation „WORKSTA3“ wird zugewiesen.
- (2) Das Archiv „WORKSTA3“ wird als Sicherungsarchiv für Knotendateien definiert. Alle Benutzer haben Leseberechtigung.

Hinweis

Damit die Directory-Datei auch von nicht-privilegierten Benutzern verwendet werden kann, muss sie mehrbenutzbar (USER-ACCESS=*ALL-USERS bzw. *SPECIAL) sein.

- (3) Das Archiv „WS3DIR“ wird für die Sicherung der Directory-Datei „NODEBACKUP.WORKSTA3.DIR“ definiert, da eine Directory-Datei (BS2000-Datei) nicht in derselben Sicherungsdatei wie die Knotendateien (UNIX-Dateien) abgelegt werden kann. Jede Sicherungsdatei des Archivs darf mehrere Sicherungsversionen enthalten. Es ist zweckmäßig, für das Directory-Datei-Archiv (WS3DIR) dieselbe Schutzfrist wie für das Knotendatei-Archiv (WORKSTA3) festzulegen.

- (4) Magnetbandkassetten werden in den Pool freier Datenträger des Archivs „WORKSTA3“ aufgenommen.

Liste der Pfadnamen erstellen, die gesichert werden sollen:

Inhalt der SAM-Datei „WORKSTA3.PATH.LIST“:

```

_____ (5)
WORKSTA3:<path-name1>
WORKSTA3:<path-name2>
WORKSTA3: . . .

```

- (5) Der HSMS-Verwalter erstellt die SAM-Datei „WORKSTA3.PATH.LIST“ mit den Pfadnamen der zu sichernden Knotendateien.

Vollsicherung starten:

```

/START-HSMS
//BACKUP-NODE-FILES - _____ (1)
//  PATH-NAMES=*FROM-FILE(LIST-FILE-NAME=WORKSTA3.PATH.LIST), -
//  SELECT-FILES=*ALL-FILES, -
//  ARCHIVE-NAME=WORKSTA3, -
//  OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=FULL.BACKUP, -
//                               WAIT-FOR-COMPLETION=*YES)
//BACKUP-FILES - _____ (2)
//  FILE-NAMES=NODEBACKUP.WORKSTA3.DIR, -
//  ARCHIVE-NAME=WS3DIR, -
//  TO-STORAGE=*S2-STORAGE-LEVEL(VOLUMES=DIRTAPE), -
//  OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=NODEBACKUP.DIR)
//END

```

- (1) Die Pfadnamen der zu sichernden Knotendateien werden der Datei „WORKSTA3.PATH.LIST“ entnommen. Es werden alle Knotendateien in das Archiv „WORKSTA3“ gesichert.
- (2) Die Directory-Datei „NODEBACKUP.WORKSTA3.DIR“ wird im Archiv „WS3DIR“ auf der Speicherebene S2 gesichert.

Restaurieren einer Knotendatei auf eine passive Workstation durch den HSMS-Verwalter

Die Knotendatei wurde mit der BACKUP-NODE-FILES-Anweisung zentral gesichert.

```

/START-HSMS
//RESTORE-NODE-FILES - _____ (1)
//  PATH-NAMES=*PATH-NAME(PATH=/DIR/FILE, -
//             NODE-ID=WORKSTA2), -
//  SELECTION-BOUNDARY=*SPECIFIED-PATHS, - _____ (2)
//  REPLACE-FILES=*YES, - _____ (3)
//  SELECT-SAVE-VERSIONS=*BY-ATTRIBUTES - _____ (4)
//  (SAVE-VERSION-DATE=00-08-27(TIME=22:19:15)), -
//  OPERATION-CONTROL=*PARAMETERS(WAIT-FOR-COMPLETION=*YES, -
//  REPORT=*FULL,OUTPUT=FULL.RESTORE)
//END

```

- (1) Die Knotendatei mit dem Pfadnamen „/DIR/FILE“ wird auf die passive Workstation „WORKSTA2“ restauriert.
- (2) Wenn die Knotendatei ein Dateiverzeichnis ist, wird nur der Indexeintrag restauriert. Die Dateien, die in diesem Dateiverzeichnis enthalten sind, werden nicht restauriert.
- (3) Eine bereits existierende Knotendatei mit demselben Pfadnamen wird beim Restaurieren überschrieben.
- (4) Diese Angabe ist nur erforderlich, wenn der Stand der zu restaurierenden Knotendatei nicht der zuletzt gesicherte sein soll.

Restaurieren von ausgewählten Dateibäumen in das lokale BS2000-UFS durch einen nicht-privilegierten Benutzer

Die Dateibäume wurden mit der BACKUP-NODE-FILES-Anweisung zentral gesichert.

```
/START-HSMS  
//SELECT-NODE-FILES - _____ (1)  
//  PATH-NAMES=*PATH-NAME(PATH=*), -  
//  OUTPUT=PATHNAMES.LIST  
//END
```

```
/START-EDT _____ (2)  
@READ `PATHNAMES.LIST` Löschen aller Pfadnamen, die nicht restauriert werden sollen  
@WRITE `PATHNAMES.LIST`  
@HALT
```

```
/START-HSMS  
//RESTORE-NODE-FILES - _____ (3)  
//  PATH-NAMES=*FROM-FILE(LIST-FILE-NAME=PATHNAMES.LIST), -  
//  SELECTION-BOUNDARY=*SPECIFIED-PATHS, -  
//  REPLACE-FILES=*YES(PROTECTION-RESPECTED=*NONE), -  
//  OPERATION-CONTROL=*PARAMETERS(WAIT-FOR-COMPLETION=*YES)  
//END
```

- (1) Zunächst werden alle Dateien des lokalen BS2000-UFS ausgewählt, um eine komplette Dateinamensliste zu erhalten. Die Dateinamensliste wird in die Datei „PATHNAMES.LIST“ ausgegeben.
- (2) Anschließend wird beispielsweise der Dateibearbeiter „EDT“ aufgerufen, um aus der Datei „PATHNAMES.LIST“ diejenigen Pfadnamen zu löschen, die nicht restauriert werden sollen.
- (3) Alle Pfadnamen, die jetzt noch in der Datei „PATHNAMES.LIST“ enthalten sind, werden restauriert. Bereits existierende Dateien mit demselben Pfadnamen werden beim Restaurieren überschrieben.

Transfer von Dateien einer Workstation auf eine andere „leere“ Workstation durch den HSMS-Verwalter

Die Workstation ist im POSIX unter /HSMS/WORKSTA3 eingehängt und über NFS zugreifbar.

Es wird auf die dezentrale Vollsicherung zurückgegriffen.

```
//START-HSMS
//RESTORE-NODE-FILES - _____ (1)
//  PATH-NAMES=*FROM-FILE(LIST-FILE-NAME=WORKSTA3.PATH.LIST), -
//  SELECTION-BOUNDARY=*ALL-FILE-SYSTEMS, - _____ (2)
//  NEW-PATH-NAMES=*BY-RULE(NEW-NODE-ID=WORKSTA9), - _____ (3)
//  OPERATION-CONTROL=*PARAMETERS(REPORT=*FULL,OUTPUT=WORKSTA9.LIST)
//END
```

- (1) Alle Dateien, die in der Datei „WORKSTA3.PATH-LIST“ aufgeführt sind, werden auf die Workstation „WORKSTA9“ übertragen.
- (2) Zusätzlich werden alle Knotendateien und Dateiverzeichnisse, die sich auf den untergeordneten Ebenen der in WORKSTA3.PATH.LIST aufgeführten Dateiverzeichnisse befinden, übertragen.
- (3) Der Transfer funktioniert nur, wenn bereits vorher die Workstation „WORKSTA9“ mit der HSMS-Anweisung MODIFY-NODE-PARAMETERS unter HSMS-Verwaltung gebracht wurde.

Archivieren des Dateibaums einer passiven Workstation durch den HSMS-Verwalter

Vorbereitende Tätigkeiten durchführen:

```
//START-HSMS
//CREATE-ARCHIVE - _____ (1)
//  ARCHIVE-NAME=ARCHIVAL, -
//  ALLOWED-USAGE=*NODEARCHIVAL, -
//  DIRECTORY-NAME=ARCHIVAL.DIR, -
//  RETENTION-PERIOD=30
//MODIFY-ARCHIVE - _____ (2)
//  ARCHIVE-NAME=ARCHIVAL, -
//  VOLUMES=*ADD(VOLUMES=(VOLUM1,VOLUM2,...))
//MODIFY-NODE-PARAMETERS -
//  NODE-ID=WORKSTA1, -
//  HSMS-CONTROL=*YES
//END
```

- (1) Das Archiv „ARCHIVAL“ wird für die Langzeitarchivierung von Knotendateien eingerichtet. Standardmäßig hat nur der Archiveigentümer darauf Zugriff. Die Directory-Datei heißt „ARCHIVAL.DIR“. Die gesicherten Knotendateien können erst nach 30 Tagen gelöscht werden.

Hinweis

Damit die Directory-Datei auch von nicht-privilegierten Benutzern verwendet werden kann, muss sie mehrbenutzbar (USER-ACCESS=*ALL-USERS bzw. *SPECIAL) sein.

- (2) Datenträger werden in den Pool freier Datenträger des Archivs „ARCHIVAL“ aufgenommen.

Archivierung starten:

```
//START-HSMS  
//ARCHIVE-NODE-FILES - _____ (3)  
//  PATH-NAMES=*PATH-NAME ( PATH=/DIR,NODE-ID=WORKSTA1 ), -  
//  SELECTION-BOUNDARY=*ALL-FILE-SYSTEMS, -  
//  ARCHIVE-NAME=ARCHIVAL, -  
//  OPERATION-CONTROL=*PARAMETERS ( REPORT=*FULL,OUTPUT=ARCHLIST )  
//END
```

- (3) Das Dateiverzeichnis „/DIR“ der Workstation „WORKSTA1“ wird archiviert. Zusätzlich werden alle Knotendateien und Dateiverzeichnisse, die sich auf den untergeordneten Ebenen des Dateiverzeichnisses „/DIR“ befinden, archiviert (SELECTION-BOUNDARY=*ALL-FILE-SYSTEMS). Die Ablage erfolgt im Archiv „ARCHIVAL“.
- Ausgegeben wird ein Report in vollem Umfang mit einer Liste aller archivierten Knotendateien. Der Report wird in die Ausgabedatei „ARCHLIST“ geschrieben.

5.8 HIPLEX und HSMS

HIPLEX ist das Konzept, mit dem mehrere Hosts zu einem Rechnerverbund zusammengefasst werden können. Dadurch kann ein Benutzer seine Aufträge an den Rechnerverbund adressieren und muss sie nicht mehr an einen der Hosts schicken. Der Vorteil dieses Konzepts liegt darin, dass selbst dann eine kontinuierliche Bearbeitung der Aufträge gewährleistet ist, wenn ein oder mehrere Hosts des Rechnerverbundes nicht zur Verfügung stehen. Dateien, die auf einem Host gesichert oder archiviert wurden, können beim Ausfall dieses Hosts auf einem anderen Host des Rechnerverbundes restauriert werden. Nachdem der ausgefallene Host wieder verfügbar ist, kann auf die Situation vor dem Ausfall zurückgegangen werden, wobei die Kontinuität der HSMS-Tätigkeiten gewährleistet ist.

Um die HSMS-Aktivitäten problemlos von einem Host des Rechnerverbundes auf einen anderen verlagern zu können, muss mit einer angepassten Umgebung gearbeitet werden: Die Daten und Metadaten, die auf dem ersten Host verfügbar sind, müssen auch auf dem Ersatz-Host verfügbar gemacht werden. Die generelle Verfügbarkeit im Rechnerverbund wird durch SM-Pubsets erreicht: Wenn sich die Daten und Metadaten auf einem SM-Pubset befinden, können die HSMS-Aktivitäten nach einem Host-Ausfall durch Importieren des SM-Pubsets auf einem Ersatz-Host fortgesetzt werden (siehe auch [Abschnitt „HSMS in einer Umgebung von SM-Pubsets“](#)).

Nachdem der HSMS-Verwalter die HSMS-Anweisung RECOVER-REQUESTS gegeben hat, werden alle Aufträge, die auf dem ausgefallenen Host erteilt wurden, in die lokalen Aufträge integriert. Mit der Anweisung SHOW-REQUESTS kann dann der Status der Aufträge zum Zeitpunkt des Host-Ausfalls ausgegeben werden. Aufträge, die auf dem ausgefallenen Host noch nicht gestartet waren, können auf einem Ersatz-Host bearbeitet werden. Aufträge, die zum Zeitpunkt des Ausfalls gerade liefen und nun unterbrochen sind, können auf dem Ersatz-Host erneut gestartet werden.

Für den Benutzer ist der Host-Wechsel transparent: Er arbeitet mit einem SM-Pubset und einigen Archiven auf diesem SM-Pubset. Nach einem Host-Ausfall werden seine Aufträge von einem anderen Host bearbeitet, wobei er selbst keine Aktionen durchführen muss.

5.8.1 Wiederherstellen von DMS-Dateien (Recovery)

Das Wiederherstellen wird für alle DMS-Dateien angeboten, die sich auf einem SM-Pubset befinden. Es basiert auf der Rekonfigurations-Fähigkeit von SM-Pubsets und auf der erweiterten HSMS-Auftragsverwaltung innerhalb eines SM-Pubsets.

Manuelles Wiederherstellen

Wenn ein Host des Rechnerverbundes nicht mehr verfügbar ist, muss der HSMS-Verwalter sicherstellen, dass das DMS auf den SM-Pubset zugreifen kann:

- Wenn der SM-Pubset als exklusives Pubset importiert war, muss der HSMS-Verwalter den SM-Pubset auf dem Ersatz-Host importieren. Die Steuerdatei, die auf dem SM-Pubset liegt, wird gelesen. Alle Metadaten (Archive, Deklarationen für das SM-Pubset, ...) werden den lokalen Metadaten hinzugefügt. Anschließend ist der SM-Pubset wieder für neue HSMS-Tätigkeiten verfügbar.
- Wenn es sich um einen Shared-SM-Pubset handelt, sind für die verbleibenden Sharer immer noch HSMS-Tätigkeiten möglich, solange auf den SM-Pubset tatsächlich zugegriffen werden kann. Bei einem Ausfall des Masters müssen die Sharer unbedingt rekonfiguriert werden. Das Subsystem MSCF stellt für eine automatische Rekonfiguration durch einen Backup-Master einen Kontrolltask bereit.

Nachdem auf den SM-Pubset wieder zugegriffen werden kann, sollte der HSMS-Verwalter entscheiden, ob die Aufträge, die von dem ausgefallenen Host gerade bearbeitet worden sind, für den Restart wieder verfügbar sein müssen. Die Aufträge des SM-Pubsets können mit der HSMS-Anweisung SHOW-REQUESTS ausgegeben werden. Für jeden Auftrag wird der sperrende Host angezeigt. Nähere Informationen über Auftragssperren in SM-Pubsets finden Sie im [Abschnitt „Auftragsverwaltung“](#).

Anschließend muss der HSMS-Verwalter die HSMS-Anweisung RECOVER-REQUESTS geben, um den Status all jener Aufträge zu normalisieren, die an einen Host gekoppelt sind und zum Wiederherstellen ausgewählt wurden. Bei der Anweisung RECOVER-REQUESTS ist beschrieben, welches Ergebnis durch das Wiederherstellen erreicht wird. Bei Shared-Pubsets kann das Wiederherstellen nur am Master-Sharer erteilt werden.

Hinweis für MSCF-Rekonfiguration

Wenn in der MSCF-Rekonfiguration ein Fehler auftritt, wird der Pubset voraussichtlich in den QUIET-Modus umgeschaltet. Aus HSMS-Sicht ist der Pubset aber immer noch in Betrieb (da er nicht exportiert wurde). Auf den Pubset kann aber nicht mehr zugegriffen werden. Der Systemadministrator sollte diese Situation unbedingt bereinigen, indem er entweder einen neuen Master für den SM-Pubset aufruft oder den Pubset auf jedem Slave exportiert.

Automatisiertes Wiederherstellen

Ein automatisiertes Wiederherstellen von HSMS-Aufträgen ist nur für Konfigurationen mit Shared-SM-Pubsets möglich. Backup-Sharer müssen in Betrieb sein, um den Ausfall eines anderen Sharers feststellen zu können. HSMS-Aufträge können nur nach einem erfolgreichem MSCF-Recovery wiederhergestellt werden.

Ein automatisiertes Wiederherstellen von HSMS-Aufträgen wird durch den Einsatz des Produkts PROP-XT erreicht. Mit PROP-XT lassen sich Ereignisse definieren, die auf Konsolmeldungen basieren. Diese Ereignisse werden in einer S-Prozedur definiert, die auf jedem Host läuft. Wenn spezifische MSCF-Meldungen festgestellt werden, die sich auf ein erfolgreiches MSCF-Recovery beziehen, wird das Wiederherstellen von HSMS-Aufträgen eingeleitet.

Nähere Informationen zu PROP-XT finden Sie im Handbuch „PROP-XT“ [\[19\]](#).

Beispiel für eine S-Prozedur zum automatisierten Wiederherstellen

Die folgende S-Prozedur beschreibt einen HSMS-Watch-Dog-Task, der automatisiert HSMS-Aufträge wiederherstellt, nachdem ein Slave-Ausfall festgestellt wurde. Nähere Informationen zu Watch-Dogs finden Sie im Handbuch „HIPLEX MSCF“ [20].

Struktur der S-Prozedur

PROP-XT-Bearbeitung beginnen;		
Operator-Rolle ROLEALL anfordern, die vorher mit SYSPRIV definiert werden muss;		
PROP-XT-Ereignis mit dem Namen DMS03B0 erstellen, das auf der Konsolmeldung DMS03B0 basiert;		
Beobachtungsschleife eingeben, die von einer Jobvariable gesteuert wird;		
	Auf Ereignis DMS03B0 warten;	
	Kommando /SHOW-SHARED-PUBSET <pubset-Id> in die Variable PUBSET-LIST ausgeben;	
	Jeden Sharer durchsuchen, der von /SHOW-SHARED-PUBSET gemeldet wird, um festzustellen, ob der lokale Host Master-Zugriff auf den Pubset hat und um den Namen des ausgefallenen Partners zu ermitteln;	
	Falls Master-Zugriff und gleichzeitig Ausfall des Slave:	
		HSMS-Anweisung RECOVER-REQUESTS ausgeben;
	Variable PUBSET-LIST löschen;	
PROP-XT ausschalten;		
Exit		

S-Prozedur

Die Erläuterungen zum Ablauf sind in der Prozedur dokumentiert.

```

/SET-PROCEDURE-OPTIONS DATA-ESCAPE-CHAR=*STD
/
/ "-----"
/ "-- Beispiel fuer einen HSMS-Watch-Dog-Task          --"
/ "--                                                  --"
/ "-- Diese SDF-P-Prozedur richtet einen Watch-Dog-Task ein, der fuer das --"
/ "-- automatisierte Wiederherstellen von HSMS-Auftraegen nach dem Ausfall--"
/ "-- eines Slave-Shareres entworfen wurde.           --"
/ "--                                                  --"
/ "-- Folgende Umgebung ist erforderlich:              --"
/ "-- * Produkt PROP-XT muss installiert und das Subsystem erzeugt sein; --"
/ "-- * Operator-Rolle ROLEALL muss mit SYSPRIV definiert worden sein:  --"
/ "--   /CREATE-OPER-ROLE ROLEALL, ROUT=*ALL          --"
/ "--   /MODIFY-OPER-ATTR USER=TSOS,ADD=ROLEALL       --"
/ "-- * Die Jobvariablen muessen verfuegbar sein, vor allem $SYSJV.HOST --"
/ "--                                                  --"
/ "-- Der Batch-Task benutzt:                          --"
/ "-- * die Jobvariable HSMS.WATCHDOG.MON zum Ueberwachen der Task      --"
/ "-- * die Datei HSMS.WATCHDOG.LST zum Ausgeben der Task              --"
/ "--                                                  --"
/ "-- Starten der Watch-Dog-Task mit dem Priv SYSHSMS:          --"
/ "-- /ENTER-PROC <filename-containing-this-proc>, JOB-NAME=HSMSWDOG --"
/ "--                                                  --"
/ "-- Anhalten der Watch-Dog-Task mit:                      --"
/ "-- /SETJV HSMS.WATCHDOG.MON,'STOP'                   --"
/ "--                                                  --"
/ "-- Der Watch-Dog-Task sollte auf jedem Host erstellt werden, der auf --"
/ "-- mindestens 1 SM-Pubset Master-Zugriff hat.             --"
/ "-- Während der Operation wartet der Watch-Dog-Task auf die Konsolmeldung--"
/ "-- DMS03B0 und gibt die HSMS-Anweisung RECOVER-REQUESTS aus, wenn die --"
/ "-- Bedingungen für den lokalen Master-Zugriff und den Slave-Ausfall --"
/ "-- eintreten.                                             --"
/ "--                                                  --"
/ "-- Hinweis:                                              --"
/ "-- Dieses Beispiel unterstuetzt nur das Wiederherstellen von Auftraegen--"
/ "-- nach Slave-Ausfaellen bei Master-Sharern. Es unterstuetzt nicht das --"
/ "-- Wiederherstellen von Auftraegen nach dem Ausfall eines Masters.  --"
/ "-----"
/
/
/ "-----"
/ "-- Deklaration der lokalen Variablen                    --"
/ "--                                                  --"
/ "-----"
/
/ " Variable SYSPOP fuer den Dialog mit PROP-XT erstellen "
/DECLARE-VARIABLE NAME=SYSPOP(TYPE=STRUCTURE)
/
/ " Variable PUBSET-LIST, um die Ausgabe von /SHOW-SHARE-PUBSET zu erhalten "
/DECLARE-VARIABLE VARIABLE-NAME=PUBSET-LIST(TYPE=*STRUCTURE(DEFINITION=-
/*DYNAMIC)),MULTIPLE-ELEMENTS=*LIST
/

```

```

/ " Variablen PUBSET und SHARER, um auf die Elemente von PUBSET-LIST "
/ " zugreifen zu koennen "
/DECLARE-VARIABLE VARIABLE-NAME=PUBSET(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC))
/DECLARE-VARIABLE VARIABLE-NAME=SHARER(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC))
/
/ " Variable LOCAL-HOST, initialisiert mit dem lokalen Hostnamen "
/DECLARE-VARIABLE VARIABLE-NAME=LOCAL-HOST(TYPE=*STRING)
/
/ " Variable CRASH-HOST fuer die Namen der ausgefallenen Hosts "
/DECLARE-VARIABLE VARIABLE-NAME=CRASH-HOST(TYPE=*STRING)
/
/ " 2 Boolean-Indikatoren, um Master- und Ausfall-Ereignisse zu ermitteln "
/DECLARE-VARIABLE VARIABLE-NAME=MASTER-IND(TYPE=*BOOLEAN)
/DECLARE-VARIABLE VARIABLE-NAME=CRASH-IND(TYPE=*BOOLEAN)
/
/ "-----"
/ "-- SYSOUT auf Datei zuweisen          --"
/ "--                                     --"
/ "-----"
/
/ASSIGN-SYSOUT TO=HSMS.WATCHDOG.LST
/
/ "-----"
/ "-- Lokalen Hostnamen mit der Jobvariable $SYSJV.HOST auflösen          --"
/ "--                                     --"
/ "-----"
/
/LOCAL-HOST = (JV('$SYSJV.HOST'))
/INFORM-OPERATOR MSG=/
    '*** HSMS-Watch-Dog auf Host &(LOCAL-HOST) gestartet ***'
/
/ "-----"
/ "-- PROP-XT-Umgebung vorbereiten          --"
/ "--                                     --"
/ "-----"
/
/BEGIN-BLOCK
/
/ " PROP-XT-Bearbeitung beginnen "
/ BEGIN-PROP-PROCESS          -
/     PROCESS-NAME=SMCHECK
/ START-PROP-OBJECT-MONITORING -
/     OBJECT-NAME=OBJ-MONITOR
/
/ " Operator-Rolle ROLEALL anfordern "
/ SEND-TO-PROP-OBJECT OBJECT-NAME=OBJ-MONITOR          -
/     ,TYPE=CMD(IMPLICIT-EVENTS=CMD-TERMINATION)        -
/     ,DATA='REQ-OPER-ROLE ROLEALL'                    -
/     ,REFERENCE-NAME=REQOPERROLE
/ WAIT-FOR-PROP-EVENTS -
/     EVENT-NAME=REQOPERROLE, TIME-LIM=10
/ IF CONDITION=(SYSPOP.MAINCODE <> '0000')
/     GOTO LABEL=PROPERR
/ ELSE-IF CONDITION=(SYSPOP.MES-ID <> 'NBR0740') -
/     OR (SYSPOP.I3 <> 'CMD0001')
/     MAINCODE=SYSPOP.I3
/     MELDUNG=MSG(MSG-ID='&MAINCODE')
/     SHOW-VARIABLE (MAINCODE,MELDUNG)
/     GOTO LABEL=PROPEND

```

```

/ END-IF
/
/ " Ereignis DMS03B0 zum Warten auf die Konsolmeldung DMS03B0 erstellen "
/ START-PROP-EVENT-MONITORING -
/     EVENT-NAME=DMS03B0 -
/     ,SELECT-EVENT=FROM-OBJECT(EVENT-DATA=*SYSTEM-MSG(MSG-ID=DMS03B0))
/
/ "-----"
/ "-- Watch-Dog-Schleife zum Verfolgen der Konsolmeldung DMS03B0 --"
/ "--"
/ "-----"
/
/ CREATE-JV HSMS.WATCHDOG.MON
/ SET-JOB-STEP
/ MODIFY-JV JV-CONTENTS=(JV-NAME=HSMS.WATCHDOG.MON),SET-VALUE='CHECK'
/
/ INFORM-OPERATOR MSG=/
/     '*** HSMS-Watch-Dog traegt Beobachtungsschleife ein ***'
/
/ WHILE ( JV('HSMS.WATCHDOG.MON') = 'CHECK' )
/
/ INFORM-OPERATOR MSG=/
/     '*** HSMS-Watch-Dog wartet auf Meldung DMS03B0 ***'
/
/ " Auf Ereignis DMS03B0 warten, Timeout nach 5 Minuten "
/ WAIT-FOR-PROP-EVENTS EVENT-NAME=DMS03B0,TIME-LIMIT=300
/
/ IF ( SYSPOP.MAINCODE <> '0B18' )
/
/     IF CONDITION=(SYSPOP.MAINCODE <> '0000')
/         GOTO LABEL=PROPERR
/     END-IF
/
/     INFORM-OPERATOR MSG=/
/     '*** HSMS-Watch-Dog ueberprueft System &(SYSPOP.I0) auf Pubset ***'//-
/     '*** &(SYSPOP.I1) ***'
/
/     "-----"
/     "-- Kommando /SHOW-SHARE-PUBSET ausgeben, umgeleitet auf --"
/     "-- OPS-Variablen --"
/     "-----"
/
/     ASSIGN-STREAM STREAM-NAME=SYSINF,TO=*VARIABLE(VARIABLE-NAME=PUBSET-LIST)
/     SHOW-SHARED-PUBSET &(SYSPOP.I1)
/     SET-JOB-STEP
/     ASSIGN-STREAM STREAM-NAME=SYSINF,TO=*DUMMY
/
/     "-----"
/     "-- Status des Pubsets fuer Partner ueberpruefen --"
/     "--"
/     "-----"
/
/     FOR PUBSET=*LIST(PUBSET-LIST)
/
/         " Sharer-Liste durchsuchen, um Master und ausgefallene Hosts zu "
/         " ermitteln "
/         MASTER-IND = FALSE
/         CRASH-IND = FALSE
/         FOR SHARER=*LIST(PUBSET.LIST)

```

```
/      " Ueberpruefen, ob der lokale Host der Master-Host ist"
/      IF (      (SHARER.PARTNER-NAME=LOCAL-HOST)  -
/              AND (SHARER.SHARER-TYPE='*MASTER') )
/              MASTER-IND = TRUE
/      END-IF
/      " Ueberpruefen, ob Partner ausgefallen ist "
/      IF (SHARER.SYS-ID='&(SYSPOP.I0)')
/              CRASH-HOST = SHARER.PARTNER-NAME
/      IF ((SHARER.SHARER-STA='*CRASH') OR (SHARER.SHARER-STA='*CHECK'))
/              CRASH-IND = TRUE
/      END-IF
/      END-IF
/      END-FOR
/
/      " Falls nicht Master, Konsolmeldung ausgeben "
/      IF NOT MASTER-IND
/              INFORM-OPERATOR MSG=-
/              '*** HSMS-Watch-Dog - Lokaler Host ist nicht Master-Sharer ***'//-
/              '*** fuer Pubset &(SYSPOP.I1) ***'
/      END-IF
/
/      " Falls kein Ausfall festgestellt wurde, Konsolmeldung senden "
/      IF NOT CRASH-IND
/              INFORM-OPERATOR MSG=/
/              '*** HSMS-Watch-Dog - Kein Ausfall fuer Partner &(CRASH-HOST) ***'//-
/              '*** festgestellt ***'
/      END-IF
/
/      " Falls Host=Master und Ausfall festgestellt wird, wird Recovery "
/      " gewuenscht "
/      IF (MASTER-IND) AND (CRASH-IND)
/              " Ausgefallenen Slave-Sharer wiederherstellen "
/              INFORM-OPERATOR MSG=/
/              '*** HSMS-Watch-Dog - Recovery fuer Partner &(CRASH-HOST) ***'//-
/              '*** auf Pubset &(SYSPOP.I1) aufgerufen***'
/              START-HSMS
//              RECOVER-REQUESTS ENV=*SYS-MAN(&(SYSPOP.I1)) -
//                                  ,HOST-NAME=&(CRASH-HOST)
//              STEP
//              END
/      END-IF
/
/      END-FOR
/      SET-JOB-STEP
/
/      FREE-VAR PUBSET-LIST
/      SET-JOB-STEP
/
/      END-IF
/
/      END-WHILE
/
/      INFORM-OPERATOR MSG=/
/              '*** HSMS-Watch-Dog - Beobachtungsschleife wird verlassen ***'
/
/END-BLOCK
/
/IF-BLOCK-ERROR
/      GOTO LABEL=PROPERR
```

```
/ELSE
/ GOTO LABEL=PROPEND
/END-IF
/
/ "-----"
/ "-- Behandlung von PROP-XT-Fehlern --"
/ "-- --"
/ "-----"
/
/PROPERR:
/ MAINCODE=MAINCODE(); SUBCODE1=SUBCODE1(); SUBCODE2=SUBCODE2()
/ MELDUNG=MSG(MSG-ID='&MAINCODE')
/ SH-VARIABLE (MAINCODE,SUBCODE1,SUBCODE2,MELDUNG)
/ SH-VARIABLE (SYSPOP)
/
/ "-----"
/ "-- Exit --"
/ "-- --"
/ "-----"
/
/PROPEND:
/ END-PROP-PROCESS
/ INFORM-OPERATOR MSG=/
/     '*** HSMS-Watch-Dog deinitialisiert ***'
/ ASSIGN-SYSOUT TO=*PRIMARY
/ EXIT-JOB MODE=NORMAL,SYSTEM-OUTPUT=NONE
/
/EXIT-PROCEDURE
```

5.8.2 HIPLEX-Fähigkeit für Knotendateien

BS2000 unterstützt die HIPLEX-Fähigkeit für Knotendateien, die von HSMS verwaltet werden. Die Knotendateien müssen entweder zum BS2000-UFS oder zu fernen passiven Knoten gehören.

Die HIPLEX-Fähigkeit für Knotendateien und für DVS-Dateien ist sehr ähnlich. Bei der HIPLEX-Fähigkeit für Knotendateien müssen ebenfalls SM-Pubsets verwendet werden.

HSMS bietet folgende Unterstützung, wenn sowohl die zu sichernden Daten als auch die Metadaten auf einem SM-Pubset abgelegt sind:

- Nach einem Host-Ausfall setzt HSMS die Aktivitäten zum Sichern/Rekonstruieren für das BS2000-UFS und für ferne passive Workstations auf einem Ersatz-Host fort, wobei das Archiv durch Importieren des SM-Pubsets verfügbar ist. Nachdem der ausgefallene Host wieder arbeitsbereit ist, ermöglicht HSMS die problemlose Rückkehr auf diesen Host.
- Aufträge, die wegen eines Systemausfalls unterbrochen wurden, startet HSMS erneut auf einem Ersatz-Host.
- HSMS akzeptiert eine Rekonfiguration von Knoten. Unter Rekonfiguration ist eine Neuorganisation von Knoten zu verstehen, wobei die betroffenen Aktivitäten von einem Host auf einen anderen Host übertragen werden, ohne dass es zu einem Systemausfall kam. Die Rekonfiguration kann entweder für den gesamten Knoten oder nur für einige POSIX-HIPLEX-Benutzer durchgeführt werden.

5.8.2.1 HSMS und das UNIX-Dateisystem (UFS)

Beim Arbeiten mit Knotendateien hat der Begriff „Dateisystem“ eine besondere Bedeutung – vor allem in der UNIX-/POSIX-Welt. In diesem Zusammenhang gilt für Dateisysteme grundsätzlich:

- Das BS2000-UFS umfasst das Root-Dateisystem und die Client-Dateisysteme, die an das Root-Dateisystem oder andere Client-Dateisysteme angehängt sind.
- Ein passiver Knoten, auf den über NFS zugegriffen wird, umfasst ein fernes passives Client-Dateisystem, das mit NFS unter das Verzeichnis /HSMS des BS2000-UFS eingehängt ist.

5.8.2.2 Allgemeine Voraussetzungen

Die folgenden Voraussetzungen müssen erfüllt sein, damit die HIPLEX-Fähigkeit für Knotendateien genutzt werden kann:

- Für jeden POSIX-Benutzer müssen die POSIX-Informationen, die im Benutzerkatalog eingetragen sind, auf allen Hosts identisch sein.
- Ein POSIX-Client-Dateisystem, dessen Behälter sich auf einem SM-Pubset befindet, muss in der Dateistruktur unter dem vereinbarten Verzeichnis „pubset“ eingehängt werden. Beispielsweise muss das Dateisystem für den SM-Pubset A unter /SM-PUB/A eingehängt werden. Nur mit Hilfe dieser Konvention kann HSMS feststellen, wo sich die zu sichernden POSIX-Daten befinden und wo die entsprechenden Metadaten gespeichert werden müssen.

Wenn ein POSIX-Dateisystem angehängt wird, dessen Pfadname /SM-PUB/A enthält, dann überprüft das POSIX-Installationsprogramm, ob sich der Dateisystem-Behälter tatsächlich auf dem SM-Pubset A befindet.

- Die Einhängpunkte müssen bei diesem Konzept auf allen Hosts identisch sein.
- Der Systemverwalter muss Widersprüche bei den Einhängpunkten innerhalb der Dateistruktur auf den verschiedenen Hosts vermeiden, indem er die Pfadnamen identisch hält.

Beim Erstellen der HSMS-Anweisungen muss die jeweilige Umgebung berücksichtigt werden. Für jede Umgebung ist eine eigene HSMS-Anweisung erforderlich.

Der Operandenwert *ALL ist nun auf eine einzige Umgebung beschränkt und kann nicht mehr alle Knoten abdecken, wenn mehrere verschiedene Umgebungen verwendet werden.

- Bei einem Systemausfall muss der Systemverwalter verschiedene Maßnahmen durchführen, damit die Aktivitäten auf einem Ersatz-Host fortgesetzt und nach Beseitigung der Ausfallursache wieder vom ursprünglichen Host übernommen werden (siehe [Abschnitt „Nutzungsmodelle“](#)).
- Die HIPLEX-Fähigkeit erfordert ein lokales Archivieren jedes einzelnen SM-Pubsets. Eine globale Archivierung ist nicht möglich, d.h. der Systemverwalter kann beispielsweise keine zentrale Archivierung in ein globales Archiv auf einem SF-Pubset durchführen.

5.8.2.3 Einschränkungen

Die HIPLEX-Fähigkeit wird wegen der POSIX-Einschränkungen nicht für das Root-Dateisystem des POSIX-Dateisystems angeboten. Die HIPLEX-Fähigkeit ist nur für die Client-Dateisysteme des POSIX-Dateisystems verfügbar.

5.8.2.4 Nutzungsmodelle

Im Folgenden sind alle unterstützten Nutzungsmodelle für das BS2000-UFS und für ferne Workstations dargestellt.

BS2000-UFS

Beschreibung der Umgebung

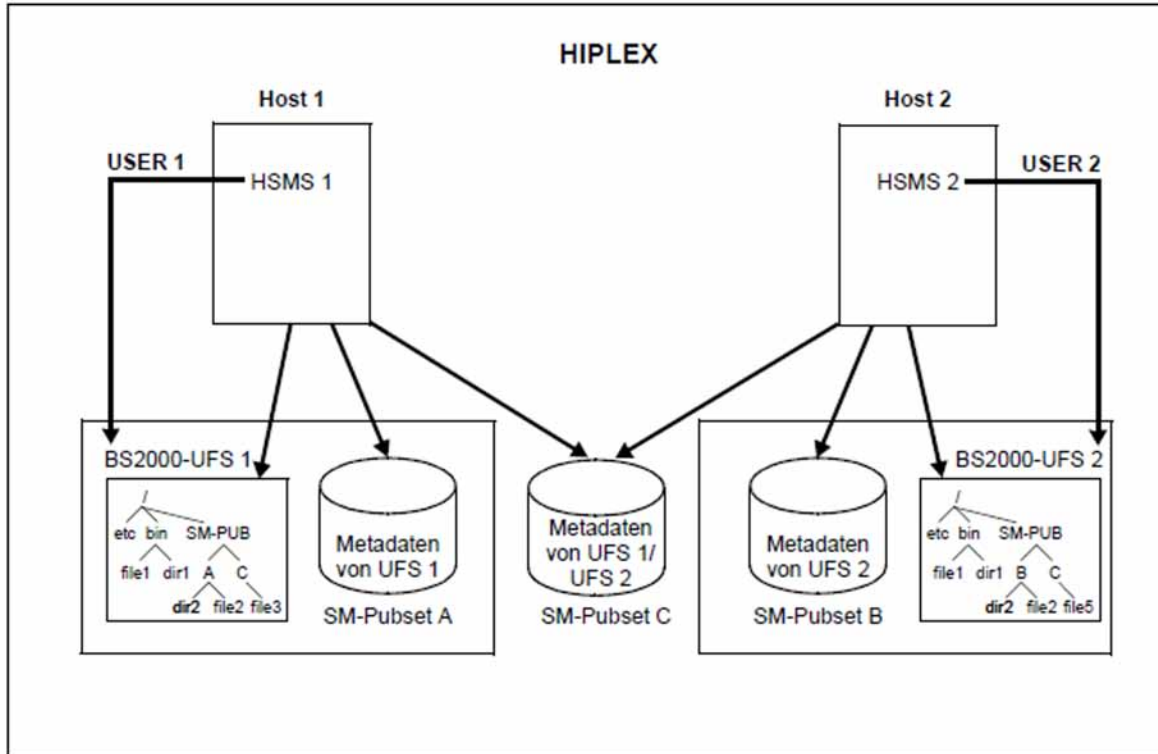


Bild 27: HSMS in einem HIPLEX für BS2000-UFS: Erreichbarkeit der POSIX-Dateien und HSMS-Metadaten

- Das HIPLEX-System umfasst mehrere Hosts, in diesem Beispiel Host 1 und Host 2.
- Jeder Host unterstützt die HSMS-Funktionen für sein BS2000-UFS.
- Der POSIX-Verwalter muss alle Dateisystem-Behälter, für die die HIPLEX-Fähigkeit gewünscht wird, auf einem der SM-Pubsets (gemeinsam benutzt oder exklusiv) ablegen. Im Bild oben sind die Dateisystem-Behälter auf dem SM-Pubset C abgelegt.
- Die betroffenen SM-Pubsets müssen unter HSMS-Kontrolle gebracht werden (CREATE-SM-PUBSET-PARAMETERS, HSMS 1 für A und C, HSMS 2 für B und C).
- Mehrere Archive müssen auf den SM-Pubsets A, B und C angelegt werden.
- Auf Host 1 muss das BS2000-UFS unter HSMS-Kontrolle gebracht werden, in jeder Umgebung entsprechend dem BS2000-UFS (SF-Pubset für das Root-Dateisystem, SM-Pubset A für den Zweig /SM-PUB/A, SM-Pubset C für den Zweig /SM-PUB/C):

```
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-F
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-M(CAT-ID=A)
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-M(CAT-ID=C)
```

- Auf Host 2 muss das BS2000-UFS unter HSMS-Kontrolle gebracht werden, in jeder Umgebung entsprechend dem BS20000-UFS (SF-Pubset für das Root-Dateisystem, SM-Pubset B für den Zweig /SM-PUB/B, SM-Pubset C für den Zweig /SM-PUB/C):

```
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-F
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-M(CAT-ID=B)
//MODIFY-NODE-PARAMETERS NODE-ID=*BS2000-UFS,ENVIRONMENT=*S-M(CAT-ID=C)
```

Während des normalen Betriebs speichert HSMS die entsprechenden Metadaten in der Umgebung, in der auch die POSIX-Daten gespeichert werden:

- Wenn ein Dateisystem gesichert wird, das sich unter dem Zweig /SM-PUB/A befindet, legt HSMS die Metadaten in einem Archiv auf dem SM-Pubset A ab.
- Wenn ein Dateisystem gesichert wird, das sich unter dem Zweig /SM-Pub/C befindet, legt HSMS die Metadaten in einem Archiv auf dem SM-Pubset C ab.

Maßnahmen nach einem Systemausfall

Nach einem Systemausfall auf Host 1 wird Host 2 als Ersatz-Host bestimmt. Nachdem folgende Maßnahmen durchgeführt wurden, kann HSMS 2 die Arbeit von HSMS 1 fortsetzen.

Maßnahmen auf Host 1	Maßnahmen auf Ersatz-Host 2
Systemausfall -> Host 1 steht nicht mehr zur Verfügung	
	<p>Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen:</p> <ol style="list-style-type: none"> 1. SM-Pubset A importieren (/IMPORT-PUBSET) 2. POSIX-Sitzung starten, falls noch keine existiert. 3. Dateisysteme von Host 1 mit dem POSIX-Installationsprogramm „anhängen“ (Datei SYSDAT.POSIX-BC.FSLIST verwenden); Jetzt stehen den Benutzern von Host 1 dieselbe Konfiguration und dieselben Zugriffsmöglichkeiten wie vor dem Systemausfall zur Verfügung. 4. HSMS laden und starten, wenn HSMS noch nicht läuft. 5. BS2000-UFS in der neuen importierten Umgebung unter HSMS-Kontrolle bringen.
	<p>Der <i>HSMS-Verwalter</i> muss folgende Maßnahme durchführen:</p> <p>HSMS-Anweisung RECOVER-REQUESTS REQUEST-DATA-ORIGIN=*NODE für SM-Pubset A geben, um die Aufträge bearbeiten zu können, die vor dem Systemausfall auf dem Host 1 eingeleitet wurden.</p>

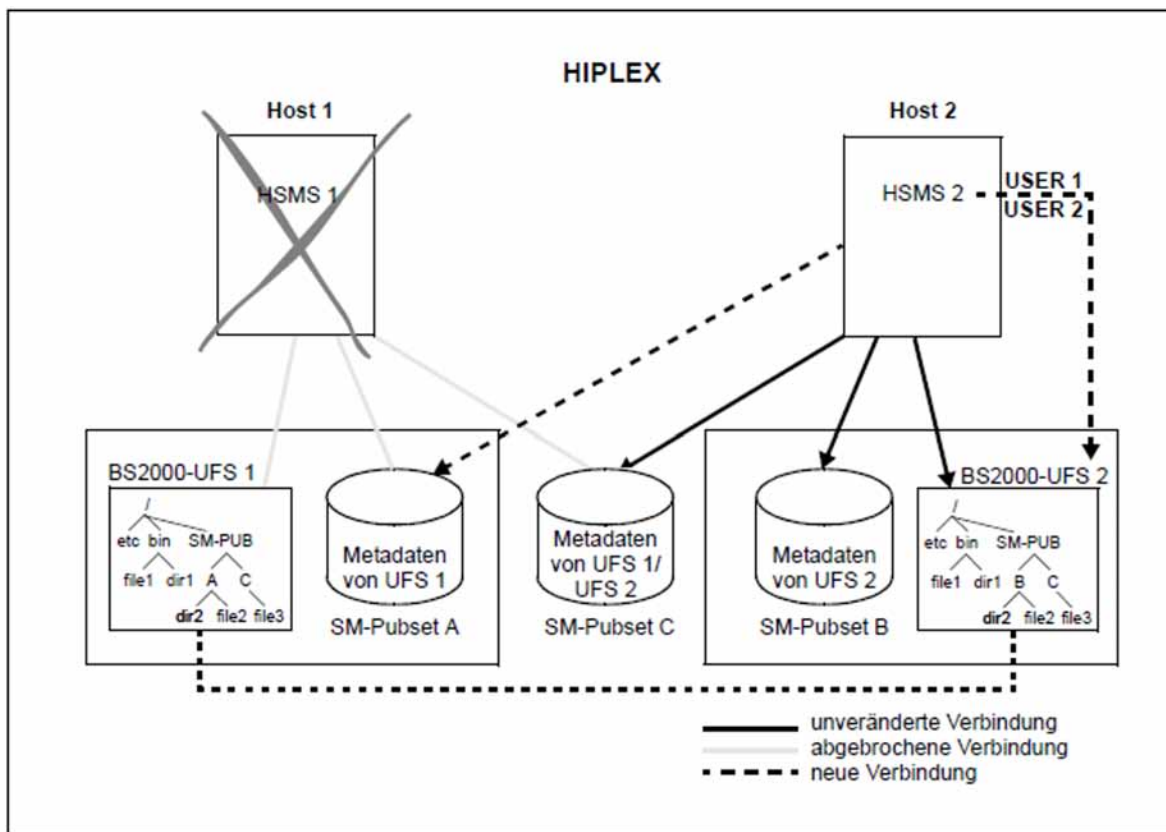


Bild 28: HSMS in einem HIPLEX für BS2000-DFS: Erreichbarkeit der POSIX-Dateien und HSMS-Metadaten nach einem Systemausfall

Maßnahmen nach Behebung des Systemausfalls

Nachdem Host 1 wieder läuft, können die Knotendateien auf den ursprünglichen Host gebracht und die HSMS-Aktivitäten zurück übertragen werden. Dazu müssen folgende Maßnahmen durchgeführt werden:

Maßnahmen auf Host 1	Maßnahmen auf Ersatz-Host 2
Ursache des Systemausfalls beseitigt -> Host 1 steht wieder zur Verfügung	
	<p>Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen:</p> <ol style="list-style-type: none"> 1. Mit dem POSIX-Installationsprogramm die Dateisysteme „löschen“ 2. BS2000-DFS im SM-Pubset A außer HSMS-Kontrolle bringen 3. SM-Pubset A exportieren (/EXPORT-PUBSET)

<p>Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen:</p>	
<ol style="list-style-type: none">1. SM-Pubset A importieren (/IMPORT-PUBSET)2. Benutzerkatalog aktualisieren: Benutzer hinzufügen, die während der Maßnahmen auf Host 2 zu Host 1 hinzugefügt wurden; dazu Informationen in der Datei SYSDAT.HIPLEX.USERS auf dem SM-Pubset A auswerten3. Dateisysteme von Host 1 mit dem POSIX-Installationsprogramm „anhängen“ (Datei SYSDAT.POSIX-BC.FSLIST verwenden)4. BS2000-UFS in der Umgebung SM-Pubset A und C unter HSMS-Kontrolle bringen, wenn dies noch nicht geschehen ist.	
<p>Der <i>HSMS-Verwalter</i> muss folgende Maßnahme durchführen:</p>	
<p>HSMS-Anweisung RECOVER-REQUESTS REQUEST-DATA-ORIGIN=*NODE für SM-Pubset A geben, um die Aufträge bearbeiten zu können, die vor dem Systemausfall auf Host 1 eingeleitet wurden.</p>	

Organisation der Dateisysteme

Im Folgenden werden die verschiedenen Möglichkeiten dargestellt, wie die POSIX-Dateisysteme organisiert werden können, damit die HIPLEX-Fähigkeit für einen Teil des Dateisystembaums gewährleistet ist.

