

Deutsch



UDS/SQL V2.9B

Datenbankbetrieb

Benutzerhandbuch

Juni 2019

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 erfüllt.

Copyright und Handelsmarken

Copyright © 2019 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

Datenbankbetrieb	12
1 Einleitung	13
1.1 Konzept der UDS/SQL-Dokumentation	14
1.2 Zielsetzung und Zielgruppen des Handbuchs	18
1.3 Konzept des Handbuchs	19
1.4 Änderungen gegenüber den Vorgänger-Handbüchern	20
1.5 Darstellungsmittel	22
1.5.1 Warnhinweise und Hinweise	23
1.5.2 Nicht-SDF-Darstellungsmittel	24
1.5.3 SDF-Syntaxdarstellung	25
2 Der Database Handler (DBH)	30
2.1 Arbeitsweise des independent DBH	33
2.2 Arbeitsweise des linked-in DBH	40
2.3 Die Session	43
2.3.1 DBH starten	44
2.3.1.1 independent DBH	45
2.3.1.2 linked-in DBH	47
2.3.1.3 DBH-Startkommandos	48
2.3.1.4 Meldungen beim DBH-Start	56
2.3.1.5 Beispiele	58
2.3.2 DB-Konfiguration festlegen und ändern	61
2.3.3 DBH beenden	63
2.4 Paralleler Datenbankbetrieb	66
3 Ladeparameter des DBH	67
3.1 Größe des 2-Kbyte System Buffer Pools festlegen (2KB-BUFFER-SIZE) ...	75
3.2 Größe des 4-Kbyte System Buffer Pools festlegen (4KB-BUFFER-SIZE) ...	76
3.3 Größe des 8-Kbyte System Buffer Pools festlegen (8KB-BUFFER-SIZE) ...	77
3.4 Administrierungsverfahren festlegen (ADM)	78
3.5 Kennwort für die Administration über DCAM festlegen (ADMPASS)	79
3.6 Präfix für Anwendertasks in UDS-D-Verteilungen festlegen (BCAM-PREFIX) ..	80
3.7 Kennwort für vom DBH einzurichtende Dateien festlegen (CATPASS)	81
3.8 Verbindung und Transaktion überwachen (CHCKTIME)	82
3.9 Meldungen des linked-in DBH auf Operatorkonsole ausgeben (CONSOLE) ...	83
3.10 Mindestgröße für einzelne Common Pools festlegen (CP-SIZE)	84
3.11 Prozessortyp angeben (CPU)	85

3.12 Mindestgröße des Communication Pools festlegen (CUP-SIZE)	86
3.13 Datenbanken benennen (DBNAME)	87
3.14 Servertasks auf Grund der Rechnerauslastung deaktivieren (DEACT) ...	91
3.15 Konfigurationsübergreifende Deadlocks bzw. Deadlocks im Zusammenhang mit openUTM auflösen (DEADTIME)	92
3.16 Mindestgröße des Distribution Pools festlegen (DIP-SIZE)	93
3.17 Anzahl entfernter Datenbanken festlegen (DISDB)	94
3.18 Eingabedatei für die Verteiltabelle festlegen (DISTABLE)	95
3.19 Teilnahme am UDS-D-Betrieb steuern (DISTRIBUTION)	96
3.20 Umfang eines Dumps beeinflussen (DUMP)	97
3.21 Parametereingabe beenden (END)	98
3.22 I/Os asynchron oder synchron ausführen (IO)	99
3.23 Sperrprotokoll festlegen (LOCK)	100
3.24 RLOG-Datei führen (LOG)	102
3.25 RLOG-Duplikatdatei führen (LOG-2)	106
3.26 Speicherplatzgröße der RLOG-Dateien festlegen (LOG-SIZE)	108
3.27 Anzahl Datenbanken der DB-Konfiguration festlegen (MAXDB)	109
3.28 Segmentgröße der Memory Pools festlegen (MPSEG)	110
3.29 Systemverhalten bei Kollision von vorgemerkten Aufträgen und UPDATE-Transaktionen steuern (ORDER-DBSTATUS)	111
3.30 Verwendete Ladeparameter auflisten (PARLIST)	112
3.31 Kennwort festlegen für Dateien, die der DBH verwendet (PASSWORD) ..	113
3.32 Behandlung der Privacy-Prüfung steuern (PRIVACY-CHECK)	115
3.33 Transaktionen im Zustand PTC bei Fehlern behandeln (PTCSYNCH)	116
3.34 Ersatzdatenträger für die RLOG-Dateien festlegen (RESERVE)	117
3.35 Auftragsantworten an die Anwendertasks bündeln (RESULT-DELAY) ...	119
3.36 Schedulingverhalten in Hochlastsituationen steuern (SCHEDULING)	120
3.37 Anzahl Servertasks festlegen (SERVERTASK)	121
3.38 Größe des SSITAB-Pools festlegen (SIP-SIZE)	122
3.39 Anzahl gleichzeitig aktiver SQL-Vorgänge festlegen (SQL)	123
3.40 Zeitspanne für nicht aktive SQL-Vorgänge einstellen (SQL-LIMIT)	124
3.41 Konsistenzpunkte in ALOG-Dateien festschreiben (STDCKPT)	125
3.42 Anzahl Subschemata festlegen (SUBSCHEMA)	128
3.43 Anzahl KDBS-Dateikennungen festlegen (SUBTRANSACTION)	129
3.44 Benutzungsarten für Transaktionen festlegen (TA-ACCESS)	130
3.45 Maximale Anzahl Transaktionen festlegen (TRANSACTION)	131
3.46 Bedienplatz zur Protokollierung der DCAM-Administration und Form der UDS /SQL-Meldungen festlegen (UCON)	133
3.47 Warte-Modus der Servertasks einstellen (WAIT)	134
3.48 Dauer eines Warmstarts beeinflussen (WARMSTART)	135

4 Administration	136
4.1 UDS/SQL über UDSADM administrieren	137
4.1.1 UDSADM-Anweisungen	138
4.1.1.1 Verbindung zur UDS/SQL-Konfiguration aufbauen (CONNECT-CONFIGURATION)	140
4.1.1.2 Verbindung zur UDS/SQL-Konfiguration abbauen (DISCONNECT-CONFIGURATION)	141
4.1.1.3 UDSADM beenden (END)	142
4.1.1.4 DAL-Kommandos ausführen (EXECUTE-DAL-CMD)	143
4.1.1.5 Hilfefunktion für DAL-Kommandos (HELP-DAL-CMD)	144
4.1.1.6 Ausgabeformat für Nachrichten einstellen (MODIFY-MSG-FORMAT)	145
4.1.1.7 Wartezeit für den Empfang von Nachrichten einstellen (MODIFY-MSG-WAIT-TIME)	146
4.1.1.8 Dauerempfangsprozess aktivieren (SET-RECEIVE-MODE)	147
4.1.1.9 Verbindungsinformationen ausgeben (SHOW-CONNECTION-ATTRIBUTES)	148
4.1.1.10 Anstehende Nachrichten ausgeben (SHOW-OUTSTANDING-MSG)	149
4.2 UDS/SQL über DCAM administrieren	150
4.3 UDS/SQL über INFORM-PROGRAM administrieren	153
4.4 Die Datenbankadministratorsprache DAL	154
4.4.1 Eine oder alle Transaktionen abbrechen (ABORT)	162
4.4.2 Zugriffssperren auf Datenbank- und Realm-Ebene behandeln (ACCESS)	163
4.4.3 Online-Erweiterbarkeit aktivieren (ACT)	165
4.4.4 Datenbanken, Realms und Kennwörter anschließen (ADD)	169
4.4.5 Neue Einträge in die Verteiltabelle bringen (&ADD DISTRIBUTION)	174
4.4.6 Anzahl bearbeiteter Base Interface Blocks ausgeben (%BIB)	176
4.4.7 Konfiguration einem anderen Verarbeitungsrechner zuordnen (&CHANGE DISTRIBUTION)	177
4.4.8 Konsistenzpunkte festschreiben (CHECKPOINT)	178
4.4.9 Session oder Administration beenden (CLOSE)	180
4.4.10 UDS-D-Betrieb beenden (&CLOSE DISTRIBUTION)	181
4.4.11 Transaktion im Zustand PTC beenden (COMMIT)	182
4.4.12 DISPLAY SQL-Ausgabe fortsetzen (CONTINUE)	183
4.4.13 Online-Erweiterbarkeit deaktivieren (DEACT)	184
4.4.14 Informationen zur DB-Konfiguration auflisten (DISPLAY)	186
4.4.14.1 DISPLAY DB - Ausgabe aller Datenbanken und Zugriffsarten	190
4.4.14.2 DISPLAY DB,RUNID=transaktionskennung - Ausgabe aller Datenbanken, die durch die Transaktionskennung angesprochen werden	192
4.4.14.3 DISPLAY USERS - Ausgabe aller Transaktionen, die derzeit arbeiten	193
4.4.14.4 DISPLAY USERS,DB=dbname - Ausgabe aller Transaktionen, die derzeit die Datenbank ansprechen	196
4.4.14.5 DISPLAY SUBSCH - Ausgabe aller Subschemas	197

4.4.14.6 DISPLAY SUBSCH,DB=dbname - Ausgabe aller Subschemas der Datenbank	199
4.4.14.7 DISPLAY MAINREF - Ausgabe von Informationen über den Zustand der Mainrefs	200
4.4.14.8 DISPLAY MAINREF,STATE=BLOCK - Ausgabe von Informationen über Mainrefs, die blockiert sind	202
4.4.14.9 DISPLAY transaktionskennung - Ausgabe von Informationen über die Transaktionen	203
4.4.14.10 DISPLAY REALMS - Ausgabe von Informationen über alle permanenten Realms	206
4.4.14.11 DISPLAY REALMS,DB=dbname - Ausgabe von Informationen über alle permanenten Realms einer Datenbank	209
4.4.14.12 DISPLAY REALMS,RN=realmname[,DB=dbname] - Ausgabe von Informationen über einen permanenten Realm	210
4.4.14.13 DISPLAY PP - Ausgabe der aktuellen DBH-Ladeparameter	211
4.4.14.14 DISPLAY FPA - Ausgabe von Informationen über freie Seiten in allen Realms	215
4.4.14.15 DISPLAY FPA,DB=dbname - Ausgabe von Informationen über freie Seiten in allen Realms einer Datenbank	217
4.4.14.16 DISPLAY FPA,RN=realmname [,DB=dbname]- Ausgabe von Informationen über freie Seiten in einem Realm	218
4.4.14.17 DISPLAY XFPA - Ausgabe von erweiterten Informationen über freie Seiten in allen Realms	219
4.4.14.18 DISPLAY XFPA,DB=dbname - Ausgabe von erweiterten Informationen über freie Seiten in allen Realms einer Datenbank	221
4.4.14.19 DISPLAY XFPA,RN=realmname [,DB=dbname]- Ausgabe von erweiterten Informationen über freie Seiten in einem Realm	222
4.4.14.20 DISPLAY INCR - Ausgabe von Informationen über die Online-Erweiterbarkeit aller zugeschalteten Realms in der DB-Konfiguration	223
4.4.14.21 DISPLAY INCR, DB=dbname - Ausgabe von Informationen über die Online-Erweiterbarkeit aller zugeschalteten Realms einer Datenbank	226
4.4.14.22 DISPLAY INCR,RN=realmname[,DB=dbname] - Ausgabe von Informationen über die Online-Erweiterbarkeit eines Realms	227
4.4.14.23 DISPLAY DBTT-INCR - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit der Satzarten der User-Schemata in der DB-Konfiguration	228
4.4.14.24 DISPLAY DBTT-INCR,DB=dbname - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit der Satzarten des User-Schemas einer Datenbank ...	231
4.4.14.25 DISPLAY DBTT-INCR,DB=dbname,RECR=recordref - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit einer Satzart	232
4.4.14.26 DISPLAY PUBSETS - Ausgabe der Einstellungen der UDS/SQL-Pubsetdeklaration	233
4.4.14.27 DISPLAY ALOG [,DB=dbname] - Ausgabe von ALOG-Einstellungen ..	234
4.4.15 Verteiltabelle ganz oder teilweise ausgeben (&DISPLAY DISTRIBUTION) ..	236
4.4.16 Informationen über einen SQL-Vorgang ausgeben (DISPLAY SQL)	239

4.4.17 Anzahl Nachrichten an den UDS/SQL-DBH ausgeben (%DML)	245
4.4.18 Datenbanken, Realms und Kennwörter ausschließen (DROP)	246
4.4.19 Einträge in der Verteiltabelle löschen (&DROP DISTRIBUTION)	250
4.4.20 Speicherabzug erstellen (DUMP)	252
4.4.21 Speicherabzug für linked-in DBH erstellen (%DUMP)	253
4.4.22 Online-DBTT-Erweiterung ausführen (EXTEND DBTT)	254
4.4.23 Online-Realm-Erweiterung ausführen (EXTEND REALM)	256
4.4.24 SQL-Betriebsmittel freigeben (FORGET SQL)	257
4.4.25 STOP-Zustand von Transaktionen aufheben (GO)	258
4.4.26 Einträge in der Verteiltabelle sperren (&LOCK DISTRIBUTION)	259
4.4.27 Einstellungen für DEFAULT-SUPPORT und RESERVE-SUPPORT der ALOG-Dateien ändern (MODIFY ALOG)	261
4.4.28 Speicherplatzzuweisung der ALOG-Dateien ändern (MODIFY ALOG-SIZE)	262
4.4.29 Datenträgerzuweisung der RLOG-Dateien ändern (MODIFY LOG)	263
4.4.30 Speicherplatzgröße der RLOG-Dateien ändern (MODIFY LOG-SIZE)	265
4.4.31 PTCSYNCH-Werte ändern (MODIFY PTCSYNCH)	266
4.4.32 Ersatzdatenträger für RLOG-Dateien ändern (MODIFY RESERVE)	267
4.4.33 Neue UDS/SQL-Pubset-Deklaration prüfen und vormerken (NEW PUBSETS)	268
4.4.34 Neue RLOG-Dateien auswählen (NEW RLOG)	271
4.4.35 Vorgemerkte Aufträge ausführen (PERFORM)	272
4.4.36 Kennwort vergeben und verändern (&PWD DISTRIBUTION)	274
4.4.37 Online-Erweiterbarkeit für einen Realm reaktivieren (REACT INCR)	275
4.4.38 Vorgemerkte Aufträge zurücknehmen (RESET ORDERS)	276
4.4.39 Verteiltabelle sichern (&SAVE DISTRIBUTION)	277
4.4.40 UDS-D-Betrieb starten (&START DISTRIBUTION)	278
4.4.41 Ausführung von Transaktionen stoppen (STOP)	279
4.4.42 Sekundäre Teiltransaktionen im Zustand PTC beenden (&SYNCHRONIZE DISTRIBUTION)	280
4.4.43 Session sofort abbrechen (%TERM)	281
4.4.44 Sperren für Einträge in der Verteiltabelle aufheben (&UNLOCK DISTRIBUTION)	282
5 Hochverfügbarkeit	284
5.1 Unterbrechungsfreier Datenbank-Betrieb	285
5.2 Zusammenwirken mit dem Transaktionsmonitor openUTM	286
5.3 Strukturierung der Datenbanken	287
5.4 Fehlertoleranz	288
5.5 Verteilte Verarbeitung	289
5.6 Langzeitsicherung	290
5.7 Restartfunktionen	291

5.8 Nutzung von BS2000-Standardverfahren zur Erhöhung der Datensicherheit ..	292
5.9 Schutz vor Ausfall lokaler Betriebsmittel	293
5.10 Einsatz von UDS/SQL-Dienstprogrammen	294
6 Ressourcen-Erweiterung und Reorganisation im laufenden Betrieb	295
6.1 Systemvoraussetzungen der Online-Erweiterung	296
6.1.1 Speicherplatz zuweisen	297
6.1.2 DVS-Aspekte der Online-Realm-Erweiterung	298
6.1.3 Erweiterbarkeit eines Realms einschätzen	299
6.1.4 Auswirkungen von Dienstprogrammen auf die Online-Erweiterbarkeit von Realms	300
6.1.5 Datenbank-Layout-Version und Realm-Layout-Version	301
6.1.6 Datenbankseitenformatierung bei Realm-Erweiterung	302
6.1.7 Freiplatzverwaltung bei Realm-Erweiterung	303
6.2 Online-Realm-Erweiterung	304
6.2.1 Ablauf der Online-Realm-Erweiterung	305
6.2.2 Verhalten im Fehlerfall	306
6.2.3 Reaktionsmöglichkeiten nach fehlgeschlagener Online-Realm- Erweiterung	307
6.2.4 DAL-Kommandos und Dienstprogramme für die Online-Realm- Erweiterung	308
6.2.5 Aktivierung der Online-Erweiterbarkeit beim Datenbankaufbau	310
6.3 Online-DBTT-Erweiterung	311
6.3.1 Online-DBTT-Erweiterung aktivieren	312
6.3.2 Ablauf der automatischen Online-DBTT-Erweiterung	313
6.3.3 Explizite Online-DBTT-Erweiterung veranlassen	314
6.3.4 Verhalten im Fehlerfall	315
6.3.5 DAL-Kommandos und Dienstprogramme für die Online-DBTT- Erweiterung .	316
6.4 Online-Reorganisation mit der UDS-Online-Utility	317
6.5 Automatische Realm-Erweiterung durch Dienstprogramme	318
7 Datenbank sichern und wiederherstellen im Fehlerfall	320
7.1 Das Sicherungskonzept von UDS/SQL	321
7.2 Transaktionssicherung	322
7.2.1 Fehlerarten und ihre Behandlung	323
7.2.1.1 Fehler im Anwenderprogramm	324
7.2.1.2 Fehler im System	325
7.2.2 Handhabung der benötigten Dateien	326
7.2.2.1 RLOG-Datei	327
7.2.2.2 Session-Log-File (SLF)	329
7.2.2.3 Status-Datei	330
7.3 Langzeitsicherung	331
7.3.1 Datenbanksicherung erstellen	332
7.3.1.1 Konsistenzpunkt	333

7.3.1.2	Sicherung einer konsistenten Datenbank	334
7.3.1.3	Online-Sicherung einer Datenbank	335
7.3.1.4	Schattendatenbank	336
7.3.2	ALOG-Dateien führen	337
7.3.3	Datenbank rekonstruieren	339
7.4	Auswahl des Sicherungskonzeptes	342
7.5	Beispiele	345
8	Leistungsoptimierung	359
8.1	I/O-Verhalten optimieren	360
8.2	Arbeitsspeichernutzung durch Subsystemfunktionalität von UDS/SQL optimieren	362
8.3	Prozessornutzung mit independent DBH optimieren	364
8.3.1	Ladeparameter, die die Prozessornutzung beeinflussen	365
8.3.2	Leistungsbezogene BS2000-Einstellungen	367
8.3.3	Optimierte Taskkommunikation	368
8.3.4	Interpretation der Kommunikationszähler des UDS/SQL-Monitors	369
8.3.5	Beispiel Hauptlastkonfiguration	370
9	Nutzung der BS2000-Funktionalität	372
9.1	Pubsets für UDS/SQL nutzen	373
9.2	Jobvariablen für UDS/SQL nutzen	376
9.2.1	Pubset-Deklarations-Jobvariable	377
9.2.1.1	Nutzung der UDS/SQL-Pubset-Deklaration im DBH	381
9.2.1.2	Nutzung der UDS/SQL-Pubset-Deklaration in den Dienstprogrammen	382
9.2.1.3	Sonstige Nutzung der UDS/SQL-Pubset-Deklaration	383
9.2.2	Session-Jobvariable	384
9.2.3	Datenbank-Jobvariable	388
9.2.4	Hinweise zur Verwendung der Jobvariablen	393
9.2.5	BMEND-Jobvariable	394
9.3	Mehrere UDS/SQL-Versionen parallel nutzen	395
9.3.1	Beispiel: Parallele Nutzung von zwei UDS/SQL-Versionen	396
9.4	DAB-Caching für UDS/SQL nutzen	399
9.5	Datenspiegelung für UDS/SQL nutzen	400
9.6	HSMS für UDS/SQL nutzen	401
9.7	FASTPAM für UDS/SQL nutzen	402
9.8	Vollen Adressraum für UDS/SQL nutzen	403
9.9	Zeitermittlung über das Subsystem GET-TIME	404
9.10	Kennwörter für UDS/SQL-Dateien vergeben	405
9.11	Accounting im UDS/SQL-openUTM-Betrieb	407
9.12	UDS/SQL-Benutzer und -Benutzergruppen	408
9.12.1	Zugriffskontrolle über openUTM	409
9.12.2	UDS/SQL-Benutzergruppen definieren und Zugriffsrechte vergeben	411

9.13 ACS (Alias Catalog Service)	414
9.14 Unicode-Konzept in UDS/SQL	415
9.14.1 Einsatz von Unicode (UTF-16) in bestehenden UDS/SQL-Datenbanken ...	416
10 Der SQL-Vorgang	418
11 DB-Betriebs- und Statuswerte ausgeben mit UDSMON	419
11.1 Beschreibung der Funktionen	421
11.2 Systemumgebung	423
11.3 Anweisungen und Kommandos für UDSMON	425
11.4 DISPLAY-Ausgabe des UDS-Monitors in Zwischendateien	431
11.5 Kommandofolge zum Starten und Bedienen von UDSMON	433
11.6 Beispiel	434
11.7 Beschreibung der UDS/SQL-Monitor-Ausgabe auf Datensichtstation	435
11.8 Beschreibung der UDS/SQL-Monitor-Ausgabe auf Drucker	451
11.9 Beschreibung der UDS/SQL-Monitor-Ausgabe in eine Datei	452
11.10 Übergabe von Monitorzählern an openSM2	459
12 Allgemeine Funktionen der Dienstprogramme	461
12.1 Dienstprogramme starten	462
12.2 Zuweisung des Datenbanknamens	464
12.3 Prüfung der UDS/SQL-Pubset-Deklaration	465
12.4 Automatische Realm-Erweiterung	466
12.5 Datenbankinformationen systemneutral ausgeben	467
12.5.1 Unterschiede der CSV-Ausgabe zur Ausgabe auf SYSLST	468
12.5.2 Steuern der CSV-Ausgabe mit DISPLAY-Anweisungen	469
12.5.3 CSV-Ausgabe in BOUTLOAD-Utility	470
12.5.4 Nutzungsbeispiele	471
12.6 Auftragsschalter	472
13 Einsatz von IQS	473
13.1 Von IQS verwendete Dateien, Module und Prozeduren	474
13.2 Hilfsmittel zur Fehlerdiagnose	476
14 Einsatz von UDS-D	477
14.1 Kurzbeschreibung des Produkts	478
14.1.1 Merkmale von UDS-D	479
14.2 Hinweise zum Einsatz von UDS-D	480
14.2.1 Entwerfen und Programmieren	481
14.2.2 Umstrukturieren	482
14.3 Arbeitsweise von UDS-D	483
14.3.1 Das Teiltransaktionskonzept	484
14.3.2 Primäre und sekundäre Teiltransaktionen	485
14.3.3 Multi-DB-Programme für verteilte Verarbeitung	486
14.3.4 Kommunikation zwischen Konfigurationen	487

14.4 Konfigurationsübergreifende Konsistenz mit dem Zwei-Phasen-Ende-Protokoll sichern	489
14.4.1 Ablauf des Zwei-Phasen-Ende-Protokolls bei Verteilung über UDS-D	490
14.4.2 Ablauf des Zwei-Phasen-Ende-Protokolls in UDS-D bei Verteilung über UDS-D und openUTM	492
14.4.3 Zustand PTC	494
14.4.4 Zustand PTC beenden	498
14.5 Datenbankbetrieb mit UDS-D	499
14.5.1 DBH beim Einsatz von UDS-D	500
14.5.2 Konfigurationsbezogener Kennwortschutz	503
14.5.3 Die Verteiltabelle	504
14.5.3.1 Aufbau der Eingabedatei für die Verteiltabelle	505
14.5.3.2 Eingabedatei einlesen	507
14.5.3.3 Aufbau der Verteiltabelle	508
14.5.3.4 Übersicht der DAL-Kommandos für Verteiltabellen	509
14.5.4 Zeitgesteuertes Überwachen von Verbindungen und Transaktionen	510
14.5.4.1 Überwachen der logischen Verbindung zu entfernten Anwenderprogrammen	511
14.5.4.2 Überwachen sekundärer Teiltransaktionen, die nicht im Zustand PTC sind	512
14.5.4.3 Überwachen sekundärer Teiltransaktionen, die im Zustand PTC sind	513
14.5.4.4 Überwachen primärer Teiltransaktionen, die durch den Datenbankadministrator beendet wurden	514
14.5.5 UDS-D-Betrieb starten und beenden	515
14.5.5.1 UDS-D-Betrieb starten	516
14.5.5.2 UDS-D-Betrieb beenden	517
14.5.5.3 UDS-D-Betrieb erneut starten	518
14.5.5.4 Beispiele für UDS-D-Betrieb	519
14.5.6 Auswirkungen des Zustands PTC	526
14.5.7 Mehrere Konfigurationen administrieren	531
14.5.7.1 Lokal administrieren	532
14.5.7.2 Zentral administrieren	533
14.5.8 DB-Konfiguration ändern - Hinweise für UDS-D	535
14.5.9 Einsatz der UDS/SQL-Konfiguration auf virtuellem Host	536
14.6 DBH-Ladeparameter für UDS-D	538
14.7 Datenbankadministratorsprache DAL für UDS-D	540
15 Anhang	542
15.1 Funktionscodes der DML-Anweisungen	543
16 Fachwörter	546
17 Abkürzungen	599
18 Literatur	602

Datenbankbetrieb

1 Einleitung

Das **Universelle Datenbank-System** UDS/SQL ist ein Datenbanksystem für hohe Durchsatzanforderungen. Es basiert auf dem Strukturkonzept von CODASYL, geht aber in seinen Möglichkeiten weit darüber hinaus und bietet koexistent auf dem gleichen Datenbestand das Relationenmodell an.

Zur Auswertung und Änderung der Daten stehen COBOL-DML, CALL-DML und SQL (ISO-konform) zur Verfügung. COBOL-DML-Anweisungen sind in die COBOL-Sprache integriert, die CALL-DML kann aus jeder Programmiersprache aufgerufen werden, SQL-Anweisungen können innerhalb von DRIVE-Programmen angewendet oder über eine ODBC-Schnittstelle genutzt werden.

UDS/SQL verhindert durch wirksame, flexibel einsetzbare Schutzmechanismen unberechtigte Zugriffe auf die Datenbank und garantiert Vertraulichkeit, Integrität und Verfügbarkeit. Diese Mechanismen sind mit dem Transaktionsmonitor openUTM abgestimmt.

Das Datensicherungskonzept von UDS/SQL schützt die Datenbestände wirkungsvoll vor Zerstörung und Verlust. Dabei werden UDS/SQL-eigene Mechanismen wie Logging veränderter Information mit BS2000-Funktionen wie DRV (Dual Recording by Volume) kombiniert.

Unter Einsatz des Zusatzproduktes UDS-D können Datenbestände in BS2000-Rechnernetzen verarbeitet werden. UDS/SQL garantiert dabei die netzweite Konsistenz der Daten. In Verbindung mit openUTM-D bzw. openUTM (Unix /Linux/Windows) lässt sich verteilte Transaktionsverarbeitung sowohl in BS2000-Rechnernetzen als auch im Verbund von BS2000 und anderen Betriebssystemen realisieren. UDS/SQL kann als Datenbank in Client-Server-Lösungen über SQL-Gateway bzw. über ODBC-Server eingesetzt werden.

UDS/SQL bietet durch seine Architekturmerkmale (z. B. Multitasking, Multithreading, DB-Cache) und durch seine vielseitigen Strukturierungsmöglichkeiten einen sehr hohen Durchsatz.

1.1 Konzept der UDS/SQL-Dokumentation

Dem Abschnitt „Wegweiser durch die Handbuchreihe“ entnehmen Sie, welche Handbücher und welche Teile daraus Ihrem Informationsbedürfnis entsprechen. Ein Fachwortverzeichnis liefert Kurzdefinitionen der im Text benutzten Fachwörter.

Außer über das Inhaltsverzeichnis können Sie die Antworten auf Ihre Fragen gezielt über das Stichwortverzeichnis und über Kolumnentitel nachschlagen.

Wegweiser durch die Handbuchreihe

Das Datenbanksystem UDS/SQL ist im Wesentlichen in fünf Handbüchern dokumentiert:

- UDS/SQL Entwerfen und Definieren
- UDS/SQL Anwendungen programmieren
- UDS/SQL Aufbauen und Umstrukturieren
- UDS/SQL Datenbankbetrieb
- UDS/SQL Sichern, Informieren und Reorganisieren

Weitere Handbücher zu UDS/SQL und Zusatzprodukten finden Sie im Kapitel "[Konzept der UDS/SQL-Dokumentation](#)".

Als Einstieg dient Ihnen das Handbuch „[Entwerfen und Definieren](#)“, Kapitel 2 und 3; hier werden erläutert:

- die Gründe für den Einsatz von Datenbanken
- das Datenbankmodell der CODASYL
- das Relationenmodell unter Berücksichtigung von SQL
- eine Abgrenzung der Modelle
- die Koexistenz der verschiedenen Datenbankmodelle bei einer UDS/SQL-Datenbank
- die charakteristischen Eigenschaften von UDS/SQL

Der weitere Umgang mit den Handbüchern richtet sich nach Ihren Vorkenntnissen und Aufgaben. Die [Tabelle 1](#) hilft Ihnen dabei, den richtigen Weg durch die Handbücher zu finden.

Beispiele

Angenommen, Ihre Aufgabe ist es, in COBOL-DML zu programmieren, so finden Sie in der zweiten Zeile der [Tabelle 1](#) unter „Aufgaben des Anwenders“ die Spalte „COBOL/CALL-DML Programm“. Im Handbuch „[Entwerfen und Definieren](#)“ brauchen Sie dann für Ihre Arbeit folgende Kapitel:

Allgemeines	E = zum Einstieg
Schema-DDL	D = zur Detailinformation
SSL	D = zur Detailinformation
Subschema-DDL	L = zum Lernen der Funktionen

Welche Kapitel Sie aus den weiteren Handbüchern brauchen, erfahren Sie in der gleichen Spalte.

Wenn Sie dagegen als Datenbankadministrator für den Datenbankbetrieb zuständig sind, orientieren Sie sich bitte in der Spalte „Verwalten und Bedienen“.

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Programmieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-D
Handbuch UDS/SQL Entwerfen und Definieren								
Einleitung	E	–	–	–	–	E	E	–
Allgemeines	E	E	E	E	E	E	–	–
Entwurf der Datenbank	E	–	–	–	–	–	–	–
Schema-DDL	L	D	–	L	L	–	–	–
SSL	L	D	–	L	L	–	–	–
Subschema DDL	L	L	–	L	L	–	–	–
relationales Schema	L	–	D	–	–	–	–	–
Aufbau der Seiten	D	–	–	D	D	–	–	–
Aufbau der Sätze und Tabellen	D	–	–	D	D	–	–	–
Nachschlageteil	S	–	–	S	–	–	–	–
Handbuch UDS/SQL Anwendungen programmieren								
Einleitung	–	E	–	–	–	E	E	–
Einführung	–	E	–	–	–	–	–	–
Transaktionskonzept	–	L	–	L	L	D	D	–
Currency-Tabelle	–	L	–	L	L	–	–	–
Funktionen der DML	D	L	–	L	–	–	–	–
Anwenden der DML	–	L	–	D	–	–	–	–
Nachschlageteil COBOL-DML	–	L	–	–	–	–	–	–
Nachschlageteil CALL-DML	–	L	–	–	–	–	–	–
Testen von DML-Funktionen mit DMLTEST	–	L	–	–	–	–	–	–
Handbuch UDS/SQL Aufbauen und Umstrukturieren								
Einleitung	–	–	–	E	–	E	E	–
Überblick	–	–	–	E	E	–	–	–
Datenbank aufbauen	–	–	–	L	–	–	–	–
Zugriffsberechtigungen festlegen	–	–	–	L	–	–	–	–
Daten speichern und entladen	D	–	–	L	–	D	–	–
Datenbank umstrukturieren	D	–	–	L	–	–	–	–
Datenbankobjekte umbenennen	D	–	–	L	–	–	–	–
Datenbank umstellen	D	–	–	L	–	–	–	–

Handbuch UDS/SQL Datenbankbetrieb								
Einleitung	-	-	-	-	E	E	E	-
Der Database Handler	-	-	-	-	L	-	-	D
Ladeparameter des DBH	-	-	-	-	L	-	-	D
Administration	-	-	-	-	L	-	-	D
Hochverfügbarkeit	-	-	-	-	E	-	-	-
Ressourcen-Erweiterung und Umorganisation im laufenden Betrieb	D	-	-	-	E	-	-	-
Datenbank sichern und wiederherstellen im Fehlerfall	D	-	-	D	L	D	-	D
Leistungsoptimierung	-	-	-	-	D	-	-	D
Nutzung der BS2000-Funktionalität	-	-	-	-	D	-	-	-
Der SQL-Vorgang	-	-	-	-	L	-	-	-
UDSMON	-	-	-	-	D	-	-	-
Einsatz von IQS	-	-	-	L	D	-	D	-
Einsatz von UDS-D	D	D	-	D	D	D	-	D
Funktionscodes der DML-Anweisungen	-	D	-	-	D	-	-	-
Handbuch UDS/SQL Sichern, Informieren und Reorganisieren								
Einleitung	-	-	-	-	E	E	E	-
Datenbank aktualisieren und rekonstruieren	D	-	-	D	L	D	-	-
Konsistenz einer Datenbank prüfen	-	-	-	-	L	-	-	-
Datenbankinformationen ausgeben	D	-	-	D	L	-	-	-
Online-Dienste durchführen	D	-	-	D	L	-	-	-
Datenbank reorganisieren	D	-	-	D	L	-	-	-
Wiederverwendung von freigewordenen Database Keys steuern	D	-	-	D	L	-	-	-

Weitere Handbücher								
UDS/SQL Meldungen	D	D	D	D	D	D	D	D
UDS/SQL Taschenbuch	S	S	–	S	S	S	S	S
IQS	–	–	–	D	D	–	L	–
ADILOS	–	–	–	–	D	–	L	–
KDBS	–	L	–	D	–	–	–	–
SQL für UDS/SQL Sprachbeschreibung	–	–	D	–	D	–	–	–

Tabelle 1: Wegweiser durch die Handbücher

E dient als Einstieg, wenn Sie bisher noch nichts mit UDS/SQL zu tun hatten

L in diesen Teilen der Handbücher steht das Lernen der Funktionen im Vordergrund

D hier können Sie hineinschauen, wenn Sie Detailinformationen suchen

S dient zum Nachschlagen von Syntaxregeln bei der praktischen Arbeit

Was Sie noch über die Handbücher wissen sollten

Literaturverweise finden Sie in Kurzform im Text. Finden Sie im Text z.B. (siehe Handbuch „Anwendungen programmieren“, CONNECT), so müssen Sie unter dem Stichwort CONNECT im Handbuch „Anwendungen programmieren“ nachschauen.

Der vollständige Handbuchtitel steht im Literaturverzeichnis.

UDS/SQL Meldungen

Das Handbuch enthält alle Meldungen, die UDS/SQL ausgibt. Die Meldungen sind aufsteigend nach Nummern oder bei einigen Dienstprogrammen alphabetisch sortiert.

UDS/SQL Taschenbuch

Das UDS/SQL-Taschenbuch enthält alle Übersichten zu den UDS/SQL-Funktionen und Formaten.

SQL für UDS/SQL Sprachbeschreibung

Das Handbuch beschreibt den SQL-DML-Sprachumfang von UDS/SQL.

Neben UDS/SQL-spezifischen Erweiterungen umfasst der beschriebene Sprachumfang die dynamische SQL als wesentliche Erweiterung der SQL-Norm.

1.2 Zielsetzung und Zielgruppen des Handbuchs

Das Handbuch ist für den Datenbankadministrator bestimmt, der die Aufgabe hat, den Datenbankbetrieb zu planen und zu überwachen.

Der Datenbankadministrator sollte Kenntnisse haben über alle Schritte des Datenbankentwurfs (Datenbankdesign, Schema-, Subschema- und SSL-Erstellen) und des Erstellens der DB-Anwenderprogramme.

Außerdem muss er über gute BS2000-Kenntnisse verfügen, das Transaktionskonzept von UDS/SQL kennen und grundlegende Kenntnisse über die Dateien einer UDS/SQL-Datenbank und die UDS/SQL-Dienstprogramme haben.

1.3 Konzept des Handbuchs

Was enthält dieses Handbuch?

Dieses Handbuch beschreibt Verwaltungs- und Bedienungsarbeiten, die für einen reibungslosen Datenbankbetrieb notwendig sind:

- Datenbankbetrieb mit dem Database Handler
- Session starten, administrieren und beenden
- Administration über UDSADM, DCAM und SEND-MSG
- Datenbank sichern und rekonstruieren
- Leistungsoptimierung
- Nutzung der BS2000-Funktionalität für UDS/SQL
- Der SQL-Vorgang
- DB-Betriebswerte und Statuswerte mit dem UDS/SQL-Monitor UDSMON ausgeben
- Allgemeine Funktionen der Dienstprogramme
- Einsatz von IQS
- Einsatz von UDS-D

Readme-Datei

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

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <https://bs2manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000

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

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

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen. Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

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

1.4 Änderungen gegenüber den Vorgänger-Handbüchern

In der folgenden [Tabelle 2](#) sind die wichtigsten Änderungen der Version UDS/SQL V2.9B gegenüber der Version V2.9 aufgeführt. Außerdem wird jeweils das Handbuch und das Kapitel genannt, in dem die Änderung beschrieben wird. Wird ein Thema in mehr als einem Handbuch beschrieben, dann wird zuerst das Handbuch aufgeführt, in dem das Thema vollständig beschrieben wird. In der Spalte „Handbuch“ bedeuten die Einträge:

ENT	Entwerfen und Definieren	DBB	Datenbankbetrieb
ANW	Anwendungen programmieren	SIR	Sichern, Informieren und Reorganisieren
AUF	Aufbauen und Umstrukturieren	MEL	Meldungen

Thema	Handbuch	Kapitel
Änderungen in V2.9A		
FIND-/FETCH-7 mit DESCENDING KEY: Aufhebung der Einschränkung		
Die Einschränkung für DESCENDING KEY entfällt	ANW	7
Satzreferenzen in COBOL-Programmen		
Die neue DDL-Anweisung GENERATE-REC-REF generiert ein Datenfeld und Bedingungsnamen für den Zugriff auf Satzreferenzen	AUF	3
Ändern der Einstellungen für ALOG-Dateien während die Datenbank in Benutzung ist		
Neues DAL-Kommando DISPLAY ALOG zum Anzeigen der ALOG-Einstellungen	DBB	4
Neue DAL-Kommandos MODIFY ALOG/MODIFY ALOG-RES und MODIFY-ALOG-SIZE zum Ändern der ALOG-Einstellungen	DBB	4
Änderung der Einschränkungen für die UDS-Online-Utility		
WAIT-FOR-TRANSACTION bietet die Möglichkeit zu warten, bis die gesperrte Quellseite von der sperrenden Transaktion freigegeben wird	SIR	8
Mit SET-RELOCATE-PARAMETERS kann auch für *INDEX-LEVELTABLE-PAGES ein Verhalten für den Fall, dass Seiten gesperrt sind, festgelegt werden (CLASH-HANDLING)	SIR	8
BRENAME mit After-Image-Logging: Die Funktion „Datenbank-Objekte umbenennen (BRENAME, BALTER)“ kann auch bei eingeschaltetem After-Image-Logging ausgeführt werden. Auf diese Weise können Logging-Lücken vermieden werden		
Neues Verhalten beim Umbenennungszyklus	AUF	7
Nach einem Umbenennungsprozess können Datenbank-Aktualisierungen durchgeführt werden	SIR	2
Größe für DBTT-Erweiterungen festlegen		
Neuer Operand EXT des DAL-Kommandos ACT DBTT-INCR	DBB	4
Die Ausgabe von BSTATUS enthält zusätzlich den Wert des EXT-Operanden	SIR	6

Neuer Datentyp FIXED REAL BINARY 63		
Erweiterung der Syntaxdarstellung	ENT	4, 9
Änderung von Meldungen, um den neuen Datentyp zu berücksichtigen	MEL	2, 4, 10
Änderungen in V2.9B		
BINILOAD von CSV-Dateien		
Erklärung, wie BINILOAD CSV-Dateien verarbeitet	AUF	2, 5
Änderungen an den Meldungen, um die neue Funktion zu berücksichtigen	MEL	3
BINILOAD mit Datensätzen variabler Länge		
BINILOAD kann mit Datensätzen variabler Länge arbeiten	AUF	5
Änderungen an den Meldungen, um die neue Funktion zu berücksichtigen	MEL	3
Defaultwerte für neue Felder in BALTER		
Neue BALTER-Anweisung, um Felder mit den angegebenen Werten zu füllen	AUF	6
Änderungen an den Meldungen, um die neue Funktion zu berücksichtigen	MEL	3
Anzahl der Extents in DB-JV		
Datenbank-Jobvariable wurde um die Anzahl der Extents der ALOG-Datei erweitert	DBB	9

Tabelle 2: Änderungen in V2.9 gegenüber V2.8

1.5 Darstellungsmittel

In diesem Abschnitt finden Sie die Erläuterung der Piktogramme für Warnhinweise und Hinweise sowie die Zeichenerklärung der Metasprache, wie sie zur Beschreibung von Syntaxregeln benutzt wird.

1.5.1 Warnhinweise und Hinweise

	Hinweis auf besonders wichtige Informationen
	Warnhinweis

1.5.2 Nicht-SDF-Darstellungsmittel

Sprachelement	Erklärung	Beispiel
<u>SCHLÜSSELWORT</u>	Schlüsselwörter sind durch Großbuchstaben mit Unterstreichung dargestellt. Sie müssen mindestens die unterstrichenen Teile des Schlüsselwortes angeben.	<u>DATABASE-KEY</u> <u>MANUAL</u>
WAHLWORT	Wahlwörter sind durch Großbuchstaben ohne Unterstreichung dargestellt. Wenn Sie Wahlwörter weglassen, hat das keinen Einfluss auf die Bedeutung einer Klausel.	NAME IS ALLOWED PAGES
<i>variable</i>	Variable sind mit kursiven Kleinbuchstaben dargestellt. Bei der Benutzung eines Formats, in dem eine Variable erscheint, müssen Sie einen aktuellen Wert an ihre Stelle setzen.	<i>feldname</i> <i>literal-3</i> <i>ganzzahl</i>
{ Entweder oder }	Genau einen der eingeklammerten Ausdrücke müssen Sie angeben. Eingerückte Zeilen gehören zum vorhergehenden Ausdruck. Die Klammer geben Sie nicht an.	{ <u>CALC</u> <u>INDEX</u> } { <u>VALUE IS</u> <u>VALUES ARE</u> }
[wahlweise]	Den eingeklammerten Ausdruck dürfen Sie weglassen. UDS /SQL benutzt dann Standardwerte. Die Klammern selbst geben Sie nicht an.	[IS <i>ganzzahl</i>] [<u>WITHIN</u> <i>realmname</i>]
... oder , ...	Den unmittelbar vorstehenden Ausdruck können Sie wahlweise mehrmals wiederholen. Die beiden Sprachelemente unterscheiden Wiederholungen mit Leerzeichen oder mit Komma als Trennzeichen.	<i>feldname</i> , ... { <u>SEARCH</u> KEY..... } ...
..... oder . .	Kennzeichnet Auslassungen aus Gründen der Übersichtlichkeit. Bei der Benutzung der Formate sind diese Auslassungen nicht erlaubt.	<u>SEARCH</u> KEY IS <u>RECORD NAME</u> . .
-	Den Punkt müssen Sie angeben, gefolgt von mindestens einem Leerzeichen. Die Unterstreichung geben Sie nicht an.	<u>SET SECTION.</u> 03 <i>feldname</i>-
Zwischenraum	Bedeutet, dass Sie mindestens ein Leerzeichen angeben müssen.	<u>USING CALC</u>

Tabelle 3: Zeichen der Metasprache

Alle übrigen Zeichen wie () , . ; „ = sind keine Metazeichen:
Sie müssen sie so angeben, wie sie im Format dargestellt sind.

1.5.3 SDF-Syntaxdarstellung

Diese Syntaxbeschreibung basiert auf der SDF-Version 4.7. Die Syntax der SDF-Kommando-/Anweisungssprache wird im Folgenden in 3 Tabellen erklärt.

Tabelle 4 : Metasyntax

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet, deren Bedeutung in [Tabelle 4](#) erläutert wird.

Tabelle 5 : Datentypen

Variable Operandenwerte werden in SDF durch Datentypen dargestellt. Jeder Datentyp repräsentiert einen bestimmten Wertevorrat. Die Anzahl der Datentypen ist beschränkt auf die in [Tabelle 5](#) beschriebenen Datentypen.

Die Beschreibung der Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 5](#) erläutert.

Tabelle 6 : Zusätze zu Datentypen

Für den Datentyp integer enthält [Tabelle 6](#) außerdem kursiv gesetzte Einheiten, die nicht Bestandteil der Syntax sind. Sie dienen lediglich als Lesehilfe.

Die Beschreibung der Zusätze zu den Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 6](#) erläutert.

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter. Einige Schlüsselwörter beginnen mit *	OPEN DATABASE COPY-NAME = <u>*NONE</u>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörigen Operandenwerten.	CONFIGURATION-NAME = <name 1..8>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird (Tabellen 27 und 31).	DATABASE = <dbname>
<u>Unterstreich</u>	Der Unterstrich kennzeichnet den Standardwert eines Operanden.	SCHEMA-NAME = <u>*STD</u>
/	Der Schrägstrich trennt alternative Operandenwerte.	CMD = <u>*ALL</u> / <dal-cmd>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	*KSET-FORMAT(...)

Einrückung	Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.	<pre> USER-GROUP-NAME = *KSET-FORMAT(...) *KSET-FORMAT(...) HOST = <host> </pre>
 	Der Strich kennzeichnet zusammengehörende Operanden einer Struktur. Sein Verlauf zeigt Anfang und Ende einer Struktur an. Innerhalb einer Struktur können weitere Strukturen auftreten. Die Anzahl senkrechter Striche vor einem Operanden entspricht der Strukturtiefe.	<pre> USER-GROUP-NAME = *ALL-EXCEPT(...) *ALL-EXCEPT(...) NAME = *KSET-FORMAT(...) *KSET-FORMAT(...) HOST = <host> ... </pre>
,	Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.	,SPACE = <u>STD</u>
list-poss(n):	Aus den <code>list-poss</code> folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muss sie in runde Klammern eingeschlossen werden.	NAME = list-poss(30): <subschemaname>

Tabelle 4: Metasyntax

Datentyp	Zeichenvorrat	Besonderheiten
alog-seq-nr	0..9	1..9 Zeichen

Tabelle 5: Datentypen

Datentyp	Zeichenvorrat	Besonderheiten
appl	A..Z 0..9 \$,#,@ Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @ Wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.
catid	A..Z 0..9	1..4 Zeichen; darf nicht mit der Zeichenfolge PUB beginnen

copyname	A..Z 0..9	1..7 Zeichen beginnend mit A..Z
c-string	EBCDIC-Zeichen	1..4 Zeichen ist in Hochkommata einzuschließen; der Buchstabe C kann vorangestellt werden; Hochkommata innerhalb des c-string müssen verdoppelt werden
csv-dateiname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen ist in Hochkomma einzuschließen
dal-cmd	A..Z 0..9 Bindestrich	1..64 Zeichen
date	0..9 Strukturkennzeichen: Bindestrich	Angabe eines Datums Eingabeformat: jjjj-mm-tt jjjj : Jahr; wahlweise 2- oder 4-stellig mm : Monat tt : Tag
dbname	A..Z 0..9	1..17 Zeichen beginnend mit A..Z
device	A..Z 0..9 \$,#,@ Strukturkennzeichen: Bindestrich	5..8 Zeichen beginnend mit A..Z oder 0..9 Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann und die einem im System verfügbaren Gerät entspricht. In der Dialogführung zeigt SDF die zulässigen Operandenwerte an. Hinweise zu möglichen Geräten sind der jeweiligen Operandenbeschreibung zu entnehmen.
host	A..Z 0..9 \$,#,@ Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @; wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt
integer	0..9,+,-	+ bzw. - kann nur erstes Zeichen sein.
kset	A..Z 0..9 \$,#,@ Strukturkennzeichen: Bindestrich	1..8 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z oder \$, #, @; wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.

name	A..Z 0..9 \$,#,@	1..8 Zeichen darf nicht nur aus 0..9 bestehen oder darf nicht mit einer Ziffer beginnen.
realmname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z
realmref	0..9	1..3 Zeichen
recordname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z Bei Satzarten mit Searchkey wird empfohlen, max. 26 Zeichen lange Namen zu verwenden, da ansonsten der implizit gebildete Setname (SYS_...) entsprechend der Begrenzung der Namenslänge von Sets gekürzt wird.
recordref	0..9	1..3 Zeichen
schemaname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z
setname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen A..Z
structured-name	A...Z 0...9 \$, #, @ Bindestrich	alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @
subschemaname	A..Z 0..9 Strukturkennzeichen: Bindestrich	1..30 Zeichen Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A..Z
time	0..9 Strukturkennzeichen: Doppelpunkt	Angabe einer Tageszeit { hh:mm:ss hh:mm hh } hh, mm, ss: bei Stunden, Minuten und Sekunden können führende Nullen weggelassen werden

userid	A..Z 0..9 \$,#,@	1..8 Zeichen beginnend mit A..Z oder \$,#,@ BPRIVACY: Wenn weniger als 8 Zeichen angegeben werden, wird intern mit Unterstrich auf 8 Zeichen aufgefüllt.
volume	A..Z 0..9 \$,#,@	1..6 Zeichen beginnend mit A..Z oder 0..9
x-string	Sedezimal: 00..FF	1..8 Zeichen ist in Hochkommata einzuschließen; der Buchstabe X muss vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.
x..y <i>unit</i>	<p>beim Datentyp integer: Intervallangabe</p> <p>x Mindestwert, der für integer erlaubt ist. x ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p>y Maximalwert, der für integer erlaubt ist. y ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>unit</i> nur bei integer: zusätzliche Einheiten. Folgende Angaben werden verwendet: <i>Mbyte</i> <i>Kbyte</i> <i>seconds</i></p>	

Tabelle 6: Zusätze zu Datentypen

2 Der Database Handler (DBH)

Auf eine UDS/SQL-Datenbank kann das Datenbank-Anwenderprogramm nicht direkt zugreifen, wie z.B. auf die Daten einer einfachen Datei. Vielmehr steuert der Database Handler (DBH) den Zugriff auf die Datenbank und nimmt dem Anwenderprogramm damit auch alle Verwaltungsarbeiten ab, die z.B. beim Speichern von Daten anfallen.

Den Database Handler gibt es in zwei Varianten:

- independent DBH
- linked-in DBH.

Der **independent DBH** ist als selbstständige Task-Familie realisiert. Er nimmt die Datenbankaufrufe der Anwenderprogramme entgegen, greift auf die Datenbank zu und transportiert die zu verarbeitenden Daten zwischen der Datenbank und dem Arbeitsbereich im Anwenderprogramm.

Außerdem überwacht der DBH alle Datenbankaktivitäten und führt den Dialog mit dem Datenbankadministrator, d. h. er meldet ihm aufgetretene Fehler und nimmt seine Anweisungen entgegen.

Der independent DBH ist multi-DB-fähig, d.h. er kann mehrere Datenbanken gleichzeitig verwalten. Ein Datenbank-Anwenderprogramm kann auch innerhalb ein und derselben Transaktion über diesen DBH auf alle Datenbanken der Multi-DB-Konfiguration zugreifen.

Das Dienstprogramm ONLINE-PRIVACY wird mit dem independent DBH genutzt.

Das Dienstprogramm UDS-ONLINE-UTILITY kann mit dem independent DBH und mit dem linked-in DBH genutzt werden.

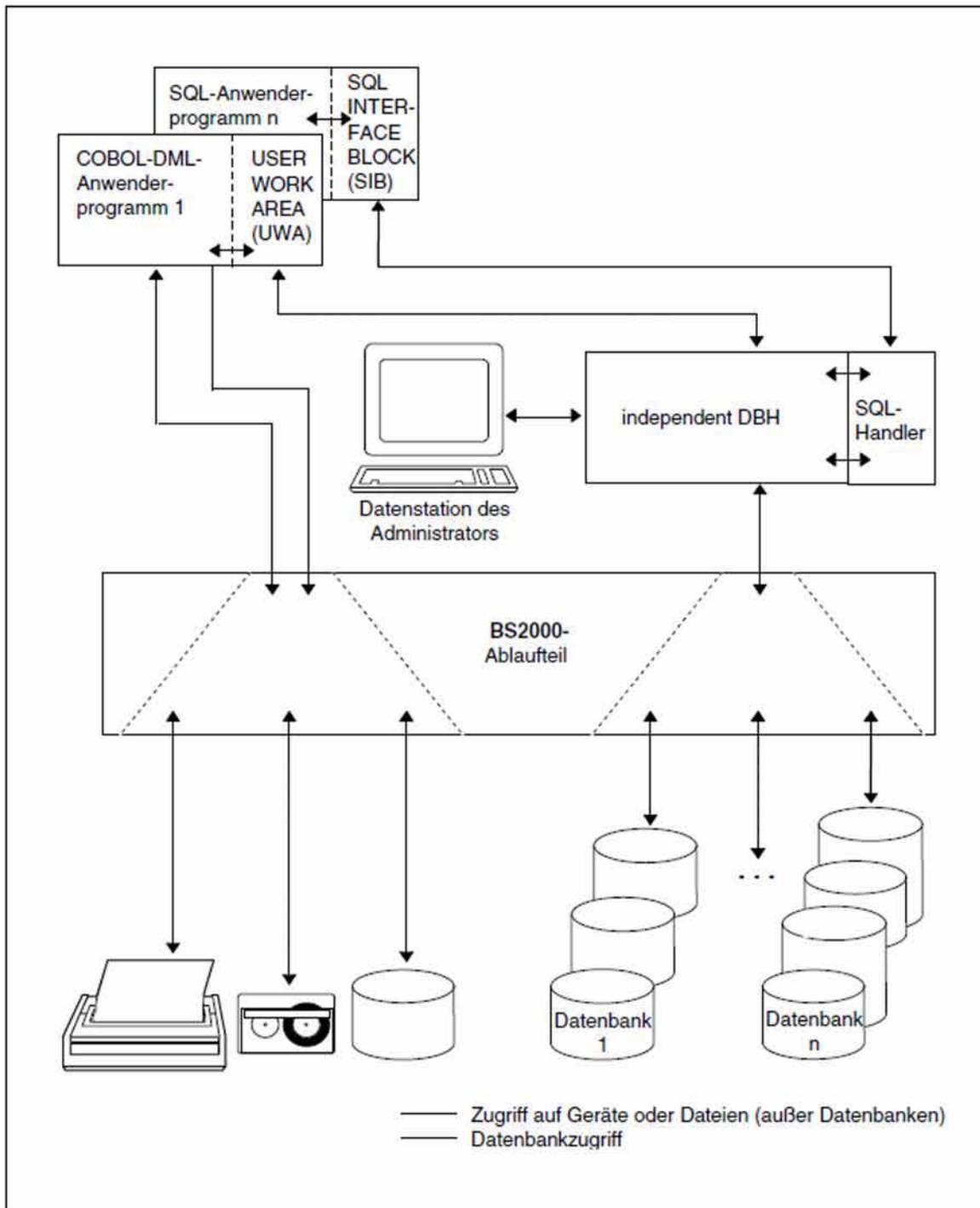


Bild 1: Prinzip des independent DBH

Der **linked-in DBH** ist kein selbstständiges Programm. Die Module des linked-in DBH werden in das Datenbank-Anwenderprogramm eingebunden; sie laufen als Teil dieses Programms. Soll vom Programm auf eine Datenbank zugegriffen werden, so ruft das Programm den DBH auf. Der DBH greift auf die Datenbank zu und schreibt Daten aus der UWA des Programms in die Datenbank oder liefert in der UWA angeforderte Daten aus der Datenbank ab.

Da bei Verwendung des linked-in DBH die Task-Kommunikation entfällt, die beim independent DBH zwischen den Tasks der Task-Familie und den Anwendertasks erforderlich ist, lässt sich die Laufzeit eines einzeln eingesetzten Anwenderprogrammes verkürzen.

i Der linked-in DBH kann keine SQL-Anweisungen bearbeiten.

Der linked-in DBH ist multi-DB-fähig, d.h. ein Datenbank-Anwenderprogramm, das mit dem linked-in DBH arbeitet, kann auf mehrere Datenbanken zugreifen. Diese Datenbanken sind während der linked-in Anwendung für jedes andere Programm gesperrt, es sei denn der DBH eröffnet sie nur lesend (SHARED-RETRIEVAL).

UDS/SQL-Dienstprogramme, die den DBH benötigen, arbeiten im Allgemeinen mit dem linked-in DBH, ebenso der DDL- und der SSL-Compiler; sie arbeiten dabei mit einer Mono-DB-Konfiguration, sprechen also immer nur eine Datenbank an.

Das Dienstprogramm UDS-ONLINE-UTILITY kann mit dem linked-in DBH und auch mit dem independent DBH genutzt werden.

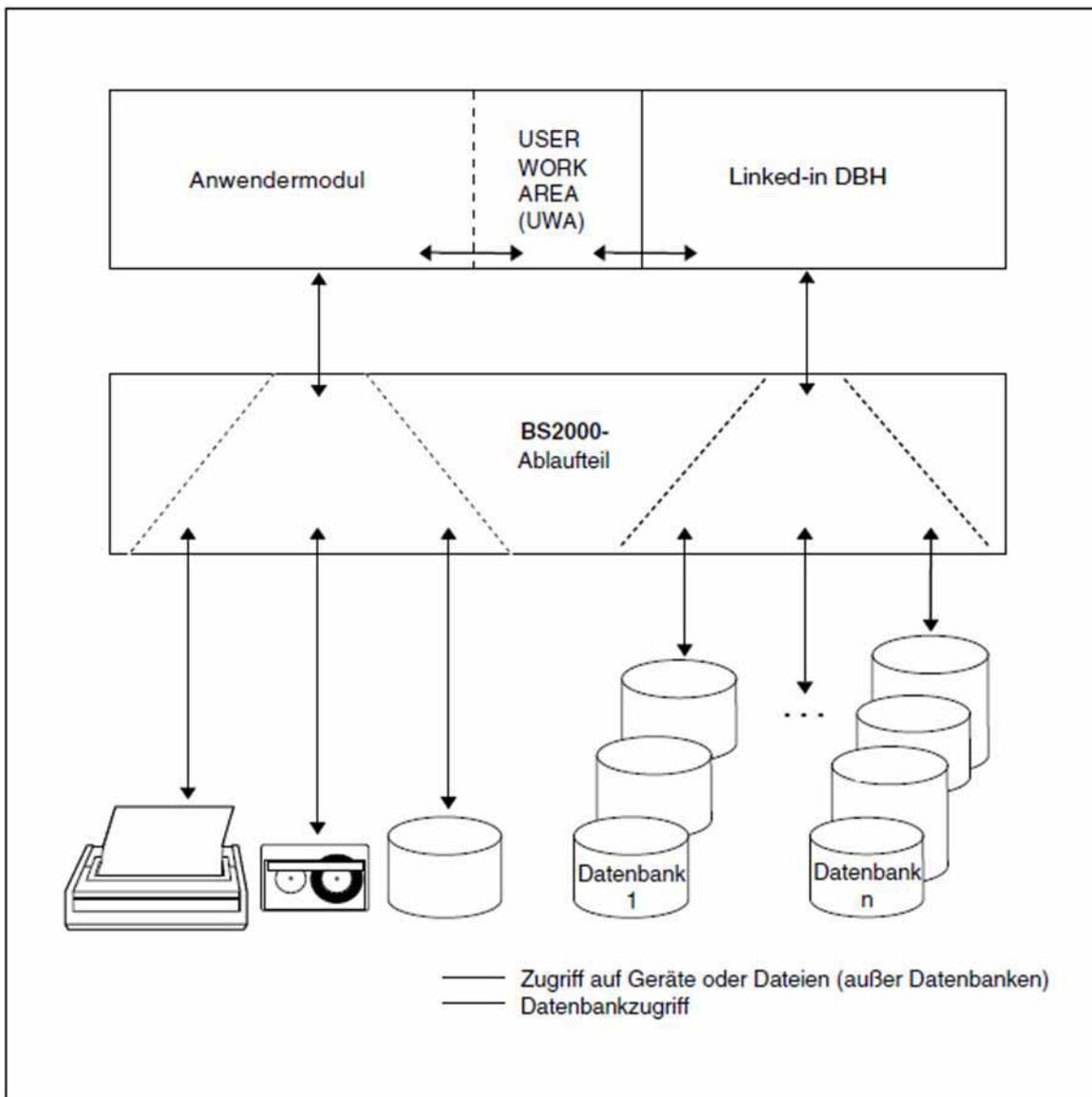


Bild 2: Prinzip des linked-in DBH

2.1 Arbeitsweise des independent DBH

Der independent DBH besteht aus mehreren Tasks (im Folgenden auch UDS/SQL-Tasks genannt):

- der Mastertask
- einer oder mehreren Servertasks
(ihre Anzahl bestimmt der Ladeparameter PP SERVERTASK)
- der Administratortask

Die **Mastertask** (Modul UDSSQL) bereitet beim DBH-Start die eigentliche Sessionphase vor und führt alle dafür notwendigen Prüfungen durch; ebenso steuert sie das DBH-Ende.

Beim DBH-Start lädt die Mastertask die Servertask(s) als ENTER-Job und beendet sie am DBH-Ende.

Die Mastertask hat darüberhinaus auch während der Session Sonderaufgaben durchzuführen, die einen reibungslosen Ablauf gewährleisten sollen.

Es wird empfohlen, aus Sicherheitsgründen die Mastertask in einem ENTER-Job zu starten.

Es ist auch möglich, die Mastertask im Dialog zu starten. In diesem Fall ist dann unbedingt darauf zu achten, dass die Mastertask nie unnötig unterbrochen wird und dass nach der Eingabe von BS2000-Kommandos immer RESUME-PROGRAM eingegeben wird.

Die **Servertask** (Modul UDSSUB) nimmt im Communication Pool den Auftrag entgegen und führt ihn aus. Dabei greift sie über den Common Pool auf die Datenbank zu und transportiert Daten, die das Anwenderprogramm in die Datenbank schreiben oder aus ihr lesen will, vom Communication Pool in den Common Pool und umgekehrt. Hat die Servertask den Aufruf fertig bearbeitet, so verständigt sie das Anwenderprogramm.

Die Eingaben und Ausgaben von den Servertasks werden standardmäßig asynchron durchgeführt (Ladeparameter PP IO=ASYN). Damit genügt es, auf einem Monoprozessor eine Servertask zu starten. Bei Multiprozessoren genügt es, pro Prozessor maximal eine Servertask zu starten (siehe [Abschnitt „Prozessornutzung mit independent DBH optimieren“](#)). Die Priorität der Servertasks muss dann geeignet eingestellt sein (siehe „RUN-PRIO“ im Kapitel ["DBH-Startkommandos"](#) und insbesondere [Abschnitt „Leistungsbezogene BS2000-Einstellungen“](#)).

Mit der **Administratortask** (Modul UDSADM) steuert der Datenbankadministrator die Session. Die Nutzung der Administratortask garantiert einen unterbrechungsfreien Betrieb der Mastertask sowie einen einfachen Anschluss des Datenbankadministrators an die Session (siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#)). Diese Form der Administration sollte im Allgemeinen verwendet werden.

Im **Communication Pool** ist für jede Transaktion ein Bereich reserviert, in dem bei einem Aufruf folgende Informationen gespeichert werden:

- die Transaktionskennung
- der Aufruf mit seinen Parametern
- die zu übertragenden Daten

Der **Common Pool** ist ein Speicherbereich, auf den alle UDS/SQL-Tasks des DBH zugreifen können. Er enthält unter anderem

- Systemtabellen
- die System Buffer Pools und die exklusiven Buffer Pools für Datenbankseiten.

Die System Buffer Pools enthalten 2-, 4- oder 8-Kbyte große Puffer, von denen jeder eine Datenbankseite aufnehmen kann. Die Größe der System Buffer Pools in Mbyte bestimmen Sie mit den Ladeparametern PP 2KB-BUFFER-SIZE, PP 4KB-BUFFER-SIZE bzw. PP 8KB-BUFFER-SIZE.

Zusätzlich zu den System Buffer Pools können Sie je Datenbank einen exklusiven Buffer Pool anlegen. Die Größe eines exklusiven Buffer Pools bestimmen Sie entweder mit dem Ladeparameter PP DBNAME oder mit dem DAL-Kommando ADD DB.

Jede Datenbankseite, die von einer Servertask bearbeitet werden soll, wird in einen Puffer des entsprechenden System Buffer Pools bzw. eines exklusiven Buffer Pools eingelesen und bleibt dort,

- bis die Session beendet wird oder
- bis die Datenbank mit DROP DB ausgeschlossen wird oder
- bis der Puffer für eine andere Datenbankseite benötigt wird.

Dadurch muss die Servertask nicht bei jedem DML-Aufruf auf die Datenbank zugreifen, sondern kann in den System Buffer Pools bzw. exklusiven Buffer Pools arbeiten.

Falls mit SQL gearbeitet wird, existiert für jede SQL-Anwendung ein Vorgang bei UDS/SQL (siehe [Kapitel „Der SQL-Vorgang“](#)). Bezüglich des Vorgangs existieren Systemtabellen im Communication Pool und im Common Pool.

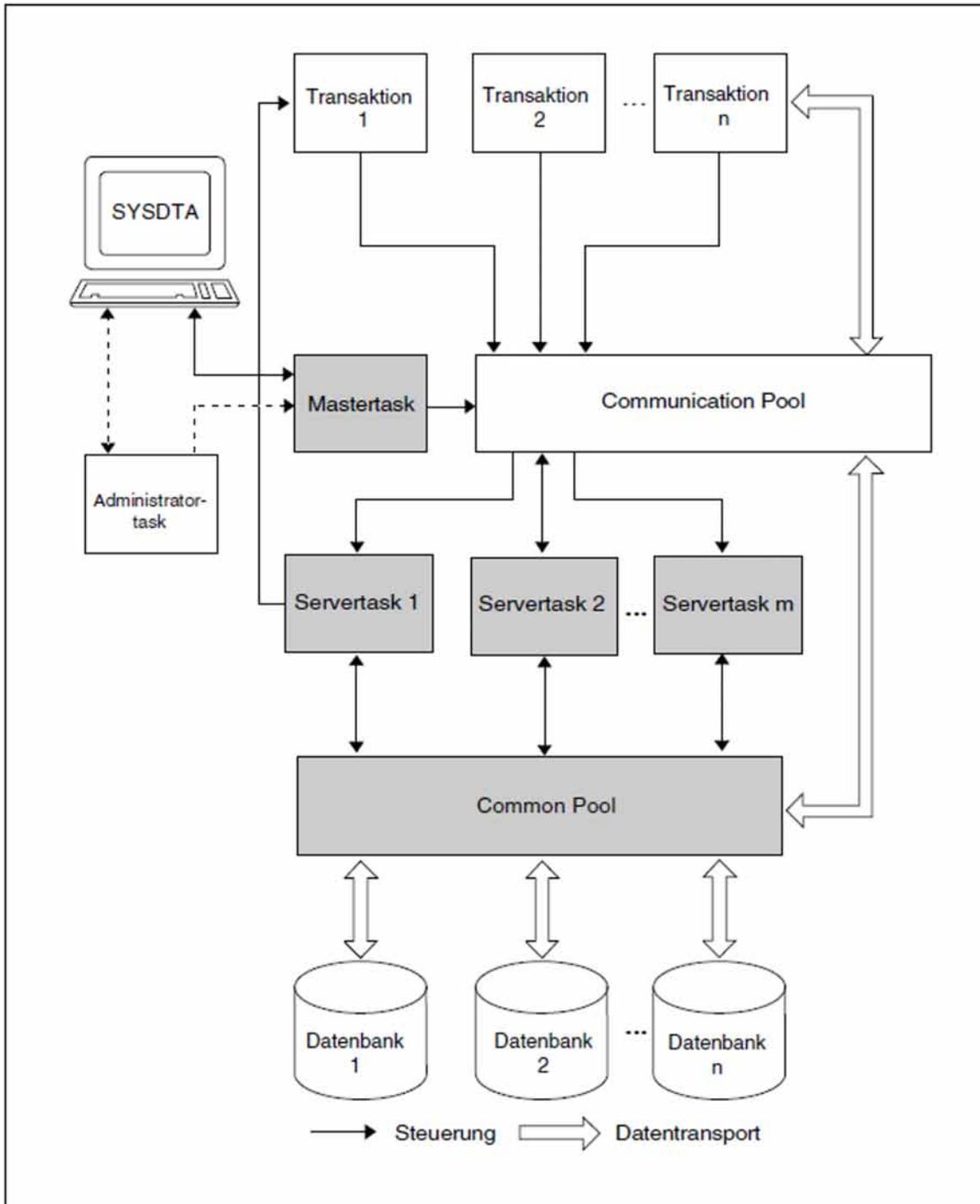


Bild 3: Der independent DBH

Bearbeiten einer COBOL-DML-Anweisung durch den independent DBH

Im Folgenden wird am Beispiel eines COBOL-Programms die Bearbeitung einer DML-Anweisung durch den independent DBH kurz dargestellt, wobei das folgende Bild den Ablauf verdeutlichen soll:

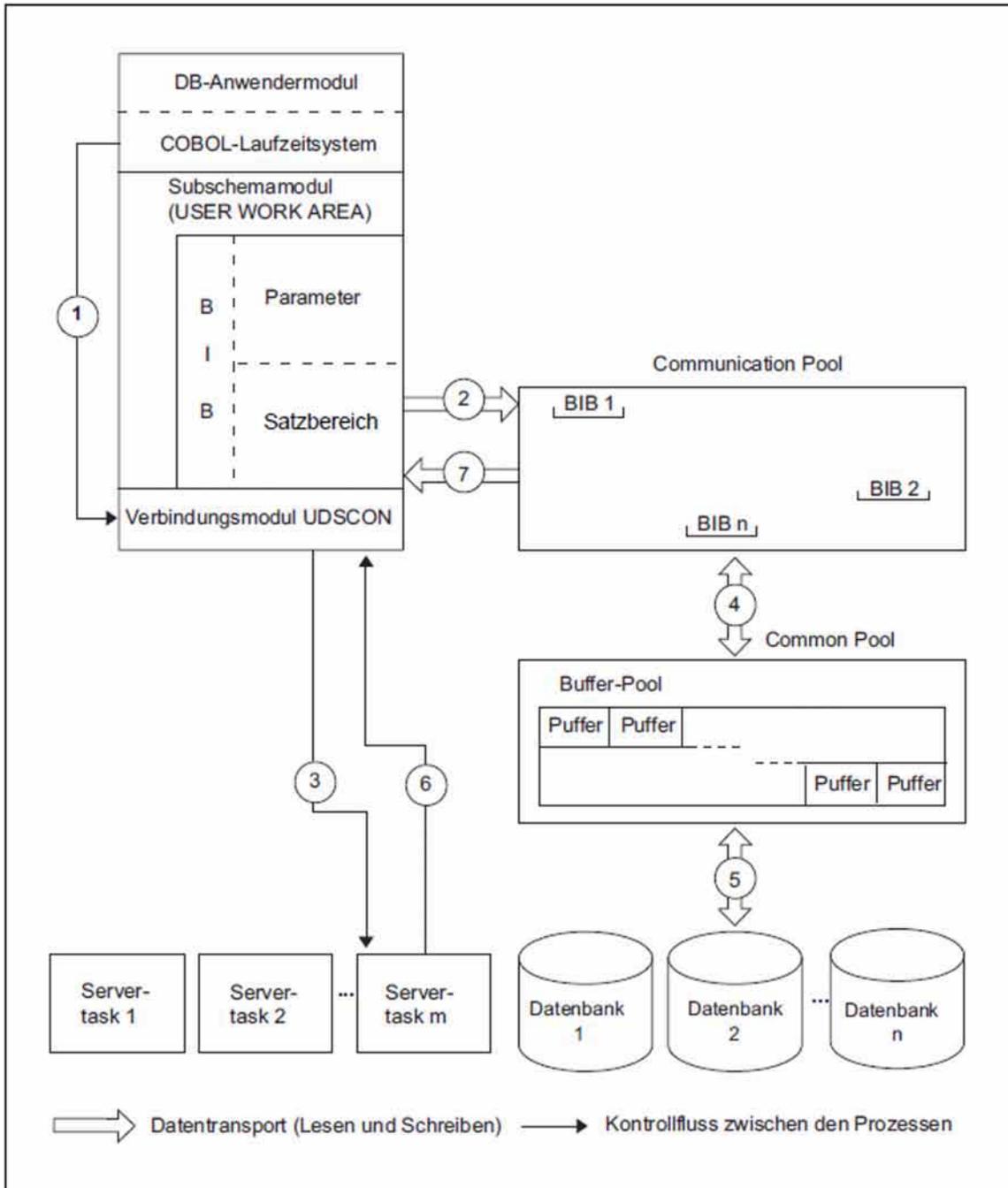


Bild 4: Abwicklung einer COBOL-DML-Anweisung durch den independent DBH

1. Dem Anwenderprogramm steht durch das Subschemasmodul ein Arbeitsbereich, der Satzbereich, zur Verfügung. Der Satzbereich ist gleichzeitig Teil des BASE INTERFACE BLOCK (BIB), der Schnittstelle zwischen UDS/SQL und dem jeweiligen Datenbankanwender.
In dem Satzbereich des BIB übergeben Sie die zu einer DML-Anweisung gehörenden Daten.
In den Parameterbereich des BIB setzt das COBOL-Laufzeitsystem die notwendigen Parameter der DML-Anweisung ein.
Über einen CALL-Aufruf übergibt das COBOL-Laufzeitsystem dem Verbindungsmodul die Adresse des BIB.
2. Das Verbindungsmodul überträgt den BIB in den Communication Pool, in dem zu Beginn einer Transaktion dafür ein Bereich reserviert wird.
3. Dieser BIB wird einer Servertask übergeben und bearbeitet.
Ist keine Servertask verfügbar, so muss der BIB warten, bis eine Servertask frei wird. Das Verbindungsmodul versetzt sich in den Wartezustand bis die DML-Anweisung bearbeitet worden ist.
4. Zur Bearbeitung der DML-Anweisung greift die Servertask auf den BIB im Communication Pool und die Buffer Pools im Common Pool zu.
Soll die DML-Anweisung Daten aus der Datenbank lesen, so überträgt die Servertask die in den Buffer Pools stehenden Daten in den entsprechenden BIB im Communication Pool. Umgekehrt beim Schreiben: soll die DML-Anweisung Daten in die Datenbank schreiben, so überträgt die Servertask die im BIB stehenden Daten in die Buffer Pools. Die Servertask schreibt aus den Buffer Pools einen Puffer jedoch erst dann in die Datenbank zurück, wenn eine andere Datenbankseite den Platz im Puffer benötigt.
5. Befindet sich die Datenbankseite, in der die DML-Anweisung die Daten des BIB schreiben soll oder aus der sie Daten lesen soll, nicht in den Buffer Pools, so wird auf der angegebenen Datenbank die entsprechende Datenbankseite gesucht und in die Buffer Pools eingelesen.
6. Nach der Bearbeitung der DML-Anweisung aktiviert die Servertask die zur Transaktion gehörende Anwendertask und hebt damit den Wartezustand des Verbindungsmoduls auf.
7. Das Verbindungsmodul schreibt den bearbeiteten BIB aus dem Communication Pool in das Anwenderprogramm zurück.
Das Verbindungsmodul gibt die Steuerung an das Anwenderprogramm zurück. Das Anwenderprogramm kann nun die Daten in dem Satzbereich bearbeiten.