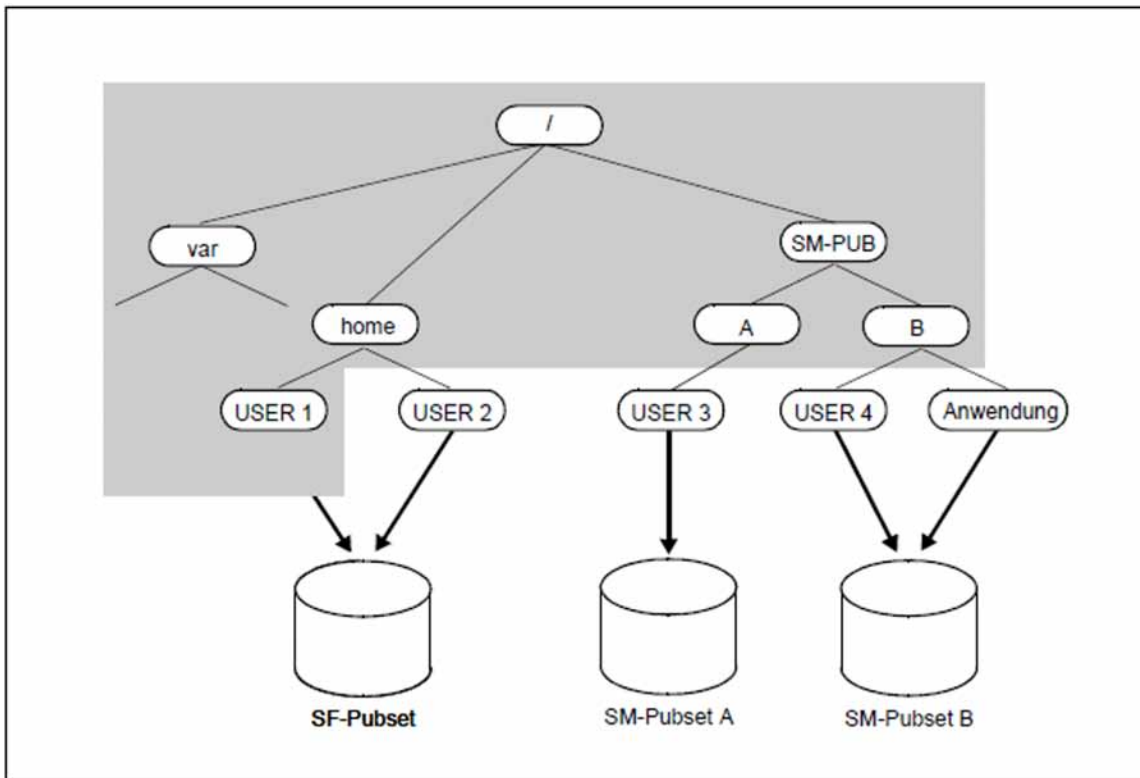


Bild 29: POSIX-/HSMS-Mechanismus: Client-Dateisysteme auf SM-Pubsets

Auf dem SF-Pubset befinden sich das Root-Dateisystem und die Dateisystem-Behälter von USER 1 und USER 2, da es sich um „Nicht-HIPLEX-Clients“ handelt.

Der Dateisystem-Behälter von USER 3 liegt auf dem SM-Pubset A. Die Dateisystem-Behälter der übrigen Benutzer sind auf dem SM-Pubset B abgelegt. Dabei müssen folgende Konventionen eingehalten werden:

- Wenn sich ein Dateisystem-Behälter auf dem SM-Pubset A befindet, muss das entsprechende Dateisystem unter dem Zweig /SM-PUB/A eingehängt werden.
- Wenn sich ein Dateisystem-Behälter auf dem SM-Pubset B befindet, muss das entsprechende Dateisystem unter dem Zweig /SM-PUB/B eingehängt werden.

Das folgende Bild zeigt, wie während des normalen Betriebs des Systems auf die POSIX-Daten und die HSMS-Metadaten zugegriffen werden kann.

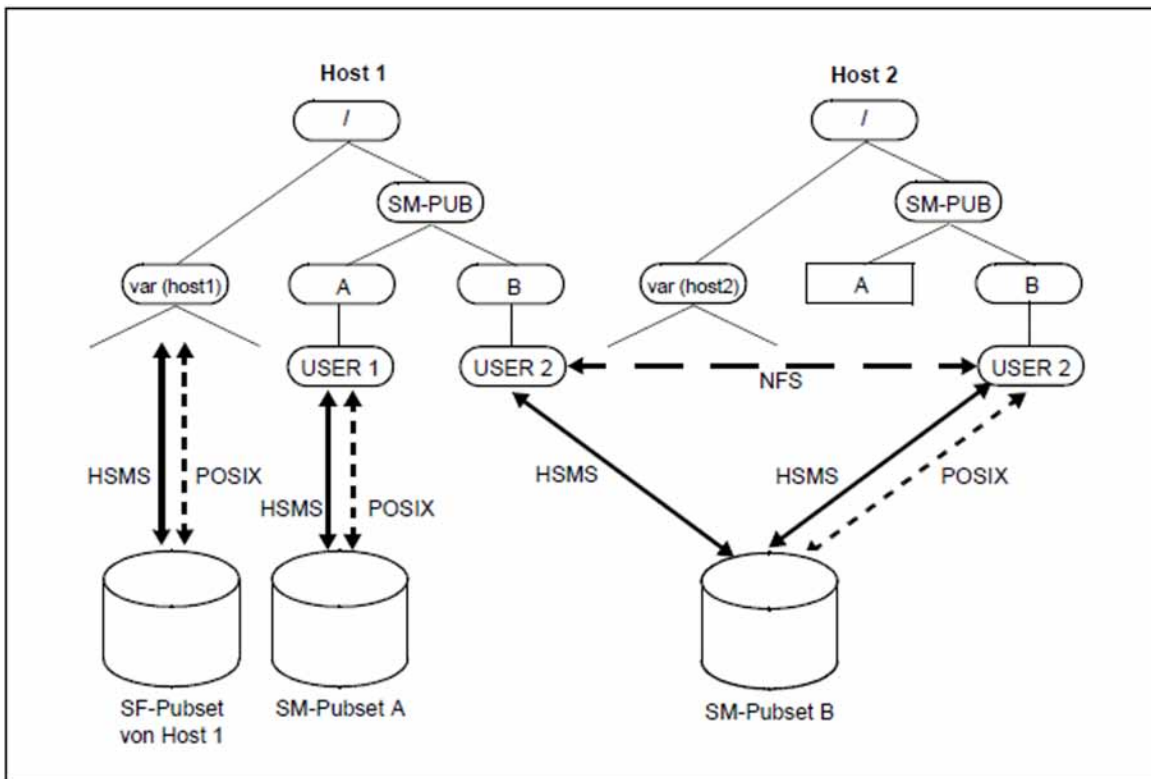


Bild 30: Zugriff auf POSIX-Dateien und HSMS-Metadaten vor einem Systemausfall

Das UNIX-Dateisystem auf dem SM-Pubset A hängt am Host 1. Das UNIX-Dateisystem auf dem SM-Pubset B hängt am Host 2.

Nach einem Systemausfall muss der Systemverwalter die auf "[BS2000-UFS: Maßnahmen nach einem Systemausfall](#)" beschriebenen Maßnahmen durchführen, damit die Aktivitäten auf einen Ersatzhost übertragen werden können. Dort wird auf die POSIX-Daten und die HSMS-Metadaten so zugegriffen, wie es im folgenden Bild zu sehen ist.

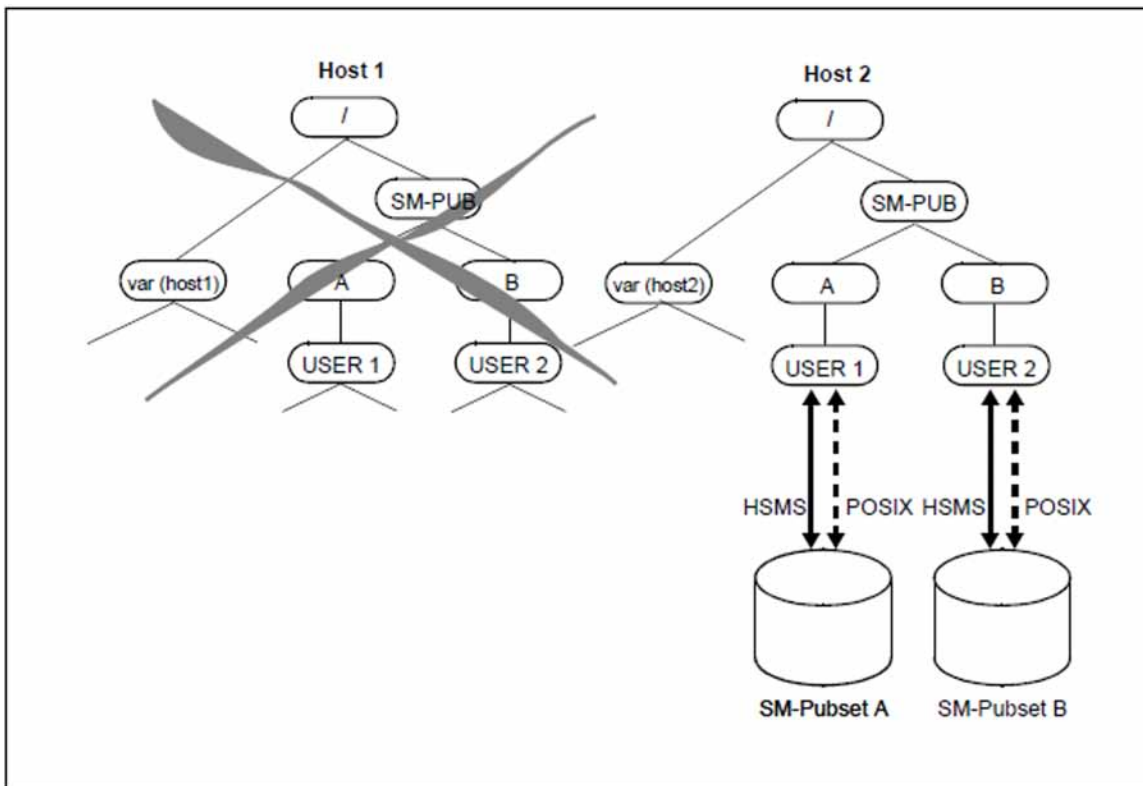


Bild 31: Zugriff auf POSIX-Dateien und HSMS-Metadaten nach einem Systemausfall

Das UNIX-Dateisystem auf dem SM-Pubset A hängt jetzt nicht mehr am Host 1, sondern am Host 2.

Ferne Workstations

Beschreibung der Umgebung

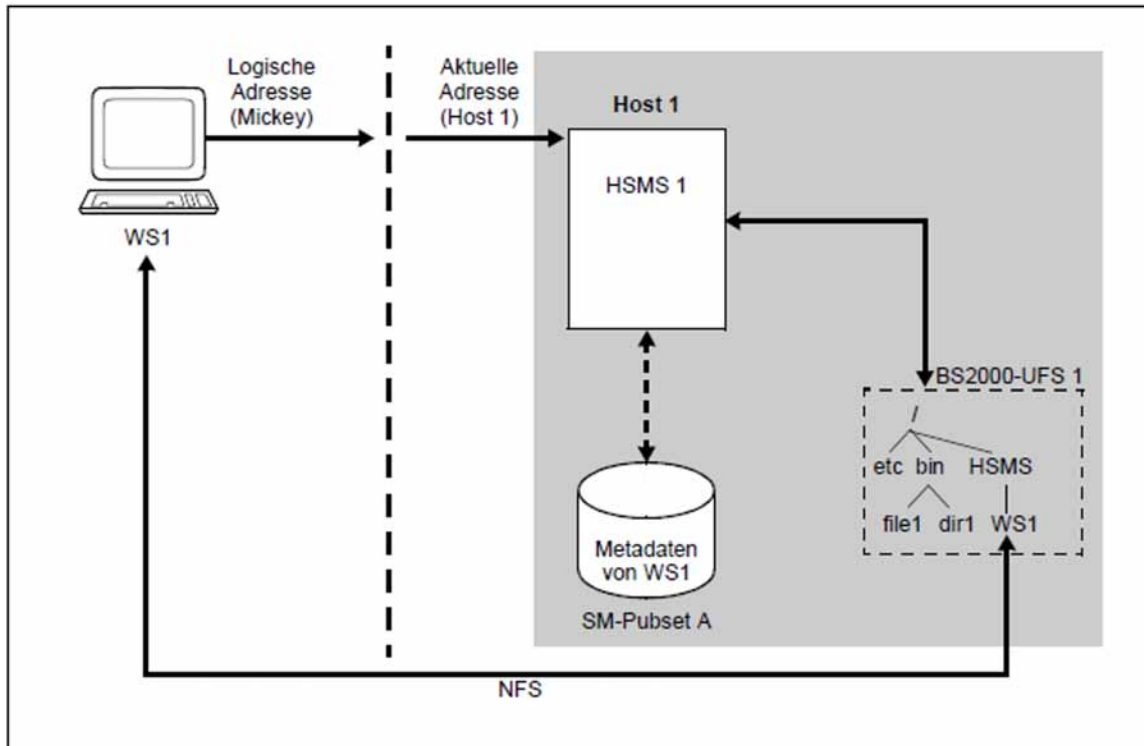


Bild 32: HSMS in einem HIPLEX für Workstations mit NFS-Zugriff: Daten-/Metadatenstrom

- Das HIPLEX-System umfasst mehrere Hosts.
- Host 1 stellt die HSMS-Funktionen für die ferne Workstation WS1 zur Verfügung.
- Der Systemverwalter muss die Daten aller Dateisysteme der Workstation auf einem einzigen SM-Pubset (gemeinsam benutzt oder exklusiv) speichern, in diesem Beispiel auf dem SM-Pubset A.
- Der betreffende SM-Pubset (hier A) muss unter HSMS-Kontrolle gebracht werden (//CREATE-SM-PUBSET-PARAMETERS, HSMS 1 für A).
- Mehrere Archive müssen auf dem SM-Pubset A angelegt werden.
- WS1 muss in der Umgebung SM-Pubset A unter HSMS 1-Kontrolle gebracht werden (//MODIFY-NODE-PARAMETERS).
- Auf die WS1-Daten wird mit NFS zugegriffen – abhängig vom Typ der Workstation (siehe Bild 32).
- Wenn auf WS1 über NFS zugegriffen wird (siehe Bild 32), muss das Dateisystem von WS1 unter /HSMS im POSIX-Root-Dateisystem eingehängt werden. Zusätzlich muss auch das BS2000-UFS unter HSMS-Kontrolle gebracht werden (//MODIFY-NODE-PARAMETERS).

Während des normalen Betriebs speichert HSMS die zugehörigen Metadaten in der Umgebung, in der die WS1-Daten gespeichert sind.

Maßnahmen nach einem Systemausfall

Nach einem Systemausfall auf Host 1 wird Host 2 als Ersatz-Host bestimmt. Nachdem folgende Maßnahmen durchgeführt wurden, kann HSMS 2 die Arbeit von HSMS 1 fortsetzen.

Maßnahmen auf Host 1	Maßnahmen auf Ersatz-Host 2
Systemausfall -> Host 1 steht nicht mehr zur Verfügung	
	<p>Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen:</p> <ol style="list-style-type: none"> 1. /CREATE-VIRTUAL-HOST Definieren des virtuellen Hosts. (Dieses BCAM-Kommando kann schon beim Starten von HIPLEX gegeben werden – also bereits vor einem Systemausfall.) 2. /BCACT HOST=Host 2 Aktivieren des virtuellen Hosts, der nun Host 1 ersetzen muss. Nach einem Systemausfall ruft dann der Systemverwalter diese Prozedur auf.) In einem TCP/IP-Netzwerk wird die Adresse des neu aktivierten virtuellen Hosts transparent verbreitet. 3. Importieren des SM-Pubsets A (/IMPORT-PUBSET) 4. Für Workstations mit Zugriff über NFS: Mit NFS die fernen Workstations „einhängen“, die vom Host 1 unter /HSMS verwaltet werden. 5. HSMS laden und starten, wenn HSMS noch nicht läuft.
	<p>Der <i>HSMS-Verwalter</i> muss folgende Maßnahme durchführen:</p> <p>HSMS-Anweisung RECOVER-REQUESTS REQUEST-DATA-ORIGIN=*NODE für SM-Pubset A geben, um die Aufträge bearbeiten zu können, die vor dem Systemausfall auf dem Host 1 eingeleitet wurden.</p>

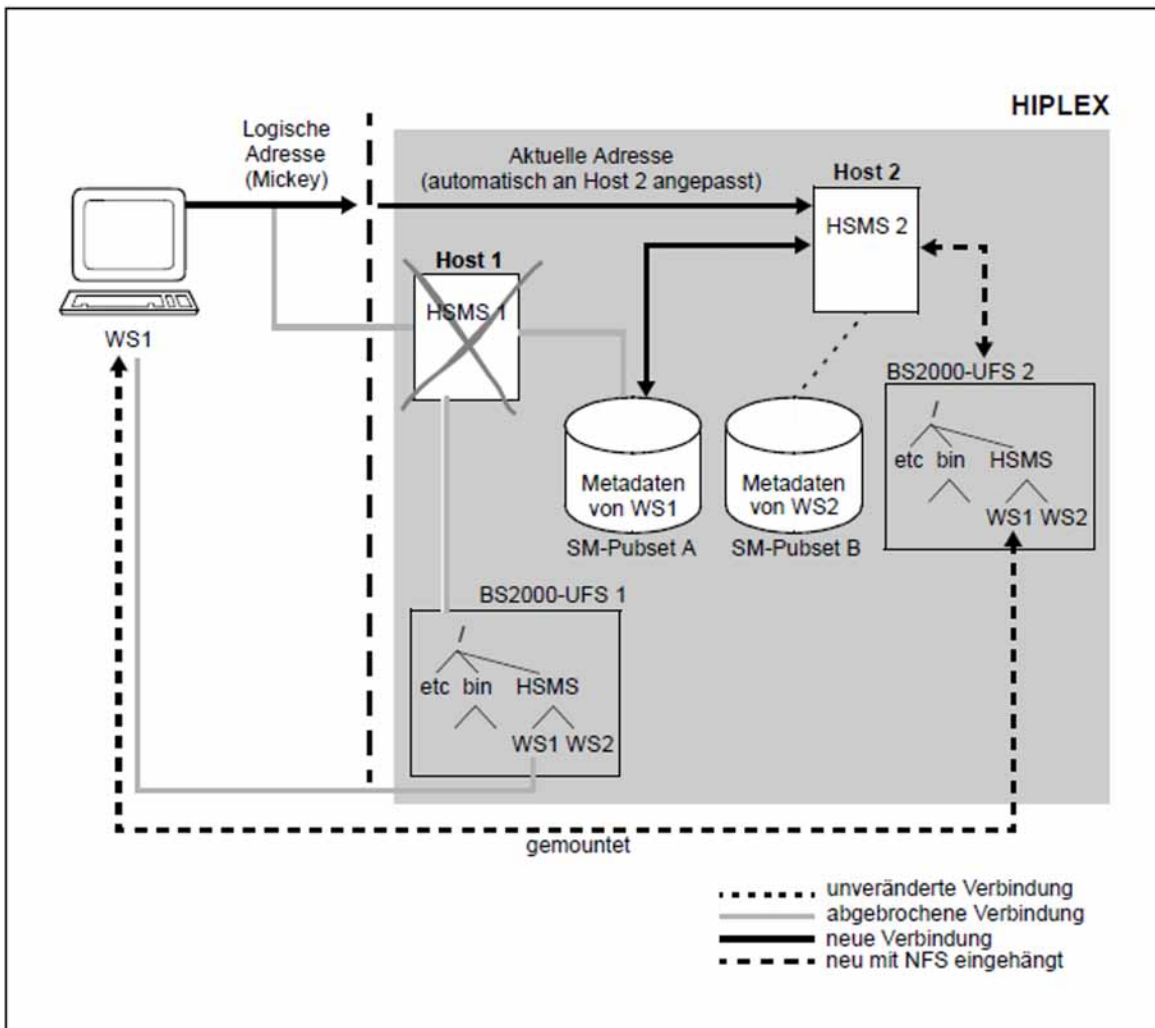


Bild 33: HIPLEX für Workstations mit NFS-Zugriff: Erreichbarkeit der Daten/Metadaten nach Systemausfall

Maßnahmen nach Behebung des Systemausfalls

Nachdem Host 1 wieder läuft, können die HSMS-Aktivitäten für die Knotendateien der Workstations auf den ursprünglichen Host zurück übertragen werden. Dazu müssen folgende Maßnahmen durchgeführt werden:

Maßnahmen auf Host 1	Maßnahmen auf Ersatz-Host 2
Ursache des Systemausfalls beseitigt -> Host 1 steht wieder zur Verfügung	
	Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen: <ol style="list-style-type: none"> 1. Nur für Workstations mit Zugriff über NFS: Mit NFS die fernen Dateisysteme „aushängen“, die auf dem SM-Pubset A gesichert wurden. 2. /CREATE-VIRTUAL-HOST Definieren des virtuellen Hosts. (Dieses BCAM-Kommando kann schon beim Starten von HIPLEX gegeben werden – also bereits vor einem Systemausfall.) 3. /BCACT HOST=Host 2 Aktivieren des virtuellen Hosts, der nun Host 1 ersetzen muss. Nach einem Systemausfall ruft dann der Systemverwalter diese Prozedur auf.) In einem TCP/IP-Netzwerk wird die Adresse des neu aktivierten virtuellen Hosts transparent verbreitet. 4. /BCDAC HOST=Host 2 Deaktivieren des virtuellen Hosts und Rückkehr auf den ursprünglichen Host 1. 5. SM-Pubset A exportieren (/EXPORT-PUBSET)
Der <i>Systemverwalter</i> muss folgende Maßnahmen durchführen:	
<ol style="list-style-type: none"> 1. SM-Pubset A importieren (/IMPORT-PUBSET) 2. Nur für Workstations mit Zugriff über NFS: Mit NFS die fernen Workstations „einhängen“, die von Host 1 unter /HSMS verwaltet werden sollen. 	

Der *HSMS-Verwalter* muss folgende Maßnahme durchführen:

1. HSMS-Anweisung RECOVER-REQUESTS REQUEST-DATA-ORIGIN=*NODE für SM-Pubset A geben, um die Aufträge bearbeiten zu können, die vor dem Systemausfall auf Host 1 eingeleitet wurden.

Organisation der Dateisysteme

Für Workstations, auf die ohne NFS zugegriffen wird, sind keine besonderen Voraussetzungen bei der Organisation der Dateisysteme der Workstation zu berücksichtigen.

Für Workstations, auf die mit NFS zugegriffen wird, müssen die Dateisysteme der Workstation unter /HSMS im POSIX-Root-Dateisystem eingehängt werden.

Das folgende Bild zeigt, wie die Dateisysteme der Workstations WS1 und WS2 bei Zugriff über NFS organisiert sein müssen, damit die HIPLEX-Fähigkeit gewährleistet ist.

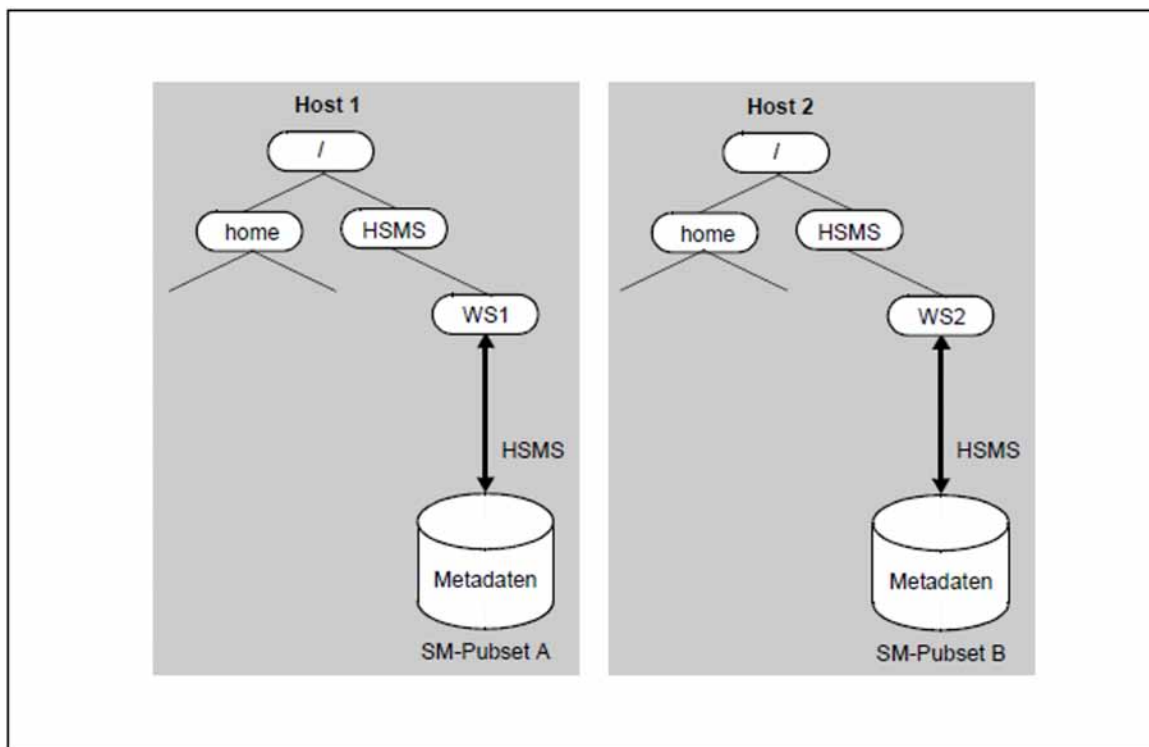


Bild 34: HIPLEX für Workstations mit NFS-Zugriff: Zugriff auf Metadaten vor einem Systemausfall

Workstation WS1 hängt am Host 1, Workstation WS2 am Host 2. HSMS greift auf die beiden Workstations über NFS zu.

Nach einem Systemausfall von Host 1 muss der Systemverwalter die auf ["Ferne Workstations: Maßnahmen nach einem Systemausfall"](#) beschriebenen Maßnahmen durchführen, damit die Aktivitäten auf einen Ersatz-Host übertragen werden können. Dort wird auf die Daten der Workstation WS1 und die HSMS-Metadaten so zugegriffen, wie es im folgenden Bild zu sehen ist.

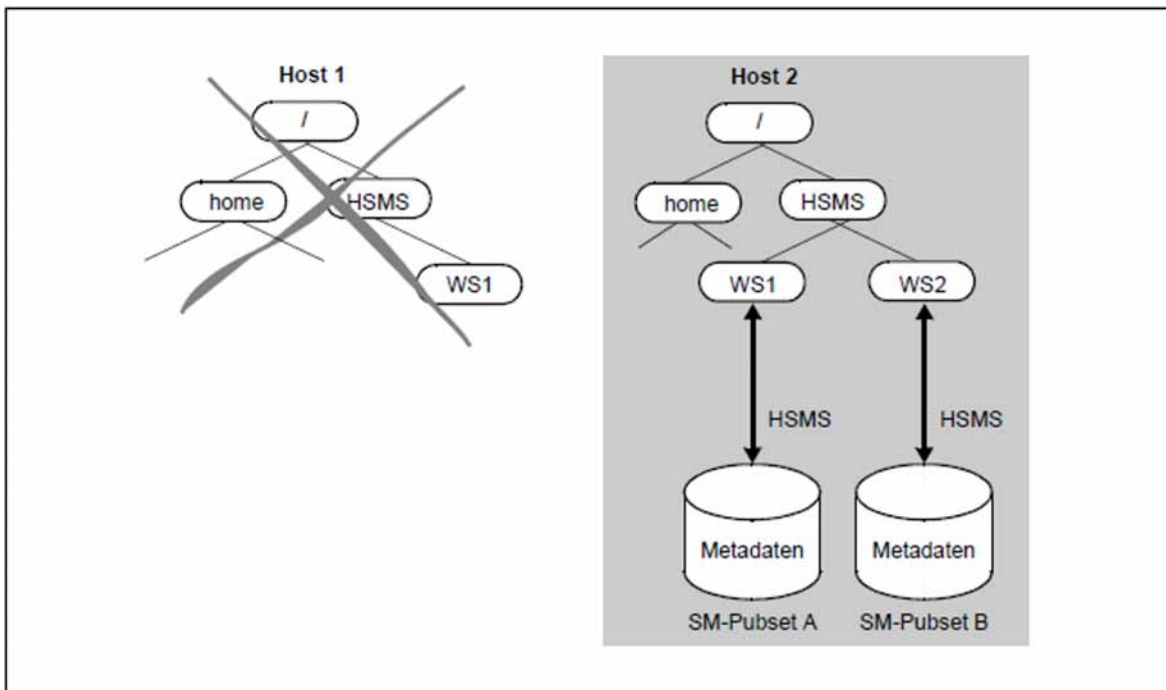


Bild 35: HIPLEX für Workstations mit NFS-Zugriff: Zugriff auf Metadaten nach einem Systemausfall

Die Workstations WS1 und WS2, die von HSMS verwaltet werden, hängen nun beide am Host 2.

5.8.3 HIPLEX-Fähigkeit für UFS-Läufe von HSMS vorbereiten

Wie bereits vorher beschrieben, sind einige SM-Pubsets nötig, um die HSMS-Metadaten für die Knotendateien zu speichern. Daher erfordert die HIPLEX-Fähigkeit auch eine Migration von einer Organisation ohne SM-Pubsets zu einer Organisation mit SM-Pubsets. Dazu muss der HIPLEX-Verwalter folgende Maßnahmen in der angegebenen Reihenfolge durchführen:

1. Daten der Knoten organisieren, die auf SM-Pubsets liegen sollen.
2. Bei POSIX-Dateisystemen die Namenskonventionen der Einhängpunkte beachten.
3. Jede Knoten-ID aus der HSMS-Kontrolle entfernen (`//MODIFY-NODE-PARAMETERS NODE-ID=<node-ID>, HSMS-CONTROL=*NO`), außer für das BS2000-UFS, das weiterhin gültig bleibt, da das POSIX-Root-Datei-system grundsätzlich in der SF-Umgebung bleibt.
4. SM-Pubsets, auf denen Daten von Knoten liegen, unter HSMS-Kontrolle bringen (`CREATE-SM-PUBSET-PARAMETERS`).
5. Mit dem POSIX-Installationsprogramm die Dateisysteme der Benutzer „anhängen“, für die POSIX-/HIPLEX-Unterstützung in der neuen SM-Umgebung bereitgestellt werden soll.
6. Neue Archive in der ausgewählten SM-Umgebung erstellen.
7. Die verschiedenen Knoten-IDs neu definieren, um sie in der richtigen SM-Umgebung unter HSMS-Kontrolle zu bringen (`//MODIFY-NODE-PARAMETERS NODE-ID= <node-id>,HSMS-CONTROL=*YES, ENVIRONMENT=*SYS-MAN(CAT-ID=<cat-id>)`).

5.9 Steuerung der Verdrängung und Verwaltung der Migrationsarchive

Im Gegensatz zu den anderen HSMS-Funktionen gibt es für die Verdrängung vielfältige Steuermöglichkeiten, aber auch eher die Notwendigkeit der Steuerung.

Anders als die Datensicherung oder die Langzeitarchivierung, die auf jedem Pubset möglich sind, ist die Verdrängung nur auf Pubsets möglich, die unter HSMS-Verwaltung genommen und der Speicherebene S0 zugeordnet wurden. Dabei sind die Dateien des Pubsets, die auf einem zugeordneten Net-Storage-Volume liegen (Speichertyp NET-STORAGE), von einer Verdrängung ausgeschlossen.

Insbesondere fällt dem HSMS-Verwalter (bzw. dem Systemverwalter) die Aufgabe zu, wichtige Systemdateien vor der Migration zu schützen (siehe [Abschnitt „Ausnahmedatei“](#)).

Verdrängung in Shared-Pubset-Systemen

Die Verdrängung einer Datei auf einem Shared-Pubset ist auch vom Slave aus möglich. Die eigentliche Auftragsbearbeitung findet aber aus Performancegründen am Master statt.

5.9.1 Verdrängung steuern

Die Steuerung geschieht „zweigleisig“; die angesprochenen Einstellungen sind in diesem Abschnitt im Einzelnen näher erläutert:

- Über die globalen Parameter (MODIFY-HSMS-PARAMETERS) kann ein HSMS-weites Migrationsarchiv festgelegt werden. Weiterhin kann die Ausnahmedatei festgelegt, das Respektieren der Migrationssperre vereinbart und das implizite Zurückholen gesteuert werden, und zwar für alle Pubsets unter HSMS-Kontrolle.
- Über die Pubset-Parameter kann pro Pubset dann einmal ein abweichendes Archiv bestimmt werden. Pubsetweise wird aber auch festgelegt, in welchem Maß für nicht-privilegierte Benutzer die Migration überhaupt zugelassen ist.

5.9.1.1 System-Migrationsarchiv

Damit Migration möglich ist, muss der HSMS-Verwalter ein Migrationsarchiv einrichten. Das Systemarchiv muss als öffentliches Archiv mit Schreibberechtigung (USER-ACCESS=*ALL-USERS und ACCESS=*WRITE) eingerichtet werden, wenn auch nicht-privilegierte Benutzer dieses Archiv für die Verdrängung nutzen sollen.

Global wird das Standard-Systemarchiv für Verdrängung zugewiesen mit:

```
//MODIFY-HSMS-PARAMETERS -  
//  DEFAULT-HSMS-STORAGE=*PAR(SYSMIGRATE=<archivname>)
```

Pubset-spezifisch kann ein anderes Migrationsarchiv zugewiesen werden mit:

```
//MODIFY-PUBSET-PARAMETERS -  
//  STORAGE-LEVEL=*S0(SYSMIGRATE=<archivname>)
```

Wenn SM-Pubsets an der Migrationsverarbeitung beteiligt sind, muss der HSMS-Verwalter Standard-Systemarchive für jeden SM-Pubset definieren, für den Migration erlaubt ist. Dies kann der HSMS-Verwalter entweder während der Definition des SM-Pubsets mit der HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS durchführen oder später, indem er mit der HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS die Parameter des SM-Pubsets ändert.

Wenn global kein Migrationsarchiv zugewiesen wird (SYSMIGRATE=*UNDEFINED), ist Verdrängung nur möglich, wenn einem Pubset ausdrücklich ein Migrationsarchiv zugewiesen wird: Der standardmäßige Rückgriff auf eine globale Voreinstellung ist nicht möglich.

5.9.1.2 Verdrängung zulassen

Wenn ein Pubset unter HSMS-Verwaltung ist, kann der HSMS-Verwalter für diesen Pubset Verdrängung für Benutzer in unterschiedlichem Maße zulassen. Dazu dient ihm die HSMS-Anweisung

```
//MODIFY-PUBSET-PARAMETERS STORAGE-LEVEL=*S0(MIGRATION=...)
```

Folgende Einstellungen für den Operanden MIGRATION sind dabei möglich:

- *ALLOWED erlaubt allen Benutzern, ihre Dateien selbst wahlweise auf die Ebene S1 oder S2 zu verdrängen.
- Bei *S2-ONLY können alle Benutzer ihre Dateien zwar verdrängen, aber nur auf die Speicherebene S2. Auf die Ebene S1 darf nur der HSMS-Verwalter verdrängen.
- *INHIBITED lässt Verdrängung insgesamt nur für den HSMS-Verwalter zu. Alle anderen Benutzer können ihre Dateien nicht selbst verdrängen.

Standardmäßig ist die Verdrängung ohne Einschränkung erlaubt (*ALLOWED).

5.9.1.3 Migrationssperre unterdrücken

Der HSMS-Verwalter muss eine Migrationssperre, die der Benutzer für seine Datei gesetzt hat, nicht respektieren (siehe [Abschnitt „Migrationssperre“](#)). Er kann sich darüber hinwegsetzen:

- In der SF-Pubset-Umgebung:

```
//MODIFY-HSMS-PARAMETERS -  
//  MIGRATION-CONTROL=*PARAMETERS(FILE-INHIBIT=*IGNORED)
```

- In einer SM-Pubset-Umgebung:

```
//MODIFY-SM-PUBSET-PARAMETERS -  
//  SM-PUBSET-ID=<sm-pubset-id>, -  
//  MIGRATION-CONTROL=*PARAMETERS(FILE-INHIBIT=*IGNORED)
```

Standardmäßig respektiert HSMS die Migrationssperre.

5.9.1.4 Ausnahmedatei

Die Verdrängung von Dateien – besonders auf die Offline-Hintergrundebene S2 – kann eine Einschränkung beim Zugriff auf diese Datei bedeuten. Dies ist nicht immer sinnvoll und gewünscht. Deshalb bieten das Datenverwaltungssystem des BS2000 und HSMS einige Möglichkeiten, bestimmte Dateien von der Verdrängung auszuschließen. Insbesondere Dateien, die zum Startup des BS2000 gebraucht werden, müssen von der Migration ausgeschlossen werden, da HSMS zu diesem Zeitpunkt nicht geladen ist und die Dateien nicht zurückgeholt werden können.

Wichtige Systemdateien werden mit einer Migrationssperre im Katalogeintrag ausgeliefert (nur mit COPY-FILE PROTECTION=*SAME kopieren!), und auch Kundendateien können mit der Migrationssperre geschützt werden. Allerdings unterscheidet HSMS beim Respektieren der Migrationssperre nicht zwischen verschiedenen Benutzern (siehe [Abschnitt „Verdrängung zulassen“](#)).

Dateien wie die Parameterdatei, CMDFILE etc. können zuverlässig vor Migration geschützt werden, indem diese Dateien (oder aus Sicherheitsgründen die ganze Benutzerkennung TSOS) durch Eintrag in eine Ausnahmedatei (Except-File) von der Migration ausgenommen werden.

Der HSMS-Verwalter listet dazu in einer SAM-Datei satzweise (nur Großbuchstaben) die Dateien auf, die HSMS nicht migrieren soll. Durch Angabe von teilqualifizierten Dateinamen können auch ganze Benutzerkennungen (wie z. B. TSOS) von der Verdrängung ausgenommen werden. Zur Bestimmung der Dateien können auch Wildcards verwendet werden.

Zugewiesen wird diese Ausnahmedatei mit:

- In der SF-Pubset-Umgebung:

```
//MODIFY-HSMS-PARAMETERS -  
//  MIGRATION-CONTROL=*PARAMETERS (EXCEPT-FILE=<dateiname>)
```

- In einer SM-Pubset-Umgebung:

```
//MODIFY-SM-PUBSET-PARAMETERS -  
//  SM-PUBSET-ID=<sm-pubset-id>, -  
//  MIGRATION-CONTROL=*PARAMETERS (EXCEPT-FILE=<dateiname>)
```

5.9.1.5 Implizites Zurückholen steuern

Der HSMS-Verwalter legt fest, ob das implizite Zurückholen von Dateien zulässig ist, die auf S2 verdrängt wurden:

```
//MODIFY-HSMS-PARAMETERS -  
//  MIGRATION-CONTROL=*PAR(RECALL-FROM-S2= . . . )
```

Dies betrifft SM- und SF-Pubset-Umgebungen.

Es gibt drei Möglichkeiten:

- *ALLOWED lässt implizites Zurückholen grundsätzlich zu.
- Bei *NOT-ALLOWED muss die Datei vor der Verarbeitung durch eine HSMS-Anweisung auf die Verarbeitungsebene geholt werden. Aufrufe durch das Datenverwaltungssystem werden zurückgewiesen.
- Bei *BATCH-ONLY werden die Dateien für Stapelaufträge automatisch zurückgeholt. In Dialogaufträgen müssen sie über HSMS-Anweisung zurückgeholt werden.

5.9.1.6 Migration / Recall mit openSM2 überwachen

HSMS stellt eine Schnittstelle zum Software Monitor openSM2 zur Verfügung, um das Migrationsverfahren zu verbessern. Das Messsystem SM2 liefert statistische Daten über die Leistung des DV-Systems und die Auslastung der Betriebsmittel. SM2 erfasst während einer Messperiode regelmäßig – beispielsweise jede Minute – HSMS-Informationen und gibt sie in eine globale SM2-Datei aus. Auf all diese HSMS-Informationen kann ein SM2-Benutzer zugreifen. Er kann einige allgemeine Parameter auswerten und ihre Entwicklung verfolgen. Nähere Informationen zu openSM2 finden Sie im Handbuch „openSM2“ [17].

Ein Migration- oder Recall-Auftrag wird auf dem Host überwacht, an dem er ausgeführt wird, aber nur dann, wenn er innerhalb derselben Messperiode gestartet und beendet wird. Wenn bei Sammelaufträgen nur ein einziger Auftrag überwacht werden soll, werden trotzdem alle Aufträge dieses Sammelauftrags überwacht.

Folgende HSMS-Informationen werden an SM2 übertragen:

- Summe der Anzahl der montierten Magnetbänder für alle Migrations- und Recall-Tätigkeiten
- Summe der Größe der migrierten Dateien (in PAM-PAGES)
- Anzahl der Migrationsläufe
- Summe der Anzahl der Extents der migrierten Dateien
- Anzahl der Dateien, die von S0 nach S1 migriert wurden
- Anzahl der Dateien, die von S0 nach S2 migriert wurden
- Anzahl der Dateien, die von S1 nach S2 migriert wurden
- Summe der Anzahl der Tage, die zwischen der letzten Benutzung einer Datei und ihrer Migration von S0 nach S1 liegen
- Summe der Anzahl der Tage, die zwischen der letzten Benutzung einer Datei und ihrer Migration von S0 nach S2 liegen
- Summe der Anzahl der Tage, die zwischen der Migration einer Datei nach S1 und ihrer Migration nach S2 liegen
- Anzahl der Recall-Läufe (explizite Recall-Läufe und implizite Recall-Ereignisse)
- Summe der Anzahl der Tage, die eine Datei auf S1 lag, bevor sie nach S0 zurückgeholt wurde
- Summe der Anzahl der Tage, die eine Datei auf S2 lag, bevor sie nach S0 zurückgeholt wurde
- Anzahl der Dateien, die explizit von S1 zurückgeholt wurden
- Anzahl der Dateien, die explizit von S2 zurückgeholt wurden
- Anzahl der Dateien, die implizit von S1 zurückgeholt wurden
- Anzahl der Dateien, die implizit von S2 zurückgeholt wurden
- Anzahl der Recall-Aufträge, die folgende Zeitspanne (in Minuten) gedauert haben: weniger als 2, 2 bis 4, 4 bis 6, 6 bis 8, 8 bis 10, 10 bis 12, 12 bis 14, 14 bis 16, 16 bis 18 und mehr als 18 Minuten
- Anzahl der Recall-Aufträge, die von ihrem Erzeugen bis zu ihrem Bearbeiten folgende Zeitspanne (in Minuten) gewartet haben: weniger als 2, 2 bis 4, 4 bis 6, 6 bis 8, 8 bis 10, 10 bis 12, 12 bis 14, 14 bis 16, 16 bis 18 und mehr als 18 Minuten
- Anzahl der Recall-Aufträge, die vom Beginn ihrer Bearbeitung bis zum ARCHIVE-Aufruf folgende Zeitspanne (in Minuten) gewartet haben: weniger als 2, 2 bis 4, 4 bis 6, 6 bis 8, 8 bis 10, 10 bis 12, 12 bis 14, 14 bis 16, 16 bis 18 und mehr als 18 Minuten
- Anzahl der ARCHIVE-Bearbeitungen für Recall-Aufträge, die folgende Zeitspanne (in Minuten) gedauert haben: weniger als 2, 2 bis 4, 4 bis 6, 6 bis 8, 8 bis 10, 10 bis 12, 12 bis 14, 14 bis 16, 16 bis 18 und mehr als 18 Minuten

- Anzahl der Magnetbandkassetten für Recall-Aufträge, die folgende Zeitspanne (in Minuten) eingehängt waren: weniger als 2, 2 bis 4, 4 bis 6, 6 bis 8, 8 bis 10, 10 bis 12, 12 bis 14, 14 bis 16, 16 bis 18 und mehr als 18 Minuten

Der HSMS-Verwalter kann mit SM2 und den gesammelten Informationen die Migrationsparameter entsprechend anpassen, wie die folgenden Beispiele zeigen.