Bearbeiten einer CALL-DML-Anweisung durch den independent DBH

Im Folgenden wird am Beispiel eines CALL-DML-Programms die Bearbeitung einer DML-Anweisung dargestellt, wobei das folgende Bild den Ablauf verdeutlichen soll.

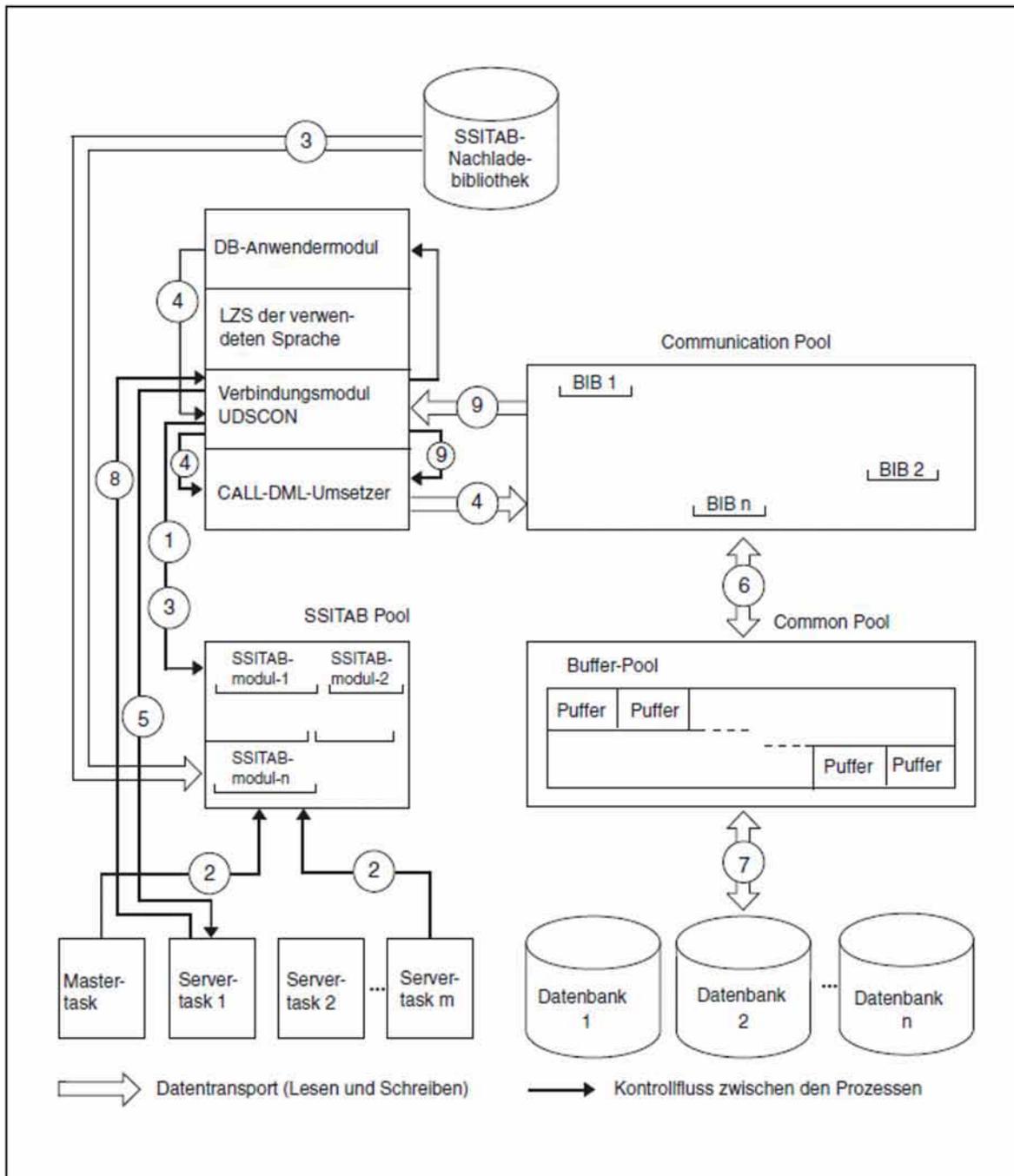


Bild 5: Abwicklung einer CALL-DML-Anweisung durch den independent DBH

Die Rolle des SSITAB-Pools und des Verbindungsmoduls UDSCON wird hier erläutert.

1. Ist das Anwenderprogramm das erste CALL-DML-Anwenderprogramm der laufenden UDS/SQL-Session, so wird vom Verbindungsmodul der SSITAB-Pool eingerichtet.
2. Alle Prozesse der UDS/SQL-Task-Familie müssen sich zu diesem Zeitpunkt ebenfalls an den SSITAB-Pool anschließen. Gelingt dies, ist ab diesem Zeitpunkt das Arbeiten mit der CALL-DML bis zum Session-Ende möglich. Der SSITAB-Pool bleibt bis zum Session-Ende bestehen.
3. Jedes Anwenderprogramm, das mit der CALL-DML arbeitet, hängt sich an den Pool an und überprüft, ob sich das SSITAB-Modul bereits im SSITAB-Pool befindet. Ist dies nicht der Fall, wird versucht, das SSITAB-Modul aus einer Bibliothek nachzuladen, die vor dem Start des Anwenderprogramms mit dem Linknamen \$UDSSSI zugewiesen wurde. Detaillierte Informationen zum Nachladen von SSITAB-Modulen finden Sie im Kapitel "[DBH-Startkommandos](#)" und im Handbuch „[Anwendungen programmieren](#)“, Abschnitt „[UDS/SQL-TIAM-Anwendungen binden, laden und starten](#)“.
4. Wird nun vom Anwendermodul an der CALL-Schnittstelle ein DML-Auftrag abgesetzt, so übernimmt ihn das Verbindungsmodul. Das Verbindungsmodul gibt den Auftrag an den CALL-DML-Umsetzer weiter, der die Anforderung in einen UDS/SQL verständlichen BIB umsetzt. Der BIB wird dann vom Verbindungsmodul in den Communication Pool geschrieben.
5. Dieser BIB wird einer Servertask übergeben und bearbeitet. Ist keine Servertask verfügbar, so muss der BIB warten, bis eine Servertask frei wird. Das Verbindungsmodul versetzt sich in den Wartezustand bis die DML-Anweisung bearbeitet worden ist. Das Anwenderprogramm bekommt erst danach die Kontrolle zurück.
6. Zur Bearbeitung der DML-Anweisung greift die Servertask auf den BIB im Communication Pool und die Buffer Pools im Common Pool zu.
Soll die DML-Anweisung Daten aus der Datenbank lesen, so überträgt die Servertask die in den Buffer Pools stehenden Daten in den entsprechenden BIB im Communication Pool. Umgekehrt beim Schreiben: soll die DML-Anweisung Daten in die Datenbank schreiben, so überträgt die Servertask die im BIB stehenden Daten in die Buffer Pools. Die Servertask schreibt im Allgemeinen aus den Buffer Pools einen Puffer jedoch erst dann in die Datenbank zurück, wenn eine andere Datenbankseite den Platz im Puffer benötigt.
7. Befindet sich die Datenbankseite, in der die DML-Anweisung die Daten des BIB schreiben soll oder aus der sie Daten lesen soll, nicht in den Buffer Pools, so wird auf der angegebenen Datenbank die entsprechende Datenbankseite gesucht und in die Buffer Pools eingelesen.
8. Nach der Bearbeitung der DML-Anweisung aktiviert die Servertask die zur Transaktion gehörende Anwendertask und hebt damit den Wartezustand des Verbindungsmoduls auf.
9. Ist der BIB von UDS/SQL bearbeitet, übernimmt das Verbindungsmodul diesen, und gibt ihn an den CALL-DML-Umsetzer weiter, der die Ergebnisse aus dem BIB in die benutzerverständliche CALL-Schnittstelle umsetzt.

2.2 Arbeitsweise des linked-in DBH

Der linked-in DBH ist kein selbstständiges Programm; seine Module werden zum Daten-bank-Anwenderprogramm geladen und laufen als Teil dieses Programms.

Der linked-in DBH ist multi-DB-fähig; ein Anwenderprogramm, das mit dem linked-in DBH arbeitet, kann also auf mehrere Datenbanken gleichzeitig zugreifen. Das Programm belegt diese Datenbanken exklusiv, d.h. kein anderes Programm kann gleichzeitig mit ihnen arbeiten, es sei denn der DBH eröffnet sie nur lesend (SHARED-RETRIEVAL).

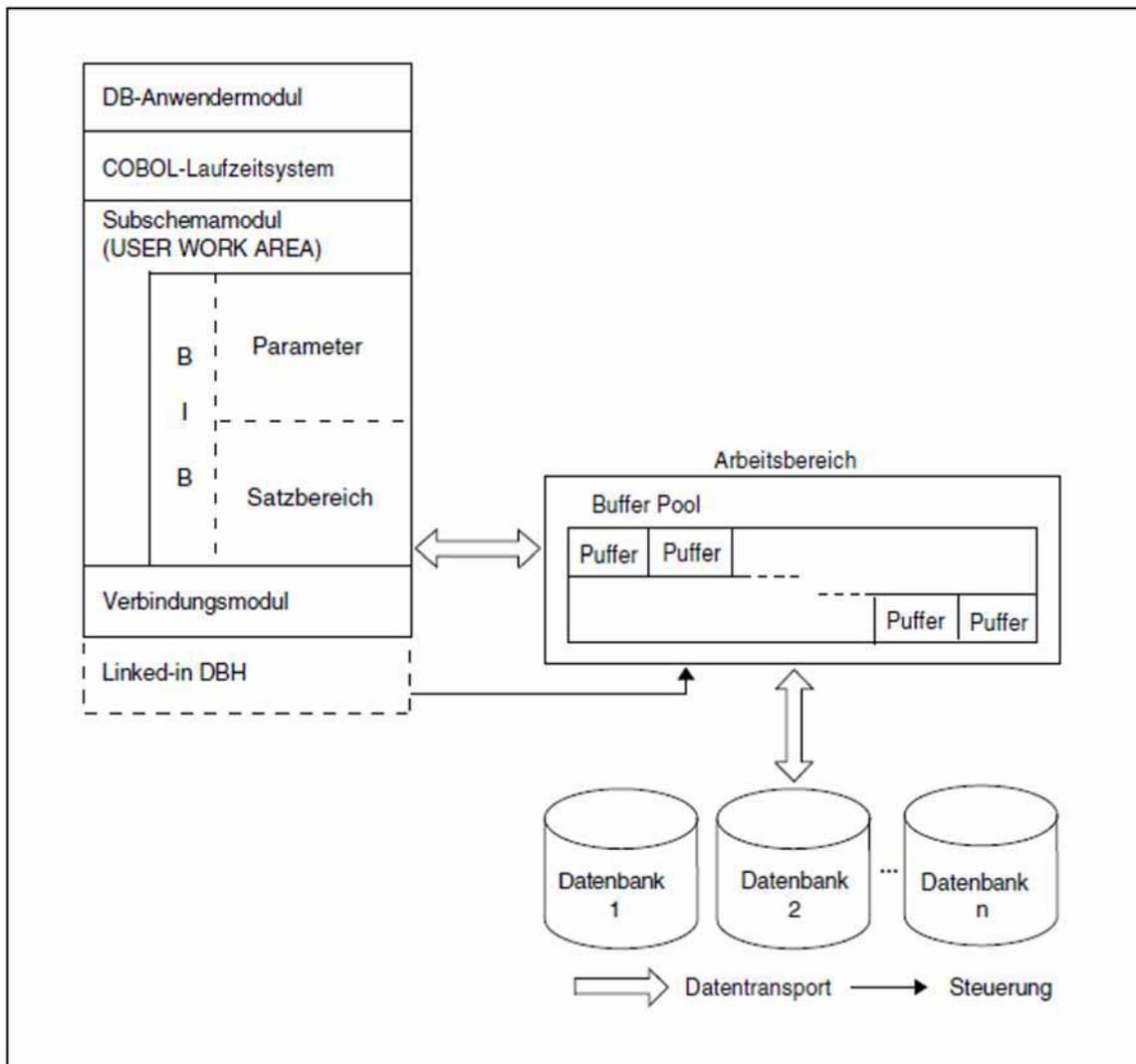


Bild 6: Der linked-in DBH

In das Anwenderprogramm müssen Sie ein spezielles Verbindungsmodul mit einbinden (UDSLNKL oder UDSLNKA, siehe Handbuch „Anwendungen programmieren“, Abschnitt UDS/SQL-TIAM-Anwendungen binden, laden und starten). Dieses sorgt beim Starten des Anwenderprogramms dafür, dass der dynamische Bindelader DBL die Module des linkedin DBH nachlädt. Gearbeitet wird, wie beim independent DBH, nicht direkt auf der Datenbank, sondern in den Buffer Pools des DBH-Arbeitsbereichs, der den Buffer Pools im Common Pool des

independent DBH entspricht.

Dieser Arbeitsbereich liegt beim linked-in DBH allerdings nicht in einem von mehreren Prozessen gemeinsam benutzbaren Speicherbereich (Common Memory), sondern im Speicherbereich des zugehörigen Prozesses.

Im Folgenden wird am Beispiel eines COBOL-Programms die Bearbeitung einer DML-Anweisung durch den linked-in DBH kurz dargestellt. Das folgende Bild soll den Ablauf verdeutlichen:

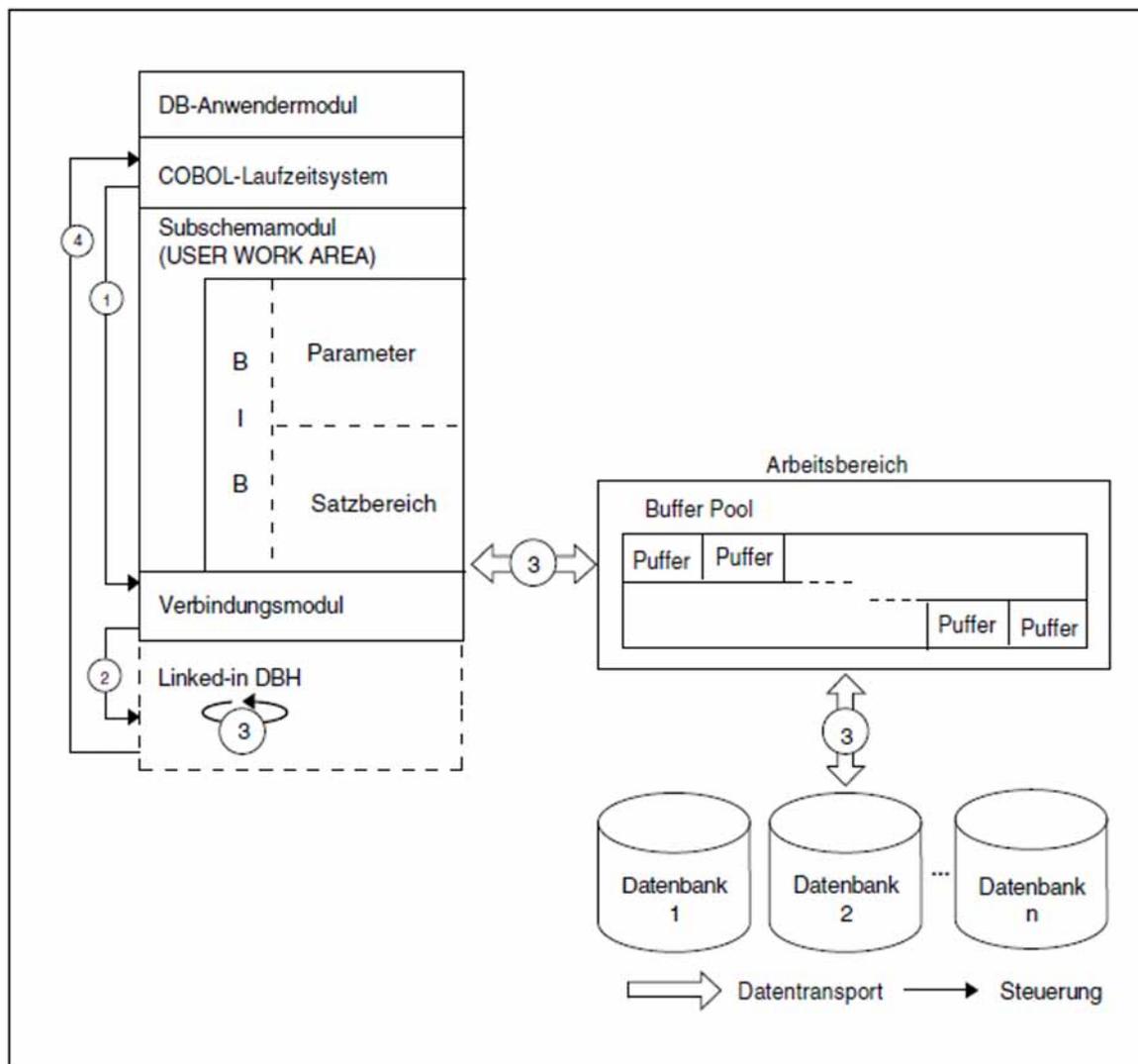


Bild 7: Abwicklung einer COBOL-DML-Anweisung durch den linked-in DBH

- Das Subschemasmodul enthält den Satzbereich, der gleichzeitig Teil des BASE INTERFACE BLOCK (BIB) ist. Der Satzbereich ist über die SUBSCHEMA SECTION durch das Anwenderprogramm adressierbar.
Das DB-Anwendermodul überträgt die zur jeweiligen DML-Anweisung gehörenden Daten in den Satzbereich des BIB.
Das COBOL-Laufzeitsystem setzt in den Parameterbereich des BIB die zur Bearbeitung des DML-Aufrufs notwendigen Angaben ein und übergibt die Adresse des BIB an das Verbindungsmodul.
- Das Verbindungsmodul ruft die Verarbeitungsmodule des linked-in DBH auf und übergibt die Adresse des BIB.

3. Der DBH bearbeitet den DML-Aufruf, d.h.
 - soll die DML-Anweisung Daten aus der Datenbank lesen, so versorgt der DBH den BIB mit den entsprechenden Daten. Dazu greift der DBH auf die Buffer Pools zu und sucht dort die entsprechende Datenbankseite. Befindet sich diese noch nicht in den Buffer Pools, so liest der DBH sie aus der Datenbank ein.
 - soll die DML-Anweisung Daten in die Datenbank schreiben, so schreibt der DBH die Daten aus dem BIB in die entsprechende Datenbankseite in den Buffer Pools. Befindet sich die noch nicht in den Buffer Pools, so liest der DBH sie aus der Datenbank ein.
4. Hat der DBH den DML-Aufruf fertig bearbeitet, so gibt er die Steuerung an das Anwenderprogramm zurück.

2.3 Die Session

Eine Session ist der Zeitraum zwischen dem DBH-Start, d.h. Session-Beginn und dem **nnormalen** DBH-Ende, dem Session-Ende. Während der Session kann eine Unterbrechung auftreten, der Session-Abbruch; der Session-Wiederanlauf gilt als Fortsetzung der Session.

Die Datenbanken, die während einer Session einem DBH zugeschaltet sind, bilden eine DB-Konfiguration.

Die DB-Konfiguration der Session bestimmen Sie entweder mit Ladeparametern des DBH (siehe [Kapitel „Ladeparameter des DBH“](#)) oder, nach dem DBH-Start, mit Kommandos der Datenbankadministratorsprache (siehe [Abschnitt „Die Datenbankadministratorsprache DAL“](#)).

Bilden mehrere Datenbanken die DB-Konfiguration, so spricht man von einer Multi-DB-Konfiguration; bei nur einer Datenbank spricht man von einer Mono-DB-Konfiguration.

Während einer Session mit dem **independent DBH** können Sie mit DAL-Kommandos die DB-Konfiguration ändern, d.h. Sie können Datenbanken an- und ausschließen, unabhängig von der Konsistenz. Inkonsistente Datenbanken werden bei Anschluss durch einen Warmstart konsistent gemacht.

Das Ändern der DB-Konfiguration nach Session-Abbruch wird Rekonfiguration genannt (siehe „DB-Konfiguration nach Session-Abbruch ändern (Rekonfiguration)“ im [„DB-Konfiguration festlegen und ändern“](#) sowie [Abschnitt „Fehlerarten und ihre Behandlung“](#)).

2.3.1 DBH starten

- independent DBH
- linked-in DBH
- DBH-Startkommandos
- Meldungen beim DBH-Start
- Beispiele

2.3.1.1 independent DBH

Den independent DBH starten Sie durch Laden der Mastertask mit dem Kommando START-UDS-DBH (weitere Start-Möglichkeiten siehe unter Punkt 11. im Kapitel "DBH-Startkommandos") und beenden Sie mit dem DAL-Kommando CLOSE. Während der Session haben Sie die Möglichkeit, mit Kommandos der Datenbankadministratorsprache steuernd in den Ablauf der Session einzugreifen.

Die DAL-Kommandos können Sie über die Administrations-Schnittstelle eingeben (siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#) und [Abschnitt „UDS/SQL über DCAM administrieren“](#)).

Syntax des Kommandos START-UDS-DBH:

START-UDS-DBH

```

VERSION = *STD /<product-version>
,MONJV = *NONE / <filename 1..54 without-gen-vers>
,CPU-LIMIT = *JOB-REST / <integer 1..32767 seconds>
,RESIDENT-PAGES = [ *PAR AMETERS](...)
  [ *PAR AMETERS](...)
    | MINIMUM = *STD / <integer 0..32767 4Kbyte>
    | ,MAXIMUM = *STD / <integer 0..32767 4Kbyte>

```

VERSION =

Produktversion des DBH, die gestartet werden soll.
Derzeit wird nur die Angabe des Werts *STD unterstützt.

VERSION = *STD

Keine explizite Angabe der Produktversion. Die Produktversion wird folgendermaßen ausgewählt:

1. Die mit dem Kommando /SELECT-PRODUCT-VERSION vorgegebene Version.
2. Die höchste mit IMON installierte DBH-Version.

VERSION = <product-version>

Explizite Angabe der Produktversion in der Form mm.n[a[kk]].

Es wird empfohlen, die Version immer vollständig anzugeben, z.B. 02.9B00, um den einfachen Versionswechsel auch bei Korrekturlieferungen zu ermöglichen.

MONJV =

Angabe einer Monitor-Jobvariablen zur Überwachung des DBH-Laufs.

MONJV = *NONE

Es wird keine Monitor-Jobvariable verwendet.

MONJV = <filename 1..54 without-gen-vers>

Name der zu verwendenden Jobvariablen.

Während des Programmlaufs setzt das System die Jobvariable auf folgende Werte:

\$R Programm läuft

\$T Programm erfolgreich beendet

\$A Programm fehlerhaft beendet

CPU-LIMIT =

Maximale CPU-Zeit in Sekunden, die der DBH beim Ablauf verbrauchen darf.

CPU-LIMIT = *JOB-REST

Es soll die verbleibende CPU-Zeit des BS2000-Jobs für die Aufgabe verwendet werden.

CPU-LIMIT = <integer 1..32767 seconds >

Es soll nur die angegebene Zeit verwendet werden.

RESIDENT-PAGES = *PARAMETERS(...)

Anzahl residente Speicherseiten, die für den DBH-Lauf benötigt werden.

Der Operand muss angegeben werden, wenn im Programm mit einem CSTAT-Makro (siehe Handbuch „[Makroaufrufe an den Ablaufteil](#)“) Seiten resident gemacht werden sollen. Die zulässige Anzahl an residenten Speicherseiten kann vom Operator beeinflusst werden. Fehlt der Operand (entspricht MIN=*STD,MAX=*STD), so werden die Speicheranforderungen dem Anfangssatz des Programms entnommen, wozu die Datei eröffnet werden muss.

MINIMUM = *STD / <integer 0..32767 4Kbyte >

Minimal benötigte Anzahl an residenten Speicherseiten.

MAXIMUM = *STD / <integer 0..32767 4Kbyte >

Maximal benötigte Anzahl an residenten Speicherseiten.

2.3.1.2 linked-in DBH

Den linked-in DBH starten Sie durch Laden und Starten des Anwenderprogramms, das den linked-in DBH verwendet. Dabei wird der linked-in DBH automatisch gestartet bei Verwendung von:

COBOL-DML

beim Starten des DB-Anwenderprogramms

ausschließlich CALL-DML

beim ersten READYC-Aufruf

KDBS

beim ersten OPDB-Aufruf

IQS

beim Starten des IQS-Ladmoduls C.L.IQS.

Der DBH wird mit der Beendigung des Anwenderprogramms beendet.

2.3.1.3 DBH-Startkommandos

i In der folgenden Kommandoübersicht wird als Standardfall angenommen, dass der DBH einer UDS/SQL-Version genutzt wird, die mit IMON installiert wurde und im SCI (Software Configuration Inventory) zentral verwaltet wird. Unter der Überschrift „Privatinstallation“ sind jeweils Kommandos aufgeführt, die nicht im Standardfall, sondern nur bei Nutzung einer privat installierten, nicht im SCI verwalteten UDS/SQL-Version relevant sind.

Nr.	Kommandos	independent DBH	linked-in DBH
1.	<code>/SET-FILE-LINK LINK-NAME=DATABASE , FILE-NAME=<i>konfigurationsname</i></code>	X	X
2.	<code>[/CREATE-FILE FILE-NAME=<i>konfname</i>.TEMPO.<i>nnn</i> , SUPPORT=*PRIVATE-DISK (VOLUME=<i>priv-vsn</i> , DEVICE-TYPE=<i>gerät</i>[, SPACE=...])]</code>	X	X
3.	<code>[/CREATE-FILE FILE-NAME=<i>konfname</i>.SLF , SUPPORT=*PRIVATE-DISK (VOLUME=<i>priv-vsn</i> , DEVICE-TYPE=<i>gerät</i>[, SPACE=...])]</code>	X	X
4.	<code>/CREATE-FILE FILE-NAME=<i>konfname</i>.DBSTAT , SUPPRESS-ERRORS=*FILE-EXISTING /CREATE-FILE FILE-NAME=<i>konfname</i>.DBSTAT.SAVE , SUPPRESS-ERRORS=*FILE-EXISTING</code>	X	X
5.	<code>[/MODIFY-JOB-SWITCHES OFF=28]</code>		X
6.	<code>[/ASSIGN-SYSDTA TO={ *LIBRARY-ELEMENT(...) *SYSCMD }]</code>	X	X
	<code>[/ADD-FILE-LINK LINK-NAME=PPFILE , FILE-NAME=<i>dateiname</i>]</code>	X	X
7.	<code>[/SET-FILE-LINK LINK-NAME=UDSDBHJV , FILE-NAME=<i>jvname</i>]</code>	X	X
8.	<code>[/SET-JV-LINK LINK-NAME=UDSPS01 , JV-NAME=<i>jvname</i>]</code>	X	X
9.	<code>[/ADD-FILE-LINK LINK-NAME=\$UDSSSI , FILE-NAME=<i>SSITAB-nachladebibliothek</i>]</code>		X
	<code>[/ADD-FILE-LINK LINK-NAME=\$UDSPLEX , FILE-NAME=<i>PLITAB-nachladebibliothek</i>]</code>		X
10.	<code>[/ADD-FILE-LINK LINK-NAME=\$UDSKONF , FILE-NAME=<i>UDSTRTAB-nachladebibliothek</i>]</code>	X	X

11.	/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL ,VERSION= <i>nn.nann</i> ,SCOPE=*TASK	X	X
	[/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D ,VERSION= <i>nn.nann</i> ,SCOPE=*TASK]	X	
<i>Privatinstallation (zusätzliche Kommandos)</i>			
	[/ADD-FILE-LINK LINKNAME=\$UDSLIB ,FILE-NAME= <i>UDS/SQL-modulbibliothek</i>]	X	X
	[/ADD-FILE-LINK LINKNAME=\$UDSDLIB ,FILE-NAME= <i>UDS-D-modulbibliothek</i>]	X	
	[/ADD-FILE-LINK LINKNAME=\$UDSKLIB ,FILE-NAME= <i>KDBS-modulbibliothek</i>]		X
12.	/START-UDS-DBH [RESIDENT-PAGES=*PAR(MIN=... ,MAX=...)]	X	
<i>Privatinstallation (alternatives Kommando)</i>			
	/START-EXECUTABLE-PROGRAM FROM-FILE= (LIB= <i>UDS/SQL-modulbibliothek</i> ,ELEM=UDSSQL)	X	
13.	/START-PROGRAM FROM-FILE=*MODULE(LIB= <i>lib</i> ,ELEM= <i>elem</i> ,RUN-MODE=*ADVANCED [(NAME-COLLISION=*STD)],PROG-MOD=*ANY [,RESIDENT-PAGES=*PAR(MIN=... ,MAX=...)]		X

Tabelle 7: DBH-Startkommandos

1. Mit diesem SET-FILE-LINK-Kommando vergeben Sie den Namen der DB-Konfiguration der Session.

konfigurationsname

frei wählbarer Name der Datenbankkonfiguration; kann im Mono-DB-Betrieb mit *dbname* identisch sein.
Es gelten folgende Regeln:

- *konfigurationsname* darf max. 41 Zeichen lang sein;
- alle *konfigurationsnamen* einer BS2000-Session müssen in den ersten sieben Zeichen eindeutig sein;
- *konfigurationsname* darf in den ersten acht Stellen kein Sonderzeichen enthalten.

konfname

Die ersten acht Zeichen von *konfigurationsname*. Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

Aus den ersten acht Zeichen des *konfigurationsnamens* bildet der DBH die Namen der folgenden Dateien:

- Session-Log-File (SLF): *konfname*.SLF
- Temporärer Realm: *konfname*.TEMPO.nnn
- RLOG-Dateien: *konfname*.RLOG.*rlogzeitstempel*{1 | 2}{.SAVE}
- Session-Jobvariable UDSSQL.DBH.*konfname*

Der DBH trägt diese Dateien in der Kennung ein, unter der Sie den DBH starten, falls Sie die Dateien nicht selbst angelegt haben.

Ebenso werden beim independent DBH die ersten acht byte von *konfigurationsname* zur Definition der Kommunikationspfade (Event-Namen des P1-Eventing und Namen des Common Memory Pools) benutzt; Konfigurationsnamen, die kürzer als acht byte sind, werden mit C'0' aufgefüllt. Über sie nehmen die Datenbank-Anwenderprogramme und die DCAM-Administration Verbindung zur DB-Konfiguration auf.

UDS/SQL erweitert die Dateien, falls sie nicht sowieso größer sind:

```
SLF      SPACE = ( 24 , 0 )

DBSTAT  SPACE = ( 24 , 48 )

RLOG    SPACE = ( 192 , 192 )

TEMPO   SPACE = ( 192 , 576 )
```

Hinweis für den independent DBH

Wenn Realms oder Dateien der DB-Konfiguration auf privater Platte liegen, so müssen die entsprechenden Platten mehreren Tasks zugänglich gemacht werden (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“) vor dem DBH-Start mit dem Operator-Kommando:

```
/SET-DISK-PARAMETER, . . .
```

- Standardmäßig legt der DBH die temporären Benutzerrealms auf gemeinschaftlicher Platte an. Wenn die Temporären Realms auf privater Platte liegen sollen, so müssen Sie pro Transaktion (siehe „PP TRANSACTION“ im Kapitel "[Maximale Anzahl Transaktionen festlegen \(TRANSACTION\)](#)") einen Temporären Realm in der Konfigurationskennung einrichten mit dem Kommando:

```
/CREATE-FILE FILE-NAME=konfname.TEMPO.nnn
, SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn, DEVICE-TYPE=gerät[ , SPACE= . . . ])
```

konfname

die ersten acht Zeichen von *konfigurationsname*

nnn

Nummer der Transaktion dreistellig mit führenden Nullen

001..*m*

m.

Wert des DBH-Ladeparameters PP TRANSACTION (siehe „PP TRANSACTION“ im Kapitel "[Maximale Anzahl Transaktionen festlegen \(TRANSACTION\)](#)")

- Standardmäßig legt der DBH die Session-Log-File (SLF) auf gemeinschaftlicher Platte an. Wenn die Session-Log-File auf privater Platte liegen soll, so müssen Sie diese mit folgendem Kommando einrichten:

```
/CREATE-FILE FILE-NAME=konfname.SLF
, SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn, DEVICE-TYPE=gerät[ , SPACE= . . . ])
```

Die SLF wird im 4-Kbyte-Seitenformat eingerichtet. Wenn eine SLF < UDS/SQL V2.0 existiert (d.h. mit 2-Kbyte-Seitenformat), und Sie wieder den gleichen Konfigurationsnamen verwenden wollen, müssen Sie diese SLF löschen.

! VORSICHT!

Die SLF darf nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

4. Die DB-Status-Dateien müssen, falls noch nicht vorhanden, eingerichtet werden. Die DB-Status-Dateien werden in 4-Kbyte-Seitenformat eingerichtet. Sie können ihre Lage auf getrennten Datenträgern bestimmen. Wenn die DB-Status-Dateien auf privater Platte liegen sollen, so müssen Sie diese mit folgendem Kommando einrichten:

```
/CREATE-FILE FILE-NAME=konfname.DBSTAT[.SAVE]
, SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn, DEVICE-TYPE=gerät[ , SPACE=... ])
, SUPPRESS-ERRORS=*FILE-EXISTING
```

! VORSICHT!

Die DB-Status-Dateien dürfen nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

5. Wenn in das Anwenderprogramm das Verbindungsmodul UDSLNKA eingebunden wurde, muss der Auftragsschalter 28 auf OFF gesetzt werden, damit der linked-in DBH gestartet wird (wenn der Auftragsschalter 28 auf ON gesetzt ist, wird die independent-Variante des Anwenderprogramms gestartet; siehe auch Handbuch „[Anwendungen programmieren](#)“, COBOL-Programm starten).
6. Mit diesen Kommandos wird die Eingabequelle festgelegt, aus der der DBH die Ladeparameter einlesen soll. Eingabequellen können sein:
- eine SAM- oder ISAM-Datei
 - ein PLAM-Bibliothekselement
 - eine Prozedurdatei (DBH-Startprozedur)

Im Folgenden sind die Kommandos aufgeführt, die Sie abhängig von der zuzuweisenden Eingabequelle jeweils benötigen:

- a. Zuweisen einer SAM- oder ISAM-Datei (independent und linked-in DBH)

```
/ADD-FILE-LINK LINK-NAME=PPFILE, FILE-NAME=dateiname
```

- b. Zuweisen eines PLAM-Bibliothekselements (independent und linked-in DBH)

```
/ASSIGN-SYSDTA TO=*LIBRARY-ELEMENT(LIB=lib, ELEM=elem, TYPE=S)
/ADD-FILE-LINK LINK-NAME=PPFILE, FILE-NAME=SYSDTA
```

Siehe „[Hinweis für den linked-in DBH](#)“.

- c. Zuweisen einer Prozedurdatei (independent DBH)

Wenn die Ladeparameter in der DBH-Startprozedur stehen, muss vor dem DBH-Startkommando folgendes Kommando angegeben werden:

```
/ADD-FILE-LINK LINK-NAME=PPFILE, FILE-NAME=SYSCMD
```

- d. Zuweisen einer Prozedurdatei (linked-in DBH)

Wenn die Ladeparameter in der Startprozedur für das linked-in-Anwenderprogramm stehen, muss vor dem Start des Anwenderprogramms folgendes Kommando angegeben werden:

```
/ASSIGN-SYSDTA TO=*SYSCMD
```

Siehe „[Hinweis für den linked-in DBH](#)“.

Hinweis für den linked-in DBH

Es wird empfohlen, die Ladeparameter aus einer Datei (PPFILE) einzugeben, wenn SYSDTA für Anwenderdaten benutzt wird. Wenn Sie die Ladeparameter auf SYSDTA stehen haben und das Anwenderprogramm ebenfalls von SYSDTA liest, so gilt Folgendes:

- Bei Verwendung der COBOL-DML richtet sich der Zeitpunkt der Eingabe der Ladeparameter danach, wann das erste COBOL-Modul mit Subschema-Deklarationen durchlaufen wurde:
 - Die Ladeparameter müssen **vor** den Daten für das Programm stehen, wenn die Daten für das Programm nach dem ersten Ansprung eines COBOL-Moduls mit Subschema-Deklarationen eingelesen werden.
 - Die Ladeparameter müssen **nach** den Daten für das Programm stehen, wenn die Daten für das Programm vor dem ersten Ansprung eines COBOL-Moduls mit Subschema-Deklarationen eingelesen werden.
 - Bei Verwendung von ausschließlich CALL-DML richtet sich der Zeitpunkt der Eingabe von Ladeparametern und Daten danach, ob das Anwenderprogramm vor oder nach dem ersten READYC-Aufruf von SYSDTA liest.
7. Mit diesem Kommando können Sie den variablen Teil des Namens der Session-Jobvariable festlegen. Siehe [Abschnitt „Session-Jobvariable“](#).
 8. Mit diesem Kommando weisen Sie eine Pubset-Deklarations-Jobvariable zu, in der eine UDS/SQL-Pubset-Deklaration festgelegt ist. Mit dieser können Sie die Eindeutigkeitsanforderung für Dateinamen auf eine ausgewählte Menge von Katalogkennungen beschränken. Es wird empfohlen, generell eine Pubset-Deklarations-Jobvariable in der Startprozedur zuzuweisen, da dies später aus einer anderen Task nicht mehr möglich ist. Wenn Sie keine Beschränkung der zu berücksichtigenden Pubsets wünschen, können Sie die Default-Einstellung "*" in die Pubset-Deklarations-Jobvariable eintragen. Siehe [Abschnitt „Pubset-Deklarations-Jobvariable“](#).

i Eine beim DBH-Start vorliegende fehlerhafte Jobvariablen-Zuweisung oder die Zuweisung einer fehlerhaften UDS/SQL-Pubset-Deklaration führt zum Abbruch der Session.

9. Diese Kommandos sind nur beim Start des linked-in DBH relevant.
 Im Falle von CALL-DML-Anwenderprogrammen muss die Bibliothek bekannt sein, aus der zum Ablaufzeitpunkt die SSITAB-Module nachgeladen werden sollen. Als Erstes wird die Bibliothek durchsucht, die mit dem Linknamen \$UDSSSI zugewiesen wurde. Wenn die SSITAB-Module nicht nur in einer Bibliothek gehalten werden, z.B. pro Datenbank in einer eigenen, können die weiteren Bibliotheken mit den Linknamen BLS-LIB00 bis BLSLIB99 zugewiesen werden. Die Zuweisung mit dem Linknamen \$UDSS-SI und ggf. zusätzlich mit BLSLIB*nn* ist das empfohlene Standardverfahren. Für dieses Verfahren wird in den Fehlermeldungen des Produkts die Abkürzung „\$UL“ verwendet.
 Wenn der Linkname \$UDSSSI nicht verwendet wird oder der Nachladevorgang nicht erfolgreich war, wird aus Kompatibilitätsgründen eine in der Ablaufkennung vorhandene Bibliothek UDS.MODLIB bzw. eine mit dem SET-TASKLIB-Kommando zugewiesene Bibliothek durchsucht. Für dieses Verfahren wird in den Fehlermeldungen des Produkts die Abkürzung „TSK“ verwendet.
 Bei KDBS-Anwendungen muss zusätzlich mit dem Linknamen \$UDSPLEX die Bibliothek zugewiesen werden, die die PLITAB-Module enthält. Das Nachladeverfahren ist analog zur Behandlung der SSITAB-Module.
10. Wenn eine Umsetztabelle (UDSTRTAB) für eine benutzerspezifische Sortierung von Character-Feldern genutzt wird, muss die Bibliothek bekannt sein, aus der das UD-STRTAB-Modul nachgeladen werden soll. Als Erstes wird eine Bibliothek in der Konfigurationskennung durchsucht, die mit dem Linknamen \$UDSKONF zugewiesen wurde. Für weitere Einzelheiten siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt „Umsetztabelle für eine anwendungsspezifische Sortierung“.
11. Es wird empfohlen, generell mit dem Kommando SELECT-PRODUCT-VERSION anzugeben, welche UDS/SQL-Version genutzt werden soll, da mit IMON im Software Configuration Inventory (SCI) mehrere UDS/SQL-Versionen parallel installiert und mehrere Versionen des UDS/SQL-Subsystems vorgeladen sein können. Dabei sollte die Versionsnummer immer inklusive Freigabe- und Korrekturstand angegeben werden (z.B. 02.9B00).

Weiterführende Informationen zum Gebrauch des SELECT-PRODUCT-VERSION-Kommandos finden Sie im [Abschnitt „Mehrere UDS/SQL-Versionen parallel nutzen“](#). Der Einsatz des dort näher beschriebenen Verfahrens ist auch dann empfehlenswert, wenn über längere Zeiträume nur eine UDS/SQL-Version auf der Anlage genutzt wird.

Beim Aufruf des independent DBH wird mit SELECT-PRODUCT-VERSION die Version des Startmoduls UDSSQL(Mastertask) ausgewählt, beim Aufruf des linked-in-Anwenderprogramms die Version des nachzuladenden, versionsabhängigen Verbindungsmoduls LCCONCT. Das eigentliche UDS/SQL-Coding wird in der passenden Version zum Startmodul bzw. linked-in-Verbindungsmodul entweder als vorgeladenes Subsystem genutzt oder aus der entsprechenden SYSLNK-Bibliothek des Produkts UDS/SQL nachgeladen. Für dieses Verfahren wird in den Fehlermeldungen des Produkts die Abkürzung „SCI“ verwendet.

Für weitere Einzelheiten siehe neben dem [Abschnitt „Mehrere UDS/SQL-Versionen parallel nutzen“](#) auch bezüglich der linked-in-Anwenderprogramme den Abschnitt „UDS/SQL-TIAM-Anwenderprogramm binden, laden und starten“ im Handbuch [„Anwendungen programmieren“](#).

UDS-D-Einsatz:

Bei Einsatz von UDS-D (nur mit dem independent DBH möglich) sollte mit einem zusätzlichen SELECT-PRODUCT-VERSION-Kommando die zur UDS/SQL-Version korrespondierende UDS-D-Version angegeben werden, um sicher zu stellen, daß immer die korrekten Produktteile von UDS-D nachgeladen werden. Auch hier wird empfohlen, SELECT-PRODUCT-VERSION auch dann zu verwenden, wenn über längere Zeiträume nur eine UDS-D-Version auf der Anlage genutzt wird.

Privatinstallation

Wenn eine Produktversion eingesetzt werden soll, die nicht rechnerweit im SCI verwaltet wird, gelten folgende Besonderheiten:

Die Bibliotheken, aus denen die Module des Produkts UDS/SQL und ggf. der Produkte UDS-D und UDSKDBS in der gewünschten Version nachgeladen werden sollen, werden explizit mit dem ADD-FILE-LINK-Kommando zugewiesen. Folgende Linknamen stehen für die Zuweisungen zur Verfügung:

\$UDSLIB Module des Produkts UDS/SQL (independent und linked-in DBH)

\$UDSDLIB Module des Produkts UDS-D (nur bei independent DBH)

\$UDSKLIB Module des Produkts KDBS (nur bei linked-in DBH)

Für dieses Verfahren wird in den Fehlermeldungen des Produkts die Abkürzung „SUL“ verwendet.

Aus Kompatibilitätsgründen wird auch weiterhin die Nutzung einer UDS.MODLIB mit ggf. TASKLIB-Zuweisung zum Nachladen der Module unterstützt, wenn die Produkte weder über SCI noch über die Linknamen \$UDSLIB, \$UDSDLIB und \$UDSKLIB bereitgestellt sind. Für dieses Verfahren wird in den Fehlermeldungen des Produkts die Abkürzung „TSK“ verwendet.

Die SELECT-PRODUCT-VERSION Kommandos für die Produkte UDS/SQL und ggf. UDS-D werden aus folgenden Gründen evtl. auch bei einer Privatinstallation benötigt:

Parallel zu einer privat installierten Produktversion können auch UDS/SQL- und UDS-D-Subsystemen vorgeladen sein. Mit dem SELECT-PRODUCT-VERSION-Kommando wird sichergestellt, dass die aus der Privatinstallation nachgeladenen Module eine Verbindung zur richtigen Subsystemversion herstellen.

12. Der independent DBH wird mit dem Kommando START-UDS-DBH gestartet.

Es wird empfohlen, im START-Kommando den Operanden RESIDENT-PAGES anzugeben, da nur in diesem Fall UDS/SQL die performantere Zugriffsmethode FASTPAM (statt UPAM) für Dateizugriffe verwendet (für

weitere Einzelheiten siehe [Abschnitt „FASTPAM für UDS/SQL nutzen“](#)).

Alternativ zum START-UDS-DBH-Kommando kann aus Kompatibilitätsgründen auch das START-EXECUTABLE-PROGRAM-Kommando (bzw. das START-PROGRAM-Kommando) zum Aufruf des independent DBH verwendet werden (Beispiel siehe unter [„Privatinstallation“](#)).

Die Mastertask liest die Ladeparameter des DBH ein. Der DBH initialisiert die Common Memory Pools und startet dann eine oder mehrere Servertasks als ENTER-Jobs. Die entsprechenden ENTER-Dateien trägt die Mastertask mit folgenden Dateinamen in der Kennung ein, unter der Sie den DBH starten (Konfigurationskennung):

UDS.ENTER.*tsn*.ST0*nn*

ENTER-Dateien für die Servertask(s)

tsn

Prozessfolgennummer der Mastertask

nn

Nummer der Servertask

Wenn beim Einsatz von UDS-D PP DISTRIBUTION=START oder STANDBY angegeben wurde, wird eine weitere ENTER-Datei angelegt und bei PP DISTRIBUTION= START gestartet:

UDS.ENTER.*tsn*.CT000

Die Mastertask startet die ENTER-Jobs für die Servertasks (und bei UDS-D-Betrieb für die UDS-D-Task) mit folgenden Angaben:

- JOB-CLASS = Jobklasse der Mastertask, falls die Mastertask als ENTER-Job läuft. Wenn die Mastertask als Dialogauftrag läuft, startet sie die ENTER-Jobs in der Standard-Jobklasse der Kennung(JOB-CLASS=*STD).
- RUN-PRIO = Priorität der Mastertask
- START=IMMEDIATELY
- Die START-Angabe wird ignoriert, falls sie weder in dem ADD-USER-Eintrag noch in der Jobklasse zugelassen ist.
- TIME=NTL oder TIME=32000, falls NTL weder in der aktuellen Benutzerkennung noch in der Jobklasse zugelassen ist.

Wenn der Start der ENTER-Jobs nicht in der Jobklasse der Mastertask gelingt, so versucht die Mastertask, die ENTER-Jobs in der Standard-Jobklasse der Kennung aber mit der gleichen RUN-PRIO- und TIME-Angabe zu starten.

Wenn auch dies erfolglos ist, wird in der Standard-Jobklasse ohne RUN-PRIO-Angabe gestartet.

Jobklassen und Benutzerkennungen richtet der BS2000-Systemverwalter für Sie ein. Die Information über die für Ihre Benutzerkennung oder Jobklassen eingetragenen Werte erhalten Sie mit den BS2000-Kommandos SHOW-USER-ATTRIBUTES bzw. SHOW-JOB-CLASS (siehe die Kommando-Handbücher zu [„BS2000 OSD /BC“](#)).

Nach dem normalen DBH-Ende löscht die Mastertask diese ENTER-Dateien automatisch.

Privatinstallation

Bei Nutzung einer Produktversion, die nicht rechnerweit im SCI verwaltet wird, kann in der Regel nicht das oben beschriebene START-Kommando verwendet werden, sondern das START-EXECUTABLE-PROGRAM-Kommando.

Beispiel

```
/START-EXECUTABLE-PROGRAM FROM-FILE=(
    LIB=$kennung.SYSLNK.UDS-SQL.version,ELEM=UDSSQL)
```

13. Den linked-in DBH laden und starten Sie durch den Aufruf des Anwenderprogramms mit dem START-PROGRAM-Kommando (bzw. mit dem START-EXECUTABLE-PROGRAM-Kommando).

Der voreingestellte Wert NAME-COLLISION=*STD sollte nicht verändert werden, da es sonst bei Anwendungen, die mit dem BINDER gebunden wurden, zu Namenskonflikten bei nachzuladenden Entries kommen kann.

Verwendet das Anwenderprogramm die **COBOL-DML**, so ruft nach dem Start des Programms das Verbindungsmodul die Module des linked-in DBH auf. Der linked-in DBH liest daraufhin die Ladeparameter des DBH ein und gibt anschließend die Steuerung an das Anwenderprogramm zurück.

Beim Verwenden von **CALL-DML** veranlasst der erste READYC-Aufruf das Verbindungsmodul dazu, die Module des linked-in DBH und des CALL-DML-Umsetzers nachzuladen. Der linked-in DBH liest daraufhin die Ladeparameter des DBH ein und bearbeitet anschließend den READYC-Aufruf.

2.3.1.4 Meldungen beim DBH-Start

Nach dem Absetzen des ENTER-JOB-Kommandos zum Starten der Servertask(s) wartet die Mastertask in einer Zeitschleife auf Startinformationen der Servertask(s).

Solange sich nicht alle Tasks bei der Mastertask gemeldet haben, erscheint an der Datensichtstation des Datenbankadministrators und an der Operatorkonsole im Abstand von ca. drei Minuten die Meldung:

```
% UDS0212 UDS IST NOCH NICHT BETRIEBSBEREIT. BITTE PRUEFEN SIE, OB DIE UDS-
STAPELPROZESSE EXISTIEREN UND ANLAUFEN KOENNEN
```

Prüfen Sie in diesem Fall, welche Tasks des DBH bisher nicht gestartet wurden und warum (Batch-Limit, ADD-USER-Eintrag), oder ob sich Tasks des DBH abnormal beendet haben.

Bei erfolgreichem DBH-Start erscheint die Meldung:

```
% UDS0201 UDS IST BETRIEBSBEREIT
```

Gleichzeitig wird die Session-Jobvariable mit UDSDBH_ACTIVE versorgt (siehe [Abschnitt „Jobvariablen für UDS/SQL nutzen“](#)).

UDS/SQL-Pubset-Deklaration

Eine beim DBH-Start berücksichtigte UDS/SQL-Pubset-Deklaration (siehe [Abschnitt „Pubset-Deklarations-Jobvariable“](#)) wird folgendermaßen protokolliert:

```
UDS0746 UDS-PUBSET-DEKLARATION (CURRENT) FOLGT (... ,tsn4)
tsn4: UDS-PUBSET-JV: name der jv
tsn4: PUBSETS: catidgroup_1
tsn4: PUBSETS: catidgroup_2
tsn4: PUBSETS: ...
```

Je einmal für jede in der UDS/SQL-Pubset-Deklaration angegebene Catid-Gruppe.

```
tsn4: DEFAULT PUBSET: <Standard-Katalogkennung der Ablaufkennung>
tsn4: -----
```

Die Zuweisung einer nicht vorhandenen oder nichtzugreifbaren Jobvariable oder einer Jobvariable der Länge 0 wird folgendermaßen abgewiesen:

```
UDS0752 UDS-ANWENDERFEHLER: ZUGRIFF ZUR UDS-PUBSET-JV UEBER JV-LINKNAMEN
*UDSPS01 SCHEITERTE (... ,tsn4)
```

Zusätzliche Hinweise zur Fehlerursache geben ggf. weitere Meldungen.

Inhaltliche Fehler der UDS/SQL-Pubset-Deklaration werden abhängig von der Art des Fehlers und dem Zeitpunkt der Fehlererkennung wie folgt gemeldet :

- Syntaxfehler, die bei der syntaktischen Analyse (Erkennung der Catid-Gruppen) erkannt wurden:

```
UDS0748 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX):
<ursache> (... ,tsn4)
```

Dabei kann <ursache> folgende Werte annehmen:

- CATID GROUP TOO LONG
- TOO MANY CATID GROUPS
- EXCLUDE CATID GROUP NOT ALONE

```
tsn4: UDS-PUBSET-JV: Name der Jobvariable
tsn4: UDS-PUBSET-JV-CONTENTS:
tsn4: Inhalt der Jobvariable, ggf. über mehrere Zeilen verteilt
tsn4: -----
```

- Während der Prüfung der Catid-Gruppen mit dem SDF-Makro CMDWCC erkannte Fehler:

```
UDS0749 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX): CATID
GROUP REJECTED BY CMDWCC (... ,tsn4)

tsn4: UDS-PUBSET-JV: name der Jobvariable
tsn4: PUBSETS: Catid-Gruppe
```

Für jede Catid-Gruppe, die vom Makro CMDWCC beanstandet wurde und zum Scheitern führte, max. 10 Mal. Gibt es mehr als 10 fehlerhafte Catid-Gruppen, werden nur die ersten 10 protokolliert und mit folgender Zeile angezeigt, dass weitere fehlerhafte Catid-Gruppen vorhanden sind:

```
tsn4: FURTHER ERRORS NOT SHOWN
tsn4: -----
```

- Während der Prüfung der Catid-Gruppen mit FSTAT erkannte Fehler:

```
UDS0749 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX): CATID
GROUP REJECTED BY FSTAT (... ,tsn4)

tsn4: UDS-PUBSET-JV: name der jv
tsn4: PUBSETS: Catid-Gruppe, FSTAT-DMS-RC: xxxx
```

Für jede Catid-Gruppe, die vom FSTAT beanstandet wurde und zum Scheitern führte, max. 10 Mal. Gibt es mehr als 10 fehlerhafte Catid-Gruppen, werden nur die ersten 10 protokolliert und mit folgender Zeile angezeigt, dass weitere fehlerhafte Catid-Gruppen vorhanden sind:

```
tsn4: FURTHER ERRORS NOT SHOWN
tsn4: -----
```

2.3.1.5 Beispiele

Beispiel 1: Independent DBH als Stapelauftrag starten

Inhalt der DBH-Startprozedur ENTER.DBH.IND:

```
/SET-LOGON-PARAMETERS
/ASS-SYSOUT TO=O.DBH.INT
/SHOW-JV JV.MANUAL.VERSION
/SET-JV-LINK LINK-NAME=VERS ,JV-NAME=JV.MANUAL.VERSION
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=&( *VERS )
/SET-JV-LINK LINK-NAME=UDSPS01 ,JV-NAME=PUBSDECL.PUBS
/SHOW-JV JV-CONTENTS=*LINK(LINK-NAME=UDSPS01)
/CREATE-FILE FILE-NAME=CONFBSPL.DBSTAT
/CREATE-FILE FILE-NAME=CONFBSPL.DBSTAT.SAVE
/SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME=CONFBSPL
/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME=UDSDBB.PP.FILE.IND
/START-UDS-DBH RESIDENT-PAGES=*PAR(MIN=0 ,MAX=1525)
/EXIT-JOB SYSTEM-OUTPUT=*NONE
```

Starten der Enter-Prozedur:

```
/ENTER-JOB FROM-FILE=ENTER.DBH.IND ,JOB-CLASS=JCB32000
% JMS0066 AUFTRAG 'P' ANGENOMMEN AM 12-01-30 UM 11:45, TSN = 1DU0
```

Protokoll des Enter-Jobs (Auszug)

```
/SHOW-JV JV.MANUAL.VERSION
% 02.9B00
...
/SHOW-JV JV-CONTENTS=*LINK(LINK-NAME=UDSPS01)
% SQL2
...
/START-UDS-DBH RESIDENT-PAGES=*PAR(MIN=0 ,MAX=1525)
...
```

Beispiel 2: Linked-in Anwendung (COBOL-DML) als Dialogauftrag starten

Inhalt der Ladeparameter-Datei PP.FILE:

```
PP LOG=PUBLIC
PP PARLIST=YES
PP PRIVACY-CHECK=OFF
PP STDCKPT=YES
PP END
```

Inhalt der Startprozedur P.VERSAND:

```

/BEGIN-PROCEDURE
/SHOW-JV JV.VERSAND
/SET-JV-LINK LINK-NAME=VERS ,JV-NAME=JV.VERSAND
/SET-JV-LINK LINK-NAME=UDSPS01 ,JV-NAME=PUBSDECL.PUB
/SHOW-JV JV-CONTENTS=*LINK(LINK-NAME=UDSPS01)
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=&( *VERS)
/ADD-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME=VERSAND.DBDIR
/ADD-FILE-LINK LINK-NAME=EINART ,FILE-NAME=DATEN.ARTLADEN.ARTIKEL
/ADD-FILE-LINK LINK-NAME=HERST ,FILE-NAME=DATEN.ARTLADEN.HERSTELLER
/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME=UDSDBB.PP.FILE
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIBRARY=PROC.MANUAL ,ELEMENT=ARTLAD)
/END-PROCEDURE

```

Aufruf der Startprozedur:

```

/CALL-PROC FROM=P.VERSAND

```

Ablaufprotokoll (Auszug):

```

...
/SHOW-JV JV.VERSAND
% 02.9B00
...
/SHOW-JV JV-CONTENTS=*LINK(LINK-NAME=UDSPS01)
% *
...
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIBRARY=PROC.MANUAL ,ELEMENT=ARTLAD)
...
% UDS0215 UDS STARTET UDS/SQL V2.9 (LINKED-IN), DATE=2017-06-28 (ILL2038,11:26:29/0YA2)
% UDS0746 UDS-PUBSET-DEKLARATION (CURRENT) FOLGT (ILL1746,11:26:29/0YA2)
0YA2: UDS-PUBSET-JV: :SQL2:$XXXXXXXXX.PUBSDECL.ALL
0YA2: PUBSETS: *
0YA2: DEFAULT PUBSET: SQL2
0YA2: -----
% UDS0722 UDS AUFTRAG ADD RLOG 150628092628 WIRD AUSGEFUEHRT (ILL1283,11:26:29/0YA2)
% UDS0354 UDS ALOG-KONSISTENZPUNKT FUER VERSAND (ILL1307,11:26:29/0YA2)
0YA2: ALOG-CKPT 20170628092635: CKPT ALREADY EXISTING.
0YA2: MAXDB = 1
0YA2: TRANSACTION = 1
0YA2: SUBSCHEMA = 1
0YA2: 2KB-BUFFER-SIZE= 1
0YA2: 4KB-BUFFER-SIZE= 1
0YA2: 8KB-BUFFER-SIZE= 0
0YA2: LOG = PUBLIC
0YA2: LOG-2 = NO
0YA2: LOG-SIZE = ( 192, 192)
0YA2: RESERVE = NONE
0YA2: WARMSTART = STD
0YA2: CONSOLE = NO
0YA2: STDCKPT = YES
0YA2: LOCK = STD
0YA2: PRIVACY-CHECK = OFF
0YA2: CONFNAME = $XXXXXXXXX.VERSAND

```

```

0YA2: DATABASES OF CONFIGURATION:
0YA2:  $XXXXXXXXX.VERSAND                ,EXCLUSIVE-UPD                ,*SYSTEM
%   UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER VERSAND  TERMINATED ( ILL1309,11:26:29/0YA2)
DB-ART-NR 000001
DB-BEZ SOMMERKLEID MIT JACKE
DB-ART-NR 000002
DB-BEZ ZWEITEILIGES SOMMERKLEID
DB-ART-NR 000003
DB-BEZ WEISSBIER
DB-ART-NR 000004
DB-BEZ HELLES BIER
DB-ART-NR 000005
DB-BEZ ZITRONENLIMONADE
DB-ART-NR 830950
DB-BEZ PLISSEE-KLEID MIT JACKE
DB-ART-NR 830993
DB-BEZ KLEID AUS JERSEY-KREPP
DB-ART-NR 835758
DB-BEZ KLEID AUS FLIESENDEM JERSEY
DB-ART-NR 831213
DB-BEZ KLEID AUS JERSEY-KREPP MIT JACKE
DB-ART-NR 835928
DB-BEZ T-SHIRT KLEID
DB-ART-NR 835952
DB-BEZ POLO-KLEID
DB-ART-NR 001140
DB-BEZ FETTARMER NATURJOGHURT
DB-ART-NR 001141
DB-BEZ BIO-NATURJOGHURT
*** NORMALES PROGRAMM-ENDE ***
%   UDS0354 UDS ALOG-KONSISTENZPUNKT FUER VERSAND ( ILL1307,11:26:29/0YA2)
0YA2: ALOG-CKPT 20170628092636: FIXED ( ALOG-NR:000000001, START-CKPT: 20170628092635 ).
%   UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK ( ILL1758,11:26:29/0YA2)
0YA2: DATABASE NAME                DMLS    LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
0YA2: -----
0YA2: VERSAND                        115      1489      101        694        103
%   UDS0213 UDS NORMAL BEENDET MIT *****115 DML-STATEMENTS 2017-06-28 ( ILLY033,11:26:
29/0YA2)

```

2.3.2 DB-Konfiguration festlegen und ändern

Eine DB-Konfiguration legen Sie fest, indem Sie dem DBH die anzuschließenden Datenbanken bekannt geben. Abhängig von der DBH-Variante haben Sie folgende Möglichkeiten:

linked-in DBH

- über DBH-Ladeparameter

independent DBH

- über DBH-Ladeparameter und
- über DAL-Kommandos

Der Konfigurationsname kann im Mono-DB-Betrieb mit dem Namen der Datenbank identisch sein (vgl. Ladeparameter DBNAME, "[Datenbanken benennen \(DBNAME\)](#)").

Den **independent DBH** können Sie auch mit „leerer DB-Konfiguration“ starten, indem Sie beim DBH-Start in den Ladeparametern keine Datenbanken angeben. Die Datenbanken können Sie dann nachträglich über DAL-Kommandos anschließen.

Eine DB-Konfiguration können Sie während der Session oder nach einem Session-Abbruch ändern.

Mit DAL-Kommandos können Sie **Datenbanken anschließen (ADD)** und **Datenbanken ausschließen (DROP)**, siehe [Abschnitt „Die Datenbankadministratorsprache DAL“](#).

DB-Konfiguration während der Session ändern

Zum Anschließen und Ausschließen von Datenbanken einer DB-Konfiguration geben Sie mit DAL-Kommando ADD bzw. DROP einen Auftrag an den DBH, eine oder mehrere Datenbanken anzuschließen bzw. auszuschließen. Dieser Auftrag wird erst durchgeführt, wenn Sie das DAL-Kommando PERFORM eingeben.

Wenn die anzuschließende Datenbank inkonsistent ist, führt der DBH einen Warmstart durch, bevor er sie für Transaktionen zugänglich macht.

Dieser Warmstart umfasst:

- den Rollback aller auf der Datenbank offenen Transaktionen
- das Nachfahren der Transaktionen, deren Änderungen noch nicht auf der Datenbank stehen

In der SLF wird die jeweils aktuelle DB-Konfiguration ausfallsicher protokolliert. Die ausfallsichere Protokollierung ermöglicht nach einem Session-Abbruch einen automatischen Session-Wiederanlauf. Das heißt, dass der DBH die zu einem Session-Wiederanlauf nötigen Informationen aus der SLF entnimmt. Diese Informationen sind unter anderem die aktuell gültigen DBH-Ladeparameter und die aktuell angeschlossenen Datenbanken.

Hinweis für UDS-D

Wenn die DB-Konfiguration während der Session verändert wird, muss der Datenbankadministrator die Verteiltabelle über die Administrationsschnittstelle anpassen (DAL-Kommandos). Gibt es Transaktionen, die auch nach Datenbank-Warmstart noch im Zustand PTC (Prepared to Commit) sind, so kann die entsprechende Datenbank nicht von der DB-Konfiguration ausgeschlossen werden (siehe [Abschnitt „DB-Konfiguration ändern - Hinweise für UDS-D“](#)).

DB-Konfiguration nach Session-Abbruch ändern (Rekonfiguration)

Nach Session-Abbruch kann der Datenbankadministrator auf einen Session-Wiederanlauf verzichten und mit einer neu zusammengestellten DB-Konfiguration und neuen DBH-Ladeparametern eine neue Session mit gleichem (oder anderem) Konfigurationsnamen und beliebiger DBH-Variante starten.

Eine Rekonfiguration ist z.B. nützlich, wenn ein Session-Abbruch kurz vor dem geplanten DBH-Ende erfolgt und eine Session-Fortsetzung über Session-Wiederanlauf sich nicht mehr lohnt.

Um eine Rekonfiguration durchzuführen, löscht der Datenbankadministrator die SLF-Datei dieser Session (siehe [Abschnitt „Session-Log-File \(SLF\)“](#)). Dadurch ist die abgebrochene Session für beendet erklärt.

Bei einer Rekonfiguration sind bestimmte Einschränkungen zu beachten (siehe „Hinweise bei Rekonfiguration“ im [Abschnitt „Datenbanken benennen \(DBNAME\)“](#)).

i Der Datenbankadministrator darf nach Session-Abbruch auf einen früheren Stand der Datenbanken zurückgreifen. Er ist allerdings selbst verantwortlich für die Konsistenzbeziehungen zwischen den Datenbanken.

2.3.3 DBH beenden

Der independent DBH lässt sich beenden durch die DAL-Kommandos

- **CLOSE RUN-UNITS:**
Normales DBH-Ende
- **CLOSE CALLS:**
Beschleunigtes normales DBH-Ende
- **%TERM:**
Sofortiges Abbrechen des DBH

DBH normal beenden

Beispiel

UDSADM-Anweisung:

```
01 //EXECUTE-DAL-CMD CMD=CLOSE RUN-UNITS
```

Meldungsausgaben:

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: CLOSE RUN-UNITS (OPOX073,09:33:46
/4TS7)
02 % UDS0241 UDS BEENDIGUNG WURDE EINGELEITET FUER TRANSACTION (OPCF024,09:33:46/4TS7)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,09:33:46
/4TS7)
03 % UDS0241 UDS BEENDIGUNG WURDE EINGELEITET FUER SERVERTASK (OPO0024,09:33:52/4TS7)
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK (OPCC758,09:33:46
/4TS7)
4TS7: DATABASE NAME          DMLS   LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
4TS7: -----
4TS7: KUNDEN                  7       66        37         21         21
04 % UDS0213 UDS NORMAL BEENDET MIT *****7 DML-STATEMENTS 2017-06-18 09:33:53
(OPOB033,09:33:53/4TS7)
```

01 Mit dem DAL-Kommando

```
CLOSE RUN-UNITS
```

starten Sie die normale Beendigungsprozedur.

02 Transaktionen werden nicht mehr zugelassen.

03 Wenn alle Transaktionen beendet sind, leitet die Mastertask die Endebehandlung für alle Servertasks ein.

04 Wenn alle Servertasks beendet sind, entwertet die Mastertask die Session-Log-File (SLF), löscht die Temporären Realms und diejenigen RLOG-Dateien, die zu diesem Konfigurationsnamen gehören und logisch leer sind.

Hinweis für UDS-D/openUTM

CLOSE RUN-UNITS wird abgewiesen, wenn sich noch sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC befinden.

Der Datenbankadministrator kann den DBH trotzdem normal beenden, wenn er vorher den Zustand PTC beendet (siehe [Abschnitt „Zustand PTC beenden“](#)).

DBH beschleunigt normal beenden

Beispiel

UDSADM-Anweisung:

```
01 //EXECUTE-DAL-CMD CMD=CLOSE CALLS
```

Meldungsausgaben:

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: CLOSE CALLS (OPOX073,09:29:25
/4TS3)
% UDS0241 UDS BEENDIGUNG WURDE EINGELEITET FUER TRANSACTION (OPCF024,09:29:27/4TS3)
02 % UDS0348 UDS BEGINNT, SAEMTLICHE TRANSAKTIONEN ZURUECKZUSETZEN (OPC1035,09:29:27/4TS3)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,09:29:27
/4TS3)
03 % UDS0241 UDS BEENDIGUNG WURDE EINGELEITET FUER SERVERTASK (OPO0024,09:29:33/4TS3)
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK (OPCC758,09:29:33
/4TS3)
4TS3: DATABASE NAME          DMLS   LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
4TS3: -----
4TS3: VERSAND                1       86        58         22         22
4TS3: KUNDEN                 0       33        22         6          6
04 % UDS0213 UDS NORMAL BEENDET MIT *****1 DML-STATEMENTS 2017-06-18 09:29:34
(OPOB033,09:29:34/4TS3)

```

01 Mit dem DAL-Kommando

```
CLOSE CALLS
```

beschleunigen Sie das DBH-Ende.

02 Für alle noch nicht abgeschlossenen Transaktionen wird ein FINISH WITH CANCEL erzwungen; d.h. alle Änderungen dieser Transaktionen werden mit Hilfe der zugehörigen RLOG-Datei rückgängig gemacht.

An die Anwenderprogramme, deren Transaktion mit FINISH WITH CANCEL abgebrochen wurde, übergibt der DBH den Statuscode 122 bzw. 151.

03 Anschließend versucht die Mastertask, den DBH normal zu beenden.

04 Die Session ist erst dann beendet, wenn die Mastertask die SLF entwertet hat.

Hinweis für UDS-D/openUTM

CLOSE-CALLS wird abgewiesen, wenn sich noch sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC befinden.

Der Datenbankadministrator kann den DBH trotzdem normal beenden, wenn er vorher den Zustand PTC beendet (siehe [Abschnitt „Zustand PTC beenden“](#)).

DBH sofort abbrechen

Beispiel

UDSADM-Anweisung:

```
01 //EXECUTE-DAL-CMD %TERM
```

Meldungsausgaben:

```

% IDA0N51 PROGRAMM-UNTERBRECHUNG BEI ADRESSE '0003AD70 (SCCDUMP), (CDUMP), CODE=UDS'
02 % IDA0N53 EIN DUMP WIRD ERSTELLT. BITTE WARTEN
% IDA0N54 'USERDUMP' AUSGEGEBEN IN DATEI '$XXXXXXXX.DUMP.DBH.7VAI.00007'
% IDA0N55 TITEL: 'TSN-7VAI UID-XXXXXXXX AC#-Z1285 USERDUMP PC-00034D98 EC-50 VERS-
120
DUMP-TIME 15:27:28 15-06-28'
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: %TERM (OPOX073,15:27:28/7VAI)
03 % UDS0202 UDS WIRD ABGEBROCHEN MIT *****23 DML-STATEMENTS 2017-06-28 15:27:31
(OPY8003,15:27:31/7VAI)
% EXC0732 ABNORMALE PROGRAMM-BEENDIGUNG. FEHLERCODE 'NRT0101': /HELP-MSG NRT0101

```

01 Mit dem DAL-Kommando

```
%TERM
```

brechen Sie unter Umgehung der normalen Terminierungsaktivitäten des DBH die Session auf dem kürzesten Wege ab (Nothalt).

02 Der DBH gibt einen Speicherabzug seines Hauptspeicherbereichs in die Datei

DUMP.*jobname.tsn.nnnnn* (bei PP DUMP=ALL) bzw.

SYS.ADUMP.*jobname.tsn.nnnnn* (bei PP DUMP=STD) aus.

03 Die Mastertask bricht die Session ab; die Datenbanken der DB-Konfiguration sind, wenn seit Ihrem letzten Konsistenzpunkt Änderungen durchgeführt wurden, als inkonsistent markiert. Die abgebrochene Session können Sie mit Session-Wiederanlauf fortsetzen.

Die Session ist erst dann beendet, wenn die SLF gelöscht oder entwertet ist.

2.4 Paralleler Datenbankbetrieb

UDS/SQL bietet die Möglichkeit des parallelen Datenbankbetriebs, d.h. in einem Verarbeitungsrechner können gleichzeitig mehrere Database Handler aktiv sein, unabhängig davon, ob es sich um independent oder linked-in DBHs handelt (zum parallelen Einsatz verschiedener DBH-Versionen auf einem Verarbeitungsrechner siehe [Abschnitt „Mehrere UDS/SQL-Versionen parallel nutzen“](#)ff).

Dabei ist es auch möglich, den DBH unter einer Kennung mehrmals für verschiedene Datenbanken bzw. DB-Konfigurationen zu laden und eine Datenbank von mehreren parallel laufenden DBHs lesend, d.h. mit der Angabe SHARED-RETRIEVAL im Ladeparameter PP DBNAME bzw. im DAL-Kommando ADD DB, anzusprechen. Dies gilt auch für Schattendatenbanken.

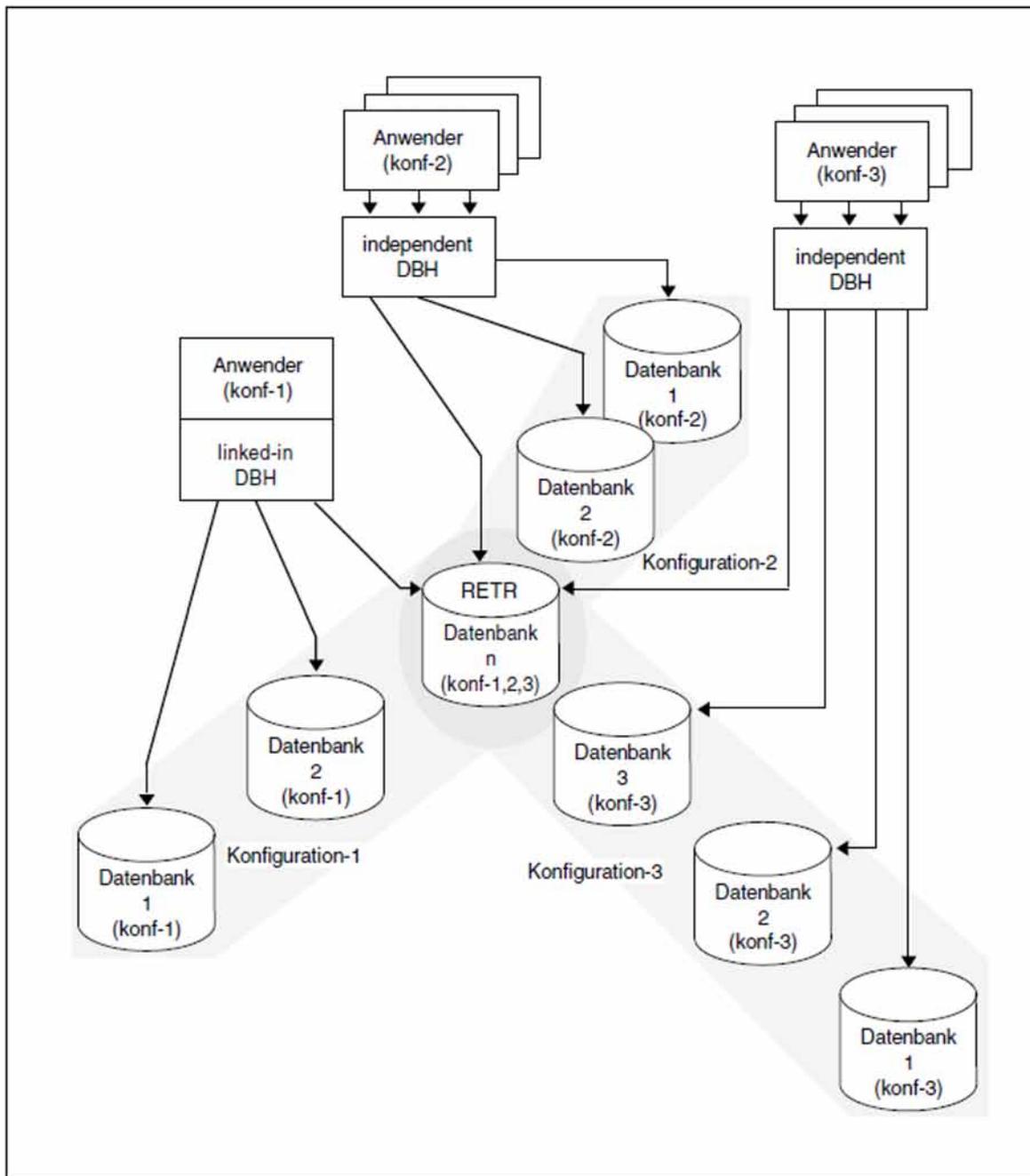


Bild 8: Beispiel eines parallelen Datenbankbetriebs

3 Ladeparameter des DBH

Die Ladeparameter definieren die wesentlichen Merkmale einer Session. Außer PP DBNAME und PP PASSWORD dürfen Sie jeden Ladeparameter nur einmal angeben. Für DBH-Ladeparameter, die Sie nicht angeben, gelten Standardwerte. Nur PP END und PP LOG müssen Sie angeben. Mit PP END beenden Sie die Eingabe und mit PP LOG bestimmen Sie, ob der DBH eine RLOG-Datei führen soll oder nicht.

Die Namen der DBH-Ladeparameter dürfen nicht abgekürzt werden. Eine Ausnahme bilden einige Operanden. Die Zeichenfolge, die Sie angeben müssen, ist unterstrichen.

Die Operanden dürfen nicht durch Leerzeichen unterbrochen werden, da jede Angabe nach dem ersten Leerzeichen als Kommentar interpretiert wird.

Beispiel

```
PP DBNAME=dbname, 'BLANK' n
```

Die Angabe „n“ wird als Kommentar interpretiert.

Eingabesätze, die mit „#“ oder „*“ beginnen, werden als Kommentar interpretiert.

Alle DBH-Ladeparameter dürfen Sie sowohl für den linked-in DBH als auch für den independent DBH verwenden. DBH-Ladeparameter, die in der aktuellen DBH-Variante nicht ausgewertet werden, werden allerdings in Form einer Warnung gemeldet.

Auswertung der DBH-Ladeparameter

Nur beim DBH-Start werden die angegebenen DBH-Ladeparameter ausgewertet. Bei Session-Wiederanlauf dagegen gelten die Werte, die bei Session-Abbruch in der SLF vorlagen: Der Datenbankadministrator hatte während der laufenden Session die Möglichkeit, über die Datenbankadministratorsprache (DAL) die Merkmale der Session zu ändern.

Der DBH weist mit folgender Meldung darauf hin, dass er bei Session-Wiederanlauf die angegebenen DBH-Ladeparameter übergeht:

```
% UDS0709 UDS SESSION-WIEDERANLAUF: DIE LADEPARAMETER WERDEN UEBERLESEN.
```

Sie können jedoch das Standard-Verhalten ändern und nach einem Session-Abbruch die DBH-Ladeparameter selbst wieder angeben. Dazu müssen Sie die SLF-Datei löschen (siehe [Abschnitt „Session-Log-File \(SLF\)“](#)). Wenn die SLF-Datei gelöscht ist, startet der DBH eine neue Session und wertet die DBH-Ladeparameter aus.

Reaktionen in Fehlersituationen

- Wenn Sie DBH-Ladeparameter mehrfach eingeben (außer bei PP DBNAME und PP PASSWORD), so wertet der DBH nur den ersten korrekten Parameter aus, die übrigen ignoriert er.
- Falls der DBH das Schlüsselwort nicht erkennen kann, gibt er eine Fehlermeldung aus und bricht den DBH-Start ab.
- Ladeparameter dürfen maximal 72 Zeichen lang sein. Längere Angaben werden vom DBH abgewiesen und wie Syntaxfehler behandelt.
- Wenn Sie DBH-Ladeparameter direkt im Dialog eingeben, so gibt der DBH bei Syntaxfehlern eine Meldung aus. In diesem Fall können Sie den Fehler korrigieren, indem Sie den richtigen Ladeparameter nochmals eingeben. Geschieht dies nicht, bricht der DBH nach PP END den DBH-Start ab.

Ausnahme

Bei PP PASSWORD und PP DBNAME ist eine direkte Korrektur nicht möglich. Wenn bei PP DBNAME ein Syntaxfehler auftritt, so können Sie zwar weitere Anweisungen PP DBNAME angeben, diese werden aber nur syntaktisch geprüft. Nach PP END bricht der DBH den DBH-Start ab, auch wenn dann korrekte Anweisungen bei PP DBNAME vorhanden sind.

- Wenn Sie DBH-Ladeparameter eingeben, die in einer katalogisierten Datei hinterlegt sind, so ist keine direkte Korrektur möglich. Im Fehlerfall bricht der DBH den DBH-Start ab. Sie müssen den Fehler in der Datei korrigieren.
- Wenn Sie DBH-Ladeparameter eingeben, bei denen die numerischen Werte außerhalb der erlaubten Grenzen liegen, so nimmt der DBH den niedrigsten bzw. höchsten erlaubten Wert und gibt als Warnung eine Meldung für den betroffenen Ladeparameter aus.

Der **independent DBH** verarbeitet folgende Ladeparameter:

Parameter	Standardwert	Bedeutung
<u>PP</u> <u>2KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 2-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
<u>PP</u> <u>4KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 4-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
<u>PP</u> <u>8KB-BUFFER-SIZE</u> = <i>n</i>	0	Größe des 8-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =0 oder <i>n</i> =3..2047
<u>PP</u> <u>ADM</u> ={ <u>REMOTE</u> <u>LOCAL</u> }	REMOTE	REMOTE: Von beliebigen Datenstationen aus über DCAM administrieren LOCAL: Die Mastertask belegt die Datensichtstation permanent. Administration nur über Konsole oder Datensichtstation des Datenbankadministrators möglich
<u>PP</u> <u>ADMPASS</u> = <i>admkenwort</i>	-	Kennwort für die Administration über DCAM festlegen
<u>PP</u> <u>BCAM-PREFIX</u> = <i>prefix</i>	SUD\$	Präfix festlegen für Namen von Anwendertasks, die auf virtuellen Hosts ablaufen
<u>PP</u> <u>CATPASS</u> ={ <u>STANDARD</u> <i>kenwort</i> }	STD	Kennwort für vom DBH einzurichtenden Dateien wie ALOG-Dateien und Temporäre Realms festlegen

<u>PP</u> <u>CHCKTIME</u> = <i>n</i>	60	Zeitintervall in Sekunden für die Verbindungs- und Transaktionsüberwachung der UDS-D-Task angeben <i>n</i> =60..900
<u>PP</u> <u>CP-SIZE</u> = <i>n</i>	1024	Mindestgröße für einzelne Common Pools in Kbyte festlegen; <i>n</i> =1..16384
<u>PP</u> <u>CPU</u> ={ <u>MONO-PROCESSOR</u> <u>MULTI-PROCESSOR</u> }	MONO-PROCESSOR	Verwendeten Prozessortyp angeben
<u>PP</u> <u>CUP-SIZE</u> = <i>n</i>	1024 ¹ 128 ²	Mindestgröße des Communication Pools in Kbyte festlegen; <i>n</i> =1..16384
<u>PP</u> <u>DBNAME</u> =[<i>\$userid.</i>] <i>dbname</i> [<i>.copyname</i>] [, <u>SHARED-RETRIEVAL</u>] [, { <i>n</i> [<i>n</i>], <i>bufferid</i> }]	-	Datenbanken der DB-Konfiguration benennen; max. 222 Datenbanken <i>n</i> : Größe des User Buffer Pools in Mbyte festlegen; <i>n</i> =0..2047 <i>bufferid</i> : Identifikator des Buffer Pools
<u>PP</u> <u>DEACT</u> ={ <u>YES</u> <u>NO</u> }	YES	Deaktivierung von UDS/SQL-Tasks auf Grund der Rechnerauslastung zulassen
<u>PP</u> <u>DEADTIME</u> = <i>n</i>	60	Zeitangabe in Sekunden, um konfigurationsübergreifende Deadlocks bzw. Deadlocks im Zusammenhang mit openUTM aufzulösen <i>n</i> =5..900
<u>PP</u> <u>DIP-SIZE</u> = <i>n</i>	1024 ¹ 64 ²	Mindestgröße des Distribution Pools in Kbyte festlegen; <i>n</i> =1..16384
<u>PP</u> <u>DISDB</u> = <i>n</i>	1	maximale Anzahl entfernter Datenbanken, die pro Transaktion angesprochen werden können; <i>n</i> =1..32
<u>PP</u> <u>DISTABLE</u> =[<i>:catid:</i>][<i>\$userid.</i>] <i>dateiname</i>	-	Eingabedatei zum Anlegen der Verteiltabelle festlegen

<u>PP DISTRIBUTION</u> ={ <u>NO</u> <u>STANDBY</u> <u>START</u> }	NO	Die Teilnahme am UDS-D-Betrieb steuern: NO: Kein UDS-D-Betrieb möglich STANDBY: UDS-D-Betrieb wird vorbereitet und kann später mit &START DISTRIBUTION gestartet werden START: Der UDS-D-Betrieb wird gestartet
<u>PP DUMP</u> ={ <u>STD</u> <u>ALL</u> }	ALL	Umfang eines Dumps beeinflussen
<u>PP END</u>	-	Eingabe der Ladeparameter beenden
<u>PP IO</u> ={ <u>ASYNC</u> <u>SYNC</u> }	ASYNC	I/Os in den Servertasks asynchron oder synchron ausführen
<u>PP LOCK</u> ={ <u>STD</u> <u>SHARED</u> <u>EXCLUSIVE</u> }	STD	Sperrprotokoll festlegen
<u>PP LOG</u> ={ <u>NO</u> :catid: <u>PUBLIC</u> (priv-vsn-1/gerät-1 [,priv-vsn-2/gerät-2 [,priv-vsn-3/gerät-3]]) (vsn-1[,vsn-2[,vsn-3]])}	-	RLOG-Datei führen
<u>PP LOG-2</u> ={:catid: <u>PUBLIC</u> (priv-vsn-1/gerät-1 [,priv-vsn-2/gerät-2 [,priv-vsn-3/gerät-3]]) (vsn-1[,vsn-2[,vsn-3]])}	-	RLOG-Duplikatdatei führen
<u>PP LOG-SIZE</u> =([primär] [, [sekundär]])	192,192	Speicherplatzgröße der RLOG-Dateien in Anzahl PAM-Seiten festlegen
<u>PP MAXDB</u> =n	Summe von PP DBNAME	max. Anzahl der Datenbanken der DB- Konfiguration festlegen; n=1..222
<u>PP MPSEG</u> ={ <u>STD</u> <u>64K</u> }	STD	Segmentgröße festlegen
<u>PP ORDER-DBSTATUS</u> ={ <u>STD</u> <u>SPECIAL</u> }	STD	Systemverhalten steuern, wenn neue UPDATE-Transaktionen oder - Verarbeitungsketten mit der Ausführung von vorgemerkten Aufträgen kollidieren

<u>PP</u> <u>PARLIST</u> ={ <u>YES</u> <u>NO</u> }	NO	Verwendete Parameter auflisten
<u>PP</u> <u>PASSWORD</u> ={ <u>NONE</u> <u>STANDARD</u> <i>kennwort</i> (<i>kennwort</i> , <i>kenn-</i> <i>wort</i> ,...)} }	STD	Kennwort für Dateien festlegen, das der DBH neben PP CATPASS zum Eröffnen von Dateien verwenden muss
<u>PP</u> <u>PRIVACY-CHECK</u> ={ <u>STD</u> <u>NO-KSET</u> <u>OFF</u> }	STD	Behandlung der Privacy-Prüfung steuern
<u>PP</u> <u>PTCSYNCH</u> = ([{ <u>WAIT</u> <u>ABORT</u> <u>COMMIT</u> }] [, [{ <u>WAIT</u> <u>ABORT</u> <u>COMMIT</u> }]])	(WAIT,WAIT)	Behandlung von Transaktionen im Zustand PTC steuern; der erste Wert gilt bei einem Warmstart, der zweite gilt während der laufenden Session, wenn der Zustand der primären Teiltransaktion nicht ermittelbar ist
<u>PP</u> <u>RESERVE</u> = { <u>NONE</u> <i>:catid:</i> <u>PUBLIC</u> (<i>priv-vsn-1/gerät-1</i> [, <i>priv-vsn-2/gerät-2</i> [, <i>priv-vsn-3/gerät-3</i>]]) (<i>vsn-1</i> [, <i>vsn-2</i> [, <i>vsn-3</i>]])}	NONE	Ersatzdatenträger für die RLOG-Dateien festlegen
<u>PP</u> <u>RESULT-DELAY</u> = <i>n</i>	0	Auftragsantworten an die Anwendertasks bündeln; <i>n</i> =1.. <i>m</i> <i>m</i> =PP TRANSACTION
<u>PP</u> <u>SCHEDULING</u> ={ <u>SYMMETRIC</u> <u>ASYMMETRIC</u> }	SYMMETRIC	Optimierung der Kommunikation zwischen Anwendertask und Servertask sowie die Abarbeitung anstehender DML-Aufträge in der Servertask steuern
<u>PP</u> <u>SERVERTASK</u> = <i>n</i>	1	Anzahl der Servertasks des independent DBH festlegen <i>n</i> =1..30
<u>PP</u> <u>SIP-SIZE</u> = <i>n</i>	1024 ¹ 128 ²	Größe des SSITAB-Pools in Kbyte festlegen; <i>n</i> =1..16384
<u>PP</u> <u>SQL</u> = <i>n</i>	4	max. Anzahl der gleichzeitig aktiven SQL-Vorgänge festlegen; <i>n</i> =0..9999
<u>PP</u> <u>SQL-LIMIT</u> = <i>n</i>	10	Zeitspanne einstellen, für die UDS/SQL die vorgangsspezifischen Daten inaktiver Vorgänge mindestens aufbewahrt; <i>n</i> =5..999 Minuten

<u>PP</u> <u>STDCKPT</u> ={ <u>YES</u> <u>NO</u> }	NO	Standard-Konsistenzpunkte im AFIM-Logging festschreiben beim DBH-Start, DBH-Ende und Session-Wiederanlauf
<u>PP</u> <u>SUBSCHEMA</u> = <i>n</i>	1	max. Anzahl der gleichzeitig innerhalb einer Datenbank verwendbaren Subschemas festlegen; <i>n</i> =1..100
<u>PP</u> <u>SUBTRANSACTION</u> = <i>n</i>	0	max. Anzahl der parallel eröffneten Dateikennungen pro Datenbank festlegen (nur KDBS); <i>n</i> =1..254
<u>PP</u> <u>TA-ACCESS</u> ={ <u>STD</u> <u>SHARED</u> }	STD	Benutzungsarten für Transaktionen festlegen
<u>PP</u> <u>TRANSACTION</u> ={ <i>n</i> ([<i>n</i>][<i>,</i>] <i>m</i>)}	(4,1)	<i>n</i> . maximale Anzahl der gleichzeitig aktiven Transaktionen und Anwendertasks <i>n</i> = 1..225; <i>m</i> . maximale Anzahl sekundärer Teiltransaktionen, die dieser DBH gleichzeitig bearbeiten kann <i>m</i> = 1.. <i>n</i> und <i>m</i> <= <i>n</i>
<u>PP</u> <u>UCON</u> =C' (<i>mn</i>) < <i>x</i> <i>nnnn</i> }' [, { <u>MSG</u> <u>UDS</u> }]	C' <U'	Bedienstation (UCON) festlegen, auf die die DCAM-Administration protokolliert werden soll
<u>PP</u> <u>WAIT</u> ={ <u>EVENT</u> <u>BUSY</u> }	EVENT	Warte-Modus einstellen
<u>PP</u> <u>WARMSTART</u> ={ <u>STD</u> <u>FAST</u> <u>VERY-FAST</u> }	STD	Dauer eines Warmstarts beeinflussen

Tabelle 8: Ladeparameter des independent DBH

¹ bei Angabe PP MPSEG=STD² bei Angabe PP MPSEG=64KDer **linked-in DBH** verarbeitet folgende Ladeparameter:

Parameter	Standardwert	Bedeutung
<u>PP</u> <u>2KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 2-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
<u>PP</u> <u>4KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 4-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047

<u>PP</u> <u>8KB-BUFFER-SIZE</u> = <i>n</i>	0	Größe des 8-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> ≠0 oder <i>n</i> ≠3..2047
<u>PP</u> <u>CATPASS</u> ={ <u>STANDARD</u> <i>kennwort</i> }	STD	Kennwort für vom DBH einzurichtende Dateien wie ALOG-Dateien und Temporäre Realms festlegen
<u>PP</u> <u>CONSOLE</u> ={ <u>YES</u> <u>NO</u> }	NO	Meldungen des linked-in DBH zusätzlich auf der Operatorkonsole ausgeben
<u>PP</u> <u>DBNAME</u> =[<i>\$userid.</i>] <i>dbname</i> [<i>.copyname</i>] [, <u>SHARED-RETRIEVAL</u>] [, { <i>n</i> [<i>n</i>] , <i>bufferid</i> }]	-	Datenbanken der DB-Konfiguration benennen; max. 222 Datenbanken <i>n</i> .Größe des User Buffer Pools in Mbyte festlegen; <i>n</i> ≠0..2047 <i>bufferid</i> : Identifikator des Bufferpools
<u>PP</u> <u>END</u>	-	Eingabe der Ladeparameter beenden
<u>PP</u> <u>LOG</u> ={ <u>NO</u> : <i>catid</i> : <u>PUBLIC</u> (<i>priv-vsn-1</i> / <i>gerät-1</i> [, <i>priv-vsn-2</i> / <i>gerät-2</i> [, <i>priv-vsn-3</i> / <i>gerät-3</i>]]) (<i>vsn-1</i> [, <i>vsn-2</i> [, <i>vsn-3</i>]]) }	-	RLOG-Datei führen
<u>PP</u> <u>LOG-2</u> = { : <i>catid</i> : <u>PUBLIC</u> (<i>priv-vsn-1</i> / <i>gerät-1</i> [, <i>priv-vsn-2</i> / <i>gerät-2</i> [, <i>priv-vsn-3</i> / <i>gerät-3</i>]]) (<i>vsn-1</i> [, <i>vsn-2</i> [, <i>vsn-3</i>]]) }	-	RLOG-Duplikatdatei führen
<u>PP</u> <u>LOG-SIZE</u> =([<i>primär</i>] [, [<i>sekundär</i>]])	192,192	Speicherplatzgröße der RLOG-Dateien in Anzahl PAM-Seiten festlegen
<u>PP</u> <u>MAXDB</u> = <i>n</i>	Summe von PP <u>DBNAME</u>	max. Anzahl Datenbanken der DB-Konfiguration festlegen; <i>n</i> ≠1..222
<u>PP</u> <u>PARLIST</u> ={ <u>YES</u> <u>NO</u> }	NO	Verwendete Ladeparameter auflisten
<u>PP</u> <u>PASSWORD</u> ={ <u>NONE</u> <u>STANDARD</u> <i>kennwort</i> (<i>kennwort</i> , <i>kenn-</i> <i>wort</i> , ...) }	STD	Kennwort für Dateien festlegen, das der DBH neben PP <u>CATPASS</u> zum Eröffnen von Dateien verwenden muss

<u>PP PRIVACY-CHECK</u> ={ <u>STD</u> <u>NO-KSET</u> <u>OFF</u> }	STD	Behandlung der Privacy-Prüfung steuern
<u>PP RESERVE</u> = { <u>NONE</u> :catid: <u>PUBLIC</u> (priv-vsn-1/gerät-1 [,priv-vsn-2/gerät-2 [,priv-vsn-3/gerät-3]]) (vsn-1[,vsn-2[,vsn-3]]) }	NONE	Ersatzdatenträger für die RLOG-Dateien festlegen
<u>PP STDCKPT</u> ={ <u>YES</u> <u>NO</u> }	NO	Standard-Konsistenzpunkte im AFIM-Logging festschreiben bei DBH-Start, DBH-Ende und Session-Wiederanlauf
<u>PP SUBSCHEMA</u> = <i>n</i>	1	max. Anzahl der gleichzeitig innerhalb einer Datenbank verwendbaren Subschemata festlegen; <i>n</i> =1..100
<u>PP SUBTRANSACTION</u> = <i>n</i>	0	max. Anzahl der parallel eröffneten Dateikennungen pro Datenbank festlegen (nur KDBS); <i>n</i> =1..254
<u>PP TRANSACTION</u> = <u>1</u>	1	Anzahl der gleichzeitig aktiven Transaktionen festlegen
<u>PP WARMSTART</u> ={ <u>STD</u> <u>FAST</u> <u>VERY-FAST</u> }	STD	Dauer eines Warmstarts beeinflussen

Tabelle 9: Ladeparameter des linked-in DBH

Der Wert des Ladeparameters PP TRANSACTION ist fest eingestellt; andere Angaben werden ignoriert. Diesen Parameter anzugeben, ist daher nicht nötig.

3.1 Größe des 2-Kbyte System Buffer Pools festlegen (2KB-BUFFER-SIZE)

```
PP 2KB-BUFFER-SIZE=n
```

Standardwert:

1

n Größe des System Buffer Pools in Mbyte, der die Realm-Seiten von Datenbanken mit einem 2-Kbyte-Seitenformat puffert

n = 1..2047

Je größer Sie den System Buffer Pool wählen, desto seltener muss der DBH auf die Realms der Datenbanken zugreifen, die der DB-Konfiguration angehören. Allerdings erhöht sich dadurch - je nach Größe des Hauptspeichers - wiederum das Paging, was den Datenbankbetrieb unter Umständen verzögert.

Es verlängert sich auch die Startphase von UDS/SQL, das Festschreiben von Konsistenzpunkten sowie die Datenbankrekonstruktion (siehe hierzu auch Ladeparameter „PP WARMSTART“ im Kapitel "[Dauer eines Warmstarts beeinflussen \(WARMSTART\)](#)".)

Voraussetzung, damit der angegebene Wert sinnvoll ist, ist eine entsprechend ausgelegte Anlagenkonfiguration. Es ist nicht sinnvoll den Maximalwert zu verwenden, da der gesamte Adressraum nur für den System Buffer Pool belegt wäre.

3.2 Größe des 4-Kbyte System Buffer Pools festlegen (4KB-BUFFER-SIZE)

```
PP 4KB-BUFFER-SIZE=n
```

Standardwert:

1

n Größe des System Buffer Pools in Mbyte, der die Realm-Seiten von Datenbanken mit einem 4-Kbyte-Seitenformat sowie die Seiten der Temporären Realms puffert

n = 1..2047

Je größer Sie den System Buffer Pool wählen, desto seltener muss der DBH auf die Realms der Datenbanken zugreifen, die der DB-Konfiguration angehören. Allerdings erhöht sich dadurch - je nach Größe des Hauptspeichers - wiederum das Paging, was den Datenbankbetrieb unter Umständen verzögert.

Es verlängert sich auch die Startphase von UDS/SQL, das Festschreiben von Konsistenzpunkten sowie die Datenbankrekonstruktion (siehe hierzu auch Ladeparameter „PP WARMSTART“ im Kapitel "[Dauer eines Warmstarts beeinflussen \(WARMSTART\)](#)".)

Voraussetzung, damit der angegebene Wert sinnvoll ist, ist eine entsprechend ausgelegte Anlagenkonfiguration. Es ist nicht sinnvoll den Maximalwert zu verwenden, da der gesamte Adressraum nur für den System Buffer Pool belegt wäre.

3.3 Größe des 8-Kbyte System Buffer Pools festlegen (8KB-BUFFER-SIZE)

```
PP 8KB-BUFFER-SIZE=n
```

Standardwert:

0

n Größe des System Buffer Pools in Mbyte, der die Realm-Seiten von Datenbanken mit einem 8-Kbyte-Seitenformat puffert

n = 0 oder

n = 3..2047

Wenn in der Session keine Datenbanken mit 8-Kbyte-Seitenformat bearbeitet werden sollen, können Sie verhindern, dass ein 8-Kbyte System Buffer Pool angelegt wird, indem Sie entweder den Ladeparameter gar nicht oder PP 8KB-BUFFER-SIZE=0 angeben.

Je größer Sie den System Buffer Pool wählen, desto seltener muss der DBH auf die Realms der Datenbanken zugreifen, die der DB-Konfiguration angehören. Allerdings erhöht sich dadurch - je nach Größe des Hauptspeichers - wiederum das Paging, was den Datenbankbetrieb unter Umständen verzögert.

Es verlängert sich auch die Startphase von UDS/SQL, das Festschreiben von Konsistenzpunkten sowie die Datenbankrekonstruktion (siehe hierzu auch Ladeparameter „PP WARMSTART“ im Kapitel "[Dauer eines Warmstarts beeinflussen \(WARMSTART\)](#)").

Voraussetzung, damit der angegebene Wert sinnvoll ist, ist eine entsprechend ausgelegte Anlagenkonfiguration. Es ist nicht sinnvoll den Maximalwert zu verwenden, da der gesamte Adressraum nur für den System Buffer Pool belegt wäre.

3.4 Administrierungsverfahren festlegen (ADM)

Nur **independent DBH!**

```
PP ADM={REMOTE | LOCAL}
```

Standardwert:

REMOTE

REMOTE

Administrieren des DBH ist über DCAM möglich, entweder mit dem Administrationsprogramm UDSADM (wird empfohlen) oder durch Anschließen einer Datensichtstation direkt an DCAM. (Siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#), bzw. [Abschnitt „UDS/SQL über DCAM administrieren“](#)). Es wird beim Initialisieren von UDS/SQL eine DCAM-Anwendung eröffnet. Als Anwendungsname gilt der Konfigurationsname bzw. die ersten acht Buchstaben von *konfigurationsname*.

LOCAL

Die Mastertask belegt die Datensichtstation permanent, bzw. die Administration erfolgt über den Bedienplatz mit dem Kommando INFORM-PROGRAM. Es wird keine DCAM-Anwendung eröffnet.

3.5 Kennwort für die Administration über DCAM festlegen (ADMPASS)

Nur **independent DBH!**

```
PP ADMPASS=admkenwort
```

admkenwort

Vereinbart ein Kennwort für die Administration, wenn PP ADM=REMOTE angegeben wurde. (Siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#), bzw. [Abschnitt „UDS/SQL über DCAM administrieren“](#)). Es kann bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C'*xxxx*':

Ein bis vier alphanumerische Zeichen

X'*nnnnnnnnri*':

Ein bis acht Sedezimalziffern

Der bei PP ADMPASS angegebene Wert kann dynamisch verändert werden mit den DAL-Kommandos ADD, siehe "[Datenbanken, Realms und Kennwörter anschließen \(ADD\)](#)" und DROP, siehe "[Datenbanken, Realms und Kennwörter ausschließen \(DROP\)](#)".

i Sofern Kleinbuchstaben genutzt werden sollen, sollten Sie das X-Format verwenden, da teilweise Eingabeschnittstellen (z.B. INTR) Kleinbuchstaben in Großbuchstaben umwandeln.

3.6 Präfix für Anwendertasks in UDS-D-Verteilungen festlegen (BCAM-PREFIX)

Nur **independent DBH!**

```
PP BCAM-PREFIX=prefix
```

Standardwert:

SUD\$

prefix Vereinbart ein Präfix für BCAM-Namen für die verteilte Kommunikation zwischen Anwendertask und entfernter UDS/SQL-Konfiguration. Dieses Präfix ist speziell dann erforderlich, wenn innerhalb eines BS2000 verschiedene virtuelle Hosts definiert sind und mehrere UDS/SQL-Konfigurationen auf virtuelle Hosts verteilt werden sollen (siehe auch [Abschnitt „Einsatz der UDS/SQL-Konfiguration auf virtuellem Host“](#)).

prefix muss vier Zeichen lang sein und so gewählt werden, dass die damit erzeugten BCAM-Namen erlaubte Namen sind und diese sich von Namen anderer BCAM-Anwendungen in den ersten vier Zeichen unterscheiden.

Das Präfix UDSD ist nicht erlaubt, weil es für die abwärtskompatible Verbindung zu DCAM_BAS-Konfigurationen reserviert ist.

i Sofern für den Ladeparameter BCAM-PREFIX ein vom Standardwert verschiedener Wert genutzt wird, wird empfohlen, auf allen über UDS-D beteiligten, entfernten Konfigurationen mindestens UDS-D V2.6 einzusetzen - auch in solchen beteiligten Konfigurationen, die nicht auf virtuellen Hosts ablaufen. Ansonsten kann es in Konfigurationen mit älteren UDS-D-Versionen zu internen Betriebsmittelengpässen kommen, die auch die sonstige DML-Bearbeitung behindern können.

3.7 Kennwort für vom DBH einzurichtende Dateien festlegen (CATPASS)

```
PP CATPASS={STANDARD | kennwort}
```

Standardwert:

STANDARD

STANDARD

UDS/SQL schützt neu einzurichtende Dateien mit dem festen Lese-Kennwort: C'UDS'BLANK' '

kennwort

Kennwort darf bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C'*xxxx*':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X'*nnnnnnnnri*:

nnnnnnnn sind ein bis acht Sedezimalziffern

d:

d'ist eine Dezimalzahl mit höchstens acht Ziffern und Vorzeichen, deren Wert zu einem Binärwert umgesetzt wird

Es gilt die BS2000-Syntax (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-PASSWORD)

Durch PP CATPASS geben Sie das Lese-Kennwort an, mit dem solche Dateien geschützt werden sollen, die vom DBH bei Bedarf eingerichtet werden. Dies können ALOG-Dateien und Temporäre Realms sein. Das angegebene Kennwort gilt pauschal für alle diese Dateien. Bei den durch die Dienstprogramme eingerichteten ALOG-Dateien ist das Kennwort C'UDS'BLANK" .

Die vom Anwender eingerichteten DB-Status-Dateien und die Session-Log-File (SLF) werden unabhängig von PP CATPASS **immer** mit C'UDS'BLANK" geschützt. Mit diesem Kennwort soll nicht der Inhalt der Dateien vor unberechtigtem Zugriff geschützt werden, sondern es soll ein versehentliches Löschen dieser sessionübergreifend genutzten Dateien verhindert werden. Diese Dateien enthalten nur DB-Verwaltungsdaten. Bei DB-Status-Dateien können für openUTM-Status-Anfragen an UDS/SQL auch konfigurationsfremde Status-Dateien gebraucht werden. Diese Regelung vermeidet dabei Kennwortkonflikte (siehe [Abschnitt „Kennwörter für UDS/SQL-Dateien vergeben“](#)). Bei der SLF ist diese Regelung notwendig, damit der DBH auf die SLF bereits zugreifen kann, bevor er die DBH-Ladeparameter ausgewertet hat.

Das Kennwort C'UDS'BLANK" und über PP CATPASS, PP PASSWORD bzw. das DAL-Kommando ADD PW angegebene Lese-Kennwörter sollten nicht für andere Dateien verwendet werden, da UDS/SQL diese Kennwörter intern absetzt und somit in Sondersituationen ein Zugriff auf Dateien möglich ist, für die dies aus Anwendersicht nicht erwünscht ist.

Die RLOG-Dateien werden durch ein Kennwort geschützt, das aus dem Zeitstempel der RLOG-Datei abgeleitet wird.

Mit dem Kennwort soll insbesondere auch ein versehentliches Löschen verhindert werden. Da die RLOG-Dateien für die Konsistenz der Daten im laufenden Betrieb ein sehr große Bedeutung haben, wird jede RLOG-Datei mit einem eigenen Kennwort geschützt. Im Allgemeinen löscht der DBH RLOG-Dateien, wenn sie nicht mehr benötigt werden. Sofern in Ausnahmefällen ein Löschen einer RLOG-Datei aus anderen Gründen nötig wird, können Sie dazu das Kennwort mit dem Dienstprogramm RLOGPASS der Liefereinheit UDS-SQL-T absetzen.

3.8 Verbindung und Transaktion überwachen (CHCKTIME)

Für UDS-D

```
PP CHCKTIME=n
```

Standardwert:

60

n Zeitintervall in Sekunden, in dem die UDS-D-Task überprüft, ob eine eingegangene DML-Anweisung noch von einer der Servertasks bearbeitet wird.

In diesem Fall sendet die UDS-D-Task auch eine Zeitquittung an Anwendertasks, die auf Antwort einer DML-Anweisung warten, welche in einem entfernten Verarbeitungsrechner bearbeitet wird.

Die Zeitquittung soll zeigen, dass die DML-Anweisung noch bearbeitet wird und dass die logische Verbindung noch besteht.

Das mit PP CHCKTIME eingestellte Zeitintervall wird für weitere Überwachungsfunktionen herangezogen (siehe [Abschnitt „Zeitgesteuertes Überwachen von Verbindungen und Transaktionen“](#)).

n = 60..900

PP CHCKTIME wirkt nur, wenn der UDS-D-Betrieb gestartet ist.

3.9 Meldungen des linked-in DBH auf Operatorkonsole ausgeben (CONSOLE)

Nur **linked-in DBH!**

```
PP CONSOLE={YES | NO}
```

Standardwert:

NO

YES UDS/SQL gibt alle Meldungen zusätzlich auf der Operatorkonsole aus. Das gilt unabhängig davon, ob der linked-in DBH als Dialogauftrag oder als ENTER-Job gestartet wurde.

NO UDS/SQL gibt alle Meldungen nach dem Einlesen der DBH-Ladeparameter nur auf der Datensichtstation aus, von der aus der linked-in DBH gestartet wurde (bzw. im entsprechenden SYSOUT-Protokoll bei Batch-Jobs).

3.10 Mindestgröße für einzelne Common Pools festlegen (CP-SIZE)

Nur **independent DBH!**

```
PP CP-SIZE=n
```

Mindestwert:

1024 Kbyte

n Speichergröße eines Common Pools in Kbyte

n = 1..16384

n wird auf ein ganzzahliges Vielfaches von 1024 aufgerundet.

Bei *n* > 16384 verwendet der DBH 16384 Kbyte.

In einer Session werden Common Pools dynamisch angelegt. Die einzelnen Common Pools werden in der Größe PP CP-SIZE oder, wenn diese Mindestgröße nicht ausreicht, in der Größe des vom DBH benötigten Wertes angelegt.

Mit dem Ladeparameter PP PARLIST oder dem DAL-Kommando DISPLAY PP wird bei CP-SIZE der Mindestwert eines Common Pools ausgegeben.

3.11 Prozessortyp angeben (CPU)

Nur **independent DBH!**

```
PP CPU= { MONO-PROCESSOR | MULTI-PROCESSOR }
```

Standardwert:

MONO-PROCESSOR

MONO-PROCESSOR

Die UDS/SQL-Session läuft auf einem Verarbeitungsrechner mit einem Monoprozessor ab.

MULTI-PROCESSOR

Die UDS/SQL-Session läuft auf einem Verarbeitungsrechner mit einem Multiprozessor ab.

Durch PP CPU geben Sie an, ob UDS/SQL auf einem Mono- oder Multiprozessor abläuft. Nach dieser Angabe wählt UDS/SQL einen optimalen Sperrmechanismus aus, um seine UDS/SQL-internen Abläufe zu serialisieren. Wenn Sie bei PP CPU Angaben machen, die nicht zutreffen, müssen Sie mit Einbußen im Laufzeitverhalten rechnen.

3.12 Mindestgröße des Communication Pools festlegen (CUP-SIZE)

Nur **independent DBH!**

```
PP CUP-SIZE=n
```

Mindestwert:

1024 Kbyte bei Angabe von PP MPSEG=STD

128 Kbyte bei Angabe von PP MPSEG=64K

n Speichergröße des Communication Pools in Kbyte

$n = 1..16384$

n wird bei Angabe von PP MPSEG=STD auf ein ganzzahliges Vielfaches von 1024 aufgerundet, bei Angabe von PP MPSEG=64K auf ein ganzzahliges Vielfaches von 64. Bei PP MPSEG=64K sind Werte über 8192 nicht sinnvoll.

Bei $n > 16384$ verwendet der DBH 16384 Kbyte.

Wenn der DBH einen größeren Speicherplatzbedarf als den Mindestwert oder den angegebenen Wert ermittelt, wird der Communication Pool in der errechneten Größe angelegt.

Der Communication Pool wird bei PP MPSEG=STD im oberen Adressraum der UDS/SQL-Tasks (> 16 MB) angelegt. In Anwendertasks wird der Communication Pool abhängig vom AMODE (24/31) im unteren oder oberen Adressraum angelegt. Bei PP MPSEG=64K liegt der Communication Pool in allen Tasks im unteren Adressraum. Falls in der Session auch Anwendungen im AMODE 24 genutzt werden, darf der Communication Pool nur so groß gewählt werden, dass er noch in den unteren Adressraum dieser Anwendung passt.

Mit dem Ladeparameter PP PARLIST oder dem DAL-Kommando DISPLAY PP wird bei CUP-SIZE ausgegeben, in welcher tatsächlichen Größe der Communication Pool angelegt wurde. Wenn die durchschnittliche Subschemagröße oder Auftragslänge einer SQL-Anweisung 16 Kbyte übersteigt, sollten Sie die tatsächliche Größe mit PP CUP-SIZE erhöhen.

3.13 Datenbanken benennen (DBNAME)

```
PP DBNAME=[ $userid. ]dbname[ .copyname ][ , SHARED-RETRIEVAL ]
[ , {n | [n],bufferid} ]
```

Standardwert:

hängt davon ab, ob auch PP MAXDB angegeben wurde oder nicht:

ohne PP MAXDB

DBH-Start mit nur einer Datenbank (Mono-DB-Betrieb)

PP DBNAME=*konfname*

mit PP MAXDB

DBH-Start mit leerer Konfiguration

(Multi-DB-Betrieb ist möglich, wenn PP MAXDB > 1 ist).

\$userid Kennung, unter der die Datenbank steht. *\$userid* ist nur erforderlich, wenn die Datenbank nicht in der Kennung katalogisiert ist, unter der der DBH gestartet wurde.

dbname Name der Datenbank

copyname

Name der Schattendatenbank

copyname besteht aus maximal sieben Zeichen.

SHARED-RETRIEVAL

Auf die anzuschließende Datenbank können auch andere DBHs SHARED-RETRIEVAL zugreifen.

Anwenderprogramme aller dieser DBHs können dann lesend auf die angeschlossene Datenbank zugreifen.

i Wenn Sie bei diesem Ladeparameter alle möglichen Parameter angeben, müssen Sie das Schlüsselwort SHARED-RETRIEVAL mit SHA abkürzen.

Standardwert:

Wenn Sie SHARED-RETRIEVAL nicht angeben, gilt EXCLUSIVE-UPDATE, d.h. auf die anzuschließende Datenbank kann der DBH exklusiv zugreifen; Anwenderprogramme können lesend und ändernd zugreifen.

Standardwert für eine Schattendatenbank: SHARED-RETRIEVAL

i Wenn der DBH auf die anzuschließende Datenbank exklusiv zugreift (EXCLUSIVE-UPDATE), Anwenderprogramme aber nur lesend zugreifen dürfen, müssen Sie zusätzlich das DAL-Kommando ACCESS RETRIEVAL,DB=*dbname* angeben.

n Größe des User Buffer Pools für die Datenbank in Mbyte. Ist der Parameter *bufferid* angegeben, so wird der mit *n* spezifizierte Buffer Pool als Shared User Buffer Pool angelegt, (s. u.).

n = 1 .. 2047

Bei einer Datenbank im 8-Kbyte-Seitenformat werden mindestens 3 Mbyte angelegt.

$n = 0$ Die Größe des User Buffer Pools ist bereits bei einer anderen Datenbank definiert.

n nicht angegeben und *bufferid* nicht angegeben

Die Datenbank wird in dem ihrem Seitenformat zugehörigen System Buffer Pool gepuffert. Wenn dieser System Buffer Pool nicht existiert, wird das Zuschalten der Datenbank abgewiesen.

n nicht angegeben und *bufferid* angegeben

Die Größe des User Buffer Pools ist bereits bei einer anderen Datenbank definiert.

i Der Wert n muss direkt nach dem Komma angegeben werden und darf nicht durch Leerzeichen unterbrochen werden (siehe "[Ladeparameter des DBH](#)").

Beispiel

PP DBNAME=*dbname* , n

bufferid Identifikator des Buffer Pools

max. 6 alphanumerische Zeichen (keine Sonderzeichen)

bufferid muss angegeben werden, wenn der mit n spezifizierte User Buffer Pool als gemeinsamer Buffer Pool mehrerer Datenbanken verwendet werden soll.

Mit PP DBNAME geben Sie an, welche Datenbanken beim DBH-Start an die DB-Konfiguration angeschlossen werden sollen. Dabei wird bei anzuschließenden inkonsistenten Datenbanken mit dem Warmstart die Konsistenz wiederhergestellt.

Außerdem können Sie angeben, ob auf die benannte Datenbank nur lesend zugegriffen werden soll. Wenn Sie die SHARED-RETRIEVAL-Angabe weglassen, so kann lesend und ändernd auf die Datenbank zugegriffen werden.

Sie können auch die Größe eines exklusiven Buffer Pools für die angegebene Datenbank spezifizieren. Dieser Buffer Pool wird zusätzlich zu den System Buffer Pools angelegt und ausschließlich für die Pufferung von Seiten der angegebenen Datenbank verwendet. Wenn Sie dem Buffer Pool mit dem Parameter *bufferid* einen Identifikator zuweisen, können Sie ihn als Shared User Buffer Pool mehreren Datenbanken als gemeinsamen exklusiven Buffer Pool zuweisen.

In einer Session kann es mehrere Shared User Buffer Pools geben.

PP DBNAME müssen Sie **einmal pro** anzuschließender Datenbank angeben. Dies gilt für alle Datenbanken, die sofort beim DBH-Start angeschlossen werden sollen.

PP DBNAME dürfen Sie max. 222 mal angeben.

Mehrere Datenbanken mit gleichem *dbname* sind innerhalb einer DB-Konfiguration unzulässig.

Wenn Datenbanken nicht unter der Kennung liegen, unter der die Session gestartet wurde, müssen alle Realms, ALOG-Dateien und die HASHLIB mehrfach benutzbar sein (SHARE=YES).

! VORSICHT!

Die Datenbankdateien dürfen im Betrieb nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

Die Auswirkungen bei RETRIEVAL-Angabe bzw. keine RETRIEVAL-Angabe können Sie der folgenden Tabelle entnehmen:

Angabe	keine Angabe	SHARED-RETRIEVAL
Auswirkung		
Mit welcher Benutzungsart werden die Realms eröffnet?	UPDATE	RETRIEVAL
Wenn AFIM-Logging eingeschaltet ist, werden die ALOG-Dateien eröffnet und geschrieben?	ja	nein
Lässt der DBH einen Warmstart zu, wenn eine inkonsistente Datenbank zugeschaltet wird?	ja	nein
Ist READY UPDATE für diese Datenbank zulässig?	ja	nein
Können Realms mit DAL dynamisch zu- und abgeschaltet werden?	ja	nein
Können auch mehrere DBHs parallel mit SHARED-RETRIEVAL auf diese Datenbank zugreifen?	nein	ja

Tabelle 10: Auswirkungen der RETRIEVAL-Angabe bei PP DBNAME

Beim Zuschalten einer DB während der Initialisierung des DBH wird die Verträglichkeit der ALOG-Vorgaben mit einer bereits geprüften UDS/SQL-Pubset-Deklaration kontrolliert. Eine DB kann nicht zugeschaltet werden falls eine der ALOG-Logging-Angaben, die mit den Parametern DEFAULT-SUPPORT oder RESERVE-SUPPORT der BMEND-Anweisung START-LOG vorgenommen wurde, außerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt.

Fehlerfälle werden mit einer oder mehrerer der folgenden Meldungen angezeigt:

```
UDS0755 UDS-ANWENDERFEHLER: CATID FEHLT IN UDS-PUBSET-DEKLARATION (CURRENT):
<catid> , ALOG-DEFAULT ,DB: <dbname> (... ,tsn4)
```

bzw.

```
UDS0755 UDS-ANWENDERFEHLER: CATID FEHLT IN UDS-PUBSET-DEKLARATION (CURRENT):
<catid> , ALOG-RESERVE ,DB: <dbname> (... ,tsn4)
```

Anschließend folgt die Meldung:

```
UDS0720 UDS ZUSCHALT-AUFTRAG GESCHEITERT: ALOG-SUPPORT NOT WITHIN SCOPE OF
UDS-PUBSET-DEKLARATION (... ,tsn4)
```

Die nachfolgende Meldung

```
UDS0719 UDS ERZWINGT DROP DB=dbname ZUR FEHLERBEHANDLUNG (... ,tsn4)
```

ermöglicht die Zuordnung des Fehlers zur Datenbank. Die Vorgaben für ALOG DEFAULT-SUPPORT und RESERVE-SUPPORT können mit der BMEND-Anweisung SHOW-LOG-INFORMATION angezeigt werden.

Das Hochfahren des DBH (ohne die betroffene Datenbank) wird dadurch nicht verhindert.

Mono-DB-Betrieb

Wenn Sie PP DBNAME und PP MAXDB nicht angeben, so ist der im Kommando

```
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname
```

angegebene Konfigurationsname gleichzeitig der Datenbankname der einzigen Datenbank, die an dieser Session teilnimmt (Mono-DB-Betrieb).

Bitte beachten Sie, dass für diese Datenbank nicht die Benutzungsart SHARED-RETRIE-VAL möglich ist, und die Datenbank dann nicht von mehreren DBHs parallel ansprechbar ist (siehe [Abschnitt „Paralleler Datenbankbetrieb“](#)).

Einen solchen Mono-DB-Betrieb können Sie nur unter der Kennung starten, unter der die Datenbank gespeichert ist.

Hinweise bei Rekonfiguration

Wenn nach einem Session-Abbruch die bestehende Konfiguration geändert werden soll (Rekonfiguration), beachten Sie bitte Folgendes:

- Eine inkonsistente Datenbank kann mit RETRIEVAL-Angabe nicht angeschlossen werden, weil der DBH zum Warmstart mit EXCLUSIVE-UPDATE auf die Datenbank zugreifen muss.
In diesem Fall muss die Datenbank also zunächst **ohne** RETRIEVAL-Angabe angeschlossen und nach erfolgreichem Warmstart wieder ausgeschlossen werden. Mit dem ACCESS-Kommando lässt sich bei Bedarf verhindern, dass diese Datenbank dabei durch Transaktionen der laufenden Session angesprochen wird.
- Mehrere inkonsistente Datenbanken können **nicht gleichzeitig** an **verschiedene** DBHs angeschlossen werden, wenn die Inkonsistenz aus ein und derselben Session stammt. Dies liegt daran, dass die Warmstarts nur zeitlich nacheinander erfolgen können, weil der DBH auf die benötigte RLOG-Datei jeweils mit EXCLUSIVE-UPDATE zugreifen muss.
- Wenn die Inkonsistenz einer Datenbank unter dem Konfigurationsnamen A entstand, kann die inkonsistente Datenbank nur dann mit Warmstart an eine Konfiguration mit Konfigurationsnamen B angeschlossen werden, wenn gerade keine Session unter dem Konfigurationsnamen A läuft. Dies liegt daran, dass es pro Konfiguration nur eine DB-Status-Datei gibt und auf die sowohl in der laufenden Session als auch beim Warmstart mit EXCLUSIVE-UPDATE zugegriffen wird.

3.14 Servertasks auf Grund der Rechnerauslastung deaktivieren (DEACT)

Nur **independent DBH!**

```
PP DEACT={YES | NO}
```

Standardwert:

YES

YES gibt an, dass eine Servertask deaktiviert werden kann, d.h. dass der Servertask die Betriebsmittel entzogen werden.

Eine Deaktivierung kann in folgenden Fällen erfolgen:

- die Rechnerauslastung ist sehr hoch (CPU, Speicher, Paging-Rate)
- die Servertask ist lange im Wartezustand (z.B. nach „PASS“, „LOCK“, „MESSAGE“)
- die Servertask verbraucht zu viel Systemleistung (z.B. CPU)

NO gibt an, dass eine Servertask nicht deaktiviert werden kann; d.h. die Servertasks werden auch bei hoher Auslastung der Anlage, hoher verbrauchter Systemleistung und in Wartezuständen von anderen Tasks im System weniger beeinflusst und laufen dadurch performanter.

Allerdings werden dadurch andere Anwendungen eventuell benachteiligt.

Mit dem Ladeparameter PP DEACT können Sie steuern, wie das Betriebssystem die Servertasks in Situationen hoher Last behandeln soll.

Siehe auch [Abschnitt „Ladeparameter, die die Prozessornutzung beeinflussen“](#).

Voraussetzung für die Wirksamkeit dieses Parameters ist, dass für die Benutzerkennung, in der der DBH gestartet werden soll, mit dem ADD-USER-Kommando der Operand INHI-BIT DEACTIVATION=YES angegeben wurde (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“).

Wenn Sie den DEACT-Parameter verwenden, ist es wichtig, dass Sie die Anlagenauslastung genau kennen. Nur wenn Sie den Parameter richtig einsetzen, können Sie die Gesamtsituation verbessern.

3.15 Konfigurationsübergreifende Deadlocks bzw. Deadlocks im Zusammenhang mit openUTM auflösen (DEADTIME)

Nur **independent DBH!**

`PP DEADTIME=n`

Standardwert:

60

n Zeitangabe in Sekunden. Ein konfigurationsübergreifender Deadlock bzw. ein Deadlock im Zusammenhang mit openUTM wird *n* bis $1,5n$ Sekunden nach seinem Auftreten aufgelöst.

$n = 5..900$

Zu kleine Werte für PP DEADTIME können zu unnötigem Zurücksetzen von Transaktionen führen, z.B. bei IQS- oder openUTM-Mehrschritt-Transaktionen.

3.16 Mindestgröße des Distribution Pools festlegen (DIP-SIZE)

Für UDS-D

```
PP DIP-SIZE=n
```

Mindestwert:

1024 Kbyte bei Angabe von PP MPSEG=STD

64 Kbyte bei Angabe von PP MPSEG=64K

n Speichergröße des Distribution Pools in Kbyte

n = 1..16384

n wird bei Angabe von PP MPSEG=STD auf ein ganzzahliges Vielfaches von 1024 aufgerundet, bei Angabe von PP MPSEG=64K auf ein ganzzahliges Vielfaches von 64. Bei PP MPSEG=64K sind Werte über 8192 nicht sinnvoll.

Bei *n* > 16384 verwendet der DBH 16384.

Wenn der DBH einen größeren Speicherplatzbedarf als den Mindestwert oder den angegebenen Wert ermittelt, wird der Distribution Pool in der errechneten Größe angelegt.

Der Distribution Pool wird bei PP MPSEG=STD im oberen Adressraum der UDS/SQL-Tasks (> 16 MB) angelegt. In Anwendertasks wird der Distribution Pool abhängig vom AMODE (24/31) im unteren oder oberen Adressraum angelegt. Bei PP MPSEG=64K liegt der Distribution Pool in allen Tasks im unteren Adressraum. Falls in der Session auch Anwendungen im AMODE 24 genutzt werden, darf der Distribution Pool nur so groß gewählt werden, dass er noch in den unteren Adressraum dieser Anwendung passt.

Mit dem Ladeparameter PP PARLIST oder dem DAL-Kommando DISPLAY PP wird bei DIP-SIZE ausgegeben, in welcher Größe der Distribution Pool angelegt wurde.

3.17 Anzahl entfernter Datenbanken festlegen (DISDB)

Für UDS-D

```
PP DISDB=n
```

Standardwert:

1

n gibt die maximale Anzahl entfernter Datenbanken an, die pro Transaktion angesprochen werden können. Pro Transaktion können also höchstens *n* entfernte Verarbeitungsketten laufen.

n = 1..32

PP DISDB wirkt nur, wenn der DBH-Ladeparameter PP DISTRIBUTION != NO ist.

i Die DBH-Ladeparameter PP DISDB, PP MAXDB und PP TRANSACTION wirken sich auf die Größe des Communication Pools und des Distribution Pools aus. Es ist nicht sinnvoll, bei allen diesen drei DBH-Ladeparametern den Maximalwert zu wählen, da dadurch die Memory Pools sehr groß werden und es abhängig von der Adressraumsituation unterhalb 16 Mbyte zu einem Speicherengpass kommen kann.

3.18 Eingabedatei für die Verteiltabelle festlegen (DISTABLE)

Für UDS-D

```
PP DISTABLE=[ :catid: ][$userid.]dateiname
```

:catid.

BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

\$userid

Benutzerkennung, in der *dateiname* steht.

dateiname

Name der Eingabedatei für die Verteiltabelle.

UDS-D liest die Eingabedatei und erstellt daraus die Verteiltabelle (siehe [Abschnitt „Aufbauder Eingabedatei für die Verteiltabelle“](#)). Das Anwenderprogramm kann nur über die in der Verteiltabelle eingetragenen Subschemas auf entfernte Datenbanken zugreifen.

Wenn der DBH-Ladeparameter PP DISTABLE fehlt, können entfernte Anwenderprogramme zwar auf die lokale Konfiguration zugreifen, aber lokale Anwenderprogramme können nicht auf entfernte Konfigurationen zugreifen.

PP DISTABLE wirkt nur, wenn der DBH-Ladeparameter PP DISTRIBUTION != NO ist.

3.19 Teilnahme am UDS-D-Betrieb steuern (DISTRIBUTION)

Für UDS-D

```
PP DISTRIBUTION={NO | STANDBY | START}
```

Standardwert:

NO

NO In dieser Session können weder entfernte Anwenderprogramme auf die Datenbanken der lokalen Konfiguration noch lokale Anwenderprogramme auf Datenbanken entfernter Konfigurationen zugreifen. UDS-D-spezifische Tabellen werden nicht angelegt.

STANDBY

Der UDS-D-Betrieb wird nur vorbereitet, d.h. UDS-D-spezifische Tabellen werden angelegt, die UDS-D-Task UDSCT jedoch nicht gestartet.

In dieser Session ist die Teilnahme am UDS-D-Betrieb erst möglich, wenn mit dem DAL-Kommando &START DISTRIBUTION der UDS-D-Betrieb gestartet wird.

START

Der UDS-D-Betrieb wird beim DBH-Start aufgenommen, indem die UDS-D-Task UDSCT gestartet wird.

Wenn Sie den DBH-Ladeparameter PP DISTRIBUTION != NO setzen, dann sollte auch der DBH-Ladeparameter PP LOG != NO sein (siehe [Abschnitt „RLOG-Datei führen \(LOG\)“](#)).

Wenn der DBH-Ladeparameter PP LOG = NO ist, dann können entfernte Anwenderprogramme nur lesend auf die lokale Konfiguration zugreifen, und lokale Anwenderprogramme können nur lesend auf entfernte Konfigurationen zugreifen.

3.20 Umfang eines Dumps beeinflussen (DUMP)

Nur **independent DBH!**

```
PP DUMP={STD | ALL}
```

Standardwert:

ALL

STD Es wird ein reduzierter Dump erzeugt, abhängig von der Fehlersituation und der genutzten BS2000-Version.

ALL Es wird ein Dump vom gesamten Benutzeradressraum erzeugt.

Der Ladeparameter DUMP=ALL sollte insbesondere dann genutzt werden, wenn bei einem reproduzierbaren Fehler der über DUMP=STD erzeugte Dump für die Diagnose nicht ausreicht.

3.21 Parametereingabe beenden (END)

PP END

Mit PP END beenden Sie die Eingabe der Ladeparameter des DBH. PP END müssen Sie immer angeben.

3.22 I/Os asynchron oder synchron ausführen (IO)

Nur **independent DBH!**

```
PP IO={ASYNC | SYNC}
```

Standardwert

ASYNC

ASYNC

Die Servertasks führen alle I/Os asynchron durch. Damit werden I/O-bedingte Wartezeiten für die Servertasks vermieden, sodass auch in Hochlastsituationen kein Engpass auftritt.

Bei zu geringer Last kann diese Einstellung jedoch zu einem CPU-Mehrverbrauch führen.

SYNC

Die Servertasks führen alle I/Os synchron durch. Dadurch treten in den Servertasks I/O-bedingte Wartezeiten auf. Eventuell muss dann die Zahl der Servertasks erhöht werden (siehe Ladeparameter PP [SERVERTASK "Anzahl Servertasks festlegen \(SERVERTASK\)"](#)).

Diese Einstellung ist deshalb nur bei geringer Last oder geringem Anteil von openUTM-Transaktionen zu empfehlen.



Bei der Angabe PP IO=SYNC hat der Ladeparameter PP SCHEDULING keine Wirkung.

3.23 Sperrprotokoll festlegen (LOCK)

Nur **independent DBH!**

```
PP LOCK={STD | SHARED | EXCLUSIVE}
```

Standardwert:

STD

STD Ein gelesener Satz (siehe „Sperrgranulat“) ist bis zum Lesen des nächsten Satzes SHARED (siehe „Sperrschärfe“) gesperrt, ein geänderter Satz ist bis zum Transaktionsende EXCLUSIVE gesperrt. Eine Sperre, die durch die DML-Anweisung KEEP gesetzt wurde, ist bis zur DML-Anweisung FREE bei einer RETRIEVAL-Verarbeitung SHARED und bei einer UPDATE-Verarbeitung EXCLUSIVE.

SHARED

Wie bei STD, mit einer Ausnahme:

Eine Sperre, die durch die DML-Anweisung KEEP gesetzt wurde, ist immer SHARED.

EXCLUSIVE

Ein gelesener Satz wird bei einer UPDATE-Verarbeitung bis zum Transaktionsende EXCLUSIVE gesperrt.

Sperrgranulat:

Gesperrt ist immer die ganze Seite, auf welcher der Satz liegt.

Sperrschärfe:

SHARED:

Sperre gegen Ändern seitens einer fremden Transaktion

EXCLUSIVE:

Sperren gegen Lesen und Ändern seitens einer fremden Transaktion

Verarbeitungstyp:

Unter der Voraussetzung, dass auf einem Realm prinzipiell Änderungen zulässig sind (Datenbank mit UPDATE zugeschaltet, kein explizites Abschalten (DAL) oder implizites Abschalten (Fehlerbehandlung) oder Sperren von Realms oder Datenbanken, keine durch BPRIVACY gesetzte Realm-Sperre), ist im SQL-Fall der Verarbeitungstyp immer UPDATE, im Nicht-SQL-Fall genau dann, falls READY-USAGE-MODE=UPDATE.

Der Parameter bezieht sich auf alle Datenbanken, die an der Session teilnehmen.

PP LOCK=SHARED sollten Sie angeben, wenn Sie die DML-Anweisung KEEP nutzen, um Daten wiederholt lesen zu können und nicht zum Zwecke der Serialisierung gegenüber anderen Transaktionen.

PP LOCK=EXCLUSIVE kann durch eine Reduzierung der parallelen Verarbeitung zu kürzeren Pfadlängen und selteneren Deadlocks führen. PP LOCK=EXCLUSIVE sollten Sie nur angeben, wenn für alle UPDATE-Transaktionen Folgendes gilt:

- Alle gelesenen Sätze werden auch geändert.
- Es gibt keine sequenziell laufenden Suchfragen (FIND-7).

- Vor jedem Zugriff auf den Membersatz eines nicht-singulären Sets wird der Ownersatz gelesen und verändert.

Wenn eine dieser Voraussetzungen fehlt, kann es dazu führen, dass die Parallelität vermindert wird oder die erhoffte Einsparung der Pfadlänge nicht zu Stande kommt. Dann sollten Sie durch Performance-Vergleich überprüfen, ob der Einsatz von PP LOCK=EXCLUSIVE sinnvoll ist.

3.24 RLOG-Datei führen (LOG)

```
PP LOG={NO |
        :catid: |
        PUBLIC
        (priv-vsn-1/gerät-1[ ,priv-vsn-2/gerät-2[ ,priv-vsn-3/gerät-3] ] )
        (vsn-1[ ,vsn-2[ ,vsn-3] ] ) }
```

Standardwert:

Es gibt keinen Standardwert.

Sie müssen PP LOG eingeben, sonst wird der DBH-Start abgebrochen.

NO Der DBH führt keine RLOG-Datei;
bei LOG=NO lässt

- der independent DBH keine ändernden Transaktionen zu
- der linked-in DBH ändernde Transaktionen zu
- der independent DBH bei verteilter Verarbeitung keine über UDS-D verteilte UPDATE-Transaktionen zu (RETRIEVAL-Transaktionen sind erlaubt).

:catid: BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Die RLOG-Datei wird auf gemeinschaftlicher Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung angelegt.

priv-vsn

Die RLOG-Datei wird auf Privatplatten angelegt. Es können maximal drei Archivnummern angegeben werden, die verschieden sein müssen. Eine Archivnummer darf bis zu sechs alphanumerische Zeichen enthalten.

gerät Gerätetyp der privaten Platte. Der Gerätetyp darf bis zu acht alphanumerische Zeichen enthalten.

vsn Die RLOG-Datei wird auf Platten angelegt, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind. Es können maximal drei VSNs angegeben werden, die verschieden sein müssen.

Die VSN kann in PUB-Notation (PUBp_{xx}) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden. Bitte beachten Sie auch die [„Zusatzbedingungen für VSN-Angaben“](#).

Mit dem Ladeparameter PP LOG bestimmen Sie, ob der DBH für die zu startende Session eine RLOG-Datei einrichten soll oder nicht.

Sie geben die Datenträger für die Originaldateien an. Bei einfach und doppelt geführter RLOG-Datei bestimmen sie damit die Lage der Originaldateien.

Die Datenträger für die Duplikatdateien geben Sie bei doppelt geführter RLOG-Datei mit dem Ladeparameter PP LOG-2 an.

Ob die RLOG-Datei einfach oder doppelt geführt werden soll, legen Sie beim DBH-Start fest. Dies können Sie nicht mit dem DAL-Kommando MODIFY ändern.

Die Datenträger für die RLOG-Dateien können Sie mit dem DAL-Kommando MODIFY ändern.

Die Speicherplatzgröße der RLOG-Dateien geben Sie mit dem Ladeparameter PP LOG-SIZE an.

Ersatzdatenträger für einfach und doppelt geführte RLOG-Dateien können Sie mit dem Ladeparameter PP RESERVE angeben.

Während der Initialisierung des DBH wird die Verträglichkeit der PP LOG-Angaben mit einer bereits geprüften UDS /SQL-Pubset-Deklaration kontrolliert.

Im Fehlerfall erfolgt nach Ausgabe des betroffenen Ladeparameters die Meldung:

```
UDS0711 UDS LADE-PARAMETER ZURUECKGEWIESEN: CATID NOT WITHIN SCOPE OF UDS
PUBSET DECLARATION (... ,tsn4)
```

Das Hochfahren des DBH wird dann nach Abschluss der Ladeparameter-Prüfung abgebrochen.

Einfach geführte RLOG-Datei

Der DBH führt pro Konfiguration **eine** RLOG-Datei (Original). Diese RLOG-Datei besteht immer aus zwei physischen Originaldateien.