Beispiel 1

- *Feststellung:*
Die Anzahl der Dateien pro Migrationslauf ist gering.
- *Ursache:*
Die Zeitspanne zwischen zwei Migrationsläufen ist wahrscheinlich zu kurz. Der HSMS-Verwalter startet einige unnötige Migrationsläufe.

Beispiel 2

- *Feststellung:*
Die durchschnittliche Zeit vor der Migration einer Datei von S0 ist kurz.
- *Ursache:*
Der nutzbare Speicherplatz auf der Platte ist für die Anzahl der Benutzer zu klein. Entweder gibt es zu viele Benutzer oder zu viele Dateien, die nicht migriert werden können.

Beispiel 3

- *Feststellung:*
Die durchschnittliche Verweilzeit auf S1 oder S2 ist kurz.
- *Ursache:*
Wenn gemäß dem Migrationsverfahren alle Dateien migriert werden, die seit X Tagen unbenutzt sind, dann ist die Zahl X zu klein gewählt.
- *Maßnahme:*
Wenn für X eine größere Zahl festgelegt wird, vergrößert sich der benutzte Speicherplatz auf S1 und die Zeitspanne vor einem Recall wird ebenfalls größer. Die Anzahl der zurückgeholten Dateien wird kleiner und die Migration wird insgesamt effektiver.

Beispiel 4

- *Feststellung:*
Die durchschnittliche Zeitspanne zwischen dem Erzeugen eines Recall-Auftrags und dem Beginn seiner Bearbeitung ist zu groß.
- *Ursache:*
Die Anzahl der HSMS-Subtasks ist wahrscheinlich zu klein.

Beispiel 5

- *Feststellung:*
Die durchschnittliche Zeit, während der die Magnetbandkassetten montiert sind, ist zu groß.
- *Ursache:*
Die Menge der zurückgeholten Dateien ist zu groß.

5.9.1.7 Verdrängung beschränken auf schnell migrierbare Dateien (Schnellmigration)

Der HSMS-Verwalter beschränkt die Verdrängung auf Dateien, die sich sofort, d.h. ohne Bandverarbeitung, verdrängen lassen:

```
//MIGRATE-FILES ... (MIGRATION-INFO=*REMIGRATION)
```

HSMS verdrängt von der ausgewählten Dateimenge nur Dateien, die seit dem letzten Recall nicht verändert wurden und bei denen die Sicherungsdatei noch vorhanden ist. Diese Dateien erhalten sofort, d.h. ohne einen Bandzugriff, nur den Verweis auf die noch vorhandene Sicherungsdatei (S1 oder S2). Der belegte Speicherplatz wird sofort freigegeben und die Verdrängung erfordert keine zusätzliche Speicherkapazität auf S1 oder S2.

Der HSMS-Verwalter kann mit dieser Maßnahme auf einem Pubset oder innerhalb einer Benutzerkennung sofort für freien Speicherplatz sorgen.

Außerdem wird durch Einsatz der Schnellmigration die Reorganisation von S2 nach S2 seltener benötigt.

5.9.2 Migrationsarchiv reorganisieren

Dateien, die bereits zurückgeholt, überschrieben oder gelöscht wurden, können nicht mehr zurückgeholt werden und sind deshalb im Migrationsarchiv ungültig. Deshalb müssen die Migrationsarchive regelmäßig reorganisiert werden; dies gewährleistet, dass nur gültige Dateien im Migrationsarchiv bleiben und alle ungültigen Dateien gelöscht werden. Durch Reorganisieren kann auch einem drohenden Engpass an Datenträgern für das Migrationsarchiv vorgebeugt werden.

Die Gültigkeit einer Datei wird allerdings nicht im Migrationsarchiv vermerkt; sie muss bei Zugriffen auf das Archiv (also auch beim Reorganisieren) anhand des Katalogeintrags geprüft werden. „Gültig“ ist eine Datei, die noch einen Katalogeintrag besitzt, in dem sie als „migriert“ gekennzeichnet ist und deren interner Dateiname und deren Versionsnummer mit dem in der Sicherungsdatei übereinstimmt.

Wenn der Katalog eines Pubsets zum Zeitpunkt der Prüfung nicht verfügbar ist, gelten dessen Dateien als gültig. Alle Dateien werden beim Reorganisieren ohne einen Hinweis übernommen, auch wenn sie tatsächlich ungültig sind.

Übersicht über wiedergewinnbaren Speicherplatz

Beim Reorganisieren von S1 wird der HSMS-Verwalter durch die Ausgabe einer Übersicht unterstützt. Sie zeigt, wie viel Speicherplatz auf S1 durch Sicherungsdateien, die nur noch teilweise mit gültigen Dateien gefüllt sind, belegt ist:

```
//SHOW-PUBSET-USAGE INFORMATION=*REUSABLE-S1-SPACE
```

Zu dieser Auskunftsmöglichkeit siehe Handbuch „HSMS Bd. 2“ [\[1\]](#), HSMS-Anweisung SHOW-PUBSET-USAGE.

Zum Reorganisieren der Migrationsarchive hat der HSMS-Verwalter vier Möglichkeiten, deren Wahl u.a. davon abhängig ist, ob überhaupt Daten auf S1 migriert werden oder nur auf S2:

5.9.2.1 Verdrängung von S1 auf S1

Der HSMS-Verwalter kann Sicherungsdateien auf der Ebene S1 reorganisieren, indem er sie von S1 auf S1 „verdrängt“:

```
//MIGRATE-FILES FROM-STORAGE=*S1-STORAGE-LEVEL(. . . ,
                          TO-STORAGE=*S1-STORAGE-LEVEL, . . . )
```

Dabei werden nur die gültigen Dateien in die neue Sicherungsdatei übernommen. Wenn die Sicherungsdatei leer war, d.h. nur noch ungültige Dateien enthielt, wird sie gelöscht. Eine neue Sicherungsdatei wird dann nicht angelegt. Die alte Sicherungsdatei wird nach dem erfolgreichen Migrationsvorgang in jedem Fall gelöscht, unabhängig von der vereinbarten Schutzfrist.

Beim Verdrängen von S1 auf S1 hat der HSMS-Verwalter folgende Einflussmöglichkeiten:

- Mit den Operanden MINIMUM-/MAXIMUM-DAYS-ON-S1 kann er die Auswahl auf Sicherungsdateien mit einem bestimmten Alter beschränken. Zum Beispiel kann er alle Sicherungsdateien reorganisieren, die schon 30 Tage auf S1 liegen.
- Mit MINIMUM-SIZE kann er die Verdrängung auf Sicherungsdateien ab einer bestimmten Größe beschränken.
- Durch den Operanden UNUSED-SPACE können die Sicherungsdateien nach dem Prozentsatz ausgewählt werden, in dem sie ungültige Dateien enthalten. Mit UNUSED-SPACE=60 beispielsweise werden nur die Sicherungsdateien reorganisiert, die schon zu mindestens 60% leer sind. Mit der Angabe UNUSED-SPACE=100 werden die leeren Sicherungsdateien gelöscht.
- Mit RELEASE-PAGES kann die Menge der zu verdrängenden Sicherungsdateien beschränkt werden, bis eine angebbare Zahl von PAM-Seiten durch das Reorganisieren frei geworden ist (analog der Verdrängung von S0).
- Beim Sichern, Migrieren und Reorganisieren auf die durch alle Volume-Sets unter HSMS-Kontrolle definierte S1-Ebene können keine spezifischen Volume-Set ausgewählt werden.

Allerdings kann der HSMS-Administrator die Nutzung der Volume-Sets mit folgendem Kommando einschränken:

```
/MODIFY-PUBSET-RESTRICTIONS PUBSET=<SM_pubset_catid>, -
/      PUBSET-TYPE=*S-M(VOL-SET=<catid_of_volume_set>, -
/      RESTRICTION = *NEW-FILE-ALLOCATION)
```

Hinweis zur Reorganisation von S1 auf S1 in einer SM-Pubset-Umgebung

Wenn die S1-Ebene durch alle unter HSMS-Kontrolle befindlichen Volume-Sets definiert ist, kann von S1-Ebene auf S1 nur in BS2000 OSD/BC V11.0A oder höher und mit SAVE-FILE-PROCESSING=*HSMS-V10-COMPATIBLE migriert werden.

5.9.2.2 Verdrängung von S1 auf S2

Bei der Verdrängung von S1 auf S2 wird die Speicherebene S1 noch stärker entlastet, indem die Sicherungsdateien nach der Verdrängung auf S1 gelöscht werden. Dafür steigt die Zugriffszeit, und das implizite Zurückholen im Dialog ist nur noch während der Bandverarbeitungszeiten möglich.

Der Katalogeintrag der migrierten Datei wird auf den neuen Stand gebracht, d.h. die Einträge für DEVICE-TYPE und STORAGE-LEVEL werden geändert (PUB/S2).

Auf S2 wird mit folgender HSMS-Anweisung verdrängt:

```
//MIGRATE-FILES FROM-STORAGE=*S1-STORAGE-LEVEL(. . . ,  
                                TO-STORAGE=*S2-STORAGE-LEVEL, . . . )
```

Dabei sind dieselben Einflussmöglichkeiten wie bei der Verdrängung von S1 auf S1 vorhanden.

Besonders empfiehlt sich die Verdrängung auch nach MINIMUM-DAYS-ON-S1, da die Wahrscheinlichkeit, dass eine Datei noch zur Verarbeitung gebraucht wird, mit der Zeit sinkt.

Unabhängig von der Einstellung der Option SAVE-FILE-PROCESSING beim Erstellen einer Datei auf der S1-Ebene kann diese mit einer beliebigen Einstellung der Option SAVE-FILE-PROCESSING von S1 nach S2 migriert werden.

5.9.2.3 Reorganisation eines Migrationsarchivs in S2

Sicherungsdateien mit ungültigen Dateien auf der Speicherebene S2 kann der HSMS-Verwalter ebenfalls mit der Anweisung MIGRATE-FILES reorganisieren:

```
//MIGRATE-FILES FROM-STORAGE=*S2-STORAGE-LEVEL(...)
```

Bei der Reorganisation einer Sicherungsdatei wird eine neue Sicherungsdatei erstellt und die Originalsicherungsdatei automatisch gelöscht. In einem Aufruf können mehrere Sicherungsdateien reorganisiert werden.

Bei der Reorganisation auf S2 hat der HSMS-Verwalter folgende Einflussmöglichkeiten:

- Mit den Operanden SAVE-FILE-ID kann er eine bestimmte Sicherungsdatei angeben oder die Auswahl auf Sicherungsdateien auf einen bestimmten Erstellungszeitraum beschränken. Voreingestellt ist die Auswahl aller Sicherungsdateien auf S2, mit Ausnahme der aktuellen Standard-Sicherungsdatei, die in keinem Fall reorganisiert wird.
- Durch den Operanden UNUSED-SPACE können die Sicherungsdateien nach dem Prozentsatz ausgewählt werden, in dem sie ungültige Dateien enthalten. Mit UNUSED-SPACE=60 beispielsweise werden nur die Sicherungsdateien reorganisiert, die schon zu mindestens 60% leer sind.
Mit der Angabe UNUSED-SPACE=100 werden die leeren Sicherungsdateien gelöscht.

5.9.2.4 Kopieren der Sicherungsdateien

Sicherungsdateien mit ungültigen Dateien auf der Speicherebene S2 können auch mit der Anweisung COPY-SAVE-FILE kopiert werden:

```
//COPY-SAVE-FILE . . . ,MIGRATION-STATE=*MIGRATED-FILE
```

Dabei werden ebenfalls nur die gültigen Dateien übernommen.

Die ursprüngliche Sicherungsdatei wird nach dem Kopieren nicht automatisch gelöscht; sie muss mit //MODIFY-ARCHIVE freigegeben werden. Außerdem wird empfohlen mehrere Sicherungsdateien mit dem gleichen Inhalt nicht über einen längeren Zeitraum zu archivieren. Ältere Sicherungsdateien sollten in diesem Fall baldmöglichst gelöscht werden, an-sonsten bleiben bei nachfolgenden Reorganisationen mit //MIGRATE-FILES überflüssige Sicherungsversionen der migrierten Dateien in ein und derselben Sicherungsdatei erhalten.

Während einer Reorganisation mit //COPY-SAVE-FILE wird nur eine einzige Sicherungsversion in der Ziel-Sicherungsdatei erstellt, selbst wenn die ursprüngliche Sicherungsdatei mehrere Sicherungsversionen enthält. Dadurch wird die Größe der Directory-Datei, die zum Migrationsarchiv gehört, verringert und die weitere Bearbeitungszeit reduziert.

Wenn von einer Sicherungsdatei eines Migrationsarchivs eine Sicherungskopie erstellt wurde, werden die Daten in folgenden Fällen aus der Kopie zurückgeholt:

- Beim Zurückholen durch HSMS-Aufruf aus der letzten Kopie, unabhängig davon, auf welcher Speicherebene sie steht,
- Beim impliziten Zurückholen aus der letzten Kopie auf der Speicherebene, die im Katalogeintrag der Datei vermerkt ist.

Selbstverständlich können auch Sicherungsdateien auf S1 mit //COPY-SAVE-FILE kopiert werden.

Die Reorganisation mit der Anweisung MIGRATE-FILES bietet aber in jedem Fall die besseren Steuerungs- und Auswahlmöglichkeiten.

5.9.3 Datensicherung und verdrängte Dateien

Wie aus dem [Abschnitt „BS2000-Dateien und Jobvariablen sichern“](#) hervorgeht, können bei der Datensicherung nicht nur die Katalogeinträge gesichert werden (Sicherungstyp MIGF), sondern auch die Daten der migrierten Dateien.

Wenn die Daten von migrierten Dateien mitgesichert werden, ist es für nach S1 migrierte Dateien nicht sinnvoll, die Sicherung auf dem S1-Pubset abzulegen; bei einem Ausfall dieses Pubsets sind dann nämlich die Daten und die Sicherung dieser Daten betroffen. Um auch für diesen Fall eine Sicherungsmöglichkeit auf Pubset zu haben, wird eine Sicherung auf einen beliebigen Pubset durch den Operanden TO-STORAGE in der BACKUP-FILES-Anweisung angeboten.

Beim Restore kann der HSMS-Verwalter wählen, ob nur der Katalogeintrag oder auch die Daten der verdrängten Dateien auf die Speicherebene S0 gebracht werden. Darüber hinaus kann er auch die Daten der verdrängten Dateien auf eine Migrationsebene restaurieren.

5.9.3.1 Inkonsistente Dateien

Mit //SHOW-PUBSET-USAGE INFORMATION=*MIGRATION-EVALUATION kann sich der HSMS-Verwalter darüber informieren, ob und auf welchen Pubsets solche Inkonsistenzen vorliegen. Ausgegeben werden – nach Pubsets und Benutzerkennungen geordnet – die Dateien, die im Katalog als migriert gekennzeichnet sind, aber nicht auf S1 bzw. nicht im Migrationsarchiv enthalten sind.

5.9.3.2 Behebung von Inkonsistenzen bei migrierten Dateien

Beim Zurückholen (Recall) einer Datei werden die Daten von der S1- oder S2-Speicherebene aus der Sicherungsdatei geholt, in die sie bei der letzten Migration bzw. Wetermigration bzw. Kopie innerhalb des Migrationsarchivs gebracht wurden. Diese Sicherungsdatei wird in der Lokalisierungsinformation im Katalogeintrag der Datei vermerkt. Andere Sicherungsdateien im Migrationsarchiv werden also beim Zurückholen nicht mehr als Ablageort der Daten migrierter Dateien interpretiert.

Eine Inkonsistenz für eine migrierte Datei liegt dann vor, wenn es die Sicherungsdatei, die in der Lokalisierungsinformation des Katalogeintrags der migrierten Datei vermerkt ist, im Migrationsarchiv nicht gibt bzw. die Sicherungsdatei auf der entsprechenden Speicherebene nicht mehr existiert. Es sind auch all jene Dateien inkonsistent, die keine Lokalisierungsinformation im Katalogeintrag enthalten.

Inkonsistenzen bei migrierten Dateien können mit dem Operanden INFORMATION= *MIGRATION-EVALUATION der SHOW-PUBSET-USAGE-Anweisung angezeigt werden. Diese Anzeigefunktion berücksichtigt auch die obige Definition der Inkonsistenz. Insbesondere kann es dann (bei vorangegangener Fehlbedienung) passieren, dass eine Datei als inkonsistent angezeigt wird, aber ordnungsgemäß zurückgeholt werden kann, wenn nämlich die im Katalogeintrag vermerkte Sicherungsdatei zwar noch existiert, aber nicht mehr im Migrationsarchiv vermerkt ist.

Inkonsistenzen bei migrierten Dateien können mit den HSMS-Anweisungen REPAIR-CATALOG-BY-EXCHANGE und REPAIR-CATALOG-BY-RESTORE aufgehoben werden.

5.9.4 Beispiele

In diesem Abschnitt finden Sie Beispiele zu folgenden Themen:

- Steuerung der Migration
- Migration von S1 auf S2
- Löschen von leeren Sicherungsdateien
- Reorganisieren von Sicherungsdateien auf S1
- Reorganisieren von Migrations-Sicherungsdateien auf S2
- Anwendung der SHOW-PUBSET-USAGE-Anweisung (mit Ausgabe von wiedergewinnbarem Speicherplatz und Inkonsistenzen)

Die Beispiele beziehen sich auf die HSMS-Konfiguration, die auf ["Einrichten einer HSMS-Konfiguration \(Beispiel\)"](#) beschrieben ist.

Steuerung der Migration

Die HSMS-Parameter werden so eingestellt, dass Dateien nur durch den HSMS-Verwalter migriert werden können. Zurückgeholt werden können die Dateien dagegen auch von nicht privilegierten Benutzern.

```
//MODIFY-ARCHIVE-ATTRIBUTES ARCH-NAME=$SYSHSMS.HSMS.MIG.Y, - _____ (1)
// USER-ACCESS=*ALL-USERS (ACCESS=*READ)
% HSM0003 HSMS STATEMENT COMPLETED

//MODIFY-HSMS-PARAMETERS -
// MIGR-CONTROL=*PAR (RECALL-FROM-S2=*ALLOWED) _____ (2)
% HSM0003 HSMS STATEMENT COMPLETED
```

- (1) Für das Migrationsarchiv wird nur Lesezugriff erlaubt, d.h. nur der HSMS-Verwalter als Archiveigentümer kann in dieses Archiv migrieren. Nicht-privilegierte Benutzer können Dateien aus diesem Archiv zurückholen.
- (2) Das implizite Zurückholen von S2 wird zugelassen.

Migration von S1 auf S2

```
//MIGRATE FILES -
// FROM-STOR=*S1-STOR(S1-PUB-ID=2BC, -
// ARCH-NAME=$SYSHSMS.HSMS.MI.2BY,MIN-DAYS-ON-S1=65), - _____ (1)
// OPER-CONTROL=*PAR (REPORT=*NONE)
% HSM0003 HSMS STATEMENT COMPLETED
```

- (1) Alle Dateien, die schon mindestens 65 Tage lang auf S1 liegen, werden auf S2 migriert. Dabei wird davon ausgegangen, dass migrierte Dateien, die gelegentlich benutzt werden, inzwischen auf S0 zurückgeholt wurden. D.h. nur Dateien, auf die seit mindestens 65+28 Tagen (Migration auf S1) nicht mehr zugegriffen wurde, müssen von S2 zurückgeholt werden. Alle anderen verbleiben auf S1.

Löschen von leeren Sicherungsdateien

```
//MIGRATE-FILES - _____ (1)
// FROM-STOR=*S1-STOR(S1-PUB-ID=2BC,UNUSED-SPACE=100, -
// TO-STOR=*S1-STOR), -
// OPER-CONTROL=*PAR(REPORT=*FULL, -
// OUT=HSMS.MAN.R.MGF.5,WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
...
Ausgabe des Reports HSMS.MAN.R.MGF.5 _____ (2)
```

- (1) Alle Sicherungsdateien im System-Migrationsarchiv, die zu 100% ungültige Dateien enthalten, die also leer sind, werden verdrängt und damit gelöscht.
- (2) Der Report zeigt auf der letzten Seite einen anderen Kopf, nämlich DELETED SAVE VERSIONS. Hier meldet HSMS die durch die Verdrängung nicht nur reorganisierten, sondern gelöschten Sicherungsdateien.

Reorganisieren von Sicherungsdateien auf S1

```
//MIGRATE-FILES - _____ (1)
// FROM-STOR=*S1-STOR(S1-PUB-ID=2BC,TO-STOR=*S1-STOR), -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.MGF.4, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
```

- (1) Die Sicherungsdateien des System-Migrationsarchivs auf S1 werden reorganisiert. Dabei werden alle Sicherungsdateien umgesetzt, die ungültige Dateien enthalten (RELEASE-PAGES=*MAXIMUM ist Standard).

Reorganisieren von Migrations-Sicherungsdateien auf S2

Das Archiv \$SYSHSMS.A.MIGRATE enthält 2 Sicherungsdateien und die Standard-Sicherungsdatei:

```
SHOW-ARCHIVE (SAVE-FILES)          INFORMATION          = SUMMARY
ENVIRONMENT      = SF              ARCHIVE-NAME       = $SYSHSMS.A.MIGRATE
SAVE-FILE-STATE = ANY             SAVE-FILE-STORAGE = ANY
CREATED-BEFORE  = LATEST          EXPIRATION-BEFORE = LATEST
-----
M SFID          CREA-DATE  EXP-DATE  OBS  ACCESS ST  DEVICE    #VOL #SV #RUNS
S.160521.130622 16-05-21  16-05-21 YES  OWNER  TAP  TAPE-C4   1   1   1
S.160521.133641 16-05-21  16-05-21 YES  OWNER  TAP  TAPE-C4   1   1   1
S.160521.135324 16-05-21  16-05-21 YES  OWNER  TAP  TAPE-C4   1   1   1
-----
NEXT-PAGE : +    (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
```

Die Sicherungsdateien vor dem Wechsel der Standard-Sicherungsdatei sollen mit der Anweisung MIGRATE-FILES reorganisiert werden:

```
//MIGRATE-FILES -
// FROM-STOR=*S2-STOR(SAVE-FILE-ID=*ALL, -
//   UNUSED-SPACE=*ANY,ARCHIVE-NAME=*SYSMIGRATE), -
// OPER-CONTROL=*PAR(OUT=PROT.S2REO)
```

Ausgabe des Protokolls:

```
*** MIGRATE - FILES                HSMS V11.0      SUMMARY          REPORT *** 2016-05-
21 14:34:27   PAGE    1
   REQUEST-ENVIRONMENT=SF
   REQUEST-NAME=MGF#0015 REQUEST-DATE=2016-05-21 14:20:28 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED   WITH WARNINGS
STATEMENT LISTING:
```

```
MGF          FROM-STORAGE=*S2-STORAGE-LEVEL,OPERATION-CONTROL=*PARAMETERS(OUTPUT=PROT.S2REO)
              ENVIRONMENT                               : SF
              ARCHIVE-NAME                               : $SYSHSMS.A.MIGRATE
              SAVE-FILE ATTRIBUTES
              TO-STORAGE                                  : S2-STORAGE-LEVEL
              DEVICE-TYPE                                 : TAPE-C4
              RETENTION-PERIOD                           : 0
              SAVE-VERSION ATTRIBUTES
              SAVE-VERSION-NAME                          : MIGRATE
```

```
*** MIGRATE - FILES                HSMS V11.0      SUMMARY          REPORT *** 2016-05-
21 14:34:27   PAGE    2
   REQUEST-ENVIRONMENT=SF
   REQUEST-NAME=MGF#0015 REQUEST-DATE=2016-05-21 14:20:28 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED   WITH WARNINGS
```

```
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160521.142625', VERSION '11.0'
% HSM0476 SAVE FILE 'S.160521.130622' DELETED DURING REORGANIZATION
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160521.142747', VERSION '11.0'
% ARC0096 MIGRATED FILE ':A:$TSOS.TTT1' NOT COPIED
% ARC0033 ARCHIVE SUBTASK TSN '0AAL' GENERATED
% ARC0815 SUBTASK '0' HAS TRANSFERRED '3' PAM PAGES FOR '1' FILES AND '0' JVS IN '10'
SECONDS
% HSM0475 SAVE FILE 'S.160521.133641' REORGANIZED INTO SAVE FILE 'S.160521.142747'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160521.143149', VERSION '11.0'
   SAVE FILE IDENTIFIER - S.160521.142747
```

```
   SUBSAVE
   NUMBER      VSNS
           0    TAPE04
```

```
***          E N D   O F                HSMS V11.0      SUMMARY          REPORT *** 2016-05-
21 14:34:27   ***
```

Die Sicherungsdatei S.160521.130622 enthielt keine gültigen Dateien mehr und wurde daraufhin gelöscht. Die Sicherungsdatei S.160521.133641 wurde reorganisiert in die Sicherungsdatei S.160521.142747.

```

SHOW-ARCHIVE (SAVE-FILES)          INFORMATION          = SUMMARY
ENVIRONMENT          = SF          ARCHIVE-NAME       = $SYSHSMS.A.MIGRATE
SAVE-FILE-STATE     = ANY          SAVE-FILE-STORAGE = ANY
CREATED-BEFORE      = LATEST       EXPIRATION-BEFORE = LATEST
-----
M SFID                CREA-DATE  EXP-DATE  OBS  ACCESS ST  DEVICE      #VOL #SV #RUNS
S.160521.135324      16-05-21  16-05-21  YES  OWNER  TAP  TAPE-C4     1   1   1
S.160521.142747      16-05-21  16-05-21  YES  OWNER  TAP  TAPE-C4     1   1   1
-----
NEXT-PAGE : +      (+, -, ++, --, E)
% HSM0012 END OF OUTPUT LIST REACHED
    
```

Anwendung der SHOW-PUBSET-USAGE-Anweisung

Dateien werden verdrängt, einige dieser Dateien anschließend gelöscht. Der durch die Reorganisation wiedergewinnbare Platz wird ausgegeben.

Außerdem wird eine Inkonsistenz im Migrationsarchiv erzeugt und ausgegeben.

```

//START-HSMS
//MIGRATE-FILES - _____ (1)
// FROM-STOR=*S0-STOR(F-NAMES=$MANUAL.FILE.*7*,TO-STOR=*S1-STOR) -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.MGF.6, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
Ausgabe des Reports HSMS.MAN.R.MGF.6 _____ (2)

.
Verdrängung weiterer Dateien, FILE.*6*, FILE.*5*, FILE.*4*, FILE.* _____ (3)
.
//BACKUP-FILES F-NAMES=$MANUAL., - _____ (4)
// TO-STOR=*S1-STOR, -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.BCF.7, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
Ausgabe des Reports HSMS.MAN.R.BCF.7 _____ (5)

//END
% HSM0014 HSMS PROGRAM TERMINATED
/DELETE-FILE FILE-NAME=$MANUAL.FILE.*1* _____ (6)
/DELETE-FILE FILE-NAME=$MANUAL.FILE.*2
//SHOW-PUBSET-USAGE PUB-ID=2BC,INF=*REUS-S1-SPACE _____ (7)

```



```
% HSM0003 HSMS STATEMENT COMPLETED
//MIGRATE-FILES- _____ (8)
// FROM-STOR=*S1-STOR(S1-PUB-ID=2BC,TO-STOR=*S1-STOR), -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.MGF.7, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
Report HSMS.MAN.R.MGF.7 (Ausschnitt): _____ (9)

***      MIGRATE - FILES                HSMS V11.0          FULL          REPORT      *** 2016-08-12
14:54:39    PAGE      2
      REQUEST-ENVIRONMENT=SF
      REQUEST-NAME=MGF#OAAK REQUEST-DATE=2016-08-12 14:52:48 USER-ID=SYSHSMS  REQUEST-
STATE=COMPLETED WITH WARNINGS
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145256', VERSION='11.0'
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.17' NOT COPIED
% ARC0033 ARCHIVE SUBTASK TSN '0ACQ' GENERATED
% HSM0475 SAVE FILE 'S.160812.145048' REORGANIZED INTO SAVE FILE 'S.160812.145256'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145314', VERSION='11.0'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145317', VERSION='11.0'
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.16' NOT COPIED
% ARC0033 ARCHIVE SUBTASK TSN '0ACR' GENERATED
usw.
% HSM0475 SAVE FILE 'S.160812.145148' REORGANIZED INTO SAVE FILE 'S.160812.145358'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145415', VERSION='11.0'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145418', VERSION='11.0'
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.01' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.02' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.11' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.12' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.13' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.21' NOT COPIED
% ARC0096 MIGRATED FILE ':2BY:$MANUAL.FILE.22' NOT COPIED
% ARC0033 ARCHIVE SUBTASK TSN '0ACU' GENERATED
% HSM0475 SAVE FILE 'S.160812.145208' REORGANIZED INTO SAVE FILE 'S.160812.145418'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.145436', VERSION='11.0'
% HSM0473 REORGANIZATION COMPLETED. '114' PAGES ON S1 STORAGE RELEASED
```

```
//RESTORE-FILES -
// F-NAMES=$MANUAL.FILE., -
// REPLACE-FILES-AND-JV=*NO, -
// OPER-CONTROL=*PAR(REPORT=*FULL,OUT=HSMS.MAN.R.RSF.7, -
// WAIT-F-C=*YES)
% HSM0003 HSMS STATEMENT COMPLETED
Report HSMS.MAN.R.RSF.7 (Ausschnitt):

*** RESTORE - FILES          HSMS V11.0      FULL      REPORT *** 2016-08-12 14:54:48
PAGE      3
REQUEST-ENVIRONMENT=SF
REQUEST-NAME=RSF#0AAK REQUEST-DATE=2016-08-12 14:54:40 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED WITH WARNINGS

          *** CATALOG - 2BY          USER - MANUAL          ***
          FILE/JOB VARIABLE NAME          LASTPG/   SAVE VERSION   SAVE   INPUT   SUB
OUTPUT
          VERS      SIZE      IDENTIFIER   TYPE   VSN      SAVE
DISK(S)

FILE.03          1          13 160812.145229  MIGF  0:2BC   0
% ARC0035 FILE TO BE RESTORED ALREADY EXISTS. FILE NOT REPLACED
FILE.04          1          18 160812.145229  MIGF  0:2BC   0
% ARC0035 FILE TO BE RESTORED ALREADY EXISTS. FILE NOT REPLACED
usw.
FILE.27          1          6 160812.145229  MIGF  0:2BC   0
% ARC0035 FILE TO BE RESTORED ALREADY EXISTS. FILE NOT REPLACED
FILE.01          1          3 160812.145229  MIGF  0:2BC   0
FILE.02          1          8 160812.145229  MIGF  0:2BC   0
FILE.11          1          3 160812.145229  MIGF  0:2BC   0
FILE.12          1          8 160812.145229  MIGF  0:2BC   0
usw.
FILE.22          1          8 160812.145229  MIGF  0:2BC   0

***      E N D      O F          HSMS V11.0      FULL      REPORT *** 2016-08-12
14:54:48      ***
//END
```

```
% HSM0014 HSMS PROGRAM TERMINATED
/DELETE-FILE FILE-NAME=:2BC:$TSOS.ARCHIVE.SAVE.FILE. - _____ (10)
/ 160812.145256.0
//SHOW-PUBSET-USAGE PUB-ID=2BY,INF=*MIG-EVAL _____ (11)
% HSM0433 DMS ERROR '0333' DURING 'CATALOG-'ACCESS TO FILE
':2BC:$TSOS.ARCHIVE.SAVE.FILE.160812.145256.0'
```

```
SHOW-PUBSET-USAGE                INFORMATION = MIGRATION-EVALUATION
SO-PUBSET: 2BY                    S1-PUBSET: 2BC    USER-ID: MANUAL
-----
INCONSISTENT FILE                ERROR
FILE.01                          NOT-IN-ARC
FILE.02                          NOT-IN-ARC
FILE.07                          NO-S1-DATA
FILE.11                          NOT-IN-ARC
FILE.12                          NOT-IN-ARC
FILE.13                          NOT-IN-ARC
FILE.14                          NOT-IN-ARC
FILE.15                          NOT-IN-ARC
FILE.16                          NOT-IN-ARC
FILE.17                          NOT-IN-ARC
FILE.21                          NOT-IN-ARC
FILE.22                          NOT-IN-ARC
FILE.27                          NO-S1-DATA
-----
NEXT-PAGE : __  (+, -, ++, --, E)
```

```
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Die Dateien der Benutzerkennung MANUAL mit Namen FILE.*7* werden in das System-Migrationsarchiv verdrängt.
- (2) Der von HSMS erzeugte Report des Migrationslaufs wird ausgegeben. Er enthält u.a. auch die SFID des Migrationslaufs.
- (3) Weitere Dateien der Kennung MANUAL werden verdrängt. Um für das Beispiel verschiedene Sicherungsdateien zu erzeugen, geschieht dies in getrennten Migrationsläufen.
- (4) Die vorher verdrängten Dateien werden in das System-Backup-Archiv gesichert. Weil der Operand SAVE-OPTIONS nicht angegeben ist, gilt der voreingestellte Wert des System-Backup-Archivs.
- (5) Der von HSMS erzeugte Report wird ausgegeben. Entsprechend der Standardeinstellung wurden nur die Katalogeinträge der migrierten Dateien gesichert (Sicherungstyp MIGF).
- (6) Ein Teil der zuvor verdrängten und dann gesicherten Dateien wird auf der Verarbeitungsebene gelöscht und damit ungültig.
- (7) Die Ausgabe mit SHOW-PUBSET-USAGE zeigt, dass einige Sicherungsdateien durch das Löschen der Katalogeinträge ungültige Dateien enthalten und wie viele Seiten durch eine Reorganisation einzusparen wären.

- (8) Durch das Migrieren innerhalb der S1-Ebene wird eine Reorganisation aller Sicherungsdateien auf dem Pubset 2BC angestoßen.

- (9) Der von HSMS erzeugte Report zeigt die übernommenen Sicherungsdateien mit der Meldung HSM0475. Die Dateien, die nicht übernommen wurden, weil sie ungültig (gelöscht) sind, werden mit der Meldung ARC0096 aufgelistet.
Die Sicherungsdatei mit S.160812.145048 wird z.B. in die Sicherungsdatei S.160812.145256 geschrieben. Dabei wird die Datei FILE.17 nicht übernommen, da sie ungültig ist. Die Sicherungsdatei enthält also noch die Dateien FILE.07 und .27.
Mit HSM0473 wird abschließend gemeldet, wie viele Seiten freigegeben wurden.
- (10) Alle vorher gesicherten Dateien werden auf die Verarbeitungsebene zurückgeholt, wobei existierende Dateien nicht überschrieben werden.
- (11) Der Report zeigt, dass die existierenden Dateien zu Warnmeldungen geführt haben. Dagegen wurden die vorher gelöschten Dateien auf die Verarbeitungsebene zurückgeholt.
- (12) Die Sicherungsdatei, die nach der Reorganisation die Dateien FILE.07 und .27 enthält (siehe oben), wird (nur zu Demonstrationszwecken) gelöscht.
- (13) Deshalb zeigt SHOW-PUBSET-USAGE jetzt Inkonsistenzen auf:
Die zuvor gelöschten Dateien waren bei der Reorganisation nicht übernommen worden und werden daher mit NOT-IN-ARC ausgegeben.
Die in der gelöschten Sicherungsdatei enthaltenen Dateien befinden sich nicht mehr in einer Sicherungsdatei auf S1 und werden mit NO-S1-DATA ausgegeben.

5.10 Weitere Steuerparameter

- Anzahl der Servertasks
- Größe des Common-Memory-Pools
- Wartezeiten für synchrone Aufträge und auf Reservierung
- Verarbeitungsmodus für Sicherungsdateien
- Zentrale Auftragsüberwachung am SE Server
- Aufträge aufbewahren

5.10.1 Anzahl der Servertasks

Zur Bearbeitung der Aufträge steht in HSMS eine maximale Anzahl von Servertasks zur Verfügung, die diese Aufträge gleichzeitig bearbeiten können. Die Anzahl der Servertasks kann der HSMS-Verwalter mit dem Operanden NUMBER-OF-SUBTASKS dauerhaft mit der HSMS-Anweisung MODIFY-HSMS-PARAMETERS festlegen. Der Systemverwalter kann den Wert temporär für die HSMS-Session mit der HSMS-Anweisung START-HSMS verändern.

Pro Standard-Systemarchiv sollte wenigstens ein Servertask vorgesehen werden. Weitere Servertasks können die Aufträge für private Archive verarbeiten. Zusätzliche Servertasks sollten für Leseaufträge von S1 vorgesehen werden. Die Zahl von Servertasks zusätzlich zu denen pro Archiv und für Aufträge auf S1 sollte der Zahl der verfügbaren Bandstationen entsprechen.

Nach einem Programmabbruch erfolgt kein automatischer Restart eines Servertasks. Dadurch kann die Anzahl der gerade aktiven Servertasks kleiner sein, als es gewünscht wird, und dazu führen, dass der Auftrag nicht bearbeitet wird. Wenn Aufträge angenommen, aber nicht gestartet werden, sollte der HSMS-Verwalter die Anzahl der gerade aktiven Servertasks mit der HSMS-Anweisung SHOW-HSMS-PARAMETERS überprüfen. Mit der HSMS-Anweisung MODIFY-HSMS-PARAMETERS kann er neue Servertasks erstellen, wenn dies erforderlich sein sollte.

5.10.2 Größe des Common-Memory-Pools

HSMS arbeitet mit gemeinschaftlichen Speicherbereichen (Common-Memory-Pools). Die Größe dieser Common-Memory-Pools kann der HSMS-Verwalter bzw. der Systemverwalter festlegen, und zwar durch den Operanden COMMON-MEMORY-SIZE der HSMS-Anweisungen MODIFY-HSMS-PARAMETERS und START-HSMS.

Wenn HSMS beim Laden auf Grund einer internen Kalkulation feststellt, dass die mit MODIFY-HSMS-PARAMETERS festgelegte Größe nicht ausreicht, richtet HSMS den Common-Memory-Pool in der errechneten Größe ein. Dagegen korrigiert HSMS den Wert nicht, der mit START-HSMS für die HSMS-Session vereinbart wurde.

HSMS gibt bei SHOW-HSMS-PARAMETERS VALID-PERIOD=*SESSION den aktuell gültigen, d.h. den gegebenenfalls korrigierten Wert aus.

In der Regel reicht der von HSMS errechnete Wert aus. Wenn die rechenzentrums-spezifische Anwendung jedoch zu weit von den Parametern abweicht, die der HSMS-Kalkulation zu Grunde liegen, kann es zu Engpässen kommen. Das Einrichten von Archiven ist dann zum Beispiel nicht mehr möglich. HSMS meldet diesen Zustand. Der HSMS-Verwalter sollte dann den Wert für den Common-Memory-Pool erhöhen. Der veränderte Wert wird aber erst in der nächsten HSMS-Session wirksam.

Bei der Einstellung ist zu beachten, dass die Größe des Common-Memory-Pools zu Lasten des Benutzeradressraums geht.

5.10.3 Wartezeiten für synchrone Aufträge und auf Reservierung

Bei der synchronen Verarbeitung gibt es maximale Wartezeiten, nach der die HSMS-Anweisung abgewiesen wird, falls bis dahin der Auftrag von HSMS noch nicht angenommen wurde. Falls der Auftrag nicht beendet ist, wird die HSMS-Anweisung asynchron weiterverarbeitet.

Die maximalen Wartezeiten legt der HSMS-Verwalter fest mit:

```
//MODIFY-HSMS-PARAMETERS REQUEST-WAIT-LIMITS=*PARAMETERS(...)
```

Für Dialog- und Stapelaufträge kann er die Wartezeiten getrennt bestimmen:

- bis zur Annahme des Auftrags (DIALOG- bzw. BATCH-REQUEST-TIME) oder
- bis zur Beendigung des Auftrags (DIALOG- bzw. BATCH-EXEC-TIME)

Wenn der Auftrag nach Ablauf der maximalen Wartezeit, die mit DIALOG- bzw. BATCH-REQUEST-TIME bis zur Auftragsannahme festgelegt wurde, noch nicht beendet ist, wird geprüft, ob der Auftrag schon von einer HSMS-Servertask angenommen ist. Die weitere Behandlung ist abhängig vom Ergebnis dieser Überprüfung:

- Ist der Auftrag bereits angenommen, wird auf die Auftragsbeendigung gewartet. Dabei gilt die maximale Wartezeit, die mit DIALOG- bzw. BATCH-EXEC-TIME als Wartezeit bis zur Auftragsbeendigung festgelegt wurde.
- Ist der Auftrag noch nicht angenommen, wird er abgewiesen.

Hinweis

HSMS setzt für die Anforderung der Eingabe- und Ausgabedatenträger ein SECURE-RESOURCE-ALLOCATION-Kommando mit Wartezeit ab. Diese Wartezeit ist gleich der Angabe beim HSMS-Parameter BATCH-REQUEST-TIME (siehe auch [Abschnitt „Datenträger- und Gerätereservierung“](#)).

5.10.4 Verarbeitungsmodus für Sicherungsdateien

Mit dem globalen Parameter SAVE-FILE-PROCESSING kann ab HSMS V10.0A der Verarbeitungsmodus für Sicherungsdateien auf der Speicherebene S1, auf gemeinschaftlicher Platte und Net-Storage festgelegt werden: *HSMS-V9-COMPATIBLE oder *HSMS-V10-COMPATIBLE.

Im Verarbeitungsmodus *HSMS-V9-COMPATIBLE erzeugte Sicherungsdateien können mit HSMS-Versionen kleiner als V10.0A ohne Einschränkung verarbeitet werden. Dagegen ist die Verarbeitung von Sicherungsdateien, die im Verarbeitungsmodus *HSMS-V10-COMPATIBLE erzeugt wurden, mit HSMS-Versionen kleiner als V10.0A nur für Dateien auf Band oder privater Platte möglich. Der Grund hierfür ist, dass Sicherungsdateien auf gemeinschaftlicher Platte und Net-Storage, die im Modus *HSMS-V10-COMPATIBLE erstellt wurden, eine andere Namensstruktur haben, als die im Modus *HSMS-V9-COMPATIBLE erstellten Dateien.

Standardmäßig ist der Modus *HSMS-V9-COMPATIBLE eingeschaltet. In diesem Modus steht jedoch nicht die vollständige Funktionalität von HSMS zur Verfügung. Wenn Sie die vollständige Funktionalität von HSMS nutzen wollen, müssen Sie explizit den Modus *HSMS-V10-COMPATIBLE einschalten.

Die folgende Tabelle gibt einen Überblick über die zur Verfügung stehende Funktionalität in Abhängigkeit von der Einstellung des Parameters SAVE-FILE-PROCESSING:

Funktionalität	SAVE-FILE-PROCESSING	
	*HSMS-V9-COMPATIBLE	*HSMS-V10-COMPATIBLE
Vom Lagerort unabhängige Sicherungsdateinamen (bei gemeinschaftlicher Platte und Net-Storage)	-	+
Nutzung von S1-SM-Pubsets	-	+
Aufteilen von Sicherungsdateien auf mehrere Volume-Sets innerhalb eines S1-SM-Pubsets	-	+
Folgenummer in Sicherungsdateinamen (bei Nutzung von S1-SM-Pubsets)	-	+
Fortschreiben von Sicherungsdateien auf gemeinschaftlicher Platte oder Net-Storage, die mit HSMS-Versionen < V10.0 erstellt wurden	+	-
Lesender Zugriff auf Sicherungsdateien, die mit HSMS-Versionen < V10.0 erstellt wurden	+	+
Zugriff mit HSMS-Versionen < V10.0 auf Sicherungsdateien auf gemeinschaftlicher Platte oder Net-Storage, die mit HSMS V10.0 oder höher erstellt wurden	+	-
Verteilung der Sicherungsdateien auf mehrere Volume-Sets in einer SM-Umgebung, wenn die Speicherebene S1 als *ALL-HSMS-CONTROLLED definiert ist	-	+

+ Funktion steht zur Verfügung

- Funktion steht nicht zur Verfügung

5.10.5 Zentrale Auftragsüberwachung am SE Server

Der BS2000 Backup Monitor ist als SE Management Anwendung im Hauptmenü **AAwendungen** des SE Managers integriert. Der BS2000 Backup Monitor informiert über den Status der Sicherungsaufträge, die in den BS2000-Systemen des SE Servers mit den Software-Produkten HSMS und FDDRL beauftragt wurden. Für beendete Aufträge stellt HSMS für den BS2000 Backup Monitor unter der Benutzerkennung SYSHSMS ein Ablaufprotokoll als PDF-Datei zur Verfügung.

Folgende Aufträge werden überwacht:

- Sicherung (Backup)
- Wiederherstellung (Restore) von Daten
- Migration (nur HSMS)
- Archivierung (nur HSMS)
- Kopieren von Sicherungsdateien inkl. Reorganisationsläufe von Archiven (nur HSMS)
- Exportieren und Importieren von Dateien (nur HSMS)
- Versions-Backup (nur HSMS)
- Versions-Backups reorganisieren (nur HSMS)

Die Anzeige der HSMS-Aufträge eines BS2000-Systems wird über folgende Einstellungen gesteuert:

- Global über den HSMS-Parameter MONITORING (Anweisung MODIFY-HSMS-PARAMETERS)
- Archivspezifisch über das Archiv-Attribut MONITORING (Anweisung CREATE-ARCHIVE bzw. MODIFY-ARCHIVE-ATTRIBUTES)

In Kombination wirken diese Einstellungen wie folgt:

- Global ist MONITORING=*ALL eingestellt: Das Monitoring ist global für alle Aufträge eingeschaltet, unabhängig von der archivspezifischen Einstellung.
- Global ist MONITORING=*BY-ARCHIVE-ATTRIBUTES eingestellt: Es werden nur Aufträge angezeigt, die sich auf Archive mit der Einstellung MONITORING=*STD beziehen.
- Global ist MONITORING=*SYSHSMS-ONLY eingestellt: Es werden nur Aufträge angezeigt, die sich auf Archive unter der Benutzerkennung SYSHSMS beziehen. Aufträge, die keine Archive betreffen (IMPORT-FILES und EXPORT-FILES), werden nur angezeigt, wenn sie unter einer Benutzerkennung mit dem Privileg HSMS-ADMINISTRATOR gestartet wurden.
- Global ist MONITORING=*NONE eingestellt: Das Monitoring ist global für alle Aufträge ausgeschaltet, unabhängig von der archivspezifischen Einstellung.

Protokoll in PDF-Datei

Am Ende des HSMS-Laufes wird eine Protokolldatei als PDF-Datei unter dem folgenden Namen erzeugt:

```
$SYSHSMS.HSM.C.SYSHSMS.<date_time>.PDF
```

Auf diese Datei kann nur das System zugreifen.

Wenn der Auftrag gelöscht wird, wird auch die PDF-Datei gelöscht.

Wenn der Auftrag mit //RESTART-REQUEST neu gestartet wird, wird die Protokolldatei und damit die PDF-Datei neu erstellt.

Die Erstellung der Protokolldatei im PDF-Format erfolgt ab HSMS V11.0 über das Subsystem CONV2PDF. Wenn ein Fehler auftritt oder das Subsystem nicht verfügbar ist, erstellt HSMS die PDF-Datei wie bisher über die Prozedur \$DHSCFTP aus der Bibliothek SYSPRC.HSSM.<version>. Nur in diesem Fall wird eine temporäre Arbeitsdatei benötigt (siehe auch „Temporäre Dateien“, "[Arbeitsdateien](#)").

5.10.6 Aufträge aufbewahren

HSMS bietet die Möglichkeit, beendete Aufträge nach einem Host-Crash oder Neustart aufrechtzuerhalten. Aufträge mit dem Status COMPLETED werden basierend auf dem Wert von KEEP-REQUESTS vorgehalten. Der Wert von KEEP-REQUEST bestimmt die Anzahl der Tage, an denen ein abgeschlossener Auftrag erhalten bleibt. Ältere abgeschlossene Anfragen werden gelöscht. Der Standardwert ist *STD, d.h. 40 Tage. Der Systemadministrator kann den Wert für die aktuelle HSMS-Session mit MODIFY-HSMS-PARAMETERN temporär ändern. Dies wird jedoch zunächst nicht wirksam, da das Löschen der abgeschlossenen Aufträge nur beim Hochfahren des HSMS-Subsystems erfolgt. Der Wert muss in der Steuerdatei über die Anweisung MODIFY-HSMS-PARAMETERS dauerhaft geändert werden. Die Tabelle unten gibt einen Überblick über mögliche Parameterwerte:

Wert	Funktionalität
	(nur für beendete Aufträge bei jedem Start des Subsystems; zugehörige PDF-Dateien auf dem SE-Server werden auch gelöscht)
*STD (Standardwert) oder 40	Aufträge werden automatisch nach 40 Tagen gelöscht
*NO oder 0	Alle Aufträge werden automatisch gelöscht
*YES	Aufträge werden nie automatisch gelöscht
Integer <0..32767>	Aufträge, die älter als <integer> Tage sind, werden automatisch gelöscht

Der Wert von KEEP-REQUESTS wird vom Systemverwalter festgelegt mittels

```
//MODIFY-HSMS-PARAMETERS OPERATION-CONTROL=*PARAMETERS(KEEP-REQUESTS=...).
```

5.11 Privilegien und Sicherheitsaspekte von HSMS

Dieser Abschnitt beschreibt die Privilegien und die Sicherheitsaspekte von HSMS.

5.11.1 Privilegien

HSMS gewährleistet die Überprüfung der Privilegien eines Benutzers.

Mit HSMS wird nur eine SDF-Systemsyntaxdatei ausgeliefert. Einschränkungen des Anweisungsspektrums können mit dem Softwareprodukt SDF-A vorgenommen werden (siehe Handbuch „SDF-A“ [14]).

HSMS prüft folgende Privilegien:

- **HSMS-Verwalter** (Privileg „HSMS-ADMINISTRATION“)
Dieses Privileg haben nur Benutzeraufträge, die unter einer Benutzerkennung laufen, die das Privileg HSMS-ADMINISTRATION besitzt. Standardmäßig sind dies die Benutzerkennungen SYSHSMS und TSOS.
Der Sicherheitsbeauftragte kann auch anderen Kennungen mit dem Kommando SET-PRIVILEGE dieses Privileg verleihen (siehe Handbuch „SECOS“ [16]). Benutzer, die unter diesen privilegierten Kennungen mit HSMS arbeiten, werden in diesem Handbuch als HSMS-Verwalter bezeichnet.
Es sollten nicht unbedingt gleichzeitig mehrere HSMS-Verwaltungen auf verschiedenen Kennungen eingerichtet werden.

Die für den HSMS-Verwalter vorgesehenen Anweisungen und bestimmte Operanden (z.B. EXPRESS-REQUEST) sind nur mit diesem Privileg zugelassen.

Der HSMS-Verwalter hat unter dem Programm HSMS dieselben Zugriffsrechte auf Dateien wie der Systemverwalter unter TSOS. Außerhalb des Programms HSMS, d.h. im Kommando-Modus auf der Benutzerkennung SYSHSMS, hat der HSMS-Verwalter dagegen keine besonderen Privilegien. Auch bei der Benutzung des Softwareprodukts ARCHIVE gilt er nicht als privilegiert.
- **Subsystem-Verwalter** (Privileg „SUBSYSTEM-MANAGEMENT“)
Dieses Privileg ist obligatorisch, um HSMS starten und beenden zu können.
- **Systemverwalter** (Privileg „TSOS“)
Die HSMS-Servertasks laufen unter diesem Privileg ab. Außerdem dient dieses Privileg dazu, die fernen Dateisysteme einzuhängen, die HSMS bearbeiten soll.
- **Benutzer** („STD-PROCESSING“)
Er besitzt keinerlei Privilegien im obigen Sinn und wird deshalb im Handbuch als „nicht-privilegierter Benutzer“ bezeichnet.

Kennung SYSHSMS

Die Kennung SYSHSMS hat in HSMS eine besondere Bedeutung: sie besitzt standardmäßig das Privileg HSMS-ADMINISTRATION. Auf dieser Kennung werden die zentralen HSMS-Arbeitsdateien angelegt, wie z.B. die Steuer- und die Auftragsdatei. SYSHSMS dient darüber hinaus als Eigentümerkennung für alle Benutzer, die das HSMS-Verwalter-Privileg besitzen (siehe [Abschnitt „Privilegien“](#)).

Mit SYSHSMS als Eigentümerkennung werden ergänzt

- Archivnamen
- Auftragsnamen
- Sicherungsversionsnamen

5.11.2 Datenschutz

Hinsichtlich Dateischutz sind folgende Mechanismen vorgesehen:

- Archivverwaltungsfunktionen wie das Ändern und Ausgeben von Archivattributen, das Löschen eines Archivs sowie der Zugriff auf Sicherungsdateien und Datenträger eines Archivs sind nur dem Archiveigentümer erlaubt. Die Verwaltung bereits eingerichteter Archive ist neben dem HSMS-Verwalter nur Benutzern unter der Archiveigentümer-Benutzerkennung möglich.
- Die Zugriffsrechte von Benutzern auf Archive werden durch die Archivdefinition geregelt. Ein öffentliches, d.h. durch andere nutzbares Archiv, darf jeder Benutzer einrichten.
Das Archivverzeichnis wird unter der Kennung des Archiveigentümers standardmäßig nicht mehrbenutzbar angelegt. Wenn das Archivverzeichnis durch ein Kennwort geschützt wird, muss dieses Kennwort vor einem Zugriff auf das Archiv eingegeben werden (auch vom HSMS-Verwalter).
- Sicherungsdateien legt HSMS standardmäßig unter der Benutzerkennung des Archivverzeichnisses und nicht mehrbenutzbar an (USER-ACCESS=*USER-ONLY). Ein Zugriff auf den Inhalt von Sicherungsdateien unter Umgehung von HSMS ist daher nur dem Archiveigentümer möglich.
- Wenn für die verdrängten Dateien beim Löschen das Überschreiben mit binären Nullen vereinbart ist (DESTROY-BY-DELETE), wird auch bei der Verdrängung der freigegebene Speicherplatz überschrieben. Entsprechend wird beim Löschen einer Sicherungsdatei eines Migrationsarchivs auf S1 verfahren; die Sicherungsdateien werden mit DESTROY-BY-DELETE angelegt.
- Beim Zurückholen oder Kopieren einer Datei durch HSMS bleiben die Schutzattribute der Datei unverändert erhalten.
- Beim automatischen Zurückholen einer verdrängten Datei, zum Beispiel bei einem OPEN, prüft das DVS, ob der Benutzer auf diese Datei überhaupt zugreifen darf.
- Jeder Benutzer – gleichgültig ob privilegiert oder nicht – darf auf die Dateien anderer Benutzerkennungen mit HSMS-Anweisungen zugreifen, wenn es das Datenverwaltungssystem (DVS) erlaubt. Jeder Benutzer kann mit der HSMS-Anweisung EXPORT-FILES fremde, zugreifbare Dateien exportieren (mit Lesezugriff).
- Die Zugriffsberechtigung (evtl. Kennwortschutz) für Dateien der eigenen Benutzerkennung prüft HSMS nur bei Zugriff auf den Dateiinhalt. Dies gilt für das Löschen bei der Archivierung und für das Ersetzen im System vorhandener Dateien beim Restore.

Zugriff auf Dateien fremder Benutzerkennungen

In Aufrufen des HSMS-Verwalters werden Dateien und Jobvariablen fremder Benutzerkennungen vollständig unterstützt.

Bei Aufrufen eines nicht-privilegierten Benutzers gelten folgende Einschränkungen:

- Fremde Dateien und Jobvariablen, für die eine Miteigentümerschaft besteht, werden unterstützt bei den Anweisungen BACKUP-FILES, MIGRATE-FILES, COPY-SAVE-FILE, RESTORE-FILES, SELECT-FILE-NAMES und SELECT-JV-NAMES.
- Das Zurückholen (implizit als auch explizit) fremder, verdrängter Dateien ist erlaubt, wenn Miteigentümerschaft besteht oder die Dateischutzattribute den Zugriff von einer anderen Benutzerkennung erlauben.
- Als Besonderheit werden beim Kopieren mit COPY-SAVE-FILE ohne Dateiselektion in das gleiche Archiv oder in das Schattenarchiv **alle** Dateien und Jobvariablen übernommen, unabhängig von den Zugriffsrechten des Aufrufers für die einzelnen Dateien.
- Bei Aufruf von SHOW-ARCHIVE mit INFORMATION=*FILES werden die Namen fremder Dateien und Jobvariablen nur ausgegeben, wenn Miteigentümerschaft besteht.

- Bei Aufruf von IMPORT- und EXPORT-FILES oder ARCHIVE-FILES werden alle Dateien und Jobvariablen unterstützt, für die der Aufrufer ein DMS-Zugriffsrecht besitzt.

5.11.3 Behandlung der Dateiattribute

Mit Einsatz von SECOS kann GUARDS als Zugriffsschutz-Mechanismus für Dateien, Bibliotheken und Bibliothekselemente sowie Jobvariablen eingesetzt werden. Die Behandlung der Dateiattribute ist im Handbuch „ARCHIVE“ [2] im Abschnitt „Verwaltung von Dateiattributen“ beschrieben. Dort finden Sie auch Informationen zur Guard-Datei und zu GUARDS. HSMS bietet genau dieselbe Unterstützung von GUARDS wie das Softwareprodukt ARCHIVE.

Zum Verständnis dieses Abschnitts müssen Sie wissen, dass die Anweisungsfolge BACKUP-FILES/RESTORE-FILES auf die ARCHIVE-Anweisungen SAVE/RESTORE abgebildet wird und die Anweisungsfolgen ARCHIVE-FILES/RESTORE-FILES und EXPORT-FILES/IMPORT-FILES auf die ARCHIVE-Anweisungen EXPORT/IMPORT abgebildet werden.

Für verdrängte Dateien ist dieser Abschnitt irrelevant, da sich verdrängte Dateien dadurch auszeichnen, dass ihr Katalogeintrag auf der Verarbeitungsebene erhalten bleibt und Änderungen während der Zeit der Migration erlaubt sind. Beim Zurückholen werden lediglich die Einträge zum Datenträger verändert, auf dem die Datei steht, und der Interne Dateiname.

5.11.4 Datensicherheit

Alle HSMS-Anweisungen, die Datenmengen manipulieren, werden auf ARCHIVE abgebildet. Insofern unterstützt HSMS alle Error-Recovery-Maßnahmen von ARCHIVE (siehe Handbuch „ARCHIVE“ [2], Abschnitt „Behandlung von Magnetbändern und Magnetbandkassetten“ und „Fehlerbehandlung bei der Plattenverarbeitung“).

5.11.5 Verschlüsselte Dateien

BS2000 unterstützt auch verschlüsselte Dateien. Diese werden von HSMS einheitlich auf folgende Weise unterstützt:

- Für Sichern und Restaurieren, Export und Import, Migration und Recall einer verschlüsselten Datei ist die Angabe des zugehörigen Crypto-Kennwortes nicht notwendig (gilt für Administrator als auch nicht-privilegierte Benutzer). Die Berücksichtigung dieser Dateien in Prozeduren bzw. Jobs zum Sichern und Restaurieren erfordert deshalb auch keine Änderungen.
- Der Dateiinhalt wird in verschlüsselter Form in der Sicherungsdatei abgelegt (d.h. in unveränderter Form von der Plattenablage im Pubset übernommen). Der Verlust oder das unberechtigte Lesen des Sicherungsdatenträgers führt also nicht zur Lesbarkeit des Dateiinhalts.
- Entsprechend wird beim Restore der verschlüsselte Dateiinhalt aus der Sicherungsdatei unverändert auf Platte zurückgeschrieben.
- Insofern entstehen beim Sichern und Restaurieren von verschlüsselten Dateien keinerlei Performance-Nachteile.
- Die restaurierte Datei besitzt die gleichen Verschlüsselungsmerkmale wie die ursprünglich gesicherte Datei. Insbesondere ist zum Öffnen der Datei das gleiche Crypto-Kennwort erforderlich.

Einschränkung

Eine verschlüsselte Datei kann nicht auf einer Privatplatte oder Net-Storage restauriert werden.

5.12 Einrichten einer HSMS-Konfiguration (Beispiel)

Folgende Schritte sind erforderlich, um eine HSMS-Konfiguration einzurichten:

- Pubsets einer Speicherebene zuordnen
- Systemarchive einrichten

Datenträger können optional einem Systemarchiv zugewiesen werden. Dies ist jedoch nicht empfohlen beim Einsatz von MAREN.

Pubsets einer Speicherebene zuordnen

```
//START-HSMS
//MODIFY-PUBSET-PAR - _____ (1)
//  PUB-ID=2BC,STOR=*S1, -
//  SYSARCHIVE=*STD,SYSBACKUP=*STD
% HSM0003 HSMS STATEMENT COMPLETED
//MODIFY-PUBSET-PAR - _____ (2)
//  PUB-ID=2BY,STOR=*S0(S1-PUB-ID=2BC,SYSMIGRATE=*STD), -
//  SYSARCHIVE=*STD,SYSBACKUP=*STD
% HSM0003 HSMS STATEMENT COMPLETED
//MODIFY-PUBSET-PAR -
//  PUB-ID=BVWC,STOR=*S0(S1-PUB-ID=2BC,SYSMIGRATE=*STD), -
//  SYSARCHIVE=*STD,SYSBACKUP=*STD
% HSM0003 HSMS STATEMENT COMPLETED
//SHOW-PUBSET-PAR _____ (3)
```

```
SHOW-PUBSET-PARAMETERS      PUBSET-ID      = ALL      SYSBACKUP  = ANY
                             STORAGE-LEVEL = ANY      SYSARCHIVE = ANY
                             S1-PUBSET-ID  = ANY      SYSMIGRATE = ANY
-----
PUBSET  ST  SYSBACKUP  SYSARCHIVE  SYSMIGRATE  S1-PUBSET  MIGRATION
BVWC    S0  STD        STD         STD         2BC       ALLOWED
2BC     S1  STD        STD         STD         2BC       ALLOWED
2BY     S0  STD        STD         STD         2BC       ALLOWED
-----
NEXT-PAGE : __  (+, -, ++, --, E)
```

```
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Zuerst wird der Pubset 2BC der Speicherebene S1 zugeordnet. Er kann dann in den folgenden HSMS-Anweisungen S0-Pubsets zugeordnet werden.
- (2) Die Pubsets 2BY und BVWC werden der Speicherebene S0 zugeordnet. Für beide Pubsets wird 2BC als S1-Pubset vereinbart. Beide Pubsets sollen mit den Standard-Systemarchiven arbeiten, die in den globalen HSMS-Parametern vereinbart werden.
- (3) Ausgabe der Parameter der Pubsets unter HSMS-Verwaltung auf dem Bildschirm.

Systemarchive einrichten

```
//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.ARCHIVE, - _____ (1)
// DIR-NAME=$SYSHSMS.HSMS.ARCHIVE.DIR,ALLOWED-USAGE=*ARCHIVAL, -
// TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(1)), -
// USER-ACCES=*ALL-USERS(*WRITE),RETENTION-PERIOD=1
% HSM0003 HSMS STATEMENT COMPLETED
//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.AR.2BY, - _____ (2)
// DIR-NAME=$SYSHSMS.HSMS.ARCHIVE.2BY.DIR,ALLOWED-USAGE= -
// *ARCHIVAL,TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(1)), -
// USER-ACCES=*ALL-USERS(*WRITE),RETENTION-PERIOD=0
% HSM0003 HSMS STATEMENT COMPLETED

//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.ARC.BVWC, -
// DIR=$SYSHSMS.HSMS.ARCHIVE.BVWC.DIR,ALLOWED-USAGE=*ARCHIVAL, -
// TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(1)), -
// USER-ACCES=*ALL-USERS(*WRITE),RETENTION-PERIOD=0
% HSM0003 HSMS STATEMENT COMPLETED
//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.BACKUP, - _____ (3)
// DIR=$SYSHSMS.HSMS.BACKUP.DIR,ALLOWED-USAGE=*BACKUP, -
// TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(3)), -
// USER-ACCES=*ALL-USERS(*READ),RETENTION-PERIOD=0
% HSM0003 HSMS STATEMENT COMPLETED
//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.MI.2BY, - _____ (4)
// DIR=$SYSHSMS.HSMS.MIGRATE.2BY.DIR,ALLOWED-USAGE=*MIGRATION, -
// TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(14)), -
// USER-ACCES=*ALL-USERS(*WRITE),RETENTION-PERIOD=365
% HSM0003 HSMS STATEMENT COMPLETED

//CREATE-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.MI.BVWC, -
// DIR=$SYSHSMS.HSMS.MIGRATE.BVWC.DIR,ALLOWED-USAGE=*MIGRATION, -
// TAPE-CONTROL=*PAR(NEW-STD-S-F=*IN-PERIODS(14)), -
// USER-ACCES=*ALL-USERS(*WRITE),RETENTION-PERIOD=365
% HSM0003 HSMS STATEMENT COMPLETED
//MODIFY-HSMS-PAR DEFAULT-HSMS-STOR=*PAR(SYSARCHIVE= - _____ (5)
// $SYSHSMS.HSMS.ARCHIVE,SYSBACKUP=$SYSHSMS.HSMS.BACKUP, -
// SYSMIGRATE=*UNDEF,S2-DEV-TYPE='TAPE-C4')
% HSM0003 HSMS STATEMENT COMPLETED
//SHOW-HSMS-PAR _____ (6)
```

```

SHOW-HSMS-PARAMETERS (01)                VALID-PERIOD = SESSION
-----
HSMS-ACCOUNT : HSMSACNB                  HSMS-VERSION : 11.0A00
                                           SAVE-FILE-PROC : HSMS-V10-COMP

OPERATION-CONTROL
  OPERATIONAL-MODUS : OPERATION          NUMBER-OF-SUBTASK : 5
  COMMON-MEMORY-SIZE : 3                 HSMS-SV-PORT-NUMBER : 1234
  FILE-SIZE-DEFAULT : 24 ,192           FILE-SIZE-RESULT : 24 ,192
  MONITORING : ALL                       OUTPUT : PRINTER

DEFAULT-HSMS-STORAGE
  S1-PUBSET-ID : SMS1                   S2-DEVICE-TYPE : TAPE-C4
  SYSMIGRATE : NOT-DEFINED              BACKUP-SERVER : ABGSE714
  SYSBACKUP : HSMS.BACKUP               SYSNODEBACKUP : NOT-DEFINED
  SYSARCHIVE : HSMS.ARCHIVE             SYSNODEARCHIVE : NOT-DEFINED

MIGRATION-CONTROL
  EXCEPT-FILE : NONE
  FILE-INHIBIT : RESPECTED
  RECALL-FROM-S2 : ALLOWED              MAXIMUM-WAIT-TIME : -
  CANCEL-AT-RECALL:NOT-ALLOWED         BACKUP-MANDATORY:YES
-----
NEXT-PAGE : __ (+, -, E)

```

```

SHOW-HSMS-PARAMETERS (02)                VALID-PERIOD = SESSION
-----
REQUEST-WAIT-LIMITS
  DIALOG-REQUEST-TIME : 1800            BATCH-REQUEST-TIME : 3600
  DIALOG-EXEC-TIME : 1800              BATCH-EXEC-TIME : 3600

DEFAULT-TAPE-CONTROL
  READ-CONTROL : PROCESS-REQUESTS       START-TIME PERIOD
  WRITE-CONTROL : PROCESS-REQUESTS
  EXPRESS-CONTROL : PROCESS-REQUESTS

REQUEST-PRIORITIES      READ WRITE      READ WRITE
  IMPORT/EXPORT :      128  128      IMPLICIT-RECALL : 128
  BACKUP :          128  128      NODEBACKUP : 128 128
  ARCHIVAL :        128  128      NODEARCHIVAL : 128 128
MIGRATION : 128 128 SHADOW : 128 128
-----
NEXT-PAGE : __ (+, -, E)

```



```
% HSM0003 HSMS STATEMENT COMPLETED

//MODIFY-PUBSET-PAR -
// PUB-ID=2BY,STOR=*S0(SYSMIGRATE=$SYSHSMS.MI.2BY), - _____ (7)
// SYSARCHIVE=$SYSHSMS.HSMS.AR.2BY
% HSM0003 HSMS STATEMENT COMPLETED
//MODIFY-PUBSET-PAR -
// PUB-ID=BVWC,STOR=*S0(SYSMIGRATE=$SYSHSMS.HSMS.MI.BVWC) - _____ (8)
% HSM0003 HSMS STATEMENT COMPLETED
//SHOW-PUBSET-PAR _____ (9)
```

```
SHOW-PUBSET-PARAMETERS      PUBSET-ID      = ALL      SYSBACKUP      = ANY
                             STORAGE-LEVEL = ANY      SYSARCHIVE     = ANY
                             S1-PUBSET-ID  = ANY      SYSMIGRATE     = ANY
-----
PUBSET  ST  SYSBACKUP  SYSARCHIVE  SYSMIGRATE  S1-PUBSET  MIGRATION
BVWC    S0  STD        STD          HSMS.MI.BVWC  2BC        ALLOWED
2BC     S1  STD        STD
2BY     S0  STD        HSMS.MI.2BY HSMS.MI.2BY  2BC        ALLOWED
-----
NEXT-PAGE : __  (+, -, ++, --, E)
```

```
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
```

- (1) Einrichtung des globalen Langzeitarchivs. Die Standard-Sicherungsdatei soll täglich gewechselt werden.
- (2) Einrichtung von pubset-spezifischen Langzeitarchiven; die Archivierung soll pro Pubset vorgenommen werden.
- (3) Einrichtung des globalen Backup-Archivs als öffentliches Archiv mit Leseberechtigung.
- (4) Einrichtung des pubset-spezifischen Versions-Backup-Archivs, Versions-Backup ist immer pubset-basiert.
- (5) Einrichtung zweier pubset-spezifischer Migrationsarchive. Die Migration soll pro Pubset vorgenommen werden, um die Archivverzeichnisse nicht zu groß werden zu lassen und um Konkurrenz beim Zugriff auf die Archive zu minimieren. Das Wechseln der Standard-Sicherungsdatei nur alle 14 Tage gewährleistet, dass bei einem nicht zu hohen Migrationsaufkommen der Datenträger besser ausgenutzt wird.
- (6) Die eingerichteten Archive werden als globale Standardarchive vereinbart; als S2-Standard-Datenträger wird TAPE-C4 festgelegt.

- (7) Die eingestellten Parameter werden auf Bildschirm ausgegeben.
- (8) Dem S0-Pubset 2BY wird pubset-spezifisch ein Migrationsarchiv und ein Langzeitarchiv zugeordnet.
- (9) Dem S0-Pubset BVWC wird pubset-spezifisch ein Migrationsarchiv zugeordnet. Die Archivierung geschieht in das globale Langzeitarchiv.
- (10) Die Parameter der Pubsets unter HSMS-Verwaltung mit den zugeordneten Systemarchiven werden auf Bildschirm ausgegeben.

Datenträger-Pool eines Systemarchivs füllen

Einem Systemarchiv werden Datenträger zugewiesen, die für Schreibvorgänge in dieses Archiv standardmäßig verwendet werden sollen.

```
//MODIFY-ARCHIVE ARCH-NAME=$SYSHSMS.HSMS.MI.2BY, -
// VOL=*ADD(VOL=(HSMS11,HSMS22,HSMS33)) - _____ (1)
% HSM0003 HSMS STATEMENT COMPLETED
//END
% HSM0014 HSMS PROGRAM TERMINATED
Report (Ausgabe auf SYSLST): _____ (2)

*** MODIFY - ARCHIVE                HSMS V11.0        SUMMARY        REPORT    *** 2016-08-12
14:01:02    PAGE    1
    REQUEST-NAME=TSOS    REQUEST-DATE=2016-08-12 14:01:01 USER-ID=TSOS    RUEST-
STATE=COMPLETED WITHOUT ERROR
% HSM0469 MODIFICATION OF ARCHIVE DIRECTORY STARTED FOR 'VOLUMES=ADD()'
% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.160812.140102', VERSION='11.0'
% ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'HSMS11' ADDED TO THE POOL
% ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'HSMS22' ADDED TO THE POOL
% ARC0010 VOLUME OF TYPE 'TAPE-C4' WITH VSN 'HSMS33' ADDED TO THE POOL
***      E N D    O F                HSMS V11.0        SUMMARY        REPORT    *** 2016-08-12 14:
01:02      ***
```

- (1) Drei Magnetbandkassetten werden dem Migrationsarchiv zugeordnet. (Im praktischen Betrieb sind selbstverständlich erheblich mehr Magnetbandkassetten zuzuordnen.)
- (2) HSMS meldet, dass die Magnetbandkassetten dem Pool hinzugefügt wurden. Der Datenträgertyp wurde den globalen Parametern entnommen.

6 Aufruf und Ablauf von HSMS

Dieses Kapitel beschreibt

- das Laden und Entladen von HSMS durch den Systemverwalter oder Operator.
- den Aufruf von HSMS durch den Benutzer oder aus Programmen heraus.
- die Arbeitsdateien, die (bis auf die zentralen Arbeitsdateien auf der Kennung SYSHSMS) während eines HSMS-Laufs auf der Benutzerkennung des Aufrufers angelegt werden.
- Maßnahmen zur Verbesserung der Performance (Reduzierung der Sicherungszeiten)

6.1 Laden und Entladen von HSMS

HSMS besteht aus einem privilegiert ablaufenden Teil (Task Privileged, TPR) und einem nicht-privilegiert ablaufenden Teil (Task Unprivileged, TU): den TPR-Modulen und dem TU-Lademodul.

Der privilegierte Teil ist als Subsystem realisiert. Dieses Subsystem HSMS wird in den Systemadressraum (Speicherklasse 4) geladen.

Das Laden von HSMS für eine BS2000-Session muss durch einen Benutzer erfolgen, der das Privileg „SUBSYSTEM-MANAGEMENT“ besitzt. Nach dem Laden steht HSMS allen Benutzern zur Verfügung. Der Aufruf von HSMS durch den einzelnen Benutzer geschieht über ein Lademodul.

Der Zeitraum vom Laden des Subsystems HSMS bis zum Entladen wird in diesem Handbuch als HSMS-Session bezeichnet.

Für das Laden von HSMS gibt es zwei Möglichkeiten:

Ein Benutzer mit dem Privileg „SUBSYSTEM-MANAGEMENT“ kann HSMS mit folgendem DSSM-Kommando laden:

```
/START-SUBSYSTEM SUBSYSTEM-NAME=HSMS
```

Entladen wird HSMS mit diesem DSSM-Kommando:

```
/STOP-SUBSYSTEM SUBSYSTEM-NAME=HSMS
```

Wenn HSMS mit DSSM-Kommandos geladen wird, können keine Steuerparameter an HSMS übergeben werden. Mit den DSSM-Kommandos kann HSMS direkt in einer CMDFILE beim BS2000-Startup geladen werden.

Ein Benutzer mit dem Privileg „SUBSYSTEM-MANAGEMENT“ kann HSMS nach dem Aufruf des TU-Programms HSMS (siehe [Abschnitt „Aufruf von HSMS“](#)) mit folgender Anweisung laden:

```
//START-HSMS
```

Mit dieser HSMS-Anweisung können Steuerparameter für den HSMS-Lauf übergeben werden (siehe Handbuch „HSMS Bd. 2“ [\[1\]](#), Anweisung START-HSMS).

Entladen wird HSMS mit der Anweisung:

```
//STOP-HSMS
```

Mit dieser HSMS-Anweisung können zusätzliche Parameter für das Entladen angegeben werden (siehe Handbuch „HSMS Bd. 2“ [\[1\]](#), Anweisung STOP-HSMS).



HSMS-SV gehört ab HSMS V9.0B nicht mehr zum Lieferumfang. Damit unterstützt HSMS ab V9.0B nur Workstations, die über NFS mit dem lokalen BS2000-UFS verbunden sind.

HSMS versucht, das Subsystem ARCHIVE zu starten. Wenn dieses Subsystem nicht gestartet werden kann, wird auch das Starten von HSMS zurückgewiesen.

Parameter, die während einer HSMS-Session temporär verändert wurden, sind auch nach einem /HOLD-SUBSYSTEM bzw. //STOP-HSMS SUBSYSTEM=HOLD und erneutem Laden verloren.

Beim Laden von HSMS werden unter anderem die BCAM-Hostnamen festgelegt, die für die HSMS-Session gelten. Wenn nicht nur auf dem lokalen Rechner gearbeitet werden soll, darf HSMS deshalb erst geladen werden, nachdem BCAM gestartet wurde.

6.2 Aufruf von HSMS

- Starten und Beenden durch den Benutzer
- Kommandoprozessor SDF
- Dialog- und Stapelbetrieb
- Fehlerbehandlung

6.2.1 Starten und Beenden durch den Benutzer

Der Benutzer ruft HSMS auf mit dem Kommando START-HSMS auf:

START-HSMS	Kurzname: HSMS
VERSION = *STD ,MONJV = *NONE / <filename 1..54 without-gen-vers> ,CPU-LIMIT = *JOB-REST / <integer 1..32767 seconds>	

VERSION = *STD

Die mit dem Kommando SELECT-PRODUCT-VERSION eingestellte Version wird geladen. Wurde keine Version eingestellt, so wird die höchste verfügbare Version von HSMS geladen.

MONJV = *NONE / <filename 1..54 without-gen-vers>

Ermöglicht die Angabe einer Monitor-Jobvariablen zur Überwachung des Programmlaufs.

CPU-LIMIT = *JOB-REST

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf. Voreingestellt ist *JOB-REST, d. h. die verbleibende CPU-Zeit wird für die Task verwendet.

CPU-LIMIT = <integer 1..32767 seconds >

Es soll maximal die angegebene Zeit verwendet werden.

Alternativ kann HSMS auch mit dem Kommando START-EXECUTABLE-PROGRAM gestartet werden, vorausgesetzt der Benutzer darf dieses Kommando ausführen:

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

Aufgerufen wird dabei das Programm HSMS auf der System-Standardkennung.

Beendet wird HSMS durch Eingabe der Anweisung:

```
//END
```

Der Zeitraum vom Aufruf von HSMS bis zur Beendigung mit END wird in diesem Handbuch HSMS-Lauf genannt.

HSMS kann synchron und asynchron aufgerufen werden (siehe [Abschnitt „Asynchrone und synchrone Verarbeitung“](#)).

6.2.2 Kommandoprozessor SDF

Mit dem Kommandoprozessor SDF (System Dialog Facility) kann der Benutzer Anweisungen geführt menügesteuert eingeben. Das Maß der Führung kann mit dem Operanden GUIDANCE des Kommandos bzw. der Standard-Anweisung MODIFY-SDF-OPTIONS festgelegt werden.

Der geführte Dialog wird mit GUIDANCE = *MINIMUM, *MEDIUM oder *MAXIMUM eingestellt, wobei die einzelnen Stufen sich durch den Umfang zusätzlicher Hilfetexte unterscheiden. Unabhängig von der Einstellung kann sich der Benutzer durch Eingabe von „?“ an Stelle eines Operandenwertes zusätzliche Erläuterungen ausgeben lassen.

Der geführte Dialog wird durch die Ausgabe von Menüs gesteuert, die die möglichen Anweisungen bzw. Operanden enthalten. Die Standardwerte der Operanden sind dabei eingesetzt. Wenn kein Standardwert eingesetzt ist, ist eine Angabe zwingend.

Wenn sich hinter einem Operandenwert ein Klammersymbol „()“ befindet, bedeutet dies, dass der Operandenwert eine Struktur einleitet, für die (durch Angabe dieses Operandenwertes) ein Unterfragebogen angefordert werden kann.

Es ist jederzeit möglich, aus dem ungeführten Dialog (GUIDANCE = *NO oder *EXPERT) temporär in den geführten Dialog zu wechseln. Dazu wird ein „?“ statt oder hinter einer Anweisung eingegeben. Auch hinter einem Operanden, der mit einem Komma abgeschlossen ist, kann durch ein „?“ in ein Menü gewechselt werden.

Die Dialog-Eingabe von Kommandos und Anweisungen ist in den Handbüchern „Dialogschnittstelle SDF“ [5] und „Kommandos“ [8] ausführlich beschrieben.

6.2.3 Dialog- und Stapelbetrieb

HSMS kann sowohl im Dialog- als auch im Stapelbetrieb ablaufen.

HSMS erwartet alle Anweisungen aus der Systemdatei SYSDTA. SYSDTA ist im Dialogbetrieb der Datenstation zugewiesen, im Stapelbetrieb der ENTER-Datei. Wenn HSMS in einer Prozedur aufgerufen werden soll, muss die Systemdatei zugewiesen werden mit:

```
/ASSIGN-SYSDTA TO=*SYSCMD
```

6.2.4 Fehlerbehandlung

Bevor ein Benutzertask nach dem Absetzen einer HSMS-Anweisung die Kontrolle zurückerhält, setzt HSMS Informationen, über die im Stapelbetrieb und in Prozeduren auf Fehlersituationen reagiert werden kann:

- Bei Warnungen wird Auftragsschalter 30 gesetzt.
Beispiele:
 - Eine für die Migration angegebene Datei ist nicht migrierbar.
 - Eine zu restaurierende Datei ist bereits vorhanden.
- Bei einfachen Fehlern wird Auftragsschalter 31 gesetzt.

Beispiel:

Eine angegebene Datei kann nicht bearbeitet werden wegen Fehler bei Ein- oder Ausgabe.

- Bei schweren Fehlern wird Spin-off ausgelöst.
Beispiele:
 - Ungültige Anweisung wurde eingegeben.
 - Maximale Wartezeit wurde überschritten.
 - Datenträger für die Verarbeitung fehlt.

Die Stellung der Auftragsschalter kann im Kommando SHOW-JOB-SWITCHES abgefragt werden. Mit dem Kommando SKIP-COMMANDS kann abhängig von der Stellung eines oder mehrerer Auftragsschalter zu einem Sprungziel verzweigt werden.

Spin-off bewirkt Überspringen aller folgenden Kommandos bis zum nächsten SET-JOB-STEP oder EXIT-JOB bzw. bis zum Ende der Prozedur- oder ENTER-Datei.

Bei asynchroner Verarbeitung führen vom HSMS-Servertask erkannte Fehlersituationen weder zum Setzen eines Auftragsschalters noch zu Spin-off.

Überwachung mit Jobvariablen

Für den HSMS-Lauf kann im Kommando START-HSMS eine Monitor-Jobvariable angegeben werden, die insgesamt den Programmlauf überwacht (siehe Handbuch „Jobvariablen“ [23]).

HSMS selbst unterstützt für seine Aufrufe keine Monitor-Jobvariablen. Bei den einzelnen Aufrufen kann jedoch eine Jobvariable angegeben werden, in die HSMS Informationen über den Status dieses Aufrufs ablegt (siehe [Abschnitt „Jobvariable zur Auftragsüberwachung“](#)).

6.3 Aufruf von HSMS aus Programmen

HSMS-Anweisungen können über den HSMS-Makro auch aus Programmen gegeben werden. Im Makro wird eine HSMS-Anweisung in einer freien Form übergeben, wie sie auch dem Programm gegeben werden kann („Free-String-Format“).

Die Ausgabe beschränkt sich auf den Returncode, der im Standard-Header zurückgegeben wird (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [9]).

Der Makro zerstört die Register 1, 14, und 15.

Wenn der Makro nicht in die System-Makrobibliothek eingemischt wurde, muss die SYSLIB.HSMS.120 (siehe [Abschnitt „Lieferumfang von HSMS“](#)) dem Assembler als ALTLIB zugewiesen werden.

Dieselbe Bibliothek muss beim Binden bzw. vor dem Aufruf des Bindemoduls als TASKLIB zugewiesen werden.

Operation	Operanden
HSMS	MF = S L C D E [,LENGTH = <number>] [,ADDR = { <adress> <(reg)> }] [,PREFIX = <prefix>] [,PARAM = { <adress> <(reg)> }]

MF Eine Operandenliste wird erzeugt, siehe Handbuch „Makroaufrufe an den Ablaufteil“ [9].

= S Standardform; Daten und Befehlscode sind nicht getrennt.

= L Es werden nur Daten erzeugt.

= C Eine CSECT wird erzeugt.

= D Eine DSECT wird erzeugt.

= E Es wird nur Befehlscode erzeugt.

LENGTH bestimmt die Länge des Speicherbereichs, der für die Anweisungseingabe zu reservieren ist. Er muss bei MF=S,L angegeben werden; ansonsten wird er ignoriert. Die maximale Länge beträgt 16372 Byte.

ADDR bestimmt die Adresse des Speicherbereichs für die Anweisungseingabe. Er muss bei MF=S,L angegeben werden; ansonsten wird er ignoriert.

PREFIX bestimmt das erste Zeichen in der HSMS-Parameterliste. Der Operand wird bei MF=E,S ignoriert.

PARAM steuert die Adressierung der Parameterliste. Bei MF=E muss er angegeben werden, ansonsten wird er ignoriert.

Standard ist „(1)“, d.h. die Adresse der Parameterliste wird in Register 1 erwartet.

Die Parameterliste besteht aus dem Standard-Header, gefolgt von der Länge der HSMS-Anweisung und der Adresse eines Feldes, das die HSMS-Anweisung enthält.

Wenn die Parameterliste dynamisch geändert wird, ist der Benutzer dafür verantwortlich, dass die richtigen Felder mit den richtigen Werten versehen werden.

Returncodes und Fehlerklassen

Übergeben werden im Standardheader ein Maincode (MC) sowie zwei Subcodes, Subcode 1 (SC1) und Subcode 2 (SC2), die in ihrer Kombination eine genaue Information über den Verlauf der mit dem Makro angestoßenen Aktion geben.

Die Returncodes werden in der Reihenfolge SC1, SC2, MC (4 Byte) übergeben.

MC	SC1	SC2	Bedeutung
X'0000'	X'00'	X'00'	Die Anweisung wurde ohne Fehler ausgeführt.
		X'01'	Die Anweisung wurde ausgeführt; eine Aktion war nicht erforderlich.
		X'02'	Die Anweisung wurde auf Gültigkeit geprüft und für die asynchrone Verarbeitung angenommen.
X'0001'	X'00'		Die Anweisung wurde mit Warnungen ausgeführt (Warnmeldungen werden ausgegeben, wenn ein vom Benutzer angegebenes Objekt nicht verarbeitet werden kann oder darf: Die Datei ist nicht migrierbar, die zu restaurierende Datei ist bereits vorhanden ...).
X'0002'	X'00'		Die Anweisung wurde mit Fehlern ausgeführt (Fehlermeldungen werden ausgegeben, wenn ein vom Benutzer angegebenes Objekt wegen eines Fehlers nicht oder nur unvollständig verarbeitet werden kann: Die Datei ist gesperrt, Ein- oder Ausgabefehler ...).
X'0003'	X'00'		Die Anweisung wurde nicht ausgeführt; beim Warten auf die Auftragsbeendigung wurde die in der HSMS-Steuerdatei festgelegte, maximal zulässige Wartezeit überschritten.
X'0004'	X'01'		Die Anweisung wurde wegen eines Syntaxfehlers zurückgewiesen.
	X'20'		Die Anweisung wurde wegen eines System- bzw. HSMS-internen Fehlers zurückgewiesen.
	X'40'		Die Anweisung wurde wegen sonstiger Fehler, z.B. Privilegienfehler, zurückgewiesen.
X'0005'			Die Ausführung der Anweisung wurde abgebrochen, weil auf Grund einer Fehlersituation eine Fortsetzung nicht sinnvoll war, z.B. wegen des Fehlens von Betriebsmitteln. Die Anweisung kann in der Regel mit RESTART-REQUESTS fortgesetzt werden.
X'FFFF'	X'01'		HSMS ist im System nicht bekannt.
	X'02'		HSMS ist nicht verfügbar, z.B. weil die Syntaxdatei fehlt.
	X'03'		Versionsfehler, z.B. ungültige Version der Syntaxdatei.
	X'..''		Die weiteren Subcodes entsprechen dem Standardheader.