Dateinamen bei einfach geführter RLOG-Datei

```
konfname.RLOG.rlogzeitstempel.1
konfname.RLOG.rlogzeitstempel.2
```

konfname

gibt den Konfigurationsnamen an.

rlogzeitstempel

gibt den Zeitpunkt an, ab dem die RLOG-Datei vom DBH benutzt wird.
Beide Exemplare der RLOG-Datei erhalten die gleichen Zeitstempel.

rlogzeitstempel setzt sich folgendermaßen zusammen:

jjmmthhmmss

1,2:

Zur Unterscheidung der beiden zyklisch benutzten, physischen RLOG-Datei-Exemplare.

Doppelt geführte RLOG-Datei

Der DBH führt pro Konfiguration zwei RLOG-Dateien (Original und Duplikat).

Sie geben mit PP LOG die Datenträger für die beiden Originaldateien und mit PP LOG-2 die Datenträger für die beiden Duplikatdateien an.

Bei Angabe der Ladeparameter ist die Reihenfolge von PP LOG und PP LOG-2 beliebig.

Eine doppelt geführte RLOG-Datei ist nur möglich, wenn die Original- und Duplikatdatei auf verschiedenen Datenträgern liegen. Beim Initialisieren prüft UDS/SQL, ob die doppelt geführte Datei auf verschiedenen Datenträgern liegt. Wenn bereits bei der syntaktischen Analyse der DBH-Ladeparameter erkennbar ist, dass die Datenträger gleich sind, wird der DBH-Start abgebrochen. Andernfalls, d.h. wenn die Gleichheit erst beim Zugriff auf die Datenträger erkannt wird, werden die an der Session beteiligten Datenbanken mit einer UPDATE-Sperre belegt.

Dateinamen bei doppelt geführter RLOG-Datei

Der DBH führt **zwei** RLOG-Dateien, d.h. er führt ein Original und ein Duplikat (SAVE).

Folgende vier physische Dateien werden angelegt:

konfname.RLOG.rlogzeitstempel.1

konfname.RLOG.rlogzeitstempel.2

konfname.RLOG.rlogzeitstempel.1.SAVE

konfname.RLOG.rlogzeitstempel.2.SAVE

Wenn in einem Exemplar der RLOG-Datei ein Fehler auftritt, so ist durch das Führen des Duplikats trotz des Fehlers noch ein Rollback möglich, sodass die Datenbanken konsistent bleiben.

i Bei LOG=NO lässt der linked-in DBH auch ändernde Transaktionen zu! Wenn während einer solchen Session allerdings ein Fehler auftritt, der einen Rollback einer Transaktion oder einen Warmstart erfordert, so sind die Datenbanken der DB-Konfiguration, in denen während der Session geändert wurde, irreparabel **inkonsistent**. Sie müssten in diesem Fall auf eine Sicherung der Datenbanken aufsetzen und die inkonsistenten Datenbanken bis zum letztmöglichen Konsistenzpunkt mit dem Dienstprogramm BMEND nachfahren (Anweisung UPDATE-DATABASE,DEADLINE=STD). Daher sollten Sie LOG=NO nur verwenden, um nach Sicherung des Datenbestandes große Datenmengen schnell in Datenbanken zu laden.

! **VORSICHT!**

Die RLOG-Dateien dürfen nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

Zusatzbedingungen für VSN-Angaben

Für die VSN-Angaben bei Anlage der RLOG-Dateien auf Platten, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind, gelten folgende Besonderheiten:

- VSN-Angaben sind nur dann zulässig, wenn die Berechtigung zur physikalischen Allokierung von gemeinschaftlichem Speicherplatz auf dem betroffenen Pubset besteht.
- Sowohl in der PUB-Notation (PUBp_{xx}), als auch in der Punkt-Notation (pp[pp].[xy]z), ist es erlaubt, die laufende Nummer der Platte (xx bzw. [xy]z) wegzulassen. Dann ist nur eine einzige VSN-Angabe möglich.

Dadurch kann erreicht werden, dass die RLOG-Datei auf beliebigen Volumes eines Volume-Sets eines SM-Pubsets angelegt wird.

Beispiele:

(PUBX01,PUBX02)	RLOG-Datei auf Volumes PUBX01 und PUBX02 im einstelligen SF-Pubset X oder Volume-Set X eines SM-Pubsets
(PUBX)	RLOG-Datei auf beliebigem Volume im einstelligen SF-Pubset X oder Volume-Set X eines SM-Pubsets
(ABC.01,ABC.02)	RLOG-Datei auf Volumes ABC.01 und ABC.02 im dreistelligen SF-Pubset ABC oder Volume-Set ABC eines SM-Pubsets

(ABC.)

RLOG-Datei auf beliebigem Volume im dreistelligen SF-Pubset
ABC oder Volume-Set ABC eines SM-Pubsets

- Falls für eine RLOG-Datei mehrere Volumes angegeben werden, müssen diese demselben Volume-Set angehören, d.h. die VSNs müssen die gleiche Katalogkennung enthalten. Eine Datei kann sich nicht über mehrere Volume-Sets erstrecken.
- Doppelt geführte Logging-Dateien müssen auf unterschiedlichen Volume-Sets angelegt werden, da beim Ausfall eines Volumes in der Regel der gesamte Volume-Set nicht mehr zur Verfügung steht.
- Außerdem darf eine der doppelt geführten Logging-Dateien auch nicht auf einem Volume-Set eines SM-Pubset angelegt werden, wenn die andere auf Grund einer :catid:- oder PUBLIC-Angabe in diesem SM-Pubset angelegt wird.
- Bei der Angabe des Control Volume-Sets eines SM-Pubsets für ein RLOG-Exemplar oder zweier Volume-Sets eines SM-Pubsets für RLOG-Original und -Duplikat ist Folgendes zu beachten:
Bei Ausfall des Control Volume-Sets eines SM-Pubsets ist vorübergehend der gesamte SM-Pubset nicht mehr zugreifbar. (Eine Rekonstruktion des SM-Pubsets ist möglich.)

3.25 RLOG-Duplikatdatei führen (LOG-2)

```
PP LOG-2={:catid: |
          PUBLIC
          (priv-vsn-1/gerät-1[ ,priv-vsn-2/gerät-2[ ,priv-vsn-3/gerät-3] ] )
          (vsn-1[ ,vsn-2[ ,vsn-3] ] ) }
```

:catid BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Die RLOG-Duplikatdatei wird auf gemeinschaftlicher Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung angelegt.

priv-vsn

Das Duplikat der RLOG-Datei wird auf Privatplatten angelegt. Es können maximal drei Archivnummern angegeben werden, die verschieden sein müssen. Eine Archivnummer darf bis zu sechs alphanumerische Zeichen enthalten.

gerät Gerätetyp der privaten Platte. Der Gerätetyp darf bis zu acht alphanumerische Zeichen enthalten.

vsn Die RLOG-Datei wird auf Platten angelegt, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind. Es können maximal drei VSNs angegeben werden, die verschieden sein müssen.

Die VSN kann in PUB-Notation (PUBp_{xx}) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden. Bitte beachten Sie auch die [„Zusatzbedingungen für VSN-Angaben“ \(RLOG-Datei führen \(LOG\)\)](#).

Geben Sie den Ladeparameter PP LOG-2 an, wird die RLOG-Datei doppelt geführt. Mit dem Wert des Ladeparameters bestimmen Sie die Datenträger für die Duplikatdateien.

Die Datenträger für die Originaldateien geben Sie mit dem Ladeparameter PP LOG an. Die Reihenfolge von PP LOG und PP LOG-2 ist beliebig.

Ob die RLOG-Datei einfach oder doppelt geführt werden soll, legen Sie beim DBH-Start fest. Dies können Sie nicht mit dem DAL-Kommando MODIFY ändern. Die Datenträger für die RLOG-Dateien können Sie mit dem DAL-Kommando MODIFY ändern.

Eine doppelt geführte RLOG-Datei ist nur möglich, wenn Originaldatei und Duplikatdatei auf verschiedenen Datenträgern liegen. Beim Initialisieren prüft UDS/SQL, ob die doppelt geführte Datei auf verschiedenen Datenträgern liegt. Wenn die Datenträger gleich sind, wird der DBH-Start abgebrochen.

Bitte beachten Sie auch die Hinweise zu „Doppelt geführter RLOG-Datei“ auf ["RLOG-Datei führen \(LOG\)"](#).

Die Speicherplatzgröße der RLOG-Dateien geben Sie mit dem Ladeparameter PP LOG-SIZE an.

Ersatzdatenträger für die einfach und doppelt geführte RLOG-Datei können Sie mit dem Ladeparameter PP RESERVE angeben.

Während der Initialisierung des DBH wird die Verträglichkeit der PP LOG-2-Angaben mit einer bereits geprüften UDS/SQL-Pubset-Deklaration kontrolliert.

Im Fehlerfall erfolgt nach Ausgabe des betroffenen Ladeparameters die Meldung:

```
UDS0711 UDS LADE-PARAMETER ZURUECKGEWIESEN: CATID NOT WITHIN SCOPE OF UDS  
PUBSET DECLARATION (... ,tsn4)
```

Das Hochfahren des DBH wird dann nach Abschluss der Ladeparameter-Prüfung abgebrochen.

3.26 Speicherplatzgröße der RLOG-Dateien festlegen (LOG-SIZE)

```
PP LOG-SIZE=( [primär] [ , [sekundär] ] )
```

Standardwert:

jeweils 192 PAM-Seiten (je 2048 byte)

primär

Primärzuweisung in PAM-Seiten

primär = 1..9999999

sekundär

Sekundärzuweisung in PAM-Seiten

sekundär = 1..32767

Sie legen mit dem Ladeparameter PP LOG-SIZE fest:

- mit der Primärzuweisung, wie viel Speicherplatz jeder RLOG-Datei neu zugewiesen werden soll,
- mit der Sekundärzuweisung, wie viel Speicherplatz automatisch zugewiesen wird, wenn im Laufe der Dateibearbeitung weiterer Speicherplatz benötigt wird.

Wenn Sie für die Sekundärzuweisung einen kleineren Wert als den Standardwert 192 angeben, wird auf ihn aufgerundet.

Die Primärzuweisung sollte der erwarteten Größe der einzurichtenden Datei entsprechen.

Bei großen Dateien sollten Sie die Werte für die Primär- und Sekundärseiten als Vielfaches der Verwaltungseinheiten von 24 oder 192 PAM-Seiten wählen.

3.27 Anzahl Datenbanken der DB-Konfiguration festlegen (MAXDB)

```
PP MAXDB=n
```

Standardwert:

Anzahl der per Ladeparameter PP DBNAME angegebenen Datenbanken. Wenn DBNAME nicht angegeben wurde, nimmt der DBH Mono-DB-Betrieb an und setzt 1 ein.

n Anzahl der maximal an der Session beteiligten lokalen Datenbanken
n = 1..222

i Zusätzlich zu den 222 lokalen Datenbanken, können bis zu 32 entfernte Datenbanken angesprochen werden, d.h. insgesamt können 254 Datenbanken an der Session beteiligt sein.

Der Ladeparameter PP MAXDB begrenzt die Anzahl der Datenbanken, die an der zu startenden Session gleichzeitig angeschlossen sein können.

Wird durch dynamisches Anschließen von Datenbanken (ADD DB) die mit MAXDB angegebene Grenze überschritten, so weist der DBH das Anschließen der Datenbank zurück.

Der DBH setzt den Wert für *n* gleich der Anzahl der Ladeparameter PP DBNAME, wenn *n* kleiner ist als die Anzahl der angegebenen Datenbanken.

PP MAXDB **muss angegeben** werden bei einer Session, in der Datenbanken dynamisch angeschlossen werden und die Summe der angeschlossenen Datenbanken größer wird als bei Session-Beginn.

PP MAXDB **muss nicht angegeben** werden bei einer Session, in der Datenbanken nicht dynamisch angeschlossen werden.

i Die DBH-Ladeparameter PP DISDB, PP MAXDB und PP TRANSACTION wirken sich auf die Größe des Communication Pools und des Distribution Pools aus. Es ist nicht sinnvoll, bei allen diesen drei DBH-Ladeparametern den Maximalwert zu wählen, da dadurch die Memory Pools sehr groß werden und es abhängig von der Adressraumsituation unterhalb 16 Mbyte zu einem Speicherengpass kommen kann.

3.28 Segmentgröße der Memory Pools festlegen (MPSEG)

Nur **independent DBH!**

```
PP MPSEG={STD | 64K}
```

Standardwert:

STD

- STD** Die Memory Pools werden in den UDS/SQL-Tasks im oberen Adressraum (> 16 Mbyte) mit einer Segmentgröße von 1 Mbyte angelegt.
Anwenderprogramme im AMODE 31 schliessen sich an diese Pool so an, dass sie ebenfalls oberhalb 16 Mbyte im oberen Adressraum liegen.
Bei Anwenderprogrammen im AMODE 24 liegen diese Pools unterhalb 16 Mbyte in deren virtuellen Adressraum.
- 64K** Der Communication Pool, der Transfer Pool, der Distribution Pool und der SSITAB-Pool werden unterhalb von 16 Mbyte mit einer Segmentgröße von 64 Kbyte angelegt.
Die restlichen Memory Pools liegen oberhalb 16 Mbyte und werden mit der Segmentgröße von 1 Mbyte angelegt.
- Dieser Operand sollte nur angegeben werden bei Speicherplatz-Engpässen durch die 1 Mbyte Segmentgröße, da sich die Segmentgröße von 64 Kbyte negativ auf die Performance auswirken kann.

3.29 Systemverhalten bei Kollision von vorgemerkten Aufträgen und UPDATE-Transaktionen steuern (ORDER-DBSTATUS)

Nur **independent DBH!**

```
PP ORDER-DBSTATUS={STD | SPECIAL}
```

Standardwert:

STD

STD

Neue UPDATE-Transaktionen oder -Verarbeitungsketten, die mit der Ausführung vorliegender CHECKPOINT- oder NEW RLOG-Aufträge kollidieren, werden mit dem allgemeinen Statuscode 12122 zurückgewiesen und die Transaktion wird mit CANCEL zurückgesetzt.

SPECIAL

Neue UPDATE-Transaktionen oder -Verarbeitungsketten, die mit der Ausführung vorliegender CHECKPOINT- oder NEW RLOG-Aufträge kollidieren, werden mit dem spezifischen Statuscode 12124 zurückgewiesen und die Transaktion wird mit CANCEL zurückgesetzt.

Mit dem Ladeparameter PP ORDER-DBSTATUS können Sie das Systemverhalten für den Fall steuern, dass der DBH durch ein PERFORM-Kommando oder durch einen systeminternen PERFORM (z. B. wegen einer Fehlersituation) veranlasst wurde, vorgemerkte Aufträge (CHECKPOINT, NEW RLOG) auszuführen, und deshalb neue UPDATE-Transaktionen oder -Verarbeitungsketten zurückweist. Bei ORDER-DBSTATUS=SPECIAL ist es damit möglich, im Anwendungsprogramm gezielt auf die spezielle Wartesituation zu reagieren.

3.30 Verwendete Ladeparameter auflisten (PARLIST)

```
PP PARLIST={YES | NO}
```

Standardwert:

NO

YES Ladeparameter auflisten

NO Ladeparameter nicht auflisten

Mit dem Ladeparameter PP PARLIST können Sie beim DBH-Start und Session-Wiederanlauf die aktuellen Ladeparameter auf der Datensichtstation des Datenbankadministrators bzw. auf SYSOUT protokollieren lassen.

3.31 Kennwort festlegen für Dateien, die der DBH verwendet (PASSWORD)

```
PP PASSWORD={NONE | STANDARD | kennwort(kennwort,kennwort,...)}
```

Standardwert:

STANDARD

NONE

UDS/SQL benötigt nur das Kennwort von PP CATPASS.

STANDARD

UDS/SQL benötigt nur das Kennwort von PP CATPASS und das feste Kennwort C'UDS'BLANK'.

kennwort

(*kennwort,kennwort,..*)

UDS/SQL benötigt zusätzlich zu PP CATPASS weitere Kennwörter. Der Datenbankadministrator übergibt mit *kennwort* das für die Session benötigte Kennwort bzw. die benötigten Kennwörter. Es können max. 100 Kennwörter auf mehrere PP PASSWORD-Angaben verteilt, angegeben werden.

kennwort darf bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C'xxxx':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X'nnnnnnnnr':

nnnnnnnn sind ein bis acht Sedezimalziffern

d:

d'ist eine Dezimalzahl mit höchstens acht Ziffern und Vorzeichen, deren Wert zu einem Binärwert umgesetzt wird

Es gilt die BS2000-Syntax (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#), ADD-PASSWORD).

Mit PP PASSWORD geben Sie das Kennwort bzw. die Kennwörter für die Dateien an, die bereits existieren wenn sie vom DBH eröffnet werden sollen, also z.B. Benutzerrealms und DBDIR sowie ggf. vorab eingerichtete ALOG-Dateien, die DB-Status-Datei und Temporäre Realms (siehe [Abschnitt „Kennwörter für UDS/SQL-Dateien vergeben“](#)). Für den Kennwortschutz der Datei, die die DBH-Ladeparameter enthält, ist der Datenbankadministrator verantwortlich. Die Menge der Kennwörter kann dynamisch mit ADD PW (siehe "[Datenbanken, Realms und Kennwörter anschließen \(ADD\)](#)") und DROP PW (siehe "[Datenbanken, Realms und Kennwörter ausschließen \(DROP\)](#)") verändert werden.

PP PASSWORD darf mehrfach angegeben werden.

Das Kennwort C'UDS'BLANK' und über PP CATPASS, PP PASSWORD bzw. das DAL-Kommando ADD PW angegebene Lese-Kennwörter sollten nicht für andere Dateien verwendet werden, da UDS/SQL diese Kennwörter intern absetzt und somit in Sondersituationen ein Zugriff auf Dateien möglich ist, für die dies aus Anwendersicht nicht erwünscht ist.

Reaktionen in Fehlersituationen

Da die DBH-Ladeparameter in der Regel in einer katalogisierten Datei hinterlegt sind, können Sie fehlerhafte Angaben zu PP PASSWORD während des UDS/SQL-Starts nicht durch Wiederholung korrigieren. Die Reaktion bei folgenden Fehleingaben ist immer Abbruch nach PP END:

- PP PASSWORD enthält einen Syntaxfehler
- PP PASSWORD=NONE/STANDARD mehrfach angegeben
- PP PASSWORD beim Einlesen abgeschnitten
- mit PP PASSWORD wurden mehr als 100 Kennwörter angegeben

3.32 Behandlung der Privacy-Prüfung steuern (PRIVACY-CHECK)

`PP PRIVACY-CHECK={STD | NO-KSET | OFF}`

Standardwert:

STD

STD Die Privacy-Prüfung wird durchgeführt.

Bei der Privacy-Prüfung wird der Benutzergruppenname so vorausgesetzt, wie im Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt BPRIVACY, Rechteprüfung, beschrieben.

NO-KSET

Die Privacy-Prüfung wird durchgeführt.

Bei der Privacy-Prüfung bei openUTM wird im Benutzergruppennamen kein KSET-Name geprüft.

OFF Es wird keine Privacy-Prüfung durchgeführt.

3.33 Transaktionen im Zustand PTC bei Fehlern behandeln (PTCSYNCH)

```
PP PTCSYNCH=([WAIT | ABORT | COMMIT][, [{WAIT | ABORT | COMMIT}]])
```

Standardwert:

(WAIT, WAIT)

Für UDS-D/openUTM

Der **erste** Wert des Parameters PTCSYNCH wirkt auf sekundäre Teiltransaktionen bzw. openUTM-Transaktionen, die sich bei **Warmstart** im Zustand PTC befinden.

Für UDS-D

Der **zweite** Wert des Parameters PTCSYNCH wirkt auf sekundäre Teiltransaktionen, die sich während der **laufenden Session** im Zustand PTC befinden und für die der Zustand der zugehörigen primären Teiltransaktion nicht ermittelbar ist.

WAIT Die sekundären Teiltransaktionen bzw. openUTM-Transaktionen warten, bis sie beendet werden entweder durch eine Nachricht von der zugehörigen primären Teiltransaktion bzw. von openUTM oder explizit durch den Datenbankadministrator mit den DAL-Kommandos ABORT, OPTION=PTC oder COMMIT.

ABORT

UDS/SQL setzt die sekundären Teiltransaktionen bzw. openUTM-Transaktionen zurück und gibt eine Warnung aus.

COMMIT

UDS/SQL beendet die sekundären Teiltransaktionen bzw. openUTM-Transaktionen mit Festschreiben der Änderungen und gibt dem Datenbankadministrator eine Warnung aus.

Mit PP PTCSYNCH können Sie steuern, ob sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC nach einer von PP CHCKTIME abhängigen Zeitspanne mit FINISH oder mit FINISH WITH CANCEL beendet werden sollen.

Die Werte des DBH-Ladeparameters PP PTCSYNCH kann der Datenbankadministrator während der Session mit dem DAL-Kommando MODIFY PTCSYNCH dynamisch verändern.

i Wenn der DBH-Ladeparameter PP PTCSYNCH nicht auf (WAIT, WAIT) steht, ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

3.34 Ersatzdatenträger für die RLOG-Dateien festlegen (RESERVE)

```
PP RESERVE=
  {NONE |
   :catid: |.
  PUBLIC
   (priv-vsn-1/gerät-1[ ,priv-vsn-2/gerät-2[ ,priv-vsn-3/gerät-3] ] )
   (vsn-1[ ,vsn-2[ ,vsn-3] ] ) }
```

Standardwert:

NONE

Es werden keine Ersatzdatenträger festgelegt.

:catid: BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Gemeinschaftliche Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung.

priv-vsn

Die RLOG-Datei wird bei Ausfall des Original-Datenträgers auf die Privatplatten angelegt, die als Ersatzdatenträger angegeben worden sind. Es können maximal drei Archivnummern angegeben werden, die verschieden sein müssen. Eine Archivnummer darf bis zu sechs alphanumerische Zeichen enthalten.

gerät Gerätetyp der privaten Platte. Der Gerätetyp darf bis zu acht alphanumerische Zeichen enthalten.

vsn Als Ersatzdatenträger für die RLOG-Datei werden Platten festgelegt, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind. Es können maximal drei VSNs angegeben werden, die verschieden sein müssen.
Die VSN kann in PUB-Notation (PUBp_{xx}) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden. Bitte beachten Sie auch die „Zusatzbedingungen für VSN-Angaben“ im [Abschnitt „RLOG-Datei führen \(LOG\)“](#).

Mit dem Ladeparameter PP RESERVE vergeben Sie Ersatzdatenträger für die Datenträger, die Sie mit den Ladeparametern PP LOG und PP LOG-2 festgelegt haben, oder legen keine Ersatzdatenträger fest.

Wenn beim Einrichten einer RLOG-Datei ein Datenträger ausfällt, den Sie mit der Erstzuweisung PP LOG oder PP LOG-2 festgelegt haben, so wird stattdessen der Ersatzdatenträger von PP RESERVE verwendet. Als Ersatzdatenträger müssen Sie andere

Datenträger bestimmen, als die, die mit der Erstzuweisung festgelegt wurden. Dies wird nicht beim Einlesen der DBH-Ladeparameter geprüft, sondern erst dann, wenn der Reservedatenträger tatsächlich benötigt wird.

Wenn auch die Datenträger der Ersatzzuweisung PP RESERVE ausfallen, so wird über die gesamte Konfiguration eine Änderungssperre verhängt. Die Änderungssperre besteht bis der Ausfall der Datenträger behoben oder ein anderer Datenträger zugewiesen ist, und mit den DAL-Kommandos NEW LOG und PERFORM die RLOG-Datei zugeschaltet ist.

Das Ausweichen auf die Ersatzdatenträger bleibt auf den Fehlerfall beschränkt, bei der nächsten Gelegenheit versucht der DBH wieder, die betreffenden Dateien entsprechend der Erstzuweisung einzurichten.

Die Ersatzdatei-Zuweisung ist pro Konfiguration einmal vorhanden.

Während der Initialisierung des DBH wird die Verträglichkeit der PP RESERVE-Angaben mit einer bereits geprüften UDS/SQL-Pubset-Deklaration kontrolliert.

Im Fehlerfall erfolgt nach Ausgabe des betroffenen Ladeparameters die Meldung:

```
UDS0711 UDS LADE-PARAMETER ZURUECKGEWIESEN: CATID NOT WITHIN SCOPE OF UDS  
PUBSET DECLARATION (... ,tsn4)
```

Das Hochfahren des DBH wird dann nach Abschluss der Ladeparameter-Prüfung abgebrochen.

3.35 Auftragsantworten an die Anwendertasks bündeln (RESULT-DELAY)

Nur **independent DBH!**

```
PP RESULT-DELAY=n
```

Standardwert:

0

n Anzahl gebündelte Auftragsantworten an die Anwendertasks.

$n = 1..m$

$m = \text{PP TRANSACTION}$

Mit dem Ladeparameter PP RESULT-DELAY wird eine Durchsatzsteigerung in Hochlastsituationen dadurch ermöglicht, dass die Auftragsantworten an die Anwendertasks gebündelt werden. Siehe auch [Abschnitt „Ladeparameter, die die Prozessornutzung beeinflussen“](#).

Diesen Parameter sollten Sie nur unter folgenden Voraussetzungen angeben:

- Es wird eine Mehrprozessor-Anlage mit mehr als drei Prozessoren genutzt.
- Die Anzahl generierter Servertasks entspricht der Anzahl Prozessoren, die durch die Servertasks ausgelastet werden können. Hierfür ist in der Regel eine hohe Priorität der Servertasks notwendig.

Wenn gleichzeitig PP SCHEDULING=ASYMMETRIC eingestellt ist, wird die tatsächliche Bündelung der Auftragsantworten jedoch eingeschränkt.

i Wenn Sie einen größeren Wert angeben als die Angabe bei PP TRANSACTION, so wird ohne Warnung auf den Wert von PP TRANSACTION geändert.

3.36 Schedulingverhalten in Hochlastsituationen steuern (SCHEDULING)

Nur **independent DBH!**

```
PP SCHEDULING={SYMMETRIC | ASYMMETRIC}
```

Standardwert:

SYMMETRIC

SYMMETRIC

Alle Servertasks sind gleichberechtigt bzgl. der Abarbeitung von anstehenden DML-Aufträgen. Diese Einstellung ist günstig bei gut ausgelasteten Servertasks oder wenn den Servertasks keine hohe BS2000-Priorität gegeben werden kann.

ASYMMETRIC

Es wird versucht, die anstehenden DML-Aufträge so auf die Servertasks zu verteilen, dass im Hochlastfall möglichst $n-1$ Servertasks zu 100% ausgelastet sind. Dadurch wird eine verbesserte Prozessornutzung, ein erhöhter Durchsatz und ein besserer Multiprozessorfaktor erreicht.

Diese Einstellung ist nur für Mehrprozessoranlagen und bei Konfigurationen mit mehr als einer Servertask sinnvoll. Außerdem müssen sinnvolle BS2000-Taskprioritäten für die Servertasks eingestellt werden, siehe Kapitel „[Leistungsoptimierung](#)“.

Durch die asymmetrische Verarbeitung kann es für einzelne Transaktionen bei Überlastsituationen zu Antwortzeitverlängerungen kommen. Diese werden aber in der Regel durch den erhöhten Gesamtdurchsatz und den damit verbundenen schnelleren Abbau der Überlastsituation kompensiert.

Mit dem Ladeparameter PP SCHEDULING können Sie das interne Verhalten mehrerer Servertasks bei Lastschwankungen im Hochlastbereich beeinflussen. Siehe auch [Abschnitt „Ladeparameter, die die Prozessornutzung beeinflussen“](#).

i Bei der Angabe PP SERVERTASK=1 oder PP IO=SYNC hat der Ladeparameter PP SCHEDULING keine Wirkung.

3.37 Anzahl Servertasks festlegen (SERVERTASK)

Nur **independent DBH!**

```
PP SERVERTASK=n
```

Standardwert:

1

n Anzahl der zu startenden Servertasks

n = 1..30

Beim Starten des DBH entnimmt die Mastertask des DBH dem Ladeparameter PP SERVERTASK, wie viele Servertasks sie starten soll (siehe [Abschnitt „Arbeitsweise desindependent DBH“](#)).

Wenn die I/Os asynchron ausgeführt werden (siehe Ladeparameter PP IO "[I/Os asynchron oder synchron ausführen \(IO\)](#)") genügt es, auf einem Monoprozessor eine Servertask zu starten. Bei Multiprozessoren genügt es, pro Prozessor maximal eine Servertask zu starten. Die Priorität der Servertasks muss dann entsprechend hoch sein (siehe „RUN-PRIO“ im Kapitel "[DBH-Startkommandos](#)").

Siehe auch [Kapitel „Leistungsoptimierung“](#).



Bei der Angabe PP SERVERTASK=1 hat der Ladeparameter PP SCHEDULING keine Wirkung.

3.38 Größe des SSITAB-Pools festlegen (SIP-SIZE)

Nur **independent DBH!**

```
PP SIP-SIZE=n
```

Standardwert:

1024 Kbyte bei Angabe von PP MPSEG=STD

128 Kbyte bei Angabe von PP MPSEG=64K

n Speichergröße des SSITAB-Pools in Kbyte

$n = 1..16384$

n wird bei Angabe von PP MPSEG=STD auf ein ganzzahliges Vielfaches von 1024 aufgerundet, bei Angabe von PP MPSEG=64K auf ein ganzzahliges Vielfaches von 64. Bei PP MPSEG=64K sind Werte über 8192 nicht sinnvoll.

Bei $n > 16384$ verwendet der DBH 16384 Kbyte.

Für das Bestimmen der Poolgröße sollten Sie folgende Größen beachten:

- der Benutzeradressraum
- die Summe der Längen aller oder parallel zu verarbeitende SSITAB-Module.

Um ein häufiges Entladen und Laden der SSITAB-Module im Pool zu vermeiden, ist es ratsam, den Pool so groß zu dimensionieren, dass alle verwendeten SSITAB-Module Platz finden.

Der SSITAB-Pool wird bei PP MPSEG=STD im oberen Adressraum der UDS/SQL-Tasks (> 16 MB) angelegt. In Anwendertasks wird der SSITAB-Pool abhängig vom AMODE (24/31) im unteren oder oberen Adressraum angelegt. Bei PP MPSEG=64K liegt der SSIT-AB-Pool in allen Tasks im unteren Adressraum. Falls in der Session auch Anwendungen im AMODE 24 genutzt werden, darf der SSITAB-Pool nur so groß gewählt werden, dass er noch in den unteren Adressraum dieser Anwendung passt.

Mit dem Ladeparameter PP PARLIST oder dem DAL-Kommando DISPLAY PP wird bei SIP-SIZE ausgegeben, in welcher Größe der SSITAB-Pool angelegt wurde.

3.39 Anzahl gleichzeitig aktiver SQL-Vorgänge festlegen (SQL)

Nur **independent DBH!**

```
PP SQL=n
```

Standardwert:

4 (Standardwert von PP TRANSACTION)

n maximale Anzahl der gleichzeitig aktiven SQL-Vorgänge

n = 0..9999

Wenn ein neuer SQL-Vorgang gestartet werden soll und dabei der Wert *n* überschritten wird, versucht der DBH einen Vorgang, der in der Zeitspanne von PP SQL-LIMIT inaktiv war, zu beenden. Gelingt das nicht, wird der neue SQL-Vorgang mit einer Fehlermeldung abgewiesen (SQL-Code: -1710).

Bei der Größe von PP SQL müssen Sie berücksichtigen, dass die SQL-Vorgänge während ihrer ganzen Lebensdauer interne Betriebsmittel (z.B. Speicherplatz) belegen. Zu viele parallele SQL-Vorgänge können daher zu Engpässen führen.

PP SQL darf auch größer als PP TRANSACTION sein, das ist in einer openUTM-Umgebung sinnvoll. PP TRANSACTION legt die maximale Anzahl der gleichzeitig aktiven Transaktionen und die maximale Anzahl der anschließbaren Anwendertasks fest. Eine Task, die nicht mit openUTM abläuft, kann nicht mehr als einen Vorgang eröffnen.

Mit PP SQL=0 können Sie die Verwendung von SQL verhindern, auch wenn der DBH mit SQL-Anschluss geladen wurde.

Siehe auch [Kapitel „Der SQL-Vorgang“](#).

3.40 Zeitspanne für nicht aktive SQL-Vorgänge einstellen (SQL-LIMIT)

Nur **independent DBH!**

```
PP SQL-LIMIT=n
```

Standardwert:

10

n Zeitspanne in Minuten, für die UDS/SQL Daten inaktiver Vorgänge mindestens aufbewahrt

n = 5..999 Minuten

Wenn ein neuer SQL-Vorgang gestartet werden soll und dabei der Wert von PP SQL überschritten wird, versucht UDS/SQL Vorgänge zu beenden, die bereits länger als die Zeitspanne von SQL-LIMIT inaktiv sind.

Siehe auch [Kapitel „Der SQL-Vorgang“](#).

3.41 Konsistenzpunkte in ALOG-Dateien festschreiben (STDCKPT)

```
PP STDCKPT={YES | NO}
```

Standardwert:

NO

YES Der DBH schreibt standardmäßig bei DBH-Start, DBH-Ende und Session-Wiederanlauf konfigurationsweit Konsistenzpunkte in den ALOG-Dateien fest.

NO Konsistenzpunkte bei den oben genannten Ereignissen werden nicht festgeschrieben.

Durch den Ladeparameter PP STDCKPT schreibt der DBH für alle Datenbanken der Konfiguration Konsistenzpunkte in ALOG-Dateien fest.

Konsistenzpunkte schreibt der DBH fest

- beim DBH-Start
- beim normalen DBH-Ende (CLOSE CALLS/CLOSE RUN-UNITS)
- nach erfolgreichem Session-Wiederanlauf

Einen Konsistenzpunkt in der ALOG-Datei festschreiben bedeutet:

- die bis zu diesem Zeitpunkt geschriebene ALOG-Datei wird abgeschlossen
- eine UDS/SQL-Meldung gibt Ihnen das Log-Intervall-Ende (Zeitstempel) der abgeschlossenen ALOG-Datei aus
- eine neue ALOG-Datei wird eröffnet.

Durch das Festschreiben der Konsistenzpunkte werden ALOG-Dateien abgeschlossen. Die ALOG-Dateien werden dadurch frei z.B. für das Nachfahren von Schattendatenbanken (mit dem Dienstprogramm BMEND). Die abgeschlossenen ALOG-Dateien können auch auf Band gesichert und anschließend auf der Platte gelöscht werden.

Konsistenzpunkte werden gleichzeitig für die gesamte DB-Konfiguration festgeschrieben. Durch das dynamische Anschließen und Ausschließen von Datenbanken mit den DAL-Kommandos ADD und DROP kann sich die Konfiguration gegenüber der Konfiguration beim DBH-Start ändern.

Bei linked-in DBH ist der Parameter PP STDCKPT die einzige Möglichkeit für den Datenbankadministrator, das Festschreiben von Konsistenzpunkten in ALOG-Dateien anzufordern.

ALOG-Datei-Wechsel

Wenn eine Datenbank auf der Konfigurationskennung liegt, richtet der DBH eine neue ALOG-Datei ein, wenn die aktuelle ALOG-Datei verbraucht ist.

Wenn die Datenbank in einer anderen Kennung liegt, muss in der anderen Kennung eine folgende ALOG-Datei vorhanden sein, die auch sharable sein muss.

Ist keine folgende ALOG-Datei vorhanden, wird

- die alte ALOG-Datei konsistent geschlossen
- im DBDIR die ALOG-Datei-Folgenummer um 1 erhöht und ein Kennzeichen gesetzt, dass die durch die ALOG-Datei-Folgenummer identifizierte ALOG-Datei noch nicht betriebsbereit ist
- die Datenbank solange für Änderungen gesperrt, bis die durch die ALOG-Datei-Folgenummer identifizierte ALOG-Datei betriebsbereit ist
- dies in einer Meldung bekannt gegeben.

Der Datenbankadministrator kann nun in der fremden Kennung eine ALOG-Datei mit dem CREATE-FILE-Kommando einrichten. Mit dem DAL-Kommando CHECKPOINT kann er den DBH dazu veranlassen, die nun vorhandene ALOG-Datei zu initialisieren und das Festschreiben des Konsistenzpunkts damit zu vervollständigen. Die Sperre für Änderungen wird aufgehoben, falls die ALOG-Datei betriebsbereit gemacht werden kann.

ALOG-Datei-Überlauf

Sollte der Platz auf der ALOG-Datei nicht ausreichen, um die ALOG-Datei konsistent zu schließen und zu wechseln, kommt es zu einem ALOG-Datei-Überlauf. Diesen gibt es in zwei Varianten:

- **der weiche ALOG-Datei-Überlauf (FILE OVERFLOW)**

Die Meldung FILE OVERFLOW zeigt an, dass die ALOG-Datei überzulaufen droht, es ist jedoch noch Platz vorhanden um einen Konsistenzpunkt zu erreichen.

Die Datenbank wird für ändernde Transaktionen gesperrt. Offene Transaktionen werden entweder zurückgesetzt oder fortgesetzt (bis COMMIT), je nachdem, wie viel Platz noch vorhanden ist auf der ALOG-Datei.

Wenn ein Konsistenzpunkt erreicht worden ist, wird die ALOG-Datei gewechselt.

- **der harte ALOG-Datei-Überlauf (ALOG OVERFLOW)**

Die Meldung ALOG OVERFLOW zeigt an, dass die ALOG-Datei voll und kein Platz vorhanden ist, um einen Konsistenzpunkt zu erreichen. Das AFIM-Logging ist nicht mehr möglich und es würde eine Logging-Lücke entstehen.

Dieser Fall kann vor allem bei langen, offenen Transaktionen auftreten.

Die Datenbank wird für alle Transaktionen gesperrt. Der DBH schaltet die Datenbank inkonsistent ab.

Die Ursache des Datei-Überlaufs können Sie dem DMS-Errorcode entnehmen und die entsprechenden Maßnahmen treffen.

Nachdem Sie Platz gemacht haben für die Erweiterung der ALOG-Datei, können Sie die Datenbank mit Warmstart wieder zuschalten.

Beim Warmstart kann es wiederum zu einem ALOG OVERFLOW kommen. Das Verfahren können Sie dann beliebig oft wiederholen, bis es gelingt, einen Konsistenzpunkt zu erreichen.

3.42 Anzahl Subschemata festlegen (SUBSCHEMA)

```
PP SUBSCHEMA=n
```

Standardwert:

1

n Anzahl der gleichzeitig verwendbaren Subschemata pro Datenbank

n = 1..100

Richtwert für *n*:

Anzahl der **häufig** angesprochenen Subschemata und zusätzlich Reserve für **sporadisch** angesprochene Subschemata.

Zum Ablaufen einer Verarbeitungskette muss die Subschema Information Area (SSIA) des angesprochenen Subschemas im Hauptspeicher stehen. Der Ladeparameter SUBSCHEMA legt fest, mit wie vielen Subschemata pro Datenbank während der Session maximal gleichzeitig gearbeitet werden kann.

Ein zu niedrig gewähltes *n* kann zu Performance-Verlusten führen, weil der DBH dann häufig gezwungen ist:

- bereits geladene SSIA's wieder zu verdrängen, um neue SSIA's einlesen zu können
- bzw. sogar Transaktionen (mit BIB-Statuscode 132) mit CANCEL zu beenden, weil die benötigten SSIA's nicht geladen sind und (aus Platzmangel) auch nicht eingelesen werden können.

3.43 Anzahl KDBS-Dateikennungen festlegen (SUBTRANSACTION)

```
PP SUBTRANSACTION=n
```

Standardwert:

0

n Anzahl der KDBS-Dateikennungen, die alle KDBS-Anwender maximal auf einer Datenbank gleichzeitig eröffnet haben können.

n = 1..254

Den Ladeparameter PP SUBTRANSACTION benötigen Sie **nur beim Einsatz von KDBS**.

3.44 Benutzungsarten für Transaktionen festlegen (TA-ACCESS)

Nur **independent DBH!**

```
PP TA-ACCESS={STD | SHARED}
```

Standardwert:

STD

STD Beim Eröffnen einer Verarbeitungskette werden alle Benutzungsarten zugelassen. Abhängig von der eingestellten Benutzungsart, werden die zu eröffnenden Realms gegen Lesen bzw. Ändern für andere Transaktionen gesperrt.

SHARED

Beim Eröffnen einer Verarbeitungskette werden nur die Benutzungsarten SHARED RETRIEVAL und SHARED UPDATE zugelassen. Die zu eröffnenden Realms werden nicht gesperrt.

Der Versuch, eine Verarbeitungskette in den Benutzungsarten PROTECTED oder EXCLUSIVE zu eröffnen, wird mit dem Statuscode „092“ abgewiesen.

PP TA-ACCESS=SHARED sollten Sie angeben, wenn alle Transaktionen in der Benutzungsart SHARED eröffnet werden. Dadurch wird beim READY Pfadlänge gespart.

Der Ladeparameter PP TA-ACCESS bezieht sich auf alle Datenbanken, die an der Session teilnehmen.

3.45 Maximale Anzahl Transaktionen festlegen (TRANSACTION)

```
PP TRANSACTION={n | ([n][,m])}
```

Standardwert:

$n: 4 / m: 1$

n Maximale Anzahl der gleichzeitig offenen Transaktionen, zugleich maximale Anzahl der Anwendertasks (einschließlich openUTM-Tasks), die gleichzeitig mit dem DBH verbunden sein können.

$n = 1..225$

Für UDS-D

m maximale Anzahl sekundärer Teiltransaktionen, die der DBH gleichzeitig bearbeiten kann.

$m = 1..n$ und $m \leq n$

Der Wert für sekundäre Teiltransaktionen wirkt nur, wenn der UDS-D-Betrieb gestartet ist.

PP TRANSACTION bestimmt

- die Anzahl der Transaktionskanäle (Mainrefs) im DBH und damit die maximale Anzahl von Transaktionen, die im DBH gleichzeitig abgewickelt werden können,
- die Anzahl der Task-Verwaltungs-Einträge des DBH und damit die maximale Anzahl von Anwendertasks, die dem DBH gleichzeitig bekannt sein dürfen,
- bei UDS-D die maximale Anzahl der sekundären Teiltransaktionen, die dieser DBH für entfernte Anwenderprogramme gleichzeitig bearbeiten kann.

Eine **Teilnehmer**-Anwendertask ist dem DBH bekannt

- ab dem Start eines UDS/SQL-Programms (bei COBOL-DML und SQL) bzw. vom ersten READYC-Aufruf (bei CALL-DML)
- bis zum Ende des UDS/SQL-Programms

Im **UDS/SQL-openUTM-Betrieb** ist dem DBH jede openUTM-Task

- vom Start bis zum Ende der Tasks bekannt.

Beim linked-in DBH ist der Wert des Ladeparameters PP TRANSACTION auf 1 fest eingestellt.

i Die DBH-Ladeparameter PP DISDB, PP MAXDB und PP TRANSACTION wirken sich auf die Größe des Communication Pools und des Distribution Pools aus. Es ist nicht sinnvoll, bei allen diesen drei DBH-Ladeparametern den Maximalwert zu wählen, da dadurch die Memory Pools sehr groß werden und es abhängig von der Adressraumsituation unterhalb 16 Mbyte zu einem Speicherengpass kommen kann.

Es empfiehlt sich, den Wert für die maximale Anzahl sekundärer Teiltransaktionen geringfügig höher zu wählen als die voraussichtlich benötigte Anzahl sekundärer Teiltransaktionen. Wird nämlich zum Beispiel die primäre Teiltransaktion asynchron zurückgesetzt (z.B. wegen Deadlocks oder durch den Administrator), so werden die Konfigurationen der sekundären Teiltransaktionen darüber nicht sofort informiert, sondern erst im Rahmen der Transaktionsüberwachung (siehe [Abschnitt „Zeitgesteuertes Überwachen von Verbindungen und Transaktionen“](#)). Die Ressourcen der sekundären Teiltransaktionen bleiben in einem solchen Fall noch über den Zeitpunkt des Rücksetzens der primären Teiltransaktion hinaus belegt, sodass sie nach dem Rücksetzen der primären Teiltransaktion bis zum Anlaufen der Transaktionsüberwachung nicht von neuen sekundären Teiltransaktionen verwendet werden können. Kann eine sekundäre Teiltransaktion wegen eines Ressourcenengpasses nicht eröffnet werden, so erhält das Anwenderprogramm den Statuscode 122.

3.46 Bedienplatz zur Protokollierung der DCAM-Administration und Form der UDS/SQL-Meldungen (UCON)

Nur **independent DBH!**

```
PP UCON=C' { (mn) | <x | nnnn} ' [ , {MSG | UDS} ]
```

Standardwert:

C'<U',MSG

Die Protokollierung wird auf dem lokalen Bedienplatz ausgegeben.

Wenn die Protokollierung auf dem anderweitig eingestellten Ausgabeort nicht möglich ist, so wird der Standardwert wieder übernommen.

Ausgabeort:

mn Zwei Zeichen langer mnemotechnischer Geräte name.

<*x* *x*= Ein Zeichen für Berechtigungsschlüssel.
Das Zeichen '<' ist anzugeben.

nnnn Vier Zeichen langer Name des berechtigten Benutzerprogrammes mit Operatorfunktionen.

Form:

MSG Die Meldungen werden ohne Vorspann protokolliert. (Standardwert)

Dieses Format ist sinnvoll, wenn Meldungen verschiedener Systeme von einem Bedienplatz einheitlich ausgewertet werden sollen (z.B. über OMNIS-PROP).

UDS Die Meldungen werden mit dem formatierten, UDS/SQL-spezifischen Vorspann protokolliert. Der Vorspann wird den über DCAM administrierten Meldungen vorangestellt, wenn über DCAM nicht direkt von einer Datensichtstation aus administriert wird. Zum Aufbau und Inhalt des Vorspanns, siehe Handbuch „Meldungen“.

Mit dem Ladeparameter PP UCON legen Sie den Bedienplatz (UCON) fest, auf dem die DCAM-Administration protokolliert werden soll, sowie die Form der UDS/SQL-Meldungen. (Siehe „PP ADM“ im Kapitel „Administrierungsverfahren festlegen (ADM)“, Abschnitt „UDS/SQL über DCAM administrieren“ und Handbuch „Einführung in die Systembetreuung“).



Meldungen an die lokale Datensichtstation (SYSOUT) werden grundsätzlich ohne Vorspann ausgegeben.

3.47 Warte-Modus der Servertasks einstellen (WAIT)

Nur **independent DBH!**

```
PP WAIT={EVENT | BUSY}
```

Standardwert:

EVENT

EVENT

Es wird auf eine Ereignismeldung gewartet. Dadurch wird der Prozessor frei und kann für andere Aufgaben genutzt werden.

BUSY

Es wird aktiv auf das Eintreffen des nächsten Auftrags gewartet. Der Prozessor wird nicht freigegeben.

Mit dem Ladeparameter PP WAIT können Sie in UDS/SQL-Hochlastanwendungen steuern, wie die letzte noch aktive Servertask auf neu eintreffende Aufträge in kurzzeitigen Unterlastsituationen warten soll. Siehe auch [Abschnitt „Ladeparameter, die die Prozessornutzung beeinflussen“](#).

PP WAIT=BUSY sollten Sie nur unter den folgenden Voraussetzungen angeben:

- Es wird eine Mehrprozessor-Anlage mit mehr als drei Prozessoren genutzt.
- Die Last ist kontinuierlich über die gesamte Session so hoch, dass mindestens eine Servertask immer Aufträge zu bearbeiten hat.
- Unterlastsituationen sind selten.

3.48 Dauer eines Warmstarts beeinflussen (WARMSTART)

```
PP WARMSTART={STD | FAST | VERY-FAST}
```

Standardwert:

STD

Mit dem Ladeparameter WARMSTART wird im laufenden Betrieb des Independent DBH gesteuert, wann bereits abgeschlossene Transaktionen endgültig in die Datenbank eingebracht werden.

Bei Nutzung des Standardwertes werden die bereits abgeschlossenen Transaktionen zunächst ausfallsicher in die RLOG-Datei geschrieben. Das Schreiben in die Datenbank erfolgt erst, wenn die entsprechend geänderten Seiten aus den System Buffer Pools verdrängt werden. Hierdurch können unnötige Lese- und Schreibvorgänge im laufenden Betrieb eingespart werden. Bei einem eventuellen Warmstart muss dann aber das Nachfahren dieser bereits abgeschlossenen Transaktionen, die aber noch nicht endgültig in die Datenbank eingebracht wurden, nachgeholt werden.

Mit den Werten FAST und VERY-FAST führt der Independent DBH das Nachfahren der bereits abgeschlossenen Transaktionen auch dann durch, wenn die entsprechend geänderten Seiten noch nicht aus den System Buffer Pools verdrängt wurden. Dadurch erhöht sich im laufenden Betrieb die Anzahl der Schreibvorgänge in die Datenbank. Ein eventuell notwendiger Warmstart wird aber dafür in kürzerer Zeit abgeschlossen, da weniger abgeschlossene Transaktionen nachgefahren werden müssen. Als Kriterium für ein vorzeitiges Einbringen bereits abgeschlossener Transaktionen in die Datenbank wird der Teil der RLOG-Datei herangezogen, der beim Warmstart gelesen werden muss.

Bei einem Warmstart müssen aber nicht nur die Teile der RLOG-Datei gelesen werden, die bereits abgeschlossene Transaktionen enthalten, sondern auch solche Teile, in denen offene Transaktionen enthalten sind. Wenn durch lange Transaktionen der Teil, der die offenen Transaktionen enthält, groß wird, wird die Wirkung des Ladeparameters WARMSTART eingeschränkt.

STD Es findet kein vorzeitiges Einbringen bereits abgeschlossener Transaktionen statt. Die Dauer eines zukünftigen Warmstarts wird nicht verkürzt.

FAST Bereits abgeschlossene Transaktionen werden vor einer eventuellen Verdrängung aus den System Buffer Pools endgültig in die Datenbank eingebracht. Der Teil der RLOG-Datei, der bei einem Warmstart gelesen werden muss, wird gegenüber STD um bis zu 90 % reduziert. Die Dauer eines zukünftigen Warmstarts wird verkürzt, gegebenenfalls zu Lasten der Performance während des normalen Betriebs.

VERY-FAST

Bereits abgeschlossene Transaktionen werden vor einer eventuellen Verdrängung aus den System Buffer Pools endgültig in die Datenbank eingebracht. Der Teil der RLOG-Datei, der bei einem Warmstart gelesen werden muss, wird gegenüber STD um bis zu 99 % reduziert. Die Dauer eines zukünftigen Warmstarts wird dadurch verkürzt, gegebenenfalls zu Lasten der Performance während des normalen Betriebs.

4 Administration

Es gibt drei Schnittstellen für die Administration einer Session:

- Administration über UDSADM
(siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#))
- Administration über DCAM
(siehe [Abschnitt „UDS/SQL über DCAM administrieren“](#))
- Administration über INFORM-PROGRAM bzw. SEND-MESSAGE
(siehe [Abschnitt „UDS/SQL über INFORM-PROGRAM administrieren“](#))

Eine Beschreibung der Kommandos der Datenbankadministratorsprache DAL finden Sie ab "[Die Datenbankadministratorsprache DAL](#)".

4.1 UDS/SQL über UDSADM administrieren

Mit dem Programm UDSADM kann eine UDS/SQL-Konfiguration administriert werden, wenn beim DBH-Start der DBH-Ladeparameter PP ADM=REMOTE angegeben wird. Verbindungswünsche des Programms UDSADM werden abgelehnt, solange bereits ein Administrationsprogramm oder eine Datensichtstation über DCAM mit der Konfiguration verbunden ist.

UDSADM kommuniziert mit der Mastertask und schließt sich dazu an die von der Mastertask eröffnete DCAM-Applikation an.

Mit UDSADM können UDS/SQL-Meldungen von UDS/SQL empfangen und DAL-Kommandos an UDS/SQL gesendet werden.

UDSADM kann zu einem Zeitpunkt nur mit einer UDS/SQL-Konfiguration verbunden sein.

Alle Datenbankadministratoren sind gleichberechtigt. Zu einem Zeitpunkt ist allerdings für **eine** UDS/SQL-Konfiguration nur **ein** Datenbankadministrator erlaubt.

UDSADM ist sowohl im Stapelbetrieb als auch im Dialogbetrieb ablauffähig.

Wenn Sie UDSADM in eine Prozedur einschalen, können Sie mit dem Kommando ASSIGN-SYSLST TO= ... alle Aktionen protokollieren lassen.

UDSADM starten

UDSADM wird gestartet mit:

```
/START-UDS-ADM
```

Zusätzlich stehen Ihnen aus Kompatibilitätsgründen folgende Alias-Namen zur Verfügung:

```
/START-UDS-ADMINISTRATION  
oder  
/UDSADM
```

Die hier beschriebenen BS2000-Kommandos gehen davon aus, dass UDS/SQL über IMON installiert wurde. Eine genauere Beschreibung der Voraussetzungen finden Sie im Abschnitt „START-Kommandos der UDS/SQL-Programme“ des Handbuchs „[Aufbauen und Umstrukturieren](#)“.

4.1.1 UDSADM-Anweisungen

Die Anweisungsformate des Administrationsprogramm UDSADM entsprechen den Regeln von SDF (System Dialog Facility), siehe Handbuch „[Dialogschnittstelle SDF](#)“ und die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“. Großbuchstaben in **Halbfett** kennzeichnen garantierte Abkürzungen der Schlüsselwörter.

Übersicht der UDSADM-Anweisungen

Anweisung	Bedeutung
CONNECT-CONFIGURATION CONFIGURATION-NAME = <name> ,PASSWORD = *NONE / <x-string> <c-string> ,HOST = *LOCAL / <name>	Verbindung zur UDS/SQL-Konfiguration eröffnen
DISCONNECT-CONFIGURATION	Verbindung zur UDS/SQL-Konfiguration schließen
END	UDSADM beenden
EXECUTE-DAL-CMD CMD = <dal-cmd>	DAL-Kommandos ausführen
HELP-DAL-CMD ?	Hilfefunktion für DAL-Kommandos
MODIFY-MSG-FORMAT HEADER = NO / YES	Ausgabeformat für Nachrichten einstellen
MODIFY-MSG-WAIT-TIME TIME = STD / <integer 1..7200 <i>seconds</i> >	Wartezeit für DAL-Kommandos einstellen
SET-RECEIVE-MODE	Dauerempfangsprozess aktivieren
SHOW-CONNECTION-ATTRIBUTES	Verbindungsinformation anzeigen
SHOW-OUTSTANDING-MSG	Anstehende Nachrichten ausgeben

Tabelle 11: UDSADM-Anweisungen

Falsch eingegebene UDSADM-Anweisungen werden mit einer spezifischen Meldung abgewiesen und können korrigiert werden. Jede richtig eingegebene UDSADM-Anweisung wird sofort ausgeführt.

Regeln zur Auswahl der Anweisungen

Anweisung	Regeln zur Auswahl
END HELP-DAL-CMD MODIFY-MSG-FORMAT MODIFY-MSG-WAIT-TIME	Diese Anweisungen sind jederzeit erlaubt.
DISCONNECT-CONFIGURATION	Diese Anweisung ist nur für eine zuvor mit CONNECT-CONFIGURATION eröffnete Verbindung erlaubt.
CONNECT-CONFIGURATION	Diese Anweisung ist für eine noch nicht eröffnete Verbindung erlaubt.
EXECUTE-DAL-CMD SET-RECEIVE-MODE SHOW-CONNECTION-ATTRIBUTES SHOW-OUTSTANDING-MSG	Diese Anweisungen sind nur zwischen CONNECT-CONFIGURATION und DISCONNECT-CONFIGURATION erlaubt.

Tabelle 12: Regeln zur Auswahl der Anweisungen

4.1.1.1 Verbindung zur UDS/SQL-Konfiguration aufbauen (CONNECT-CONFIGURATION)

Mit dieser Anweisung wird eine Verbindung von UDSADM zu einer UDS-Konfiguration via DCAM aufgebaut. Wenn bereits eine Verbindung zum Kommunikationspartner existiert, wird diese Anweisung abgewiesen und eine Meldung ausgegeben.

CONNECT-CONFIGURATION

CONFIGURATION-NAME = <name>

, **PASSWORD** = ***NONE** / <x-string> / <c-string>

, **HOST** = ***LOCAL** / <name>

CONFIGURATION-NAME = <name>

Name der DCAM-Applikation, an die sich UDSADM anschließen soll. Der Name lässt sich aus dem UDS/SQL-Konfigurationsnamen ableiten und entspricht den ersten acht Zeichen.

Beispiel

UDS/SQL-Konfigurationsname	CONFIGURATION-NAME
UDSADMTESTDB	UDSADMTE
UDSADM	UDSADM
UDSA	UDSA

PASSWORD = ***NONE**

Wenn für die UDS/SQL-Konfiguration kein Administrations-Kennwort vereinbart wurde, müssen Sie keine Angabe machen für PASSWORD.

PASSWORD = <x-string> / <c-string>

Administrations-Kennwort für die Administration via DCAM. Es kann bis zu vier byte lang sein und wird folgendermaßen dargestellt:

X' nnnnnnnr'

ein bis zu acht Sedezimalziffern

C' xxxx' ein bis zu vier alphanumerische Zeichen

Wenn für die UDS/SQL-Konfiguration ein Administrations-Kennwort vereinbart wurde, dann müssen Sie dieses hier angeben.

HOST = ***LOCAL**

Name des lokalen Verarbeitungsrechners. Wenn die zu administrierende UDS/SQL-Konfiguration auf dem lokalen Verarbeitungsrechner betrieben wird, müssen Sie keine Angabe für HOST machen.

HOST = <name>

Name des Verarbeitungsrechners, auf dem die zu administrierende UDS/SQL-Konfiguration betrieben wird. Er darf maximal acht Zeichen lang sein.

4.1.1.2 Verbindung zur UDS/SQL-Konfiguration abbauen (DISCONNECT-CONFIGURATION)

Mit dieser Anweisung wird eine Verbindung von UDSADM zu einer UDS/SQL-Konfiguration abgebaut.

Wenn keine Verbindung zu einem Kommunikationspartner existiert, wird diese Anweisung abgewiesen und eine UDSADM-Meldung ausgegeben.

Wenn die Verbindung existiert, wird die Verbindung zum Kommunikationspartner abgebaut und eine Bestätigung ausgegeben.

DISCONNECT-CONFIGURATION

Diese Anweisung hat keine Operanden.

4.1.1.3 UDSADM beenden (END)

Mit dieser Anweisung wird UDSADM beendet.

Eine noch bestehende Verbindung zum Kommunikationspartner wird abgebaut und eine Bestätigung ausgegeben.

END

Diese Anweisung hat keine Operanden.

4.1.1.4 DAL-Kommandos ausführen (EXECUTE-DAL-CMD)

Mit dieser Anweisung wird das für *dal-cmd* eingegebene UDS/SQL- oder UDS-D-DAL-Kommando an die UDS/SQL-Konfiguration gesendet. UDSADM wartet auf die Empfangsbestätigung von UDS/SQL und gibt eine entsprechende Meldung aus.

Bei DAL-Kommandos mit Antwort wartet UDSADM maximal die eingestellte Wartezeit auf das Eintreffen der DAL-Antwort.

Trifft innerhalb dieser Zeit die DAL-Antwort ein, wird diese komplett empfangen und ausgegeben. Anschließend verzweigt UDSADM wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

Während UDSADM auf die DAL-Antwort wartet, werden alle von der UDS/SQL-Konfiguration eintreffenden Nachrichten empfangen und ausgegeben.

Trifft innerhalb dieser Zeit keine DAL-Antwort ein, wird eine UDSADM-Meldung ausgegeben. UDSADM verzweigt wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

Die ausstehende DAL-Antwort kann dann zu einem späteren Zeitpunkt mit einer der Empfangsfunktionen von UDSADM empfangen werden.

EXECUTE-DAL-CMD

CMD = <dal-cmd>

CMD = <dal-cmd>

dal-cmd kann bis zu 64 Zeichen lang sein. Die Syntax der DAL-Kommandos wird nicht von UDSADM geprüft.

i

- Bei zu kurz eingestellter Wartezeit (< 5 Sekunden) kann es vorkommen, dass der DCAM-Nachrichten-Puffer zwischen der UDS/SQL-Konfiguration und UDSADM kurzzeitig nicht ausreicht. UDSADM meldet nach Ablauf der Wartezeit, dass keine Meldungen mehr vorhanden sind. Mit der UDSADM-Anweisung SET-RECEIVE-MODE kann der Rest des DAL-Ergebnisses geholt werden. Wenn mehr als 30 Sekunden vergangen sind, sind möglicherweise DAL-Meldungen verloren gegangen. In diesem Fall sollte das DAL-Kommando mit einer längeren Wartezeit wiederholt werden.
- Sofern UDSADM in einer Prozedur aufgerufen wird, müssen Sie darauf achten, dass die DAL-Kommandos für UDS-D nicht mit symbolischen Prozedurnamen verwechselt werden können (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, Parameter ESCAPE-CHARACTER im Kommando BEGIN-PROCEDURE). Dies können Sie erreichen durch Verdopplung des Zeichens „&“ im UDS-D-DAL-Kommando.
- Bitte beachten Sie, dass UDSADM einen abschliessenden Bindestrich als Fortsetzungszeichen interpretiert. Dies kann zu Schwierigkeiten führen, wenn die Eingabe von DAL-Kommandos mit einem im DAL-Kommando enthaltenen Bindestrich endet (siehe „DAL-Syntax“ im [Abschnitt „Die Datenbankadministratorsprache DAL“](#)).

4.1.1.5 Hilfefunktion für DAL-Kommandos (HELP-DAL-CMD)

Mit dieser Anweisung kann die Bedeutung einzelner DAL-Kommandos abgefragt werden. Die Syntax der DAL-Kommandos wird mit Beispielen beschrieben.

Nach Eingabe des Kommandos wird eine Übersichtsliste der DAL-Kommandos ausgegeben. Durch Eingabe von Fragezeichen kann die Bedeutung einzelner DAL-Kommandos erfragt werden.

Die Ausführung von DAL-Kommandos ist nicht auf diesem Wege möglich.

HELP-DAL-CMD
?

Das Fragezeichen muss angegeben werden.

4.1.1.6 Ausgabeformat für Nachrichten einstellen (MODIFY-MSG-FORMAT)

Mit dieser Anweisung wird das Ausgabeformat für Nachrichten eingestellt. UDS/SQL-Meldungen, UDS-D-Meldungen und DAL-Antworten werden von UDSADM in dem eingestellten Ausgabeformat auf SYSOUT ausgegeben.

MODIFY-MSG-FORMAT

HEADER = <u>NO</u> / YES

HEADER = NO

Die Nachricht wird **ohne** Meldungsvorspann auf SYSOUT ausgegeben.

HEADER = YES

Die Nachricht wird **mit** Meldungsvorspann auf SYSOUT ausgegeben, siehe Handbuch „[Meldungen](#)“, Formatierte Meldungsausgabe.

4.1.1.7 Wartezeit für den Empfang von Nachrichten einstellen (MODIFY-MSG-WAIT-TIME)

Mit dieser Anweisung wird die Zeit in Sekunden eingestellt, die auf das Eintreffen einer Nachricht (DAL-Antwort, UDS/SQL- oder UDS-D-Meldung) gewartet werden soll.

Wenn innerhalb dieser Wartezeit die Nachricht eintrifft, wird sie komplett empfangen und ausgegeben.

Wenn innerhalb dieser Antwortzeit die Nachricht nicht eintrifft, verzweigt UDSADM wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

Die ausstehende Nachricht kann dann zu einem späteren Zeitpunkt mit einer der Empfangsfunktionen von UDSADM empfangen werden.

MODIFY-MSG-WAIT-TIME

TIME = <u>STD</u> / <integer 1..7200 seconds>
--

TIME = STD

Die Wartezeit wird auf den Standard-Wert eingestellt. Der Standard-Wert beträgt für Dialogbetrieb 12 Sekunden und für Stapelbetrieb 60 Sekunden.

TIME = <integer 1..7200 seconds >

Der Wert für die Wartezeit muss in den angegebenen Grenzen liegen. Es sind Werte von 1 bis 7200 Sekunden möglich.

i Wenn der Parameter TIME zu niedrig eingestellt wird, können Meldungen verloren gehen (siehe Hinweis auf "DAL-Kommandos ausführen (EXECUTE-DAL-CMD)").

4.1.1.8 Dauerempfangsprozess aktivieren (SET-RECEIVE-MODE)

Mit dieser Anweisung wird ein Empfangsprozess in UDSADM aktiviert. UDSADM ist nun ständig empfangsbereit und empfängt alle eintreffenden Nachrichten (DAL-Antworten, UDS/SQL-, UDS-D-Meldungen) und gibt sie aus. Wenn UDSADM sich in diesem Zustand befindet, können keine Eingaben gemacht werden.

Der Empfangsprozess kann mit der Taste [K2] unterbrochen werden. UDSADM verzweigt dann wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

SET-RECEIVE-MODE

Diese Anweisung hat keine Operanden.

- i** Normalerweise sollte UDSADM im Dauerempfangsprozess aktiv sein, damit keine Meldungen verloren gehen.
Wenn seitens UDSADM für längere Zeit keine Empfangsfunktion genutzt wurde, kann es sein, dass auf dem Transportweg über DCAM Meldungen verloren gehen.

4.1.1.9 Verbindungsinformationen ausgeben (SHOW-CONNECTION-ATTRIBUTES)

Mit dieser Anweisung werden die Verbindungsinformationen ausgegeben.

```
SHOW-CONNECTION-ATTRIBUTES
```

Diese Anweisung hat keine Operanden.

Die Verbindungsinformation hat folgendes Format:

UDSADM *own-appl*/IN HOST *own-name* CONNECTED TO:

CONFNAME	HOSTNAME	SECLEVEL	CON-DATE	CON-TIME
<i>konfname</i>	<i>rechner</i>	<i>n</i>	<i>jjjj-mm-dd</i>	<i>hh:mm:ss</i>

Die Verbindungsinformationen sind wie folgt zu interpretieren:

own-appl

DCAM-Anwendungsname von UDSADM im Format UAD@ *tsn*.

UAD@ ist ein fest definierter String

tsn ist die Prozessfolgennummer der UDSADM-Task.

own-name

Name des Verarbeitungsrechners, in dem UDSADM abläuft.

konfname

UDS/SQL-Konfigurationsname; dient als DCAM-Anwendungsname.

rechner

BCAM-Name des Verarbeitungsrechners, auf dem die zu administrierende UDS/SQL-Konfiguration betrieben wird.

SECLEVEL

Sicherheitsstufe der zu administrierenden UDS/SQL-Konfiguration.

Diese Spalte wird nur noch aus Kompatibilitätsgründen ausgegeben.

n=NO

normale Sicherheitsstufe

CON-DATE

Datum bei Verbindungsaufbau zur UDS/SQL-Konfiguration.

CON-TIME

Zeit bei Verbindungsaufbau zur UDS/SQL-Konfiguration.

4.1.1.10 Anstehende Nachrichten ausgeben (SHOW-OUTSTANDING-MSG)

Mit dieser Anweisung werden anstehende Nachrichten (DAL-Antworten, UDS/SQL-, UDS-D-Meldungen) ausgegeben. Stehen zur Zeit dieser Anweisung Nachrichten an, werden diese ausgegeben. Wenn alle Nachrichten ausgegeben sind, verzweigt UDSADM wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

Wenn zu dieser Zeit keine Nachrichten anstehen, wird eine UDSADM-Meldung ausgegeben. UDSADM verzweigt wieder in den Basiszustand und kann weitere UDSADM-Eingaben bearbeiten.

SHOW-OUTSTANDING-MSG

Diese Anweisung hat keine Operanden.

4.2 UDS/SQL über DCAM administrieren

Wenn beim DBH-Start der DBH-Ladeparameter PP ADM=REMOTE angegeben wird, eröffnet UDS/SQL eine DCAM-Anwendung. Der Datenbankadministrator kann dann von jeder Datensichtstation Verbindung mit UDS/SQL aufnehmen und Datenbankadministrator-Kommandos eingeben.

Solange die Verbindung besteht, erhält der Datenbankadministrator alle Meldungen von UDS/SQL an dieses Gerät.

Verbindung zu UDS/SQL aufbauen

In einer Terminalemulation können Sie eine Verbindung zu der Anwendung *konfname* auf dem BS2000-Rechner *rechner* aufbauen. Ein evtl. notwendiges Administrationspasswort *kennwort1* übergeben Sie als Benutzernachricht.

Der Anschluss über OMNIS ist mit folgendem Kommando möglich (siehe Handbuch „[OMNIS/OMNIS-MENU \(BS2000\) - Funktionen und Kommandos](#)“):

```
O[PNCON] pac,PT=konfname,PR=rechner,TYP=DCAM[ ,CMMSG | LMSG=kennwort2]
```

konfname

UDS/SQL-Konfigurationsname; dient als DCAM-Anwendungsname. Es sind die ersten acht Zeichen des Konfigurationsnamens aus dem Kommando

```
/SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME=konfigurationsname
```

rechner

BCAM-Name des Verarbeitungsrechners

pac

Partner-Adress-Code der UDS/SQL-Konfiguration

MSG=*kennwort1* oder

CMMSG=*kennwort2* oder

LMSG=*kennwort2*

Kennwort zur Administration mit DCAM-Anschluss. Muss nur angegeben werden, wenn mit PP ADMPASS ein Kennwort vereinbart wurde.

kennwort1

Ein bis vier byte langes Schlüsselwort

C' *xxxx*':

Ein bis vier alphanumerische Zeichen

X' *nnnnnnnr*':

Ein bis acht Sedezimalziffern

kennwort2

Ein bis vier byte langes Schlüsselwort

xxxx

Ein bis vier Zeichen ohne Datentyp-Angabe und ohne Hochkommata.

Das Kennwort darf nicht im Format C'...' oder X'...' angegeben werden, da OMNIS diese Zeichen nicht als Formatkennzeichnung interpretiert, sondern sie unverändert in der Verbindungsnachricht weitergibt.

Weiter müssen Sie beachten, dass der Parameter LCASE bei OPNCON noch ohne Wirkung ist und Kleinbuchstaben bei CMSG oder LMSG immer in Großbuchstaben umgewandelt werden. Diese Einschränkungen müssen Sie gegebenenfalls bei der Vergabe des Administrationskennworts berücksichtigen.

Nach dem Verbindungsaufbau betrachtet UDS/SQL diese Datensichtstation als Datensichtstation des Datenbankadministrators. Ein Verbindungswunsch wird abgelehnt, wenn das Programm UDSADM oder eine andere Datensichtstation des Datenbankadministrators bereits mit der Konfiguration verbunden ist. Ebenso ist es nicht möglich, über das Programm UDSADM oder die Operatorkonsole UDS/SQL-Kommandos einzugeben, während die Datensichtstation des Datenbankadministrators über DCAM verbunden ist.

Nachdem Sie sich an die DCAM-Anwendung angeschlossen haben, geben Sie das Kommando direkt an, d.h. ohne INFORM-PROGRAM oder SEND-MESSAGE:

dal-kommando

Wenn Sie die Taste [K1] drücken, wird das Senden einer Nachricht von UDS/SQL um 20 Sekunden verzögert, so dass eine Dateneingabe nicht gestört wird.

Verbindung zu UDS/SQL abbauen

Mit dem folgenden Kommando löst der Datenbankadministrator seine Verbindung zu UDS/SQL:

```
CLOSE ADMINISTRATION
```

Die DCAM-Anwendung bleibt erhalten. Ein späteres Wiederankoppeln einer beliebigen Datensichtstation oder eines Programms ist möglich.

Solange über DCAM keine Verbindung zu UDS/SQL besteht, wird die Administration über den lokalen Bedienplatz abgewickelt. Als lokaler Bedienplatz gilt die Datensichtstation, über die Kommandos einzugeben sind, wenn der Ladeparameter PP ADM=LOCAL angegeben ist. Das können sein:

- Die Operatorkonsole, wenn die Mastertask als ENTER-Job läuft.
- Die Datensichtstation, die an die Mastertask angeschlossen ist, wenn die Mastertask als Dialogauftrag läuft.

Administration protokollieren

Wenn eine Datensichtstation eine Verbindung mit DCAM aufgebaut hat, werden die Nachrichten an UDS/SQL und die Meldungen von UDS/SQL an dem ursprünglichen Bedienplatz zusätzlich protokolliert.

Als ursprünglicher Bedienplatz gilt:

- Die Operatorkonsole, wenn die Mastertask als ENTER-Job läuft.
- Die Datensichtstation, die an die Task angeschlossen ist, wenn die Mastertask als Dialogauftrag läuft.

Mit Hilfe des BS2000-Kommandos:

```
/MODIFY-JOB-OPTIONS logging=*par(listing=*yes)
```

und einer geeigneten Zuweisung der SYSLST-Datei können Sie das Protokoll auch in einer Datei ablegen.

Mit dem Ladeparameter PP UCON kann ein anderer Bedienplatz als der ursprüngliche als Ausgabeort für die Protokollierung festgelegt werden, siehe „PP UCON“ im Kapitel ["Bedienplatz zur Protokollierung der DCAM-Administration und Form der UDS/SQL-Meldungen festlegen \(UCON\)"](#).

Reaktionen in Fehlersituationen

Wenn während des UDS/SQL-Betriebs Fehler auftreten (z.B. Leitungsverlust) oder wenn die DCAM-Anwendung gewaltsam beendet wird (z.B. durch das Netzmanagementkommando BCAPPL MODE=DEACTIVATE), versucht UDS/SQL sofort, seine Anwendung neu zu eröffnen. Wenn der Versuch misslingt, wird er alle zehn Minuten wiederholt. Sobald eine neue Anwendung eröffnet ist, kann der Datenbankadministrator erneut Verbindung aufnehmen.

Meldungen der UDS/SQL-DCAM-Administration

Der Zustand der DCAM-Administration wird über eine UDS/SQL-Meldung am lokalen Bedienplatz protokolliert. Die Meldung heißt:

```
% UDS0347 UDS-ADMINISTRATION: ( s1) ( s2)
```

'BLANK' s1: Beschreibung des Ereignisses

'BLANK' s2: UDS/SQL-Diagnose Insert

Es werden die folgenden Ereignisse mit entsprechenden Zusatzangaben gemeldet:

- Eröffnen der DCAM-Anwendung mit Anwendungsname
- Verbindungsaufbau mit Datensichtstationsname
- Verbindungsabbau mit Datensichtstationsname
- Verbindungsabbruch mit Feedback-Information
- DCS-Abbruch mit Feedback-Information
- SVC-Fehler in der Administrationsroutine mit Makroname und Fehlerstatus bzw. Feedback-Information.

Weitere Informationen können Sie den Handbüchern zu „[DCAM \(BS2000\)](#)“ entnehmen.

4.3 UDS/SQL über INFORM-PROGRAM administrieren

Wenn Sie die UDS/SQL-Session über INFORM-PROGRAM steuern wollen, gibt es zwei verschiedene Eingabemöglichkeiten (siehe unten).

Möglichkeit 1: Eingabe von der Operatorkonsole

Wenn der DBH durch einen ENTER-Job gestartet wurde, können Sie DAL-Kommandos von der Operatorkonsole mit dem INFORM-PROGRAM-Kommando eingeben:

```
/INFORM-PROGRAM MSG='dal-kommando' ["kommentar"], JOB-ID=*TSN( tsn),
```

dabei bedeutet:

tsn Prozessfolgennummer der Mastertask

dal-kommando müssen Sie in Großbuchstaben eingeben. Abhängig von der Betriebssystem-Version muss das Zeichen „&“ doppelt angegeben werden. Eventuelle Hochkommata müssen ebenso doppelt angegeben werden.

Möglichkeit 2: Eingabe von der Datensichtstation des Datenbankadministrators

Wenn der independent DBH von der Datensichtstation des Datenbankadministrators aus als Dialogauftrag gestartet wurde, läuft der gesamte Dialog mit dem DBH über diese Datensichtstation. Die Mastertask hat darüberhinaus auch während der Session Sonderaufgaben durchzuführen, die einen reibungslosen Ablauf Gewähr leisten sollen.

Es ist daher darauf zu achten, dass die Mastertask nie unnötig und nur kurzzeitig zur Eingabe von DAL-Kommandos unterbrochen wird, wenn sie als Dialogauftrag geladen ist. Sie sollten darauf achten, dass Sie nach Eingabe von BS2000-Kommandos immer RESUME-PROG eingeben.

Zur Eingabe eines Kommandos müssen Sie die Tasten [EM],[DUE] bzw. die Taste [K2] drücken und damit die Mastertask unterbrechen. Danach können Sie das jeweilige DAL-Kommando durch ein INFORM-PROGRAM-Kommando an die Mastertask übergeben:

```
/INFORM-PROGRAM MSG='dal-kommando' ["kommentar"] [, JOB-ID=*OWN]
```

i Die Mastertask darf nicht für längere Zeit unterbrochen werden, da sonst die Kommunikation zwischen den UDS/SQL-Tasks unterbrochen wird.

dal-kommando müssen Sie in Großbuchstaben eingeben. Abhängig von der Betriebssystem-Version muss das Zeichen „&“ doppelt angegeben werden. Eventuelle Hochkommata müssen ebenso doppelt angegeben werden.

4.4 Die Datenbankadministratorsprache DAL

Bei einer Session mit dem independent DBH können Sie mit den Kommandos der Datenbankadministratorsprache DAL (Database Administrator Language) dem independent DBH Anweisungen geben und steuernd in den Ablauf der Session eingreifen.

DAL-Syntax

Die Namen der DAL-Kommandos und ihrer Operanden dürfen Sie abkürzen. Es gelten folgende Abkürzungsregeln:

- Es genügt, die ersten drei Zeichen des Namens anzugeben; dies gilt allerdings nicht für %BIB, %DML, %TERM.
- Werden mehr als drei Zeichen angegeben, so müssen Sie auch bei diesen nachfolgenden Zeichen die korrekte Syntax angeben.
- Bei Angabe von mehr als drei Zeichen kann die Eingabe des DAL-Kommandos auch so abgekürzt werden, dass sie mit einem im DAL-Kommando enthaltenen Bindestrich endet. Bei Eingabe über UDSADM ist hierbei aber zu beachten, dass UDSADM einen abschließenden Bindestrich als Fortsetzungszeichen interpretiert.
- Namen, die kürzer als drei Zeichen sind, müssen vollständig angegeben werden.

Bei den **UDS-D**-spezifischen DAL-Kommandos genügen die ersten drei Zeichen nach „&“.

Die Gesamtlänge der DAL-Kommandos darf max. 64 Zeichen betragen. Das kann bei DAL-Kommandos, in denen Datenbankname und Realm-Name angegeben werden müssen, zu Problemen führen.

Diese Probleme lassen sich vermeiden, wenn die Realm-Namen konfigurationsweit eindeutig gehalten werden (dann ist die Angabe des Datenbanknamens überflüssig), oder wenn die Datenbanknamen und Realm-Namen nicht zu lang gewählt werden (siehe Handbuch „[Entwerfen und Definieren](#)“, Nachschlageteil).

Eine Hilfe für die Korrektur möglicher Fehler wegen zu langer DAL-Kommandos gibt UDS/SQL mit der Meldung UDS0220.

Beim Empfang eines DAL-Kommandos wird von UDS/SQL folgende Meldung ausgegeben:

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN
```

Diese Meldung enthält als Insert (&01) das DAL-Kommando, das UDS/SQL entgegengenommen hat. Aus diesem Insert ist ersichtlich, was bei UDS/SQL tatsächlich angekommen ist. Insbesondere ist erkennbar, ob Teile des Kommandos, das Sie eingegeben haben, abgeschnitten wurden.

Das Ende der Kommandobearbeitung wird bei fehlerfreien DAL-Kommandos mit folgender Meldung angezeigt:

```
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL KOMMANDOS ABGESCHLOSSEN
```

bzw. bei UDS-D DAL-Kommandos

```
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT
```

Bei Fehlern in DAL-Kommandos erhalten Sie die Meldung

```
% UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN
```

Diese kann mehrfach mit unterschiedlich detaillierten Angaben zur Ursache erscheinen.

Bei einigen DAL-Kommandos muss *transaktionskennung* angegeben werden:

transaktionskennung:

bis zu achtstellige Dezimalzahl, die der DBH zur Kennzeichnung einer Transaktion vergibt; sie kann mit dem DISPLAY-Kommando erfragt werden.

Ausführung der Kommandos

Bei den meisten DAL-Kommandos wird die gewünschte Funktion sofort ausgeführt oder zumindest angestoßen. Bei den folgenden DAL-Kommandos wird die gewünschte Funktion nur vorgemerkt:

ACT INCR
ACT DBTT-INCR
ADD DB
ADD RN
DEACT INCR
DEACT DBTT-INCR
DROP DB
DROP RN
NEW PUBSETS
NEW RLOG
CHECKPOINT
ACCESS DB (wenn eine Kombination mit ADD DB vorliegt)

Die vorgemerkten Aufträge werden erst ausgeführt, wenn das Kommando PERFORM gegeben wird. Durch dieses Verfahren haben Sie die Möglichkeit, während der laufenden Session mehrere Aufträge zu sammeln. Nach dem PERFORM kann der DBH die Aufträge auf einmal bearbeiten und braucht interne Sperren nur einmal zu setzen.

Sie dürfen die Aufträge in beliebiger Reihenfolge eingeben. Der DBH arbeitet die Aufträge aber in bestimmter Reihenfolge ab. Die Reihenfolge richtet sich danach, ob

- die Aufträge sich auf Datenbanken beziehen, die zum Zeitpunkt, wenn das Kommando PERFORM gegeben wird, der aktuellen Konfiguration **angeschlossen sind**
- die Aufträge sich auf Datenbanken beziehen, die zum Zeitpunkt, wenn das Kommando PERFORM gegeben wird, der aktuellen Konfiguration **nicht angeschlossen sind**.

Bei Datenbanken, die angeschlossen sind, wird z.B. ADD RN vor DROP DB ausgeführt. Das bedeutet, dass ausgefallene Realms, falls sie repariert wurden, zuerst verfügbar gemacht werden und erst dann die gesamte Datenbank ausgeschlossen wird.

In welcher Reihenfolge diese Aufträge abgearbeitet werden, können Sie der folgenden Tabelle entnehmen.

Der Auftrag bezieht sich auf Datenbanken, die angeschlossen sind.	Der Auftrag bezieht sich auf Datenbanken, die nicht angeschlossen sind.
ACT INCR DEACT INCR ACT DBTT-INCR DEACT DBTT-INCR ADD RN= DROP RN= CHECKPOINT [DB=] DROP DB=	ADD DB= CHECKPOINT DB= ACCESS DB=

Tabelle 13: Reihenfolge der Abarbeitung von Aufträgen nach PERFORM

Folgende DAL-Kommandos kennt der independent DBH:

Die DAL-Kommandos für UDS-D beginnen mit dem Sonderzeichen &. Sie werden nur ausgewertet, wenn der UDS-D-Betrieb gestartet ist.

Sie sind in die alphabetische Reihenfolge der übrigen DAL-Kommandos eingeordnet.

DAL-Kommando	Bedeutung
<code><u>ABORT</u> { <i>transaktionskennung</i> [, <u>OPTION</u>=<u>PTC</u>] . <u>ALL</u> }</code>	Angegebene offene Transaktionen zurücksetzen OPTION=PTC: ohne Rücksicht auf konfigurationsübergreifende Konsistenz bzw. UDS/SQL-openUTM-Konsistenz
<code><u>ACCESS</u> { <u>LOCK</u> <u>RETRIEVAL</u> <u>UPDATE</u> } , { <u>DB</u>=<i>dbname</i> . <u>RN</u>=<i>realmname</i> [, <u>DB</u>=<i>dbname</i>] }</code>	Zugriffssperren auf Datenbank- und Realm-Ebene behandeln
<code><u>ACT</u> { <u>DBTT</u>-<u>INCR</u> , <u>DB</u>=<i>dbname</i> [, <u>RECR</u>=<i>recordref</i>] [, <u>EXT</u>=<i>extnr</i>] [, <u>SCAN</u>={ <u>YES</u> <u>NO</u> } <u>INCR</u> , <u>DB</u>=<i>dbname</i> [, <u>RR</u>=<i>realmref</i>] [, <u>EXT</u>=(<i>nr-pages</i> , <i>min-pages</i>)] }</code>	Online-Erweiterbarkeit von DBTTs bzw. Realms aktivieren
<code><u>ADD</u> { <u>DB</u>=[<i>\$userid</i> .] <i>dbname</i> [. <i>copyname</i>] [, <u>OPTION</u>=[<u>SHARED</u>-<u>RETRIEVAL</u>] [, <u>OWN</u>-<u>BUFFER</u>-<u>SIZE</u>=<i>n</i>] [, <u>ID</u>=<i>bufferid</i>] . <u>RN</u>=<i>realmname</i> [, <u>DB</u>=<i>dbname</i>] <u>PW</u>=<i>kennwort</i> <u>ADM</u>=<i>admkenwort</i> }</code>	Datenbanken, Realms und Kennwörter anschließen

<pre>&ADD DISTRIBUTION, {<u>N</u>ODE=prozessorname, <u>C</u>ONF=konfname[,<u>D</u>B=dbname] <u>D</u>B=dbname, <u>S</u>S=subschemaname . <u>F</u>ILE=dateiname }</pre>	<p>Neue Einträge in die Verteiltabelle bringen</p>
<pre>%BIB</pre>	<p>Anzahl der Nachrichten an den UDS/SQL-DBH ausgeben, die UDSCON weitergeleitet hat. Gezählt werden nicht nur die DML-Anweisungen, die von der COBOL-DML, CALL-DML, SQL oder KDBS kommen, sondern auch die Nachrichten der Laufzeit-Systeme und von UDSCON.</p>
<pre>&CHANGE DISTRIBUTION, <u>N</u>ODE=prozessorname, . <u>C</u>ONF=konfname</pre>	<p>Eine Konfiguration einem anderen Verarbeitungsrechner zuordnen</p>
<pre><u>C</u>HECKPOINT[<u>D</u>B=dbname[,<u>O</u>PTION=<u>E</u>VEN-WITHOUT-<u>A</u>LOG]]</pre>	<p>Konsistenzpunkte festschreiben im AFIM-Logging für einzelne Datenbanken der Konfiguration oder die gesamte Konfiguration</p>
<pre><u>C</u>LOSE {<u>R</u>UN-UNITS <u>C</u>ALLS <u>A</u>DMINISTRATION}</pre>	<p>Laufende Session normal beenden bzw. DCAM-Administration beenden</p>
<pre>&CLOSE DISTRIBUTION</pre>	<p>UDS-D-Betrieb beenden</p>
<pre><u>C</u>OMMIT <i>transaktionskennung</i></pre>	<p>Transaktion im Zustand PTC beenden mit Festschreiben der Änderungen (FINISH) ohne Rücksicht auf konfigurationsübergreifende Konsistenz bzw. UDS/SQL-openUTM-Konsistenz</p>
<pre><u>C</u>ONTINUE</pre>	<p>In der Ausgabe des DISPLAY SQL-DAL-Kommandos „blättern“</p>
<pre><u>D</u>EACT {<u>D</u>BT-<u>I</u>NCR, <u>D</u>B=dbname[, <u>R</u>ECR=recordref] <u>I</u>NCR, <u>D</u>B=dbname[, <u>R</u>R=realmref] }</pre>	<p>Online-Erweiterbarkeit von DBTTs bzw. Realms deaktivieren</p>

<pre> DISPLAY {DB[,RNID=transaktionskennung] USERS[,DB=dbname] SUBSCH[,DB=dbname] [,LINES={n ALL}] MAINREF[,STATE=BLOCK] transaktionskennung REALMS[,DB=dbname][,RN=realmname] [,LINES={n ALL}] PP FPA [,DB=dbname][,RN=realmname] [,LINES={n ALL}] INCR [,DB=dbname][,RN=realmname] [,LINES={n ALL}] DBTT-INCR [,DB=dbname [, RECR=recordref]] [,LINES={n ALL}] PUBSETS ALOG [,DB=dbname] } </pre>	<p>Datenbanken, Transaktionen, Subschemas, Mainrefs, noch verfügbaren Freiplatz oder Programmparameter der Konfiguration auflisten, bzw. Informationen über die Online-Erweiterbarkeit von Realms oder DBTTs ausgeben oder UDS/SQL-Pubset-Deklaration ausgeben</p>
<pre> &DISPLAY DISTRIBUTION [,NODE=prozessorname] [,CONF=konfname] [,DB=dbname] [,SS=subschemaname] </pre>	<p>Verteiltabelle ausgeben</p>
<pre> DISPLAY SQL{ ,VG=vgnr ,OPTION=ALL[,VG=vgnr] ,OPTION=IDLE[,TIME=z] } </pre>	<p>Detaillierte Informationen über einen oder mehrere SQL-Vorgänge ausgeben</p>
<pre> %DML </pre>	<p>Anzahl der Nachrichten an den UDS/SQL-DBH ausgeben, die UDSCON weitergeleitet hat. Gezählt werden nicht nur die DML-Anweisungen, die von der COBOL-DML, CALL-DML, SQL oder KDBS kommen, sondern auch die Nachrichten der Laufzeit-Systeme und von UDSCON.</p>
<pre> DROP {DB=dbname RN=realmname[,DB=dbname] PW=kennwort ADM=admkenwort} </pre>	<p>Datenbanken, Realms und Kennwörter ausschließen</p>
<pre> &DROP DISTRIBUTION, {ALL NODE=prozessorname, CONF=konfname[,DB=dbname][,ALL] DB=dbname, SS=subschemaname } </pre>	<p>Einträge in der Verteiltabelle löschen</p>
<pre> DUMP [{ALL STD transaktionskennung}] </pre>	<p>Speicherabzug vom DBH erstellen</p>

<code>EXTEND DBTT, DB=dbname, RECR=recordref [,EXT=extnbr]</code>	Online-DBTT-Erweiterung ausführen
<code>EXTEND REALM, DB=dbname, RR=realmref, EXT=nr-pages</code>	Online-Realm-Erweiterung ausführen
<code>FORGET SQL, VG=vg-nr</code>	Betriebsmittel freigeben, die zu dem SQL-Vorgang mit der Nummer <i>vg-nr</i> gehören
<code>GO {transaktionskennung ALL OLD}</code>	Ausführung der angegebenen Transaktion(en) fortsetzen
<code>&LOCK DISTRIBUTION, {NODE=prozessorname CONF=konfname DB=dbname SS=subschemaname }</code>	Einträge der Verteiltabelle sperren
<code>MODIFY {ALOG ALOG-RES}, DB=dbname[,VALUE={:catid: PUBLIC}] [,SHARE={YES NO}]</code>	Einstellungen für DEFAULT-SUPPORT und RESERVE-SUPPORT der ALOG-Dateien ändern
<code>MODIFY ALOG-SIZE, DB=dbname, VALUE=([primär] [, [sekundär]])</code>	Speicherplatzzuweisung für die ALOG-Dateien ändern
<code>MODIFY {LOG LOG-2}, VALUE={:catid: PUBLIC (priv-vsn-1/gerät-1 [,priv-vsn-2 gerät-2 [,priv-vsn-3 gerät-3]]) (vsn-1[,vsn-2[,vsn-3]])}</code>	Datenträgerzuweisung für eine neu einzurichtende Original- und Duplikat-RLOG-Datei ändern
<code>MODIFY LOG-SIZE, VALUE=([primär] [, [sekundär]])</code>	Speicherplatzgröße der RLOG-Dateien ändern
<code>MODIFY PTCSYNCH, VALUE=([{WAIT ABORT COMMIT}] [, [{WAIT ABORT COMMIT}]])</code>	Wert des DBH-Ladeparameters PTCSYNCH ändern
<code>MODIFY RESERVE, VALUE={NONE :catid: PUBLIC (priv-vsn-1/gerät-1 [,priv-vsn-2 gerät-2 [,priv-vsn-3 gerät-3]]) (vsn-1[,vsn-2[,vsn-3]])}</code>	Ersatzdatenträger für RLOG-Dateien ändern
<code>NEW PUBSETS</code>	Neue UDS/SQL-Pubset-Deklaration prüfen und vormerken
<code>NEW RLOG</code>	Neue RLOG-Dateien auswählen

<code><u>PERFORM</u> {<u>NOCANCEL</u> <u>CANCEL</u>}</code>	Aufträge anstoßen aus den Kommandos: DROP, ADD, NEW bzw. CHECKPOINT Standardwert: NOCANCEL
<code>&<u>PWD</u> <u>DISTRIBUTION</u>,<u>CONF</u>=<i>konfname</i>, {<u>PWN</u>=<i>kennwortneu</i> <u>PWO</u>=<i>kennwortalt</i> <u>PWO</u>=<i>kennwortalt</i>,<u>PWN</u>=<i>kennwortneu</i>}</code>	Kennwort vergeben und verändern
<code><u>REACT</u> <u>INCR</u>,<u>DB</u>=<i>dbname</i>[, <u>RR</u>=<i>realmref</i>]</code>	Online-Realm-Erweiterbarkeit reaktivieren
<code><u>RESET</u> <u>ORDERS</u></code>	Vorgemerkte Aufträge zurücknehmen
<code>&<u>SAVE</u> <u>DISTRIBUTION</u>,<u>FILE</u>= [:<i>catid</i>:] [<i>\$userid.</i>] <i>dateiname</i></code>	Verteiltabelle sichern
<code>&<u>START</u> <u>DISTRIBUTION</u></code>	UDS-D-Betrieb starten
<code><u>STOP</u> { <i>transaktionskennung</i> <u>NEW</u> <u>ALL</u> }</code>	Ausführung der angegebenen Transaktion(en) stoppen
<code>&<u>SYNCHRONIZE</u> <u>DISTRIBUTION</u></code>	Sekundäre Teiltransaktionen im Zustand PTC beenden
<code><u>%TERM</u></code>	Session auf dem kürzesten Weg abbrechen (Nothalt) und wahlweise einen kompletten Speicherabzug des DBH ausgeben ! VORSICHT! %TERM sollten Sie nur dann verwenden, wenn die Mastertask wegen eines Fehlers keine anderen DAL-Kommandos mehr akzeptiert.
<code>&<u>UNLOCK</u> <u>DISTRIBUTION</u>, {<u>NODE</u>=<i>prozessorname</i> <u>CONF</u>=<i>konfname</i> <u>DB</u>=<i>dbname</i> <u>SS</u>=<i>subschemaname</i> }</code>	Sperren für Einträge in der Verteiltabelle aufheben

Tabelle 14: DAL-Kommandos für den independent DBH

Folgende DAL-Kommandos kennt der linked-in DBH:

DAL-Kommando	Bedeutung
<u>%BIB</u>	Anzahl der abgearbeiteten <u>B</u> ase <u>I</u> nterface <u>B</u> locks ausgeben, die in diesem Session-Abschnitt bearbeitet wurden
<u>%DUMP</u>	Kompletten Speicherabzug des DBH mit aufbereiteten Tabellen auf Datei ausgeben. Das Programm wird nicht abgebrochen.

Tabelle 15: DAL-Kommandos für den linked-in DBH

Im Folgenden sind alle DAL-Kommandos ausführlich beschrieben.

4.4.1 Eine oder alle Transaktionen abbrechen (ABORT)

```
ABORT { transaktionskennung [ ,OPTION=PTC ] | ALL }
```

Standardwert:

ohne OPTION=PTC

transaktionskennung

bezeichnet eine offene Transaktion, die zurückgesetzt werden soll.

OPTION=PTC

setzt die angegebene Transaktion auch dann zurück, wenn sie sich im Zustand PTC befindet (nur für UDS-D/openUTM).

ALL setzt alle zur Zeit offenen Transaktionen zurück, mit Ausnahme derjenigen, die sich im Zustand PTC befinden.

Das ABORT-Kommando setzt die angegebene Transaktion bzw. alle zurzeit offenen Transaktionen zurück. Das betroffene COBOL(/CALL)-DML-Anwenderprogramm erhält den Statuscode '122'. SQL-Programme erhalten einen Statuscode < -1000 (SQL-Code).

Der DBH akzeptiert ein ABORT-Kommando grundsätzlich nur für solche Transaktionen, die ohne Konsistenzverlust für den Datenbestand zurückgesetzt werden können.

Ohne den Zusatz OPTION=PTC wird ABORT ignoriert, wenn sich die Transaktion im Zustand PTC befindet.

Mit dem Zusatz OPTION=PTC setzt der Datenbankadministrator 'per Hand' eine sekundäre Teiltransaktion bzw. eine openUTM-Transaktion zurück, auch wenn sie sich im Zustand PTC befindet und somit auf eine Beendigungsnachricht von ihrer primären Teiltransaktion bzw. von openUTM wartet.

Die Wirkung des DBH-Ladeparameters PP PTCSYNCH ist für diese Transaktion aufgehoben.

Kann eine Transaktion, z.B. wegen fehlerhafter oder unvollständiger RLOG-Datei nicht zurückgesetzt werden, so wird das ABORT-Kommando vom DBH ignoriert.

i Wenn Sie eine sekundäre Teiltransaktion bzw. eine openUTM-Transaktion im Zustand PTC zurücksetzen, gefährden Sie die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz (siehe [Abschnitt „Zustand PTC beenden“](#)).

4.4.2 Zugriffssperren auf Datenbank- und Realm-Ebene behandeln (ACCESS)

```
ACCESS {LOCK | RETRIEVAL | UPDATE},
       {DB=dbname | RN=realmname [,DB=dbname] }
```

dbname

Name einer Datenbank der DB-Konfiguration

realmname

Name eines Realms der Datenbank

Mit dem ACCESS-Kommando können Sie die Zugriffsrechte auf die Realms einer Datenbank bzw. auf die gesamte Datenbank beeinflussen. Sie können Zugriffe sperren, einschränken oder Zugriffssperren aufheben. Die Sperren wirken nur logisch, enthalten aber kein physisches Ausschließen der Dateien.

Zugriffe sperren

```
ACCESS LOCK,DB=dbname
```

auf die mit *dbname* angegebene Datenbank.

```
ACCESS LOCK,RN=realmname [,DB=dbname]
```

auf den mit *realmname* angegebenen Realm. Die Angabe von DB=*dbname* ist nur nötig, wenn der Realm-Name nicht innerhalb der gesamten DB-Konfiguration eindeutig ist. Sie spezifiziert dann die gewünschte Datenbank.

ACCESS LOCK bewirkt, dass der DBH keinerlei Zugriffe mehr auf den Realm bzw. die Datenbank zulässt.

ACCESS LOCK wird nur bei READY (Statuscode 12022) wirksam, d.h. Transaktionen, die auf dem Realm bzw. der Datenbank schon arbeiten, können noch bis zum FINISH weiterlaufen. Der ändernde Zugriff wird aber dann gesperrt, wenn Sie versuchen, mit READY UPDATE eine Verarbeitungskette für einen weiteren Realm zu eröffnen, der bereits mit ACCESS LOCK für Änderungen gesperrt wurde.

Änderungszugriffe sperren

ACCESS RETRIEVAL,DB=*dbname*

auf die mit *dbname* angegebene Datenbank.

ACCESS RETRIEVAL,RN=*realmname* [,DB=*dbname*]

auf den mit *realmname* angegebenen Realm. Die Angabe von DB=*dbname* ist nur nötig, wenn der Realm-Name nicht innerhalb der gesamten DB-Konfiguration eindeutig ist. Sie spezifiziert dann die gewünschte Datenbank.

ACCESS RETRIEVAL sperrt den angegebenen Realm bzw. die angegebene Datenbank solange für ändernde Transaktionen, bis ein ACCESS UPDATE-Kommando die Sperre aufhebt.

Lesende Transaktionen sind dagegen möglich.

ACCESS RETRIEVAL wird in CODASYL-Anwendungen bei READY (Statuscode 12123) wirksam, in SQL-Anwendungen beim ersten ändernden DB-Zugriff; d.h. ändernde Transaktionen, die auf dem Realm bzw. der Datenbank schon arbeiten, können noch bis zum FINISH weiterlaufen. Der ändernde Zugriff wird aber dann nicht mehr zugelassen, wenn Sie versuchen, mit READY UPDATE eine Verarbeitungskette für einen weiteren Realm zu eröffnen, der bereits mit ACCESS RETRIEVAL für Änderungen gesperrt wurde.

In SQL-Anwendungen werden ändernde SQL-Anweisungen abgewiesen.

Zugriffssperre aufheben

ACCESS UPDATE,DB=*dbname*

bei der mit *dbname* angegebenen Datenbank.

ACCESS UPDATE,RN=*realmname* [,DB=*dbname*]

bei dem mit *realmname* angegebenen Realm. Die Angabe von DB=*dbname* ist nur nötig, wenn der Realm-Name nicht innerhalb der gesamten DB-Konfiguration eindeutig ist. Sie spezifiziert dann die gewünschte Datenbank.

ACCESS UPDATE hebt ein vorangegangenes ACCESS LOCK oder ACCESS RETRIEVAL-Kommando auf; d.h. es hebt die Sperre auf, die auf den angegebenen Realm bzw. die angegebene Datenbank gegeben wurde.

Das ACCESS-Kommando darf für eine Datenbank auch zwischen den Kommandos

ADD DB und PERFORM angegeben werden, d.h. sobald die Datenbank zum Anschließen angemeldet ist.

Damit kann erreicht werden, dass nach dem Anschließen der Datenbank die Zugriffssperre gilt, die mit ACCESS vereinbart wurde.

Beispielsweise ist dies sinnvoll, wenn ändernde Transaktionen auf einer Datenbank vorläufig verhindert werden sollen.

4.4.3 Online-Erweiterbarkeit aktivieren (ACT)

```
ACT {DBTT-INCR,DB=dbname[ , RECR=recordref][ , EXT=extnr][ , SCAN={YES | NO} |
INCR,DB=dbname[ , RR=realmref][ , EXT=(nr-pages,min-pages)]}
```

i Zwischen ACT und DBTT-INCR bzw. INCR ist genau ein Leerzeichen anzugeben. Weitere Leerzeichen sind in diesem DAL-Kommando nicht erlaubt.

Mit dem Kommando ACT können Sie die Online-Erweiterbarkeit für Realms und DBTTs aktivieren.

Online-DBTT-Erweiterbarkeit für eine Satzart aktivieren (ACT DBTT-INCR)

dbname

Name der Datenbank, zu der die Satzart bzw. die Satzarten gehören, für die die Online-DBTT-Erweiterung aktiviert werden soll. Die Datenbank und der entsprechende Realm müssen zum Zeitpunkt des DAL-Kommandos bereits zugeschaltet sein.

recordref

Nummer der Usersatzart, für die die Online-DBTT-Erweiterung aktiviert werden soll. Soll sie für alle Satzarten aktiviert werden, kann diese Angabe entfallen.

Die *recordrefs* der Satzarten können Sie mit dem DAL-Kommando DISPLAY DBTT-INCR anzeigen lassen oder Sie können sie dem BPSIA-Protokoll entnehmen.

recordref≠1 wird UDS/SQL-intern genutzt und kann hier nicht angegeben werden.

extnmbr

Anzahl der Einträge, um die die DBTT mindestens erweitert werden soll.

Die Angabe muss größer oder gleich 1 und kleiner oder gleich 99999999 sein.

Der Default-Wert ist die Anzahl von DBTT-Einträgen in einer DBTT-Erweiterung.

Den aktuellen Wert des EXT-Parameters können Sie mit dem DAL-Kommando DISPLAY DBTT-INCR ausgeben oder der BSTATUS-Ausgabe entnehmen.

In der Regel wird die tatsächliche Anzahl der DBTT-Einträge, um die erweitert wird, etwas größer als die gewünschte Anzahl sein. Die tatsächliche Anzahl hängt von folgenden Faktoren ab:

- Seitengröße der Datenbank (2K, 4K oder 8K)
- DBTT-Zeilenlänge der jeweiligen Satzart

Falls es innerhalb der DBTT-Erweiterung zu Engpässen kommt, kann die Anzahl der Einträge, um die erweitert wurde, auch geringer als die gewünschte Anzahl sein. In jedem Fall wird die Anzahl der DBTT Einträge, um die tatsächlich erweitert wurde, mit entsprechender Meldung ausgegeben.

SCAN=YES

Bei einem STORE wird die Online-DBTT-Erweiterung erst ausgeführt, wenn in der gesamten DBTT kein freier Eintrag mehr gefunden wurde - es sei denn, dieser Transaktion ist aus anderen Quellen bekannt, dass alle DBTT-Einträge belegt sind. SCAN=YES ist die Standardeinstellung und entspricht insoweit dem Verhalten bei nicht aktivierter Online-DBTT-Erweiterbarkeit, bei der bei einem STORE die DBTT noch einmal komplett durchlaufen wird, bevor „DBTT voll“ gemeldet wird.

Diese Einstellung hat zur Folge, dass diese eine Transaktion möglicherweise zu einer langen Antwortzeit führt und andere Transaktionen blockiert.

SCAN=NO

Die Online-DBTT-Erweiterung wird ausgeführt, sobald das DBTT-Ende beim Suchen nach einem freien Eintrag erreicht wird; aktuell bestehende DBTT-Bereiche werden immer nur einmal gescannt, solange diese Einstellung aktiv ist. Sinnvoll ist dies für Satzarten, in denen keine oder nur selten Sätze gelöscht werden oder bei denen die durch Löschen entstandenen Lücken in der DBTT tolerierbar sind.

Mit dem Kommando ACT DBTT-INCR wird ein Auftrag zur Aktivierung der Online-DBTT-Erweiterung einer oder mehrerer Satzarten vorgemerkt oder ein vorhergehender DEACT DBTT-INCR-Auftrag aufgehoben.

Der Auftrag wird erst durchgeführt, nachdem das Kommando PERFORM angegeben wurde. Sie werden mit der Meldung UDS0722 folgendermaßen über die Ausführung informiert:

- Wurde ein Aktivierungsauftrag für sämtliche Satzarten ausgeführt, so wird eine einzige Meldung für die Datenbank ausgegeben.
- Wurde ein Aktivierungsauftrag für eine einzelne Satzart ausgeführt, so wird eine satzartspezifische Meldung ausgegeben.
- Wurden sowohl Aktivierungsaufträge für einzelne Satzarten als auch für sämtliche Satzarten gegeben, so wird nur eine einzige Meldung für die Datenbank ausgegeben.

Werden mehrere ACT DBTT-INCR-Aufträge für dieselbe Satzart abgesetzt, dann gelten die Angaben des letzten Auftrags.

Ein DEACT DBTT-INCR-Auftrag hebt einen vorhergehenden ACT DBTT-INCR-Auftrag für eine Satzart auf und umgekehrt. Auf diese Weise können falsch abgesetzte ACT DBTT-INCR- bzw. DEACT DBTT-INCR-Aufträge rückgängig gemacht werden, bevor sie mit PERFORM wirksam werden.

Die Online-Erweiterung einer Satzart bleibt aktiviert, bis sie mittels DEACT DBTT-INCR deaktiviert wird.

Falls eine Online-DBTT-Erweiterung an einer suspendierten (d.h. wegen aufgetretener Fehler ausgesetzten) Realm-Erweiterung scheitert, wird maximal einmal pro Satzart eine Meldung ausgegeben, dass die Online-DBTT-Erweiterung gescheitert ist. Weitere Meldungen über gescheiterte Online-DBTT-Erweiterungen werden unterdrückt, bis diese Suspendierung der Online-Realm-Erweiterung beendet ist.

Falls bei der Aktivierung der Online-DBTT-Erweiterung ohne Angabe einer *recordref*, d.h. Aktivierung für alle Satzarten einer Datenbank, ein Realm mit DBTTs nicht zugeschaltet ist, unterbleibt die Aktivierung für die entsprechenden Satzarten. Meldung UDS0745 informiert über den betroffenen Realm.

Falls bei der Aktivierung der Online-DBTT-Erweiterung ohne Angabe einer *recordref*, d.h. Aktivierung für alle Satzarten einer Datenbank, die Online-Realm-Erweiterung für einen Realm mit DBTTs nicht aktiviert ist, oder ein Auftrag zur Deaktivierung der Online-Realm-Erweiterung bereits vorliegt, unterbleibt die Aktivierung für die entsprechenden Satzarten. Meldung UDS0744 informiert über den betroffenen Realm.

Das Kommando ACT DBTT-INCR wird mit Meldung UDS0209 sofort zurückgewiesen

- wenn die angegebene Satzartnummer nicht existiert.
- wenn die spezifizierte Datenbank nicht zugeschaltet oder im Modus SHARED-RETRIEVAL zugeschaltet ist.

- wenn bei einer angegebenen Satzart der entsprechende DBTT-Realm nicht zugeschaltet ist. Meldung UDS0745 informiert über den betroffenen Realm.
- wenn bei einer angegebenen Satzart die Online-Realm-Erweiterung des entsprechenden DBTT-Realms nicht aktiviert ist oder ein Auftrag zur Deaktivierung der Online-Realm-Erweiterung des DBTT-Realms bereits vorliegt. Meldung UDS0744 informiert über den betroffenen Realm.
- wenn ein Auftrag zur Aktivierung der Online-DBTT-Erweiterung für alle Satzarten einer Datenbank bei keiner Satzart zu einer Aktivierung oder zur Rücknahme eines gegenteiligen Auftrags führt.

Online-Erweiterbarkeit für einen Realm aktivieren (ACT INCR)

dbname

Name der Datenbank des Realms bzw. der Realms, für den bzw. die Online-Erweiterbarkeit aktiviert werden soll.

dbname muss zum Zeitpunkt, an dem dieses Kommando abgesetzt wird, zugeschaltet sein.

dbname darf sich nicht im Modus SHARED-RETRIEVAL befinden.

realmref

Nummer des Realms, für den die Online-Erweiterbarkeit aktiviert werden soll. Der Realm muss zum Zeitpunkt der Kommandoausführung zugeschaltet sein. Die Nummern der zugeschalteten Realms können Sie sich mit DISPLAY INCR anzeigen lassen.

Zulässig sind die Nummern aller Benutzerrealms und des DBDIR. Die Angabe der Nummer des DBCOM ist nicht zulässig.

Ist $RR=$ *realmref* nicht angeben, so wird das Kommando für alle gerade zugeschalteten Benutzerrealms und den DBDIR der Datenbank *dbname* mit den jeweils gleichen Werten ausgeführt.

nr-pages

Anzahl der Datenbankseiten, die bei einer Online-Realm-Erweiterung an den Realm angefügt werden sollen.

nr-pages=64 ... 16777215

Der Default-Wert ist 64.

nr-pages wird vom DBH bei der Online-Realm-Erweiterung um bis zu 64 erhöht, falls von dem Erweiterungsbereich Platz für einen neuen FPA-Extent abgezweigt werden muss.

Unabhängig von *nr-pages* erfolgt eine Online-Realm-Erweiterung höchstens bis zur Anzahl maximal möglicher Datenbankseiten im Realm.

Bei *nr-pages* handelt es sich nicht um die Anzahl der Seiten, um die der Realm gegebenenfalls durch das DVS erweitert wird, sondern um die Anzahl Datenbankseiten.

Im Falle eines Realms, welcher nur DBTTs enthält und der demzufolge nur zwecks Online-DBTT-Erweiterung zu aktivieren ist, ist die Angabe von *nr-pages* überflüssig, da UDS/SQL nach internen Vorgaben den Realm erweitert.

min-pages

Grenzwert der freien Datenbankseiten im Realm.

min-pages=0 ... *nr-pages*

Der Default-Wert ist 16.

min-pages wirkt sich wie folgt aus:

min-pages=0

Die Realm-Erweiterung wird angestoßen, wenn eine Freiplatzsuche erfolglos war.

min-pages>0

Unterschreitet die Anzahl der aktuell vorhandenen freien Datenbankseiten *min-pages*, so führt eine Freiplatzsuche zu einer Realm-Erweiterung.

Die Realm-Erweiterung wird auch durchgeführt, wenn dies für die Ausführung des aktuellen Auftrages noch nicht nötig wäre. Dadurch können während der Online-Realm-Erweiterung weitere Transaktionen ausgeführt werden, die freien Platz im Realm benötigen.

Im Falle eines Realms, welcher nur DBTTs enthält und der demzufolge nur zwecks Online-DBTT-Erweiterung zu aktivieren ist, ist die Angabe von *min-pages* überflüssig, da UDS/SQL nach internen Vorgaben den Realm erweitert.

Mit dem Kommando ACT INCR können Sie für den Realm *realmref* der Datenbank *dbname* einen Auftrag zur Aktivierung der Online-Erweiterbarkeit vormerken. Ist *RR=realmref* nicht angegeben, wird der Auftrag zur Aktivierung der Online-Erweiterbarkeit für alle zugeschalteten Realms der Datenbank *dbname* vorgemerkt.

Mit ACT INCR können Sie auch ein zuvor abgesetztes Kommando DEACT INCR widerrufen, sofern die Kommandos noch nicht mit PERFORM zur Durchführung angestoßen sind. Grundsätzlich wird ACT INCR erst ausgeführt, nachdem PERFORM angegeben wurde. Falls Sie für einen Realm mehrere ACT INCR-Aufträge abgesetzt haben, gelten die Angaben des letzten Auftrags.

Die eventuell zusätzlich zu ACT INCR vorgemerkten DAL-Aufträge dürfen auch Aufträge zum Abschalten der betroffenen Realms bzw. Datenbanken enthalten. Dabei ist es gleichgültig, ob die DROP-Aufträge vor oder nach den ACT INCR-Aufträgen eingegeben werden. Bei PERFORM werden in jedem Fall zuerst die ACT INCR-Aufträge bezüglich einer Datenbank bearbeitet.

Falls für den oder die angegebenen Realms die Online-Erweiterbarkeit ausgesetzt war, wird die Aussetzung durch ACT INCR aufgehoben, d. h. ACT INCR schließt die Funktionalität des Kommandos REACT INCR ein.

Die Online-Erweiterbarkeit eines Realms bleibt aktiviert, bis sie mittels DEACT INCR deaktiviert wird.

Bei aktivierter Online-Realm-Erweiterung kann es nach der Aufhebung einer Suspendierung in der Folge wiederum zu Meldungen über das Scheitern von DBTT-Erweiterungen in diesem Realm kommen, wenn erneut eine Suspendierung vorgenommen wird.

i Soll ein Realm online-erweiterbar sein, so liegt es in Ihrer Verantwortung als Administrator, dem Realm rechtzeitig genügend Speicherplatz oder eine Sekundärzuweisung größer 0 zuzuweisen. Dies kann aus DVS-Gründen nur offline geschehen. Eine Überprüfung dieser Voraussetzung wird vom DBH nicht durchgeführt (siehe auch [Kapitel „Ressourcen-Erweiterung und Reorganisation im laufenden Betrieb“](#)).

Das Kommando ACT INCR wird mit Meldung UDS0209 sofort zurückgewiesen,

- wenn für *nr-pages* oder *min-pages* ein ungültiger Wert angegeben wurde.
- wenn die Datenbank *dbname* nicht zugeschaltet oder im Modus SHARED-RETRIEVAL zugeschaltet ist.
- wenn die angegebene Realm-Nummer *realmref* nicht existiert.
- wenn als Realm-Nummer *realmref* diejenige des DBCOM angegeben wurde.
- wenn der angegebene Realm nicht zugeschaltet ist.

4.4.4 Datenbanken, Realms und Kennwörter anschließen (ADD)

```
ADD {DB=[ $userid. ] dbname[ .copyname ] [ ,OPTION=[ SHARED-RETRIEVAL ]
    [ ,OWN-BUFFER-SIZE=n ] [ ,ID=bufferid ] |
    RN=realmname [ ,DB=dbname ]
    PW=kennwort
    ADM=admkenntwort }
```

\$userid

Kennung, in der die betroffene Datenbank steht. Sie ist nur erforderlich, wenn die Datenbank nicht in der Kennung steht, in der der DBH gestartet wurde.

dbname

Name der Datenbank

copyname

Suffix der Schattendatenbank.

Ist *copyname* angegeben, so wird die entsprechende Schattendatenbank angeschlossen.

n Größe des User Buffer Pools für die Datenbank in Mbyte. Ist der Parameter *bufferid* angegeben, so wird der mit *n* spezifizierte Buffer Pool als Shared User Buffer Pool angelegt, (siehe unten).

n = 1 .. 2047

Bei einer Datenbank im 8-Kbyte-Seitenformat werden mindestens 3 Mbyte angelegt.

n = 0 Die Größe des User Buffer Pools ist bereits bei einer anderen Datenbank definiert.

n nicht angegeben und *bufferid* nicht angegeben

Die Datenbank wird in dem ihrem Seitenformat zugehörigen System Buffer Pool gepuffert. Wenn dieser System Buffer Pool nicht existiert, wird das Zuschalten der Datenbank abgewiesen.

n nicht angegeben und *bufferid* angegeben

Die Größe des User Buffer Pools ist bereits bei einer anderen Datenbank definiert.

bufferid

Identifikator eines Shared User Buffer Pools

max. 6 alphanumerische Zeichen (keine Sonderzeichen)

Ist *bufferid* angegeben, wird der Datenbank der mit *bufferid* bezeichnete Shared User Buffer Pool zugewiesen.

realmname

Name des Realms

kennwort

kennwort darf bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C' xxxx':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X' nnnnnnnr':

nnnnnnnn sind ein bis acht Sedezimalziffern

d:

d ist eine Dezimalzahl mit höchstens acht Ziffern und Vorzeichen, deren Wert zu einem Binärwert umgesetzt wird

Es gilt die BS2000-Syntax (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-PASSWORD)

adm kennwort

adm kennwort darf bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C' xxxx':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X' nnnnnnnr':

nnnnnnnn sind ein bis acht Sedezimalziffern

Mit dem Kommando ADD können Datenbanken, Realms und Kennwörter in eine laufende Session angeschlossen werden. Es kann aber auch ein DROP-Kommando mit ADD widerrufen werden, solange DROP und ADD noch nicht mit PERFORM zur Durchführung angestoßen sind. Grundsätzlich werden ADD DB und ADD RN erst ausgeführt, nachdem PER-FORM angegeben wurde.

Datenbank anschließen

ADD DB=[*\$userid.*] *dbname*

Der DBH notiert einen Auftrag zum Anschließen der genannten Originaldatenbank. Damit wird die aktuelle DB-Konfiguration verändert.

ADD DB=[*\$userid.*] *dbname.copyname*

Anschließen der entsprechenden Schattendatenbank. Damit wird die aktuelle DB-Konfiguration verändert.

Wenn für eine bereits angeschlossene Datenbank (Original oder Kopie) ein Abschaltauftrag (DROP DB) vorliegt, so wird dieser Abschaltauftrag durch ein folgendes DAL-Kommando ADD DB nur dann widerrufen, wenn der beim ADD DB angegebene *copyname* mit dem aktuell zugeschalteten Exemplar übereinstimmt (im Falle eines DB-Originals bedeutet dies: es ist kein *copyname* angegeben.) Andernfalls wird beim folgenden PERFORM das aktuelle Exemplar der Datenbank abgeschaltet und das Exemplar mit dem angegebenen Copynamen zugeschaltet.

Auf diese Weise kann bei einer PERFORM-Bearbeitung (d.h. in einem Schritt)

- ein DB-Original durch eine DB-Kopie
- eine DB-Kopie durch ein DB-Original
- ein DB-Kopie durch eine andere DB-Kopie

ersetzt werden.

OPTION-Standardwert

Bei Originaldatenbank:

Keine RETRIEVAL-Option, d.h. der DBH eröffnet die Datenbank exklusiv, die Anwenderprogramme können lesend und schreibend zugreifen.

Bei Schattendatenbank: SHARED-RETRIEVAL.

OPTION=SHARED-RETRIEVAL

Auf die anzuschließende Datenbank greift der DBH shared zu. Auch andere DBHs können auf die anzuschließende Datenbank mit OPTION=SHARED-RETRIEVAL zugreifen. Anwenderprogramme aller dieser DBHs können dann lesend auf der zugeschalteten Datenbank arbeiten.

i Um die zulässige Gesamtlänge von 64 Zeichen des Kommandos ADD DB nicht zu überschreiten, sollten Sie für die Eingabe des Schlüsselwortes SHARED-RETRIEVAL die Abkürzung SHA verwenden.

OWN-BUFFER-SIZE=*n*

Für die anzuschließende Datenbank kann die Größe eines User Buffer Pools angegeben werden. Bei einer Datenbank im 8-Kbyte-Seitenformat werden mindestens 3 Mbyte angelegt. Dieser Buffer Pool wird dynamisch beim Zuschalten der Datenbank zusätzlich zu den System Buffer Pools angelegt. Ist ID=*bufferid* nicht angegeben, so wird dieser Buffer Pool ausschließlich für die Pufferung von Seiten der betreffenden Datenbank verwendet. Ist ID=*bufferid* angegeben, wird der Datenbank ein Shared User Buffer Pool zugewiesen (s.u.)

Wenn OWN-BUFFER-SIZE nicht angegeben ist, wird die Datenbank in dem ihrem Seitenformat zugehörigen System Buffer Pool gepuffert. Wenn dieser System Buffer Pool nicht existiert, wird das Zuschalten der Datenbank abgewiesen.

ID=*bufferid*

Der mit OWN BUFFER-SIZE=*n* spezifizierte Buffer Pool wird mit dem Identifikator *bufferid* als Shared User Buffer Pool der Datenbank zugewiesen. Ein Shared User Buffer Pool kann mehreren Datenbanken als gemeinsamer exklusiver Buffer Pool zugewiesen werden.

Ist noch kein Buffer Pool *bufferid* vorhanden, so wird unter diesem Namen ein neuer Shared User Buffer Pool angelegt und der Datenbank zugewiesen. Andernfalls wird der Buffer Pool *bufferid* auch für die Pufferung der Seiten der betroffenen Datenbank verwendet. Hierfür müssen folgende Voraussetzungen gegeben sein:

- Die Seitengröße der Datenbank und des Shared User Buffer Pool passen zusammen
- Die mit OWN-BUFFER-SIZE=*n* angegebene Größe des Buffer Pools stimmt mit der Größe des vorhandenen Shared User Buffer Pools überein
 - oder
 - OWN-BUFFER-SIZE hat den Wert 0
 - oder
 - OWN-BUFFER-SIZE=*n* ist nicht angegeben.

Reaktionen in Fehlersituationen

ADD DB-Kommando wird sofort zurückgewiesen

- wenn mehr Datenbanken zum Anschließen angegeben wurden, als bei PP MAXDB festgelegt wurde.
- wenn eine Originaldatenbank gleichen Namens oder eine Schattendatenbank gleichen Namens bereits angeschlossen ist und kein Auftrag zum Ausschließen der Datenbank existiert.
- wenn für eine Datenbank gleichen Namens bereits ein Auftrag zum Anschließen vorliegt.
- wenn die mit OWN-BUFFER-SIZE=*n* angegebene Größe des Buffer Pools nicht mit der Größe des mit ID=*bufferid* angegebenen Shared User Buffer Pools übereinstimmt.
- wenn OWN-BUFFER-SIZE=*n* den Wert 0 hat, ID=*bufferid* angegeben ist und der Buffer Pool *bufferid* noch nicht (bei einer anderen Datenbank) definiert ist.

Während der Bearbeitung des Auftrags zum Anschließen einer Datenbank können Fehler auftreten

- wenn die angegebene Datenbank nicht existiert oder sich nicht in der gewünschten Benutzungsart zuschalten lässt.
- wenn die angegebene Datenbank inkonsistent ist und die Konsistenz nicht wiederhergestellt werden kann mit einem Warmstart, weil die Datenbank mit RETRIEVAL belegt ist, oder weil der DBH nicht auf die zugehörige RLOG-Datei und/oder DB-Status-Datei exklusiv zugreifen kann (siehe „[Auswertung der DBH-Ladeparameter](#)“ (Ladeparameter des DBH)).
- wenn auf Grund verteilter Verarbeitung mit UDS-D/openUTM auf der (inkonsistenten) Datenbank Transaktionen im Zustand Prepared to Commit (PTC) sind und bei UDS-D der PTCSYNCH-Wert für die laufende Session auf WAIT steht.
- wenn beim Anschließen der Dateien dieser Datenbank Fehler auftreten.
- wenn die Seitengröße der Datenbank und die des mit ID=*bufferid* angegebenen Shared User Buffer Pools nicht übereinstimmen.
- wenn eine der ALOG-Logging-Angaben, die mit den Parametern DEFAULT-SUPPORT oder RESERVE-SUPPORT der BMEND-Anweisung START-LOG vorgenommen wurden, außerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt.

Wenn beim Anschließen der Datenbank durch einen dieser Fehler die Konsistenz nicht wiederhergestellt werden kann, wird die Datenbank sofort wieder ausgeschlossen.

Kann die Konsistenz der Datenbank trotz der oben genannten Schwierigkeiten wiederhergestellt werden, so wird die Datenbank so vollständig wie möglich angeschlossen, damit wenigstens mit einer Teil-Realm-Konfiguration gearbeitet werden kann.

Realm anschließen

ADD RN=*realmname*[,DB=*dbname*]

Der DBH notiert einen Auftrag zum Anschließen des mit *realmname* angegebenen Realms bzw. streicht einen vorliegenden DROP-Auftrag.

Die Angabe von DB=*dbname* ist nur nötig, wenn der Realm-Name nicht innerhalb der gesamten DB-Konfiguration eindeutig ist. Sie spezifiziert dann die gewünschte Datenbank.

Mit dem Kommando ADD RN kann ein Realm, der für Reparaturmaßnahmen oder bei Geräte-Engpässen ausgeschlossen wurde (DROP), wieder in die laufende Session angeschlossen werden.

Der ADD RN-Auftrag wird erst ausgeführt, wenn Sie das PERFORM-Kommando geben.

Das ADD RN-Kommando kann dazu benutzt werden, ein vorangegangenes DROP RN-Kommando zu widerrufen. Bereits eingeleitete DROP-Aufträge können nicht mehr widerrufen werden.

Reaktionen in Fehlersituationen

Das ADD RN-Kommando wird zurückgewiesen, wenn:

- der genannte Realm bei den angeschlossenen Datenbanken nicht vorhanden ist
- der Realm-Name in der aktuellen DB-Konfiguration nicht eindeutig ist, weil die Angabe *DB=dbname* fehlt
- der genannte Realm schon angeschlossen ist und kein Auftrag vorliegt, diesen Realm auszuschließen
- der genannte Realm nicht angeschlossen werden kann, weil es ein Temporärer Realm ist oder dieser Realm ein Teil einer Datenbank ist, für die SHARED RETRIEVAL gilt
- beim Anschließen des Realms Fehler auftreten
- der Realm als defekt markiert oder inkonsistent ist.

Das Anschließen eines Realms ist erst dann ausfallsicher im DBDIR protokolliert, wenn der Auftrag komplett abgearbeitet wurde. Ausfallsicher heißt, es überdauert Session-Unterbrechungen und das DBH-Ende.

Kennwort für Dateien anschließen

ADD PW=*kennwort*

Anschließen des genannten Datei-Kennworts. Damit wird die aktuelle Menge der Kennwörter geändert.

Reaktionen in Fehlersituationen

Das ADD PW-Kommando wird zurückgewiesen, wenn beim Anschließen des genannten Kennworts die Grenze von 100 gleichzeitig bekannten Kennwörtern überschritten wird (siehe [Abschnitt „Kennwörter für UDS/SQL-Dateien vergeben“](#)).

Kennwort für die Administration anschließen

ADD ADM=*adm kennwort*

Anschließen des genannten Datenbankadministrator-Kennworts. Damit wird das Datenbankadministrator-Kennwort entweder neu angegeben, falls bei PP ADM-PASS noch kein Kennwort angeschlossen wurde oder geändert, sofern mit dem DAL-Kommando DROP ADM zuvor das aktuelle Datenbankadministrator-Kennwort ausgeschlossen wurde.

Reaktionen in Fehlersituationen

Das ADD ADM-Kommando wird zurückgewiesen, wenn beim DBH bereits ein Datenbankadministrator-Kennwort definiert wurde.

4.4.5 Neue Einträge in die Verteiltabelle bringen (&ADD DISTRIBUTION)

Für UDS-D

```
&ADD DISTRIBUTION,
  {NODE=prozessorname,
    CONF=konfname [ ,DB=dbname] |
    DB=dbname ,SS=subschemaname |
    FILE=dateiname }
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*; er muss in den ersten sieben Zeichen eindeutig sein.
Das achte Zeichen darf nicht '@' sein.
Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

dbname

Name einer Datenbank;
er darf bis zu 17 Zeichen lang sein;
er muss netzweit eindeutig sein.

subschemaname

Name eines Subschemas;
er darf maximal 30 Zeichen lang sein;
er muss in den ersten sechs Zeichen netzweit eindeutig sein.

Es dürfen nur Benutzersubschemas angegeben werden. PRIVACY-AND-IQF-Subschema und COMPILER-Subschema sind in der Verteiltabelle nicht erlaubt.

:catid.

BS2000-Katalogkennung
Hier ist die Angabe von *:catid.* zulässig
(siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#)).

userid

Benutzerkennung, in der *dateiname* steht

dateiname

Name einer Datei mit den Einträgen, um die die bestehende Verteiltabelle erweitert werden soll.

Sie bringen neue Einträge in die Verteiltabelle (siehe [Abschnitt „Aufbau der Verteiltabelle“](#)).

NODE=...,CONF=...

Mit diesen Operanden geben Sie einen NODE-/CONF-Eintrag an.

Falls diese Konfiguration bereits einem anderen Verarbeitungsrechner zugeordnet ist, wird das Kommando abgewiesen. Verwenden Sie dann bitte das DAL-Kommando &CHANGE DISTRIBUTION.

NODE=...,CONF=...,DB=...

Mit diesen Operanden geben Sie einen DB-Eintrag an.

Der DB-Eintrag wird verkettet mit dem dazugehörigen NODE-/CONF-Eintrag, falls dieser schon existiert oder der NODE-/CONF-Eintrag wird neu angelegt, falls der angegebene Eintrag noch nicht existiert. Die beiden neu angelegten Einträge werden miteinander verkettet.

DB=...,SS=...

Mit diesen Operanden geben Sie einen SS-Eintrag an.

Der SS-Eintrag wird verkettet mit dem dazugehörigen DB-Eintrag, falls dieser schon existiert oder der DB-Eintrag wird neu angelegt, falls der angegebene noch nicht existiert. Die beiden neu angelegten Einträge werden miteinander verkettet.

FILE=...

Mit diesem Operanden erweitern Sie die bestehende Verteiltabelle um die Einträge dieser Eingabedatei.

Mit &ADD DISTRIBUTION ändern Sie nur die lokale Verteiltabelle.

Mit diesem DAL-Kommando können Sie keine konfigurationsbezogenen Kennwörter vergeben. Zur Vergabe konfigurationsbezogener Kennwörter siehe [Abschnitt „Konfigurationsbezogener Kennwortschutz“](#) und die Beschreibung des DAL-Kommandos &PWD DISTRIBUTION im Kapitel "[Kennwort vergeben und verändern \(&PWD DISTRIBUTION\)](#)".

i Änderungen in der Verteiltabelle wirken sich erst bei der nächsten READY-Anweisung aus.

Beispiel

DAL-Kommando	Inhalt der Verteiltabelle
&ADD DIS ,NODE=PROZ3 ,CONF=CONFD	PROZ3 / CONFD
&ADD DIS ,NODE=PROZ3 ,CONF=CONFD ,DB=DBD1	<pre> ^ DBD1-----+</pre>
&ADD DIS ,NODE=PROZ4 ,CONF=CONF E ,DB=DBE1	DBE1-->PROZ4 / CONF E
&ADD DIS ,DB=DBE1 ,SS=SSE11	<pre> ^ SSE11----+</pre>
&ADD DIS ,DB=DBF1 ,SS=SSF11	SSF11-->DBF1

4.4.6 Anzahl bearbeiteter Base Interface Blocks ausgeben (%BIB)

```
%BIB
```

Durch das %BIB-Kommando wird die Anzahl der Base Interface Blocks ausgegeben, die in diesem Session-Abschnitt bearbeitet wurden. Die Ausgabe geht auf die Datensichtstation des Datenbankadministrators bzw. der Operatorkonsole.

Das %BIB-Kommando gilt auch für den linked-in DBH.

Zusätzlich wird die Anzahl der COBOL-/CALL-DML-Anweisungen ausgegeben, die an entfernte Konfigurationen zur Bearbeitung weitergeleitet wurden und von dieser Konfiguration für entfernte Anwenderprogramme bearbeitet wurden.