Beispiel

Das folgende Programm ermöglicht dem Benutzer, HSMS-Anweisungen einzugeben; es ruft HSMS über die Programmschnittstelle auf.

```

ASSEMBH
LISTING
09:05:55 2016-02-29 PAGE 0003
LOCTN OBJECT CODE ADDR1 ADDR2 STMT M SOURCE STATEMENT
000000 1 HSMSMAC CSECT ,
2 HSMSMAC AMODE ANY
3 HSMSMAC RMODE ANY
4 GPARMOD 31
5 1 * ,MACRO: GPARMOD,

VERSION: VER121
000000 0D A0 6 BASR 10,0
000002 00000002 7 USING *,10
000002 41 70 A096 00000098 8 LA 7,LHSMS
000006 00000000 9 USING HSMSREF,7
10 PRINT GEN
11 WHAT RDATA MF=(E,LRDATA)
000006 41 10 A0A6 000000A8 12 1 WHAT LA 1,LRDATA
LOAD ADDR PARAM LIST INTO R1
00000A 0A 27 13 1 SVC 39
SYSFILE SVC
14 1 *
00000C D5 03 A0EAA1F6 000000EC 000001F8 15 CLC STMT(4),=C'END '
000012 47 80 A05C 0000005E 16 BE TERM
17 CALL HSMS MF=E,PARAM=LHSMS
18 1 CALL MFCHK
MF=E, C
18 1 SUPPORT=(C,D,E,L,
S), C
18 1
PREFIX=D, C
18 1
MACID=HSM, C
18 1
DMACID=HSM, C
18 1
DNAME=HSMPAR, C
18 1
PARAM=LHSMS, C
18 1
ENTRY=IDHSASS, C
18 1 ALIGN=F
000016 19 2 CALL DS 0Y
000016 58 F0 A1FA 000001FC 20 2 L 15,=V(IDHSASS)
00001A 41 10 A096 00000098 21 2 LA 1,LHSMS
00001E 0D EF 22 2 BASR 14,15
000020 D2 03 A1BE7004 000001C0 00000004 23 MVC MPACK,DHSMRET
000026 F3 84 A1B2A1BE 000001B4 000001C0 24 UNPK MRS(9),MPACK(5)
00002C D4 07 A1B2A1D3 000001B4 000001D5 25 NC MRS(8),ANDMASK
000032 DC 07 A1B2A1C3 000001B4 000001C5 26 TR MRS(8),MTAB
000038 D2 03 A1E7A1B6 000001E9 000001B8 27 MVC RCMC,MRS+4
00003E D2 01 A1ECA1B4 000001EE 000001B6 28 MVC RCSUB1,MRS+2

```

```

000044 D2 01 A1EFA1B2 000001F1 000001B4      29          MVC   RCSUB2,MRS
                                           30   WRITERC  WROUT MF=(E,LWROUT)
00004A 41 10 A0CA      000000CC      31   1 WRITERC  LA    1,LWROUT
LOAD ADDR PARAM LIST INTO R1
00004E 0A 27          32   1          SVC   39
SYSFILE SVC
                                           33   1 *
000050 92 40 A0EA      000000EC      34          MVI   STMT,C' '
000054 D2 C6 A0EBA0EA 000000ED 000000EC      35          MVC   STMT+1(L'STMT-1),STMT
00005A 47 F0 A004      00000006      36          B    WHAT
                                           37   TERM    TERM
00005E          38   1 TERM    DS
0H          206
00005E 41 10 A066      00000068      39   1          LA    1,
S0006D          205
000062 47 F0 A076      00000078      40   1          B
S0006S          200
000068          41   1 S0006D  DS
0F          200
VERS=1          207          42   1          FHDR  UNIT=6,FUNCT=40,
000068          43   2          DS    0A
000068          44   2          DS    0XL8
GENERAL OPERAND LIST HEADER
000068 0006          45   2          DC    AL2(6)
FUNCTION UNIT NUMBER
00006A 28          46   2          DC    AL1(40)
FUNCTION NUMBER
00006B 01          47   2          DC    AL1(1)
FUNCTION INTERFACE VERSION NUMBER
00006C FFFFFFFF      48   2          DC    X'FFFFFFFF'
Returncode is virgin
000070 01          49   1          DC
XL1'01'          207
000071 00          50   1          DC    XL1'00'
000072 00          51   1          DC    XL1'00'
000073 04          52   1          DC    XL1'04'
000074 40404040      53   1          DC    CL4' '
000078          54   1 S0006S  DS
0Y          200
000078 0A 09          55   1          SVC   9
                                           56   TERMD   TERM  DUMP=Y,MODE=A

ASSEMBH
LISTING
09:05:55 2016-02-29 PAGE 0004
LOCTN  OBJECT CODE  ADDR1  ADDR2  STMT  M  SOURCE STATEMENT

00007A          57   1 TERMD   DS
0H          206
00007A 41 10 A082      00000084      58   1          LA    1,
S0008D          205
00007E 47 F0 A092      00000094      59   1          B
S0008S          200
000084          60   1 S0008D  DS
0F          200
VERS=1          207          61   1          FHDR  UNIT=6,FUNCT=40,
000084          62   2          DS    0A
000084          63   2          DS    0XL8

```

GENERAL OPERAND LIST HEADER					
000084 0006	64	2	DC	AL2(6)	
FUNCTION UNIT NUMBER					
000086 28	65	2	DC	AL1(40)	
FUNCTION NUMBER					
000087 01	66	2	DC	AL1(1)	
FUNCTION INTERFACE VERSION NUMBER					
000088 FFFFFFFF	67	2	DC	X'FFFFFFFF'	
Returncode is virgin					
00008C 01	68	1	DC		
XL1'01'		207			
00008D 01	69	1	DC	XL1'01'	
00008E 04	70	1	DC	XL1'04'	
00008F 04	71	1	DC	XL1'04'	
000090 40404040	72	1	DC	CL4'	
000094	73	1	S0008S	DS	
0Y		200			
000094 0A 09	74	1	SVC	9	
	75	*			
	76	LHSMS	HSMS	ADDR=STMT,LENGTH=200,MF=L	
	77	1 LHSMS	MFCHK		
MF=L,		C			
S),	C	77	1	SUPPORT=(C,D,E,L,	
PREFIX=D,		77	1		C
MACID=HSM,		77	1		C
DMACID=HSM,		77	1		C
DNAME=HSMPAR,		77	1		C
PARAM=,		77	1		C
ENTRY=IDHSASS,		77	1		C
ALIGN=F					
000098	78	2	LHSMS	DS	0F
000098	79	1	DHSMST	DS	
0H					
000098	80	1	HSM20010	DS	
0H					
	81	1		FHDR	MF=L,UNIT=73,FUNCT=1,
VERS=1					
000098	82	2	DS	0A	
000098	83	2	DS	0XL8	
GENERAL OPERAND LIST HEADER					
000098 0049	84	2	DC	AL2(73)	
FUNCTION UNIT NUMBER					
00009A 01	85	2	DC	AL1(1)	
FUNCTION NUMBER					
00009B 01	86	2	DC	AL1(1)	
FUNCTION INTERFACE VERSION NUMBER					
00009C FFFFFFFF	87	2	DC	X'FFFFFFFF'	
Returncode is virgin					
0000A0 00C8	88	1	DC	Y(200)	LENGTH OF
STATEMENT					
0000A2 00	89	1	DC	X'0'	

UNUSED								
0000A3 FF			90	1	DC	AL1(255)	REGISTER	
NUMBER								
0000A4 000000EC			91	1	DC	A(STMT)	ADDRESS OF	
STATEMENT								
	00000010		92	1	DHSMLE	EQU	*-	
DHSMLE								
			93		*			
			94		LRDATA	RDATA	MSGOUT,TERMD,STMT#,	
KEYOUT=Y,MF=L								
0000A8			95	1	S0013D	DS		
0F						A340		
			96	1	LRDATA	FHDR	UNIT=36,FUNCT=18,VERS=2	
0000A8			97	2		DS	0A	
0000A8			98	2	LRDATA	DS	0XL8	
GENERAL OPERAND LIST HEADER								
0000A8 0024			99	2		DC	AL2(36)	
FUNCTION UNIT NUMBER								
0000AA 12			100	2		DC	AL1(18)	
FUNCTION NUMBER								
0000AB 02			101	2		DC	AL1(2)	
FUNCTION INTERFACE VERSION NUMBER								
0000AC FFFFFFFF			102	2		DC	X'FFFFFFFF'	
Returncode is virgin								
			103	1	*			
0000B0 0000007A			104	1		DC	A(TERMD)	
ERROR ADDRESS								
0000B4 000000E8			105	1		DC	AL4(MSGOUT)	
READ IN AREA ADDRESS								
0000B8			106	1		DS	AL1(0)	
PLACE FOR I.EDIT BYTE 1								
0000B9			107	1		DS	AL1(0)	
PLACE FOR I.EDIT BYTE 2								
0000BA 00			108	1		DC	AL1(0)	
SYSDTA ASSIGNMENT								
0000BB 00			109	1		DC	AL1(0)	
FLAG BYTE 1								
0000BC 00CC			110	1		DC	AL2(STMT#)	
LENGTH OF READ								
			111	1	*			
0000BE 80			112	1		DC	AL1(128)	
FLAG TABLE BYTE								
0000BF 00			113	1		DC	AL1(0)	
ASSIGNMENT CHANGE INDICATOR								
0000C0 0000			114	1		DC	H'0'	
KEY-POSITION								
0000C2 0000			115	1		DC	H'0'	
KEY-LENGTH								
0000C4 00000000			116	1		DC	AL4(0)	
VTSUCB ADDRESS								
0000C8 0000			117	1		DC	AL2(0)	
INPUT TIMER VALUE	009							
ASSEMBH								
LISTING								
09:05:55	2012-02-29	PAGE 0005						
LOCTN	OBJECT CODE	ADDR1	ADDR2	STMNT	M	SOURCE	STATEMENT	
0000CA	0000			118	1		DC	H'0'
RES_FOR_TIAM		007						
				119	1	*		

ICFD=,	C	120	1	@DCEI	DCEDIT=,MODE=,IGETFC=,
IGETBS=,	C	120	1		ITRSUP=,ILINEND=,
IHDR=,	C	120	1		IMANUAL=,ILCASE=,
0000CC		121	2	ORG	*-20
0000B8 00		122	2	DC	AL1(0)
0000B9	000000CC	123	2	ORG	*+20-1
0000CC	000000B9	124	2	ORG	*-19
0000B9 00		125	2	DC	AL1(0)
0000BA	000000CC	126	2	ORG	*+19-1
921011	53531002	127	2		*,@DCEI 999
		128	1	*	
		129	1	*	
0000CC		130	1	LWROUT	WROUT RC,TERMD,MF=L
0F		131	1	S0016D	DS
				A340	
0000CC		132	1	LWROUT	FHDR UNIT=36,FUNCT=17,VERS=2
0000CC		133	2	DS	0A
0000CC		134	2	LWROUT	DS 0XL8
GENERAL OPERAND LIST HEADER					
0000CC 0024		135	2	DC	AL2(36)
FUNCTION UNIT NUMBER					
0000CE 11		136	2	DC	AL1(17)
FUNCTION NUMBER					
0000CF 02		137	2	DC	AL1(2)
FUNCTION INTERFACE VERSION NUMBER					
0000D0 FFFFFFFF		138	2	DC	X'FFFFFFFF'
Returncode is virgin					
		139	1	*	
0000D4 0000007A		140	1	DC	AL4(TERMD)
ERROR ADDRESS					
0000D8 000001E0		141	1	DC	AL4(RC)
MESSAGE AREA ADDRESS					
0000DC		142	1	DS	AL1(0)
PLACE FOR EDIT BYTE 1					
0000DD		143	1	DS	AL1(0)
PLACE FOR EDIT BYTE 2					
0000DE 00		144	1	DC	AL1(0)
RESERVED					
0000DF 00		145	1	DC	AL1(0)
FLAG BYTE 1					
0000E0 00000000		146	1	DC	AL4(0)
VTSUCB ADDRESS					
0000E4 00000000		147	1	DC	F'0'
RES_FOR_TIAM					
		148	1	*	
OEXTEND=,	C	149	1	@DCEO	DCEDIT=,MODE=,
OMANUAL=,	C	149	1		OTRSUP=,OLINEND=,
ONOPSN=,	C	149	1		OPTAPE=,OHCOPY=,
OINFO=,	C	149	1		OHDR=,OETB=,OHOM=,
		149	1		OBELL=,OTRANS=,

```

ONOLOGC=,
C
149 1 RDA1=-12,RDA2=-11
0000E8 000000DC 150 2 ORG *-12
0000DC 00 151 2 DC AL1(0)
0000DD 000000E8 152 2 ORG *+12-1
0000E8 000000DD 153 2 ORG *-11
0000DD 00 154 2 DC AL1(0)
0000DE 000000E8 155 2 ORG *+11-1
156 2 *,@DCEO 999
921011 53531004
157 1 *
158 *
0000E8 159 MSGOUT DS 0F
0000E8 00CC 160 STMT_L DC Y(STMT#)
0000EA 0000 161 STMT_R DC X'0000'
0000EC 4040404040404040 162 STMT DC CL200' '
000000CC 163 STMT# EQU *-STMT_L
164
*
0001B4 4040404040404040 165 MRS DC CL9' '
0001C0 166 MPACK DS F
0001C4 00 167 DC X'00'
0001C5 F0F1F2F3F4F5F6F7 168 MTAB DC C'0123456789ABCDEF'
0001D5 0F0F0F0F0F0F0F0F 169 ANDMASK DC X'0F0F0F0F0F0F0F0F'
0001E0 170 RC DS 0F
0001E0 0013 171 DC Y(RC#)
0001E2 0000 172 DC X'0000'
0001E4 00 173 DC X'00'
0001E5 D9C37A40 174 DC C'RC: '
0001E9 175 RCMC DS CL4
0001ED 40 176 DC C' '
0001EE 177 RCSUB1 DS CL2
0001F0 40 178 DC C' '
0001F1 179 RCSUB2 DS CL2
00000013 180 RC# EQU *-RC
ASSEMBH
LISTING
09:05:55 2012-02-29 PAGE 0006
LOCTN OBJECT CODE ADDR1 ADDR2 STMNT M SOURCE STATEMENT
181 HSMSREF HSMS MF=D
182 1 HSMSREF MFCHK
MF=D,
C
182 1 SUPPORT=(C,D,E,L,
S),
C
182 1
PREFIX=D,
C
182 1
MACID=HSM,
C
182 1
DMACID=HSM,
C
182 1
DNAME=HSMPAR,
C
182 1
PARAM=,
C
182 1
ENTRY=IDHSASS,
C
182 1
ALIGN=F
000000 183 2 HSMSREF DSECT ,

```


SPECIFIC REACTION POSSIBLE	218	2 * CLASS E	X'80' - X'82'	WAIT AND
RETRY	219	2 *		
00000000	220	2 DHSMRFSP EQU	X'00'	
FUNCTION SUCCESSFULLY PROCESSED	221	2 DHSMRPER EQU	X'01'	
PARAMETER SYNTAX ERROR	222	2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN		
CLASS X'01' - X'1F'	223	2 DHSMRFNS EQU	X'01'	
00000001	224	2 DHSMRFNA EQU	X'02'	
CALLED FUNCTION NOT SUPPORTED	225	2 DHSMRVNA EQU	X'03'	
00000002	226	2 *		
CALLED FUNCTION NOT AVAILABLE	227	2 DHSMRAER EQU	X'04'	
00000003	228	2 DHSMRIER EQU	X'20'	
INTERFACE VERSION NOT SUPPORTED	229	2 DHSMRCAR EQU	X'40'	
00000004	230	2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN		
ALIGNMENT ERROR	231	2 DHSMRECR EQU	X'41'	
00000020	232	2 *		
INTERNAL ERROR	233	2 DHSMRECN EQU	X'42'	SS
00000040	234	2 *		
CORRECT AND RETRY	235	2 DHSMRWAR EQU	X'80'	WAIT
CLASS X'40' - X'7F'	236	2 DHSMRWLR EQU	X'81'	
00000041	237	2 DHSMRWUR EQU	X'82'	WAIT
SUBSYSTEM (SS) MUST BE CREATED	238	2 *		BUT
EXPLICITELY BY CREATE-SS	239	2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN		
00000042	240	2 DHSMRTNA EQU	X'81'	SS
MUST BE EXPLICITELY CONNECTED	241	2 DHSMRDH EQU	X'82'	SS IN
00000080	242	2 *		
FOR A SHORT TIME AND RETRY				
00000081				
" LONG "				
00000082				
TIME IS UNCALCULABLY LONG				
RETRY IS POSSIBLE				
CLASS X'80' - X'82'				
00000081				
TEMPORARILY NOT AVAILABLE				
00000082				
DELETE / HOLD				
ASSEMBH				
LISTING				
09:05:55 2012-02-29 PAGE 0007				
LOCTN OBJECT CODE ADDR1 ADDR2	STMNT	M	SOURCE STATEMENT	
000006	243	2 DHSMRET DS	0AL2	6 MAIN
RETURN CODE	244	2 DHSMR2 DS	AL1	6 MAIN
000006	245	2 DHSMR1 DS	AL1	7 MAIN
RETURN CODE 2				
000007				
RETURN CODE 1	246	2 *		
	247	2 * SPECIAL LAYOUT OF		

```

LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
                                248  2  *
                                0000FFFF 249  2  DHSMLNK EQU  X'FFFF'
LINKAGE ERROR / REQ. NOT PROCESSED
                                00000008 250  2  DHSMFHL EQU  8                8
GENERAL OPERAND LIST HEADER LENGTH
                                251  2  *
                                00000008 252  1  DHSMLIST EQU
*
000008 253  1  DHSMLGTH DS  H                LENGTH OF
STATEMENT STRING | |
00000A 254  1  DS  X
UNUSED | |
00000B 255  1  DHSMREG DS  AL1            REGISTER #
CONTAINING STRING ADDRESS | |
00000C 256  1  DHSMADDR DS  F                ADDRESS OF
STATEMENT STRING | |
DHSMLIST 00000008 257  1  DHSMLENG EQU  *-
                                | |
                                258  END ,  *** ,NO END CARD
0001F8 C5D5C440 259  =C'END '
0001FC 00000000 260  =V(IDHSASS)
000200 0309290905446314 261  =X'0309290905446314'
CONSISTENCY CONSTANT FOR AID
FLAGS IN 00000 STATEMENTS, 000 PRIVILEGED FLAGS, 000 MNOTES
HIGHEST ERROR-WEIGHT : NOTE
THIS PROGRAM WAS ASSEMBLED BY ASSEMBH V01.2C00 ON 2012-02-29 AT 09:05:44
ASSEMBH
LISTING
09:05:55 2012-02-29 PAGE 0008
USED FILES AND LIBRARIES
SOURCE FILE : :2OSG:$USER1.ALF.SRC.HSMSMAC
MACRO-LIBRARIES LINKNAME LIBRARY-NAME
                :2OSG:$USER1.ALFX2.SYSLIB.HSMS.090
                :2RZV:$RZV.SM.SSG.BS2CP.V18.0.GCLIB.UR
ASSEMBLY TIME : 0.672 SEC.
THIS LISTING WAS GENERATED BY THE LISTING GENERATOR V 1.2C00.

```

Der HSMS-Makro wird in der Form MF=L verwendet. Die Adresse der Anweisung ist angegeben; ihre maximale Länge beträgt 200 Bytes.

```

/START-BINDER _____ (1)
//START-LLM-CREATION INTERNAL-NAME=HSMS-MACRO-EXEC
//INCLUDE-MODULES MODULE-CONTAINER=*OMF(ELEMENT=*ALL)
//INCLUDE-MODULES MODULE-CONTAINER=*LIBRARY-ELEMENT -
// (LIBRARY=$SYSHSMS.SYSLIB.HSMS.090,ELEMENT=HSMSGC) _____ (2)
//SAVE-LLM MODULE-CONTAINER=*LIBRARY-ELEMENT(LIBRARY=HSMS.MACRO.PL)
//END
/
/START-EXECUTABLE-PROGRAM FROM-FILE=*LIBRARY-ELEMENT -
/ (LIBRARY=HSMS.MACRO.PL,ELEMENT=HSMS-MACRO-EXEC)

```

(1) Der Binder „Binder“ wird aufgerufen.

- (2) Beim Binden des erzeugten Moduls muss die HSMS-Systembibliothek zugewiesen werden, um den Entry HSMSGC einzubinden.

6.4 Arbeitsdateien

HSMS benötigt für den laufenden Betrieb eine Reihe von Arbeitsdateien. Die Steuer- und die Auftragsdatei sind für den HSMS-Betrieb von zentraler Bedeutung. Sie werden im Zuge der Installation angelegt. Die anderen Dateien dienen als Ausgabe- oder Hilfsdateien und werden nur zeitweise angelegt.

Ab HSMS V7.0 werden die Dateien neutral im NK4-Format angelegt (mit BLOCK-CONTROL-INFO=*WITHIN-DATA-4K-BLOCK und BUFFER-LENGTH=*STD(SIZE=4)).

Die maximale Größe der HSMS-Arbeitsdateien bleibt auf 32 GB beschränkt. Wenn diese Größe z.B. durch Eingriffe des Benutzers überschritten wird, kann HSMS die Arbeitsdatei nicht mehr öffnen und verarbeiten.

Steuerdateien

Alle Parameter, die für den HSMS-Betrieb vereinbart werden, werden in Steuerdateien hinterlegt. Deshalb sind Steuerdateien für den HSMS-Betrieb wichtig.

Eine zentrale Steuerdatei verwaltet die globalen HSMS-Parameter, alle Definitionen für die SF-Pubset-Umgebung (Archive und Pubsets) und die Definitionen für die Workstations. Es gibt nur eine zentrale Steuerdatei. Sie wird unter der Benutzerkennung SYSHSMS auf der Standard-Katalogkennung angelegt. Der Name dieser zentralen Steuerdatei lautet:

```
HSM.1.CONTROL.FILE
```

Zusätzlich zur zentralen Steuerdatei wird für jeden SM-Pubset unter HSMS-Kontrolle eine lokale Steuerdatei angelegt. Diese lokale Steuerdatei wird unter der Benutzerkennung SYSHSMS auf dem Control-Volume-Set des SM-Pubsets angelegt. Sie enthält die globalen HSMS-Parameter, die für die SM-Pubset-Umgebung spezifisch sind, sowie die Archive, die innerhalb dieser SM-Pubset-Umgebung definiert sind.

Der Name der lokalen Steuerdatei für einen SM-Pubset lautet:

```
HSM.1.SM.CONTROL.FILE
```

Die Steuerdateien werden als ISAM-Dateien mit USER-ACCESS=*OWNER-ONLY ohne Kennwortschutz eingerichtet.

Die Steuerdateien werden bei jedem Start einer HSMS-Session überprüft. Alle Definitionen, die in den Steuerdateien enthalten sind, werden in interne Tabellen gelesen. Da die Aufwärtskompatibilität der Steuerdateien gewährleistet ist, werden alte Datensätze automatisch erkannt und aktualisiert.

Eine Steuerdatei muss vor einem Upgrade kopiert werden, wenn ein Rückstieg nicht ausgeschlossen werden kann. Zur Kompatibilität von Steuerdateien siehe [Abschnitt „Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen“](#).

Eine neue zentrale Steuerdatei wird angelegt, wenn beim Start einer HSMS-Session keine zentrale Steuerdatei gefunden wird. Sie enthält bereits Voreinstellungen. Diese Voreinstellungen sind im Handbuch „HSMS Bd. 2“ [1] am Ende der HSMS-Anweisung MODIFY-HSMS-PARAMETERS aufgeführt.

Während einer HSMS-Session können die globalen HSMS-Parameter mit folgender Anweisung geändert werden:

```
//MODIFY-HSMS-PARAMETERS VALID-PERIOD=*PERMANENT,...
```

Die Steuerdatei eines SM-Pubsets wird während der Deklaration des SM-Pubsets mit der Anweisung CREATE-SM-PUBSET-PARAMETERS angelegt.

Eine neue Steuerdatei für einen SM-Pubset wird angelegt, wenn beim Start einer HSMS-Session oder beim Import eines Pubsets keine gefunden wird. Sie enthält bereits Voreinstellungen. Diese Voreinstellungen sind im Handbuch „HSMS Bd. 2“ [\[1\]](#) am Ende der Anweisung MODIFY-SM-PUBSET-PARAMETERS aufgeführt.

Die Parameter eines spezifischen SM-Pubsets können während einer HSMS-Session mit der Anweisung MODIFY-SM-PUBSET-PARAMETERS geändert werden.

Die zentrale HSMS-Steuerdatei enthält auch die Parameterdefinitionen der Pubsets und Workstations, die von HSMS verwaltet werden.

i Beim Start von HSMS wird überprüft, ob ferne Workstations in der Steuerdatei eingetragen sind. Da HSMS-SV im Normalfall ab HSMS V9.0B nicht mehr vorhanden ist, werden diese Workstations übersprungen und der HSMS-Start wird fortgesetzt. Die fernen Workstations werden nicht aus der Steuerdatei gelöscht.

Auftragsdatei

Die Aufträge, die HSMS auf Grund von Aktionsanweisungen oder DVS-Zugriffen auf verdrängte Dateien erzeugt hat, werden in die Auftragsdatei der SF- oder SM-Pubset-Umgebung eingetragen, für die sie erteilt wurden.

Die zentrale Auftragsdatei (Aufträge für eine SF-Umgebung) wird unter der Benutzerkennung SYSHSMS und unter der Standard-Katalogkennung angelegt. Der Name der Auftragsdatei lautet:

```
HSM.2.REQUEST.FILE
```

In einer SM-Pubset-Umgebung wird eine Auftragsdatei für jeden SM-Pubset unter der Benutzerkennung SYSHSMS und unter der Katalogkennung dieses SM-Pubsets angelegt. Der Name der Auftragsdatei für einen SM-Pubset lautet:

```
HSM.2.SM.REQUEST.FILE
```

Auftragsdateien, die mit früheren HSMS-Versionen angelegt wurden, sind nicht kompatibel. Die zentrale Auftragsdatei wird beim Start einer HSMS-Session überprüft. Wenn dabei eine inkompatible Auftragsdatei festgestellt wird, wird der Start abgebrochen. Zur Kompatibilität von Auftragsdateien siehe den [Abschnitt „Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen“](#).

Die Auftragsdateien für SM-Pubsets werden beim Start einer HSMS-Session (für die zugreifbaren Umgebungen) oder bei der Import-Bearbeitung überprüft. Wenn dabei eine inkompatible Auftragsdatei festgestellt wird, misslingt der Start der spezifischen Umgebung.

Wenn ein Task feststellt, dass keine Auftragsdatei vorhanden ist, wird automatisch eine neue Auftragsdatei angelegt. Sie wird leer und als ISAM-(alt)-Datei mit USER-ACCESS= *OWNER-ONLY ohne Kennwortschutz eingerichtet.

Aus dieser Datei heraus werden asynchrone Aufträge (wieder)gestartet. Zur Verwaltung der Aufträge stehen die HSMS-Anweisungen SHOW-REQUESTS, RESTART-REQUESTS und DELETE-REQUESTS zur Verfügung.

Dateien zur Bearbeitung einer Aktionsanweisung

Die *Ergebnisdatei* mit dem Namen:

```
HSM.A.<user-id><date><time>.RESULT
```

enthält das Ergebnis des ARCHIVE-Laufes, der zur Bearbeitung einer HSMS-Aktionsanweisung erzeugt wurde. Die Ergebnisdatei wird beim Datentransfer unter der Benutzerkennung des Aufrufers angelegt, ansonsten unter der Benutzerkennung SYSHSMS und unter der Standard-Katalogkennung der Umgebung, auf die sich der Auftrag bezieht (Standard-Katalogkennung oder SM-Pubset-Kennung).

Die Ergebnisdatei wird als ISAM-Datei mit USER-ACCESS=*OWNER-ONLY ohne Kennwortschutz eingerichtet. Die <user-id> im Dateinamen ist die Kennung des Benutzers, der den Auftrag erzeugt hat.

Die Ergebnisdatei benötigt sehr viel Plattenspeicher, wenn eine Vielzahl von Objekten bearbeitet wird; pro Objekt wird ein Satz erzeugt.

ARCHIVE benötigt die Ergebnisdatei zum Wiederanlauf durch die RESTART-REQUESTS-Anweisung. Sie wird automatisch nach normaler Beendigung oder nach einer DELETE-REQUESTS-Anweisung gelöscht.

Bei Aufträgen in einer SF-Umgebung für einen im Slave-Modus importierten Shared-Pubset, die am Master abgearbeitet werden, wird die Ergebnisdatei unter folgendem Namen angelegt:

```
HSM.A.<user-id><date><time>.<sysid>.RESULT
```

<sysid> bezeichnet die Systemidentifikation des auftraggebenden Systems im Shared-Pubset-Verbund.

Bei Aufträgen in einer SM-Umgebung wird eine Ergebnisdatei mit folgendem Namen im SM-Pubset erstellt:

```
HSM.A.<user-id><date><time>.<rq-name>.RESULT
```

<rq-name> ist der Auftragsname, den der Benutzer vergeben hat. Da der Auftragsname maximal 8 Zeichen lang sein darf, kann das Suffix „RESULT“ gekürzt sein, damit die Dateinamenslänge den DMS-Konventionen entspricht. Nur die ersten drei Zeichen des Suffix „RESULT“ sind garantiert.

Die *Berichtdatei* mit dem Namen

```
HSM.B.<user-id><date><time>.REPORT
```

enthält u.a. die Auswertung der ARCHIVE-Ergebnisdatei durch HSMS. Sie wird wie die Ergebnisdatei unter der Benutzerkennung SYSHSMS angelegt und unter der Katalogkennung der Umgebung, auf die sich der Auftrag bezieht (Standard-Katalogkennung oder SM-Pubset-Kennung).

Die Berichtdatei wird nicht mehrbenutzbar ohne Kennwortschutz eingerichtet. Sie wird nach dem Erzeugen der Druckdatei gelöscht.

Die Berichtdatei benötigt sehr viel Plattenspeicher, wenn eine Vielzahl von Objekten bearbeitet wird; pro Objekt wird ein Satz erstellt.

Bei Aufträgen in einer SF-Umgebung für einen im Slave-Modus importierten Shared-Pubset, die am Master abgearbeitet werden, wird die Berichtdatei auf dem (bzw. einem der) betroffenen Shared-Pubset unter folgendem Namen angelegt:

```
HSM.B.<user-id><date><time>.<sysid>.REPORT
```

<sysid> bezeichnet die Systemidentifikation des auftraggebenden Systems im Shared-Pubset-Verbund.

Bei Aufträgen in einer SM-Umgebung wird eine Berichtdatei mit folgendem Namen im SM-Pubset erstellt:

```
HSM.A.<user-id><date><time>.<rq-name>.REPORT
```

<rq-name> ist der Auftragsname, den der Benutzer vergeben hat. Da der Auftragsname maximal 8 Zeichen lang sein darf, kann das Suffix „REPORT“ gekürzt sein, damit die Dateinamenslänge den DMS-Konventionen entspricht. Nur die ersten drei Zeichen des Suffix „REPORT“ sind garantiert.

Die *Druckdatei* mit dem Namen

```
HSM.C.<user-id><date><time>.PRINT
```

enthält die druckaufbereitete Auswertung der Berichtdatei für den Benutzer, falls die Ausgabe des Reports auf Drucker vereinbart wurde. Sie wird unter der Benutzerkennung des Aufrufers (bzw. SYSHSMS für Aufträge des HSMS-Verwalters) angelegt und nach der Druckausgabe gelöscht.

Bei Aufträgen in einer SM-Umgebung wird eine Druckdatei mit folgendem Namen im SM-Pubset erstellt:

```
HSM.A.<user-id><date><time>.<rq-name>.PRINT
```

<rq-name> ist der Auftragsname, den der Benutzer vergeben hat. Da der Auftragsname maximal 8 Zeichen lang sein darf, kann das Suffix „PRINT“ gekürzt sein, damit die Dateinamenslänge den DMS-Konventionen entspricht. Nur die ersten drei Zeichen des Suffix „PRINT“ sind garantiert.

Dateien bei der Veränderung eines Archivverzeichnisses

Die Ergebnisdatei mit dem Namen

```
HSM.E.<user-id><date><time>.RESULT
```

enthält das Ergebnis eines ARCHIVE-Laufs, der zur Veränderung des Archivverzeichnisses erzeugt wurde. Die Ergebnisdatei wird unter der Benutzerkennung des Aufrufers (bzw. SYSHSMS für Aufträge des HSMS-Verwalters) angelegt.

Die Ergebnisdatei wird nach der Auswertung durch HSMS zum Erzeugen einer Berichtdatei gelöscht.

Die Berichtdatei mit dem Namen

```
HSM.F.<user-id><date><time>.REPORT
```

enthält die Auswertung der ARCHIVE-Ergebnisdatei durch HSMS. Sie wird wie die Ergebnisdatei unter der Benutzerkennung des Aufrufers (bzw. SYSHSMS für Aufträge des HSMS-Verwalters) angelegt.

Die Berichtdatei wird nach der druckaufbereiteten Ausgabe in die Systemdatei SYSLST gelöscht.

Die Berichtdatei benötigt sehr viel Plattenspeicher, wenn eine Vielzahl von Objekten bearbeitet wird; pro Objekt wird ein Satz erstellt.

Temporäre Dateien

Beim Zurückholen einer migrierten Datei wird auf dem Pubset, auf dem die Datei vor der Migration lag, eine Hilfsdatei mit dem Namen

```
HSM.D.<user-id><date><time>.<version><cfid>
```

unter der Benutzerkennung SYSHSMS kurzzeitig angelegt und nach der Verarbeitung gelöscht.

Wenn die HSMS-Anweisung RESTORE-LIBRARY-ELEMENTS bearbeitet wird, wird eine temporäre Datei mit dem Namen

HSM.P.<user-id><date><time>.PLAM

unter der Benutzerkennung SYSHSMS kurzzeitig angelegt und nach der Verarbeitung gelöscht.

Zur Erstellung der Protokolldatei für den BS2000 Backup Monitor wird unter Umständen eine temporäre Datei mit folgendem Namen angelegt:

\$SYSHSMS.HSM.C.SYSHSMS.<date_time>

Diese temporäre Datei wird nur erstellt, wenn beide folgenden Bedingungen erfüllt sind:

- Erweiterung des Benutzerprotokolls war erforderlich (Benutzerprotokoll wird in eine bereits existierende Datei geschrieben)
- Das Subsystem CONV2PDF ist nicht verfügbar oder es ist ein Fehler aufgetreten (siehe auch [Abschnitt „Zentrale Auftragsüberwachung am SE Server“](#))

Diese Datei wird wieder gelöscht, sobald die PDF-Protokoll-Datei \$SYSHSMS.HSM.C.SYSHSMS.<date_time>.PDF erzeugt wurde.

Arbeitsdateien für die Sicherung mit Concurrent Copy

Während einer Sicherung mit Concurrent Copy werden standardmäßig zwei temporäre Arbeitsdateien mit folgenden Namen angelegt:

S.DCH.<sysid>.<CC.session-id>.DATA

S.DCH.<sysid>.<CC.session-id>.META

Falls mit dem Operanden WORK-FILE-NAME der HSMS-Anweisung BACKUP-FILES ein Name festgelegt wurde, wird nur eine Arbeitsdatei angelegt, die den angegebenen Namen erhält.

Datei ARCHIVE.ASN zur Serialisierung paralleler Läufe

Mit HSMS V12.0A02 wird eine neue Arbeitsdatei mit Namen \$TSOS.ARCHIVE.ASN eingeführt.

Sie garantiert eindeutige SFIDs für Sicherungsdateien, die auf einer Shared S1-Ebene oder einem Shared SF-Pubset von gleichzeitig auf verschiedenen Systemen laufenden HSMS-Aufträgen angelegt werden. Die Datei ARCHIVE.ASN wird auf dem jeweiligen SF- oder S1-SM-Pubset während des ersten HSMS Sicherungslaufs auf dieses Pubset unter der Kennung TSOS angelegt. Die Datei speichert die zuletzt generierte ARCHIVE Sequence Number (ASN) des letzten Sicherungslaufs auf das Shared Pubset. Sie ist mit einem Lese-Passwort versehen; der Systemverwalter kann sie mit DELETE-FILE .. IGNORE-PROTECTION=*READ-PASSWORD gelöscht werden, falls das Pubset nicht mehr zur Aufbewahrung von Sicherungsdateien benutzt werden soll.

6.5 Möglichkeiten zur Performance-Verbesserung

Sowohl Benutzer als auch der HSMS-Administrator können bei Beachtung der nachfolgenden Hinweise ihre Sicherungszeiten reduzieren.

- Dateimenge sinnvoll eingrenzen durch Nutzung der Funktionen zur Dateiselektion
- dezentrale Verwaltung von SF-Pubsets
- Archivverzeichnisse nicht unnötig groß werden lassen und ggf. sinnvoll aufteilen (z.B. nach Katalogkennungen bzw. nach UFS-Laufwerken)
- zwischen den Vollsicherungen Differenzsicherungen bzw. Sicherungen nur nach Änderung von Katalogeinträgen durchführen
- Sicherung verkürzen durch Parallelbetrieb von MBK-Laufwerken
- bei Sicherung des BS2000-UFS Multiplexing nutzen
- Ausfallzeit der Anwendung verringern durch Sichern von "eingefrorenen" Kopien (BACKUP-FILES mit CCOPY)
- Verbesserung der Ein-/Ausgaben (maximale Bandgröße und Anbindung der Geräte)
- bei Shared-Pubset-Betrieb Aufträge auf dem Master-Rechner ausführen lassen (vermeidet Zeitstrafen beim Verschicken vieler Katalogoperationen)
- Reportumfang bei großen Dateimengen möglichst reduzieren (mit REPORT= *SUMMARY statt *FULL)
- ohne Wiederaufsetzpunkte arbeiten (WRITE-CHECKPOINTS=*NO)

6.5.1 Maximale Bandblockgröße

Für den besseren Durchsatz beim Sichern auf Band empfiehlt sich die Wahl großer Bandblöcke (Operand BLOCKING-FACTOR). Dabei wird auch der Anteil der Zwischenräume (Gaps) auf den Bändern kleiner und die Bandnutzung verbessert sich. Die Bandblockgröße kann entweder pro Anweisung, als Archiv-Attribut oder als globaler Parameter in ARCHIVE eingestellt werden.

Details sind im [Abschnitt „Größe von Bandblöcken“](#) beschrieben.

6.5.2 Beschleunigtes Kopieren von Sicherungsdateien

Beschleunigtes Kopieren von Sicherungsdateien kann erreicht werden, indem bei //COPY-SAVE-FILE keine Dateiauswahl getroffen wird. Dadurch können aufwendige Zugriffe auf das Archivverzeichnis vermieden werden.

7 Installation von HSMS

HSMS muss in Zusammenarbeit von Systemverwalter und HSMS-Verwalter installiert werden. Der Systemverwalter muss die entsprechende IMON-Installation für HSMS durchführen. Im Zuge dieser IMON-Installation werden ebenfalls alle notwendigen Dateien für ARCHIVE installiert.

Wenn der Systemverwalter HSMS das erste Mal lädt, geschieht dies im Modus DEFINE-SHOW, d.h. HSMS führt noch keine Aktionsanweisungen aus. Anschließend nimmt der HSMS-Verwalter die Einstellung der Steuerparameter und den Wechsel zum OPERATION-Modus vor.

Vor dem Installieren einer neuen HSMS-Version muss sich der Systemverwalter unbedingt über die Anforderungen an die Systemumgebung und über die Kompatibilität der verschiedenen HSMS-Versionen informieren. Der [Abschnitt „Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen“](#) erklärt, wie ein Versionsumstieg und -rückstieg durchgeführt wird und welche HSMS-Versionen für einen Ablauf von Shared-Pubsets kompatibel sind. Dieser Abschnitt sollte sorgfältig gelesen werden, da der Verlust einer Steuerdatei auch den Verlust der gesamten HSMS-Umgebung zur Folge haben kann.

Häufig war vor der Einführung von HSMS schon das Softwareprodukt ARCHIVE zur Datensicherung im Einsatz. Die von ARCHIVE erzeugten Directory-Dateien können von HSMS übernommen werden (siehe [Abschnitt „Besonderheiten bei Übergang von ARCHIVE- zu HSMS-Betrieb“](#), „ARCHIVE-Directory-Dateien übernehmen“).

7.1 Systemvoraussetzungen

- Bearbeitung von Daten im BS2000

7.1.1 Bearbeitung von Daten im BS2000

HSMS V12.0A kann mit BS2000 OSD/BC ab V10.0 eingesetzt werden.

Weiterhin benötigt HSMS V12.0A die folgende Software-Umgebung:

- ARCHIVE V12.0A (für die Bearbeitung der Aktionsanweisungen)
Sicherungsdateien, die mit älteren ARCHIVE-Versionen angelegt wurden, können mit ARCHIVE V12.0A gelesen und fortgesetzt werden. ARCHIVE ist Lieferbestandteil von HSMS.
- JV (Jobvariablen, für die automatische Verdrängung, Concurrent Copy; für die Bearbeitung von Jobvariablen, bei denen es Miteigentümer gibt)
- POSIX (für die Bearbeitung von Knotendateien des BS2000-UFS)
- NFS (zum Sichern/Restaurieren von Knotendateien auf passiven Workstations)
- SECOS (für die Bearbeitung von Dateien, bei denen es Miteigentümer gibt)
- SHC-OSD (für Einsatz von Concurrent Copy)
- MAREN (zur Datenträgerverwaltung)
- DAB (für die Pufferung von HSMS-Metadaten)

7.2 Lieferumfang von HSMS

Die folgenden Dateien sind Bestandteil der Lieferung von HSMS (<ver>=120):

Datei	Inhalt
SYSPRG.HSMS.<ver>	Lademodul für den Aufruf (START-PROGRAM) von HSMS
SYSLNK.HSMS.<ver>	Objektmodul-Bibliothek mit den Modulen der TU-Komponente von HSMS
SYSLNK.HSMS.<ver>.TPR SKMLNK.HSMS.<ver>.TPR	Objektmodul-Bibliothek mit den Modulen des Subsystems HSMS
SYSLIB.HSMS.<ver>	Bibliothek, enthält den HSMS-Makro und das für Programmaufrufe nötige Bindemodul
SYSFHS.HSMS.<ver>	Bibliothek, enthält die FHS-Bildschirmmasken;
SYSTEMS.HSMS.<ver>	Vollständige Meldungs-Primärdatei (MSGMAKER-Format)
SYSSDF.HSMS.<ver>	System-Syntaxdatei mit den HSMS-Anweisungen und Anweisungsoptionen für alle Benutzer. Die Anweisungen bzw. Operanden für den HSMS-Verwalter sind mit dem Privileg „HSMS-Verwalter“ gekennzeichnet.
SYSRMS.HSMS.<ver>	Datei mit Korrekturen für das Subsystem HSMS
SYSSSC.HSMS.<ver>	Datei mit den Subsystemdeklarationen zum Laden von HSMS
SYSNRF.HSMS.<ver>	NOREF-Datei für das Subsystem HSMS
SYSGM.HSMS.<ver>.D	Freigabemitteilung in deutscher Sprache
SYSGM.HSMS.<ver>.E	Freigabemitteilung in englischer Sprache
SYSSII.HSMS.<ver>	Datei mit IMON-Installationsinformation für HSMS
SYSPRC.HSMS.<ver>	Bibliothek mit kompilierten Prozeduren zur Unterstützung des BS2000 Backup Monitors
SYSLIB.ARCHIVE.<ver>	Makrobibliothek mit dem ARCHIVE-Makro
SYSLNK.ARCHIVE.<ver>	Lademodulbibliothek für den nichtprivilegierten Teil von ARCHIVE.
SYSLNK.ARCHIVE.<ver>.TPR	Lademodulbibliothek für den privilegierten Teil von ARCHIVE für den Einsatz auf Servern und Server Units mit /390-Architektur.
SKMLNK.ARCHIVE.<ver>.TPR	Lademodulbibliothek für den privilegierten Teil von ARCHIVE für den Einsatz auf Server Units mit x86-Architektur.
SYSNRF.ARCHIVE.<ver>	NOREF-Datei für das Subsystem ARCHIVE.
SYSPAR.ARCHIVE.<ver>	Parameterdatei (SAM) mit Voreinstellungen für die meisten ARCHIVE-Operanden
SYSTEMS.ARCHIVE.<ver>	Vollständige Meldungsdatei mit den ARCHIVE-Meldungen

SYSMH.ARCHIVE.<ver>	ISAM-Datei mit den deutschen und englischen Hilfetexten der HELP-Anweisung.
SYSPRG.ARCHIVE.<ver> (ARCHIVE)	Lademodul für die Kompatibilität mit dem Kommando /START-PROGRAM ARCHIVE
SYSPRG.ARCHIVE.<ver>. DIRCONV	Konvertierungsprogramm für Directory-Dateien.
SYSRMS.ARCHIVE.<ver>	RMS-Liefermenge für das Subsystem ARCHIVE.
SYSSII.ARCHIVE.<ver>	Datei mit Informationen über die Struktur und die Installation von ARCHIVE mit IMON.
SYSSDF.ARCHIVE.<ver>	Syntaxdatei für die Anweisungen im SDF-Format (DIRCONV) und die Kommandos /START-ARCHIVE und /START-DIRCONV.
SYSSSC.ARCHIVE.<ver>	Datei mit den Subsystemdeklarationen für DSSM

Mit HSMS wird eine SYSSII-Datei zur Verwendung unter IMON ausgeliefert. Sie enthält Informationen über die Struktur und die Installation des Produkts. Unter anderem enthält die SYSSII-Datei Informationen über das Produkt (Release-Unit) und über die Release-Items. Jedes Release-Item hat einen logischen Namen, der bei der Installation verwendet wird. Der logische Name ist versionsunabhängig; er ist der funktionsbeschreibende Teil des Dateinamens.

Im Folgenden finden Sie die Zuordnung der logischen Namen zu den Release-Items von HSMS:

Struktur der Release-Unit HSMS V12.0

Release-Item	Logischer Name
SYSPRG.HSMS.<ver>	SYSPRG
SYSFGM.HSMS.<ver>.E	SYSFGM.E
SYSFGM.HSMS.<ver>.D	SYSFGM.D
SYSFHS.HSMS.<ver>	SYSFHS
SYSLIB.HSMS.<ver>	SYSLIB
SYSLNK.HSMS.<ver>	SYSLNK
SYSLNK.HSMS.<ver>.TPR SKMLNK.HSMS.<ver>.TPR	SYSLNK.TPR
SYSMES.HSMS.<ver>	SYSMES
SYSREP.HSMS.<ver>	SYSREP
SYSRMS.HSMS.<ver>	SYSRMS
SYSNRF.HSMS.<ver>	SYSNRF
SYSSDF.HSMS.<ver>	SYSSDF
SYSSSC.HSMS.<ver>	SYSSSC
SYSPRC.HSMS.<ver>	SYSPRC

Struktur der Release-Unit ARCHIVE V12.0

Release item	Logischer Name
SYSLIB.ARCHIVE.<ver>	SYSLIB
SYSLNK.ARCHIVE.<ver>	SYSLNK
SYSLNK.ARCHIVE.<ver>.TPR SKMLNK.ARCHIVE.<ver>.TPR	SYSLNK.TPR
SYSNRF.ARCHIVE.<ver>	SYSNRF
SYSPAR.ARCHIVE.<ver>	SYSPAR
SYSMES.ARCHIVE.<ver>	SYSMES
SYSMESH.ARCHIVE.<ver>	SYSMESH

SYSPRG.ARCHIVE.<ver> (ARCHIVE)	SYSPRG
SYSPRG.ARCHIVE.<ver>.DIRCONV	SYSPRG.DIRCONV
SYSRMS.ARCHIVE.<ver>	SYSRMS
SYSSDF.ARCHIVE.<ver>	SYSSDF
SYSSSC.ARCHIVE.<ver>	SYSSSC

Nähere Informationen zu Release-Items, Release-Units und logischen Namen finden Sie im Handbuch „IMON“ [\[7\]](#).

7.3 Kompatibilität von HSMS-Definitionen in verschiedenen HSMS-Umgebungen

Vor dem Installieren einer neuen HSMS-Version sollten Sie unbedingt die Kompatibilität der verschiedenen HSMS-Versionen berücksichtigen. Bei HSMS spielt der Versionsstand eine besondere Rolle für

- die Archivdefinitionen, die in der Steuerdatei abgelegt sind.
- die Auftragsdefinitionen, die in der Auftragsdatei abgelegt sind.

Außerdem müssen Sie Folgendes berücksichtigen:

- Wenn Shared-Pubsets installiert sind:
Die Kriterien für den Auftragsversand vom Slave-Host zum Master-Host (siehe [Abschnitt „Shared-Pubsets und verschiedene HSMS-Versionen“](#)).
- Wenn SM-Pubsets installiert sind:
Die HSMS-Steuer- und Auftragsdateien, die auf diesen SM-Pubsets abgelegt sind.

7.3.1 SF-Steuerdatei (Home-Pubset)

Die SF-Steuerdatei von HSMS kann immer fortgeschrieben werden. Eine SF-Steuerdatei, die von einer niedrigeren HSMS-Version erstellt wurde, kann immer von einer höheren HSMS-Version gelesen und fortgeschrieben werden. Das Fortschreiben geschieht automatisch beim Erzeugen des Subsystems.