4.4.7 Konfiguration einem anderen Verarbeitungsrechner zuordnen (&CHANGE DISTRIBUTION)

Für UDS-D

```
&CHANGE DISTRIBUTION ,NODE=prozessorname ,CONF=konfname
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*; er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht '@' sein. Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

Sie ändern *prozessorname* im NODE-/CONF-Eintrag der Verteiltabelle. Nach Ausführung des DAL-Kommandos leitet UDS-D DML-Anweisungen, die die Konfiguration *konfname* ansprechen, an den Verarbeitungsrechner *prozessorname*.

Mit &CHANGE DISTRIBUTION ändern Sie nur die lokale Verteiltabelle.



Änderungen in der Verteiltabelle wirken sich erst bei der nächsten READY-Anweisung aus.

4.4.8 Konsistenzpunkte festschreiben (CHECKPOINT)

```
CHECKPOINT[ DB=dbname [ ,OPTION=EVEN-WITHOUT-ALOG ] ]
```

dbname

Name der Datenbank, für die ein Konsistenzpunkt festgeschrieben werden soll.

CHECKPOINT
(ohne DB)

Für alle Datenbanken der Konfiguration sollen Konsistenzpunkte festgeschrieben werden.

OPTION=EVEN-WITHOUT-ALOG

Auf der Datenbank *dbname* wird auch dann ein Konsistenzpunkt festgeschrieben, wenn sie ohne AFIM-Logging betrieben wird. Ohne Angabe von OPTION=EVEN-WITHOUT-ALOG wird für solche Datenbanken das CHECKPOINT-Kommando abgewiesen. Für Datenbanken mit AFIM-Logging ist die Angabe von OPTION=EVEN-WITHOUT-ALOG ohne Wirkung.

Mit dem CHECKPOINT-Kommando wird ein Auftrag zum Festschreiben von Konsistenzpunkten erteilt. Beim Festschreiben von Konsistenzpunkten schließt der DBH je Datenbank die ALOG-Datei ab und eröffnet eine neue ALOG-Datei. Erst nach PERFORM wird der Auftrag ausgeführt.

Alle RETRIEVAL-Transaktionen laufen unbehindert weiter, auch neu hinzukommende. Alle laufenden UPDATE-Transaktionen werden zu Ende geführt. Alle UPDATE-Transaktionen, die während des Checkpoint-Schreibens neu hinzukommen, werden bei COBOL-DML im Allgemeinen mit dem Statuscode 12122 abgewiesen. Mit PP ORDER-DBSTATUS kann dieser Statuscode beeinflusst werden.

Neue SQL-Transaktionen können als RETRIEVAL-Transaktionen ablaufen. SQL-UPDATE-Anweisungen werden abgewiesen (SQL-CODE -810).

Für solche Datenbanken, für die AFIM-Logging eingeschaltet ist und die mit SHARED-RETRIEVAL-Option angeschlossen wurden, wird ein CHECKPOINT-Kommando zurückgewiesen.

Das CHECKPOINT-Kommando kann auch für eine Datenbank gegeben werden, die noch nicht an die DB-Konfiguration angeschlossen ist, für die aber ein Anschließer existiert. Dies ist sinnvoll, damit auch für solche Datenbanken Konsistenzpunkte festgeschrieben werden können, die mit dem DAL-Kommando ADD DB angeschlossen werden. Der DBH-Ladeparameter PP STDCKPT wirkt nicht beim dynamischen Anschließen und Ausschließen von Datenbanken.

i Der DBH kann eine neue ALOG-Datei nur eröffnen, wenn sie vorab eingerichtet wurde mit dem CREATE-FILE-Kommando oder die Datenbank in der Konfigurationskennung liegt.

Wenn es dem DBH nicht gelingt, eine neue ALOG-Datei betriebsbereit zu machen, sperrt der DBH intern die betroffene Datenbank für ändernde Zugriffe (UPDATE), d.h. er lässt nur noch lesende Zugriffe zu.

Die Sperre kann vom Datenbankadministrator **nicht** mit dem DAL-Kommando ACCESS UPDATE aufgehoben werden.

Der Datenbankadministrator kann aber die fehlende ALOG-Datei mit dem CREATE-FILE-Kommando einrichten und erneut ein CHECKPOINT-Kommando und ein PERFORM für die Datenbank geben und damit die interne UPDATE-Sperre wieder aufheben. Siehe auch „ALOG-Datei-Wechsel“ und „ALOG-Datei-Überlauf“ im Kapitel "[Konsistenzpunkte in ALOG-Dateien festschreiben \(STDCKPT\)](#)".

Sind bei der Ausführung eines Auftrages CHECKPOINT ohne Angabe von DB=*dbname* Datenbanken an die Konfiguration angeschlossen, für die kein AFIM-Logging eingeschaltet ist, so wird für jede dieser Datenbanken gemeldet, dass kein Konsistenzpunkt festgeschrieben wurde.

Reaktionen in Fehlersituationen

Das CHECKPOINT-Kommando wird zurückgewiesen,

- wenn die Datenbank mit OPTION=SHARED-RETRIEVAL betrieben wird.
- wenn keine Datenbank mit dem angegebenen Namen vorhanden ist und auch kein Auftrag vorliegt, eine Datenbank mit dem angegebenen Namen anzuschließen.
- wenn eine Datenbank mit dem angegebenen Namen zwar vorhanden ist, diese Datenbank jedoch ohne AFIM-Logging gefahren wird und auch OPTION=EVEN-WITHOUT-ALOG nicht angegeben wurde.

4.4.9 Session oder Administration beenden (CLOSE)

```
CLOSE {RUN-UNITS | CALLS | ADMINISTRATION}
```

Das CLOSE-Kommando beendet eine Session bzw. die DCAM-Administration normal (siehe [Abschnitt „Verbindung zur UDS/SQL-Konfiguration abbauen \(DISCONNECT-CONFIGURATION\)“](#)).

RUN-UNITS

stößt die normale Beendigungsprozedur an:

Jede aktive Transaktion kann normal weiterarbeiten, der DBH lässt aber keine neuen Transaktionen zu (Statuscode 151).

Läuft keine Transaktion mehr, so beendet die Mastertask alle Servertasks.

Mit Ausgabe der folgenden Meldung schließt die Mastertask dann die Session:

```
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK
% UDS0213 UDS NORMAL BEENDET MIT nnnnnnnnnnnnnnnnn DML-STATEMENTS
```

Transaktionen, die sich im STOP-Zustand befinden, werden automatisch aktiviert (siehe „STOP-Kommando“ im Kapitel ["Ausführung von Transaktionen stoppen \(STOP\)"](#)).

Die Anzahl der DML-Anweisungen pro Datenbank und die Anzahl der Ein- und Ausgaben pro Datenbank werden bei folgenden Gelegenheiten nach SYSOUT ausgegeben:

- Wenn die Datenbank ausgeschlossen wird (DROP DB).
- Sobald der DBH beendet wurde.

CALLS

bewirkt eine beschleunigte Beendigung der Session:

Der DBH setzt alle offenen Transaktionen zurück (Statuscode 122, SQL-Code -1700). Der Versuch, eine Transaktion zu beginnen, wird mit Statuscode 151 bzw. 122 zurückgewiesen (SQL-Code -1820 bzw. -1700).

Hinweis für UDS-D

Ein Rollback wird nicht durchgeführt, wenn sich die Transaktion im Zustand PTC befindet. Damit CLOSE CALLS wirksam werden kann, müssen Transaktionen, die im Zustand PTC sind, einzeln beendet werden (siehe [Abschnitt „Zustand PTC beenden“](#)).

ADMINISTRATION

Die laufende Administration über DCAM wird geschlossen. Bei Bedarf kann die Administration über DCAM wieder eröffnet werden.

4.4.10 UDS-D-Betrieb beenden (&CLOSE DISTRIBUTION)

Für UDS-D

```
&CLOSE DISTRIBUTION
```

Sie fordern UDS/SQL auf, den UDS-D-Betrieb zu beenden, d.h. die UDS-D-Task UDSCCT zu beenden. Die UDS-D-Task wird erst beendet, wenn alle sekundären Teiltransaktionen in der lokalen Konfiguration beendet sind oder sich seit einer Zeitspanne im Zustand PTC befinden, die abhängig ist vom Wert des DBH-Ladeparameters PP CHCKTIME (siehe [Abschnitt „UDS-D-Betrieb beenden“](#)).

Wenn Sie &CLOSE DISTRIBUTION angeben, werden READY-Anweisungen von entfernten Anwenderprogrammen abgewiesen und keine verteilten Transaktionen von lokalen Anwendern zugelassen.

Das Anwenderprogramm erhält den Statuscode 141.

Wenn die UDS-D-Task UDSCCT beendet ist, ist der UDS-D-Betrieb in dieser Konfiguration beendet, d.h. es kommen keine entfernten Nachrichten mehr und es werden auch keine Nachrichten mehr an entfernte Konfigurationen geschickt.

4.4.11 Transaktion im Zustand PTC beenden (COMMIT)

Für UDS-D/openUTM

```
COMMIT transaktionskennung
```

transaktionskennung

bezeichnet eine sekundäre Teiltransaktion bzw. eine openUTM-Transaktion im Zustand PTC, die beendet werden soll.

Mit COMMIT beenden Sie eine sekundäre Teiltransaktion bzw. openUTM-Transaktion im Zustand PTC mit Festschreiben der Änderungen (FINISH).

Wenn die angegebene Transaktion nicht im Zustand PTC ist, weist UDS/SQL das Kommando zurück.

Die Wirkung des DBH-Ladeparameters PP PTCSYNCH ist für diese Transaktion aufgehoben.

i Indem Sie eine sekundäre Teiltransaktion bzw. eine openUTM-Transaktion im Zustand PTC beenden, gefährden Sie die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz (siehe [Abschnitt „Zustand PTC beenden“](#)).

4.4.12 DISPLAY SQL-Ausgabe fortsetzen (CONTINUE)

CONTINUE

Mit dieser Funktion können Sie in der Ausgabe des DAL-Kommandos DISPLAY SQL „blättern“. Wenn nicht alle Vorgänge auf einem Bildschirm ausgegeben werden konnten, weil die Menge zu groß war, können Sie die Ausgabe mit CONTINUE fortsetzen.

Es ist zu berücksichtigen, dass immer der aktuelle Stand ausgegeben wird. Das bedeutet, dass sich die Werte zwischen dem Eingeben des DISPLAY SQL-Kommandos und des CONTINUE-Kommandos verändern können.

Ist zum Beispiel in der Zwischenzeit ein Vorgang neu eröffnet worden, wird beim CONTINUE einer mehr als beim DISPLAY SQL-Kommando ausgegeben.

CONTINUE bezieht sich immer auf das zuletzt eingegebene „blätterfähige“ DISPLAY SQL-Kommando, auch wenn inzwischen andere DAL-Kommandos eingegeben worden sind.

Ausgabeformate

Die Ausgabe des CONTINUE-Kommandos ist bei den verschiedenen DISPLAY SQL-Kommandos beschrieben.

4.4.13 Online-Erweiterbarkeit deaktivieren (DEACT)

```
DEACT { DBTT-INCR , DB=dbname [ , RECR=recordref ] |
       INCR , DB=dbname [ , RR=realmref ] }
```

i Zwischen DEACT und DBTT-INCR bzw. INCR ist genau ein Leerzeichen anzugeben. Weitere Leerzeichen sind in diesem DAL-Kommando nicht erlaubt.

Mit dem Kommando DEACT können Sie die Online-Erweiterbarkeit für DBTTs und Realms deaktivieren.

Online-DBTT-Erweiterbarkeit für eine Satzart deaktivieren (DEACT DBTT-INCR)

dbname Name der Datenbank, zu der die Satzart bzw. die Satzarten gehören, für die die Online-DBTT-Erweiterung deaktiviert werden soll. Die Datenbank und der entsprechende Realm müssen zum Zeitpunkt des DAL-Kommandos bereits zugeschaltet sein.

recordref

Nummer der Usersatzart, für die die Online-DBTT-Erweiterung deaktiviert werden soll. Soll sie für alle Satzarten deaktiviert werden, kann diese Angabe entfallen.

Die *recordrefs* der Satzarten können Sie mit dem DAL-Kommando DISPLAY DBTT-INCR anzeigen lassen oder Sie können sie dem BPSIA-Protokoll entnehmen.

recordref=1 wird UDS/SQL-intern genutzt und kann hier nicht angegeben werden.

Mit dem Kommando DEACT DBTT-INCR wird ein Auftrag zur Deaktivierung der Online-DBTT-Erweiterung einer oder mehrerer Satzarten vorgemerkt oder ein vorhergehender ACT DBTT-INCR-Auftrag aufgehoben.

Der Auftrag wird erst durchgeführt, nachdem das Kommando PERFORM angegeben wurde. Sie werden mit der Meldung UDS0722 folgendermaßen über die Ausführung informiert:

- Wurde ein Deaktivierungsauftrag für sämtliche Satzarten ausgeführt, so wird eine einzige Meldung für die Datenbank ausgegeben.
- Wurde ein Deaktivierungsauftrag für eine einzelne Satzart ausgeführt, so wird eine satzartspezifische Meldung ausgegeben.
- Wurden sowohl Deaktivierungsaufträge für einzelne Satzarten als auch für sämtliche Satzarten gegeben, so wird nur eine einzige Meldung für die Datenbank ausgegeben.

Falls bei der Deaktivierung der Online-DBTT-Erweiterung für alle Satzarten einer Datenbank ein Realm mit DBTTs nicht zugeschaltet ist, unterbleibt die Deaktivierung für die entsprechenden Satzarten. Meldung UDS0745 informiert über den betroffenen Realm.

Mit DEACT DBTT-INCR, wird die in einem vorangegangenen ACT DBTT-INCR EXT=*extr-mbr* festgelegte Erweiterungsgröße zurückgesetzt.

Das Kommando DEACT DBTT-INCR wird sofort mit Meldung UDS0209 zurückgewiesen

- wenn die angegebene Satzartnummer nicht existiert.
- wenn die spezifizierte Datenbank nicht zugeschaltet oder im Modus SHARED-RETRIEVAL zugeschaltet ist.

- wenn bei einer angegebenen Satzart der entsprechende DBTT-Realm nicht zugeschaltet ist. Meldung UDS0745 informiert über den betroffenen Realm.
- wenn ein Auftrag zur Deaktivierung der Online-DBTT-Erweiterung für alle Satzarten einer Datenbank bei keiner Satzart zu einer Deaktivierung oder zur Rücknahme eines gegenteiligen Auftrags führt.

Online-Erweiterbarkeit für einen Realm deaktivieren (DEACT INCR)

dbname

Name der Datenbank des Realms bzw. der Realms, für den bzw. für die Online-Erweiterbarkeit deaktiviert werden soll.

Die Datenbank muss zum Zeitpunkt der Kommandoausführung zugeschaltet sein.

Die Datenbank darf sich nicht im Modus SHARED-RETRIEVAL befinden.

realmref

Nummer des Realms, für den die Online-Erweiterbarkeit deaktiviert werden soll. Der Realm muss zum Zeitpunkt der Kommandoausführung zugeschaltet sein. Die Nummern der zugeschalteten Realms können Sie sich mit DISPLAY INCR anzeigen lassen.

Zulässig sind die Nummern aller Benutzerrealms und des DBDIR. Die Angabe der Nummer des DBCOM ist nicht zulässig.

Mit dem Kommando DEACT INCR können Sie für den Realm *realmref* der Datenbank *dbname* einen Auftrag zur Deaktivierung der Online-Erweiterbarkeit vormerken. Ist *RR=realmref* nicht angeben, so wird das Kommando für alle gerade zugeschalteten Benutzerrealms und den DBDIR der Datenbank *dbname* ausgeführt.

Mit DEACT INCR können Sie aber auch ein zuvor eingegebenes Kommando ACT INCR widerrufen, solange die Kommandos noch nicht mit PERFORM zur Durchführung angestoßen sind. Grundsätzlich wird DEACT INCR erst ausgeführt, nachdem PERFORM angegeben wurde.

Die eventuell zusätzlich zu DEACT INCR vorgemerkten DAL-Aufträge dürfen auch Aufträge zum Abschalten der betroffenen Realms bzw. Datenbanken enthalten. Dabei ist es gleichgültig, ob die DROP-Aufträge vor oder nach den DEACT INCR-Aufträgen eingegeben werden. Bei PERFORM werden in jedem Fall zuerst die DEACT INCR-Aufträge bezüglich einer Datenbank bearbeitet. Zur Ausführung vorgemerakter DAL-Kommandos siehe auch „Ausführung der Kommandos“ im [Abschnitt „Die Datenbankadministratorsprache DAL“](#).

Falls der oder die angegebenen Realms nicht online erweiterbar ist/sind, wird DEACT INCR ignoriert.

Anstehende EXTEND-DBTT-Aufträge, die noch nicht ausgeführt sind, weil seit dem Absetzen des Auftrags noch kein Satz eingespeichert worden ist, werden bei der Ausführung von DEACT INCR ohne zusätzliche Meldung annulliert.

Das Kommando DEACT INCR wird sofort mit Meldung UDS0209 zurückgewiesen,

- wenn die Datenbank *dbname* nicht zugeschaltet oder im Modus SHARED-RETRIEVAL zugeschaltet ist.
- wenn die angegebene Realm-Nummer *realmref* nicht existiert.
- wenn als Realm-Nummer *realmref* diejenige des DBCOM angegeben wurde.
- wenn der angegebene Realm nicht zugeschaltet ist.
- wenn noch mindestens eine Online-DBTT-Erweiterung auf dem Realm aktiviert ist oder bereits ein Auftrag zur Online-DBTT-Erweiterung für den Realm vorgemerkt ist. Meldung UDS0738 informiert über den betroffenen Realm.

4.4.14 Informationen zur DB-Konfiguration auflisten (DISPLAY)

```

DISPLAY {DB[ ,RUNID=transaktionskennung] |
        USERS[ ,DB=dbname] |
        SUBSCH[ ,DB=dbname][ ,LINES={n | ALL}] |
        MAINREF[ ,STATE=BLOCK] |
        transaktionskennung |
        REALMS[ ,DB=dbname][ ,RN=realmname][ ,LINES={n | ALL}] |
        PP |
        FPA [ ,DB=dbname][ ,RN=realmname][ ,LINES={n | ALL}] |
        INCR [ ,DB=dbname][ ,RN=realmname][ ,LINES={n | ALL}] |
        DBTT-INCR [ ,DB=dbname[ , RECR=recordref]][ ,LINES={n | ALL}] |
        PUBSETS |
        ALOG [ ,DB=dbname] }

```

transaktionskennung

Kennung einer aktuellen Transaktion

dbname

Name einer Datenbank der DB-Konfiguration (auch Name einer Schattendatenbank).

Bei DISPLAY DBTT-INCR Name der Datenbank, zu der die Satzart bzw. die Satzarten gehören, für die die Informationen über das Userschema ausgegeben werden sollen. Wird der Parameter DB nicht angegeben, so werden alle Satzarten aller aktuell zugeschalteten Datenbanken ausgegeben.

realmname

Name eines Realms

recordref

Nummer der Usersatzart, für die die Informationen ausgegeben werden sollen. Wird der Parameter RECR nicht angegeben, so werden alle Satzarten des Userschemas der angegebenen Datenbank ausgegeben.

Das Kommando DISPLAY DBTT-INCR ohne Angabe von *recordref* kann insbesondere dafür genutzt werden, um die Satznummern den Satzartnamen zuzuordnen.

recordref≠1 wird UDS/SQL-intern genutzt und kann hier nicht angegeben werden.

n

Maximalzahl der Ausgabezeilen für dieses DISPLAY-Kommando (Voreinstellung: 100). Für *n* können Werte zwischen 5 und 999 angegeben werden. Durch die Begrenzung der Anzahl der Ausgabezeilen können Sie Behinderungen des DBH-Betriebes bei unbeabsichtigter Erzeugung sehr vieler Ausgabezeilen vermeiden.

ALL

Alle Ausgabezeilen werden erzeugt.

Das DISPLAY-Kommando protokolliert auf der Datensichtstation des Datenbankadministrators bzw. auf der Operatorkonsole bei Angabe von

DB

alle Datenbanken der aktuellen DB-Konfiguration

DB,RUNID=*transaktionskennung*

alle Datenbanken, die von der Transaktion *transaktionskennung* angesprochen werden

USERS

alle Transaktionen, die derzeit in der DB-Konfiguration arbeiten

USERS,DB=*dbname*

alle Transaktionen, die die Datenbank *dbname* ansprechen

SUBSCH

alle Subschemas der DB-Konfiguration

SUBSCH,DB=*dbname*

alle Subschemas der Datenbank *dbname*

MAINREF

Informationen über den Zustand der Transaktionskanäle (Mainrefs)

MAINREF,STATE=BLOCK

Informationen zu Mainrefs, die blockiert sind

transaktionskennung

Information zur Transaktion *transaktionskennung*

REALMS

Information über alle permanenten Realms der DB-Konfiguration

REALMS,DB=*dbname*

Information über alle permanenten Realms der Datenbank *dbname*

REALMS,RN=*realmname*

[,DB=*dbname*]

Information über den permanenten Realm *realmname*
[der Datenbank *dbname*]

PP

alle Ladeparameter, mit denen der DBH zurzeit läuft

FPA

Information über freie Seiten in allen Realms in der DB-Konfiguration

FPA,DB=*dbname*

Information über freie Seiten in den Realms der Datenbank *dbname*

FPA,RN=*realmname*

[,DB=*dbname*]

Information über freie Seiten des Realms *realmname*
[der Datenbank *dbname*]

INCR

Information über die Online-Erweiterbarkeit aller zugeschalteten Benutzerrealms sowie aller DBDIRS in der DB-Konfiguration

INCR,DB=*dbname*

Information über die Online-Erweiterbarkeit aller zugeschalteten Benutzerrealms sowie des DBDIR der Datenbank *dbname*

INCR,RN=*realmname*

[,DB=*dbname*]

Information über die Online-Erweiterbarkeit des Realms *realmname* der Datenbank *dbname*. Falls *realmname* in der DB-Konfiguration eindeutig ist, kann die Angabe von DB=*dbname* entfallen.

DBTT-INCR

Information über die Online-DBTT-Erweiterbarkeit der Satzarten der User-Schemata in der DB-Konfiguration

DBTT-INCR,DB=*dbname*

Information über die Online-DBTT-Erweiterbarkeit der Satzarten des User-Schemas der Datenbank *dbname*

DBTT-INCR,DB=*dbname*,RECR=*recordref*

Information über die Online-DBTT-Erweiterbarkeit der Satzart mit der Satzartnummer *recordref* der Datenbank *dbname*

PUBSETS

Einstellungen der UDS/SQL-Pubsetdeklaration

ALOG

Aktuelle ALOG-Einstellungen für alle ALOG-Dateien der DB-Konfiguration.

ALOG DB=*dbname*

Aktuelle ALOG-Einstellungen Der Datenbank *dbname*.

LINES

Anzahl der Ausgabezeilen für dieses DISPLAY-Kommando.

LINES=*n*

maximale Anzahl der Ausgabezeilen für dieses DISPLAY-Kommando

Falls die Anzahl der maximalen Ausgabezeilen für ein DISPLAY-Kommando nicht ausreicht, um die Information für alle betroffenen Objekte auszugeben, wird ein entsprechender Hinweis ausgegeben:

```
OUTPUT OF nnnn FURTHER LINES SUPPRESSED
```

Es gibt kein Kommando, um dieses abgebrochene DISPLAY-Kommando fortzusetzen. Um die Information für alle betroffenen Objekte zu erhalten, muss das Kommando z.B. mit `LINES=ALL` wiederholt werden.

Abschließend wird in jedem Fall die folgende Meldung ausgegeben:

```
UDS0218 UDS HAT DIE BEARBEITUNG DES DAL KOMMANDOS ABGESCHLOSSEN
```

`LINES=ALL`

alle Ausgabezeilen für dieses DISPLAY-Kommando

Im Folgenden ist beschrieben, was die verschiedenen DISPLAY-Kommandos ausgeben.

4.4.14.1 DISPLAY DB - Ausgabe aller Datenbanken und Zugriffsarten

Eingabeformat: DISPLAY DB

Kommentarzeile: USERID: DATABASE-NAME: DB-OPTION: BUF-SIZE: BUF-ID: PAGE:

Informations-

zeile(n): *userid dbname db-option bbbb buffid pgKB*

Endzeile: UDS/SQL *Vn.n nnn* DATABASE(S) IN *konfname*

Erklärungen:

userid Kennung, in der die betroffene Datenbank steht.

dbname

Name der Datenbank

db-option

EXC-UPD

EXCLUSIVE-UPDATE:

Der DBH greift exklusiv auf die Datenbank *dbname* zu. Die Anwenderprogramme können ändernd auf die Datenbank zugreifen.

SHA-RTR

SHARED-RETRIEVAL:

Alle DBHs können auf die Datenbank *dbname* shared zugreifen, d.h. auch an andere DBHs kann diese Datenbank angeschlossen werden (aber auch nur SHA-RTR). Die Anwenderprogramme aller DBHs können nur lesend auf der Datenbank arbeiten.

bbbb Größe des User Buffer Pool der Datenbank in Mbyte ohne führende Nullen. Wenn kein User Buffer Pool vorhanden ist, werden Leerzeichen ausgegeben.

buffid Name des Shared User Buffer Pools
Falls der Datenbank ein Shared User Buffer Pool zugewiesen ist.

*EXCL

zeigt an, dass der Datenbank ein exklusiver User Buffer Pool zugewiesen ist.

*SYSTEM

zeigt an, dass der Datenbank der System Buffer Pool zugewiesen ist.

pg Seitenformat der Realm-Dateien

Vn.n UDS/SQL-Versionsnummer

nnn Anzahl der Datenbanken der DB-Konfiguration

*Beispiel***//EXECUTE-DAL-CMD CMD=DISPLAY DB**

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY DB (OPOX073,11:27:28/0YA4)
0YA4: USERID:    DATABASE-NAME:    DB-OPTION: BUF-SIZE: BUF-ID: PAGE:
0YA4: -----
0YA4: $XXXXXXXX.VERSAND             EXC-UPD             *SYSTEM  4KB
0YA4: $XXXXXXXX.KUNDEN             EXC-UPD             1      *EXCL   4KB
0YA4:
0YA4: UDS/SQL V2.9    2 DATABASES IN CONFBSPL
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:28/0YA4)
```

4.4.14.2 DISPLAY DB,RUNID=transaktionskennung - Ausgabe aller Datenbanken, die durch die Transaktionskennung angesprochen werden

Eingabeformat: DISPLAY DB,RUNID=*transaktionskennung*

Kommentarzeile: USERID: DATABASE-NAME: OPEN-MODE:

Informations-
zeile(n): *userid* *dbname* *open-mode*

Endzeile: UDS/SQL V*n.n* *nnn* DATABASE(S) USED FROM *transaktionskennung*

Erklärungen:

transaktionskennung

Interne von UDS/SQL vergebene Transaktionskennung.

userid Kennung, in der die betroffene Datenbank steht.

dbname

Name der Datenbank

open-mode

Benutzungsart, mit der die Transaktion *transaktionskennung* den (die) Realm(s) der Datenbank eröffnet hat:

UPD

UPDATE

RTR

RETRIEVAL

V*n* . UDS/SQL-Versionsnummer

n

nnn Anzahl der Datenbanken, die die Transaktion *transaktionskennung* anspricht

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY DB,RUNID=1

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY DB,RUNID=1 (OPOX073,11:27:30
/OYA4)
OYA4:  USERID:        DATABASE-NAME:                    OPEN-MODE:
OYA4:  -----
OYA4:  $XXXXXXXXX.VERSAND                            RTR
OYA4:
OYA4:  UDS/SQL V2.9    1 DATABASE    USED FROM                    1
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN.   (OPCC074,11:27:30/OYA4)
```

4.4.14.3 DISPLAY USERS - Ausgabe aller Transaktionen, die derzeit arbeiten

Eingabeformat: DISPLAY USERS

Kommentarzeile: PROGRAM TSN/TERM RUNUNIT-ID STATE PTC FC MR-NR DLY

Informations-

zeile(n): *programe tsn/term ta-kennung state ptc fc mr-nr dly*

Endzeile: UDS/SQL V*n.n* *nnn* USER(S) OF *konfname*

Erklärungen:

PROGRAM

Anwenderprogramm, das die Transaktion gestartet hat:

programe

Name des Anwenderprogramms oder des openUTM-Operanden DBKEY bzw. 'UTM'

TSN/TERM

Prozesskennzeichen:

tsn

bei Teilnehmerbetrieb: Prozessfolgenummer, unter der das Programm *programe* läuft

term-name

bei Teilhaberbetrieb: logischer Name des openUTM-Clients oder Partners (LTERM-, LPAP-, OSI-LPAP-Name)

Bei verteilter TA bezieht sich diese Angabe auf deren primäre Teiltransaktion (PTT), die möglicherweise in einer anderen Konfiguration abläuft.

RUNUNIT-ID

ta-kennung

Interne von UDS/SQL vergebene Transaktionskennung.

Die Transaktionen werden innerhalb eines Session-Abschnitts fortlaufend nummeriert.

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

STATE

state

gibt den gegenwärtigen Zustand der Transaktion an:

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

STOP

die Transaktion befindet sich im STOP-Zustand

AREA

die Transaktion befindet sich bei der Ausführung einer READY-Anweisung in der Realm-Warteschlange; sie kann zur Zeit auf einen angeforderten Realm nicht zugreifen

LOCK

die Transaktion wartet darauf, dass ein gesperrtes Betriebsmittel, z.B. eine Seite freigegeben wird

DEACT

die Transaktion ist deaktiviert

WAIT-ST

die Transaktion wartet auf eine freie Servertask

TO-STOP

für die Transaktion liegt ein STOP-Auftrag vor

TO-CANC

für die Transaktion liegt ein CANCEL-Auftrag vor

IN-CANC

für die Transaktion wird gerade ein CANCEL-Auftrag bearbeitet

USER

für diese Transaktion wird gegenwärtig keine DML-Anweisung bearbeitet, die Kontrolle ist beim Anwenderprogramm bzw. bei UDS-D in der UDS/SQL-Konfiguration der PTT.

CONNEC

die Kontrolle über die Transaktion ist im Verbindungsmodul bzw. bei UDS-D in der UDS/SQL-Konfiguration einer STT.

DBH

für die Transaktion wird gegenwärtig vom DBH eine DML-Anweisung bearbeitet; die Kontrolle ist beim DBH.

PTC *ptc*

YES

Die verteilte Transaktion ist im Zustand Prepared to Commit (PTC); d.h. der FINISH der Transaktion ist eingeleitet, aber noch nicht abgeschlossen.

-

Die verteilte Transaktion ist nicht im Zustand Prepared to Commit (PTC).

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt oder um eine durch openUTM verteilte Transaktion.

FC *funktionscode*

Funktionscode der DML-Anweisung, die derzeit bearbeitet wird bzw. als letzte bearbeitet wurde (siehe [Abschnitt „Funktionscodes der DML-Anweisungen“](#)).

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

MR-NR *mainrefnr*

vom DBH vergebene interne Nummer, die beim READY der Transaktion zugeordnet wird; sie ist im Gegensatz zu *transaktionskennung* nicht fortlaufend (Wertebereich: 1 bis Wert von PP TRANSACTION)

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

DLY *dly*

YES

Die Transaktion behindert die Ausführung von vorgemerkten Aufträgen (DLY entspricht DELAYING), d. h. sie muss vor der Durchführung der Aufträge beendet sein (siehe DAL-Kommando „PERFORM“ im Kapitel "[Vorgemerkte Aufträge ausführen \(PERFORM\)](#)" und „RESET ORDERS“ im Kapitel "[Vorgemerkte Aufträge zurücknehmen \(RESET ORDERS\)](#)").

-

Die Transaktion behindert die Ausführung von vorgemerkten Aufträgen nicht.

V*n.n* UDS/SQL-Versionsnummer

nnn Anzahl der derzeit offenen Transaktionen

4.4.14.4 DISPLAY USERS,DB=dbname - Ausgabe aller Transaktionen, die derzeit die Datenbank ansprechen

Eingabeformat: DISPLAY USERS,DB=*dbname*

Ausgabe wie bei Ausgabeformat DISPLAY USERS für die Anwenderprogramme, die die Datenbank *dbname* mit READY eröffnet haben.

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY USERS,DB=VERSAND

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY USERS,DB=VERSAND
(OPOX073,11:27:32/0YA4)
0YA4: PROGRAM   TSN/TERM      RUNUNIT-ID  STATE    PTC  FC   MR-NR  DLY
0YA4: -----
0YA4: DMLTEST   TSN 0YA7                1   USER    -   18    1    -
0YA4:
0YA4: UDS/SQL V2.9      1 USER  OF VERSAND
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:32/0YA4)
```

4.4.14.5 DISPLAY SUBSCH - Ausgabe aller Subschemata

Eingabeformat: DISPLAY SUBSCH

Kopfzeile: *n* SUBSCHEMA(S) IN *konfname*

Informationszeilen:

Datenbankzeile: DB = *dbname*

Schemazeile: SCHEMA = *schemaname*

Subschemazeile(n) *subschemaname* [-NAME AMBIGUOUS]

Erklärungen:

n Anzahl der Subschemata dieser DB-Konfiguration

konfname

die ersten acht Zeichen des Konfigurationsnamens

Datenbankzeile:

dbname

Datenbankname

Schemazeile:

schemaname

Schemaname

Subschemazeile(n):

für jedes Subschemata der Datenbank *dbname* eine Zeile. Sie enthält:

subschemaname

Subschemaname (max. 30 Zeichen)

NAME AMBIGUOUS

Hinweis bei allen Subschemanamen, die sich innerhalb der aktuellen DB-Konfiguration in den ersten sechs Zeichen nicht unterscheiden. Ein READY auf ein solches Subschemata würde vom DBH mit dem Statuscode 141 abgewiesen werden.

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY SUBSCH

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY SUBSCH (OPOX073,11:27:34
/OYA4)
OYA4: 2 SUBSCHEMAS IN CONFBSPL
OYA4: =====
OYA4: DB      = VERSAND
OYA4: SCHEMA = ARTIKELVERSAND
OYA4: -----
OYA4: ADMIN
OYA4: =====
OYA4: DB      = KUNDEN
OYA4: SCHEMA = KUNDENKARTEI
OYA4: -----
OYA4: VERWALTUNG
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:34/OYA4)
```

4.4.14.6 DISPLAY SUBSCH,DB=dbname - Ausgabe aller Subschemas der Datenbank

Eingabeformat: DISPLAY SUBSCH,DB=*dbname*

Kopfzeile: *n* SUBSCHEMA(S) OF *dbname*

Informationszeile(n):

Schemazeile: SCHEMA = *schemaname*

Subschemazeile *subschemaname*

n Anzahl der Subschemas der Datenbank *dbname*

Schemazeile:

schemaname

Schemaname

Subschemazeile:

subschemaname

Subschemaname (max. 30 Zeichen)

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY SUBSCH,DB=VERSAND

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY SUBSCH,DB=VERSAND
(OPOX073,11:27:36/0YA4)
0YA4: 1 SUBSCHEMA OF VERSAND
0YA4: SCHEMA = ARTIKELVERSAND
0YA4: -----
0YA4: ADMIN
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:36/0YA4)
```

4.4.14.7 DISPLAY MAINREF - Ausgabe von Informationen über den Zustand der Mainrefs

Eingabeformat: DISPLAY MAINREF

Kommentarzeile:	MR-NR	STATE	RUNUNIT-ID	TEMP
Informationszeilen:	<i>mainrefnr</i>	<i>state</i>	<i>transaktionskennung</i>	<i>temp</i>

MR-NR

mainref

vom DBH vergebene interne Nummer, die beim READY der Transaktion zugeordnet wird; sie ist im Gegensatz zu *transaktionskennung* nicht fortlaufend.

Wertebereich 1 bis Wert von PP TRANSACTION

STATE

state

gibt den gegenwärtigen Zustand der Mainref an

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

FREE

die Mainref ist von keiner Transaktion belegt und steht für eine neue Transaktion zur Verfügung.

STOP-NEW

die Mainref ist zurzeit gegen die Belegung durch neue Transaktionen gesperrt:

- durch ein DAL-Kommando oder
- durch ein DBH-internes Kommando

STOP

die Mainref ist von einer Transaktion mit dem Status STOP belegt

USED

die Mainref ist durch eine Transaktion belegt

BLOCK

die Mainref ist wegen eines Defekts ihres Temporären Realms für den aktuellen Session-Abschnitt gegen die Belegung durch neue Transaktionen gesperrt

Die STATE-Werte schließen sich gegenseitig aus. Sie werden nach folgender Hierarchie angezeigt:

BLOCK, STOP, STOP-NEW, FREE, USED

RUNUNIT-ID

transaktionskennung

Interne, vom UDS/SQL vergebene Transaktionskennung.

Die Transaktionen werden innerhalb eines Session-Abschnitts fortlaufend nummeriert.

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

TEMP

temp

zeigt Defekte im Temporären Realm dieser Mainref an

-

kein Defekt im Temporären Realm dieser Mainref

DSTR

Temporärer Realm dieser Mainref ausgefallen

Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal, ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

Bei permanenten Fehlern in einem Temporären Realm reagiert der DBH mit Blockieren der entsprechenden Mainref.

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY MAINREF

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY MAINREF (OPOX073,11:27:36
/OYA4)
OYA4: MR-NR    STATE      RUNUNIT-ID  TEMP
OYA4: -----
OYA4:   1     USED           1          -
OYA4:   2     FREE           -          -
OYA4:   3     FREE           -          -
OYA4:   4     FREE           -          -
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:38/OYA4)
```

4.4.14.8 DISPLAY MAINREF,STATE=BLOCK - Ausgabe von Informationen über Mainrefs, die blockiert sind

Eingabeformat: DISPLAY MAINREF,STATE=BLOCK

Kommentarzeile:	MR-NR	STATE	RUNUNIT-ID	TEMP
Informationszeile(n):	<i>mainrefnr</i>	<i>block</i>	<i>transaktionskennung</i>	<i>temp</i>

Falls keine Mainref blockiert ist:

Informationszeile: NO MAINREF IS BLOCKED

Erklärungen:

Nur solche Mainrefs werden aufgelistet, die gegen eine Neubelegung von Transaktionen gesperrt, d.h. blockiert sind. Ansonsten wie DISPLAY MAINREF.

Beispiel

```
//EXECUTE-DAL-CMD CMD=DISPLAY MAINREF,STATE=BLOCK
```

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY MAINREF,STATE=BLOCK
(OPOX073,11:27:38/0YA4)
0YA4: NO MAINREF IS BLOCKED
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:38/0YA4)
```

4.4.14.9 DISPLAY transaktionskennung - Ausgabe von Informationen über die Transaktionen

Eingabeformat: DISPLAY *transaktionskennung*

Kommentar	Information	Erklärung
PROGRAM-NAME :	<i>progname</i>	Name des Anwenderprogramms oder des openUTM-Operanden DBKEY bzw. 'UTM'. Bei verteilter TA bezieht sich diese Angabe auf ein Anwenderprogramm, das möglicherweise in einer anderen Konfiguration läuft.
[UTM-TAC / -USER-ID :	<i>utm-tac / utm-kennung</i>	openUTM-Transaktionskennzeichen /openUTM-Benutzerkennung aus dem KDCSIGN-Kommando von openUTM. Bei verteilter TA bezieht sich diese Angabe auf deren primäre Teiltransaktion
TSN :	<i>tsn</i>	bei Teilnehmerbetrieb: Prozessfolgennummer, unter der das Anwenderprogramm läuft. Bei verteilter TA bezieht sich diese Angabe auf deren PTT.
TERMINAL-NAME :	<i>term-name</i>	bei Teilhaberbetrieb: Logischer Name des openUTM-Clients oder Partners (LTERM-, LPAP-, OSI-LPAP-Name). Bei verteilter TA bezieht sich diese Angabe auf deren PTT. Der Terminal ist möglicherweise an einen anderen Rechner angeschlossen
RUNUNIT-ID / - STATE :	<i>transaktionskennung/</i> { STOP AREA LOCK DEACT WAIT-ST TO-STOP TO-CANC IN-CANC USER CONNEC DBH }	Transaktionskennung; wird intern von UDS /SQL vergeben. Bedeutung der Stati wie bei DISPLAY USERS. Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

MAINREF-NR / - STATE :	<i>mainref/</i> { STOP-NEW STOP USED BLOCK }	Nummer der Mainref; wird beim READY der Transaktion zugeordnet. Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal ob es sich um eine PTT oder STT handelt. STOP-NEW, STOP, BLOCK, USED, siehe DISPLAY MAINREF
FC :	<i>funktionscode</i>	Funktionscode der aktuellen bzw. letzten DML-Anweisung der Transaktion (siehe Abschnitt „Funktionscodes der DML-Anweisungen“). Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt
PTT-PROC-NAME :	<i>rechnername</i>	Name des Verarbeitungsrechners, auf dem die primäre Teiltransaktion (PTT) läuft.
PTT-CONF-NAME :	<i>konfname</i>	Die ersten 8 Zeichen des Konfigurationsnamens der primären Teiltransaktion (PTT).
PTT-RLOG-DATE :	<i>rlogzeitstempel</i>	Zeitstempel, der das Erstellungsdatum der zugehörigen RLOG-Datei darstellt. Diese Angabe bezieht sich auf die RLOG-Datei der primären Teiltransaktion (PTT).
PTT-RUNUNIT-ID :	<i>ptt-kennung</i>	Transaktionskennung der zugehörigen primären Teiltransaktion.
LOCAL-TT-STATE :	{ PTT STT } [IN PTC]	Die lokale Teiltransaktion ist PTT bzw. STT einer verteilten Transaktion. IN PTC: ist im Zustand Prepared to Commit (PTC), d.h. der konfigurationsübergreifende FINISH ist eingeleitet, aber noch nicht abgeschlossen.
[UTM-D-TA :	IN PTC]	Die openUTM-Transaktion ist im PTC.
TEMP :	DSTR	Der Temporäre Realm dieser Mainref ist ausgefallen. Bei verteilter TA bezieht sich diese Angabe auf die lokale Teiltransaktion, egal ob es sich um eine primäre oder sekundäre Teiltransaktion (PTT oder STT) handelt.

Tabelle 16: Eingabeformat für die Ausgabe von Informationen über die Transaktion

*Beispiel***//EXECUTE-DAL-CMD CMD=DISPLAY 1**

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY 1 (OPOX073,11:27:40/0YA4)
0YA4: PROGRAM-NAME:          DMLTEST
0YA4: TSN:                   0YA7
0YA4: RUNUNIT-ID / -STATE:   1 / USER
0YA4: MAINREF-NR / -STATE:   1 / USED
0YA4: FC:                    18
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:40/0YA4)
```

4.4.14.10 DISPLAY REALMS - Ausgabe von Informationen über alle permanenten Realms

Eingabeformat: DISPLAY REALMS

Kommentarzeile:		DATABASE	ACC-USER	ACC-SYS	DB-OPT	DESTR
		(USERID: <i>userid</i>)				
Informations-						
zeile(n):		<i>dbname</i>	<i>acc-user</i>	<i>acc-sys</i>	<i>db-opt</i>	<i>destr</i>
Kommentar-	CATID	REALM	ACC-USER	ACC-SYS	STATE	DEFECT
zeile:						
Informations-	<i>katalog-</i>	<i>realm-</i>	<i>acc-user</i>	<i>acc-sys</i>	<i>state</i>	<i>defect</i>
zeile(n):	<i>kennung</i>	<i>name</i>				

Erklärungen:

Realms werden in bestimmten Fällen vom DBH abgeschaltet. Der Datenbankadministrator kann sich über den Zustand der Realms informieren.

userid Kennung, in der die betroffene Datenbank steht.

DATABASE

dbname
Name der Datenbank

CATID

katalogkennung
BS2000-Katalogkennung des Realms

REALM

realmname
Name des Realms

ACC-USER

acc-user
zeigt den Zugriffsmodus an, der mit dem ACCESS-Kommando vom Datenbankadministrator erlaubt wurde

UPD
lesende und schreibende Zugriffe erlaubt

RTR
nur lesende Zugriffe erlaubt

LOCK
weder lesende noch schreibende Zugriffe erlaubt

ACC-SYS

acc-sys

zeigt den Zugriffsmodus an, der vom DBH intern erlaubt wird und nicht durch das ACCESS-Kommando beeinflusst werden kann

UPD

lesende und schreibende Zugriffe erlaubt

RTR

nur lesende Zugriffe erlaubt

LOCK

weder lesende noch schreibende Zugriffe erlaubt

DB-OPT

db-opt

zeigt den Zugriff auf die Datenbank an.

EXC-UPD

EXCLUSIVE UPDATE

Der DBH greift exklusiv auf die Datenbank *dbname* zu.

Die Anwenderprogramme können ändernd auf die Datenbank zugreifen.

SHA-RTR

SHARED RETRIEVAL

Alle DBHs können auf die Datenbank *dbname* shared zugreifen, d.h. auch zu anderen DBHs kann diese Datenbank angeschlossen werden (aber auch nur SHA-RTR). Die Anwenderprogramme aller DBHs können nur lesend auf der Datenbank arbeiten.

STATE

state

zeigt den Status des Realms an.

TMPRLM

der Realm ist temporär

TO DROP

für diesen Realm liegt ein Auftrag zum Ausschließen vor, der noch nicht (vollständig) bearbeitet ist.

SWTOFF

der Realm ist ausgeschlossen worden entweder durch das DAL-Kommando DROP RN =, die BMEND-Funktion REMOVE-REALM oder die Fehlerbehandlung des DBH

TO ADD

für diesen Realm liegt ein Wiederanschließauftrag vor, der noch nicht (vollständig) bearbeitet ist

OPENED

der Realm ist physisch eröffnet, er nimmt an der Session teil

DESTR

destr

zeigt an, ob die Datenbank defekt ist oder nicht (DESTR=DESTROYED)

- YES die Datenbank ist defekt
- die Datenbank ist nicht defekt

DEFECT

defect

zeigt an, ob der Realm im DBH als defekt markiert ist

- YES der Realm ist defekt
- der Realm ist nicht defekt

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY REALMS

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY REALMS (OPOX073,11:27:40
/OYA4)
OYA4:          DATABASE (USERID: $XXXXXXXXX.)  ACC-USER  ACC-SYS  DB-OPT  DESTR
OYA4: -----
OYA4:          VERSAND                        UPD        UPD        EXC-UPD  -
OYA4: =====
OYA4: CATID   REALM                          ACC-USER  ACC-SYS  STATE   DEFECT
OYA4: -----
OYA4: :SQL2:  DATABASE-DIRECTORY              UPD        UPD        OPENED   -
OYA4: :SQL2:  AUFTRAGSRLM                     UPD        UPD        OPENED   -
OYA4: :SQL2:  BESTELLRLM                     UPD        UPD        OPENED   -
OYA4: :SQL2:  KLEIDUNG                        UPD        UPD        OPENED   -
OYA4: :SQL2:  HAUSHALT                       UPD        UPD        OPENED   -
OYA4: :SQL2:  SPORT                          UPD        UPD        OPENED   -
OYA4: :SQL2:  LEBENSMITTEL                   UPD        UPD        OPENED   -
OYA4: :SQL2:  SPIELE-HOBBY                   UPD        UPD        OPENED   -
OYA4: :SQL2:  SCHREIBWAREN                   UPD        UPD        OPENED   -
OYA4: :SQL2:  ARTIKELRLM                     UPD        UPD        OPENED   -
OYA4: -     SUCHRLM                          -          -          TMPRLM   -
OYA4:
OYA4:          DATABASE (USERID: $XXXXXXXXX.)  ACC-USER  ACC-SYS  DB-OPT  DESTR
OYA4: -----
OYA4:          KUNDEN                        UPD        UPD        EXC-UPD  -
OYA4: =====
OYA4: CATID   REALM                          ACC-USER  ACC-SYS  STATE   DEFECT
OYA4: -----
OYA4: :SQL2:  DATABASE-DIRECTORY              UPD        UPD        OPENED   -
OYA4: :SQL2:  KUNDENRLM                       UPD        UPD        OPENED   -
OYA4: :SQL2:  FINANZRLM                       UPD        UPD        OPENED   -
OYA4: -     TEMPRLM                          -          -          TMPRLM   -
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:40/OYA4)
    
```

4.4.14.11 DISPLAY REALMS, DB=dbname - Ausgabe von Informationen über alle permanenten Realms einer Datenbank

Eingabeformat: DISPLAY REALMS, DB=*dbname*

Erklärung: wie DISPLAY REALMS; die Ausgabe bezieht sich nur auf die mit *dbname* angegebene Datenbank

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY REALMS, DB=VERSAND

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY REALMS, DB=VERSAND
(OPOX073, 11:27:46/0YA4)
0YA4:      DATABASE (USERID: $XXXXXXXX.)  ACC-USER  ACC-SYS  DB-OPT  DESTR
0YA4: -----
0YA4:      VERSAND                        UPD      UPD      EXC-UPD  -
0YA4: =====
0YA4: CATID  REALM                        ACC-USER  ACC-SYS  STATE  DEFECT
0YA4: -----
0YA4: :SQL2: DATABASE-DIRECTORY            UPD      UPD      OPENED  -
0YA4: :SQL2: AUFTRAGSRLM                    UPD      UPD      OPENED  -
0YA4: :SQL2: BESTELLRLM                      UPD      UPD      OPENED  -
0YA4: :SQL2: KLEIDUNG                        UPD      UPD      OPENED  -
0YA4: :SQL2: HAUSHALT                        UPD      UPD      OPENED  -
0YA4: :SQL2: SPORT                          UPD      UPD      OPENED  -
0YA4: :SQL2: LEBENSMITTEL                    UPD      UPD      OPENED  -
0YA4: :SQL2: SPIELE-HOBBY                    UPD      UPD      OPENED  -
0YA4: :SQL2: SCHREIBWAREN                    UPD      UPD      OPENED  -
0YA4: :SQL2: ARTIKELRLM                      UPD      UPD      OPENED  -
0YA4: -      SUCHRLM                        -        -        TMPRLM  -
0YA4:
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074, 11:27:46/0YA4)
    
```

4.4.14.12 DISPLAY REALMS,RN=*realmname*[,DB=*dbname*] - Ausgabe von Informationen über einen permanenten Realm

Eingabeformat: DISPLAY REALMS,RN=*realmname*[,DB=*dbname*]

Erklärung: wie DISPLAY REALMS; die Ausgabe bezieht sich nur auf den mit *realmname* angegebenen Realm [der Datenbank *dbname*].

Die Angabe von DB=*dbname* kann entfallen, falls der genannte Realm-Name in der Konfiguration eindeutig ist.

4.4.14.13 DISPLAY PP - Ausgabe der aktuellen DBH-Ladeparameter

Eingabeformat: DISPLAY PP

Informationszeile(n):

MAXDB	=	n_1
TRANSACTION	=	n_2 bzw. (n_2, n_3) bei UDS-D
[SUBTRANSACTION	=	n_4]
SUBSCHEMA	=	n_5
SERVERTASK	=	n_6
2KB-BUFFER-SIZE	=	n_{7a}
4KB-BUFFER-SIZE	=	n_{7b}
8KB-BUFFER-SIZE	=	n_{7c}
CP-SIZE	=	n_8
CUP-SIZE	=	n_9
SIP-SIZE	=	n_{10}
[DIP-SIZE	=	n_{11}] bei UDS-D
[DISDB	=	n_{12}] bei UDS-D
[CHCKTIME	=	n_{13}] bei UDS-D
DEADTIME	=	n_{14}
DISTRIBUTION	=	{NO STANDBY START}
PTCSYNCH	=	({ <u>WAIT</u> <u>ABORT</u> <u>COMMIT</u> })
LOG	=	{NO :catid: <u>PUBLIC</u> (priv-vsn-1/gerät-1[,priv-vsn-2 gerät-2 [,priv-vsn-3 gerät-3]]) (vsn-1[,vsn-2[,vsn-3]])}
LOG-2	=	{NO :catid: <u>PUBLIC</u> (priv-vsn-1/gerät-1[,priv-vsn-2 gerät-2 [,priv-vsn-3 gerät-3]]) / (vsn-1[,vsn-2[,vsn-3]])}
LOG-SIZE	=	(primär, sekundär)

RESERVE = {NONE |
 :*catid*: |
 (*priv-vsn-1/gerät-1[,priv-vsn-2 gerät-2*
 [,*priv-vsn-3 gerät-3*]) |
 (*vsn-1[,vsn-2[,vsn-3]*)}

WARMSTART = {STD | FAST | VERY-FAST}

DEACT = {YES | NO}

STDCKPT = {YES | NO}

ADM = {LOCAL | REMOTE}

CPU = {MONO-PROCESSOR | MULTI-PROCESSOR}

[SQL = *n*₁₅]

[SQL-LIMIT = *n*₁₆]

[DISTABLE = {NO | [:*catid*:][*\$userid.*]dateiname}] bei
 UDS-D

TASKLIB = {NO | [:*catid*:][*\$userid.*]udsmo_{lib}}

DUMP = {STD | ALL}

MPSEG = {STD | 64K}

UCON = C' {(*mn*) | <*x* | *nnnn*} ' [, {MSG | UDS}]

SECLEVEL = {F2 | F2-EXCEPT | NO}, NO-AUDIT

LOCK = {STD | SHARED | EXCLUSIVE}

TA-ACCESS = {STD | SHARED}

WAIT = {EVENT | BUSY}

RESULT-DELAY = *n*₁₇

SCHEDULING = {SYMMETRIC | ASYMMETRIC}

IO = {ASYNC | SYNC}

ORDER-DBSTATUS = {STD | SPECIAL}

PRIVACY-CHECK = {STD | NO-KSET | OFF}

BCAM-PREFIX = *prefix*

CONFNAME = [*\$userid.*]konfname

DATABASES OF CONFIGURATION:

\$userid.dbname, {EXCLUSIVE-UPD[,*bbbb*][,*buffid* |
 SHARED-RTR[,*bbbb*][,*buffid*]

Bis zu MAXDB-Zeilen; entfällt bei leerer Konfiguration

- n_1 Anzahl der Datenbanken der DB-Konfiguration
- n_2 max. Anzahl der gleichzeitig aktiven Transaktionen
- n_3 max. Anzahl der gleichzeitig für entfernte Anwenderprogramme aktiven Transaktionen
- n_4 Anzahl der KDBS-Dateikennungen (nur KDBS)
- n_5 max. Anzahl der gleichzeitig pro Datenbank verwendbaren Subschemas
- n_6 Anzahl der Servertasks
- n_{7a} Größe des System Buffer Pools für Datenbanken im 2-Kbyte-Seitenformat in Mbyte
- n_{7b} Größe des System Buffer Pools für Datenbanken im 4-Kbyte-Seitenformat in Mbyte
- n_{7c} Größe des System Buffer Pools für Datenbanken im 8-Kbyte-Seitenformat in Mbyte
- n_8 Mindestgröße eines Common Pools in Kbyte
- n_9 Größe des Communication Pools in Kbyte
- n_{10} Größe des SSITAB-Pools in Kbyte
- n_{11} Größe des Distribution Pools in Kbyte
- n_{12} max. Anzahl entfernter Datenbanken pro Transaktion
- n_{13} Zeitangabe in Sekunden für die Verbindungs- und Transaktionsüberwachung der UDS-D-Task
- n_{14} Zeitangabe in Sekunden zur Auflösung konfigurationsübergreifender Deadlocks bzw. Deadlocks im Zusammenhang mit openUTM
- n_{15} Anzahl der gleichzeitig aktiven SQL-Vorgänge
- n_{16} Zeitspanne für nicht aktive Vorgänge
- n_{17} Anzahl Auftragsantworten die gebündelt werden sollen
- bbbb* Größe des User Buffer Pools der Datenbank in Mbyte ohne führende Nullen; fehlt ganz, wenn der Datenbank der System Buffer Pool zugewiesen ist.
- buffid* Name des Shared User Buffer Pools, falls der Datenbank ein Shared User Buffer Pool zugewiesen ist.
*EXCL zeigt an, dass der Datenbank ein exklusiver User Buffer Pool zugewiesen ist.
*SYSTEM zeigt an, dass der Datenbank der System Buffer Pool zugewiesen ist.
- prefix* Aus *prefix* und TSN werden die BCAM-Namen der Anwendertasks gebildet.

Erklärungen:

Der Aufruf gibt die aktuellen Werte der Ladeparameter des DBH inklusive DAL-Änderungen aus.

Sie werden in der Form aufgelistet wie bei Angabe von PP PARLIST=YES während der Startphase.

Die Ladeparameter SECLEVEL und TASKLIB sind ohne Wirkung. Sie werden nur noch aus Kompatibilitätsgründen ausgegeben.

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY PP

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY PP (OPOX073,11:27:50/0YA4)
0YA4: MAXDB          =      2
0YA4: TRANSACTION   = (    4,    1)
0YA4: SUBSCHEMA     =      1
0YA4: SERVERTASK    =      1
0YA4: 2KB-BUFFER-SIZE=      1
0YA4: 4KB-BUFFER-SIZE=      1
0YA4: 8KB-BUFFER-SIZE=      0
0YA4: CP-SIZE       =    1024
0YA4: CUP-SIZE     =    1024
0YA4: SIP-SIZE     =    1024
0YA4: DIP-SIZE     =    1024
0YA4: DISDB        =      1
0YA4: CHCKTIME     =      60
0YA4: DEADTIME     =      60
0YA4: DISTRIBUTION = START
0YA4: PTCSYNCH     = (WAIT ,WAIT )
0YA4: LOG           = PUBLIC
0YA4: LOG-2        = NO
0YA4: LOG-SIZE     = (    192,    192)
0YA4: RESERVE      = NONE
0YA4: WARMSTART    = STD
0YA4: DBDCSYNCH   = YES
0YA4: DEACT        = YES
0YA4: STDCKPT     = YES
0YA4: ADM          = REMOTE
0YA4: CPU          = MONO-PROCESSOR
0YA4: DISTABLE    = UDSDBB.VT.BEISPIEL.START
0YA4: SQL          =      4
0YA4: SQL-LIMIT   =     10
0YA4: TASKLIB     = NO
0YA4: DUMP        = ALL
0YA4: MPSEG       = STD
0YA4: UCON        = C'<U ',MSG
0YA4: SECLEVEL    = NO      ,NO-AUDIT
0YA4: LOCK        =          = STD
0YA4: TA-ACCESS   = STD
0YA4: WAIT        = EVENT
0YA4: RESULT-DELAY =      0
0YA4: SCHEDULING  = SYMMETRIC
0YA4: IO          = ASYNC
0YA4: ORDER-DBSTATUS = STD
0YA4: PRIVACY-CHECK = OFF
0YA4: BCAM-PREFIX = SUD$
0YA4: CONFNAME    = $XXXXXXXXX.CONFBSP
0YA4: DATABASES OF CONFIGURATION:
0YA4:   $XXXXXXXXX.VERSAND          ,EXCLUSIVE-UPD      ,*SYSTEM
0YA4:   $XXXXXXXXX.KUNDEN          ,EXCLUSIVE-UPD,    1,*EXCL
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:27:50/0YA4)

```

4.4.14.14 DISPLAY FPA - Ausgabe von Informationen über freie Seiten in allen Realms

Eingabeformat: DISPLAY FPA

Kommentarzeile: DATABASE

Informationszeile(n): *dbname*

Kommentarzeilen:	RLM	REALM	NR	NR FREE	SECOND
	REF		PAGES	PAGES	SCAN
Informationszeile(n):	<i>realm-</i>	<i>realmname</i>	<i>n</i>	<i>n</i>	<i>second</i>
	<i>referenz</i>				<i>scan</i>

Erklärungen:

Der Datenbankadministrator kann sich über die Anzahl der (freien) Seiten der Realms informieren.

DATABASE

dbname

Name der Datenbank

RLMREF

realmreferenz

Nummer des Realms

REALM

realmname

Name des Realms

NR PAGES *n*

Anzahl der Seiten des Realms

NR FREE PAGES *n*

Anzahl der freien Seiten des Realms

SECOND SCAN

second scan

Suchmodus

-

Der Realm wurde noch nicht vollständig nach freien Seiten durchsucht. Es wird der Suchmodus „First Scan“ verwendet (Platz für neu einzuspeichernde Daten wird im freien Bereich am Ende des Realms gesucht).

YES

Der Realm wurde bereits einmal vollständig nach freien Seiten durchsucht. Es wird der Suchmodus „Second Scan“ verwendet (Platz für neu einzuspeichernde Daten wird ab dem Beginn des Realms gesucht).

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY FPA

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY FPA (OPOX073,11:28:00/OYA4)
OYA4: DATABASE
OYA4: -----
OYA4: VERSAND
OYA4: =====
OYA4: RLM  REALM                NR      NR FREE  SECOND
OYA4: REF                PAGES   PAGES   SCAN
OYA4: -----
OYA4:  1  DATABASE-DIRECTORY      100     50     -
OYA4:  3  AUFTRAGSRLM             36      22     -
OYA4:  4  BESTELLRLM             60      33     -
OYA4:  5  KLEIDUNG                42      14     -
OYA4:  6  HAUSHALT               24      15     -
OYA4:  7  SPORT                  44      36     -
OYA4:  8  LEBENSMITTEL           18       8     -
OYA4:  9  SPIELE-HOBBY           44      36     -
OYA4: 10  SCHREIBWAREN           24      17     -
OYA4: 11  ARTIKELRLM            2062    1981     -
OYA4:
OYA4: DATABASE
OYA4: -----
OYA4: KUNDEN
OYA4: =====
OYA4: RLM  REALM                NR      NR FREE  SECOND
OYA4: REF                PAGES   PAGES   SCAN
OYA4: -----
OYA4:  1  DATABASE-DIRECTORY      100     55     -
OYA4:  3  KUNDENRLM               126     23     -
OYA4:  4  FINANZRLM               126    117     -
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:28:00/OYA4)

```

4.4.14.15 DISPLAY FPA,DB=*dbname* - Ausgabe von Informationen über freie Seiten in allen Realms einer Datenbank

Eingabeformat: DISPLAY FPA,DB=*dbname*

Erklärung: wie DISPLAY FPA; die Ausgabe bezieht sich nur auf die mit *dbname* angegebene Datenbank.

4.4.14.16 DISPLAY FPA,RN=*realmname* [,DB=*dbname*]- Ausgabe von Informationen über freie Seiten in einem Realm

Eingabeformat: DISPLAY FPA,RN=*realmname*[,DB=*dbname*]

Erklärung: wie DISPLAY FPA; die Ausgabe bezieht sich nur auf den mit *realmname* angegebenen Realm [der Datenbank *dbname*].

Die Angabe von DB=*dbname* kann entfallen, falls der genannte Realm-Name in der Konfiguration eindeutig ist.

4.4.14.17 DISPLAY XFPA - Ausgabe von erweiterten Informationen über freie Seiten in allen Realms

Eingabeformat: DISPLAY XFPA

Kommentarzeile: DATABASE

Informationszeile(n): *dbname*

Kommentar-	RLM	NR	NR FREE	NR FULL	PARTIAL-FILLED	SECOND SEARCH
zeilen:	REF	PAGES	PAGES	PAGES	NR PAGES FIL%	SCAN IN FPA
Informations-	<i>realm-</i>	<i>n</i>	<i>n</i>	<i>n</i>	<i>n</i>	<i>second search</i>
zeile(n):	<i>referenz</i>					<i>scan in fpa</i>

Erklärungen:

Der Datenbankadministrator kann sich über die Anzahl der (freien) Seiten der Realms informieren.

DATABASE

dbname

Name der Datenbank

RLMREF

realmreferenz

Nummer des Realms

NR PAGES

Anzahl der Seiten des Realms

NR FREE PAGES

Anzahl der freien Seiten des Realms

NR FULL PAGES

Anzahl der vollen Seiten des Realms

PARTIAL-FILLED NR PAGES

Anzahl der teilgefüllten Seiten des Realms

FIL%

Durchschnittlicher Füllgrad aller teilgefüllten Seiten des Realms

SECOND SCAN

second scan

Suchmodus

-

Der Realm wurde noch nicht vollständig nach freien Seiten durchsucht. Es wird der Suchmodus „First Scan“ verwendet (Platz für neu einzuspeichernde Daten wird im freien Bereich am Ende des Realms gesucht).

YES

Der Realm wurde bereits einmal vollständig nach freien Seiten durchsucht. Es wird der Suchmodus „Second Scan“ verwendet (Platz für neu einzuspeichernde Daten wird ab dem Beginn des Realms gesucht).

SEARCH IN FPA*search in fpa*

Der von der UDS-Online-Utility hinterlegte Wert von SET-FPA-SCAN-PARAMETERS bzw. von BMODTT der Wert SET/RESET REUSE-FREE-SPACE

REUSE

Es wird ab dem Beginn des Realms nach Platz für neu einzuspeichernde Sätze und Tabellen gesucht (Second Scan).

NOREUSE

Es wird am Realm-Ende hinter der ersten Seite des zusammenhängenden Bereichs freier Seiten nach Platz für neu einzuspeichernde Sätze und Tabellen gesucht (First Scan).

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY XFPA

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY XFPA (OPOX073,11:28:06/OYA4)
OYA4: DATABASE
OYA4: -----
OYA4: VERSAND
OYA4: =====
OYA4: RLM      NR      NR FREE  NR FULL  PARTIAL-FILLED  SECOND  SEARCH
OYA4: REF      PAGES  PAGES   PAGES   NR PAGES  FIL%   SCAN   IN FPA
OYA4: -----
OYA4:  1        100     50      43       7    76    -    NOREUSE
OYA4:  3         36     22      13       1     2    -    NOREUSE
OYA4:  4         60     33      25       2   37    -    NOREUSE
OYA4:  5         42     14      17      11     8    -    NOREUSE
OYA4:  6         24     15       9       0     -    -    NOREUSE
OYA4:  7         44     36       8       0     -    -    NOREUSE
OYA4:  8         18      8       6       4     7    -    NOREUSE
OYA4:  9         44     36       8       0     -    -    NOREUSE
OYA4: 10         24     17       7       0     -    -    NOREUSE
OYA4: 11        2062   1981    74       7     7    -    NOREUSE
OYA4:
OYA4: DATABASE
OYA4: -----
OYA4: KUNDEN
OYA4: =====
OYA4: RLM      NR      NR FREE  NR FULL  PARTIAL-FILLED  SECOND  SEARCH
OYA4: REF      PAGES  PAGES   PAGES   NR PAGES  FIL%   SCAN   IN FPA
OYA4: -----
OYA4:  1        100     55      38       7   74    -    NOREUSE
OYA4:  3        126     23     100       3   39    -    NOREUSE
OYA4:  4        126    117       7       2   22    -    NOREUSE
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:28:06/OYA4)
```

4.4.14.18 DISPLAY XFPA,DB=dbname - Ausgabe von erweiterten Informationen über freie Seiten in allen Realms einer Datenbank

Eingabeformat: DISPLAY XFPA,DB=*dbname*

Erklärung: wie DISPLAY XFPA; die Ausgabe bezieht sich nur auf die mit *dbname* angegebene Datenbank.

4.4.14.19 DISPLAY XFPA,RN=*realmname* [,DB=*dbname*]- Ausgabe von erweiterten Informationen über freie Seiten in einem Realm

Eingabeformat: DISPLAY XFPA,RN=*realmname*[,DB=*dbname*]

Erklärung: wie DISPLAY XFPA; die Ausgabe bezieht sich nur auf den mit *realmname* angegebenen Realm [der Datenbank *dbname*].

Die Angabe von DB=*dbname* kann entfallen, falls der genannte Realm-Name in der Konfiguration eindeutig ist.

4.4.14.20 DISPLAY INCR - Ausgabe von Informationen über die Online-Erweiterbarkeit aller zugeschalteten Realms in der DB-Konfiguration

Eingabeformat: DISPLAY INCR

Informationszeile(n): DATABASE (USERID: \$xxxxxxxx.)

Informationszeile(n): *dbname*

Kommentarzeilen:	RLM REALM	INCR	NR	MIN	FMT	EXT	EXT
	REF	ACT	PAGES	PAGES		PND	SIZE

Informationszeile(n):	<i>n</i>	<i>realmname</i>	<i>incr</i>	<i>n</i>	<i>n</i>	<i>fmt</i>	<i>ext</i>	<i>n</i>
			<i>act</i>				<i>pnd</i>	

Erklärungen:

Der Datenbankadministrator kann sich über die Online-Erweiterbarkeit der Realms informieren.

USERID

Kennung, in der die betroffene Datenbank steht.

DATABASE

dbname

Name der Datenbank

RLM REF *n*

Nummer des Realms

REALM

realmname

Name des Realms

INCR ACT

incr act

Online-Erweiterbarkeit

YES

Für den Realm ist die Online-Erweiterbarkeit aktiviert und nicht ausgesetzt.

NO

Für den Realm ist die Online-Erweiterbarkeit nicht aktiviert.

SUSP

Für den Realm ist die Online-Erweiterbarkeit aktiviert, aber im aktuellen Sessionabschnitt ausgesetzt.

NR PAGES *n*

Anzahl der Seiten, um die der Realm im Bedarfsfall erweitert wird.

MIN PAGES *n*

Anzahl der freien Seiten des Realms, bei deren Unterschreitung eine Online-Realm-Erweiterung angestoßen wird.

FMT

fmt

Formatierung der neu hinzugefügten Datenbankseiten

YES

Formatierung der neu hinzugefügten Datenbankseiten durch den DBH ist nötig.

NO

Formatierung der neu hinzugefügten Datenbankseiten durch den DBH ist nicht nötig.

EXT PND

ext pnd

Anzeige zu unbedingter Realmerweiterung.

YES

Unbedingte einmalige Realmerweiterung ist vorgemerkt.

NO

Keine Vormerkung für unbedingte einmalige Realmerweiterung.

EXT SIZE *n*

Gewünschte Anzahl von Datenbankseiten, um die erweitert werden soll.

Wird nur angezeigt, falls unbedingte einmalige Realmerweiterung vorgemerkt ist, also EXT PND gleich YES ist.

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY INCR

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY INCR (OPOX073,11:28:12/OYA4)
OYA4: DATABASE (USERID: $XXXXXXXXX.)
OYA4: -----
OYA4: VERSAND
OYA4: =====
OYA4: RLM REALM          INCR NR          MIN          FMT EXT EXT
OYA4: REF              ACT          PAGES        PAGES        PND SIZE
OYA4: -----
OYA4:  1 DATABASE-DIRECTORY      YES          64          0 YES NO
OYA4:  3 AUFTRAGSRML             YES         2000         400 NO NO
OYA4:  4 BESTELLRLM              YES          64          0 NO NO
OYA4:  5 KLEIDUNG                YES          64          0 YES NO
OYA4:  6 HAUSHALT                YES          64          0 NO NO
OYA4:  7 SPORT                   YES          64          0 NO NO
OYA4:  8 LEBENSMITTEL            YES          500         50 NO NO
OYA4:  9 SPIELE-HOBBY            YES          64          0 NO NO
OYA4: 10 SCHREIBWAREN            YES          64          0 NO NO
OYA4: 11 ARTIKELRLM             YES         1000         100 NO NO
OYA4:
OYA4: DATABASE (USERID: $XXXXXXXXX.)
OYA4: -----
OYA4: KUNDEN
OYA4: =====
OYA4: RLM REALM          INCR NR          MIN          FMT EXT EXT
OYA4: REF              ACT          PAGES        PAGES        PND SIZE
OYA4: -----
OYA4:  1 DATABASE-DIRECTORY      YES          64          0 YES NO
OYA4:  3 KUNDENRLM               YES          500         30 NO NO
OYA4:  4 FINANZRLM               YES          64          0 NO NO
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:28:12/OYA4)

```

4.4.14.21 DISPLAY INCR, DB=dbname - Ausgabe von Informationen über die Online-Erweiterbarkeit aller zugeschalteten Realms einer Datenbank

Eingabeformat: DISPLAY INCR, DB=*dbname*

Erklärung: wie DISPLAY INCR; die Ausgabe bezieht sich nur auf die zugeschalteten Realms der Datenbank *dbname*.

4.4.14.22 DISPLAY INCR,RN=*realmname*[,DB=*dbname*] - Ausgabe von Informationen über die Online-Erweiterbarkeit eines Realms

Eingabeformat: DISPLAY INCR[, DB=*dbname*],RN=*realmname*

Erklärung: wie DISPLAY INCR; die Ausgabe bezieht sich nur auf den Realm *realmname*.

Die Angabe von DB=*dbname* kann entfallen, falls der genannte Realm-Name in der Konfiguration eindeutig ist.

4.4.14.23 DISPLAY DBTT-INCR - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit der Satzarten der User-Schemata in der DB-Konfiguration

Eingabeformat: DISPLAY DBTT-INCR

Informationszeile(n): DATABASE (USERID: \$xxxxxxxx.)

Informationszeile(n): *dbname*

Kommentarzeilen: REC RECORD RLM DBTT NR EXT HIGHEST EXT
REF REF ACT ENTRIES PND RSQ SIZE

Informationszeile(n): *n recordname n dbtt n ext n n*
act pnd

Erklärungen:

Der Datenbankadministrator kann sich über die Online-Erweiterbarkeit der DBTTs informieren.

USERID

Kennung, in der die betroffene Datenbank steht.

dbname

Name der Datenbank

REC REF *n*

Nummer der Satzart

RECORD

recordname

Name der Satzart

RLM REF *n*

Nummer des Realms

Mit dem DAL-Kommando DISPLAY INCR,DB=*dbname* kann der Name des Realms ermittelt werden.

DBTT ACT

dbtt act

Anzeige, ob die Online-DBTT-Erweiterung aktiviert ist:

SCAN

Die Online-DBTT-Erweiterung ist aktiviert und nicht ausgesetzt. Die Online-DBTT-Erweiterung wird erst vorgenommen, wenn in der gesamten DBTT kein freier Eintrag gefunden wurde.

NOSCAN

Die Online-DBTT-Erweiterung ist aktiviert und nicht ausgesetzt. Die Online-DBTT-Erweiterung wird vorgenommen, wenn die Suche nach einem freien Eintrag bei der höchsten aktuell möglichen RSQ erfolglos ist. Es wird also nicht mehr erneut nach freien Einträgen mit niedriger RSQ gesucht.

NO

Die Online-DBTT-Erweiterung ist nicht aktiviert.

SUSP

Die Online-DBTT-Erweiterung ist im aktuellen Sessionabschnitt ausgesetzt.

Dies kann seine Ursache in der Suspendierung der Online-Realm-Erweiterung im entsprechenden Realm oder im Erreichen der Maximalgröße der DBTT haben.

SWTOFF

Der Realm, in dem die DBTT liegt, ist derzeit nicht verfügbar. Der Zustand der Aktivierung ist unklar.

NR ENTRIES *n*

Anzahl von Einträgen bei einer DBTT-Erweiterung.

EXT PND*ext pnd*

Information über noch nicht ausgeführte EXTEND-DBTT-Aufträge:

YES

Für diese Satzart ist ein Auftrag vorgemerkt, der beim nächsten Abspeichern eines Satzes ausgeführt wird.

NO

Es ist kein Auftrag vorgemerkt.

HIGHEST RSQ *n*

Höchstmögliche RSQ der Satzart in der aktuellen DBTT

EXT SIZE *n*

Gewünschte Anzahl von DBTT Einträgen, um die erweitert werden soll.

Wird nur angezeigt, falls eine unbedingte einmalige DBTT-Erweiterung vorgemerkt ist, also EXT PND gleich YES ist.

Falls die Anzahl der maximalen Ausgabezeilen für ein DISPLAY-Kommando nicht ausreicht, um die Information für alle Satzarten auszugeben, wird ein entsprechender Hinweis ausgegeben:

```
OUTPUT OF nnnn FURTHER LINES SUPPRESSED
```

Es gibt kein Kommando, um dieses abgebrochene DISPLAY-Kommando fortzusetzen. Um die Information für alle Satzarten zu erhalten, muss das Kommando z.B. mit LINES=ALL wiederholt werden. Für einzelne Datenbanken können entsprechende Informationen auch mit dem Dienstprogramm BSTATUS bereitgestellt werden.

Abschließend wird in jedem Fall die folgende Meldung ausgegeben:

```
UDS0218 UDS HAT DIE BEARBEITUNG DES DAL KOMMANDOS ABGESCHLOSSEN
```

Ist eine Online-DBTT-Erweiterung gescheitert, dann können Sie sich folgendermaßen über die Gründe der Suspendierung informieren:

- mit dem DAL-Kommando DISPLAY INCR
- mit Hilfe des Wertes HIGHEST RSQ des Kommandos DISPLAY DBTT-INCR

Beispiel

//EXECUTE-DAL-CMD CMD=DISPLAY DBTT-INCR

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY DBTT-INCR (OPOX073,12:26:15
/OLME)
OLME: DATABASE (USERID: $XXXXXXXXX.)
OLME: -----
OLME: VERSAND
OLME: =====
OLME: REC   RECORD                RLM DBTT   NR           EXT HIGHEST   EXT
OLME: REF                REF ACT   ENTRIES   PND           RSQ  SIZE
OLME: -----
OLME:    2 KUNDE                3 SCAN     21184 NO           331
OLME:    3 AUFTRAG              3 SCAN     31808 NO           497
OLME:    4 AUFTR-POS            3 SCAN     63680 YES        1990   63680
OLME:    5 RATENZAHLUNG         3 SCAN     63680 NO           995
OLME:    6 ARTIKELART           11 NOSCAN   31808 NO           497
OLME:    7 ARTIKELAUSWAHL       11 SCAN     31808 NO           497
OLME:    8 ARTIKELBESCHR        11 SCAN     31808 NO           497
OLME:    9 ARTIKEL              11 NOSCAN   63680 YES        995   63680
OLME:   10 TEILMENGE            6 SCAN     63680 NO           995
OLME:   11 FARBEN                11 SCAN     63680 NO           995
OLME:   12 MATERIALIEN          11 NOSCAN   63680 NO           995
OLME:   13 LIEFERANT             4 SCAN     21184 NO           662
OLME:   14 BESTELLUNG           4 SCAN     31808 NO           497
OLME:   15 BESTELL-POS          4 SCAN     63680 NO           995
OLME:
OLME: DATABASE (USERID: $XXXXXXXXX.)
OLME: -----
OLME: KUNDEN
OLME: =====
OLME: REC   RECORD                RLM DBTT   NR           EXT HIGHEST   EXT
OLME: REF                REF ACT   ENTRIES   PND           RSQ  SIZE
OLME: -----
OLME:    2 KUNDE                3 SCAN     21184 NO           662
OLME:    3 OFFENE-POSTEN         3 SCAN     63680 NO           995
OLME:    4 KREDIT                 4 SCAN     63680 NO        1990
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:26:17/OLME)

```

4.4.14.24 DISPLAY DBTT-INCR,DB=*dbname* - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit der Satzarten des User-Schemas einer Datenbank

Eingabeformat: DISPLAY DBTT-INCR,DB=*dbname*

Erklärung: wie DISPLAY DBTT-INCR; die Ausgabe bezieht sich nur auf die Satzarten des User-Schemas der Datenbank *dbname*.

4.4.14.25 DISPLAY DBTT-INCR,DB=*dbname*,RECR=*recordref* - Ausgabe von Informationen über die Online-DBTT-Erweiterbarkeit einer Satzart

Eingabeformat: DISPLAY DBTT-INCR,DB=*dbname*,RECR=*recordref*

Erklärung: wie DISPLAY DBTT-INCR; die Ausgabe bezieht sich nur auf die Satzart mit der Satzartnummer *recordref* des User-Schemas der Datenbank *dbname*.

4.4.14.26 DISPLAY PUBSETS - Ausgabe der Einstellungen der UDS/SQL-Pubsetdeklaration

Eingabeformat: DISPLAY PUBSETS

Erklärungen:

Kopfzeile(n): UDS0746 UDS-PUBSET-DEKLARATION (CURRENT) FOLGT

Informationszeile: UDS-PUBSET-JV: *jv-name*

Informationszeile(n): PUBSETS: *catid-group*

Endzeile: DEFAULT PUBSET: *default-catid*

jv-name

Name der Pubset-Deklarations-Jobvariable

catid-group

Catid-Gruppe, die in der UDS/SQL-Pubset-Deklaration angegeben wurde.

default-catid

Standard-Katalogkennung der Ablaufkennung

Der Datenbankadministrator kann sich über die aktuellen Einstellungen der UDS/SQL-Pubset-Deklaration informieren. Zwischen den DAL-Kommandos NEW PUBSETS und PERFORM werden zusätzlich die Einstellungen der neuen UDS-Pubsetdeklaration in gleichem Format ausgegeben, jedoch mit dem Insert "PENDING" statt "CURRENT" bei der Meldung UDS0746.

Falls keine UDS-Pubsetdeklaration vorliegt, erfolgt zur Unterscheidung von einer explizit angegebenen Catid-Gruppe "*" die Ausgabe:

```
UDS0747 UDS-PUBSET-DEKLARATION NICHT VORHANDEN, DEFAULT IST CATID * (CURRENT) (... ,  
tsn4)
```

4.4.14.27 DISPLAY ALOG [,DB=dbname] - Ausgabe von ALOG-Einstellungen

Eingabeformat: DISPLAY ALOG [,DB=*dbname*]

Kommentar- USERID DATABASE-NAME DEF-SUP RES-SUP PRIM-ALL SEC-ALL SHARE
zeile:

Informations *userid* *.dbname* *def-cat* *res-cat* *prim* *sec* *share*
zeile(n):

Endzeile: UDS/SQL *Vn.n nnn* DATABASE(S) IN konfname

Erklärungen:

Der Datenbankadministrator kann sich über die Einstellungen für die ALOG-Dateien informieren.

USERID

userid

Kennung, in der die betroffene Datenbank steht.