Allerdings ist die SF-Steuerdatei nicht abwärtskompatibel. Eine SF-Steuerdatei, die von einer höheren HSMS-Version erstellt wurde, kann nicht von einer niedrigeren HSMS-Version gelesen werden. Deshalb empfehlen wir dringend, die aktuelle SF-Steuerdatei zu kopieren, bevor eine neue HSMS-Version gestartet wird. Die Kopie der Steuerdatei ist bei einem eventuellen Rückstieg sehr wichtig.

7.3.2 SF-Auftragsdatei (Home-Pubset)

Die SF-Auftragsdatei von HSMS ist nicht kompatibel. Sie muss deshalb gelöscht oder geleert werden, bevor eine neue HSMS-Version gestartet wird.

Eine SF-Auftragsdatei wird geleert, indem alle in ihr enthaltenen Aufträge gelöscht werden. Dies geschieht mit der HSMS-Anweisung DELETE-REQUESTS der HSMS-Version, die die Aufträge erstellt hat.

Wenn beim Erzeugen des Subsystems ein nicht kompatibler Auftrag in der Auftragsdatei gefunden wird, kann das System nicht hochgefahren werden.

7.3.3 Shared-Pubsets und verschiedene HSMS-Versionen

Für Aufträge an Shared-Pubsets gilt ganz allgemein, dass ein Slave-Sharer die Aufträge zur Bearbeitung an den Master-Sharer sendet, falls in der Archiv-Definition BACKUP-SERVER-USAGE=*NO eingestellt ist. Wenn *STD eingestellt ist, werden die Aufträge zur Bearbeitung, abhängig vom HSMS-Parameter BACKUP-SERVER, an den Master-Sharer oder an den explizit eingestellten Backup-Server gesendet.

HSMS unterstützt das Senden von Aufträgen ohne Einschränkung, wenn am Slave-Sharer eine niedrigere HSMS-Version als am Master-Sharer im Einsatz ist. In HSMS < V10.0 werden die Aufträge an den Master-Sharer gesendet, da die Backup-Server-Funktion in diesen Versionen noch nicht unterstützt wird.

Wenn am Master-Sharer eine niedrigere HSMS-Version als am Slave-Sharer im Einsatz ist, wird der Auftrag nur angenommen, wenn der Master-Sharer alle Funktionen unterstützt, die zur korrekten Bearbeitung des Auftrags nötig sind (siehe [Abschnitt „Arbeiten mit SM-Pubsets und verschiedenen HSMS-Versionen“](#)).

7.3.3.1 Arbeiten mit S1-SM-Pubsets in verschiedenen HSMS-Versionen

Die Nutzung von S1-SM-Pubsets ist nur möglich, wenn auf allen auf das Pubset zugreifenden Systeme mindestens HSMS V10.0A läuft und der HSMS-Parameter SAVE-FILE-PROCESSING auf *HSMS-V10-COMPATIBLE eingestellt ist.

7.3.4 Steuer- und Auftragsdateien auf SM-Pubsets

SM-Pubsets können von Host zu Host transportiert oder von verschiedenen HSMS-Versionen gemeinsam benutzt werden. Deshalb sind die Steuer- und Auftragsdateien auf SM-Pubsets aufwärts- und abwärtskompatibel.

i Während der HSMS-Startup-Verarbeitung und der HSMS-Import-Verarbeitung sind die Auftragsdateien nach einer maximalen Wartezeit von 10 Minuten zugreifbar.

Eine Archivdefinition oder ein Auftrag, der von einer beliebigen HSMS-Version erstellt wird, kann von allen anderen HSMS-Versionen gelesen werden. Die Archivdefinition/der Auftrag kann aber nur bearbeitet werden, wenn die betreffende HSMS-Version die geforderte Funktion unterstützt (siehe [Abschnitt „Arbeiten mit SM-Pubsets und verschiedenen HSMS-Versionen“](#)).

7.3.5 Arbeiten mit der erweiterten Speicherebene S1 in einer SM-Umgebung

- Einrichten der erweiterten S1-Ebene
- Sicherung, Archivierung, Versions-Backup und Verdrängung auf die erweiterte S1-Ebene
- Erweiterte Speicherebene S1 wieder auf ein Volume-Set reduzieren
- Kompatibilität mit kleineren Versionen oder bei inhomogener Umgebung

7.3.5.1 Einrichten der erweiterten S1-Ebene

Mit der Anweisung CREATE-SM-PUBSET-PARAMETERS bzw. MODIFY-SM-PUBSET-PARAMETERS kann die Speicherebene S1 für eine SM-Umgebung definiert werden.

Dabei legt S1-VOLUME-SET=<cat-id> explizit das angegebene Volume-Set als S1-Ebene fest. Ab HSMS V11.0 kann mit S1-VOLUME-SET=*ALL-HSMS-CONTROLLED eine erweiterte Speicherebene S1 definiert werden. Damit stehen alle Volume-Sets des SM-Pubsets, die sich unter HSMS-Kontrolle befinden, als S1-Ebene zur Verfügung.

Zur Nutzung der erweiterten S1-Ebene in einer SM-Umgebung gelten folgende Voraussetzungen:

- Auf dem System wird BS2000 OSD/BC ab V11.0 und HSMS ab V11.0 eingesetzt. Im Falle eines Shared-Pubset-Verbunds müssen alle Pubset-Sharer des SM-Pubsets diese Voraussetzung erfüllen.
- Für den HSMS-Parameter SAVE-FILE-PROCESSING muss der Wert *HSMS-V10-COMPATIBLE angegeben sein. Im Falle eines Shared-Pubset-Verbunds gilt dies für alle Systeme, die auf den Pubset zugreifen.

 Alle Volume-Sets, die der S1-Ebene zugeordnet werden, müssen NK2 formatiert sein.

Die Anweisung CREATE-SM-PUBSET-PARAMETERS bzw. MODIFY-SM-PUBSET-PARAMETERS wird in folgenden Fällen abgewiesen:

- Wenn kein Volume-Set des SM-Pubsets für die Nutzung durch HSMS vorgesehen ist, wird die Angabe von *ALL-HSMS-CONTROLLED abgewiesen.
- Auf einem BS2000-System mit BS2000 OSD/BC kleiner V11 wird die Angabe von *ALL-HSMS-CONTROLLED abgewiesen.

Sobald auf einem SM-Pubset die erweiterte S1-Speicherebene eingerichtet ist, können Systeme mit HSMS V10 den SM-Pubset nicht mehr unter HSMS-Kontrolle bringen.

7.3.5.2 Sicherung, Archivierung, Versions-Backup und Verdrängung auf die erweiterte S1-Ebene

Nachdem die erweiterte S1-Ebene wie unter [Abschnitt „Einrichten der erweiterten S1-Ebene“](#) eingerichtet worden ist, kann die S1-Ebene wie bisher genutzt werden. Die Anweisungen zur Sicherung, Archivierung, Versions-Backup und Migration haben keine besondere Syntax zur Nutzung der erweiterten S1-Ebene.

Die Ablage der Sicherungsdateien wird durch den Allocator gesteuert und entzieht sich dem Einfluss des Benutzers. Allerdings kann der Systemverwalter das Anlegen neuer Sicherungsdateien auf bestimmten Volume-Sets mit folgendem Kommando verbieten:

```
/MODIFY-PUBSET-RESTRICTIONS PUBSET=<sm_pubset_catid>,  
  PUBSET-TYPE=*S-M(VOL-SET=<volset_catid>,  
  RESTRICTION=*NEW-FILE-ALLOCATION)
```

7.3.5.3 Erweiterte Speicherebene S1 wieder auf ein Volume-Set reduzieren

Wenn die erweiterte Speicherebene S1 bereits eine Weile genutzt wurde, können sich einzelne Sicherungsdateien über mehrere Volume-Sets verteilen. Deshalb muss eine erweiterte Speicherebene S1 leer sein, bevor sie wieder auf ein einzelnes Volume-Set reduziert werden kann.

Um die erweiterte Speicherebene S1 für einen SM-Pubset *SM01* auf das Volume-Set *VS01* zu reduzieren, ist folgende Anweisung notwendig:

```
//MODIFY-SM-PUBSET-PARAMETERS SM-PUBSET-ID=sm01,  
S1-DEFINITION=*PAR(S1-VOLUME-SET=vs01)
```

HSMS überprüft, ob sich auf Volume-Sets unter HSMS-Kontrolle noch Sicherungsdateien befinden. Ist dies der Fall, wird die Anweisung MODIFY-SM-PUBSET-PARAMETERS mit der Meldung HSM0500 zurückgewiesen. Das gilt auch, wenn die S1-Ebene mit SM-PUBSET-ID=*UNDEFINED ganz entfernt werden soll.

Wenn nach Nutzung der erweiterten Speicherebene S1 das Umschalten auf ein Volume-Set notwendig wird, muss der Administrator vorher die S1-Ebene leeren:

- Die Migrationsarchive mit der Anweisung MIGRATE-FILES von S1 nach S2 migrieren. Sicherungsdateien, die keine ungültigen Dateien enthalten, werden dabei nicht reorganisiert.
- Backup- und Langzeitarchive mit der Anweisung COPY-SAVE-FILE von S1 nach S2 kopieren. Dann die Sicherungsdateien mit folgender Anweisung von S1 löschen.
- Reorganisieren von Versions-Backup-Archiven auf S2 mit der Anweisung REORGANIZE-VERSION-BACKUP

```
//MODIFY-ARCHIVE ... SAVE-FILE=*DELETE
```

7.3.5.4 Kompatibilität mit kleineren Versionen oder bei inhomogener Umgebung

Vor dem Umschalten auf die erweiterte S1-Ebene muss sichergestellt werden, dass auf allen BS2000-Systemen, die auf den SM-Pubset zugreifen, BS2000 OSD/BC V11.0 oder höher und HSMS V11.0 oder höher im Einsatz sind.

Sobald auf einem Pubset-Sharer mit HSMS V11.0A oder höher die erweiterte S1-Ebene für einen SM-Pubset eingerichtet wurde, können alle Pubset-Sharer mit HSMS V10.0A oder früher den Pubset nicht mehr unter HSMS-Kontrolle bringen, weder beim IMPORT-PUBSET noch mit CREATE-SM-PUBSET-PARAMETERS.

Falls ein SM-Pubset auf einem Pubset-Sharer mit HSMS V10.0 benutzt wurde, bevor auf einem anderen Sharer die erweiterte S1-Ebene für diesen Pubset eingerichtet wurde, bleibt dieser Pubset unter der Kontrolle von HSMS V10. Allerdings sind folgende Funktionen nicht mehr möglich: Das Sichern, Kopieren und Migrieren auf die S1-Ebene und das Ändern von Pubset-Parametern. Lesezugriffe auf Sicherungsdateien in der S1-Ebene sind noch möglich: Wiederherstellen von gesicherten Dateien, Zurückholen migrierter Dateien, Kopieren von Sicherungsdateien in die S2-Ebene, Migrieren von der S1-Ebene zur S2-Ebene, Ändern von Dateien in der S1-Ebene und Löschen von Dateien von der S1-Ebene.

Systeme mit HSMS V9.0 oder älter können in keinem Fall auf Dateien in der erweiterten S1-Ebene zugreifen.

Informationsausgabe mit SHOW-SM-PUBSET-PARAMETERS

Auf einem Pubset-Sharer mit HSMS kleiner V11.0 zeigt die Anweisung SHOW-SM-PUBSET-PARAMETERS bei der erweiterten S1-Ebene statt **ALL-HSMS-CONTROLLED** den Wert **NOT-DEFINED** an. Das Feld HSMS VERSION hat den Wert 110.

Informationsausgabe mit SHOW-PUBSET-USAGE

Wenn auf einem Pubset-Sharer (mit HSMS V11.0 oder höher) die erweiterte S1-Ebene für den Pubset eingerichtet wurde, verhält sich die Anweisung SHOW-PUBSET-USAGE auf einem Sharer desselben Pubsets mit HSMS kleiner als V11.0 so, als ob keine S1-Ebene definiert sei:

- mit INFORMATION=*REUSABLE-S1-SPACE wird die Anweisung mit der Meldung HSM0213 abgewiesen.
- mit INFORMATION=*MIGRATION-EVALUATION, werden Dateien die auf die S1-Ebene migriert wurden, mit 'S1-NOT-DEF' gekennzeichnet.

7.3.6 Versions-Backup in einer Shared-Umgebung

In einer inhomogenen BS2000/HSMS-Umgebung wird die Anweisung `BACKUP-FILE-VERSIONS` nicht auf den Master-Rechner oder Backup-Server mit BS2000 OSD/BC < V11.0B oder HSMS < V12.0A übertragen. In diesem Falle wird die Anweisung `BACKUP-FILE-VERSIONS` mit entsprechender Meldung auf dem Host abgebrochen, der den Auftrag startet.

Hinweise:

- Es wird dringend empfohlen, den gleichen Wert des Systemparameters `NUMBACK` für alle Sharer mit BS2000 OSD/BC ab V11.0B anzugeben. Sonst werden neu erstellte Dateien verschiedene Werte von `NUMBER-OF-BACKUP-VERSIONS` haben, je nach dem, von welchem System sie erstellt wurden.
- Die Anweisung `//BACKUP-FILE-VERSIONS` ist nur im Modus `*HSMS-V10-COMPATIBLE` möglich.
- Restaurieren aus dem Versions-Backup-Archiv ist nur ab HSMS V12.0A möglich. Im Falle von Slave-, Master- oder Backup-Servern muss HSMS die Version V12.0A oder höher haben.
- Restaurieren aus dem Versions-Backup-Archiv ist nur ab BS2000 OSD/BC V11.0B oder höher möglich.
- `//CHECK-CATALOGED-FILES` ist nur für Systeme mit BS2000 OSD/BC V11.0B oder höher möglich.
- `//REORGANIZE-VERSION-BACKUP` erfordert BS2000 OSD/BC V11.0B oder höher und ist nur im Modus `*HSMS-V10-COMPATIBLE` möglich.

Kompatibilität in einer Shared-SM-Umgebung.

Sobald einem SM-Pubset unter HSMS ab V12.0A ein System-Archiv für Versions-Backup zugeordnet worden ist, ist das Ändern der SM-Pubset-Parameter mit `MODIFY-SM-PUBSET-PARAMETERS` auf allen Sharern mit niedrigeren HSMS-Versionen nicht mehr möglich. Die Funktion steht wieder zur Verfügung, wenn das System-Backup-Archiv wieder als `*UNDEFINED` eingestellt wird.

7.4 Benutzererkennung SYSHSMS

Die Benutzererkennung SYSHSMS ist für den Betrieb von HSMS nötig. Auf diese Kennung müssen die Produktdateien von HSMS eingespielt werden; auf dieser Kennung richtet HSMS zentrale Arbeitsdateien ein. Das System richtet die Benutzererkennung SYSHSMS bei einem First-Start automatisch ein. Ihre Attribute kann der Systemverwalter mit dem Kommando MODIFY-USER-ATTRIBUTES ändern. Die Standard-Pubsetkennung (DEFAULT-PUBSET) der Kennung SYSHSMS darf nicht auf einen Shared-Pubset gelegt werden.

Das PUBLIC-SPACE-LIMIT sollte so hoch gesetzt werden, dass alle nötigen Dateien angelegt werden können. Um einen Abbruch beim Anlegen größerer Arbeitsdateien zu vermeiden, sollte das Überschreiten dieses Limits (PUBLIC-SPACE-EXCESS) erlaubt werden. Diese Empfehlung gilt besonders, wenn die Migrationsfunktion von HSMS genutzt wird. Der automatische Recall-Mechanismus holt eine migrierte Datei zuerst unter eine temporäre Datei unter der Benutzererkennung SYSHSMS auf dem Pubset der migrierten Datei zurück, bevor die Zuweisung der Datei auf den endgültigen Katalogeintrag geändert wird. Zum Zurückholen von sehr großen Dateien muss ausreichend Speicherplatz für SYSHSMS vorhanden sein.

Beim Festlegen des PUBLIC-SPACE-LIMIT ist auch zu berücksichtigen, dass unter der SYSHSMS-Kennung auch die Protokolldateien zur Anzeige im BS2000 Backup Monitor des SE Managers abgelegt werden. Wenn der Auftrag gelöscht wird, wird auch die PDF-Datei gelöscht.

7.5 Umgebung zur Sicherung ferner Knoten-S0 erzeugen

Für die Sicherung von Knotendateien mit NFS muss der Systemverwalter zum Erstellen der benötigten Umgebung folgende Tätigkeiten ausführen:

- Er muss im lokalen BS2000-UFS unterhalb des Root-Verzeichnisses das Dateiverzeichnis „HSMS“ definieren.
- Er muss für jeden fernen Rechner, auf den HSMS Zugriff haben soll, ein Dateiverzeichnis innerhalb des Dateiverzeichnisses „HSMS“ erzeugen. Der Name dieses Dateiverzeichnisses muss mit dem Namen des fernen Rechners identisch sein.

7.6 Erster Start von HSMS

Der erste Start von HSMS wird als normaler Start einer HSMS-Session durchgeführt, außer dass in der Regel die Steuer- und Auftragsdateien fehlen oder noch von einer vorhergehenden HSMS-Version vorhanden sind. Die normale Startup-Bearbeitung berücksichtigt dies aber.

Bei jedem Start einer HSMS-Session werden alle Steuer- und Auftragsdateien in folgender Reihenfolge überprüft:

1. Zuerst wird die zentrale Steuerdatei überprüft und in interne Tabellen gelesen.
 - Wenn keine zentrale Steuerdatei vorhanden ist, wird eine neue angelegt. Diese Steuerdatei enthält bereits Voreinstellungen, die dann für den Start verwendet werden.
 - Wenn die zentrale Steuerdatei vorhanden ist und Definitionen enthält, die von einer früheren HSMS-Version stammen, werden die Definitionen erkannt und automatisch berichtigt. Der Rückstieg auf eine kleinere Version wird nicht unterstützt!
 - Wenn die zentrale Steuerdatei ungültige Sätze enthält oder inkonsistent ist, wird der Start abgebrochen. Zusätzlich werden Fehlermeldungen am Bedienplatz ausgegeben.
2. Als Nächstes wird die zentrale Auftragsdatei überprüft.
 - Wenn die zentrale Auftragsdatei nicht vorhanden ist, wird eine neue, leere Auftragsdatei angelegt.
 - Wenn die zentrale Auftragsdatei vorhanden ist und Datensätze enthält, die von einer früheren HSMS-Version stammen, wird der Start mit Fehlern abgebrochen. Die Auftragsdatei kann nicht berichtigt werden; sie muss deshalb gelöscht werden, bevor die Startup-Bearbeitung erneut gestartet wird.
3. Anschließend werden die lokale Steuerdatei und die lokale Auftragsdatei eines jeden zugreifbaren SM-Pubsets überprüft. Zuerst wird die lokale Auftragsdatei überprüft und in interne Tabellen geschrieben.
 - Wenn keine lokale Steuerdatei für eine gegebene SM-Pubset-Umgebung vorhanden ist, wird eine neue angelegt. Diese Steuerdatei enthält bereits Voreinstellungen, die für den Start der SM-Pubset-Umgebung verwendet werden.
 - Wenn eine lokale Steuerdatei vorhanden ist und Definitionen enthält, die von einer früheren HSMS-Version stammen, werden die Definitionen erkannt und automatisch berichtigt.
 - Wenn eine lokale Steuerdatei ungültige Sätze enthält oder inkonsistent ist, wird der lokale Start abgebrochen und die SM-Pubset-Umgebung übersprungen. Der Start wird mit der nächsten SM-Pubset-Umgebung fortgesetzt.
 - Wenn auf einen SM-Pubset, der sich unter HSMS-Kontrolle befindet, beim Start der HSMS-Session nicht zugegriffen werden kann, wird er übersprungen. Er wird aber gestartet, wenn der SM-Pubset an den Host importiert wird. Dort wird dann derselbe Start durchgeführt (überprüfen und lesen der lokalen Auftragsdatei).

Nachdem der Start von HSMS abgeschlossen ist, können die globalen HSMS-Parameter mit folgender HSMS-Anweisung an die Bedürfnisse des jeweiligen Systems angepasst werden:

```
//MODIFY-HSMS-PARAMETERS VALID-PERIOD=*PERMANENT, . . .
```

Die Voreinstellungen nach dem ersten Start von HSMS sind im Handbuch „HSMS Bd.2“ [1] bei der HSMS-Anweisung MODIFY-HSMS-PARAMETERS beschrieben.

Die SM-Pubset-spezifischen Parameter können mit folgender HSMS-Anweisung geändert werden:

```
//MODIFY-SM-PUBSET-PARAMETERS
```

Die Voreinstellungen für die Steuerdatei eines SM-Pubsets sind im Handbuch „HSMS Bd. 2“ [\[1\]](#) bei der HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS beschrieben.

Schrittweise kann dann über die Betriebsweise SIMULATION, in der Aktionsanweisungen nur simuliert werden, also unerwünschte Manipulationen an Daten noch ausgeschlossen sind, zum Normalbetrieb übergegangen werden. In diesem Modus können Archive eingerichtet und Pubsets unter HSMS-Verwaltung einbezogen werden.

Nach dem ersten Start von HSMS ist nur der Standard-Pubset der Kennung SYSHSMS unter der Verwaltung von HSMS. Er ist der Speicherebene S0 zugeordnet.

Für alle anderen Pubsets gelten die Voreinstellungen, die im Handbuch „HSMS Bd. 2“ [\[1\]](#) bei der HSMS-Anweisung MODIFY-PUBSET-PARAMETERS aufgelistet sind.

Das Einbeziehen der SF-Pubsets in die HSMS-Verwaltung ist im [Abschnitt „SF-Pubsets verwalten“](#) beschrieben.

Wie SM-Pubsets unter HSMS-Verwaltung gebracht werden, ist im [Abschnitt „Bearbeitung von SM-Pubsets“](#) beschrieben.

7.7 Aufgaben bei jedem Startup des BS2000-Systems

Der Systemverwalter muss bei jedem Startup des BS2000-Systems alle fernen Dateisysteme, die bearbeitet werden sollen, am richtigen Knotenpunkt des lokalen BS2000-UFS einhängen. Ein fernes Dateisystem kann aber nur dann eingehängt werden, wenn es an dem Rechner, der Eigentümer dieses Dateisystems ist, als mehrfachbenutzbar deklariert wurde.

Wenn ein Dateisystem auf einem neuen, fernen Rechner bearbeitet werden soll, muss der Systemverwalter im lokalen BS2000-UFS einen entsprechenden Knoten innerhalb des Dateiverzeichnisses „HSMS“ definieren.

7.8 Hinweise zur Einführung der Verdrängung

Bei der Einführung der Verdrängung sind bestimmte Maßnahmen zu treffen:

- Nicht alle Dateien dürfen dem Verdrängungsmechanismus unterliegen. Der HSMS-Verwalter sollte die Hinweise im [Abschnitt „Ausnahmedatei“](#) beachten, alle Benutzer die Hinweise im [Abschnitt „Migrationssperre“](#).
- Es muss unbedingt beachtet werden, dass in vielen Fällen Dateien für sehr lange Zeit migriert bleiben. Es wird zwar empfohlen nur Dateien zu migrieren, die bereits gesichert worden sind (dies wird gewährleistet durch den HSMS-Parameter `MIGRATION-CONTROL(BACKUP-MANDATORY=*YES)`), allerdings werden diese Sicherungen nach Ablauf der Retention-Period obsolet und werden aus dem Backup-Archiv gelöscht (`//MODIFY-ARCHIVE SAVE-FILE=*DELETE`). Ab diesem Zeitpunkt liegt keine Kopie der Datei mehr vor.
Deshalb sollten die migrierten Dateien, wenn die Sicherungszeitfenster es zulassen, wenigstens von Zeit zu Zeit Sicherungen der Migrations-Archive durchgeführt werden: `BACKUP-FILES ... SAVE-OPTIONS=*PARAMETER (SAVE-DATA=*S2-S2-S0)`. So kann gewährleistet werden, dass stets Kopien der Daten von allen Speicherebenen existieren. Versions-Backup kann dazu auch verwendet werden.
- Abgesehen davon sollte HSMS möglichst frühzeitig nach der Systemeinleitung gestartet werden, um das Zurückholen der Dateien so schnell wie möglich zu gewährleisten.
- Jobs, die bisher mit `/SHOW-FILE-ATTRIBUTES` oder `/ADD-FILE-LINK` geprüft haben, ob während des Ablaufs benötigten Dateien vorhanden sind, sollten dies stattdessen mit `/SECURE-RESOURCE-ALLOCATION` mit Wartezeit tun (siehe [Abschnitt „Implizites Zurückholen bei Dateizugriff“](#)).
- Anwendungen, die den Internen Dateinamen (CFID) zur Prüfung und Bearbeitung von Dateien (z.B. UPAM) benutzen, müssen Folgendes beachten: Der Vergleich des Katalogeintrags vor und nach dem OPEN ist nicht zulässig, da beim OPEN auf eine verdrängte Datei diese zurückgeholt und der Interne Dateiname (CFID) dabei geändert wird. Beim Absetzen eines FSTAT-Makros vor der Verarbeitung darf man sich also den Katalogeintrag nicht für eine Prüfung merken. Besser ist ein `/SECURE-RESOURCE-ALLOCATION` mit Wartezeit, da die Datei dann bereits zurückgeholt wird.
- Für die Benutzerkennung SYSHSMS sollte der Speicherplatz (`PUBLIC-SPACE-LIMIT`) nicht begrenzt werden, wenn sehr große Dateien implizit zurückgeholt werden. Beim Zurückholen wird die Datei auf dem Pubset, auf der sie vor der Migration lag, zuerst unter der Benutzerkennung SYSHSMS importiert, bevor die Zuweisung auf die endgültige Benutzerkennung geändert wird. Beim Einrichten der Benutzerkennung SYSHSMS sollte aus diesem Grund die Speicherplaterweiterung (`PUBLIC-SPACE-EXCESS`) zugelassen werden

Hinweis

Im Anhang sind die Informationen, die Benutzer vor der Einführung der Verdrängung erhalten sollten, noch einmal gesondert gesammelt. Deshalb ist der Anhang vom Kopierverbot ausgenommen. Der Systemverwalter darf Kopien an zukünftige Benutzer bzw. von der Verdrängung betroffene Benutzer verteilen.

8 Zusammenarbeit von HSMS mit ARCHIVE

HSMS arbeitet eng mit dem BS2000-Softwareprodukt ARCHIVE zusammen.

8.1 ARCHIVE-Funktionen in HSMS

HSMS setzt auf dem Softwareprodukt ARCHIVE auf. HSMS bietet die Funktionen von ARCHIVE an, jedoch mit einer moderneren Benutzeroberfläche und zusätzlichen Steuerungsmöglichkeiten.

Während ARCHIVE die Funktionen Datensicherung und Langzeitarchivierung nicht trennt, unterscheidet HSMS diese Funktionen klar. Es bietet getrennte Anweisungen für die einzelnen Funktionen und eine getrennte Verwaltung der jeweiligen Bestände.

Die meisten Funktionen von ARCHIVE unterstützt auch HSMS. Da aber HSMS eine andere Benutzerschnittstelle anbietet, müssen dieselben Funktionen unter HSMS mit anderen Anweisungen angesprochen werden. Als Hilfe für Benutzer, die mit dem Softwareprodukt ARCHIVE vertraut sind, sind in der folgenden Tabelle einige Beispiele für die Abbildung von ARCHIVE-Anweisungen und -Operanden auf HSMS aufgeführt.

In einer weiteren Tabelle sind die Operanden der PARAM-Anweisung von ARCHIVE und die entsprechenden Operanden von HSMS-Anweisungen abgebildet.

ARCHIVE bleibt für alle Benutzer weiterhin zugänglich.

Folgende Funktionen sind nur mit HSMS möglich:

- alle Aktionen, die Migration und Knotendateien betreffen
- die Fortsetzung von Sicherungen (Several-SVID-Format)
- Versions-Backups
- einige Zusatzfunktionen wie der Restore von Bibliothekselementen (ermöglicht durch Sichern der Meta-Informationen von PLAM-Bibliotheken) oder alle an Archiv-Attribute gebundenen Funktionen
- über Operanden „nur Dateien auf Net-Storage auswählen“ oder „Dateien auf Net-Storage ausschließen“ (eine Steuerung besteht nur über die Parameterdatei für die gesamte ARCHIVE-Session, siehe „Sicherung von Net-Storage-Dateien“, "[Einschränkungen durch neue Funktionen](#)").

Gerätereservierung in HSMS

HSMS reserviert die für den Lauf notwendigen Bandgeräte selbst. Bandgeräte, die der Benutzer vor einem HSMS-Lauf in seiner Task selbst reserviert, stehen HSMS nicht zur Verfügung.

Gegenüberstellung von ARCHIVE- und HSMS-Anweisungen

ARCHIVE-Anweisung	HSMS-Anweisung
DELETE	DELETE-REQUESTS
EXPORT	EXPORT-FILES
,DIRSAVE=*YES	,DIRECTORY-NAME=<dateiname> – (SAVE-DIRECTORY=*YES)
IMPORT	IMPORT-FILES
FILES	SELECT-FILE-NAMES und Operand FILE-NAMES (SELECT-FILES bei COPY-SAVE-FILE)
,FROM=*PRDISC	BACKUP-FILES SUPPORT=*PRIVATE-DISK (VOLUMES=*ALL)
,FROM=*PRTAPE	BACKUP-FILES SUPPORT=*TAPE
INQUIRE	SHOW-ARCHIVE
JOBVAR	SELECT-JV-NAMES und Operand JV-NAMES (SELECT-JV bei COPY-SAVE-FILE)
LIST	LIST-VOLUMES
POOL	MODIFY-ARCHIVE VOLUMES=*ADD/*REMOVE
PURGE	MODIFY-ARCHIVE SAVE-FILES=*DELETE
PROCESS	RESTART-REQUESTS
RESTORE	RESTORE-FILES
,FROM=*LATEST	,SELECT-SAVE-VERSIONS=*ALL
,FROM=*LATEST,*STATE	,SELECT-SAVE-VERSIONS=*LATEST
SAVE	ARCHIVE-FILES, BACKUP-FILES
,CHANGED=*NO	BACKUP-FILES SELECT-FILES=*ALL-FILES
,CHANGED=*YES,*LARGE	BACKUP-FILES SELECT-FILES=*MODIFIED-FILES (PARTIAL-FILE-SAVE=*LARGE-FILES)
,DIRSAVE=*YES	BACKUP-FILES SAVE-OPTIONS=*PAR (SAVE-DIRECTORY=*YES)
,DRIVES=	BACKUP-FILES OPERATION-CONTROL (PARALLEL-RUNS=)

STATUS	SHOW-REQUESTS
--------	---------------

Tabelle 6: Abbildung von ARCHIVE-Anweisungen auf HSMS-Anweisungen

Operanden der PARAM-Anweisung von ARCHIVE	HSMS-Operanden
CNS=*YES/*NO	REPORT=*FULL/*SUMMARY
RESTART=*YES/*NO	WRITE-CHECKPOINTS=*YES/*NO
UNLOAD=*YES/*NO	UNLOAD-TAPE=*YES/*NO
OPERATOR=*YES/*NO	OPERATOR-INTERACTION=*YES/*NO
WRCHK=*YES/*NO	WRITE-CHECK=*YES/*NO
SNR=*YES/*NO	REPORT=*FULL/*RESTORED-FILES
DESTROY=*YES/*NO	DESTROY-BY-DELETE=*YES/*NO
CATID=*YES/*NO	CATALOG-ID-MODE=*YES/*NO (nur bei Export/Import, ansonsten arbeitet HSMS nur mit Katalogkennung)
OLS=*YES/*NO	SAVE-OPTIONS=*PAR (SAVE-ONLINE-FILES=*YES/*NO) bei BACKUP-FILES

Tabelle 7: Abbildung von PARAM-Operanden auf HSMS-Operanden

8.2 Besonderheiten bei Übergang von ARCHIVE- zu HSMS- Betrieb

ARCHIVE-Directory-Dateien übernehmen

Häufig war vor der Einführung von HSMS schon das Softwareprodukt ARCHIVE zur Datensicherung im Einsatz. Die von ARCHIVE erzeugten Directory-Dateien kann HSMS übernehmen, falls die Directory-Datei Katalogkennungen zulässt, d.h. mit der ARCHIVE-Anweisung PARAM CATID=*YES erzeugt wurde. Eine Directory-Datei ohne Katalogkennungen kann mit dem Programm DIRCONV, das zum Lieferumfang von ARCHIVE gehört, in eine Directory-Datei mit Katalogkennungen umgewandelt werden.

HSMS kann die Datensicherung mit dem bisherigen Datenträger-Pool und dem Bestand an Sicherungsdateien fortsetzen, wie er mit ARCHIVE aufgebaut wurde. Dazu wird ein HSMS-Archiv eingerichtet. Die alte ARCHIVE-Directory-Datei kann diesem HSMS-Archiv als Archivverzeichnis zugeordnet werden. Dies geschieht mit:

```
//CREATE-ARCHIVE ARCHIVE-NAME=<archivname>, -  
// DIRECTORY-NAME=<dateiname> (NEW-DIRECTORY=*NO)
```

Für dieses Archiv steht anschließend der volle HSMS-Funktionsumfang zur Verfügung.

ARCHIVE-Directory-Dateien sollten nur in Backup-Archive übernommen werden, weil dies mit den geringsten Einschränkungen verbunden ist.

Umsetzung von Export-Sicherungen ohne Directory in Several-SVID-Sicherungen mit Directory

Beim Übergang von Magnetbandgeräten mit kleinen Kassetten zu Geräten mit sehr großen Kassetten (z.B. LTO) ist auch ein Übergang von ARCHIVE- auf HSMS-Betrieb ratsam, da HSMS die Fortsetzung von Sicherungen (Several-SVID-Format) ermöglicht. Dabei sollten auch Export-Sicherungen ohne Directory, die im ARCHIVE-Betrieb auf kleinen Volumes erstellt wurden, kopiert werden auf fortgeschriebene HSMS-Langzeitsicherungen auf großen Volumes. Diese Sonderfunktion für den Übergang von ARCHIVE- auf HSMS-Betrieb steht ab HSMS V8.0A innerhalb der Anweisung COPY-EXPORT-SAVE-FILE zur Verfügung. Dabei kann die Eingabe-Sicherungsdatei als Volume-Liste angegeben werden und die Ausgabe-Sicherungsdatei im Several-SVID-Format mit der Option zum Fortsetzen erstellt werden.

Das bei COPY-EXPORT-SAVE-FILE erzeugte Directory kann beim Anlegen eines Langzeitarchivs als HSMS-Archivverzeichnis angegeben werden.

8.3 Einschränkungen für den Rückstieg auf ARCHIVE

Für den Rückstieg auf ARCHIVE sind einige Einschränkungen zu beachten, die in diesem Abschnitt beschrieben sind.

8.3.1 Kompatibilität der Directory-Dateien

Im Rahmen eines Umstiegs von ARCHIVE auf HSMS, können bereits vorhandene ARCHIVE-Directory-Dateien in neu anzulegende HSMS-Backup-Archive aufgenommen werden. Wenn die Directory-Datei mit dem Archiv-Attribut `SAVE-FILE-STRUCTURE= *SEVERAL-SVID` aufgenommen wurde und Sicherungen unter HSMS erstellt worden sind, ist ein Rückstieg in den ARCHIVE-Betrieb unter Nutzung dieser Directory-Datei nicht mehr möglich. Grundsätzlich dürfen Archivverzeichnisse (Directorys) von Backup-Archiven in SEVERAL-SVID-Struktur, sowie Versions-Backup-, Migrations- und Langzeitarchive keinesfalls unter ARCHIVE verwendet werden.

8.3.2 Einschränkungen durch neue Funktionen

Einschränkungen für den Rückstieg auf ARCHIVE ergeben sich aus folgenden Funktionen, die in HSMS neu sind:

- neue Schutzmechanismen
- Fortschreiben von Sicherungsdatenträgern
- Sicherungen auf S1-Pubsets oder Volume-Sets
- Einführung des Versions-Backups
- Einführung migrierter Dateien
- Sicherung von Knotendateien des BS2000-UFS (POSIX) oder ferner Knoten-S0 über POSIX
- Sicherung/Wiedergewinnung von Dateien, bei denen es Miteigentümer gibt
- Sicherung von Dateien auf Net-Storage

Die Einschränkungen beim Rückstieg auf ARCHIVE sind im Allgemeinen abhängig von der ARCHIVE-Version, auf die zurückgegangen wird. Sie sind im Folgenden beschrieben.

Einschränkungen durch andere Schutzmechanismen

Da HSMS die Sicherungsaufträge durch einen privilegierten Servertask unter TSOS abarbeitet und nicht wie bei ARCHIVE durch einen Subtask unter der eigenen Kennung, kann auf Sicherungen von Dateien der eigenen Benutzerkennung nicht ohne weiteres zugegriffen werden.

Wenn unter der Kennung gearbeitet wird, unter der die Directory-Datei katalogisiert ist, oder unter TSOS, so können mit ARCHIVE Dateien aus dieser Directory-Datei restauriert werden. Ansonsten ist der Zugriff mit ARCHIVE nur für solche Sicherungsdateien in einer Directory-Datei unter TSOS möglich, die als mehrbenutzbare Dateien eingerichtet wurden (siehe Handbuch „HSMS Bd. 2“ [1], Anweisung BACKUP-FILES, Operand USER-ACCESS=*ALL-USERS).

Eine Directory-Datei, die HSMS auf der Benutzerkennung SYSHSMS erzeugt hat, muss also gegebenenfalls auf TSOS kopiert werden.

Einschränkungen durch Sicherungen auf Band

Bei Sicherungen auf Band im Format SEVERAL-SVID sowie generell bei Langzeitarchivierung und Verdrängung legt HSMS Sicherungsdateien so an, dass sie aus mehreren Sicherungsversionen bestehen können. Dies hat Auswirkungen auf den Rückstieg.

Die Sicherungsdatenträger können unter TSOS gelesen werden. Allerdings muss bei Datenträgern, die mehrere Sicherungsversionen enthalten, eine Save-Version-ID (SVID) angegeben werden. Die LIST-Anweisung kann nur verwendet werden, wenn die SVIDs auf dem Datenträger in aufsteigender Reihenfolge vorliegen.

Die Bearbeitung der Directory-Datei wird nur bei IMPORT/EXPORT-Läufen unterstützt. Mehrfachsicherungen einer Datei in einer Sicherungsdatei können mit IMPORT gezielt eingespielt werden.

ARCHIVE unterstützt nicht das Fortschreiben einer Sicherungsdatei mit mehreren Sicherungsversionen.

Einschränkungen durch Sicherungen auf Pubsets

Bei HSMS sind Sicherungen auf Pubsets (S1) möglich. Dies hat für einen Rückstieg auf ARCHIVE folgende Konsequenzen:

Mit Directory-Datei sind Restaurieren oder Auflisten ohne Einschränkung möglich, sofern die Sicherung unter HSMS mit der Einstellung SAVE-FILE-PROCESSING=*HSMS-V9-COMPATIBLE stattgefunden hat. Ohne Directory-Datei kann nur auf eine Sicherungsdatei zugegriffen werden, die auf dem Home-Pubset liegt. Gegebenenfalls muss sie dorthin kopiert werden.

Einschränkungen durch migrierte Dateien

Mit HSMS wurden migrierte Dateien eingeführt. Für den Rückstieg auf ARCHIVE hat dies folgende Konsequenzen:

Ein EXPORT einer migrierten Datei oder ein SAVE auf einer Benutzerkennung != TSOS wird mit einer Fehlermeldung zurückgewiesen.

Der Katalogeintrag einer migrierten Datei kann nur unter der Benutzerkennung TSOS gesichert werden; beim Restaurieren ist eine Umbenennung nicht zugelassen. Der SPACE-Operand wird ignoriert.

Restaurieren von Dateien von Knoten-S0:

Zum Restaurieren von Dateien des BS2000-UFS (POSIX) oder ferner Knoten-S0 (z.B. UNIX-Workstations), die über POSIX gemountet und mit HSMS gesichert wurden (BACKUP-NODE-FILES, ARCHIVE-NODE-FILES), ist HSMS notwendige Voraussetzung.

Sicherung von Net-Storage-Dateien

Bei der Sicherung mit HSMS können Sie auf Anweisungsebene festlegen, welche Dateien gesichert werden sollen:

- Dateien auf Pubset, Net-Storage und Privatplatte
- nur Dateien auf Pubset und Privatplatte
- nur Dateien auf Net-Storage

In ARCHIVE kann dieses Verhalten nicht auf Anweisungsebene gesteuert werden. Über den Parameter STORAGE-TYPE in der Parameterdatei SYSPAR.ARCHIVE bestehen folgende Steuermöglichkeiten für die gesamte ARCHIVE-Session:

- **STORAGE-TYPE=ANY** oder keine Angabe des Parameters

Wenn in der FILES-Anweisung der Operand FROM nicht angegeben wurde, werden alle beim NAME-Operanden angegebenen Dateien gesichert, die auf Pubset, Privatplatte oder Net-Storage befinden. Von Banddateien werden nur die Katalogeinträge gesichert.

Wenn FROM=PUBLIC angegeben wurde, werden nur die Dateien gesichert, die im NAME-Operanden angegeben sind und die sich auf gemeinschaftlichen Datenträgern oder auf Net-Storage befinden.

- **STORAGE-TYPE=PUBLIC-SPACE**

Wenn in der FILES-Anweisung der Operand FROM nicht angegeben wurde, werden alle beim NAME-Operanden angegebenen Dateien gesichert, die auf Pubset oder Privatplatte befinden. Von Banddateien werden nur die Katalogeinträge gesichert.

Wenn FROM=PUBLIC angegeben wurde, werden nur die Dateien gesichert, die im NAME-Operanden angegeben sind und die sich auf gemeinschaftlichen Datenträgern befinden.

Da Dateien auf Net-Storage in beiden Fällen von der Sicherung ausgeschlossen sind, kann die Datensicherheit für diese Dateien in diesem Fall nur durch Backups des entsprechenden Dateisystems auf dem Net-Server selbst hergestellt werden.

Entsprechendes gilt für den TO-Operanden der FILES-Anweisung für Rekonstruktion:

- Wenn STORAGE-TYPE = ANY angegeben ist und der TO-Operand fehlt, werden die Dateien wieder auf ihre ursprünglichen Medien rekonstruiert. Bei TO=PUBLIC werden alle Dateien, auch Dateien von privaten Datenträgern, auf gemeinschaftliche Datenträger zurückgeschrieben. Dabei werden Dateien, die von Net-Storage gesichert wurden, wieder auf Net-Storage restauriert. Dateien auf Net-Storage zählen zu den Public-Dateien.
- Wenn STORAGE-TYPE = PUBLIC-SPACE angegeben ist, werden alle Dateien auf dem festgelegten Pubset rekonstruiert, unabhängig von dem Medium von dem sie gesichert wurden. ARCHIVE sichert die SAM-Struktur von SAM-Node-Files nicht mit, daher können diese Dateien nur mit dem Typ Node-File auf Net-Storage rekonstruiert werden. ARCHIVE gibt für jede Datei, die aus diesem Grund beim Restore auf gemeinschaftlichen Datenträger nicht verarbeitet werden konnte, eine Meldung in die Protokolldatei aus.

9 Meldungen

In diesem Kapitel finden Sie eine Übersicht über die Meldungsklassen von HSMS sowie Hinweise zu Meldungen anderer Produkte, die beim Aufruf von HSMS unverändert ausgegeben werden.

Mit dem Kommando bzw. der Standardanweisung HELP-MSG-INFORMATION können Sie weitere Informationen über die Bedeutung einzelner Meldungen abfragen.

Alle HSMS-Meldungen finden Sie auch über die HTML-Anwendung „Systemmeldungen“ auf dem Manual-Server (URL: <http://bs2manuals.ts.fujitsu.com>) und auf der DVD „BS2000 Soft-Books“.

9.1 Meldungsklassen

Meldungen gibt HSMS mit einem 7-stelligen Meldungsschlüssel aus. Die Meldungskennung ist „HSM“. Eine HSMS-Meldung hat die Form „HSM0ynn“, wobei „y“ die Meldungsklasse bezeichnet; nn ist die laufende Nummer innerhalb der Meldungsklasse.

Folgende Meldungsklassen werden in HSMS unterschieden:

Meldungsklasse	Art der Meldung
Klasse-0	Ergebnismeldung
Klasse-1	Bedienplatzmeldung
Klasse-2	Fehlermeldung mit Zurückweisung (einer Anweisung)
Klasse-4	Warn- oder Fehlermeldung (bei Anweisungsverarbeitung)
Klasse-9	Fehlermeldung mit Speicherabzug (Dump)

Meldungen mit den Meldungsklassen 0, 2 und 4 haben nur eine anweisungsspezifische Bedeutung und Auswirkung.

Meldungen mit den Meldungsklassen 1 und 9 sind nur für den HSMS-Verwalter (bzw. Operator) von Interesse, nicht aber für nicht-privilegierte Benutzer.

Die Ausgabestation für HSMS-Meldungen hängt in erster Linie von der Ablaufumgebung der betreffenden Task ab (Dialog- oder Stapelbetrieb, Benutzerauftrag, Servertask oder Maintask).

HSMS gibt die Meldungen auf folgende Ziele aus:

Auftragsart	Meldungsklasse	Ausgabeziel
Benutzerauftrag Dialog	0, 2, 4	SYSOUT
	9	Bedienplatz
Benutzerauftrag Stapel	0, 2, 4	SYSLST
	9	Bedienplatz
Servertask	4	Report
	1, 9	Bedienplatz, SYSOUT
Maintask	9	Bedienplatz

Ergebnismeldungen (Klasse-0)

Ergebnismeldungen werden ausgegeben, um einen Benutzer über das Ergebnis einer Anweisungsbearbeitung zu informieren. Im Standardfall (asynchrone Verarbeitung) ist dieses Ergebnis vorläufig.