DATABASE-NAME

dbname

Name der Datenbank

DEF-SUP

def-cat

BS2000-Katalogkennung der ALOG-Einstellung DEFAULT-SUPPORT

RES-SUP

res-cat

BS2000-Katalogkennung der ALOG-Einstellung RESERVE-SUPPORT

PRIM-ALL

prim

Primärzuweisung der ALOG-Datei

SEC-ALL

sec

Sekundärzuweisung der ALOG-Datei

SHARE

share

Zugriff auf die ALOG-Datei

YES

Alle Benutzer können auf die ALOG-Datei zugreifen

NO

Nur der Eigentümer kann auf die ALOG-Datei zugreifen

Vn.n UDS/SQL-Versionsnummer

nnn Anzahl der Datenbanken in der DB-Konfiguration

Beispiel

```
0L9M: USERID      DATABASE-NAME      DEF-SUP  RES-SUP  PRIM-ALL  SEC-ALL  SHARE
0L9M: -----
0L9M: $XXXXXXXXX.VERSAND          IUDS     ABN2     192      576      NO
0L9M: $XXXXXXXXX.KUNDE           IUDS     IUDS     300      600      YES
0L9M:
0L9M: UDS/SQL V2.9    2 DATABASES IN CONFEXMP
```

4.4.15 Verteiltabelle ganz oder teilweise ausgeben (&DISPLAY DISTRIBUTION)

Für UDS-D

```
&DISPLAY DISTRIBUTION
[ ,NODE=prozessorname ]
[ ,CONF=konfname ]
[ ,DB=dbname ]
[ ,SS=subschemaname ]
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*;
er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht '@' sein. Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

dbname

Name einer Datenbank;
er muss netzweit eindeutig sein.

Die Angabe :*catid.\$userid.dbname.realmname.copypname* darf maximal 54 Zeichen lang sein.

subschemaname

Name eines Subschemas;
er darf max. 30 Zeichen lang sein;
er muss in den ersten sechs Zeichen netzweit eindeutig sein.



Gesperrte Einträge sind durch ein vorangestelltes Minuszeichen gekennzeichnet.

Die Ausgabe der Verteiltabelle ist auf 32 KB beschränkt. Ist die Verteiltabelle größer, dann wird sie nicht vollständig ausgegeben und Sie werden in einer Meldung hierüber informiert.

UDS-D gibt Folgendes auf die Datensichtstation aus:

Bei &DISPLAY DISTRIBUTION ohne Operanden wird die ganze lokale Verteiltabelle ausgegeben:

Beispiel 1

```
//EXECUTE-DAL-CMD CMD=&DISPLAY DISTRIBUTION
```

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &DISPLAY DISTRIBUTION (OPOX073,11:28:26/0YA4)
0YA4: =====
0YA4:  NODE-NAME  CONF-NAME  DB-NAME                SS-NAME
0YA4:  =====
0YA4:    PROZ1      CONF3      DB11                   SS111
0YA4:                                     -SS112
0YA4:  -----
0YA4:                                     DB12                   SS121
0YA4:                                     SS122
0YA4:  -----
0YA4:    PROZ2      CONF4      DB21                   SS21
0YA4:  -----
0YA4:                                     DB22                   SS22
0YA4:  -----
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (OPCC382,11:28:26/0YA4)
```

Bei &DISPLAY DISTRIBUTION mit einem Operanden wird der angegebene Operand mit seiner unmittelbaren Umgebung in der Verteiltabelle ausgegeben:

Beispiel 2

```
//EXECUTE-DAL-CMD CMD=&DISPLAY DISTRIBUTION,DB=DB11
```

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &DISPLAY DISTRIBUTION,DB=DB11
(OPOX073,11:28:28/0YA4)
0YA4: =====
0YA4:  NODE-NAME  CONF-NAME  DB-NAME                SS-NAME
0YA4:  =====
0YA4:                CONF3      DB11                   SS111
0YA4:                                     -SS112
0YA4:  -----
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (OPCC382,11:28:28/0YA4)
```

Bei &DISPLAY DISTRIBUTION mit mehreren Operanden werden die angegebenen Operanden mit ihrer unmittelbaren Umgebung in der Verteiltabelle ausgegeben:

Beispiel 3

```
//EXECUTE-DAL-CMD CMD=&DISPLAY DISTRIBUTION,CONF=CONF3,DB=DB11
```

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &DISPLAY DISTRIBUTION,CONF=CONF3,
DB=DB11
(OPOX073,11:28:30/0YA4)
0YA4: =====
0YA4:  NODE-NAME  CONF-NAME  DB-NAME                SS-NAME
```

```
OYA4: =====  
OYA4:  PROZ1      CONF3      DB11                      SS111  
OYA4:                                          -SS112  
OYA4: -----  
%  UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (OPCC382,11:28:30/OYA4)
```

4.4.16 Informationen über einen SQL-Vorgang ausgeben (DISPLAY SQL)

```
DISPLAY SQL{ ,VG=vg-nr |  
             ,OPTION=ALL[ ,VG=vg-nr] |  
             ,OPTION=IDLE[ ,TIME=z ] }
```

vg-nr Nummer des Vorgangs;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs;
Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

z Zeitintervall von 1..999 Minuten

Das DISPLAY SQL-Kommando protokolliert auf der Datensichtstation des Datenbankadministrators bzw. auf der Operatorkonsole bei Angabe von

VG=*vg-nr*

alle Informationen zu dem Vorgang *vg-nr*

OPTION=ALL

alle Vorgänge aufsteigend sortiert nach Vorgangsalter

OPTION=ALL, VG=*vg-nr*

alle Vorgänge ab dem angegebenen aufsteigend sortiert nach Vorgangsalter

OPTION=IDLE

die am längsten inaktiven Vorgänge nach Dauer der Inaktivität aufsteigend sortiert

OPTION=IDLE, TIME=*z*

alle inaktiven Vorgänge, die mindestens *z* Minuten inaktiv sind, nach Dauer der Inaktivität aufsteigend sortiert

Im Folgenden ist beschrieben, was die verschiedenen DISPLAY SQL-Kommandos ausgeben.

Ausgabe alle Informationen zu einem Vorgang**Eingabeformat:** DISPLAY SQL,VG=*vg-nr*

Informationszeilen:	VG-NR:	<i>vg-nr</i>
	UTM-APPL-/USER-NAME:	<i>appl-/user-name</i>
	BATCH/TSN:	<i>tsn</i>
	VG-START:	<i>date</i>
	NR-TA:	<i>nr-ta</i>
	NR-STMT:	<i>nr-stmt</i>
	MEM:	<i>mem</i> (KB)
	SQL-OPC:	<i>opcode</i>
	TA-ID:	<i>ta-id</i>
	IDLE-TIME:	<i>time</i> (MIN)

Erklärungen:

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Sessionabschnitts fortlaufend nummeriert.

appl-/user-name

openUTM-Anwendungsname/Benutzername, maximal siebzehnstellig

tsn Prozessfolgenummer, unter der das Anwenderprogramm läuft, maximal vierstellig

date Zeitstempel für Beginn des Vorgangs in UDS/SQL:
jjjj-mm-tt hh.mm.ss

nr-ta Anzahl abgeschlossener SQL-Transaktionen des Vorgangs, maximal vierstellig

nr-stmt

Anzahl bearbeiteter SQL-Anweisungen des Vorgangs, maximal vierstellig

mem Umfang des SQL-spezifischen Memory in Kbyte, maximal vierstellig

opcode

aktueller bzw. jüngster SQLU-Anweisungscode, maximal sechsstellig, wie folgt:

PERMIT	PERMIT
CLOCUR	CLOSE CURSOR
COMWOR	COMMIT WORK
CRETV	CREATE TEMPORARY VIEW
DCLCUR	DECLARE CURSOR
DELPOS	DELETE POSITIONED
DELSRC	DELETE SEARCHED
FETCH	FETCH
INSERT	INSERT
OPECUR	OPEN CURSOR
RESCUR	RESTORE CURSOR
ROLWOR	ROLLBACK WORK
SELECT	SELECT
SETTA	SET TRANSACTION
STOCUR	STORE CURSOR
UPDPOS	UPDATE POSITIONED
UPDSRC	UPDATE SEARCHED

ta-id globale Transaktions-Id der momentan offenen TA des Vorgangs, maximal achtstellig, bzw. '- - -', wenn keine TA offen ist, oder wenn die TA wegen eines DAL-Kommandos STOP NEW noch nicht eröffnet wurde.

time Zeitraum der Inaktivität in Minuten, maximal vierstellig, bzw. '- - -', wenn eine TA offen ist, also der Vorgang aktiv ist.

Wenn die Anzahl der Stellen des jeweiligen Ausgabefeldes zu klein ist, wird anstatt des Wertes der String '>>..' ausgegeben.

Ausgabe aller aktiven und inaktiven Vorgänge

Eingabeformat: DISPLAY SQL,OPTION=ALL

Kopfzeile: VG-NR TYPE VG-START MEM TA-ID IDLT

Informationszeile(n): *vg-nr type date mem ta-id time*

Endzeile: {END OF DISPLAY | CONTINUE FOR *n* FURTHER VGS}

Erklärungen:

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

type Typ der Anwendung, die den Vorgang ausgelöst hat:

UTM

Es handelt sich um eine UDS/SQL-openUTM-Anwendung

BATCH

Es handelt sich um eine UDS/SQL-TIAM-Anwendung

<i>date</i>	Zeitstempel für Beginn des Vorgangs in UDS/SQL: <i>jjjj-mm-tt hh.mm.ss</i>
<i>mem</i>	Umfang des SQL-spezifischen Memory in Kbyte, maximal vierstellig
<i>ta-id</i>	globale Transaktions-Id der momentan offenen TA des Vorgangs, maximal achtstellig, bzw. '- - -', wenn keine TA offen ist, oder wenn die TA wegen eines DAL-Kommandos STOP NEW noch nicht eröffnet wurde.
<i>time</i>	Zeitraum der Inaktivität in Minuten, maximal vierstellig, bzw. '- - -', wenn eine TA offen, also der Vorgang aktiv ist
<i>n</i>	Anzahl der noch nicht ausgegebenen Vorgänge, maximal vierstellig

Wenn die Anzahl der Stellen des jeweiligen Ausgabefeldes zu klein ist, wird anstatt des Wertes der String '>>..>' ausgegeben.

Beginnend mit dem jüngsten Vorgang werden alle aktiven Vorgänge, geordnet nach steigendem Vorgangsalter, ausgegeben.

Es werden nur so viele Vorgänge ausgegeben, wie auf den Bildschirm passen. Die Ausgabe wird ggf. abgeschlossen durch die Anzahl der noch nicht ausgegebenen Vorgänge. Diese Vorgänge können Sie mit dem DAL-Kommando CONTINUE ausgeben.

Ausgabe aller aktiven und inaktiven Vorgänge ab der angegebenen Vorgangsnummer

Eingabeformat: DISPLAY SQL,OPTION=ALL,VG=*vg-nr*

Kopfzeile:	VG-NR	TYPE	VG-START	MEM	TA-ID	IDLT
Informationszeile(n):	<i>vg-nr</i>	<i>type</i>	<i>date</i>	<i>mem</i>	<i>ta-id</i>	<i>time</i>
Endzeile:	{END OF DISPLAY CONTINUE FOR <i>n</i> FURTHER VGS}					

Erklärungen:

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

type Typ der Anwendung, die den Vorgang ausgelöst hat:

UTM

Es handelt sich um eine UDS/SQL-openUTM-Anwendung

BATCH

Es handelt sich um eine UDS/SQL-TIAM-Anwendung

date Zeitstempel für Beginn des Vorgangs in UDS/SQL:
jjjj-mm-tt hh.mm.ss

mem Umfang des SQL-spezifischen Memory in Kbyte, maximal vierstellig

ta-id globale Transaktions-Id der momentan offenen TA des Vorgangs, maximal achtstellig, bzw. '- - -', wenn keine TA offen ist, oder wenn die TA wegen eines DAL-Kommandos STOP NEW noch nicht eröffnet wurde.

time

Zeitraum der Inaktivität in Minuten, maximal vierstellig, bzw. '- - -', wenn eine TA offen, also der Vorgang aktiv ist

n Anzahl der noch nicht ausgegebenen Vorgänge, maximal vierstellig

Wenn die Anzahl der Stellen des jeweiligen Ausgabefeldes zu klein ist, wird anstatt des Wertes der String '>>..>' ausgegeben.

Ab dem angegebenen Vorgang werden alle weiteren Vorgänge, geordnet nach steigendem Vorgangsalter, ausgegeben.

Es werden nur so viele Vorgänge ausgegeben, wie auf den Bildschirm passen. Die Ausgabe wird ggf. abgeschlossen durch die Anzahl der noch nicht ausgegebenen Vorgänge.

Diese Vorgänge können Sie mit dem DAL-Kommando CONTINUE ausgeben.

Ausgabe der am längsten inaktiven Vorgänge

Eingabeformat: DISPLAY SQL,OPTION=IDLE

Kopfzeile:	VG-NR	TYPE	VG-START	MEM	IDLT
Informationszeile(n):	<i>vg-nr</i>	<i>type</i>	<i>date</i>	<i>mem</i>	<i>time</i>
Endzeile:	{END OF DISPLAY CONTINUE FOR <i>n</i> FURTHER VGS}				

Erklärungen:

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

type Typ der Anwendung, die den Vorgang ausgelöst hat:

UTM
Es handelt sich um eine UDS/SQL-openUTM-Anwendung

BATCH
Es handelt sich um eine UDS/SQL-TIAM-Anwendung

date Zeitstempel für Beginn des Vorgangs in UDS/SQL:
jjjj-mm-tt hh.mm.ss

mem Umfang des SQL-spezifischen Memory in Kbyte, maximal vierstellig

time Zeitraum der Inaktivität in Minuten, maximal vierstellig, bzw. '- - -', wenn eine TA offen ist, also der Vorgang aktiv ist.

n Anzahl der noch nicht ausgegebenen inaktiven Vorgänge, maximal vierstellig

Wenn die Anzahl der Stellen des jeweiligen Ausgabefeldes zu klein ist, wird anstatt des Wertes der String '>>..>' ausgegeben.

Es werden die am längsten inaktiven Vorgänge ausgegeben, geordnet nach steigender Dauer ihrer Inaktivität.

Es werden nur so viele Vorgänge ausgegeben, wie auf den Bildschirm passen. Die Ausgabe wird ggf. abgeschlossen durch die Anzahl der noch nicht ausgegebenen inaktiven Vorgänge.

Diese Vorgänge können Sie mit dem DAL-Kommando CONTINUE ausgeben.

Ausgabe aller inaktiven Vorgänge, die seit einer bestimmten Zeit inaktiv sind

Eingabeformat: DISPLAY SQL,OPTION=IDLE,TIME=*z*

Kopfzeile:	VG-NR	TYPE	VG-START	MEM	IDLT
Informationszeile(n):	<i>vg-nr</i>	<i>type</i>	<i>date</i>	<i>mem</i>	<i>time</i>
Endzeile:	{END OF DISPLAY CONTINUE FOR <i>n</i> FURTHER VGS}				

Erklärungen:

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

type Typ der Anwendung, die den Vorgang ausgelöst hat:

UTM

Es handelt sich um eine UDS/SQL-openUTM-Anwendung

BATCH

Es handelt sich um eine UDS/SQL-TIAM-Anwendung

date Zeitstempel für Beginn des Vorgangs in UDS/SQL:

jjjj-mm-tt hh.mm.ss

mem Umfang des SQL-spezifischen Memory in Kbyte, maximal vierstellig

time Zeitraum der Inaktivität in Minuten, maximal vierstellig

n Anzahl der noch nicht ausgegebenen inaktiven Vorgänge, maximal vierstellig

Wenn die Anzahl der Stellen des jeweiligen Ausgabefeldes zu klein ist, wird anstatt des Wertes der String '>>..>' ausgegeben.

Diejenigen Vorgänge, die seit mindestens *z* Minuten inaktiv sind, werden geordnet nach steigender Dauer ihrer Inaktivität ausgegeben.

Es werden nur so viele Vorgänge ausgegeben, wie auf den Bildschirm passen. Die Ausgabe wird ggf. abgeschlossen durch die Anzahl der noch nicht ausgegebenen inaktiven Vorgänge.

Diese Vorgänge können Sie mit dem DAL-Kommando CONTINUE ausgeben.

4.4.17 Anzahl Nachrichten an den UDS/SQL-DBH ausgeben (%DML)

%DML

Das %DML-Kommando gibt die Anzahl der Nachrichten an den UDS/SQL-DBH aus. Gezählt werden nicht nur die DML-Anweisungen, die von COBOL-DML, CALL-DML, SQL oder KDBS kommen, sondern auch mögliche Aufträge der Laufzeitsysteme und von UDSCON.

4.4.18 Datenbanken, Realms und Kennwörter ausschließen (DROP)

```
DROP { DB=dbname |
       RN=realmname [ , DB=dbname ] |
       PW=kennwort |
       ADM=admkenwort }
```

dbname

Name der Datenbank

realmname

Name des Realms

kennwort

kennwort darf bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C' *xxxx*':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X' *nnnnnnnnri*:

nnnnnnnn sind ein bis acht Sedezimalziffern

d:

d'ist eine Dezimalzahl mit höchstens acht Ziffern und Vorzeichen, deren Wert zu einem Binärwert umgesetzt wird

Es gilt die BS2000-Syntax (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-PASSWORD)

admkenwort

admkenwort kann bis zu vier byte lang sein und wird folgendermaßen dargestellt:

C' *xxxx*':

xxxx sind ein bis vier alphanumerische Zeichen und Sonderzeichen

X' *nnnnnnnnri*:

nnnnnnnn sind ein bis acht Sedezimalziffern

d:

d'ist eine Dezimalzahl mit höchstens acht Ziffern und Vorzeichen, deren Wert zu einem Binärwert umgesetzt wird

Es gilt die BS2000-Syntax (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-PASSWORD)

Mit dem Kommando DROP kann der Datenbankadministrator Datenbanken, Realms und Kennwörter aus einer laufenden Session ausschließen.

Datenbank ausschließen

DROP DB=*dbname*

Ausschließen der genannten Datenbank. Alle beteiligten Realms erreichen einen Konsistenzpunkt. Damit wird die aktuelle DB-Konfiguration verändert.

Wenn die angegebene Datenbank nicht angeschlossen ist, und für sie kein Auftrag zum Anschließen vorliegt, wird dieses Kommando sofort zurückgewiesen.

Wenn ein Auftrag zum Anschließen eines Datenbankexemplars (Original oder Kopie) vorliegt, das derzeit nicht angeschlossen ist, so wird dieser Auftrag durch DROP DB widerrufen.

Wenn die Datenbank ausgeschlossen wird, werden die Anzahl der DML-Anweisungen pro Datenbank und die Anzahl der Ein- und Ausgaben pro Datenbank nach SYSOUT ausgegeben.

Beispiele

1. DROP DB für die VERSAND-Datenbank

```
//EXECUTE-DAL-CMD CMD=DROP DB=VERSAND
% UDS0347 UDS-ADMINISTRATION LOGON = UAD@0E9P (OPDM239,12:58:52/0E9X)
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DROP DB=VERSAND (OPOX073,12:58:52/0E9X)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:58:52/0E9X)
//EXECUTE-DAL-CMD CMD=PERFORM
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: PERFORM (OPOX073,12:58:52/0E9X)
% UDS0206 UDS HAT DAS KOMMANDO ANGENOMMEN (OPCC012,12:58:52/0E9X)
% UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER CONFBSPL STARTED (OPCC309,12:58:53/0E9X)
% UDS0722 UDS AUFTRAG DROP DB=VERSAND WIRD AUSGEFUEHRT (OPCC283,12:58:53/0E9X)
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK (OPCC758,12:58:53/0E9X)
0E9X: DATABASE NAME          DMLS    LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
0E9X: -----
0E9X: VERSAND                0        65         57          3           3
% UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER CONFBSPL TERMINATED (OPCC309,12:58:55/0E9X)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:58:56/0E9X)
```

2. DROP DB für mehrere Datenbanken

```
//EXECUTE-DAL-CMD CMD=DROP DB=VERSAND
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DROP DB=VERSAND (OPOX073,12:59:00/0E9X)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:59:00/0E9X)
//EXECUTE-DAL-CMD CMD=DROP DB=KUNDEN
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DROP DB=KUNDEN (OPOX073,12:59:00/0E9X)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:59:00/0E9X)
//EXECUTE-DAL-CMD CMD=PERFORM
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: PERFORM (OPOX073,12:59:00/0E9X)
% UDS0206 UDS HAT DAS KOMMANDO ANGENOMMEN (OPCC012,12:59:00/0E9X)
% UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER CONFBSPL STARTED (OPCC309,12:59:00/0E9X)
% UDS0722 UDS AUFTRAG DROP DB=VERSAND WIRD AUSGEFUEHRT (OPCC283,12:59:03/0E9X)
% UDS0722 UDS AUFTRAG DROP DB=KUNDEN WIRD AUSGEFUEHRT (OPCC283,12:59:03/0E9X)
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK (OPCC758,12:59:03/0E9X)
0E9X: DATABASE NAME          DMLS    LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
0E9X: -----
0E9X: VERSAND                0        65         50          3           3
0E9X: KUNDEN                 0        29         22          2           2
% UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER CONFBSPL TERMINATED (OPCC309,12:59:03/0E9X)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:59:03/0E9X)
```

Realm ausschließen

DROP RN=*realmname*
[,DB=*dbname*]

Der DBH notiert einen Auftrag zum Ausschließen des mit *realmname* angegebenen Realms bzw. streicht einen vorliegenden ADD-Auftrag. Der Realm erreicht vor dem Ausschließen einen Konsistenzpunkt.

Die Angabe von DB=*dbname* ist nur nötig, wenn der Realm-Name nicht innerhalb der gesamten DB-Konfiguration eindeutig ist. Sie spezifiziert dann die gewünschte Datenbank.

Das DROP RN-Kommando kann nicht ausgeführt werden, wenn mit PP DBNAME=*dbname*,SHA die Datenbank für ändernde Zugriffe gesperrt wurde. Zwischen den DBHs, die auf diese Datenbank parallel zugreifen, findet keine Kommunikation statt.

Das Ausschließen von Realms ist sinnvoll:

- bei Geräte-Engpässen:
Realms können in konsistentem Zustand sowohl logisch als auch physisch ausgeschlossen werden.
- bei Hardwarefehlern, bei denen der DBH den Realm nicht automatisch ausschließt:
Der Datenbankadministrator kann dann nach dem DROP-Kommando mit anschließendem PERFORM parallel zur laufenden Session mit BMEND den Realm rekonstruieren. Dazu müssen alle zur Reparatur nötigen ALOG-Dateien vorhanden sein. Wenn die gerade aktuelle ALOG-Datei der zugehörigen Datenbank zur Reparatur gebraucht wird, muss zusätzlich ein CHECKPOINT-Kommando gegeben werden.

Die Aufträge zu DROP DB und DROP RN werden erst ausgeführt, wenn ihre Ausführung mit dem PERFORM-Kommando angestoßen wird.

Das Ausschließen eines Realm ist erst dann ausfallsicher im DBDIR protokolliert, wenn der Auftrag komplett abgearbeitet wurde. Ausfallsicher heißt, es überdauert Session-Unterbrechungen und das DBH-Ende.

Das DROP-Kommando ist ein direktes Gegenstück zum ADD-Kommando. Ein vorangegangenes ADD-Kommando kann solange durch DROP widerrufen werden, wie die Bearbeitung des ADD-Auftrags noch nicht durch PERFORM eingeleitet wurde.

 Der Realm DBDIR kann nicht ausgeschlossen werden.

Kennwort für Dateien ausschließen

DROP PW=*kennwort*

Ausschließen des genannten Datei-Kennworts aus der Menge der Kennwörter.

Bei DROP PW prüft der DBH, ob das angegebene Datei-Kennwort bereits bekannt ist. Wenn dies der Fall ist, streicht der DBH das angegebene Datei-Kennwort.

Das DROP PW-Kommando wird von DBH zurückgewiesen, wenn das angegebene Datei-Kennwort nicht bekannt ist.

Datei-Kennwörter sollten Sie erst ausschließen, nachdem Sie die entsprechenden Dateien ausgeschlossen haben (siehe [Abschnitt „Kennwörter für UDS/SQL-Dateien vergeben“](#)).

Kennwort für die Administration ausschließen

DROP ADM=*admkenwort*

Ausschließen des genannten Datenbankadministrator-Kennworts.

Bei DROP ADM prüft der DBH, ob das angegebene Datenbankadministrator-Kennwort bereits vorhanden ist. Wenn dies der Fall ist, streicht der DBH das angegebene Kennwort. Das DROP ADM-Kennwort wird vom DBH zurückgewiesen, wenn das angegebene Kennwort nicht bekannt ist.

Die Kommandos DROP PW und DROP ADM werden sofort ohne PERFORM ausgeführt.

4.4.19 Einträge in der Verteiltabelle löschen (&DROP DISTRIBUTION)

Für UDS-D

```
&DROP DISTRIBUTION,
  { ALL |
    NODE=prozessorname , CONF=konfname [ , DB=dbname ] [ , ALL ] |
    DB=dbname , SS=subschemaname }
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*; er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht '@' sein. Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

dbname

Name einer Datenbank;
er darf max. 17 Zeichen lang sein; er muss netzweit eindeutig sein.

subschemaname

Name eines Subschemas;
er darf max. 30 Zeichen lang sein;
er muss in den ersten sechs Zeichen netzweit eindeutig sein.

Sie löschen Einträge in der Verteiltabelle (siehe [Abschnitt „Aufbau der Verteiltabelle“](#)).

&DROP DISTRIBUTION,ALL

Mit diesen Operanden löschen Sie sämtliche Einträge. Die Verteiltabelle ist nach Ausführung des DAL-Kommandos leer. Eine neue Verteiltabelle erstellen Sie, indem Sie eine neue Eingabedatei einlesen mit &ADD-DISTRIBUTION,FILE=...

&DROP DISTRIBUTION, NODE=...,CONF=...[,ALL]

Mit diesen Operanden löschen Sie den angegebenen NODE-/CONF-Eintrag.
Mit dem Zusatz ALL löschen Sie zusätzlich alle mit dem NODE-/CONF-Eintrag verketteten DB- und SS-Einträge.

&DROP DISTRIBUTION, NODE=...,CONF=...,DB=...[,ALL]

Mit diesen Operanden löschen Sie den angegebenen DB-Eintrag, falls er mit dem angegebenen NODE-/CONF-Eintrag verkettet ist oder mit keinem NODE-/CONF-Eintrag verkettet ist.

Mit dem Zusatz ALL löschen Sie zusätzlich alle mit dem DB-Eintrag verketteten SS-Einträge.

&DROP DISTRIBUTION,
DB=...,SS=...

Mit diesen Operanden löschen Sie den angegebenen SS-Eintrag, falls er mit dem angegebenen DB-Eintrag verkettet ist oder mit keinem DB-Eintrag verkettet ist.

Beispiel

Es existiert an jeder Konfiguration folgende Zuordnung über die Verteiltabellen:

PROZ1	CONF1	DB1
PROZ2	CONF2	DB2
PROZ3	CONF3	DB3

Der Datenbankadministrator gibt an der Konfiguration CONF2 das DAL-Kommando

```
&DROP DISTRIBUTION ,NODE=PROZ1 ,CONF=CONF1 ,DB=DB1
```

Damit löscht er in der Verteiltabelle für die Konfiguration CONF2 den DB-Eintrag für die Datenbank DB1.

Das bedeutet, dass

- Anwenderprogramme, die mit `SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME=CONF2` gestartet wurden, nicht auf Subschemas der Datenbank DB1 zugreifen können.
- Anwenderprogramme, die mit einer anderen Konfiguration gestartet wurden, von der Änderung dieser Verteiltabelle nicht betroffen sind.

i Änderungen in der Verteiltabelle wirken sich erst bei der nächsten READY-Anweisung aus.

4.4.20 Speicherabzug erstellen (DUMP)

```
DUMP [ {ALL | STD | transaktionskennung } ]
```

Standardwert:

ALL

Abhängig vom angegebenen Operanden bewirkt das DUMP-Kommando Folgendes:

ALL gibt den gesamten vom DBH belegten Adressraum auf Datei aus; wenn dies abgeschlossen ist, wird die Bearbeitung normal fortgesetzt.

STD Es wird ein reduzierter Dump erzeugt, abhängig vom Systemzustand und der genutzten BS2000-Version. Falls aus Sicht von UDS/SQL ein reduzierter Dump nicht sinnvoll erscheint, wird ein vollständiger User-Dump erstellt. Nach Abschluss des Dumps wird die Bearbeitung normal fortgesetzt.

transaktionskennung

Diese Angabe wird nur noch aus Kompatibilitätsgründen unterstützt und wirkt wie STD.

Bei einem User-Dump wird die Datei in folgender Form angelegt:

```
[$userid.]DUMP.jobname.tsn.nnnnn
```

Bei einem reduzierten Dump (Area-Dump) wird die Datei in folgender Form angelegt:

```
[$userid.]SYS.ADUMP.jobname.tsn.nnnnn
```

Dabei bedeutet:

userid.

Aktuelle Kennung

jobname.

Name des Auftrags

tsn.

Prozessfolgennummer des Auftrags

nnnnn.

Folgennummer, fünfstellig mit führenden Nullen



Während ein Dump geschrieben wird, ignoriert UDS/SQL alle anderen DAL-Kommandos außer %TERM.

4.4.21 Speicherabzug für linked-in DBH erstellen (%DUMP)

Nur für linked-in DBH!

Gibt den gesamten vom DBH belegten Hauptspeicher mit aufbereiteten Tabellen auf Datei aus.

Das Programm wird nicht abgebrochen.

4.4.22 Online-DBTT-Erweiterung ausführen (EXTEND DBTT)

```
EXTEND DBTT ,DB=dbname ,RECR=recordref[ ,EXT=extnbr]
```

i Zwischen EXTEND und DBTT ist genau ein Leerzeichen anzugeben. Weitere Leerzeichen sind in diesem DAL-Kommando nicht erlaubt.

dbname

Name der Datenbank, zu der die Satzart gehört, für die die Online-DBTT-Erweiterung durchgeführt werden soll. Die Datenbank und der entsprechende Realm müssen zum Zeitpunkt des DAL-Kommandos bereits zugeschaltet sein und die Online-Realm-Erweiterung muss aktiviert sein.

recordref

Nummer der Usersatzart, für die die Online-DBTT-Erweiterung durchgeführt werden soll.

Die *recordrefs* der Satzarten können Sie mit dem DAL-Kommando DISPLAY DBTT-INCR anzeigen lassen oder Sie können sie dem BPSIA-Protokoll entnehmen.

recordref=1 wird UDS/SQL-intern genutzt und kann hier nicht angegeben werden.

extnbr

Anzahl der Einträge, um die die DBTT mindestens erweitert werden soll.

Die Angabe muss größer oder gleich 1 und kleiner oder gleich 99999999 sein.

Falls der Parameter nicht angegeben ist, wird die Erweiterungsgröße intern berechnet.

Durch das DAL-Kommando EXTEND DBTT wird unabhängig von einer Aktivierung der Online-DBTT-Erweiterung beim nächsten Einspeichern eines Satzes in dieser Satzart genau eine Online-DBTT-Erweiterung durchgeführt. Dabei ist Voraussetzung, dass die Online-Erweiterbarkeit des betreffenden Realms aktiviert ist und der Realm zugeschaltet ist. Die erfolgreiche Erweiterung wird mit der Meldung UDS0741 mitgeteilt.

Da das DAL-Kommando erst beim nächsten Einspeichern eines Satzes in dieser Satzart ausgeführt wird, hängt der Zeitraum zwischen dem Absetzen des Kommandos und der Durchführung der Erweiterung nur von der Einspeicherung neuer Sätze durch die aktuell aktiven Anwendungen ab.

Ob ein EXTEND-DBTT-Auftrag noch vorgemerkt ist, können Sie mit DISPLAY DBTT ermitteln.

Wird der Realm, in dem die DBTT enthalten ist, abgeschaltet, dann wird ein noch nicht ausgeführter EXTEND-DBTT-Auftrag ohne zusätzliche Meldung annulliert. Diese Annullierung wird insbesondere auch beim Abschalten der entsprechenden Datenbank sowie beim Sessionende des DBH durchgeführt.

Der EXTEND-DBTT-Auftrag wird beim nächsten Speichern eines Satzes ausgeführt, auch wenn es dabei bei aktivierter Online-DBTT-Erweiterung zeitnah zu zwei Erweiterungen kommen kann, weil parallel in einer anderen Transaktion aufgrund der aktivierten Online-DBTT-Erweiterung ebenfalls eine DBTT-Erweiterung ausgeführt wird.

Werden mehrere EXTEND-DBTT-Aufträge hintereinander abgesetzt, bevor der nächste Satz der entsprechenden Satzart gespeichert wird, so wird die DBTT-Erweiterung nur einmal ausgeführt und zwar mit der zuletzt angegebenen Erweiterungsgröße.

In der Regel wird die tatsächliche Anzahl der DBTT-Einträge, um die erweitert wird, etwas größer als die gewünschte Anzahl sein. Die tatsächliche Anzahl hängt von folgenden Faktoren ab:

- Seitengröße der Datenbank (2K, 4K oder 8K)
- DBTT-Zeilenlänge der jeweiligen Satzart

Abhängig von diesen Faktoren gibt es eine Obergrenze für die mögliche Erweiterungsanzahl, auf die bei größeren Angaben automatisch reduziert wird.

Falls es innerhalb der DBTT-Erweiterung zu Engpässen kommt, kann die Anzahl der Einträge, um die erweitert wurde, auch geringer als die gewünschte Anzahl sein. In jedem Fall wird die Anzahl der DBTT Einträge, um die tatsächlich erweitert wurde, mit entsprechender Meldung ausgegeben.

Scheitert ein vorliegender EXTEND-DBTT-Auftrag, weil die Online-Realm-Erweiterung suspendiert oder weil die Maximalgröße der DBTT erreicht ist, wird die Online-Erweiterung der betreffenden DBTT im aktuellen Sessionabschnitt ausgesetzt und es wird eine entsprechende Fehlermeldung ausgegeben. Diese Meldung wird bei jedem neuen EXTEND-DBTT-Auftrag wieder ausgegeben, unabhängig davon, ob in dem aktuellen Suspendierungsabschnitt bereits eine diesbezügliche Meldung ausgegeben wurde oder nicht.

Liegt ein EXTEND-DBTT-Auftrag vor, wenn die Online-Realm-Erweiterung des betreffenden Realms deaktiviert wird, so wird der EXTEND-DBTT-Auftrag ohne zusätzliche Meldung annulliert.

4.4.23 Online-Realm-Erweiterung ausführen (EXTEND REALM)

```
EXTEND REALM , DB=dbname , RR=realmref , EXT=nr-pages
```

dbname

Name der Datenbank des Realms, für den die Realmerweiterung durchgeführt werden soll.

realmref

Nummer des Realms, für den die Erweiterung durchgeführt werden soll. Zulässig sind alle Realm-Nummern außer derjenigen für den DBCOM.

nr-pages

Anzahl von Datenbankseiten, um die der Realm erweitert werden soll.

nr-pages=0, 64 ... 16777215

Beachten Sie bitte, dass gegebenenfalls bis zu 64 PAM-Seiten zusätzlich angefordert werden, falls zur Verwaltung der neu hinzukommenden Seiten ein oder mehrere neue FPA-Extents benötigt werden. Mit der speziellen Angabe *nr-pages*=0 kann ein vorheriger EXTEND REALM-Auftrag zurückgenommen werden, allerdings nur, wenn nicht schon mit der Ausführung begonnen wurde. Da nach einer derartigen Auftragsrücknahme kein EXTEND REALM mehr ansteht, wird keine weitere Meldung ausgegeben.

Mit dem Kommando EXTEND REALM können Sie bei der nächsten Freiplatzsuche eine einmalige Realmerweiterung um die angegebene Seitenzahl durchführen lassen, unabhängig davon, ob keine freien Seiten mehr oder nur noch weniger freie Seiten als der per ACT INCR spezifizierte Grenzwert verfügbar sind. Das Kommando erfordert keine PERFORM-Anweisung.

Für die erfolgreiche Ausführung des EXTEND REALM-Kommandos ist es notwendig, dass der Administrator dem Realm genügend Speicherplatz oder eine Sekundärzuweisung größer 0 zugewiesen hat.

Das Kommando kann auch dann ausgeführt werden, wenn die ONLINE-Realmerweiterung nicht mit ACT INCR aktiviert ist.

Nach der durchgeführten Erweiterung (ob mit oder ohne Erfolg) ist die unbedingte Realmerweiterung erledigt, eine evtl. mit ACT INCR aktivierte ONLINE-Realmerweiterung bleibt aber weiterhin aktiviert.

Wird ein EXTEND REALM-Kommando abgesetzt, solange für ein vorheriges EXTEND REALM-Kommando noch nicht mit der Ausführung begonnen wurde, so wird das vorherige EXTEND REALM-Kommando durch das neuere ersetzt, d.h. die Erweiterung findet mit der zuletzt angegebenen Größe statt.

Unabhängig von *nr-pages* erfolgt eine Realmerweiterung höchstens bis zur Anzahl maximal möglicher Datenbankseiten im Realm.

Die Ausführung des EXTEND REALM-Kommandos wird mit einer entsprechenden Meldung quittiert.

Das Kommando EXTEND REALM wird in den folgenden Fällen sofort zurückgewiesen:

- die Datenbank *dbname* ist nicht zugeschaltet.
- die Realm-Nummer *realmref* existiert nicht.
- als Realm-Nummer wurde die Realm-Nummer des DBCOM angegeben.
- der angegebene Realm ist nicht zugeschaltet.

4.4.24 SQL-Betriebsmittel freigeben (FORGET SQL)

```
FORGET SQL, VG=vg-nr
```

vg-nr Vorgangsnummer, maximal achtstellig;
intern von UDS/SQL vergebene Nummer des SQL-Vorgangs; Vorgänge werden während eines Session-Abschnitts fortlaufend nummeriert.

Der Vorgang mit der Nummer *vg-nr* wird aus dem UDS/SQL-Tabellenwerk gelöscht. Das Kommando wird ausgeführt, wenn in dem angegebenen Vorgang keine Transaktion mehr eröffnet, also der Vorgang inaktiv ist. Wenn in dem angegebenen Vorgang die TA gerade aktiv ist, so wird das DAL-Kommando für diesen Vorgang vorgemerkt, und bei TA-Ende ausgeführt.

Wenn der Vorgang trotz eröffneter Transaktion gelöscht werden soll, ist vorher ein ABORT-Kommando für die TA erforderlich. Wenn der Vorgang wegen eines DAL-Kommandos STOP NEW eine noch nicht eröffnete TA hat, kann der Vorgang nur nach dem DAL-Kommando GO gelöscht werden.

Siehe auch [Kapitel „Der SQL-Vorgang“](#).

4.4.25 STOP-Zustand von Transaktionen aufheben (GO)

```
GO { transaktionskennung | ALL | OLD }
```

Das GO-Kommando ist ein direktes Gegenstück zum STOP-Kommando, d.h. abhängig vom angegebenen Operanden setzt es bei einer oder mehreren Transaktionen, die sich im STOP-Zustand befinden, die Ausführung fort:

transaktionskennung

Kennung einer aktuellen Transaktion;
hebt den STOP-Zustand der angegebenen Transaktion auf

ALL hebt für alle Transaktionen den STOP-Zustand auf; d.h. die Ausführung bereits bekannter Transaktionen wird fortgesetzt, neue Transaktionen werden zugelassen und gestartet.

OLD hebt den STOP-Zustand nur für die Transaktionen auf, für die ein Anwenderprogramm bereits einen READY abgesetzt hat.

4.4.26 Einträge in der Verteiltabelle sperren (&LOCK DISTRIBUTION)

Für UDS-D

```
&LOCK DISTRIBUTION,
  {NODE=prozessorname |
  CONF=konfname |
  DB=dbname |
  SS=subschemaname }
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*;
er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht '@' sein.
Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0
werden nicht unterschieden.

dbname

Name einer Datenbank;
er darf max. 17 Zeichen lang sein, er muss netzweit eindeutig sein.

subschemaname

Name eines Subschemas;
er darf max. 30 Zeichen lang sein; er muss in den ersten sechs Zeichen netzweit eindeutig sein.

Sie können in der lokalen Verteiltabelle sperren:

- Verarbeitungsrechner
Alle betroffenen NODE-/CONF-Einträge werden gesperrt.
- Konfigurationen
Der betroffene NODE-/CONF-Eintrag wird gesperrt.
- Datenbanken
Der betroffene DB-Eintrag wird gesperrt.
- Subschemas
Der betroffene SS-Eintrag wird gesperrt.

READY-Anweisungen, die sich auf einen gesperrten Eintrag beziehen, weist UDS-D ab, nachdem es das DAL-Kommando &LOCK DISTRIBUTION bearbeitet hat.

Das Anwenderprogramm erhält den Statuscode 141.

Mit &LOCK DISTRIBUTION sperren Sie den angegebenen Eintrag.

Z.B., wenn Sie ein Subschema sperren, wird eine READY-Anweisung für dieses Subschema abgewiesen. Wenn Sie eine Konfiguration sperren, werden als Folge READY-Anweisungen für alle Subschemas dieser Konfiguration abgewiesen.

Bei lokaler Administration können Sie mit &LOCK DISTRIBUTION nur die lokale Verteiltabelle ändern.

i Änderungen in der Verteiltabelle wirken sich erst bei der nächsten READY-Anweisung aus.

Beispiel

Es existiert an jeder Konfiguration folgende Zuordnung über die Verteiltabellen:

PROZ1	CONF1	DB1
PROZ2	CONF2	DB2
PROZ3	CONF3	DB3

Der Datenbankadministrator gibt an der Konfiguration CONF2 das DAL-Kommando

```
&LOCK DISTRIBUTION, DB=DB1
```

Damit sperrt er in der Verteiltabelle für die Konfiguration CONF2 die Datenbank DB1.

Das bedeutet, dass

- Anwenderprogramme, die mit `SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=CONF2` gestartet wurden, nicht auf Subschemas der Datenbank DB1 zugreifen können.
- Anwenderprogramme, die mit einer anderen Konfiguration gestartet wurden, von der Änderung dieser Verteiltabelle nicht betroffen sind.

4.4.27 Einstellungen für DEFAULT-SUPPORT und RESERVE-SUPPORT der ALOG-Dateien ändern (MODIFY ALOG)

```
MODIFY {ALOG | ALOG-RES},DB=dbname[ ,VALUE={:catid: | PUBLIC}]
      [,SHARE={YES | NO}]
```

ALOG

Die Einstellungen für DEFAULT-SUPPORT der ALOG-Datei werden geändert.

ALOG-RES

Die Einstellungen für RESERVE-SUPPORT der ALOG-Datei werden geändert.

dbname

Name der Datenbank (max. 17 Zeichen)

:catid.

BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Die ALOG-Datei wird auf gemeinschaftlicher Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung angelegt.

SHARE=YES

Der Zugriff auf die Datei ist allen Benutzern erlaubt.

SHARE=NO

Der Zugriff auf die Datei ist nur dem Eigentümer erlaubt.

i Die SHARE-Angabe kann sowohl bei MODIFY ALOG als auch bei MODIFY ALOG-RES angegeben werden. In der Ausgabe des Kommandos DISPLAY ALOG wird jedoch nur der Wert für Default Support ausgegeben.

Beispiel

Siehe Beispiel zu MODIFY ALOG-SIZE.

4.4.28 Speicherplatzzuweisung der ALOG-Dateien ändern (MODIFY ALOG-SIZE)

```
MODIFY ALOG-SIZE ,DB=dbname ,VALUE=( [primär] [ , [sekundär] ]
```

dbname

Name der Datenbank (max. 17 Zeichen)

primär

Primärzuweisung in PAM-Seiten

primär = 192..50337645

sekundär

Sekundärzuweisung in PAM-Seiten

sekundär = 576..32767

Mit dem DAL-Kommando MODIFY ALOG-SIZE ändern Sie die Speicherplatzzuweisung der ALOG-Dateien. Sie können die Primär- und Sekundärzuweisung ändern. Das Kommando ist bei der Erzeugung der nächsten ALOG-Datei wirksam.

Beispiel

```
//EXECUTE-DAL-CMD CMD=MODIFY ALOG,DB=CUSTOMER,VALUE=PUB,SHARE=YES
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: MODIFY ALOG,DB=CUSTOMER,VALUE=PUB,
SHARE=YES
(OPOX073,12:28:04/0L9M)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:28:04/0L9M)
//EXECUTE-DAL-CMD CMD=MODIFY ALOG-SIZE,DB=CUSTOMER,VALUE=(300,600)
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: MODIFY ALOG-SIZE,DB=CUSTOMER,VALUE=
(300,600)
(OPOX073,12:28:04/0L9M)
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:28:04
/0L9M)
//EXECUTE-DAL-CMD CMD=DISPLAY ALOG
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY ALOG (OPOX073,12:28:04
/0L9M)
0L9M: USERID      DATABASE-NAME      DEF-SUP  RES-SUP  PRIM-ALL  SEC-ALL  SHARE
0L9M: -----
0L9M: $XXXXXXXXX.SHIPPING          IUDES    ABN2      192      576      NO
0L9M: $XXXXXXXXX.CUSTOMER        IUDES    IUDES      300      600      YES
0L9M:
0L9M: UDS/SQL V2.9    2 DATABASES IN CONFEXMP
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,12:28:04/0L9M)
```

4.4.29 Datenträgerzuweisung der RLOG-Dateien ändern (MODIFY LOG)

```

MODIFY {LOG | LOG-2},
      VALUE={:catid: |
            PUBLIC |
            (priv-vsn-1/gerät-1[,priv-vsn-2 gerät-2
            [,priv-vsn-3 gerät-3]])
            (vsn-1[,vsn-2[,vsn-3]])}

```

LOG Die Datenträger der RLOG-Originaldatei werden geändert.

LOG-2 Die Datenträger der RLOG-Duplikatdatei werden geändert.

:catid. BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Die RLOG-Datei oder das Duplikat der RLOG-Datei wird auf gemeinschaftlicher Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung angelegt.

priv-vsn

Die RLOG-Datei oder das Duplikat der RLOG-Datei wird auf Privatplatten angelegt. Es können maximal drei Archivnummern angegeben werden, die verschieden sein müssen. Eine Archivnummer darf bis zu sechs alphanumerische Zeichen enthalten.

gerät Gerätetyp der privaten Platte. Der Gerätetyp darf bis zu acht alphanumerische Zeichen enthalten.

vsn Die RLOG-Datei oder das Duplikat der RLOG-Datei wird auf Platten angelegt, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind. Es können maximal drei VSNs angegeben werden, die verschieden sein müssen.

Die VSN kann in PUB-Notation (PUBp_{xx}) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden. Bitte beachten Sie auch die [„Zusatzbedingungen für VSN-Angaben“](#).

Sie können maximal drei Datenträger angeben.

Ein DAL-Kommando darf 64 Zeichen lang sein. Bei langen Datenträgerangaben können Sie die Abkürzungen MOD und VAL verwenden. LOG-2 müssen sie ausschreiben, damit es eindeutig ist.

Mit dem DAL-Kommando MODIFY LOG und MODIFY LOG-2 können Sie die Datenträgerzuweisung für eine neu einzurichtende RLOG-Original- und Duplikatdatei ändern. Die Wertänderungen werden für einen Session-Wiederanlauf ausfallsicher in der Session-Log-File (SLF) hinterlegt.

Ob die RLOG-Datei einfach oder doppelt geführt werden soll, legen Sie beim DBH-Start fest. Dies können Sie nicht mit dem DAL-Kommando MODIFY ändern. Die Datenträger für die RLOG-Dateien können Sie mit dem DAL-Kommando MODIFY ändern.

Bei doppelt geführter RLOG-Dateien müssen die Original- und Duplikatdatei auf verschiedenen Datenträgern liegen. Der angegebene Datenträger wird gegen den vorhandenen geprüft, sind sie gleich, wird das MODIFY-Kommando zurückgewiesen.

Die Speicherplatzgröße der RLOG-Dateien können Sie mit dem DAL-Kommando MODIFY LOG-SIZE verändern.

Ersatzdatenträger für einfach und doppelt geführte RLOG-Dateien können Sie mit dem DAL-Kommando MODIFY RESERVE angeben.

Die neue Einstellung muss einem Pubset entsprechen, der innerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt. Bei einer MOD LOG-Anweisung zwischen einem akzeptierten DAL NEW PUBSETS und dem darauffolgenden PERFORM muss die neue Einstellung einem Pubset entsprechen, der auch innerhalb des Pubset-Raumes der neuen UDS/SQL-Pubset-Deklaration liegt. Sind diese Bedingungen nicht erfüllt, wird das MODIFY RESERVE-Kommando abgewiesen.

Zusatzbedingungen für VSN-Angaben

Für die VSN-Angaben bei Anlage der RLOG-Dateien auf Platten, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind, gelten folgende Besonderheiten:

- VSN-Angaben sind nur dann zulässig, wenn die Berechtigung zur physikalischen Allokierung von gemeinschaftlichem Speicherplatz auf dem betroffenen Pubset besteht.
- Sowohl in der PUB-Notation (PUBp_{xx}), als auch in der Punkt-Notation (pp[pp].[xy]z), ist es erlaubt, die laufende Nummer der Platte (xx bzw. [xy]z) wegzulassen. Dann ist nur eine einzige VSN-Angabe möglich.

Dadurch kann erreicht werden, dass die RLOG-Datei auf beliebigen Volumes eines Volume-Sets eines SM-Pubsets angelegt wird.

Beispiele:

(PUBX01, PUBX02)	RLOG-Datei auf Volumes PUBX01 und PUBX02 im einstelligen SF-Pubset X oder Volume-Set X eines SM-Pubsets
(PUBX)	RLOG-Datei auf beliebigem Volume im einstelligen SF-Pubset X oder Volume-Set X eines SM-Pubsets
(ABC.01, ABC.02)	RLOG-Datei auf Volumes ABC.01 und ABC.02 im dreistelligen SF-Pubset ABC oder Volume-Set ABC eines SM-Pubsets
(ABC.)	RLOG-Datei auf beliebigem Volume im dreistelligen SF-Pubset ABC oder Volume-Set ABC eines SM-Pubsets

- Falls für eine RLOG-Datei mehrere Volumes angegeben werden, müssen diese demselben Volume-Set angehören, d.h. die VSNs müssen die gleiche Katalogkennung enthalten. Eine Datei kann sich nicht über mehrere Volume-Sets erstrecken.
- Doppelt geführte Logging-Dateien müssen auf unterschiedlichen Volume-Sets angelegt werden, da beim Ausfall eines Volumes in der Regel der gesamte Volume-Set nicht mehr zur Verfügung steht.
- Außerdem darf eine der doppelt geführten Logging-Dateien auch nicht auf einem Volume-Set eines SM-Pubset angelegt werden, wenn die andere auf Grund einer :catid:- oder PUBLIC-Angabe in diesem SM-Pubset angelegt wird.
- Bei der Angabe des Control Volume-Sets eines SM-Pubsets für ein RLOG-Exemplar oder zweier Volume-Sets eines SM-Pubsets für RLOG-Original und -Duplikat ist Folgendes zu beachten:

Bei Ausfall des Control Volume-Sets eines SM-Pubsets ist vorübergehend der gesamte SM-Pubset nicht mehr zugreifbar. (Eine Rekonstruktion des SM-Pubsets ist möglich.)

4.4.30 Speicherplatzgröße der RLOG-Dateien ändern (MODIFY LOG-SIZE)

```
MODIFY LOG-SIZE,VALUE=( [primär] [ , [sekundär] ] )
```

primär

Primärzuweisung in PAM-Seiten

primär = 1..9999999

sekundär

Sekundärzuweisung in PAM-Seiten

sekundär = 1..32767

Für das Kommando LOG-SIZE gilt die übliche Abkürzungsregel der ersten drei Buchstaben nicht. Diese sind nicht eindeutig. Sie müssen mindestens LOG-S angeben.

Mit dem DAL-Kommando MODIFY LOG-SIZE ändern Sie die Speicherplatzgröße der RLOG-Dateien, die mit dem Ladeparameter PP LOG-SIZE festgelegt wurde. Sie können die Primär- und Sekundärzuweisung ändern.

Die Wertänderungen werden für einen Session-Wiederanlauf ausfallsicher in der Session-Log-File (SLF) hinterlegt.

Wenn Sie für die Sekundärzuweisung einen kleineren Wert als den Standardwert 192 angeben, wird auf ihn aufgerundet.

4.4.31 PTCSYNCH-Werte ändern (MODIFY PTCSYNCH)

```
MODIFY PTCSYNCH,
  VALUE=( [{WAIT | ABORT | COMMIT} ]
          [, [{WAIT | ABORT | COMMIT} ]])
```

Für UDS-D/openUTM

Der **erste** Wert der Anweisung MODIFY PTCSYNCH wirkt auf sekundäre Teiltransaktionen bzw. openUTM-Transaktionen, die sich bei einem **Warmstart** im Zustand PTC befinden.

Für UDS-D

Der **zweite** Wert der Anweisung MODIFY PTCSYNCH wirkt auf sekundäre Teiltransaktionen, die sich während der **laufenden Session** im Zustand PTC befinden und für die der Zustand der zugehörigen primären Teiltransaktion nicht ermittelbar ist.

WAIT

Die sekundären Teiltransaktionen bzw. openUTM-Transaktionen warten, bis sie beendet werden, entweder durch eine Nachricht der zugehörigen primären Teiltransaktion bzw. von openUTM oder explizit durch den Datenbankadministrator mit dem DAL-Kommando ABORT,OPTION=PTC oder COMMIT.

ABORT

UDS/SQL setzt die sekundären Teiltransaktionen bzw. openUTM-Transaktionen zurück (FINISH WITH CANCEL) und gibt dem Datenbankadministrator eine Warnung aus.

COMMIT

UDS/SQL beendet die sekundären Teiltransaktionen bzw. openUTM-Transaktionen mit Festschreiben der Änderungen (FINISH) und gibt dem Datenbankadministrator eine Warnung aus.

Mit dem DAL-Kommando MODIFY PTCSYNCH können Sie die Werte des DBH-Ladeparameters PP PTCSYNCH dynamisch verändern. Dadurch können Sie sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC beenden. Wenn Sie einen der beiden Werte nicht angeben, bleibt der aktuelle Wert unverändert.

Ein Verändern des Standardwerts (WAIT) für Warmstart kann z.B. erforderlich sein, wenn der Datenbankadministrator eine Datenbank mit Warmstart anschließen möchte, auf der eine sekundäre Teiltransaktion bzw. eine openUTM-Transaktion im Zustand PTC ist.

Indem der Datenbankadministrator den PTCSYNCH-Wert von WAIT auf COMMIT oder ABORT setzt, erzwingt er ein Anschließen dieser Datenbank.

i Wenn der PTCSYNCH-Wert nicht auf (WAIT, WAIT) steht (Standardwert), ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

Die physische Konsistenz innerhalb einer Konfiguration bleibt unabhängig davon gesichert.

4.4.32 Ersatzdatenträger für RLOG-Dateien ändern (MODIFY RESERVE)

```

MODIFY RESERVE,
  VALUE={NONE |
         :catid: |
         PUBLIC |
         (priv-vsn-1/gerät-1[ ,priv-vsn-2 gerät-2
          [ ,priv-vsn-3 gerät-3]]) |
         (vsn-1[ ,vsn-2[ ,vsn-3]])}

```

:catid: BS2000-Katalogkennung (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#))

PUBLIC

Gemeinschaftliche Platte (PUBLIC VOLUME) unter der Standard-Katalogkennung der Konfigurationskennung.

priv-vsn

Die RLOG-Datei wird auf die Privatplatten angelegt, die als Ersatzdatenträger angegeben worden sind. Es können maximal drei Archivnummern angegeben werden, die verschieden sein müssen. Eine Archivnummer darf bis zu sechs alphanumerische Zeichen enthalten.

gerät Gerätetyp der privaten Platte. Der Gerätetyp darf bis zu acht alphanumerische Zeichen enthalten.

vsn Als Ersatzdatenträger für die RLOG-Datei werden Platten festgelegt, die einem SF-Pubset oder einem Volume-Set eines SM-Pubsets zugeordnet sind. Es können maximal drei VSNs angegeben werden, die verschieden sein müssen.
Die VSN kann in PUB-Notation (PUBpxx) oder in Punkt-Notation (pp[pp].[xy]z) angegeben werden. Bitte beachten Sie auch die „Zusatzbedingungen für VSN-Angaben“ im [Abschnitt „Datenträgerzuweisung der RLOG-Dateien ändern \(MODIFY LOG\)“](#).

Sie können maximal drei Datenträger angeben.

Ein DAL-Kommando darf 64 Zeichen lang sein. Bei langen Datenträgerangaben können Sie die Abkürzungen MOD, RES und VAL verwenden.

Mit dem DAL-Kommando MODIFY RESERVE ändern Sie die Ersatzdatenträger für neu einzurichtende Dateien, oder legen keine Ersatzdatenträger fest. Die Ersatzdatenträger wurden mit dem Ladeparameter PP RESERVE festgelegt.

Die geänderten Datenträger werden für einen Session-Wiederanlauf ausfallsicher in der Session-Log-File SLF hinterlegt.

Als Ersatzdatenträger müssen Sie andere Datenträger bestimmen, als die aktuell für PP LOG und PP LOG-2 gültigen. Dies wird geprüft, wenn der Reservedatenträger benötigt wird.

Die neue Einstellung muss einem Pubset entsprechen, der innerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt. Zwischen einem akzeptierten DAL NEW PUBSETS und dem darauffolgenden PERFORM muss die neue Einstellung einem Pubset entsprechen, der auch innerhalb des Pubset-Raumes der neuen UDS/SQL-Pubset-Deklaration liegt. Sind diese Bedingungen nicht erfüllt, wird das MODIFY RESERVE-Kommando abgewiesen.

4.4.33 Neue UDS/SQL-Pubset-Deklaration prüfen und vormerken (NEW PUBSETS)

NEW PUBSETS

Mit dem Kommando NEW PUBSETS wird eine neue UDS/SQL-Pubset-Deklaration vormerkbar. UDS/SQL prüft erneut, ob eine gültige UDS/SQL-Pubset-Deklaration in der Jobvariablen mit dem LINK-Namen UDSPS01 vorliegt. Die Prüfung erfolgt unmittelbar durch das Kommando NEW PUBSETS. Die geprüfte UDS/SQL-Pubset-Deklaration wird jedoch erst beim nächsten PERFORM (ggf. auch intern ausgelöst) wirksam.

Läuft der DBH im Dialog, dann kann die neue UDS/SQL-Pubset-Deklaration durch die Zuweisung einer anderen Jobvariablen bekannt gemacht werden, in der Praxis, bei einer im Hintergrund laufenden Mastertask, dürfte jedoch nur ein Verändern des Jobvariablen-Inhalts relevant sein.

Die Änderung der aktuellen UDS/SQL-Pubset-Deklaration mit DAL NEW PUBSETS wird abgewiesen, falls die neue UDS/SQL-Pubset-Deklaration nicht die Pubsets umfasst, die für LOG, LOG-2, und RESERVE des RLOG-Logging sowie für DEFAULT-SUPPORT und RESERVE-SUPPORT des ALOG-Logging jeder zugeschalteten Datenbank vorgegeben wurden.

Die Ausgabe nach einer erfolgreichen Prüfung hat folgendes Layout:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0746 UDS-PUBSET-DEKLARATION (PENDING) FOLGT (... ,tsn4)
tsn4: UDS-PUBSET-JV: name der jv
tsn4: PUBSETS: catidgroup_1
tsn4: PUBSETS: catidgroup_2
tsn4: PUBSETS: ...
Je einmal für jede in der UDS/SQL-Pubset-Deklaration angegebene Catid-Gruppe.
tsn4: DEFAULT PUBSET: <Standard-Katalogkennung der Ablaufkennung>
tsn4: -----
UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (... ,tsn4)
```

Falls keine UDS/SQL-Pubset-Deklaration zugewiesen ist, wird die Catid-Gruppe "*" verwendet. Zur Unterscheidung von einer explizit angegebenen Catid-Gruppe "*" erfolgt die Ausgabe:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0747 UDS-PUBSET-DEKLARATION NICHT VORHANDEN, DEFAULT IST CATID * (PENDING) (... ,tsn4)
UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (... ,tsn4).
```

Die Zuweisung einer UDS/SQL-Pubset-Deklarations-Jobvariable, die nur Leerzeichen enthält, ist zulässig; es wird der Default Public Volume Set der Ablaufkennung berücksichtigt.

Falls bei der Prüfung der UDS/SQL-Pubset-Deklaration ein Fehler erkannt wird, bleibt die bisher wirksame UDS/SQL-Pubset-Deklaration weiterhin wirksam.

Ein Kommando DAL NEW PUBSETS (auch ein fehlerhaftes) löscht ein seit dem letzten PERFORM evtl. vorhergegangenes Kommando DAL NEW PUBSETS.

Fehlerhafte Zuweisung einer Pubset-Deklarations-Jobvariable

Die Zuweisung einer nicht vorhandenen oder nichtzugreifbaren Jobvariable oder einer Jobvariable der Länge 0 wird folgendermaßen abgewiesen:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0752 UDS-ANWENDERFEHLER: ZUGRIFF ZUR UDS-PUBSET-JV UEBER JV-LINKNAMEN UDSPS01 SCHEITERTE (... ,tsn4)
UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN ERROR IN UDS PUBSET DECLARATION (... ,tsn4)
```

Zusätzliche Hinweise zur Fehlerursache geben ggf. weitere Meldungen.

Inhaltliche Fehler der UDS/SQL-Pubset-Deklaration werden abhängig von der Art des Fehlers und dem Zeitpunkt der Fehlererkennung wie folgt gemeldet:

- Syntaxfehler, die bei der syntaktischen Analyse (Erkennung der Catid-Gruppen) erkannt wurden:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0748 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX): <ursache> (... ,tsn4)
```

Dabei kann <ursache> folgende Werte annehmen:

- CATID GROUP TOO LONG
- TOO MANY CATID GROUPS
- EXCLUDE CATID GROUP NOT ALONE

```
tsn4: UDS-PUBSET-JV: Name der Jobvariable
tsn4: UDS-PUBSET-JV-CONTENTS:
tsn4: Inhalt der Jobvariable, ggf. über mehrere Zeilen verteilt
tsn4: -----
UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN
ERROR IN UDS PUBSET DECLARATION (... ,tsn4)
```

- Während der Prüfung der Catid-Gruppen mit dem SDF-Makro CMDWCC erkannte Fehler:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0749 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX): CATID GROUP REJECTED BY
CMDWCC (... ,tsn4)
tsn4: UDS-PUBSET-JV: name der Jobvariable
tsn4: PUBSETS: Catid-Gruppe
```

Für jede Catid-Gruppe, die vom Makro CMDWCC beanstandet wurde und zum Scheitern führte, max. 10 Mal.

Gibt es mehr als 10 fehlerhafte Catid-Gruppen, werden nur die ersten 10 protokolliert und mit folgender Zeile angezeigt, dass weitere fehlerhafte Catid-Gruppen vorhanden sind:

```
tsn4: FURTHER ERRORS NOT SHOWN
tsn4: -----
UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN
ERROR IN UDS PUBSET DECLARATION (... ,tsn4)
```

- Während der Prüfung der Catid-Gruppen mit FSTAT erkannte Fehler:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0749 UDS-ANWENDERFEHLER IN DER UDS-PUBSET-DEKLARATION (SYNTAX): CATID GROUP REJECTED BY
FSTAT (... ,tsn4)
tsn4: UDS-PUBSET-JV: name der jv
tsn4: PUBSETS: Catid-Gruppe, FSTAT-DMS-RC: xxxx
```

Für jede Catid-Gruppe, die vom FSTAT beanstandet wurde und zum Scheitern führte. Die Catid-Gruppe wird in der tatsächlichen Länge ausgegeben, das Zeilenformat ist daher variabel.

```
tsn4: -----
UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN
ERROR IN UDS PUBSET DECLARATION (... ,tsn4)
```

- Die neue UDS/SQL-Pubsetdeklaration umfasst nicht die Pubsets, die für LOG, LOG-2, RESERVE des RLOG-Logging sowie für DEFAULT-SUPPORT und RESERVE-SUPPORT des ALOG-Logging jeder zugeschaltete DB vorgegeben wurden:

```
UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: NEW PUBSETS (... ,tsn4)
UDS0755 UDS-ANWENDERFEHLER: CATID FEHLT IN UDS-PUBSET-DEKLARATION (NEW): <loginfo>: (... ,
tsn4)
```

Die Meldung wird ggf. mehrfach ausgegeben.

<loginfo> gibt an, welche Katalogkennung für welches Loggingobjekt fehlt:

- :catid: , RLOG
- :catid: , RLOG-2
- :catid: , RLOG-RESERVE
- :catid: , ALOG-DEFAULT ,DB: <dbname>
- :catid: , ALOG-RESERVE ,DB: <dbname>

```
UDS0209 UDS-ANWENDERFEHLER: DAS KOMMANDO WIRD ABGEWIESEN
ERROR IN UDS PUBSET DECLARATION (... ,tsn4)
```

4.4.34 Neue RLOG-Dateien auswählen (NEW RLOG)

```
NEW RLOG
```

Mit dem Kommando NEW RLOG wird beim DBH der Auftrag hinterlegt, die momentan eröffnete RLOG-Datei zu schließen, eine neue RLOG-Datei anzulegen und sie zu eröffnen. NEW RLOG wird erst ausgeführt, wenn dieser Auftrag mit PERFORM angestoßen wird.

NEW RLOG muss vom Datenbankadministrator eingesetzt werden, wenn es dem DBH nicht gelingt, eine RLOG-Datei auf den Platten einzurichten, die mit den Ladeparametern PP LOG, PP LOG-2 und PP RESERVE zugewiesen wurden. Dann werden UPDATE-Transaktionen und verteilte Transaktionen gesperrt. Diese Sperre ist nur durch NEW RLOG aufzuheben.

Mit NEW RLOG kann der Datenbankadministrator den DBH veranlassen, die RLOG-Datei zu wechseln.

4.4.35 Vorgemerkte Aufträge ausführen (PERFORM)

```
PERFORM {NOCANCEL | CANCEL}
```

Standardwert:

NOCANCEL

NOCANCEL

Alle offenen Transaktionen, die die Ausführung der vorliegenden Aufträge behindern, werden normal beendet.

Anschließend werden die vorgemerkten Aufträge ausgeführt.

CANCEL

Alle offenen Transaktionen, die die Ausführung der vorliegenden Aufträge behindern, werden durch Rollback zurückgesetzt.

Vorgemerkte Aufträge resultieren aus folgenden DAL-Kommandos:

ACT INCR, ACT DBTT-INCR, ADD DB, ADD RN, CHECKPOINT, DEACT INCR, DEACT DBTT-INCR, DROP DB, DROP RN, NEW PUBSETS, NEW RLOG.

Das DAL-Kommando PERFORM bewirkt, dass der DBH alle vorgemerkten Aufträge durchführt, sobald das mit Rücksicht auf den Konsistenz-Aspekt und die offenen Transaktionen möglich ist. Hierzu müssen zunächst alle offenen Transaktionen, die eine konsistente Durchführung der vorliegenden Aufträge **behindern**, beendet werden. Diese Transaktionen werden bei DISPLAY USERS angezeigt. Neue störende Transaktionen werden mit einem Statuscode abgewiesen.

Während des Setzens eines Checkpoints werden alle neuen UPDATE-Transaktionen mit dem Statuscode 12122 bzw. 12124 zurückgewiesen (der Statuscode 12124 wird im Fall von PP ORDER-DBSTATUS=SPECIAL ausgegeben).

Neue SQL-Transaktionen können als RETRIEVAL-Transaktionen ablaufen. SQL-UPDATE-Anweisungen werden abgewiesen (SQL-CODE -810).

Nach der Angabe von PERFORM werden folgende Kommandos zurückgewiesen, solange die Aufträge nicht fertig bearbeitet sind:

ACT, ADD, CHECKPOINT, DEACT, DROP, NEW RLOG.

Eine zuvor mit DAL NEW PUBSETS geprüfte fehlerfreie UDS/SQL-Pubset-Deklaration wird beim nächsten (DAL oder DBH-internen) PERFORM wirksam. Dies wird mit folgender Meldung angezeigt:

```
UDS0754 UDS WECHSELT ZUR NEUEN PUBSET DECLARATION
```

Eine genaue Beschreibung zur Abarbeitung der DAL-Kommandos finden Sie im Abschnitt „Ausführung der Kommandos“ im [Abschnitt „Die Datenbankadministratorsprache DAL“](#).

i Wenn Sie PERFORM NOCANCEL angegeben haben, kann der Fall eintreten, dass durch langlaufende Transaktionen die Bearbeitung der Aufträge hinausgezögert wird. Um lange Wartezeiten zu vermeiden, haben Sie die Möglichkeit, nachträglich PERFORM CANCEL zu geben.

Hinweis für UDS-D/openUTM

Bei UDS-D/openUTM können auch verteilte Transaktionen die Durchführung der Aufträge hinauszögern, z.B. wenn verteilte Transaktionen im Zustand PTC geblieben sind. Solche Transaktionen werden vom PERFORM CANCEL-Kommando nicht beeinflusst, um die konfigurationsübergreifende Konsistenz bzw. UDS/SQL-openUTM-Konsistenz nicht zu gefährden. In solchen Fällen sollten Sie sich über das DAL-Kommando DISPLAY die Zustände der aktuellen Transaktionen ausgeben lassen und notfalls mit ABORT oder COMMIT eingreifen.

4.4.36 Kennwort vergeben und verändern (&PWD DISTRIBUTION)

Für UDS-D

```
&PWD DISTRIBUTION,CONF=konfname,
  {PWN=kennwortneu |
  PWO=kennwortalt |
  PWO=kennwortalt,PWN=kennwortneu}
```

Standardwert:

kein Kennwort

konfname

Name der Konfiguration, der das Kennwort zugeordnet werden soll.

kennwortneu

Neues Kennwort für die Konfiguration *konfname*.

kennwortalt

Bestehendes Kennwort der Konfiguration *konfname*.

PWN=kennwortneu

Festlegen eines neuen Kennworts.

In die Verteiltabelle wird das neue Kennwort eingetragen. Dies ist nur möglich, wenn für die angegebene Konfiguration kein Kennwort besteht.

Maximale Länge: acht Zeichen.

Ein Leerzeichen gilt als Ende des Kennworts. Eine längere Eingabe wird abgeschnitten. Sinnvoll sind lesbare Zeichen, dies wird jedoch nicht geprüft.

PWO=kennwortalt

Löschen des bestehenden Kennworts. In der Verteiltabelle wird das alte Kennwort mit Leerzeichen überschrieben.

PWO=kennwortalt,PWN=kennwortneu

Ändern des bestehenden Kennworts. In der Verteiltabelle wird das alte Kennwort in das neue geändert.

Geben Sie für *kennwortalt* ein Kennwort an, das nicht mit dem bestehenden Kennwort übereinstimmt, so wird das Kommando abgewiesen und eine Fehlermeldung ausgegeben.

Mit dem Kennwort für UDS-D können Sie Ihre DB-Konfiguration schützen. Auf die Daten dieser Konfiguration kann nur von solchen Konfigurationen zugegriffen werden, die das Kennwort kennen. Das Kennwort wird in der Verteiltabelle abgelegt (siehe auch [Abschnitt „Konfigurationsbezogener Kennwortschutz“](#)).