Das Meldungsgewicht für Klasse-0-Meldungen ist standardmäßig „10“; nur bei nicht ordnungsgemäßer Beendigung der Anweisung ist das Gewicht „50“.

Bedienplatzmeldungen (Klasse-1)

Bedienplatzmeldungen werden ausgegeben, um Operator und HSMS-Verwalter über den Status von HSMS oder eine Fehlersituation in HSMS mit bekannter Ursache zu informieren. Dies sind z.B. Meldungen während des Ladens von HSMS.

Klasse-1-Meldungen werden in der Regel vom HSMS-Task oder den HSMS-Servertasks ausgegeben, und zwar am Bedienplatz mit Routing-Code „U“. Bei Servertasks erfolgt die Ausgabe zusätzlich nach SYSOUT.

Fehlermeldungen mit Zurückweisung (Klasse-2)

Fehlermeldungen der Klasse-2 werden ausgegeben, wenn HSMS eine Anweisung zurückweist wegen

- einer ungültigen Eingabe,
- fehlender Privilegierung oder
- eines nicht verfügbaren Betriebsmittels (keine Bandverarbeitungszeit).

Das Meldungsgewicht für Klasse-2-Meldungen ist standardmäßig „99“.

Meldungen bei der Verarbeitung (Klasse-4)

Meldungen der Klasse-4 werden während der Verarbeitung einer HSMS-Anweisung ausgegeben. Sie melden einen Fehler, der während der Verarbeitung auftritt oder melden den Grund, weshalb die Verarbeitung abgebrochen wird. Klasse-4-Meldungen lassen sich unterteilen:

- Warnmeldungen werden ausgegeben, wenn ein vom Benutzer angegebenes Objekt (Datei, Datenträger usw.) nicht bearbeitet werden kann (Objekt nicht vorhanden, mangelnde Privilegierung).
Nach Warnmeldungen wird die Verarbeitung grundsätzlich fortgesetzt. Das Meldungsgewicht ist standardmäßig „30“ (normal error).
- Fehlermeldungen, nach denen die Verarbeitung fortgesetzt wird, werden ausgegeben, wenn ein vom Benutzer angegebenes Objekt wegen eines aufgetretenen Fehlers nicht oder nur unvollständig bearbeitet werden kann. Das Meldungsgewicht ist standardmäßig „50“ (important error).
- Fehlermeldungen, nach denen die Verarbeitung abgebrochen wird, werden ausgegeben, wenn ein Fehler in HSMS-Basisfunktionen aufgetreten ist, der die Bearbeitung unmöglich machte. Das Meldungsgewicht ist standardmäßig „99“.

Fehlermeldungen mit Dump (Klasse-9)

Eine Fehlermeldung der Klasse-9 wird ausgegeben, wenn in HSMS ein Fehler aufgetreten ist, der nicht bekannt ist und für dessen Diagnose Fehlerunterlagen erzeugt werden.

Klasse-9-Meldungen werden am Bedienplatz mit Routing-Code „U“ ausgegeben. Bei Dialogaufträgen erfolgt die Ausgabe zusätzlich nach SYSOUT, bei Stapelaufträgen nach SYSLST.

Das Meldungsgewicht für Klasse-9-Meldungen ist standardmäßig „99“.

9.2 Meldungen anderer Produkte

Beim Aufruf von HSMS werden auch Meldungen folgender Produkte unverändert ausgegeben:

- ARCHIVE-Meldungen
ARCHIVE-Meldungen am Bedienplatz werden unverändert ausgegeben. Alle übrigen ARCHIVE-Meldungen werden nur in die Reportliste des ARCHIVE-Laufs übernommen. Die Reportliste wird bei HSMS-Anweisungen erzeugt, die ARCHIVE aufrufen.
- SDF-Meldungen
Sie werden beim Feststellen eines Syntaxfehlers durch SDF auf SYSOUT ausgegeben.
- DSSM-Meldungen
Sie werden beim Startup bzw. Shutdown von HSMS ausgegeben.

10 Abrechnungssätze

HSMS stellt ein genaues Abrechnungsverfahren zur Verfügung:

Der Benutzer bezahlt exakt das, was er für den belegten Speicherplatz, die verbrauchte CPU-Zeit und die durchgeführten Ein-/Ausgaben benötigt hat.

10.1 CPU-Zeit und Ein-/Ausgaben bei HSMS-Tätigkeiten

Bei diesem Abrechnungstyp wird zwischen dem Definitions-/Auskunfts-Betrieb (siehe auch "Definitions-/Auskunfts-Betrieb") und den Aktionsanweisungen unterschieden:

Definitions-/Auskunftsbetrieb

Für den Definitions-/Auskunftsbetrieb gibt es kein neues Abrechnungsverfahren, da die beanspruchten Betriebsmittel bereits beim gegenwärtigen BS2000-Abrechnungsverfahren berechnet werden.

Aktionsanweisungen

Für Aktionsanweisungen (Aufträge) werden besondere neue Abrechnungssätze in die Abrechnungsdatei geschrieben. Diese Abrechnungssätze informieren über den HSMS-Benutzer, die verbrauchte CPU-Zeit und die Anzahl der Ein-/Ausgaben. Zusätzlich werden Datum und Uhrzeit des Auftrags, für den die Abrechnungssätze geschrieben werden, protokolliert.

Die Abrechnungssätze werden für alle Tätigkeiten geschrieben, die einen Auftrag zur Folge haben. Im Einzelnen sind dies:

- Sichern/Restaurieren für DVS- und UFS-Läufe
- Archivieren/Restaurieren für DVS- und UFS-Läufe
- Migrieren/Zurückholen für DVS-Läufe
- Exportieren/Importieren für DVS-Läufe
- Kopieren von Sicherungsdateien für DVS- und UFS-Läufe
- Aktualisieren von Sicherungen (Anweisung UPDATE-EXPORT-SAVE-FILE) für DVS-Läufe
- Alle Verwaltungstätigkeiten, die mit einem Servertask arbeiten (Anweisungen REPAIR-CATALOG-BY-RESTORE und REPAIR-SAVE-FILE-BY-RESTORE)

Die Abrechnungsmessungen werden pro Auftrag für folgende vier Tasks getrennt durchgeführt:

- den Benutzertask
- den Servertask
- die ARCHIVE-Subtasks
- den Kommunikationstask

Zu Beginn und am Ende eines jeden Tasks wird jeweils ein Abrechnungssatz geschrieben. Er enthält die Messungen, die zu Beginn und am Ende des Tasks durchgeführt wurden. Um für einen bestimmten Task die momentan verbrauchten Betriebsmittel zu erhalten, muss die Differenz zwischen den Messungen im ersten und zweiten Abrechnungssatz ermittelt werden.

Die Anzahl der Abrechnungssätze, die pro Auftrag geschrieben werden, ist von der Art des Auftrags abhängig:

1. Für einen einzelnen „normalen“ HSMS-Auftrag werden mindestens sechs Abrechnungssätze geschrieben:
 - 2 Abrechnungssätze für den Benutzertask
 - 2 Abrechnungssätze für den Servertask
 - 2 Abrechnungssätze pro generiertem ARCHIVE-Subtask.

Die Abrechnungssätze sind durch die Benutzerkennung des Auftraggebers, die TSN und die Abrechnungsnummer des Benutzertasks gekennzeichnet.

2. Für einen Sammelauftrag werden geschrieben:

- 2 Abrechnungssätze für jeden Auftrag in den Benutzertask
- 2 Abrechnungssätze für den gesamten Sammelauftrag in den Servertask und
- 2 Abrechnungssätze pro generiertem ARCHIVE-Subtask.

Alle Aufträge werden unabhängig von einander im Benutzertask bearbeitet. Deshalb sind die Abrechnungssätze durch die Benutzerkennung des Auftraggebers, die TSN und die Abrechnungsnummer der Benutzertask gekennzeichnet.

Die Betriebsmittel, die ein Servertask verbraucht, betreffen mehrere Sammelaufträge und daher häufig auch verschiedene Benutzer. Sie werden unter der Benutzerkennung TSOS registriert. Abrechnungssätze für einen Servertask werden durch die Benutzerkennung TSOS, die TSN des Servertasks und die in den HSMS-Parametern festgelegte Abrechnungsnummer für Servertasks gekennzeichnet. Wenn diese Abrechnungsnummer nicht festgelegt ist, wird „ADMINSTR“ verwendet.

Damit die Betriebsmittel, die jeder Sammelauftrag im Servertask verbraucht hat, richtig berechnet werden können, enthalten die Abrechnungssätze des Servertask eine zusätzliche Erweiterung. Diese Erweiterung enthält die Benutzerkennungen, die Abrechnungsnummern sowie die TSNs und Abrechnungsnummern der Sammelaufträge. Die Betriebsmittel, die der Servertask verbraucht hat, werden zwischen allen Sammelaufträgen in gleiche Teile aufgeteilt und in die Abrechnungen der betroffenen Sammel-Benutzerkennungen aufgenommen. In die ARCHIVE-Subtasks werden für jede behandelte Katalog- und Benutzerkennung zwei Abrechnungssätze geschrieben.

3. Die Abrechnungssätze von Knotenaufträgen, die von einer Workstation kommen, werden durch die Benutzerkennung SYSHSMS, die TSN und die Abrechnungsnummer des Kommunikations-Subtasks gekennzeichnet.
4. Für Aufträge, die Objekte auf einem Shared-Pubset betreffen und am Master-Host bearbeitet werden, wird ein Abrechnungssatz am Slave-Host und einer am Master-Host erstellt.
5. Für implizite Rückholaufträge, die an einen Master-Host geschickt werden, wird das Auftragsdatum und die Auftragszeit nicht eingetragen, da am Slave-Host kein Auftrag erzeugt wird.
6. Bei einer RESTART-REQUESTS-Anweisung, die mehrere Aufträge erneut startet, werden folgende Abrechnungssätze geschrieben:
- nur 2 Abrechnungssätze für alle Aufträge in den Benutzertask, wobei das Auftragsdatum und die Auftragszeit nicht eingetragen werden.
 - 2 Abrechnungssätze bei erneut gestarteten Aufträgen in den Servertask
 - 2 Abrechnungssätze bei generiertem ARCHIVE-Subtask

Eine RESTART-REQUESTS-Anweisung, die nur einen einzigen Auftrag betrifft, wird wie ein einzelner „normaler“ Auftrag bearbeitet (siehe Punkt 1).

Format des HSMS-Abrechnungssatzes:

(A) Satzbeschreibung (Länge: 20 Bytes)

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	0	4	A	Kennzeichen des Abrechnungssatzes: „HSMS“
2	04	4	8	B2	Zeitstempel der TOD-Systemuhr
3	0C	12	2	B	Länge des Identifikations-Abschnitts
4	0E	14	2	B	Länge der Grundinformation
5	10	16	4	B	– reserviert –

(B) Benutzer-Kennzeichnung (Länge: 28 Bytes)

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	14	20	8	A	Benutzerkennung
2	1C	28	8	A	Abrechnungsnummer
3	24	36	4	A	Task Sequence Number (TSN)
4	28	40	8	A	Gruppenname

(C) Grundinformation (Länge: 40 Bytes)

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	30	48	4+4	B2	Task-CPU-Zeit (TU und TPR) ¹⁾
2	38	56	4	B	Anzahl der Ein-/Ausgaben ²⁾
3	3C	60	17	A	Zeitstempel des Auftrags ³⁾
4	4D	77	4	A	Kennzeichen des Tasktyps ⁴⁾
5	51	81	4	A	TSN der ablaufenden Task
6	55	85	1	A	Satzindex ⁵⁾
7	56	86	2	B	– reserviert –

Anmerkung

- 1) Die Task-CPU-Zeit enthält die von Benutzerprogrammen im TU-Zustand verbrauchte CPU-Zeit und die im TPR-Zustand verbrauchte CPU-Zeit für die Bearbeitung von SVC-Aufrufen, Programmfehlern und Kommandos. Das erste Wort enthält die vollen Sekunden, das zweite den verbleibenden Bruchteil in Nanosekunden.
- 2) Gemeint sind hier Ein- /Ausgaben auf lokale periphere Geräte, soweit sie von den DMS-Zugriffsmethoden (SAM, ISAM, UPAM, BTAM, PPAM) oder vom SYSDISK-Management unterstützt werden. Nicht mitgezählt werden die Terminal- und Seitenwechsel-Ein-/Ausgaben.
- 3) Zeitstempel des Auftrags im Format „yy-mm-dd hh-mm-ss“
- 4) Kennzeichen des Tasktyps:
 - USER für einen Benutzertask
 - SERV für einen Servertask
 - ASUB für einen ARCHIVE-Subtask
 - COMM für einen Kommunikationstask
- 5) Satzindex:
 - „A“ für den ersten Satz eines Satzpaares.
 - „B“ für den zweiten Satz eines Satzpaares.
 - Ein Satz mit dem Index „B“ kann nicht ohne einen entsprechenden Satz mit dem Index „A“ vorkommen.
 - Aber ein Satz mit dem Index „A“ kann ohne den entsprechenden Satz mit dem Index „B“ vorkommen. Dieser Satz muss dann ignoriert werden.

(D) Variable Information (Länge: 8 Bytes)

Die variable Information des HSMS-Abrechnungssatzes enthält eine Satzerweiterung.

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	58	88	2	B	Anzahl der Erweiterungen ¹
2	5A	90	2	B	Distanz zur ersten Erweiterung
3	5C	92	2	B	Distanz zur zweiten Erweiterung
4	5E	94	2	B	Distanz zur dritten Erweiterung

¹ Die Anzahl der Erweiterungen ist immer 3. Falls die dritte Erweiterung nicht vorhanden ist, hat die Distanz zur dritten Erweiterung den Wert 0.

1. Erweiterung (max. Länge: 12 Bytes)

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	00	2	A	Erweiterungskennung "ID"
2	02	02	1	B	X'00'
3	03	03	1	B	Länge der Abrechnungskennung
4	04	04	L ¹	C/X	Abrechnungskennung

- 1 Die Abrechnungskennung kann maximal 8 Bytes lang sein. Sie stimmt mit der maximal 8 Byte langen Benutzerkennung übereinoder mit dem Makro AREC (Operand ID) eingegeben hat. Falls für die Durchführung eines Auftrags keine Benutzerkennung angegeben wurde, wird dieses Feld mit X'FFFFFFFFFFFFFFFF' gefüllt.

2. Erweiterung (Länge: 24 Bytes)

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	00	2	A	Erweiterungskennung "IO"
2	02	02	1	B	Anzahl der Elemente (X'01')
3	03	03	1	B	Länge des Elements (X'14')
4	04	04	4	B	Anzahl der Ein-/Ausgaben pro Gerät für Pubsets
5	08	8	4	B	Anzahl der Ein-/Ausgaben pro Gerät für gemeinsam benutzbare Privatplatten
6	EC	12	4	B	Anzahl der Ein-/Ausgaben pro Gerät für exklusive Privatplatten
7	10	16	4	B	Anzahl der Ein-/Ausgaben pro Gerät für Magnetbandkassetten
8	14	20	4	B	Anzahl der Ein-/Ausgaben pro Gerätegruppe für Unit-Record-Geräte

3. Erweiterung

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	00	2	A	Erweiterungskennung "CO"
2	02	02	1	B	Anzahl der Elemente (X'xx')
3	03	03	1	B	Länge des Elements (X'20')
4	04	04	8	A	Benutzerkennung
5	0C	12	8	A	Abrechnungsnummer
6	14	20	4	A	Task Sequence Number (TSN)
7	18	24	4	A	Länge der Abrechnungsnummer der Sammelaufträge
8	1C	28	8	A	Abrechnungsnummer der Sammelaufträge
$4 + x * 32 \text{ Bytes}$ (wobei x die Anzahl der Elemente ist)					

In der 3. Erweiterung ist die Anzahl der Elemente variabel (X'xx').

Die maximale Länge eines Abrechnungssatzes mit der „IO“-Erweiterung, die immer vorhanden ist, beträgt 132 Bytes.

Die minimale Länge eines Abrechnungssatzes mit der „CO“-Erweiterung (nur für Sammelaufträge vorhanden) beträgt 132 Bytes + 36 Bytes (vollständige Erweiterung: ID, Länge, Information über den 1. Sammelbenutzer) + 32 Bytes (nur Information über den nächsten Sammelbenutzer).

Da ein Abrechnungssatz maximal 496 Bytes lang sein kann, können nur Informationen über höchstens 11 Sammelbenutzer ausgegeben werden. Falls noch mehr Sammelbenutzer in diesem Auftrag vorhanden sind, werden diese im Abrechnungssatz ignoriert.

10.2 Belegter Speicherplatz (Speicherebene S1 und S2)

Der Speicherplatz-Bestandsaufnahme-Abrechnungssatz (Satzkennung DSPC; vgl. auch im Handbuch „Abrechnungssätze“ [28]) enthält für einen Pubset pro Benutzerkennung:

- Anzahl belegter PAM-Blöcke auf S0
- Anzahl der durch Migration belegten PAM-Blöcke auf S1
- Anzahl der durch Migration belegten PAM-Blöcke auf S2

11 Anhang

Die nachfolgenden Abschnitte „Einführung von HSMS im Rechenzentrum“, „Auswirkungen auf Benutzer-Anwendungen“ und „Hinweise zur Erstellung automatischer Verfahren“ sind vom Vervielfältigungsverbot ausgenommen. Der System- bzw. HSMS-Verwalter darf sie kopieren und an die betroffenen Benutzer in einem Rechenzentrum vor der Einführung von HSMS verteilen. Im Wesentlichen handelt es sich um Auszüge aus dem vorliegenden Handbuch.

Zusätzlich enthält der Anhang Hinweise zur Fehler-Diagnose (siehe [Abschnitt „Diagnosehilfen und Trace-Funktionen“](#)).

11.1 Einführung von HSMS im Rechenzentrum

HSMS (**H**ierarchisches **S**peicher **M**anagement **S**ystem – Hierarchical Storage Management System) ist ein Softwareprodukt

- zur Organisation von Datensicherung und Archivierung
- zur Nutzungsoptimierung von externen Schnellspeichern in einem BS2000-System
- zum Datentransfer
- zum Versions-Backup

HSMS unterscheidet die drei Speicherebenen S0, S1 und S2, die unter diesem Kürzel auch in HSMS-Anweisungen und BS2000-Kommandos angesprochen werden.

Übersicht über die HSMS-Speicherebenen:

Speicherebene	HSMS-Nutzung	Datenträger	Zugriff	Zugriffszeit
S0 Verarbeitungsebene	Archivverwaltung	Platten	online	sehr kurz
S1 Hintergrundebene	Langzeitarchivierung, Datensicherung, Versions-Backup und Verdrängung	Platten	online	kurz
S2 Archivebene	Langzeitarchivierung, Datensicherung, Versions-Backup und Verdrängung	Magnetbandkassetten	online	lang

Der Einsatz vom HSMS steht auf ihrer Anlage unmittelbar bevor.

Im Folgenden sind die Auswirkungen der Nutzungsoptimierung durch die sog. Verdrängung (Migration) ausführlich dargestellt, weil davon auch alle Nicht-HSMS-Benutzer betroffen sein werden.

Weitergehende Informationen zur Datensicherung, Archivierung und zum Datentransfer finden Sie im Handbuch „HSMS Band 1: Funktionen, Verwaltung und Installation“.

11.1.1 Motivation zum Einsatz der Migrationsfunktion

HSMS bietet mit der Verdrängung eine Nutzungsoptimierung für schnelle Plattenspeicher an. Vor allem bei häufig wechselnden Anwendungen ist es sinnvoll, gerade nicht benötigte (inaktive) Daten von der Verarbeitungsebene auf eine Hintergrundebene zu verdrängen; dabei bleiben die Daten jedoch über ihren Katalogeintrag ansprechbar. Dadurch lassen sich Sättigungszustände auf gemeinschaftlichen Datenträgern (pubspace-saturation) oder das Erreichen des Pub-Space-Limits für Benutzerkennungen vermeiden, ohne dass die Verfügbarkeit der Dateien merklich eingeschränkt wird.

Für verdrängte Dateien erzeugt das Datenverwaltungssystem (DVS) bei Zugriffsversuchen HSMS-Aufträge: Die Dateien werden ohne Eingriff des Benutzers auf die Verarbeitungsebene zurückgeholt. Teure, schnelle Externspeicher lassen sich durch den Verdrängungsmechanismus wirtschaftlicher einsetzen, ohne den personellen Aufwand zu erhöhen. Da nur die tatsächlich belegten Seiten übertragen werden, wird auf der Hintergrundebene gegenüber der Verarbeitungsebene auch Speicherplatz eingespart.

11.1.2 Verdrängte Dateien

Nach der Verdrängung gilt eine Datei als verdrängt (migriert): Die Datei besitzt auf der Verarbeitungsebene einen Katalogeintrag, beansprucht aber dort keinen Speicherplatz. Im Katalogeintrag wird vermerkt, auf welche Hintergrundebene (S1 oder S2) die Daten dieser Datei verdrängt wurden.

Als SUPPORT wird, je nach Hintergrundebene, PUB/S1 oder PUB/S2 ausgegeben. Die Angaben für Dateigröße und Last-Page-Pointer ändern sich durch die Verdrängung nicht.

Eine Übersicht über verdrängte Dateien erhalten Sie mit folgendem Kommando:

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=*ALL, -
/ SELECT=*BY-ATTR(STORAGE-LEVEL=( *S1, *S2 ))
```

Beispiel für Ausgaben von verdrängten Dateien

```
/SHOW-FILE-ATTRIBUTES FILE-NAME=TEST.HSMS, INFORMATION=*ALL-ATTRIBUTES
%00000543#:10SN:$MANUALE.TEST.HSMS
% ----- HISTORY -----
% CRE-DATE   = 2016-01-29  ACC-DATE   = 2016-02-05  CHANG-DATE = 2016-01-29
% CRE-TIME   =   15:04:05  ACC-TIME   =   15:58:22  CHANG-TIME =   15:04:20
% ACC-COUNT  = 2          S-ALLO-NUM = 61
% ----- SECURITY -----
% READ-PASS  = NONE       WRITE-PASS  = NONE       EXEC-PASS  = NONE
% USER-ACC   = ALL-USERS  ACCESS      = WRITE      ACL         = NO
% AUDIT      = NONE       FREE-DEL-D  = *NONE       EXPIR-DATE = 2016-01-29
% DESTROY    = NO        FREE-DEL-T  = *NONE       EXPIR-TIME =   00:00:00
% SP-REL-LOCK= NO        ENCRYPTION  = *NONE
% ----- BACKUP -----
% BACK-CLASS = A          SAVED-PAG  = COMPL-FILE  VERSION    = 1
% MIGRATE    = ALLOWED    STOR-LEVEL = S2
% #BACK-VERS = 2
% ----- ORGANIZATION -----
% FILE-STRUC = ISAM      BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
% IO(USAGE)  = READ-WRITE IO(PERF)   = STD          DISK-WRITE = IMMEDIATE
% REC-FORM   = (V,N)     REC-SIZE   = 0
% AVAIL      = *STD
% WORK-FILE  = *NO       F-PREFORM  = *K          S0-MIGR    = *ALLOWED
% ----- ALLOCATION -----
% SUPPORT    = PUB/S2    S-ALLOC   = 16          HIGH-US-PA = 543
% EXTENTS    VOLUME     DEVICE-TYPE
% NONE       NONE       NONE
%:10SN: PUB/S2:      1 FILE RES=      543 FRE=      0 REL=      0 PAGES

/SHOW-FILE-ATTRIBUTES FILE-NAME=TEST.
%      543#:10SN:$MANUALE.TEST.HSMS
%      3 :10SN:$MANUALE.TEST.MSG
%      3#:10SN:$MANUALE.TEST.VOLX
%:10SN: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
%:10SN: PUB/S2:      2 FILES RES=     546 FRE=      2 REL=      0 PAGES
```

Bei Übersichten sind verdrängte Dateien durch das Nummernzeichen (#) zwischen der Dateigrößenangabe und der Katalogkennung gekennzeichnet.

Während der Zeit ihrer Verdrängung gilt für Zugriffe auf diese Datei und ihren Katalogeintrag:

- Veränderungen der Dateiattribute, wie z.B. der Schutzattribute, Kennwörter etc., sind möglich und bleiben auch nach dem Zurückholen wirksam.
- Die Datei darf allerdings nicht umbenannt werden. Ein entsprechender Versuch wird mit einer Fehlermeldung zurückgewiesen.
- Die Dateigröße (Operand SPACE bei MODIFY-FILE-ATTRIBUTES) können Sie verändern. Die Änderung wird allerdings erst nach dem Zurückholen wirksam und sichtbar.
- Die Datei können Sie jederzeit löschen. Eine verdrängte Datei, deren Katalogeintrag gelöscht wurde, ist für HSMS ungültig. Denn ohne Katalogeintrag kann sie nicht mehr zurückgeholt werden.
- Das Überschreiben der Datei (OPEN OUTPUT, OUTIN oder UPDATE) auf der Verarbeitungsebene ist zugelassen. Danach gilt die Datei nicht mehr als verdrängt und ist im Rahmen des Migrationsarchivs ebenfalls ungültig.

11.1.3 Verdrängte Dateien zurückholen

Der zur Verdrängung umgekehrte Vorgang, nämlich das Kopieren der verdrängten Daten auf die Verarbeitungsebene, heißt Zurückholen.

Es können nur so viele Dateien auf S0 eingespielt werden, wie auf den jeweiligen Benutzerkennungen Platz haben.

Zurückholen kann man nur Dateien, die als verdrängt katalogisiert sind, und nur den zum Katalogeintrag (CFID und Version) passenden Stand im Migrationsarchiv. Der aktuelle Stand einer Datei auf der Verarbeitungsebene wird durch das Zurückholen nicht verändert.

Der nicht-privilegierte Benutzer kann die Dateien unter der eigenen Benutzerkennung zurückholen. Dateien einer anderen Benutzerkennung kann er nur zurückholen, wenn er Miteigentümer der Datei ist oder die Dateizugriffsrechte den Zugriff anderer Benutzerkennungen erlauben.

Insbesondere beim Zurückholen einer migrierten Datei einer anderen Benutzerkennung sollte berücksichtigt werden, dass sich dadurch auch die Speicherplatzbelegung dieser Benutzerkennung entsprechend erhöht.

Achtung

Beim Zurückholen einer Datei ändern sich folgende Dateiattribute: interner Dateiname (CFID), DEVICE-TYPE, SUPPORT, VOLUME und EXTENTS. Anwendungen, die diese Dateiattribute auswerten, müssen dies berücksichtigen.

Für das Zurückholen bietet HSMS zwei unterschiedliche Möglichkeiten:

- Zurückholen durch HSMS-Aufruf (Explizites Zurückholen)
- Zurückholen beim Zugriff (Implizites Zurückholen)

11.1.3.1 Explizites Zurückholen durch HSMS-Aufruf

Jeder Benutzer kann die Daten einer oder mehrerer verdrängter Dateien auf die Verarbeitungsebene S0 zurückholen, indem er das Programm HSMS aufruft und die HSMS-Anweisung RECALL-MIGRATED-FILES absetzt. Die genaue Beschreibung dieser HSMS-Anweisung sowie deren Operanden finden Sie im Handbuch „HSMS Bd. 2“ [\[1\]](#).

11.1.3.2 Implizites Zurückholen durch Zugriff

Der Benutzer kann eine verdrängte Datei auch ohne expliziten HSMS-Aufruf zurückholen.

Zu Beginn eines jeden Zugriffs kann das Datenverwaltungssystem (DVS) des BS2000 anhand des Katalogeintrags feststellen, dass die zu dieser Datei gehörigen Daten nicht auf der Verarbeitungsebene verfügbar sind.

Wurde festgestellt, dass die Datei verdrängt ist, und will der Benutzer bzw. ein Benutzerprogramm lesend auf die Datei zugreifen (OPEN-Modus INPUT, INOUT, aber auch UPDATE, REVERSE und EXTEND), so werden die Daten der Datei automatisch (implizit) auf die Verarbeitungsebene zurückgeholt.

Will ein Benutzer nur schreibend auf die Datei zugreifen (OPEN-Modus OUTPUT oder OUTIN), so sollen alle Daten der Datei überschrieben werden, und ein vorheriges Zurückholen ist nicht nötig.

Im Falle eines solchen sog. impliziten Recalls werden die Daten der Datei aus einer Sicherungsdatei auf der Speicherebene zurückgeholt, die im Katalog eingetragen ist.

Etwas anders ist die Behandlung bei Dateizugriffen mit dem Kommando SECURE-RESOURCE-ALLOCATION. Mit diesem BS2000-Kommando können Sie den bevorstehenden Zugriff auf eine oder mehrere Dateien ankündigen. Wie auch im Zusammenhang mit Geräten und exklusiv belegten Datenträgern (z.B. Magnetbandkassetten) dient auch in diesem Fall das Kommando dazu, alle benötigten Ressourcen (hier: Dateien und deren Daten) verfügbar zu machen, d.h. falls nötig einen Rückholvorgang einzuleiten.

11.2 Auswirkungen auf Benutzer-Anwendungen

- Zurückholen beim Dateizugriff
- Zurückholen durch SECURE-RESOURCE-ALLOCATION
- Schutz der Dateien vor Verdrängung

11.2.1 Zurückholen beim Dateizugriff

Folgende Auswirkungen ergeben sich beim Zurückholen verdrängter Dateien:

- Der Rückholvorgang wird implizit beim Eröffnen der Datei angestoßen. Dies kann prinzipiell für jede Datei geschehen.
- Der Rückholvorgang benötigt eine gewisse Laufzeit, abhängig von der aktuellen Anlagen- und Gerätebelastung und von der Art des Hintergrundspeichers (Platten bei S1, Magnetbandkassetten bei S2).
- Die Verarbeitung innerhalb einer Benutzeranwendung kann somit an mehreren Stellen (jeweils während des Eröffnens einer Datei) unterbrochen, d.h. verzögert werden.
- Je Rückholauftrag erfolgt ein eigener Zugriff auf den Hintergrundspeicher. Insbesondere bei Verdrängung auf S2 kann durch implizite Rückholvorgänge die Montagerate erheblich erhöht werden, was i.a. auch die Laufzeit entsprechend erhöht.

11.2.2 Zurückholen durch SECURE-RESOURCE-ALLOCATION

Wenn der Rückholvorgang mit dem Kommando SECURE-RESOURCE-ALLOCATION gesteuert wird, ergeben sich folgende Auswirkungen:

- Ein implizites Zurückholen wird nur angestoßen, wenn der Benutzer zu warten bereit ist (Operand WAIT=(TIME=wartezeit)). Standardwert für Dialogtasks ist: Kein Warten! Diese Zeit ist hinreichend lang zu dimensionieren, um auch Bandzugriffe zu ermöglichen (z.B. 2 Minuten).
- Sämtliche Dateien (bis zu 48), die für einen Job benötigt werden, können mit einem Kommando angefordert und mit einem Auftrag zurückgeholt werden. Dadurch sind weniger HSMS-Aufrufe und evtl. weniger Bandanforderungen nötig.
- Es existiert ein definierter Wartepunkt, nachdem die Verarbeitung entweder starten kann oder abgebrochen werden muss. So lassen sich „unliebsame Überraschungen“ während der Verarbeitung durch fehlende Dateien vermeiden.
- Für derartig angestoßene Zurückholaufträge gibt HSMS einen zusammenfassenden Report aus, der sonst bei implizitem Zurückholen nur im Fehlerfall ausgegeben (ausgedruckt) wird.

Allgemein lassen sich Bandzugriffe durch Bandverarbeitungszeiten, die der Systemverwalter definiert hat, koordinieren. Wenn derartige Einschränkungen beim Zugriff auf eine verdrängte Datei gemeldet werden, wenden Sie sich bitte an den zuständigen Systemverwalter.

11.2.3 Schutz der Dateien vor Verdrängung

Eine Datei kann auf folgende Arten vor der Verdrängung geschützt werden:

- Dateiattribut „MIGRATE“:
Die Migrationssperre wird gesetzt mit `MIGRATE=*INHIBITED` im Kommando `CREATE-FILE` bzw. `MODIFY-FILE-ATTRIBUTES` (Aufhebung mit `MIGRATE=*ALLOWED`).
Der Systemverwalter kann diesen Schutz umgehen, d.h. die Datei ist nur vor „Standard-Migration“ gesichert. Entsprechend sollte dieser Schutz nur selektiv eingesetzt werden, um allein durch Migration der nicht geschützten Dateien den Plattenspeicher entlasten zu können und so den Systemverwalter nicht zur Migration dieser geschützten Dateien zu „zwingen“.
Für Benutzerkennungen mit dem Recht zur physikalischen Allokierung gibt es zusätzlich noch die Option `MIGRATE=*FORBIDDEN`. Dadurch wird die Migration absolut verboten.
- Der Systemverwalter kann eine Liste besonders schützenswerter Dateien führen, in der er auch Benutzerdateien eintragen kann. Neben dem Setzen des Datei-Attributes `MIGRATE=*FORBIDDEN` schützt nur der Eintrag in diese Liste die betreffende Datei vor Migration.
- Alle Dateien, die während des Migrationsauftrages geöffnet sind, werden ebenfalls nicht migriert. Dies kann aber im Allgemeinen dadurch vermieden werden, dass der Migrationsauftrag außerhalb der normalen Betriebszeiten abgearbeitet wird.

11.3 Hinweise zur Erstellung automatischer Verfahren

- Empfohlene Umstellungen
- Zwingend nachzuprüfendes Verhalten
- Unkritische Behandlung verdrängter Dateien

11.3.1 Empfohlene Umstellungen

Die im Folgenden dargestellten Aspekte dienen nur der Laufzeit-Optimierung der Anwendungen und können entsprechend ignoriert werden. Zur Reduktion der Gesamtlast auf dem Rechner wird aber das dargestellte Verfahren empfohlen.

- Statt durch die Kommandos SHOW-FILE-ATTRIBUTES bzw. ADD-FILE-LINK das Vorhandensein einer Datei zu prüfen bzw. sicherzustellen, sollte das Kommando SECURE-RESOURCE-ALLOCATION verwendet werden.
- Vor Beginn der Verarbeitung sollte möglichst für alle nachfolgend zum Lesen zu öffnenden Dateien sichergestellt werden, dass mit dem Kommando SECURE-RESOURCE-ALLOCATION (*mit* Angabe einer Wartezeit) deren Daten auf der Verarbeitungsebene bereitstehen. Als Nebeneffekt werden so mehrere Dateien mit einem gemeinsamen Auftrag zurückgeholt, was (s.o.) die Anlagenlast reduzieren kann.
- Wenn absehbar ist, dass beim Löschen einer Datei deren Daten überschrieben werden sollen (Operand DESTROY), so sollte dies nicht erst beim Löschen der Datei (Kommando DELETE-FILE) angegeben sondern als Attribut der Datei definiert werden (Kommando MODIFY-FILE-ATTRIBUTES). Andernfalls erfolgt das Löschen der Daten im Rahmen des Verdrängungsvorganges *ohne* Überschreiben der Daten.
- Bei Änderung des zeitlich befristeten Dateischutzes (Retention Period) einer auf S2 verdrängten Datei kann nicht sichergestellt werden, dass die Daten der Datei während dieses Intervalls verfügbar sind. Ein vorhergehendes Zurückholen wird empfohlen.

11.3.2 Zwingend nachzuprüfendes Verhalten

Die nachfolgend dargestellten Aspekte sind unbedingt zu beachten, um den sicheren Ablauf der Anwendungen zu gewährleisten.

- Wenn nicht sichergestellt werden kann, dass die Datei durch ein vorhergehendes /SECURE-RESOURCE-ALLOCATION zurückgeholt wurde, darf der Inhalt des Katalogeintrages einer Datei *vor* deren Eröffnung *nicht* mit dem *nach* der Eröffnung verglichen werden. Dies betrifft die Attribute:
 - Interner Dateiname (CFID)
 - Lokalitätsbeschreibung (VOLUME, Extent-Count und Extent-List)
- Anwendungen, die den Internen Dateinamen (CFID) zur Prüfung und Bearbeitung von Dateien (z.B. mit UPAM) benutzen, dürfen die CFID erst nach dem evtl. erfolgten Rückholvorgang ermitteln.
- Verdrängte Dateien dürfen nicht umbenannt werden. Entsprechende Aktionen werden mit Fehlermeldungen zurückgewiesen.

Wenn eine Anwendung Dateien umbenennen will, die vorher nicht bearbeitet wurden, so muss z.B. mit dem Kommando SECURE-RESOURCE-ALLOCATION sichergestellt werden, dass die Dateien nicht verdrängt sind.

11.3.3 Unkritische Behandlung verdrängter Dateien

- Dateien können unabhängig von ihrem Speicherort (S0, S1 oder S2) gelöscht werden. Bei verdrängten Dateien wird lediglich der Katalogeintrag gelöscht.
- Die Dateiattribute von verdrängten Dateien, wie z.B. die Schutzattribute, Kennwörter etc. können geändert werden. Die Änderungen werden sofort wirksam und bleiben auch nach dem Zurückholen der Daten erhalten.
- Wenn die Größe einer verdrängten Datei geändert wird (Operand SPACE im Kommando MODIFY-FILE-ATTRIBUTES), wird zunächst nur der Katalogeintrag geändert. Erst nach dem Zurückholen der Daten auf die Verarbeitungsebene beeinflusst dies den Speicherplatz.

11.4 Diagnosehilfen und Trace-Funktionen

Die nachfolgend genannten Hilfsmittel dienen der Diagnose von bestimmten Problemen im Kundenbetrieb.

Sowohl Auswahl als auch Einsatz eines Hilfsmittels sollte immer in Absprache mit den zuständigen Service-Mitarbeitern erfolgen.

11.4.1 HSMS-Trace

Der Einsatz ist sinnvoll, wenn Ablaufprobleme bei Aufträgen in HSMS vermutet werden.

Der HSMS-Trace wird mit den privilegierten HSMS-Anweisungen START-TRACE und STOP-TRACE ein- bzw. ausgeschaltet.

START-TRACE	<i>startet den HSMS-Trace</i>
DOMAIN = *ALL / list-poss(32): *REPOSITORY / *MESSAGES / *SHARE-PVS / *STATEMENTS	
,TSN = *ALL / <alphanum-name 1..4>	
STOP-TRACE	<i>beendet den HSMS-Trace</i>
CLOSE-MODE = *NORMAL-CLOSE / *PSEUDO-CLOSE	

Der Operand DOMAIN bestimmt, ob der Trace alle oder mehrere bestimmte Funktionsbereiche erfassen soll:

- Der Operandenwert *ALL veranlasst Trace-Einträge für alle nachfolgend genannten Funktionsbereiche.
- Der Operandenwert *REPOSITORY erzeugt bisher keinerlei Trace-Einträge.
- Der Operandenwert *MESSAGES erzeugt Trace-Einträge für alle Meldungsausgaben von HSMS (Parameterliste von MSG7) durch den Modul DHSBMSG.
- Der Operandenwert *SHARE-PVS erzeugt Trace-Einträge für die im Shared-Pubset-Betrieb verschickten HSMS-Aufträge (Internal HSMS Commands) durch den Modul DHSBCIH.
- Der Operandenwert *STATEMENTS erzeugt Trace-Einträge für alle eingegebenen HSMS-Anweisung durch den Modul DHSEVAL.

Mit dem Ausschalten des Traces werden die Diagnose-Informationen in eine Datei geschrieben. Der Operand CLOSE-MODE=*PSEUDO-CLOSE bewirkt, dass der Trace weiterläuft, aber die bis zu diesem Zeitpunkt geschriebenen Daten gelesen werden können.

Name der Trace-Datei: \$SYSHSMS.HSM.Y.TRACE.<yyyymmdd>.<hhmmss>

Der Zustand des Traces wird mit der HSMS-Anweisung SHOW-LOGGING-STATUS ausgegeben.

Der Inhalt der Trace-Datei kann mit der HSMS-Anweisung LIST-LOGGING-FILE aufbereitet und nach SYSLST geschrieben werden.

11.4.2 ARCHIVE-Trace

Der Einsatz ist sinnvoll, wenn Ablaufprobleme bei Aufträgen in ARCHIVE vermutet werden.

Der ARCHIVE-Trace wird mit den privilegierten Kommandos START-ARCHIVE-TRACE und STOP-ARCHIVE-TRACE ein- bzw. ausgeschaltet. Der Trace kann auch nur für bestimmte Funktionseinheiten bzw. Tasks gestartet werden.

START-ARCHIVE-TRACE	<i>startet den ARCHIVE-Trace</i>
TSN = <u>*ALL</u> / <alphanum-name 1..4> ,DOMAIN = <u>*ALL</u> / list-poss(32): *DEVICE / *MULTIPLEXING / *PACKET / *SYNCRONIZATION / *PLAM / *MEMORY / *DIRECTORY	
STOP-ARCHIVE-TRACE	<i>beendet den ARCHIVE-Trace</i>
CLOSE-MODE = <u>*NORMAL-CLOSE</u> / *PSEUDO-CLOSE	
MODIFY-ARCHIVE-TRACE	<i>ändert den laufenden ARCHIVE-Trace</i>
TSN = <u>*ALL</u> / <alphanum-name 1..4> ,DOMAIN = <u>*ALL</u> / list-poss(32): *DEVICE / *MULTIPLEXING / *PACKET / *SYNCRONIZATION / *PLAM / *MEMORY / *DIRECTORY	

Der Operand DOMAIN bestimmt, ob der Trace alle oder nur (mehrere) bestimmte Funktionsbereiche erfassen soll:

- Der Operandenwert *ALL veranlasst Trace-Einträge für alle nachfolgend genannten Funktionsbereiche.
- Der Operandenwert *DEVICE veranlasst Trace-Einträge für alle Volume- und Bandgeräte-Aufrufe an MAREN und NDM in den Modulen FAPO, FASC, FASR und FAUTDM.
- Der Operandenwert *MULTIPLEXING veranlasst Trace-Einträge für alle Aufrufe zwischen Bandtreiber- und Multiplexing-Subtask.
- Der Operandenwert *PACKET veranlasst Trace-Einträge zur Paketverarbeitung in den Modulen FAFN, FAFNSV und FARPX.
- Der Operandenwert *SYNCHRONISATION veranlasst Trace-Einträge zur Synchronisation zwischen ARCHIVE-Maintask und Subtasks Modul FASC.
- Der Operandenwert *PLAM veranlasst Trace-Einträge zum Schreiben und Lesen der PLAM-Info im Modul FAFNPL.
- Der Operandenwert *MEMORY veranlasst Trace-Einträge zu Speicheranforderungen bzw. Speicherfreigaben im Modul FAUTMR.
- Der Operandenwert *DIRECTORY veranlasst Trace-Einträge zu Zugriffen auf Verzeichnisse in den Modulen FARAM und FAUTRM.

Beim Starten bzw. Stoppen des ARCHIVE-Traces wird die zugehörige Trace-ID ausgegeben.

Mit dem Ausschalten des Traces werden die Diagnose-Informationen in eine Datei geschrieben und der zugehörige Dateiname wird ausgegeben. Der Operand `CLOSE-MODE=*PSEUDO-CLOSE` bewirkt, dass der Trace weiterläuft, aber die bis zu diesem Zeitpunkt geschriebenen Daten gelesen werden können.

11.4.3 Performance-Statistik

Im Normalfall liefern die Meldungen ARC0815/ARC0825 und ARC0816/ARC0826 Informationen über die Performance einer Sicherung oder eines Restore. Ausgegeben werden die Größe und Anzahl der übertragenen Dateien und Jobvariablen sowie die dafür benötigte Zeit. Bei der Zeiterfassung wird die Wartezeit während der Bandmontage nicht berücksichtigt. Der Zeitgeber läuft nur, während die Sicherungsdatei ARCHIVE.SAVE.FILE geöffnet ist.

Für jeden ARCHIVE-Subtask wird eine eigene Meldung ausgegeben. Beispiele:

```
ARC0815 SUBTASK '0' HAS TRANSFERRED '10' PAM PAGES FOR '1' FILES AND '3' JVS  
IN '4' SECONDS  
ARC0816 SUBTASK '1' HAS TRANSFERRED '27' MEGABYTES FOR '1305' FILES IN '324'  
SECONDS
```

Wenn die Sicherungszeiten unerwartet hoch sind oder Probleme bei einzelnen Dateien oder beim Bandübergang vermutet werden, kann eine zusätzliche Performance-Statistik bei der Problemdiagnose helfen.

Angefordert wird diese Performance-Statistik mit dem Operanden PERFORMANCE-ANALYSIS=*YES der entsprechenden Anweisung zum Sichern bzw. Restaurieren.

Die Performance-Statistik wird pro Subtask in eine Datei unter der Benutzerkennung SYSHSMS unter folgendem Namen abgelegt:

```
HSM.S.SYSHSMS<user-id><year><julian date><hour><min><secs>.STATS.<subtask-nr>
```

Der Namensbestandteil ist nur bei nicht-privilegierten Benutzern enthalten. Der nicht-privilegierte Benutzer kann die Erzeugung dieser Datei unter der Benutzerkennung SYSHSMS zwar veranlassen, aber es sind keine Zugriffsrechte für ihn gesetzt.

In folgenden Fällen wird eine neue Informationszeile in die Statistikdatei geschrieben:

- beim Öffnen einer Sicherungsdatei ARCHIVE.SAVE.FILE
- beim Schließen der Sicherungsdatei
- bei Beginn des Sicherns oder Restaurierens einer neuen Datei
- alle 2 Sekunden während des Sicherns bzw. Restaurierens

Die Performance-Statistik gibt somit einen detaillierten Überblick über alle Aktivitäten einer Subtask, u.a. kann pro Datei die zum Sichern bzw. Restaurieren benötigte Zeit ermittelt werden. Um eine programmtechnische Auswertung der Statistikdaten zu erleichtern, sind die einzelnen Felder mit einem Trennzeichen, das bei Anforderung der Statistikdaten frei vereinbart werden kann (Voreinstellung ist Semikolon), von einander getrennt. Der Aufbau einer Informationszeile ist in der folgenden Tabelle beschrieben.

Feld	Position	Länge	Wert	Bedeutung
Zeitstempel	1	19		YYYY-MM-DD HH:MM:SS
Separator 1	20	1	;	Trennzeichen
Anzahl der Dateien	21	11		Anzahl der Dateien/Pfade, die bis jetzt bearbeitet wurden
Separator 2	32	1	;	Trennzeichen
Anzahl der Jobvariablen	33	11		Anzahl der Jobvariablen, die bis jetzt bearbeitet wurden (immer 0 bei einem UFS-Dateisystem)
Separator 3	44	1	;	Trennzeichen
Anzahl der intern übertragenen PAM-Seiten oder MBytes	45	20		PAM-Seiten bei DMS-Dateien, MBytes bei UFS; Transfer von/zur Sicherungsdatei (interner Transfer)
Separator 4	65	1	;	Trennzeichen
Anzahl der extern übertragenen xBytes; x= K / M / G	66	11		KBytes/MBytes/GBytes, die mit TCP/IP über das Netz übertragen wurden (externerTransfer)
Separator 5	77	1	;	Trennzeichen
Einheit für xBytes	78	1		'1' = K; '2' = M; '3' = G
Separator 6	79	1	;	Trennzeichen
Name der Workstation	80	48		Name der aktuellen Workstation; leer, wenn DMS-Datei oder *BS2-UFS
Separator 7	128	1	;	Trennzeichen

Obwohl die Ausgabe der Performance-Statistik nur eine geringe Lasterhöhung verursacht, sollte sie im Normalbetrieb nicht ständig eingesetzt werden.

Bei folgenden Anweisungen kann die Performance-Statistik angefordert werden (siehe Handbuch „HSMS Anweisungen“ [1]):

Anweisung	Funktion
ARCHIVE-FILES	Dateien und Jobvariablen archivieren
ARCHIVE-NODE-FILES	Knotendateien archivieren
BACKUP-FILES	Dateien und Jobvariablen sichern
BACKUP-NODE-FILES	Knotendateien sichern
BACKUP-FILE-VERSIONS	Dateiversionen sichern
COPY-EXPORT-SAVE-FILE	Sicherungsdatei kopieren
COPY-NODE-SAVE-FILE	Knoten-Sicherungsdatei kopieren
COPY-SAVE-FILE	Sicherungsdatei kopieren
EXPORT-FILES	Dateien und Jobvariablen exportieren
IMPORT-FILES	Dateien und Jobvariablen importieren
MIGRATE-FILES	Dateien verdrängen
MOVE-SAVE-FILES	Sicherungs- bzw. Knoten-Sicherungsdateien verlagern
RECALL-MIGRATED-FILES	Verdrängte Dateien zurückholen
REORGANIZE-VERSION-BACKUP	Versions-Backup-Archive reorganisieren
REPAIR-CATALOG-BY-EXCHANGE	Inkonsistenzen bei migrierten Dateien durch EXCHANGE beheben
REPAIR-CATALOG-BY-RESTORE	Inkonsistenzen bei migrierten Dateien durch RESTORE aufheben
REPLACE-SAVE-FILE-BY-EXCHANGE	Sicherungsdatei im Migrationsarchiv durch EXCHANGE ersetzen
REPLACE-SAVE-FILE-BY-RESTORE	Sicherungsdatei im Migrationsarchiv durch RESTORE ersetzen
RESTORE-FILES	Dateien und Jobvariablen restaurieren
RESTORE-LIBRARY-ELEMENTS	Elemente einer Bibliotheksdatei restaurieren
RESTORE-NODE-FILES	Knotendateien restaurieren
UPDATE-EXPORT-SAVE-FILE	Sicherung aktualisieren

Tabelle 8: HSMS-Anweisungen, die Performance-Statistik unterstützen

11.4.4 Weitere Hinweise zur Diagnose

Wenn bei Störungen des Bandbetriebs eine Problemanalyse über ARCHIVE hinaus notwendig wird, sollte in Absprache mit den zuständigen Service-Mitarbeitern ein NDM-Trace erstellt werden. Die Erstellung des NDM-Traces ist im „Diagnose-Handbuch“ [\[27\]](#) beschrieben.

12 Fachwörter

Aktionsanweisung

Anweisung an HSMS, die zur Bearbeitung Ein- oder Ausgaben auf Benutzerdateien erfordert; im Einzelnen:

ARCHIVE-FILES, ARCHIVE-NODE-FILES , BACKUP-FILES, BACKUP-FILE-VERSIONS, BACKUP-NODE-FILES, COPY-EXPORT-SAVE-FILE, COPY-NODE-SAVE-FILE, COPY-SAVE-FILE, EXPORT-FILES, IMPORT-FILES, MIGRATE-FILES, MOVE-SAVE-FILES, RECALL-MIGRATED-FILES, REORGANIZE-VERSION-BACKUP, REPAIR-CATALOG-BY-RESTORE, REPLACE-SAVE-FILE-BY-RESTORE, RESTORE-FILES, RESTORE-NODE-FILES, SELECT-FILE-NAMES, SELECT-JV-NAMES, SELECT-NODE-FILES

Anzahl der Backup-Versionen

Eine Zahl im Bereich von 0 bis 32, die die maximale Anzahl der Dateiversionen festlegt, die garantiert im Versions-Backup-Archiv aufbewahrt werden. Die Anzahl der Backup-Versionen (NUM-OF-BACKUP-VERS) ist ein Dateiattribut, das bei CREATE-FILE angegeben oder später mit MODIFY-FILE-ATTRIBUTES geändert werden kann. In einer SM-Umgebung kann es innerhalb einer Management-Klasse verwaltet werden. Der Wert wird während des Sicherungslaufs ins Archivverzeichnis des Versions-Backup-Archivs (//BACKUP-FILE-VERSION) geschrieben oder dort unabhängig davon mit //CHECK-CATALOG-FILES aktualisiert.

Archiv

Verwaltungseinheit für Dateien unter HSMS-Verwaltung. Es besteht aus der Archivdefinition und dem zugehörigen Archivverzeichnis (Directory-Datei). Ein Archiv wird über die -> Eigentümerkennung und den Archivnamen angesprochen.

HSMS unterscheidet verschiedene Archivtypen:

Für DVS-Dateien: Archive für -> Datensicherung (Backup-Archive), -> Versions-Backup (Versions-Backup-Archive), -> Langzeitarchivierung (Langzeitarchive), -> Verdrängung (Migrationsarchive) sowie -> Schattenarchive.

Für Knotendateien: Archive für Knotensicherung (Knoten-Backup-Archive) und Knotenarchivierung (Knoten-Langzeitarchive).

Ein Archiv kann privat sein, d.h. nur dem -> Archiveigentümer zur Verfügung stehen, oder aber öffentlich sein, d.h. allen Benutzern zugänglich sein.

Additional Mirror Unit

-> BCV-Spiegel oder -> Clone-Unit

Archival

-> Langzeitarchivierung

ARCHIVE

BS2000-Softwareprodukt zur logischen Sicherung von Dateien und Jobvariablen. ARCHIVE besitzt eine interne Schnittstelle zu HSMS und realisiert die -> Aktionsanweisungen von HSMS.

Archiveigentümer

Benutzer haben das Recht, mit CREATE-ARCHIVE Archive einzurichten; sie sind dann Archiveigentümer. Nur der HSMS-Verwalter darf bei der Anweisung CREATE-ARCHIVE im Archivnamen eine beliebige Benutzerkennung eingeben, wodurch der entsprechende Benutzer Eigentümer des Archivs wird.

Archivierung

-> Langzeitarchivierung

Archivtyp

Bestimmt, für welche der HSMS-Grundfunktionen das Archiv genutzt wird.

Archivverzeichnis

Datei, in der die in einem Archiv verwalteten Objekte, also Dateien, Jobvariablen, -> Sicherungsdateien, -> Sicherungsversionen und der -> Datenträger- Pool, eingetragen werden. Realisiert wird das Archivverzeichnis durch eine ARCHIVE-Directory-Datei (-> ARCHIVE).

Auftrag

Wird erzeugt zur Bearbeitung von -> Aktionsanweisungen und bei -> implizitem Zurückholen. Ein Auftrag wird in die -> Auftragsdatei eingetragen.

Auftragsdatei

Datei unter der Benutzerkennung SYSHSMS; sie enthält die Aufträge. Aus der Auftragsdatei heraus werden die Prozesse zur Abarbeitung der -> Aktionsanweisungen gestartet.

Ausnahmedatei

Datei, die vom Systemverwalter geführt wird. Sie enthält satzweise die (auch teilqualifizierten) Namen der Dateien, die von der Verdrängung ausgenommen werden.

Backup

-> Datensicherung

Backup-Archiv

HSMS-Archiv, das der Datensicherung dient.

Backup-Klasse

Sicherungsstufe einer Datei, die im Katalog eingetragen ist. Sie bestimmt, wie oft eine Datei gesichert wird. Mögliche Werte sind *A, *B, *C, *D und *E. Dateien mit Backup-Klasse *A werden immer gesichert. Dateien mit Backup-Klasse *E werden nur gesichert, wenn explizit MAXIMUM-BACKUP-CLASS=*E angegeben wurde.

Backup Monitor

-> BS2000 Backup Monitor

Backup-Server

Im SPVS-Verbund explizit vereinbarter Rechner zur Ausführung von HSMS-Aufträgen zur Entlastung der Produktivsysteme.

Bandverarbeitungszeit

Zeitpunkt, zu dem die bis dahin aufgelaufenen Ein- und Ausgaben von und nach -> S2 bearbeitet werden.

BCV-Spiegel

Zusätzliche -> Spiegelplatte in einem -> Plattenspeichersystem, die ohne Beeinträchtigung des laufenden Betriebs für andere Zwecke (Sicherung, Testverarbeitung usw.) abgetrennt werden kann. Der BCV-Spiegel wird auch als Additional-Mirror-Unit bezeichnet.

BS2000 Backup Monitor

Der BS2000 Backup Monitor ist als SE Management Anwendung im Hauptmenü **Anwendungen** des SE Managers integriert. Der BS2000 Backup Monitor informiert über den Status der Sicherungsaufträge, die in den BS2000-Systemen des SE Servers mit den Software-Produkten HSMS und FDDRL beauftragt wurden.

BS2000-Datei

Bezeichnet eine Datei, die ausschließlich von BS2000 angelegt und bearbeitet wird. BS2000-Dateien auf Net-Storage (FILE-TYPE=BS2000) werden seit BS2000/OSD-BC V9.0 bedient. Sie liegen direkt auf einem Net-Storage-Volume. Offene Systeme dürfen ausschließlich lesend darauf zugreifen.

BS2000-Net-Storage-Datei

-> Net-Storage-Datei

BS2000-UFS

UNIX-Dateisystem (POSIX), das sich auf einem BS2000-Server befindet.

Cataloged-Not-Saved (CNS)

Datei wurde nicht gesichert, weil sie entweder bei einer Differenzsicherung nicht geändert war oder wegen eines Fehlers (z.B. Open-Error) nicht gesichert werden konnte.

CFID

Coded File ID -> Interner Dateiname

Client

Rechner, der von einem anderen Rechner (-> Server) Dienste über das Netz in Anspruch nimmt. Ein Rechner kann gleichzeitig für bestimmte Funktionen als Client Dienste anfordern und für andere Rechner als Server Dienste zur Verfügung stellen.
siehe auch -> HSMS-Client

Client-Server-Architektur

Systemarchitektur, in der Rechnerkapazitäten und Anwendungen auf -> Clients und -> Server verteilt werden.
Client-Funktionen haben vor allem PCs, Workstations und -> UNIX-Systeme. Server-Funktionen werden überwiegend von Mainframe- und UNIX-Systemen ausgeübt, die dafür bestimmten Bedingungen genügen müssen Client- und Server-Systeme können beliebig kombiniert werden; jeder Client kann grundsätzlich auf jeden Server zugreifen.

Clone-Unit

Eine Clone-Unit ist die Kopie einer (Original-)Unit zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“), die abhängig vom Plattenspeichersystem von der Komponente EquivalentCopy, TimeFinder /Clone bzw. SnapViewClone erstellt wird. Nach der Aktivierung sind Unit und Clone-Unit von einander getrennt und Anwendungen können auf beide zugreifen. Eine Clone-Unit wird auch als Additional-Mirror-Unit bezeichnet.

CNS

-> Cataloged-Not-Saved

Coded-File-ID

-> Interner Dateiname

Collector-Request

-> Sammelauftrag

Concurrent Copy

Option der HSMS-Anweisungen BACKUP-FILES und BACKUP-FILE-VERSIONS, die das Ändern von BS2000-Dateien während einer Sicherung ermöglicht.

Continuation-Period

Zeitraum, währenddessen die Standard-Sicherungsdatei eine Archivs fortgeschrieben wird; zwischen einem Tag und einem Monat.

Data-Transfer

-> Datentransfer

Dateisystem

Hierarchische Sammlung von -> Dateiverzeichnissen und anderen Dateien, die in einer Baumstruktur angeordnet sind. Die Wurzel der Baumstruktur ist das -> Root-Verzeichnis (/). Alle anderen Dateiverzeichnisse sind Zweige, die vom Root-Verzeichnis ausgehen. Jede Datei eines Dateisystems ist über genau einen Pfad des Dateisystems erreichbar.

Dateiverzeichnis

Ein Dateiverzeichnis wird in UNIX-Dateisystemen verwendet, um Dateien oder Dateiverzeichnisse zu gruppieren und zu organisieren.

Daten

Im Rahmen von HSMS Dateien, Jobvariablen und Katalogeinträge von Dateien (von Magnetbandkassette oder Privatplatte)

Datensicherung

Regelmäßiges Erstellen von Kopien des Datenbestandes zur Wiederherstellung von Daten nach Datenverlust durch Hardware- oder Software-Fehler oder durch versehentliches Löschen usw. Kann auch zum Reorganisieren von Plattenspeichern verwendet werden.

Datenträger-Pool

Menge von Datenträgern, die von einem Archiv verwaltet werden und im Archivverzeichnis eingetragen sind. Aus dem Pool freier Datenträger werden Datenträger für Sicherungsaufträge standardmäßig angefordert.

Datentransfer

Übertragen von Dateien, Jobvariablen oder Katalogeinträgen von Dateien auf andere BS2000-Systeme oder andere Benutzerkennungen; realisiert durch Exportieren auf Magnetbandkassette, gemeinschaftlichen Datenträger oder -> Net-Storage und Importieren am Ziel.

Directory

-> Dateiverzeichnis

Directory-Datei

Synonym verwendeter Begriff für -> Archivverzeichnis. Einzelheiten über den Aufbau einer Directory-Datei siehe Handbuch „ARCHIVE“ [\[2\]](#), Abschnitt „Directory-Datei“.

Ebene

-> Speicherhierarchie

Eigentümerkennung

Die Benutzerkennung des -> Archiveigentümers wird bei der Einrichtung eines -> Archivs in der HSMS-Steuerdatei als Eigentümerkennung hinterlegt.
Gibt der HSMS-Verwalter beim Einrichten des Archivs keine Kennung an, so ist die Eigentümerkennung SYSHSMS.

ETERNUS-Plattensystem

-> Plattenspeichersystem der Fujitsu Technology Solutions GmbH.

Ethernet

Standardverfahren zur Kopplung von zwei Rechnern; daraus entsteht ein -> lokales Rechnernetz (LAN).

Except-File

-> Ausnahmedatei

Exportieren

Schreiben von Daten auf Magnetbandkassette, gemeinschaftlichen Datenträger oder -> Net-Storage zum -> Datentransfer.

Express-Request

-> Expressauftrag

Expressauftrag

Dem HSMS-Verwalter vorbehaltene Kategorie von Aufträgen, für die eigene -> Bandverarbeitungszeiten festgelegt werden können.

Ferner Rechner

In einem -> Netz werden ferne und -> lokale Rechner unterschieden. Alle Rechner im Netz, an denen ein Benutzer nicht direkt arbeitet, sind für diesen Benutzer ferne Rechner. Er kann mit allen fernen Rechnern im Netz kommunizieren. Wenn sich ein Benutzer an einen fernen Rechner anschließt, wird dieser für ihn zum lokalen Rechner.

File-Expiration-Date

-> Schutzfrist für die Sicherungsversion in einem Langzeitarchiv.

gültige Datei

-> Verdrängte Datei, die bisher weder zurückgeholt noch auf der Verarbeitungsebene gelöscht oder überschrieben wurde. Sie wird im Katalog als „verdrängt“ gekennzeichnet (# vor dem Namen).

HSMS

Hierarchisches Speicher Management System (Hierarchical Storage Management System): BS2000-Softwareprodukt mit Funktionen für -> Verdrängung (Migration), -> Datensicherung (Backup), -> Langzeitarchivierung (Archival) und -> Datentransfer, realisiert in einem -> Speicherhierarchie-Konzept und -> Archiven.

HSMS-Administrator

-> HSMS-Verwalter

HSMS-Funktionen

-> HSMS

HSMS-Lauf

Zeit zwischen dem Aufruf des Programms HSMS bis zur Beendigung mit END.

HSMS-Session

Zeit zwischen dem Laden des Subsystems HSMS (mit /START-SUBSYSTEM oder //START-HSMS) bis zum Entladen (mit /STOP-SUBSYSTEM oder //STOP-HSMS); siehe auch -> SM-Pubset-Session.

HSMS-Verwalter

Benutzer, der das HSMS-Verwalter-Privileg besitzt. Das Privileg haben Benutzer, die unter den Benutzerkennungen SYSHSMS und TSOS arbeiten.

Der HSMS-Verwalter kann sämtliche Funktionen von HSMS ohne Einschränkung nutzen. Er ist u.a. zuständig für die Verwaltung der Speicherhierarchie, die Einrichtung von Standard-Systemarchiven, die Systemsicherung und die Steuerung der Bandverarbeitung.

Implizites Zurückholen

-> Zurückholen von verdrängten Dateien nicht durch eine HSMS-Anweisung, sondern automatisch aufgrund von DVS-Zugriffsversuchen auf diese Dateien.

Importieren

Einlesen von exportierten Daten am Ziel zum -> Datentransfer.

Inactive Date

Datum, seit dem auf eine Datei nicht mehr zugegriffen wurde.

Inaktive Dateien

Dateien mit hohem -> Inactive Date, d.h. die seit längerer Zeit nicht mehr referenziert worden sind, bieten sich zur -> Verdrängung an.

Interner Dateiname

Neben dem Dateinamen im Katalog intern geführter Name, der die Datei eindeutig kennzeichnet und bei jeder Änderung der Datei verändert wird. Er wird beim FSTAT-Makro übergeben, bei SHOW-FILE-ATTRIBUTES jedoch nicht ausgegeben.

Knoten

Rechner (Workstation oder PC), der an ein Rechnernetz angeschlossen ist.

Knotendatei

Bedeutung bei UNIX-Workstations:

Knotendateien sind Dateien oder -> Dateiverzeichnisse im UNIX-Dateisystem (UFS) einer UNIX-Workstation. Beim Einsatz von -> NFS versteht man unter Knotendateien die Dateien oder -> Dateiverzeichnisse einer Workstation, auf die über das -> BS2000-UFS zugegriffen werden kann.

Bei Knotendateien wird Groß- und Kleinschreibung unterschieden.

Wenn Verzeichnisse oder Dateisysteme über das BS2000-UFS (POSIX) gemounted wurden, können die dort enthaltenen Dateien mit HSMS BACKUP-NODE-FILES oder ARCHIVE-NODE-FILES gesichert bzw. archiviert werden, sie können aber nicht direkt vom BS2000 DVS verarbeitet werden.

Knoten-Langzeitarchiv

HSMS-Archiv, das der -> Archivierung von Knotendateien dient.

Knoten-S0

Kennzeichnet ein fernes -> Dateisystem, das der Systemverwalter am Knotenpunkt „HSMS“ eines lokalen -> BS2000-UFS eingehängt hat.

Das lokale BS2000-UFS wird ebenfalls als Knoten-S0 angesehen.

Knoten-Sicherungsarchiv

HSMS-Archiv, das der -> Sicherung von Knotendateien dient.

LAN (Local Area Network)

Hardware-Konfiguration eines lokalen -> Netzes, in dem alle Datensichtgeräte und sonstigen Geräte in relativ geringem Abstand zueinander aufgestellt sind, z.B. innerhalb desselben Gebäudes. Die geringe Entfernung erlaubt einfachere Übertragungstechniken und damit höhere Geschwindigkeiten zu einem geringeren Preis.

In der Bundesrepublik Deutschland ist die Größe eines LAN auf das Grundstück des Anwenders beschränkt. Ein LAN kann als privates Subnetz mit anderen -> Rechnernetzen verbunden und so Teil eines größeren Netzes sein, etwa eines WAN.

Synonyme: Lokales Rechnernetz, lokales Netz.

Langzeitarchiv

HSMS-Archiv, das der Langzeitarchivierung dient.

Langzeitarchivierung

Langfristige Aufbewahrung ausgewählter Dateien oder Jobvariablen, z.B. aus Gründen der Produkthaftung oder Revisionsicherheit. Die Dateien können nach der Archivierung von der Bearbeitungsebene gelöscht werden.

lokal importiert

-> Pubset

Lokaler Rechner

Für einen Benutzer ist immer derjenige Rechner lokal, an dem er arbeitet. Alle anderen Rechner im -> Rechnernetz sind dann für ihn -> ferne Rechner. Wenn sich ein Benutzer an einen fernen Rechner anschließt, wird dieser für ihn zum lokalen Rechner.

Master-Rechner

Rechner, auf dem alle DVS-Verwaltungsfunktionen eines -> Shared-Pubsets ausgeführt werden.

MDS

-> Modified-during-save

Migration

-> Verdrängung

Migrationsarchiv

HSMS-Archiv, das der Verdrängung dient.

Miteigentümerschaft

Nicht-privilegierte Benutzer dürfen auch Dateien und Jobvariablen bearbeiten, bei denen sie Miteigentümer sind.

Modified-during-save

zeigt an, dass eine -> Knotendatei während der Sicherung von einem anderen Benutzer geöffnet wurde.

Monitoring

Wenn HSMS mit der Anwendung SM2 verbunden ist, können SM2-Benutzer Migrations- bzw. Rückhol-Aktivitäten besser überwachen. Die dabei gewonnenen Informationen können zu dem Ergebnis führen, dass die Migrationsparameter neu eingestellt werden müssen.

MPVS

Multiple-Public-Volume-Set; die gemeinschaftlichen Datenträger sind auf mehrere -> Pubsets verteilt. Die Benutzer können gezielt auf die Pubsets verteilt werden.

Multiplexbetrieb

Multiplexbetrieb ist nur für die Backup-Server-Funktionen von HSMS möglich, also für das Sichern und Restaurieren von Dateien auf fernen Workstations. Im Multiplexbetrieb können sich mehrere ARCHIVE-Subtasks gleichzeitig parallel dieselben MBK-Geräte und Magnetbandkassetten teilen. Dadurch wird eine höhere Performance, eine optimale Ausnutzung der MBK-Geräte und eine optimale Füllung der Magnetbandkassetten erreicht.

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-Datei

Bezeichnet eine Datei, die auf einem Net-Storage-Volume angelegt ist. Auf Net-Storage wird zwischen zwei Dateitypen BS2000-Datei und Node-File unterschieden.

Netz

Komplexes Gebilde aus Leitungen und Steuerungseinrichtungen, das der Datenübertragung dient.

NFS (Network File System)

BS2000-Softwareprodukt, mit dem verteilte Datenhaltung in einem heterogenen -> Rechnernetz möglich ist. Der Benutzer kann auf ferne Dateien so zugreifen, als ob sie an seinem -> lokalen Rechner vorhanden wären.

NFS dient somit der Konnektivität zwischen Systemen. Außerdem können Dateien mit NFS automatisch und zuverlässig durch das BS2000 gesichert werden.

NODE

-> Knoten

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. Node-Files werden ab BS2000 OSD/BC V10.0 unterstützt. Sie liegen auf einem Net-Storage-Volume in einem benutzerspezifischen Verzeichnis (Name der Benutzererkennung) und die Dateinamen entsprechen den BS2000-Namenskonventionen.

Bedeutung bei UNIX-Workstations:

-> Knotendatei

Owner-Id

-> Eigentümererkennung

Passiver HSMS-Client

bei UNIX-Workstations:

Das Softwareprodukt -> NFS wird zusätzlich benötigt.

Ein passiver HSMS-Client wird über -> NFS im POSIX-Filesystem des BS2000 gemountet. Die -> Sicherung/Archivierung erfolgt über HSMS mit der Anweisung BACKUP-NODE-FILES bzw. ARCHIVE-NODE-FILES. Eine zusätzliche Software ist auf dem Knoten nicht erforderlich.

Pfadname

Unter -> UNIX besitzt jede Datei und jedes -> Dateiverzeichnis einen eindeutigen Pfadnamen. Der Pfadname gibt die Position der Datei bzw. des Dateiverzeichnisses innerhalb des -> Dateisystems an und zeigt, wie darauf zugegriffen werden kann. Der Pfadname besteht aus den Namen aller darüberliegenden Dateiverzeichnisse, ausgehend von der Wurzel des Dateisystems, und dem eigentlichen Namen der Datei oder des Dateiverzeichnisses. Die Namen der Dateiverzeichnisse werden jeweils durch einen Schrägstrich (Slash) voneinander getrennt.

(Beispiel: /verzeichnis1/verzeichnis2/protokoll)

In UNIX wird zwischen absoluten und relativen Pfadnamen unterschieden.

Plattenspeichersystem

Externes Plattensystem. Für Plattenspeichersysteme, die die BS2000 Host-Komponente SHC-OSD verwaltet, können Spiegelungsfunktionen in BS2000 genutzt werden (z.B. Sicherung auf Snapsets).

Pool

-> Datenträger-Pool

Pubset

Public-Volume-Set; Satz zusammengehöriger gemeinschaftlicher Platten, die z.B. einen gemeinsamen Katalog besitzen. Ein Pubset gilt als lokal verfügbar, wenn sein Katalog und die Daten importiert wurden (IMPORT-PUBSET). Ein Pubset wird durch eine ein- bis vierstellige Pubsetkennung identifiziert.

Recall

-> Zurückholen

Rechnernetz

Zusammenschluss mehrerer Rechner über eine physikalische Verbindung mit dem Ziel, einen gleichberechtigten Datenaustausch zwischen diesen Rechnern zu ermöglichen. Es gibt lokale (-> LAN) und nichtlokale (-> WAN) Rechnernetze.

Rekonstruktion

-> Restore

Reorganisation

In HSMS vor allem das Reorganisieren der Migrationsarchive, d.h. das Umwälzen der Sicherungsdateien ohne Übernahme „ungültiger“ Dateien.

Beim Reorganisieren von Langzeitarchiven werden nur die Sicherungsversionen umkopiert, deren -> File-Expiration-Date noch nicht erreicht ist. Obsolete Sicherungsversionen werden dabei nicht übernommen.

Es können jedoch auch -> Pubsets und Privatplatten mit HSMS reorganisiert werden (-> Datensicherung).

Repository

-> Archivverzeichnis für Knotendateien

Request

-> Auftrag

Restore

Das Einspielen von Daten aus einem HSMS-Archiv auf die Verarbeitungsebene -> S0.

Retention-Period

-> Schutzfrist

Root-Verzeichnis

Hauptdateiverzeichnis in einem baumartig strukturierten -> Dateisystem, von dem alle anderen -> Dateiverzeichnisse abzweigen. Das Root-Verzeichnis muss auf jeden Fall vorhanden sein.

S0

Online-Verarbeitungsebene, bestehend aus Plattenspeichern mit kurzen Zugriffszeiten. Verwaltungseinheiten sind Pubsets.

S1

Online verfügbare Hintergrundebene, bestehend aus Plattenspeichern (mit evtl. höherer Kapazität und langsameren Zugriffszeiten als -> S0); Verwaltungseinheiten sind Pubsets.

S1-SM-Pubset

-> SM-Pubset, der als Speicherebene -> S1 für SF-Pubsets zugewiesen ist. Dadurch ist der für die Sicherung und Verdrängung zur Verfügung stehende Speicherplatz in der Speicherebene S1 nicht mehr auf die Größe eines SF-Pubsets von 4TB beschränkt, sondern kann nahezu bis 1000TB groß sein.

S2

Nur offline verfügbare Hintergrundebene, bestehend aus Magnetbandkassetten.

Sammelauftrag

Migrationsaufträge auf S2 und Archivierungsaufträge in die -> Standard-Sicherungsdatei eines Systemarchivs können, wenn sie nicht vom HSMS-Verwalter angestoßen wurden, zu einem Sammelauftrag zusammengefasst werden. Die Benutzeraufträge werden zusammen ausgeführt (weniger Bandzugriffe und Bandpositionierungen); die Reporte werden getrennt erzeugt. Bei Abbruch eines Sammelauftrags muss der HSMS-Verwalter den Restart veranlassen.

Save-File

-> Sicherungsdatei

Save-Version

-> Sicherungsversion

Schattenarchiv

Archiv, das mit einem -> Backup- oder -> Langzeitarchiv verbunden ist. In einem Schattenarchiv werden die automatisch erstellten Kopien von Daten, die in einem Backup- oder Langzeitarchiv gespeichert sind, abgelegt.

Schutzattribute

Sicherheitsrelevante Eigenschaften eines Objekts (Datei, Jobvariable etc.), die die Art und Möglichkeit des Zugriffs auf dieses Objekt festlegen.

Für Dateien gibt es beispielsweise folgende Schutzattribute: ACCESS, USER-ACCESS, AUDIT, READ-PASSWORD, WRITE-PASSWORD, EXEC-PASSWORD, RETENTION-PERIOD, BASIC-ACL und GUARD.

Schutzfrist

Zeitraum, während dem ein Ändern oder Löschen von Daten nicht zugelassen ist. Die *physische* Schutzfrist (Retention-Period) verhindert ein Überschreiben von Sicherungsdateien und Sicherungsdatenträgern während dieser Frist. Die *logische* Schutzfrist, durch das Freigabedatum (File-Expiration-Date) festgelegt, verhindert das Verändern oder Löschen von Dateien.

Server

Rechner, der anderen Rechnern (-> Clients) Dienste im Netzwerk zur Verfügung stellt.

SFID

Kennzeichnet eine -> Sicherungsdatei. Die Save-File-ID hat folgendes Format:
S.yymmdd.hhmmss

SF-Pubset

Abkürzung für Single-Feature-Pubset.

Ein SF-Pubset besteht aus einer Reihe von öffentlichen Plattenspeichern mit einem gemeinsamen Katalog; er wird durch seine Katalogkennung definiert. Ein SF-Pubset unterscheidet sich von einem -> SM-Pubset.

SF-Pubset-Umgebung

Reihe von HSMS-Parametern, die für alle -> SF-Pubsets spezifisch sind. Die Parameter der globalen Steuerdatei und der globalen Auftragsdatei sind darin eingeschlossen.

Shared-Pubset

Pubset, auf den gleichzeitig mehrere Rechner zugreifen können.

Sharer

Rechner, die einen bestimmten Pubset (-> Shared-Pubset) gleichzeitig importiert haben.

Sichern

Allgemein das Schreiben (Kopieren) von Daten in eine -> Sicherungsdatei, gleichgültig für welche Grundfunktion, aber auch speziell gebraucht für den Vorgang der -> Datensicherung.

Sicherung, logisch

Daten werden von einem oder mehreren Datenträgern gelesen und zusammenhängend auf einen oder mehrere Datenträger geschrieben.

Sicherung, physisch (physikalisch)

Sämtliche Daten eines Datenträgers, einschließlich der Datenträgerkennsätze, werden blockweise auf einen zweiten Datenträger geschrieben. Dieser ist in Inhalt und Aufbau identisch mit dem Originaldatenträger.

Sicherungsdatei

Behälter, in dem gesicherte Dateien und Jobvariablen abgelegt werden; enthält eine oder mehrere -> Sicherungsversionen und besteht aus einer Menge von Datenträgern mit demselben Eigentümer und derselben -> Schutzfrist (Retention-Period). Die Unterscheidung zwischen Sicherungsdatei und Sicherungsversion wird durch HSMS eingeführt, da im Rahmen von Archivierung, Migration und Knotensicherung mehrere Sicherungsversionen in einer Sicherungsdatei abgelegt werden können. Eine Sicherungsdatei kann nur als Ganzes freigegeben werden. Sie wird durch eine SAVE-FILE-ID (-> SFID) gekennzeichnet, die durch Datum und Zeit gebildet wird.

Sicherungsversion

Ergebnis eines Sicherungs- oder Archivierungsauftrags. Die Sicherungsversion wird intern gekennzeichnet durch eine SVID (Save-Version-ID). Der Benutzer kann sie über das Erzeugungsdatum ansprechen oder durch den Namen, der bei der Erzeugung vergeben wurde.

Single-Feature-Pubset

-> SF-Pubset

SM-Pubset

Abkürzung für systemverwaltetes Pubset (System-Managed-Pubset).

Das DVS-Objekt SM-Pubset besteht aus einer Reihe von -> Volume-Sets. Ein SM-Pubset unterscheidet sich von einem -> SF-Pubset.

In diesem Handbuch wird der Begriff SM-Pubset auch statt von -> „SM-Pubset unter HSMS-Kontrolle“ verwendet.

SM-Pubset außer HSMS-Kontrolle

SM-Pubset, der nicht für HSMS deklariert wurde und deshalb auch nicht von HSMS verwaltet werden kann.

SM-Pubset unter HSMS-Kontrolle

Geschlossener Behälter, der die Daten und die Metadaten enthält. In diesem Handbuch oft auch als -> SM-Pubset abgekürzt.

SM-Pubset-Session

Für HSMS die Zeitspanne zwischen dem Start des Subsystems HSMS (oder dem Import des SM-Pubsets, wenn dieser später durchgeführt wird) und dem Stop des Subsystems HSMS (oder dem Export des SM-Pubsets, wenn dieser vorher durchgeführt wird).

SM-Pubset-Umgebung

Reihe von HSMS-Parametern, die für einen -> SM-Pubset spezifisch sind. Die Parameter der Steuerdatei des SM-Pubsets und der Auftragsdatei des SM-Pubsets sind darin eingeschlossen.

Snapset

Kopie eines -> Pubsets auf zusätzlichen -> Spiegelplatten (Snap-Platten), die ab BS2000/OSD-BC V7.0 mit SHC-OSD in einem -> Plattenspeichersystem als Pubset-Sicherung erstellt werden können. Von einem Snapset können Dateien und Jobvariablen per DMS-Kommando restauriert bzw. mit HSMS gesichert werden.

Speicherhierarchie

Zuordnung von Speichermedien zu Ebenen, abhängig von Kapazität, Zugriffszeit und Speicherkosten (-> S0, -> S1, -> S2).

Spiegelplatte

Plattensatz, der aus mindestens zwei Platten mit identischem Inhalt besteht.

Standard-Sicherungsdatei

Sicherungsdatei, in die -> Sicherungen für ein -> Archiv standardmäßig geschrieben werden.

Standard-Systemarchiv

-> Archiv, auf das systemweit oder SF-/SM-pubset-spezifisch standardmäßig zugegriffen wird, wenn kein Archiv ausdrücklich angegeben ist. Standard-Systemarchive gibt es getrennt für die einzelnen HSMS-Grundfunktionen -> Verdrängung (-> SYSMIGRATE), -> Versions-Backup (-> SYSVERSION (nur SF-/SM-pubset-spezifisch)), -> Datensicherung (-> SYSBACKUP), -> Langzeitarchivierung (-> SYSARCHIVE), Knotensicherung (-> SYSNODEBACKUP) und Knotenarchivierung (-> SYSNODEARCHIVE).

Steuerdatei

Datei unter der Benutzerkennung SYSHSMS des Home-Pubsets; sie enthält die HSMS-Steuerparameter und die Archivdefinitionen.

Bei systemverwalteten Speichermedien (System Managed Storage) befindet sich die Steuerdatei unter der Benutzerkennung SYSHSMS eines -> SM-Pubsets; sie enthält die Steuerparameter dieses SM-Pubsets.

Storage-Level

-> Speicherhierarchie

SYSARCHIVE

-> Standard-Systemarchiv für die -> Langzeitarchivierung

SYSBACKUP

-> Standard-Systemarchiv für die -> Datensicherung

SYSMIGRATE

-> Standard-Systemarchiv für die -> Verdrängung

SYSNODEARCHIVE

-> Standard-Systemarchiv zum Sichern von -> Knotendateien.

SYSNODEBACKUP

-> Standard-Systemarchiv zum Archivieren von -> Knotendateien.

System Managed Pubset

-> SM-Pubset

Systemarchiv

-> Standard-Systemarchiv

Systemeinleitung

Laden der Betriebssystem-Software des BS2000. Es gibt folgende Arten der Systemeinleitung: AUTOMATIC-Startup, DIALOG-Startup und FAST-Startup. Diese Arten unterscheiden sich durch unterschiedlichen Automatisierungsgrad

SYSDIR

-> Pubset-spezifisches Standardarchiv für -> Versions-Backup von Dateien

TAPE

Datenträger der Klasse „TAPE“ werden der Speicherebene -> S2 zugeordnet.

Tape-Session

-> Bandverarbeitungszeit

UFS (UNIX File System)

Komponente von -> NFS zum Zugriff auf lokale Dateisysteme.

ungültige Datei

In einer -> Sicherungsdatei des -> Migrationsarchivs enthaltene Datei, die auf der Verarbeitungsebene S0 bereits gelöscht, überschrieben oder zurückgeholt wurde.

UNIX

Ein im Dialogbetrieb arbeitendes Betriebssystem, das 1969 von Bell Laboratories für Kleincomputer entwickelt wurde, inzwischen aber in allen Rechnerklassen eingesetzt werden kann. Da nur ein zentraler Kern von UNIX hardwareabhängig ist, wird UNIX auf vielen unterschiedlichen Systemen verschiedener Hersteller eingesetzt.

verdrängte Datei

Datei, deren Daten auf der Verarbeitungsebene gelöscht wurden, die aber ihren Katalogeintrag dort behält. In ihrem Katalogeintrag wird vermerkt, auf welche Hintergrundebene die Daten verdrängt wurden.

Verdrängung

Auslagerung -> inaktiver Dateien aus der Verarbeitungsebene auf eine Hintergrundebene ohne Löschung des Katalogeintrags.

Versions-Backup-Archive

Ein neuer Archivtyp, der speziell für das Versions-Backup eingeführt wurde. Mit Hilfe des Versions-Backup kann der Benutzer dafür sorgen, dass eine bestimmte Anzahl von Dateiversionen im Archiv aufbewahrt wird. Der Benutzer steuert dies über die Dateieigenschaft NUM-OF-BACKUP-VERS (anzugeben bei CREATE-FILE oder MODIFY-FILE-ATTRIBUTES), bzw. bei SM-Pubsets über Management-Klassen. Für den Fall, dass der Benutzer Dateien aus seiner Benutzerkennung versehentlich gelöscht hat, werden besondere Mechanismen zur Verfügung gestellt, um den Benutzer vor Datenverlust zu schützen.

Verzeichnis

Verzeichnis der exportierten Daten, analog zum -> Archivverzeichnis.

Volume-Pool

-> Datenträger-Pool

Volume-Set

Reihe von Platten, die durch ihre Volume-Set-ID angesprochen werden können. Mehrere Volume-Sets bilden einen -> SM-Pubset.

WAN (Wide Area Network)

-> Rechnernetz, das nicht auf ein räumlich begrenztes Gebiet beschränkt ist. Ein WAN kann unter anderem aus mehreren -> LAN bestehen.

Zurückholen

Einspielen verdrängter Dateien von einer Hintergrundebene auf die Verarbeitungsebene -> S0. Das Zurückholen kann explizit durch die Anweisung RECALL-MIGRATED-FILES oder implizit durch das Öffnen der Datei angestoßen werden.

13 Abkürzungen

ACS	Alias Catalog System
AML	Automatic Media Library
ASCII	American Standard Code for Information Interchange
BACL	Basic Access Control List
BCAM	Basic Communication Access Method
BCV	Business Continuance Volumes
BLS	Binder-Lader-Starter
Catid	Catalog Identifier
CCOPY	Concurrent Copy
CFID	Coded File ID
CHKPT	Checkpoint
CL	Client
CNS	Cataloged Not Saved
DAB	Disk Access Buffer
DCAM	Data Communication Access Method
DEFLUID	Default User ID
DFS	Distributed File System
DGG	Dateigenerationsgruppe
DMS	Data Management System
DSECT	Dummy Section
DSSM	Dynamic Subsystem Management
DVS	Datenverwaltungssystem
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDT	Editor
EOF	End of File
EOV	End of Volume
FARMTSAV	File Archiving Metadata Save

FCB	File Control Block
FDDRL	Fast Disk Dump and Reload
FGG	File Generation Group
FHS	Format Handling System
FIFO	First In First Out
FTP	File Transfer Protocol
GCF	Generic Catalog Facility
GUARDS	Generally Usable Access Control Administration System
HIPLEX	Highly Integrated System Complex
HSMS	Hierarchical Storage Management System
IFG	Interactive Format Generator
IMON	Installations-Monitor
INOP	Inoperable (Gerätezustand)
ISAM	Indexed Sequential Access Method
JV	Jobvariable
K-Format	Key-Format
KB	Kilobyte
LMS	Library Maintenance System
MAREN	Magnetdatenträger-Archivierungssystem im Rechnernetz
MBK	Magnetbandkassette
MIP	Message Improvement Processing
MPVS	Multiple Public Volume Set
MRSCAT	Mehrrechner-Systemkatalog
MSCF	Multiple System Control Facility
MSG	Message
NDM	Nucleus Device Management
NFS	Network File System
NK-Format	Nonkey-Format

OPS	Output Presentation Services
OSD	Open Server Dimension (bisher: Open Systems Direction)
PAM	Primary Access Method
PCS	Performance Control Subsystem
PFA	Performant File Access
POSIX	Portable Open System Interface for UNIX
PVS	Public Volume Set (Pubset)
RAID	Redundant Array of Independent Disks
RFA	Remote File Access
ROBAR	Roboterarchiv
RSO	Remote SPOOL Output
RU	Release Unit
SAM	Sequential Access Method
SDF	System Dialog Facility
SECOS	Security Control System
SF-Pubset	Single-Feature-Pubset
SHC-OSD	Storage Host Component
SM-Pubset	System-Managed-Pubset
SMS	System Managed Storage
SNMP	Simple Network Management Protocol
SOLIS	Software Liefer- und Informationssystem
SPVS	Shared Public Volume Set
SFID	Save File Identification
SRDF	Symmetrix Remote Data Facility
SRPM	System Resources and Privileges Management
SV	Server
SVID	Save Version Identification
SYSSII	Structure-and-Installation-Information

TCP/IP	Transmission Control Protocol / Internet Protocol
TEMPFILE	Temporäre Datei
TFT	Task File Table
TIAM	Terminal Interactive Access Method
TLS	Tape Library System
TPR	Task Privileged
TSN	Task Sequence Number
TSOS	Time Sharing Operating System
TU	Task Unprivileged
UDS	Universelles Datenbank System
UFS	Unix File System
VAV	Verbessertes Aufzeichnungsverfahren
VOL	Volume Header Label
VSN	Volume Serial Number

14 Literatur

Die Handbücher finden Sie im Internet unter <http://bs2manuals.ts.fujitsu.com>.

- [1] **HSMS V11.0A** (BS2000)
Hierarchisches Speicher Management System
Band 2: Anweisungen
Benutzerhandbuch
- [2] **ARCHIVE** (BS2000)
Benutzerhandbuch
- [3] **BS2000 OSD/BC**
Dienstprogramme
Benutzerhandbuch
- [4] **BS2000 OSD/BC**
Einführung in das DVS
Benutzerhandbuch
- [5] **SDF** (BS2000)
Dialogschnittstelle SDF
Benutzerhandbuch
- [6] **BS2000 OSD/BC**
Einführung in die Systembetreuung
Benutzerhandbuch
- [7] **IMON** (BS2000)
Installationsmonitor
Benutzerhandbuch
- [8] **BS2000 OSD/BC**
Kommandos
Benutzerhandbuch
- [9] **BS2000 OSD/BC**
Makroaufrufe an den Ablaufteil
Benutzerhandbuch
- [10] **MAREN** (BS2000)
Bandverwaltung in BS2000
Benutzerhandbuch
- [11] **NFS** (BS2000)
Network File System
Benutzerhandbuch
- [12] **POSIX** (BS2000)
Grundlagen für Anwender und Systemverwalter
Benutzerhandbuch

- [13] **POSIX (BS2000)**
Kommandos
Benutzerhandbuch
- [14] **SDF-A (BS2000)**
Benutzerhandbuch
- [15] **SDF-P (BS2000)**
Programmieren in der Kommandosprache
Benutzerhandbuch
- [16] **SECOS (BS2000)**
Security Control System
Benutzerhandbuch
- [17] **openSM2 (BS2000)**
Software Monitor
Benutzerhandbuch
- [18] **BS2000 OSD/BC**
System Managed Storage
Benutzerhandbuch
- [19] **PROP-XT (BS2000)**
Programmiertes Operating mit komfortablen Sprachmitteln von SDF-P

Produktbuch
- [20] **HIPLEX MSCF (BS2000)**
BS2000-Rechner im Verbund
Benutzerhandbuch
- [21] **SHC-OSD (BS2000)**
Storage-Management für BS2000
Benutzerhandbuch
- [22] **BS2000 OSD/BC**
DVS-Makros
Benutzerhandbuch
- [23] **JV (BS2000)**
Jobvariablen
Benutzerhandbuch
- [24] **openNet Server (BS2000)**
BCAM
Benutzerhandbuch
- [25] **SE700 / SE500 /SE300**
Bedienen und Verwalten
Benutzerhandbuch

[26] **BS2000 OSD/BC**
Performance Handbuch

- [27] **BS2000 OSD/BC**
Diagnosehandbuch
Benutzerhandbuch
- [28] **BS2000 OSD/BC**
Abrechnungssätze
Benutzerhandbuch
- [29] **interNet Services (BS2000)**
Administratorhandbuch