4.4.37 Online-Erweiterbarkeit für einen Realm reaktivieren (REACT INCR)

```
REACT INCR ,DB=dbname [ ,RR=realmref ]
```

i Zwischen REACT und INCR ist genau ein Leerzeichen anzugeben. Weitere Leerzeichen sind in diesem DAL-Kommando nicht erlaubt.

dbname

Name der Datenbank des Realms bzw. der Realms, für den bzw. für die Online-Erweiterbarkeit reaktiviert werden soll.

Die Datenbank muss zum Zeitpunkt der Kommandoausführung zugeschaltet sein.

Die Datenbank darf sich nicht im Modus SHARED-RETRIEVAL befinden.

realmref

Nummer des Realms, für den die Online-Erweiterbarkeit reaktiviert werden soll. Der Realm muss zum Zeitpunkt der Kommandoausführung zugeschaltet sein. Die Nummern der zugeschalteten Realms können Sie sich mit DISPLAY INCR anzeigen lassen.

Zulässig sind die Nummern aller Benutzerrealms und des DBDIR. Die Angabe der Nummer des DBCOM ist nicht zulässig.

In bestimmten Fehlerfällen wird durch den DBH für einen Realm eine Aussetzung der Online-Erweiterbarkeit veranlasst (siehe "[Verhalten im Fehlerfall](#)"). Die Online-Erweiterbarkeit als permanente, potenzielle Eigenschaft des Realms bleibt dabei aktiviert.

Mit dem Kommando REACT INCR können Sie für den Realm *realmref* der Datenbank *dbname* die Aussetzung der Online-Erweiterbarkeit aufheben. Ist RR=*realmref* nicht angegeben, so wird das Kommando für alle gerade zugeschalteten Benutzerrealms und den DB-DIR der Datenbank *dbname* ausgeführt.

REACT INCR wird sofort ausgeführt. Es ist kein PERFORM erforderlich.

Falls für den oder die angegebenen Realms die Online-Erweiterbarkeit nicht ausgesetzt ist, wird das Kommando ignoriert.

Wird die Aussetzung einer Online-Realm-Erweiterung aufgehoben, dann führt dies auch dazu, dass Online-DBTT-Erweiterungen nun wieder möglich sind.

Verhalten in Fehlersituationen

Das Kommando REACT INCR wird sofort mit Meldung UDS0209 zurückgewiesen,

- wenn die Datenbank *dbname* nicht zugeschaltet oder im Modus SHARED-RETRIEVAL zugeschaltet ist.
- wenn die angegebene Realm-Nummer *realmref* nicht existiert.
- wenn als Realm-Nummer *realmref* diejenige des DBCOM angegeben wurde.
- wenn der angegebene Realm nicht zugeschaltet ist.

4.4.38 Vorgemerkte Aufträge zurücknehmen (RESET ORDERS)

```
RESET ORDERS
```

Mit diesem DAL-Kommando können vorgemerkte Aufträge, deren Ausführung mit PER-FORM angestoßen wurde, zurückgenommen werden (siehe DAL-Kommando „PERFORM“ auf "[Vorgemerkte Aufträge ausführen \(PERFORM\)](#)"). Das ist möglich, solange der DBH mit der Ausführung der Aufträge noch nicht begonnen hat.

Nach der Rücknahme der vorgemerkten Aufträge ist der Transaktionsbetrieb wieder ungehindert möglich. Ebenso ist die Eingabe von neuen Aufträgen über DAL-Kommandos wieder möglich.

Die Funktion ist vor allem für solche Fälle sinnvoll, wo nach der Eingabe des PERFORM-Kommandos festgestellt wird, dass störende Transaktionen vorliegen, die nicht durch ROLLBACK zurückgesetzt werden sollen und die das Ausführen der Aufträge über eine längere Zeit verzögern.

Das Kommando wirkt nicht auf vorliegende Aufträge zur Aktivierung oder Deaktivierung der Online-Erweiterbarkeit von Realms oder DBTTs (ACT INCR, DEACT INCR, ACT DBTT-INCR, DEACT DBTT-INCR), weil diese Aufträge nicht durch offene Transaktionen behindert werden.

4.4.39 Verteiltabelle sichern (&SAVE DISTRIBUTION)

Für UDS-D

```
&SAVE DISTRIBUTION,FILE=[ :catid: ][$userid.]dateiname
```

:catid.

BS2000-Katalogkennung

userid

Benutzerkennung in der *dateiname* stehen soll.

dateiname

Name einer SAM- oder ISAM-Datei.

Hier ist die Angabe von *:catid.* zulässig (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#)).

Sie sichern den aktuellen Stand der Verteiltabelle in die angegebene Datei. Falls die angegebene Datei existiert, wird sie überschrieben.

UDS-D bereitet den Inhalt der Verteiltabelle so auf, dass die gesicherte Datei so aufgebaut ist wie eine Eingabedatei für die Verteiltabelle (siehe [Abschnitt „Aufbau der Eingabedatei für die Verteiltabelle“](#)).

Die gesicherte Datei enthält dann Datenbanksätze und Subschemasätze.

Wenn *dbname*, *konfname*, *prozessorname* oder *subschemaname* gesperrt sind, haben sie ein Minuszeichen vorangestellt.

UDS-D sichert nicht die folgenden Einträge der Verteiltabelle:

- Kennwörter
- SS-Einträge, die mit keinem DB-Eintrag verkettet sind,
- DB-Einträge, die mit keinem NODE-/CONF-Eintrag verkettet sind.

4.4.40 UDS-D-Betrieb starten (&START DISTRIBUTION)

Für UDS-D

```
&START DISTRIBUTION
```

Sie starten den UDS-D-Betrieb, d.h. Sie starten die UDS-D-Task UDSCT, wenn

- der DBH mit dem DBH-Ladeparameter PP DISTRIBUTION=STANDBY gestartet wurde
- der DBH mit dem DBH-Ladeparameter PP DISTRIBUTION=START gestartet wurde, aber der UDS-D-Betrieb inzwischen mit dem DAL-Kommando &CLOSE DISTRIBUTION beendet wurde
- der UDS-D-Betrieb abnormal beendet wurde (siehe [Abschnitt „UDS-D-Betrieb erneutstarten“](#)).

Wenn der UDS-D-Betrieb gestartet wurde, sind verteilte Transaktionen möglich.

4.4.41 Ausführung von Transaktionen stoppen (STOP)

```
STOP {transaktionskennung | NEW | ALL}
```

Das STOP-Kommando stoppt, abhängig vom angegebenen Operanden, die Ausführung einer oder mehrerer Transaktionen. Die bezeichneten Transaktionen versetzt es in den STOP-Zustand.

Der STOP-Zustand einer Transaktion kann durch folgende Ereignisse wieder aufgehoben werden:

- GO-Kommando
- CANCEL-Kommando für die Transaktion
- CLOSE-Kommando
- PERFORM-Kommando

transaktionskennung

Kennung einer aktuellen Transaktion;
stoppt die angegebene Transaktion

NEW lässt zukünftige Transaktionen zu, startet sie aber nicht

ALL stoppt alle gegenwärtig bekannten Transaktionen und lässt zukünftige Transaktionen zu, ohne sie jedoch zu starten

Gestoppt wird eine Transaktion erst dann, wenn sie eine DML-Anweisung abgibt. Wenn der DBH gerade eine DML-Anweisung dieser Transaktion bearbeitet, so wird diese Anweisung bis zum Ende abgearbeitet; erst wenn die Transaktion die nächste DML-Anweisung abgibt, wird sie gestoppt.

Zur Vermeidung von globalen Deadlocks werden neue Transaktionen aus UTM-Anwendungen und in sekundären Teiltransaktionen mit Statuscodes abgewiesen.

4.4.42 Sekundäre Teiltransaktionen im Zustand PTC beenden (&SYNCHRONIZE DISTRIBUTION)

Für UDS-D

`&SYNCHRONIZE DISTRIBUTION`

Mit &SYNCHRONIZE DISTRIBUTION versucht der Datenbankadministrator z.B. nach Rechner- oder Leitungszusammenbrüchen, alle sekundären Teiltransaktionen zu beenden, die sich im Zustand PTC befinden, um die Wartezeiten der sekundären Teiltransaktionen zu verkürzen. Wartezeiten können entstehen, wenn der Wert des DBH-Ladeparameters PP PTCSYNCH für die laufende Session auf WAIT steht.

Um eine sekundäre Teiltransaktion im Zustand PTC zu beenden, muss beim DBH der primären Teiltransaktion erfragt werden, ob die primäre Teiltransaktion mit FINISH oder mit FINISH WITH CANCEL beendet wurde, damit die sekundäre Teiltransaktion gleichermaßen beendet werden kann.

Wenn die Konfiguration der primären Teiltransaktion nicht erreichbar ist, werden sofort alle sekundären Teiltransaktionen, für die der Zustand der primären Teiltransaktion nicht ermittelbar ist, gemäß dem eingestellten PTCSYNCH-Wert für die laufende Session behandelt (siehe [Abschnitt „Überwachen sekundärer Teiltransaktionen, die im Zustand PTC sind“](#)).

Ohne &SYNCHRONIZE DISTRIBUTION reagiert UDS-D nach Ablauf einer Zeitspanne, die abhängig ist von PP CHCKTIME, gemäß des eingestellten PTCSYNCH-Wertes für die laufende Session.

4.4.43 Session sofort abbrechen (%TERM)

%TERM

Mit dem %TERM-Kommando bricht der DBH unter Umgehung der normalen Terminierungsaktivitäten die Session auf dem kürzesten Weg ab (Nothalt).

UDS/SQL gibt wahlweise einen reduzierten oder einen kompletten Speicherabzug aus (siehe „PP DUMP“ auf ["Umfang eines Dumps beeinflussen \(DUMP\)"](#)).

i %TERM sollten Sie nur dann verwenden, wenn die Mastertask wegen eines Fehlers keine anderen DAL-Kommandos mehr akzeptiert. Die Session wird auf jeden Fall abnormal beendet. Alle momentan inkonsistenten Datenbanken der Konfiguration brauchen bei ihrem nächsten Anschließen einen Warmstart.

4.4.44 Sperren für Einträge in der Verteiltabelle aufheben (&UNLOCK DISTRIBUTION)

Für UDS-D

```
&UNLOCK DISTRIBUTION,
  {NODE=prozessorname |
   CONF=konfname |
   DB=dbname |
   SS=subschemaname }
```

prozessorname

Name eines Verarbeitungsrechners;
er darf max. acht Zeichen lang sein.

konfname

Die ersten acht Zeichen von *konfigurationsname*;
er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht '@' sein.
Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0
werden nicht unterschieden.

dbname

Name einer Datenbank;
er darf max. 17 Zeichen lang sein; er muss netzweit eindeutig sein.

subschemaname

Name eines Subschemas;
er darf max. 30 Zeichen lang sein; er muss in den ersten sechs Zeichen netzweit eindeutig sein.

Sie können in der Verteiltabelle Sperren aufheben für

- Verarbeitungsrechner
- Konfigurationen
- Datenbanken und
- Subschemata

In der lokalen Verteiltabelle gelten die Einträge als gesperrt, die Sie mit dem DAL-Kommando &LOCK DISTRIBUTION oder in der Eingabedatei für die Verteiltabelle mit einem vorangestellten Minuszeichen gesperrt haben (siehe [Abschnitt „Aufbau der Eingabedatei für die Verteiltabelle“](#)).

Mit &UNLOCK DISTRIBUTION heben Sie die Sperre für den angegebenen Eintrag auf. Bei lokaler Administration können Sie mit &UNLOCK DISTRIBUTION nur die lokale Verteiltabelle ändern.

Beispiel

Wenn Sie die Sperre für ein Subschema aufheben, werden READY-Anweisungen für dieses Subschema wieder akzeptiert. Wenn Sie die Sperre für eine Konfiguration aufheben, werden READY-Anweisungen für Subschemata dieser Konfiguration wieder akzeptiert, es sei denn, Datenbanken oder Subschemata sind explizit gesperrt.

i Änderungen in der Verteiltabelle wirken sich erst bei der nächsten READY-Anweisung aus.

Beispiel

Es existiert an jeder Konfiguration folgende Zuordnung über die Verteiltabellen:

PROZ1	CONF1	DB1
PROZ2	CONF2	DB2
PROZ3	CONF3	DB3

Der Datenbankadministrator gibt an der Konfiguration CONF2 das DAL-Kommando

```
&UNLOCK DISTRIBUTION, DB=DB1
```

Damit hebt er in der Verteiltabelle für die Konfiguration CONF2 die Sperre für die Datenbank DB1 auf.

Das bedeutet, dass

- Anwenderprogramme, die mit `SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=CONF2` gestartet wurden, jetzt auf Subschemas der Datenbank DB1 zugreifen können, außer Subschemas der Datenbank DB1 sind explizit gesperrt.
- Anwenderprogramme, die mit einer anderen Konfiguration gestartet wurden, von der Änderung dieser Verteiltabelle nicht betroffen sind.

5 Hochverfügbarkeit

In UDS/SQL verwaltete Daten sind in der Regel für das nutzende Unternehmen geschäftskritisch. Sie müssen daher in besonderem Maße gegen Datenverlust geschützt werden und unterliegen sehr hohen Verfügbarkeitsanforderungen. UDS/SQL bietet daher eine Vielzahl von Funktionen, mit denen – teilweise in Zusammenarbeit mit anderen Produkten des BS2000 – diesen Anforderungen Rechnung getragen wird.

5.1 Unterbrechungsfreier Datenbank-Betrieb

Zur Vermeidung von Unterbrechungen des Datenbankbetriebs können Sie viele Konfigurationseinstellungen der Datenbanken im laufenden Betrieb ändern:

- Sie können Datenbanken zu- oder abschalten. Hierfür stehen Ihnen die DAL-Kommandos ADD DB und DROP DB zur Verfügung.
- Ebenso können Sie einzelne Realms zu- oder abschalten. Verwenden Sie hierfür die Kommandos ADD RN bzw. DROP RN. Für notwendige Reparatur- bzw. Strukturierungsmaßnahmen ist es somit in vielen Fällen nicht erforderlich, die ganze Datenbank abzuschalten. Die Behinderung des Betriebs kann dadurch minimiert werden.
- Sie können Datenträgerzuweisungen von neu einzurichtenden RLOG-Dateien ändern. Hierfür stehen Ihnen die DAL-Kommandos MODIFY LOG, MODIFY LOG-2 und MODIFY LOG-SIZE zur Verfügung.
- Sie können Ersatzdatenträger für RLOG-Dateien ändern, indem Sie das Kommando MODIFY RESERVE anwenden.
- Sie können Konfigurationsparameter der verteilten Verarbeitung ändern. Hierfür stehen Ihnen die DAL-Kommandos &ADD, &CHANGE, &DROP, &LOCK, &PWD, &SAVE, &SYN und &UNLOCK zur Verfügung.
- Sie können ALOG- und RLOG-Dateien wechseln. Für diesen Zweck stehen Ihnen die Kommandos DAL CHECKPOINT und NEW RLOG zur Verfügung.
- Wenn durch umfangreiche Einfügungen der Freiplatz in einem Realm aufgebraucht ist oder einen bestimmten Schwellwert unterschreitet, kann der DBH den Realm automatisch während des Betriebs erweitern (siehe [Kapitel „Ressourcen-Erweiterung und Reorganisation im laufenden Betrieb“](#)).
- Sie können eine bestehende UDS-SQL-Pubset-Deklaration ändern, die Änderung mit dem Kommando DAL NEW PUBSETS vormerken und sie anschließend mit DAL PERFORM aktivieren.

Wenn beim Einspeichern neuer Sätze der Höchstwert für die Anzahl Sätze einer Satzart erreicht ist, kann der DBH die DBTT automatisch während des Betriebs erweitern (siehe [Kapitel „Ressourcen-Erweiterung und Reorganisation im laufenden Betrieb“](#)).

5.2 Zusammenwirken mit dem Transaktionsmonitor openUTM

Bei einem Wiederanlauf einer openUTM-Anwendung mit UDS/SQL wird eine Synchronisierung der in den Datenbanken vorhandenen Daten mit denen der Anwendung automatisch durchgeführt. Damit wird die Konsistenz und Integrität der Daten sichergestellt. Dies wird insbesondere dann gewährleistet, wenn während der Durchführung einer Transaktion Störungen, wie z.B. Netzunterbrechung oder Systemausfall, auftreten.

Bei der Beendigung einer openUTM-Anwendung mit UDS/SQL aufgrund von Störungen wird eine koordinierte Beendigung der gerade aktiven Transaktionen soweit möglich durchgeführt, um den später notwendigen Wiederanlauf auf das unbedingt Notwendige zu beschränken. Belegte Betriebsmittel werden sofort freigegeben, damit diese von anderen Anwendungen umgehend wieder verwendet werden können.

5.3 Strukturierung der Datenbanken

Ein wesentliches Konzept von UDS/SQL besteht darin, die Daten einer Datenbank auf mehrere Dateien, sogenannte Realms zu verteilen. Dadurch wird die Verfügbarkeit der mit UDS/SQL verwalteten Daten erhöht, denn die Auswirkungen von Gerätefehlern oder Schreib-/Lese Fehlern können so auf Untereinheiten der Datenbank eingegrenzt werden.

Dienstprogramme benötigen nicht immer alle Realms zur Durchführung von Änderungen. Hierdurch können Einschränkungen eines parallelen DBH-Betriebes minimiert werden.

Sie können Kopien von Datenbanken auf mehreren lokalen oder entfernten Servern erstellen. Die Aktualisierung der Replikat erfolgt mit den ALOG-Dateien der Original-Datenbank. Damit steht Ihnen nach einem Ausfall des Original-Datenbestandes schnell ein Ersatz-Datenbestand zur Bearbeitung zur Verfügung. Ein weiterer Vorteil dieser verteilten Datenhaltung besteht darin, dass mit den Replikaten auch räumlich verteilter lesender Zugriff beschleunigt werden kann (Filialkonzept).

5.4 Fehlertoleranz

Wird in einem Anwendungsprogramm ein Fehler erkannt, bleibt die Wirkung lokal. Die betroffene Transaktion wird zurückgesetzt. Bei schwereren Fehlern wird das Anwendungsprogramm beendet. Andere Transaktionen werden insofern nicht behindert.

Bei STXIT-Ereignissen in Anwendungsprogrammen, die zur Beendigung desselben führen, werden belegte Betriebsmittel sofort freigegeben, damit diese von anderen Anwendungen umgehend wieder verwendet werden können.

5.5 Verteilte Verarbeitung

Die Verteilung eines logisch zusammengehörenden Datenbestandes auf mehrere BS2000-Server ist in einem Rechnernetz möglich. Der Einsatz mehrerer Datenbank-Server reduziert zum einen deutlich die Belastung der einzelnen Server gegenüber der Belastung beim Einsatz eines einzelnen Datenbank-Servers, zum anderen wird durch verteilte Datenbanken der maximale Schaden bei Ausfall eines Datenbank-Servers begrenzt.

Sie können die Datenbankverteilung durch jederzeit veränderbare Tabellen steuern. Somit können Sie die Verteilung zu jedem Zeitpunkt dynamisch an veränderte Gegebenheiten anpassen.

Sie können die verteilte Verarbeitung entkoppelt von der Independent-DBH-Session starten und beenden. Insbesondere können in einer Independent-DBH-Session nacheinander mehrere UDS-D-Sessions gestartet werden. Hierfür stehen Ihnen die DAL-Kommandos &START und &CLOSE zur Verfügung.

Die administrative Behandlung von Transaktionen, die sich im Zustand PTC („prepare to commit“) befinden, kann dynamisch verändert werden. Damit ist auch im Falle globaler Störungen und bei eingeschränkter Verfügbarkeit von Netzkomponenten die vollständig autonome Administrierbarkeit („site autonomy“) gewährleistet (siehe [Abschnitt „PTCSYNCH-Werte ändern \(MODIFY PTCSYNCH\)“](#)).

5.6 Langzeitsicherung

Eine Schattendatenbank oder Backup-Datenbank ist eine Kopie der Originaldatenbank, auf der zeitversetzt alle Änderungen nachgezogen werden. Falls die Original-Datenbank zerstört ist, kann mit einer Schattendatenbank nach Aktualisierung durch die letzten Afterimages der Datenbankbetrieb schnell fortgesetzt werden (siehe [Abschnitt „Schattendatenbank“](#)).

Im laufenden Datenbank-Betrieb können Sie sogenannte Online-Sicherungen über B2000-Betriebssystemfunktionen, z. B. mit den BS2000-Dienstprogrammen ARCHIVE bzw. HSMS durchführen. Das Erstellen einer Sicherung ist somit ohne Beeinträchtigung des laufenden Betriebs möglich. Eine solchermaßen erstellte Online-Kopie ist im Allgemeinen inkonsistent, außer es wird durch Administrationsmaßnahmen dafür gesorgt, dass die Datenbank zu Beginn des Sicherungsvorgangs konsistent ist und während der gesamten Dauer der Sicherung keine Änderungen stattfinden.

Im Anschluss an einen mit BS2000-Mitteln durchgeführten Kopiervorgang kann die Kopie mithilfe der entsprechenden Log-Dateien aktualisiert und damit zu einer Schattendatenbank gemacht werden. Welche Log-Dateien für die Aktualisierung benötigt werden, ist UDS/SQL intern bekannt (siehe [Abschnitt „Online-Sicherung einer Datenbank“](#)).

5.7 Restartfunktionen

Eine der wichtigsten Aufgaben von UDS/SQL besteht darin, die Konsistenz und Integrität der Daten sicherzustellen. Dies muss insbesondere dann gewährleistet sein, wenn während eines Transaktionsprozesses Störungen, wie z. B. ein Systemausfall, auftreten. Bei einem Wiederanlauf führt UDS/SQL eine Synchronisierung der in den Datenbanken vorhandenen Daten durch und bringt diese in einen konsistenten Stand (DB-Warmstart).

Ein DB-Warmstart kann auch für einzelne Datenbanken isoliert durchgeführt werden.

Eine abgebrochene Independent-DBH-Session wird bei einem Neustart des DBH ohne gesonderte Konfigurationsmaßnahmen genau in dem Zustand fortgesetzt, in dem sie zum Zeitpunkt des Abbruchs war. Dies wird durch die Nutzung des Session-Log-File (SLF) ermöglicht, in das der DBH die DBH-Ladeparameter der aktuellen Konfiguration ablegt (siehe [Abschnitt „Session-Log-File \(SLF\)“](#)).

Über den Ladeparameter WARMSTART können Sie Einfluss auf die Warmstartdauer nehmen. Mit diesem Parameter wird gesteuert, in welchem Umfang die Blöcke der Log-Dateien bereits im laufenden Betrieb aus dem Puffer in die Datenbank zurück geschrieben werden. Bei WARMSTART=VERY-FAST werden die Seiten sehr häufig in die Datenbank geschrieben, wodurch sich die Dauer eines Warmstarts deutlich verkürzt.

5.8 Nutzung von BS2000-Standardverfahren zur Erhöhung der Datensicherheit

Beim Einsatz von UDS/SQL können alle wesentlichen BS2000-Verfahren zur Erhöhung der Datensicherheit bzw. Verfügbarkeit genutzt werden.

UDS/SQL arbeitet problemlos mit der TimeFinder-Funktion der Symmetrix-Plattensysteme zusammen. Dabei wird auch das Abtrennen und Anhängen eines BCV-Plattenspiegels ohne Unterbrechung des Datenbankbetriebs möglich. Je nach Zustand der Datenbank zum Zeitpunkt des Abtrennens, wird mit dem BCV entweder eine konsistente Kopie erstellt, oder eine inkonsistente „Online“-Kopie, die mithilfe von Log-Files auf einen konsistenten Stand gebracht werden muss.

Die BS2000-Funktion Dual Recording by Volume (DRV) kann ohne Einschränkung bei UDS/SQL-Dateien genutzt werden. Falls nach einem Systemabsturz eine Egalisierung der auf den DRV-Platten gespeicherten Daten erforderlich sein sollte, kann diese mit UDS/SQL-Mitteln, d. h. durch DBH und Dienstprogramme, zielgenauer und damit kürzer als durch das Standardverfahren des BS2000 durchgeführt werden (siehe [Abschnitt „Datenspiegelung für UDS/SQL nutzen“](#)).

Mit dem Produkt HIPLEX AF (Highly Integrated Systems Complex) können in einem Verfügbarkeits- und Lastverbund mehrerer BS2000-Server Anwendungen mit hoher Verfügbarkeit realisiert werden. Eine Ausfallzeit wird dabei durch eine automatische Ausfallerkennung und durch die Einsparung des Systemwiederanlaufs minimiert. Innerhalb eines solchen Verbundes garantiert UDS/SQL wegen seiner globalen Transaktionssicherung und der Wiederanlauffunktionen, dass bei einem Wiederanlauf auf einem anderen Rechner keine Daten verloren gehen und keine Inkonsistenzen auftreten können.

Zum Erstellen einer Datenbank-Sicherung können Sie die im BS2000 zur Verfügung stehenden Möglichkeiten nutzen, üblicherweise die BS2000-Dienstprogramme HSMS bzw. ARCHIVE. Mit beiden Dienstprogrammen ist es möglich, eine sogenannte „Online-Kopie“ einer Datenbank zu erzeugen. Bei diesem Verfahren wird die Datenbank während des laufenden Betriebs gesichert. Eine solche im Allgemeinen inkonsistente Sicherung kann parallel zum normalen Datenbank-Betrieb durch Nachfahren von ALOG-Dateien im offline-Modus konsistent gemacht werden.

5.9 Schutz vor Ausfall lokaler Betriebsmittel

Neben der Unterstützung von BS2000-Standardverfahren bietet UDS/SQL auch die Möglichkeit, durch eigenständige Funktionen die Datensicherheit und Verfügbarkeit bei Ausfall einzelner Betriebsmittel sicherzustellen.

Für hohe Verfügbarkeitsansprüche können in UDS/SQL die RLOG-Dateien, welche die Before- und After-Images jeder Änderung enthalten und zur Durchführung des DB-Warmstarts nach Ausfällen dienen, auch doppelt geführt werden. Diese Art der redundanten Datenhaltung stellt eine Alternative zur RAID Technik oder zu DRV dar. Falls RLOG-Dateien doppelt geführt werden, überprüft UDS/SQL soweit möglich automatisch, ob für die RLOG-Dateien zwei unabhängige Datenträger angegeben wurden. Ist dies nicht der Fall, wird das Einrichten der RLOG-Dateien verweigert.

Für die Speicherung von RLOG- und ALOG-Dateien können Ersatzdatenträger angegeben werden, die verwendet werden, wenn die eigentlich dafür vorgesehenen Datenträger ausfallen. Damit können Betriebsunterbrechungen wegen Störungen bei Logging-Datenträgern stark reduziert werden. Für die Administration von Ersatzdatenträgern stehen Ihnen der DBH-Ladeparameter PP RESERVE, das DAL-Kommando MODIFY RESERVE und die BMEND-Anweisung START-LOG zur Verfügung. Bei Einschränkungen in der Verfügbarkeit von aktuell zugewiesenen Logging-Dateien wechselt der DBH automatisch zu den zugewiesenen Ersatz-Datenträgern.



VORSICHT!

Datenbank- und Loggingdateien dürfen nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

5.10 Einsatz von UDS/SQL-Dienstprogrammen

Lesende Dienstprogramme können parallel zum DBH auf aktuell in Bearbeitung befindliche Datenbanken zugreifen.

Die UDS-Online-Utility bietet die Möglichkeit, einige Funktionen, die sonst teilweise auf ähnliche Weise durch andere Dienstprogramme realisiert sind, online (d.h. während des DBH-Betriebs) durchzuführen. Es werden folgende Funktionen in Form von DMLs angeboten, die eine Datenbank selbst oder deren Bearbeitung verändern:

- Mit der Online-DML RELOCATE verlagern Sie Sätze und Tabellen, mit dem Ziel, die Belegungsdichte der Datenseiten zu erhöhen. Des Weiteren können Sie einen Realm entlasten durch Verlagern von Seiten einer verteilbaren Liste in einen anderen Realm.
- Mit der Online-DML FPASCAN wird eine Möglichkeit geboten, freien Platz, der z.B. durch Löschen von Sätzen oder Tabellen frei geworden ist, bei einer Neueinspeicherung ab sofort wieder zu nutzen.
- Mit der Online-DML PREFRLM verändern Sie die Einstellung des Preferred-Realm für eine verteilbare Liste. In diesem Realm sucht der DBH beim Abspeichern von Membersätzen dieser verteilbaren Liste freie Seiten auf der untersten Tabellenstufe (Stufe-0).

Eine ausführliche Beschreibung der UDS-Online-Utility finden Sie im Handbuch „[Sichern, Informieren und Reorganisieren](#)“.

Mit dem Dienstprogramm ONLINE-PRIVACY, das sich wie ein normales Anwendungsprogramm an den Independent-DBH anschließt, können die Zugriffsrechte dynamisch in der laufenden Session geändert werden (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, ONLINE-PRIVACY).

Sie können die durch die Prüfung der physischen Konsistenz einer Datenbank verursachte Einschränkung des Datenbankbetriebs wesentlich verkürzen, indem sie mit dem Dienstprogramm BCHECK eine vorsorgliche Prüfung der physischen Konsistenz auf einer konsistenten Kopie der Datenbank durchführen. Die eigentliche Prüfung der physischen Konsistenz der Datenbank kann dann inkremental gegen die konsistente und geprüfte Kopie der Datenbank durchgeführt werden. Dabei werden nur die seither geänderten Daten in die Prüfung einbezogen und die Laufzeit der Prüfung wird gegenüber einer Gesamtprüfung erheblich verringert.

Mit dem in UDS/SQL integrierten Dienstprogramm UDSMON lassen sich eine Vielzahl von Informationen über den aktuellen Datenbankbetrieb abfragen. Dazu gehören auch Informationen über den Status der aktuell durchgeführten Transaktionen. Mit dieser Information können Sonderaktionen gezielt in Zeiten durchgeführt werden, in denen das Aufkommen an DMLs und Transaktionen gering ist.

Umfangreichere Reorganisationen des Datenbestandes und Änderungen in den Datenstrukturen können in der Regel nur offline durchgeführt werden. Um die Laufzeiten solcher Aktionen so gering wie möglich zu halten, werden dabei in den Dienstprogrammen eine Vielzahl von Optimierungen eingesetzt, die bei parallelem DBH-Betrieb nicht möglich wären. Auch wegen der in der Regel hohen Komplexität solcher Änderungen wird der höheren Verfahrenssicherheit des Offline-Betriebes an dieser Stelle eine Priorität gegenüber der Verfügbarkeit eingeräumt.

6 Ressourcen-Erweiterung und Reorganisation im laufenden Betrieb

UDS/SQL bietet die Möglichkeit, sowohl Benutzer-Realms und DBDIR als auch die DBTTs und damit die Anzahl der möglichen Sätze der Satzart eines User-Schemas im laufenden Datenbankbetrieb zu erweitern. Die Erweiterung der Realms und der DBTTs wird bei Bedarf durch den Independent DBH wie auch durch den Linked-in DBH automatisch durchgeführt. Durch diese im Folgenden als „Online-Realm-Erweiterung“ bzw. „Online-DBTT-Erweiterung“ bezeichnete Funktionalität können Unterbrechungen des Datenbankbetriebes vermieden werden. Dadurch wird eine sehr hohe Verfügbarkeit der in UDS/SQL verwalteten Daten ermöglicht (siehe auch [Kapitel „Hochverfügbarkeit“](#)).

Die Administration der Online-Erweiterung umfasst folgende Schritte:

- Systemvoraussetzungen der Online-Erweiterung prüfen bzw. einstellen (siehe [Abschnitt „Systemvoraussetzungen der Online-Erweiterung“](#)).
- Online-Realm-Erweiterung bzw. Online-DBTT-Erweiterung aktivieren und überwachen (siehe [Abschnitt „Freiplatzverwaltung bei Realm-Erweiterung“](#) und [Abschnitt „Online-DBTT-Erweiterung aktivieren“](#)).

Unabhängig von der Aktivierung der Online-Realm-Erweiterung und Online-DBTT-Erweiterung erweitern die ändernden Dienstprogramme bei Bedarf automatisch die Realms der bearbeiteten Datenbank bzw. die DBTTs der Satzarten in DBDIR und DBCOM (siehe [Abschnitt „Automatische Realm-Erweiterung durch Dienstprogramme“](#)).

Im laufenden Betrieb können auch Verlagerungen und Verdichtungen von abgespeicherten Daten durchgeführt werden (vgl. [Abschnitt „Online-Reorganisation mit der UDS-Online-Utility“](#))

6.1 Systemvoraussetzungen der Online-Erweiterung

Um die Funktionalität der Online-Erweiterung von Realms bzw. DBTTs sinnvoll nutzen zu können, müssen Sie vor der Aktivierung die hierfür relevanten Eigenschaften des Realms prüfen und gegebenenfalls anpassen.

Voraussetzung für die Aktivierung der Online-DBTT-Erweiterung ist, dass die Online-Realm-Erweiterung des Realms, in dem die DBTT liegt, aktiviert ist.

6.1.1 Speicherplatz zuweisen

Ein Realm, der online erweiterbar sein soll, muss zum Zeitpunkt der Erweiterung über genügend Speicherplatz verfügen. Diese Bedingung ist erfüllt, wenn der Realm zum Zeitpunkt der Erweiterung entweder über eine ausreichende Anzahl von Dateiseiten verfügt, oder wenn für den Realm die Möglichkeit der dynamischen Dateierweiterung gegeben ist.

Speicherplatz weisen Sie beim Anlegen der Realm-Datei mit dem BS2000-Kommando CREATE-FILE zu (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Datenbankaufbau vorbereiten). Mit dem Kommando MODIFY-FILE-ATTRIBUTES können Sie die Speicherplatzzuweisung nachträglich ändern.

Bei der Speicherplatzzuweisung ist zwischen der Primärzuweisung und der Sekundärzuweisung zu unterscheiden:

- Mit der Primärzuweisung wird für eine neue Datei eine feste Anzahl Dateiseiten reserviert. Für eine bereits existierende Datei wird mit der Primärzuweisung eine feste Anzahl Dateiseiten zusätzlich reserviert.
- Mit der Sekundärzuweisung bewirken Sie eine dynamische Erweiterbarkeit der Datei, falls ihr Wert größer 0 ist. Den Wert der Sekundärzuweisung können Sie entsprechend der jeweiligen Erfordernisse und Randbedingungen frei wählen. Er sollte auf den Umfang der Online-Realm-Erweiterung, den Sie bei der Aktivierung mit dem Kommando ACT INCR angeben, abgestimmt sein.

Die Zuweisung des Speicherplatzes für eine Realm-Datei muss aus DVS-Gründen offline erfolgen. Es ist daher ratsam, die aktuellen Sekundärzuweisungswerte der Realm-Dateien zu beobachten, um sie bei Bedarf anpassen zu können, wenn die Datenbank bzw. der Realm aus Wartungsgründen offline geschaltet ist.

Die Dateigröße eines Realms aus DVS-Sicht lässt keinen direkten Rückschluss auf die Anzahl der Datenbankseiten zu, da die Datei mehr PAM-Seiten umfassen kann, als tatsächlich für Datenbankseiten genutzt werden. Bei einer Realm-Erweiterung kommt es immer dann zu einer Dateierweiterung, wenn Datenbankseiten, die vom DBH dem Realm logisch hinzugefügt werden sollen, nicht innerhalb der Datei liegen. Der Umfang der Dateierweiterung hängt dabei ab von der Anzahl solcher Datenbankseiten und von der eingestellten Sekundärzuweisung.

6.1.2 DVS-Aspekte der Online-Realm-Erweiterung

Beachten Sie bitte folgende DVS-Aspekte einer Online-Realm-Erweiterung:

Die Dateierweiterung kann nur in ganzen Allokierungseinheiten erfolgen. Die Größe der Allokierungseinheiten wird vom Systemverwalter unter Berücksichtigung des Plattenformats festgelegt (z. B. 3, 4, 32 PAM-Pages).

Bei jeder erfolgten Dateierweiterung verdoppelt DVS den Wert für die Sekundärzuweisung bis zu einem Maximalwert. Dieser Maximalwert ist bestimmt durch den Klasse-2-Systemparameter DMMAXSC (Systemglobal für alle Privatplatten, Default für Pubsets) bzw. durch die folgenden Kommandos:

- ADD-MASTER-CATALOG-ENTRY, Operand MAXIMAL-ALLOCATION,
- MODIFY-MASTER-CATALOG-ENTRY, Operand MAXIMAL-ALLOCATION,
- MODIFY-PUBSET-SPACE-DEFAULTS, Operand MAXIMAL-ALLOCATION für Pubsets.

Falls die Erweiterung einer Realm-Datei an eine Grenze stößt, z. B. weil das Speicherplatzkontingent der Kennung ausgeschöpft ist, so erfolgt in manchen Fällen keine Teilzuweisung. Daher kann ein zu großer Wert für die Sekundärzuweisung eine Realm-Erweiterung um eine bestimmte Anzahl neuer Datenbank-Seiten verhindern, die bei einem kleineren Wert für die Sekundärzuweisung noch erfolgreich gewesen wäre.

Ist eine Realm-Datei, bei der die Online-Realm-Erweiterung genutzt werden soll, kopiert worden, so ist vor dem Zuschalten zur Session die aktuelle Einstellung der Sekundärzuweisung zu überprüfen. Dieses ist z. B. bei Rückgriff auf eine Sicherungskopie oder vor dem Nachfahren einer Schatten-DB zu empfehlen.

Häufige Dateierweiterungen eines Realms können zu einer Zersplitterung der Datei in viele Extents führen. Diesem Umstand sollten Sie durch entsprechende Kontrollen und Maßnahmen Rechnung tragen. Informationen zu diesem Problemkreis finden Sie in den Handbüchern „[BS2000 OSD/BC Einführung in das DVS](#)“ und „[SPACEOPT \(BS2000\) Optimierung und Reorganisation von Platten](#)“).

6.1.3 Erweiterbarkeit eines Realms einschätzen

UDS/SQL kann maximal 16777214 Datenbankseiten in einem Realm verwalten. Daraus ergeben sich folgende Grenzwerte für die maximalen Dateigrößen:

Datenbankseitengröße	Maximale Dateigröße in PAM-Seiten
2 KB	16777214
4 KB	33554428
8 KB	67108856

Tabelle 17: Grenzwerte für Dateigrößen

Folgende Möglichkeiten stehen Ihnen zur Verfügung, um sich über die logische Realm-Größe zu informieren:

- Mit dem DAL-Kommando `DISPLAY FPA` können Sie sich die aktuelle logische Realm-Größe `NR PAGES` und die Anzahl der darin enthaltenen freien DB-Seiten `NR FREE PAGES` ausgeben lassen
- Mit der Anweisung `DISPLAY REALM` des Dienstprogrammes `BSTATUS` können Sie sich die logische Realm-Größe (`SPACE USED BY REALM / TOTAL NR OF PAGES`) sowie ausführliche Informationen zum darin enthaltenen Freiplatz anzeigen lassen.
- Mit der Anweisung `DISPLAY PAGE ZERO` des Dienstprogramms `BPRECORD` können Sie sich die logische Realm-Größe (`FPA BASE / TOTAL NR PAGES`) anzeigen lassen.

Verwenden Sie das `BS2000`-Kommando `SHOW-FILE-ATTRIBUTES`, um Informationen über die physikalische Realm-Größe, d.h. über die Anzahl bereits reservierter PAM-Seiten, sowie um die aktuelle Einstellung der Sekundärzuweisung (`SECONDARY-ALLOCATION`) der Realm-Datei ausgeben zu lassen. Diese beiden Angaben sind auch bei einer offenen Realm-Datei aktuell.

6.1.4 Auswirkungen von Dienstprogrammen auf die Online-Erweiterbarkeit von Realms

Wenn Sie einen Realm, für den Sie die Online-Erweiterbarkeit aktiviert haben, mit dem Dienstprogramm BPGSIZE bearbeiten, müssen Sie auf eine Sekundärzuweisung größer 0 achten.

Bei der Ausführung des Dienstprogramms BALTER mit Erzeugen neuer Realms müssen Sie zusätzlich auf eine mögliche nachfolgende Aktivierung der Online-Erweiterbarkeit dieser Realms achten.

Alle anderen UDS/SQL-Dienstprogramme verändern die Werte, die Sie mit dem DAL-Kommando ACT INCR eingestellt haben nicht.

6.1.5 Datenbank-Layout-Version und Realm-Layout-Version

Die Datenbank-Layout-Version kennzeichnet die Struktur einer Datenbank; ebenso kennzeichnet die Realm-Layout-Version die Struktur eines Realms. Mit UDS/SQL V2.9 können Sie Datenbanken und Realms mit Layout-Version 004.00 und 005.00 bearbeiten. Mit der Anweisung DISPLAY PAGE ZERO des Dienstprogramms BPRECORD können Sie sich den ACT-KEY-0 eines Realms aufbereiten lassen. In dieser Aufbereitung wird u.a. im DBDIR die Datenbank-Layout-Version und in den übrigen Realms die Realm-Layout-Version ausgegeben.

6.1.6 Datenbankseitenformatierung bei Realm-Erweiterung

Die folgenden Informationen zur Datenbankseitenformatierung gelten sowohl für Online-Realm-Erweiterungen als auch für Realm-Erweiterungen, die Sie im Offline-Modus mit dem Dienstprogramm BREORG durchführen.

Bei einer Erweiterung des DBDIR werden die hinzukommenden Datenbankseiten grundsätzlich formatiert. Dies gilt auch für eine Erweiterung des DBCOM, die allerdings nur offline erfolgen kann.

Bei der Erweiterung eines Benutzer-Realms werden nur diejenigen neu hinzukommenden Datenbankseiten formatiert, auf die möglicherweise Probable Position Pointer (PPP) verweisen. Alle übrigen neu hinzukommenden Datenbankseiten werden einzeln bei der ersten Nutzung formatiert. Die Anzahl zu formatierender Datenbankseiten hat einen deutlichen Einfluss auf die Dauer einer Realm-Erweiterung. Ebenso wird durch die Anzahl zu formatierender Datenbankseiten das Anwachsen der ALOG-Datei stark beeinflusst, denn für jede zu formatierende Seite werden dort je nach Datenbankseitenlänge ein, zwei bzw. drei 4KB-Blöcke (bei 2-, 4-, 8KB-Datenbanken) geschrieben. Die Anzahl zu formatierender Datenbankseiten sollte daher möglichst klein gehalten werden.

Potenziell können Probable Position Pointer (PPP) auf Seiten außerhalb der Datenbank verweisen, wenn ein Realm mit dem Dienstprogramm BREORG (Anweisung MODIFY-REALM-SIZE) verkleinert wurde. Bei der Verkleinerung werden nämlich die einzelnen Probable Position Pointer (PPP) aus Gründen der Performance nicht aktualisiert. Deshalb sollten Sie im Anschluss an Realm-Verkleinerungen für sämtliche Benutzer-Realms BREORG-Läufe mit der Anweisung REORGANIZE-POINTERS durchführen.

PPPs können auch über die Funktion REORGPDP DML der UDS-Online-Utility während des laufenden Betriebs aktualisiert werden. Wenn alle PPPs aller Benutzer-Realme aktualisiert werden, kann das Formatieren von neuen Seiten während der Online-Realm-Erweiterung vermieden werden.

Wenn die höchste zu formatierende Seite eines Benutzerrealms (vgl. HIGHEST PAGE NR FOR FORMATTING in der BPRECORD-Ausgabe von DISPLAY PAGE ZERO) nicht größer als die Anzahl der Seiten dieses Realms ist, dann ist garantiert, dass keine Probable Position Pointer (PPP) mehr auf Seiten außerhalb der Datenbank verweisen. Bei einer Realm-Erweiterung müssen somit neu hinzukommende Datenbankseiten nicht formatiert werden, sodass eine Online-Realm-Erweiterung entsprechend performant ablaufen kann.

Wenn die höchste zu formatierende Seite eines Benutzerrealms noch nicht bekannt ist, genügt es einmal einen BREORG-Lauf mit einer Anweisung REORGANIZE-POINTERS für jeden Realm der Datenbank durchzuführen.

Sie können sich folgendermaßen darüber informieren, ob die Erweiterungsseiten eines Realms formatiert werden müssen:

- mit dem Dienstprogramm BPRECORD (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, BPRECORD).
- mit dem DAL-Kommando DISPLAY INCR (siehe Abschnitt „[DISPLAY INCR - Ausgabe von Informationen über die Online-Erweiterbarkeit aller zugeschalteten Realms in der DB-Konfiguration](#)“). Die Information, ob formatiert werden muss, wird für alle (zugeschalteten) Realms ausgegeben und Sie können so abschätzen, welche Auswirkungen die Aktivierung der Online-Realm-Erweiterung haben wird.

6.1.7 Freiplatzverwaltung bei Realm-Erweiterung

Die folgenden Informationen zu den Freiplatzverwaltungstabellen gelten sowohl für Online-Realm-Erweiterungen als auch für Realm-Erweiterungen, die Sie im Offline-Modus mit dem Dienstprogramm BREORG durchführen.

Bei einer Realm-Erweiterung können zusätzlich zur Basis-Freiplatzverwaltungstabelle (FPA-Basis) weitere Freiplatzverwaltungstabellen, sogenannte FPA-Extents, entstehen.

Jeder FPA-Extent ist 128KB groß. In 2KB-Datenbanken besteht er aus 64 Seiten, in 4KB-Datenbanken aus 32 Seiten und in 8KB-Datenbanken aus 16 Seiten. Mit dem Dienstprogramm BPRECORD können Sie sich Informationen zur FPA-Basis und den FPA-Extents ausgeben lassen.

Die Tatsache, dass in einem Realm FPA-Extents vorhanden sind, besagt jedoch nichts über die aktuelle Online-Erweiterbarkeit. Die FPA-Extents können durch einen BREORG-Lauf erzeugt worden sein, ohne dass der Realm jemals online erweiterbar war. Es ist auch möglich, dass der Realm in der Vergangenheit online erweiterbar war, diese Eigenschaft jedoch inzwischen wieder zurückgenommen wurde.

Mit dem Dienstprogramm BPGSIZE können Sie FPA-Extents entfernen. BPGSIZE fasst bei unveränderter Seitenlänge für jede Datei FPA-Basis und möglicherweise vorhandene FPA-Extents zu einer neuen FPA-Basis zusammen (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, BPGSIZE).

6.2 Online-Realm-Erweiterung

Sind die Systemvoraussetzungen gegeben, können Sie für einen Realm die Online-Erweiterbarkeit aktivieren. Verwenden Sie dafür das DAL-Kommando ACT INCR (siehe „Online-Erweiterbarkeit für einen Realm aktivieren (ACT INCR)“, [„Online-Erweiterbarkeit aktivieren \(ACT\)“](#)). Bei der Aktivierung müssen Sie zwei Werte angeben:

- Mit dem ersten Wert legen Sie den Umfang einer Online-Realm-Erweiterung fest, d. h. die Anzahl der Datenbankseiten, die dem Realm jeweils hinzugefügt werden sollen.
- Mit dem zweiten Wert bestimmen Sie die Anzahl freier Seiten, bei deren Unterschreitung eine Online-Realm-Erweiterung vorsorglich von UDS/SQL angestoßen werden soll.

Der Auftrag wird erst ausgeführt, nachdem das Kommando PERFORM abgesetzt wurde.

Die Online-Erweiterbarkeit des Realms bleibt so lange aktiv, bis das Kommando DEACT INCR für diesen Realm abgesetzt wird (siehe [Abschnitt „Online-Erweiterbarkeit deaktivieren \(DEACT\)“](#)).

Verwenden Sie das DAL-Kommando DISPLAY INCR oder die Dienstprogramme BPSIA bzw. BPRECORD, um Informationen über die Online-Erweiterbarkeit eines Realms zu erhalten.

Beim Neuaufbau von Datenbanken wird von den Dienstprogrammen BCREATE und BFORMAT die Online-Realm-Erweiterbarkeit automatisch für alle Realms und ggf. auch die Online-DBTT-Erweiterbarkeit für DBTTs in diesen Realms aktiviert, für die bei der Speicherplatzzuweisung eine Sekundärzuweisung > 0 festgelegt wurde, siehe hierzu [Abschnitt „Aktivierung der Online-Erweiterbarkeit beim Datenbankaufbau“](#).

6.2.1 Ablauf der Online-Realm-Erweiterung

Die eigentliche Erweiterung eines Realms wird vom DBH automatisch angestoßen, wenn für die Bearbeitung einer DML-Anweisung Platz in einem Realm benötigt wird, dessen Mindestanzahl freier Seiten unterschritten ist. Die Realm-Erweiterung wird innerhalb der User-Transaktion dieser DML durchgeführt. Die Antwortzeit der DML wird dadurch zwar vergrößert, insgesamt ist die Behinderung des Gesamtbetriebs durch diese Vorgehensweise aber gering.

Dadurch dass eine Online-Realm-Erweiterung vorsorglich angestoßen wird, wenn im Realm noch eine bestimmte Anzahl freier Datenbankseiten vorhanden sind, können parallel zur Durchführung der Realm-Erweiterung auch Transaktionen weiterlaufen, die freien Platz im Realm benötigen. Diese Transaktionen finden den von ihnen benötigten Speicherplatz im ursprünglichen Teil des Realms. Aus Performance-Gründen warten diese Transaktionen nämlich nicht auf das Ende der Realm-Erweiterung, sondern verfolgen die optimistische Strategie dass sie immer Platz finden.

Die Dauer der Online-Realm-Erweiterung hängt wesentlich von der Anzahl der Seiten ab, die im Zuge dieser Erweiterung formatiert werden müssen (siehe [Abschnitt „Datenbankseitenformatierung bei Realm-Erweiterung“](#)).

In der Regel erfolgt die logische Erweiterung des Realms um die Anzahl von Datenbankseiten, die bei der Aktivierung der Online-Erweiterbarkeit angegeben wurde, höchstens jedoch bis zur Anzahl maximal möglicher Datenbankseiten im Realm (siehe Tabelle 17, [„Erweiterbarkeit eines Realms einschätzen“](#)). Werden zur Verwaltung der neu hinzukommenden Seiten ein oder mehrere neue FPA-Extents benötigt, werden gegebenenfalls bis zu 64 PAM-Seiten mehr angefordert als durch das DAL-Kommando ACT INCR vorgegeben. Neue FPA-Extents liegen lückenlos am Beginn des Erweiterungsbereichs, damit die nachfolgenden neuen freien Seiten am Stück entstehen.

Konnte der Realm erweitert werden, wird folgende Meldung ausgegeben:

```
UDS0737 REALM WURDE ERWEITERT: (&00) DATENBANKSEITEN
```

6.2.2 Verhalten im Fehlerfall

Konnte der Realm auf Grund eines vom DVS gemeldeten Fehlers nicht erweitert werden, wird die folgende Meldung mit dem DVS-Fehler ausgegeben:

```
UDS0700 UDS BETRIEBSMITTEL-FEHLER (&00) (&01) (&02)
```

Wenn beim Erweitern des Realms ein DVS-Fehler auftritt oder wenn der Realm bereits vor dem Erweiterungsversuch seine Maximalgröße erreicht hat, wird vom DBH die Online-Erweiterung für den betroffenen Realm ausgesetzt. Dadurch wird verhindert, dass im aktuellen Sessionabschnitt immer wieder vergeblich versucht wird, diesen Realm zu erweitern. Die DML-Anweisung, die die fehl geschlagene Realm-Erweiterung ausgelöst hat, kann dann möglicherweise trotzdem erfolgreich beendet werden. Dies ist dann der Fall, wenn für die Bearbeitung der DML-Anweisung noch genügend Speicherplatz im Realm zur Verfügung steht. Die diesbezügliche Rückmeldung erfolgt wie üblich über den Statuscode.

Wird die Online-Erweiterung für einen Realm ausgesetzt (suspendiert), wird folgende Meldung ausgegeben:

```
UDS0740 ONLINE-ERWEITERBARKEIT DES REALMS (&00) WIRD AUSGESETZT
```

6.2.3 Reaktionsmöglichkeiten nach fehlgeschlagener Online-Realm- Erweiterung

Wie die Online-Erweiterbarkeit eines Realms reaktiviert werden kann, hängt von der Fehlerursache ab, die zur Aussetzung der Online-Erweiterbarkeit führte.

Muss der betroffene Realm zur Beseitigung der Fehlerursache nicht ausgehängt werden, können Sie für diesen Realm mit dem DAL-Kommando REACT INCR die Online-Erweiterbarkeit im aktuellen Session-Abschnitt wiederherstellen. Dies ist z. B. der Fall, wenn unter der Benutzerkennung zusätzlicher Plattenspeicherplatz zur Verfügung gestellt werden musste.

Muss der betroffene Realm zur Beseitigung der Fehlerursache ausgehängt werden, so ist nach Beseitigung der Fehlerursache die Online-Erweiterbarkeit allein durch Wiedereinschalten des Realms reaktiviert. Dies ist z. B. der Fall, wenn die Sekundärzuweisung der Realm-Datei von 0 auf einen Wert größer 0 gesetzt werden muss. Die implizite Reaktivierung der Online-Erweiterbarkeit eines Realms erfolgt auch, wenn zur Fehlerbeseitigung die Datenbank abgeschaltet und neu angeschlossen werden musste, oder wenn die Session beendet und neu gestartet werden musste.

Mit dem DAL Kommando DISPLAY INCR können Sie erfahren, ob für einen Realm die Online-Erweiterung ausgesetzt ist.

6.2.4 DAL-Kommandos und Dienstprogramme für die Online-Realm- Erweiterung

Im Folgenden sind den Aufgaben, die sich aus der Administration der Online-Realm-Erweiterung ergeben, den dafür zur Verfügung stehenden Dienstprogrammen und DAL-Kommandos zugeordnet.

Funktion	DAL/Dienstprogramm	Parameter /Anweisung
Online-Realm-Erweiterung aktivieren	DAL	ACT INCR
Online-Realm-Erweiterung einmalig bei der nächsten Freiplatzsuche im Realm durchführen	DAL	EXTEND REALM
Informationen zur Aktivierung der Online-Realm-Erweiterung des DBDIR bzw. eines Benutzer-Realms beschaffen	DAL	DISPLAY INCR
	BPSIA	DISPLAY SCHEMA
	BPRECORD	REALM NAME IS <i>realmname</i> DISPLAY PAGE ZERO
Online-Realm-Erweiterung deaktivieren	DAL	DEACT INCR
ermitteln, ob eine Online-Realm-Erweiterung ausgesetzt ist	DAL	DISPLAY INCR
Online-Realm-Erweiterung reaktivieren	DAL bzw. Realm abschalten, Fehlerursache beheben und Realm wieder zuschalten	REACT INCR
logische Größe eines Realms ermitteln	DAL	DISPLAY FPA
	BPRECORD	REALM NAME IS <i>realmname</i> DISPLAY PAGE ZERO
	BSTATUS	DISPLAY REALM <i>realmname</i>
ermitteln, ob bei einer Realm-Erweiterung formatiert wird	DAL	DISPLAY INCR
ermitteln, bis zu welcher Seite bei einer Realm-Erweiterung formatiert wird	BPRECORD	REALM NAME IS <i>realmname</i> DISPLAY PAGE ZERO
Lage und Anzahl der FPA-Extents des DBDIR, des DBCOM, bzw. eines Benutzer-Realms ermitteln	BPRECORD	REALM NAME IS <i>realmname</i> DISPLAY PAGE ZERO
FPA-Basis und FPA-Extents zusammen fassen	BPGSIZE	
Nummer eines Realms ermitteln (erforderlich für ACT INCR und DEACT INCR)	DISPLAY INCR	
	BPRECORD	

Tabelle 18: DAL-Kommandos und Dienstprogramme für die Administration der Online-Realm-Erweiterung

6.2.5 Aktivierung der Online-Erweiterbarkeit beim Datenbankaufbau

Beim Neuaufbau von Datenbanken wird von den Dienstprogrammen BCREATE und BFORMAT die Online-Erweiterbarkeit automatisch für alle Realms aktiviert, für die bei der Speicherplatzzuweisung eine Sekundärzuweisung > 0 festgelegt wurde.

Die Aktivierung der Online-Erweiterbarkeit gilt somit für alle neu aufgebauten Realms mit einer Sekundärzuweisung > 0 von Anfang an und muss nicht mit den entsprechenden DAL-Kommandos eingestellt werden. Sie können jedoch die Aktivierung der Online-Erweiterbarkeit unterbinden, indem Sie für den Realm eine Sekundärzuweisung = 0 festlegen.

Eine Änderung der Sekundärzuweisung, nachdem die Dienstprogramme BCREATE und BFORMAT gelaufen sind, hat keine automatische Anpassung dieser Aktivierung zur Folge.

Wird die Online-Realm-Erweiterung auf diese Weise aktiviert, so werden für *nr-pages* und *min-pages* die Standardwerte *nr-pages*=64 und *min-pages*=0 eingetragen (siehe „Online-DBTT-Erweiterbarkeit für eine Satzart aktivieren (ACT DBTT-INCR)“ im [Abschnitt „Online-Erweiterbarkeit aktivieren \(ACT\)“](#)).

Für alle DBTTs, die in Realms liegen, für die auf diese Weise die Online-Erweiterbarkeit aktiviert wurde, wird ebenfalls bereits beim Datenbankaufbau die Online-DBTT-Erweiterung aktiviert. Dabei wird für den Parameter SCAN der Standardwert SCAN=YES verwendet (siehe „Online-DBTT-Erweiterbarkeit für eine Satzart aktivieren (ACT DBTT-INCR)“ im [Abschnitt „Online-Erweiterbarkeit aktivieren \(ACT\)“](#)).

Eine spätere Änderung der Einstellungen (insbesondere eine Deaktivierung) sowohl der Online-Realm-Erweiterung als auch der Online-DBTT-Erweiterung mit den DAL-Kommandos ist möglich.

Unabhängig von der Online-Erweiterbarkeit eines Realms können Sie mit dem DAL-Kommando EXTEND REALM eine einmalige Realmerweiterung vormerken lassen, die bei der nächsten Freiplatzsuche im Realm durchgeführt wird.

6.3 Online-DBTT-Erweiterung

In diesem Abschnitt erfahren Sie, wie Sie die Online-DBTT-Erweiterung aktivieren, wie diese abläuft, was im Fehlerfall passiert und welche DAL-Kommandos und Dienstprogramme für die Online-DBTT-Erweiterung zur Verfügung stehen.

6.3.1 Online-DBTT-Erweiterung aktivieren

Voraussetzung für die Online-DBTT-Erweiterung ist, dass die Online-Realm-Erweiterung des Realms, in dem die DBTT liegt, aktiviert ist.

Ist die Online-Realm-Erweiterung aktiviert und sind die Systemvoraussetzungen für die Realm-Erweiterung gegeben, können Sie für eine DBTT die Online-Erweiterbarkeit aktivieren. Verwenden Sie dafür das DAL-Kommando ACT DBTT-INCR (siehe Abschnitt „Online-DBTT-Erweiterbarkeit für eine Satzart aktivieren (ACT DBTT-INCR)“ im [Abschnitt „Online-Erweiterbarkeit aktivieren \(ACT\)“](#)).

Der Auftrag wird erst ausgeführt, nachdem das Kommando PERFORM abgesetzt wurde.

Die Online-Erweiterbarkeit der DBTT bleibt so lange aktiv, bis das Kommando DEACT DBTT für diese DBTT abgesetzt wurde (siehe [Abschnitt „Online-Erweiterbarkeit deaktivieren \(DEACT\)“](#)).

Verwenden Sie das DAL-Kommando DISPLAY DBTT oder das Dienstprogramm BSTATUS, um Informationen über die Online-Erweiterbarkeit einer DBTT zu erhalten.

Beim Neuaufbau von Datenbanken wird von den Dienstprogrammen BCREATE und BFORMAT die Online-DBTT-Erweiterung automatisch für alle DBTTs aktiviert, die in Realms liegen, für die die Online-Realm-Erweiterung automatisch aktiviert wurde. Siehe hierzu [Abschnitt „Aktivierung der Online-Erweiterbarkeit beim Datenbankaufbau“](#).

6.3.2 Ablauf der automatischen Online-DBTT-Erweiterung

Die eigentliche Erweiterung einer DBTT wird vom DBH automatisch angestoßen, wenn bei der Suche nach einem freien Eintrag festgestellt wird, dass der letzte Eintrag belegt ist (SCAN=NO) oder in der gesamten DBTT kein freier Eintrag mehr zur Verfügung steht (SCAN=YES). Die DBTT-Erweiterung wird innerhalb der User-Transaktion dieser DML durchgeführt. Die Antwortzeit der DML wird dadurch zwar vergrößert, insgesamt ist die Behinderung des Gesamtbetriebs durch diese Vorgehensweise aber gering.

Die Dauer der Online-DBTT-Erweiterung hängt wesentlich von der Dauer der damit verbundenen Online-Realm-Erweiterung ab.

Bei der Online-DBTT-Erweiterung wird jedesmal der Realm, in dem die DBTT liegt, erweitert. Durch diese obligatorische Realm-Erweiterung wird der laufende Transaktionsbetrieb am wenigsten belastet.

Der Realm wird nur um das notwendige Maß erweitert, nicht gemäß den bei der Aktivierung der Online-Realm-Erweiterung angegebenen Werten. Somit entstehen bei der Online-Realm-Erweiterung für eine DBTT keine überflüssigen leeren Seiten.

Konnte die DBTT erweitert werden, wird folgende Meldung ausgegeben:

```
UDS0741 DBTT WURDE ERWEITERT: (&00) DBTT EINTRAEGE
```

Im Insert (&00) wird der Recordname mit der Anzahl der DBTT-Einträge ausgegeben, um die die DBTT erweitert wurde.

Falls mit dem EXT-Parameter eine Anzahl der Einträge angegeben wurde, wird die DBTT um eine aufgerundete Anzahl von DBTT-Einträgen erweitert. Diese Anzahl hängt von der Seitengröße der Datenbank und der DBTT-Zeilenlänge der jeweiligen Satzart ab.

6.3.3 Explizite Online-DBTT-Erweiterung veranlassen

Mit dem DAL-Kommando EXTEND DBTT (siehe "[Online-DBTT-Erweiterung ausführen \(EXTEND DBTT\)](#)") können Sie unabhängig davon, ob die Online-DBTT-Erweiterung aktiviert ist, beim nächsten Einspeichern eines Satzes in der entsprechenden Satzart eine Online-DBTT-Erweiterung veranlassen. Auch in diesem Fall müssen die Voraussetzungen einer Online-Realm-Erweiterbarkeit erfüllt sein.

6.3.4 Verhalten im Fehlerfall

Konnte die DBTT nicht erweitert werden, wird eine Meldung ausgegeben, die die Ursache angibt.

Wurde das Scheitern der DBTT-Erweiterung durch das Scheitern der notwendigen Realm-Erweiterung verursacht, unterbleiben weitere Erweiterungsversuche für diese DBTT, bis die Suspendierung der Realm-Erweiterung aufgehoben ist.

Besteht wegen gescheiterter Online-Realm-Erweiterung eine Suspendierung, die aufgrund eines großen Erweiterungsgranulats entstanden ist, können Sie evtl. mit den DAL-Kommandos REACT INCR und EXTEND DBTT dennoch eine erfolgreiche Online-DBTT-Erweiterung durchführen.

6.3.5 DAL-Kommandos und Dienstprogramme für die Online-DBTT- Erweiterung

Im Folgenden sind den Aufgaben, die sich aus der Administration der Online-DBTT-Erweiterung ergeben, die dafür zur Verfügung stehenden Dienstprogrammen und DAL-Kommandos zugeordnet.

Funktion	DAL /Dienstprogramm	Parameter /Anweisung
Online-DBTT-Erweiterung aktivieren	DAL	ACT DBTT-INCR
Online-DBTT-Erweiterung einmalig beim nächsten Einspeichern in der Satzart durchführen	DAL	EXTEND DBTT
Informationen zur Aktivierung der Online-DBTT-Erweiterung beschaffen	DAL	DISPLAY DBTT-INCR
	BSTATUS	DISPLAY RECORD
Online-DBTT-Erweiterung deaktivieren	DAL	DEACT DBTT-INCR
Ermitteln, ob eine Online-DBTT-Erweiterung für eine Satzart ausgesetzt ist	DAL	DISPLAY DBTT-INCR
DBTT-Basis und DBTT-Extents zusammenfassen	BPGSIZE	
Referenz einer Satzart ermitteln (erforderlich für ACT, EXTEND und DEACT DBTT)	DAL	DISPLAY DBTT
	BPSIA	

Tabelle 19: DAL-Kommandos und Dienstprogramme für die Administration der Online-DBTT-Erweiterung

6.4 Online-Reorganisation mit der UDS-Online-Utility

Die UDS-Online-Utility bietet die Möglichkeit, einige Funktionen, die sonst teilweise auf ähnliche Weise durch Dienstprogramme realisiert sind, online (d.h. während des DBH-Betriebs) durchzuführen.

Offline können Sie die Verdichtung von Datenbeständen folgendermaßen erreichen:

- mit BPGSIZE (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“) oder
- durch Entladen und Neuladen der Daten mit BOUTLOAD/BINILOAD (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“) oder
- durch Neuaufbau von Tabellen mit BREORG (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“)

Mit der UDS-Online-Utility können Sie die Verlagerung und Verdichtung von Datensätzen und kleinen Tabellen gezielt in einzelnen Realms im laufenden Betrieb durchführen.

Für verteilbare Listen können Sie eine Verlagerung von Stufe-0-Seiten in einen anderen beteiligten Realm im laufenden Betrieb durchführen. Sie können im laufenden Betrieb den Realm ändern, in dem freier Platz für zusätzlich benötigte Stufe-0-Seiten gesucht wird (Preferred-Realm).

Außerdem können Sie mit der UDS-Online-Utility die vergleichbaren Funktionen SET und RESET des Dienstprogrammes BMODTT (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“) auch im laufenden Betrieb aufrufen und wirksam machen.

Mit der UDS-Online-Utility können sie auch PPPs während des laufenden Betriebs mit den häufigsten Werten aktualisieren.

Eine ausführliche Beschreibung der UDS-Online-Utility finden Sie im Handbuch „[Sichern, Informieren und Reorganisieren](#)“.

6.5 Automatische Realm-Erweiterung durch Dienstprogramme

Dienstprogramme mit Linked-in DBH

Die UDS/SQL-Dienstprogramme DDL-Compiler, SSL-Compiler, BPRIVACY, BGSIA und BGSSIA bearbeiten DBDIR und DBCOM (und ggf. COSSD), nicht jedoch die Benutzerrealms der betr. Datenbank. Beim Ablauf dieser Dienstprogramme mit Linked-in DBH (= Utility-Sessions) werden sowohl DBDIR als auch DBCOM bei Bedarf dynamisch erweitert. Diese Erweiterungen erfolgen unabhängig von der Aktivierung der Online-Erweiterbarkeit für diese Realms. Voraussetzung hierfür ist jedoch, dass bei der Speicherplatzzuweisung für DBDIR und/oder DBCOM eine Sekundärzuweisung > 0 festgelegt ist. Somit können Sie, falls gewünscht, die automatische Erweiterung von DBDIR oder DBCOM durch die Festlegung einer Sekundärzuweisung = 0 unterbinden.

Die via DAL-Kommando mögliche Aktivierung der Online-Erweiterbarkeit des DBDIR wirkt nur für User-Sessions. Diese Aktivierung können Sie auch weiterhin wie bisher mit den DAL-Kommandos ACT INCR bzw. DEACT INCR des Independent DBH ändern. Die Online-Erweiterbarkeit des DBCOM können Sie nur über die Sekundärzuweisung beeinflussen.

In Utility-Sessions mit Linked-in DBH werden automatische Erweiterungen aus folgenden Gründen mit dem Standardwert 64 für ADDITIONAL_NR_PAGES und dem Wert 0 für MIN_NR_FREE_PAGES durchgeführt:

- Der DBH fordert bei der Freiplatzsuche immer nur einzelne freie Seiten an. Deshalb sind keine größeren Erweiterungsschritte erforderlich und weder DBDIR noch DBCOM werden allzu groß.
- Bei Dienstprogrammläufen ist keine Parallelität gegeben, daher muss kein Vorhalt in Form von MIN_NR_FREE_PAGES > 0 getroffen werden.

Die DBTTs der Satzarten in DBDIR und DBCOM (mit Ausnahme der DBTT des SSIA_RECORDs) werden ebenfalls unabhängig von der nur für User-Sessions erforderlichen Aktivierung der Online-Erweiterbarkeit im Bedarfsfall automatisch erweitert, wenn die Voraussetzung fuer die entsprechende Realm-Erweiterung (Sekundärzuweisung > 0) erfüllt ist.

Die beim Datenbankaufbau zum Tragen kommende Dimensionierung der Satzarten in DBDIR und DBCOM wird beibehalten. Die DBTTs dieser Satzarten sind damit im Allgemeinen ausreichend groß angelegt. Dadurch soll erreicht werden, dass Erweiterungen der DBTTs dieser Satzarten auf Ausnahmefälle beschränkt bleiben. Andernfalls würden DBDIR und DBCOM durch häufige DBTT-Erweiterungen unverhältnismäßig stark anwachsen.

Dienstprogramme ohne DBH

Die UDS/SQL-Dienstprogramme BCREATE, BCHANGE, BALTER, BRENAME, BREORG, BFORMAT und BINILOAD erweitern bei Bedarf automatisch die Realms der bearbeiteten Datenbank.

Unabhängig davon, ob für diese Realms die Online-Erweiterbarkeit aktiviert ist, veranlassen die genannten Dienstprogramme eine Erweiterung des Realms immer dann, wenn der verfügbare Freiplatz für den Ablauf des Dienstprogramms nicht mehr ausreicht.

Voraussetzung hierfür ist jedoch, dass bei der Speicherplatzzuweisung für den betreffenden Realm eine Sekundärzuweisung > 0 festgelegt ist. Somit können Sie, falls gewünscht, die automatische Realm-Erweiterung durch die Festlegung einer Sekundärzuweisung = 0 unterbinden.

Außerdem gelten die im [Abschnitt „DVS-Aspekte der Online-Realm-Erweiterung“](#) beschriebenen Randbedingungen.

Der betreffende Realm wird jeweils um so viele Datenbankseiten erweitert, wie in die aktuell beim DVS eingestellte Sekundärzuweisung passen, mindestens jedoch um den Standardwert 64 Seiten (dabei können noch 64 Seiten hinzukommen, wenn ein neuer FPA-Extent benötigt wird).

Falls im Ablauf des Dienstprogramms die Anforderung eines größeren zusammenhängenden Bereichs an Freiplatz im vorhandenen Realm nicht erfüllt werden kann, wird die Größe dieses Bereichs als Maß für die Erweiterung genommen (wobei auch hier die Anzahl Seiten, die in die Sekundärzuweisung passen und der Standardwert 64 nicht unterschritten werden).

Die Durchführung einer automatischen Realm-Erweiterung, sowie deren Umfang und die neue Größe des Realms werden mit der folgenden Meldung angezeigt:

```
0074 REALM <realmname> HAS BEEN EXTENDED BY nnn DATABASE-PAGES  
      NEW NR OF PAGES : mmm
```

Das Scheitern einer automatischen Realm-Erweiterung führt zur Ausgabe der Meldung

```
0073 DYNAMIC EXTENSION BY nnn DATABASE-PAGES NOT POSSIBLE FOR REALM  
<realmname>
```

- i** Online-Kopien, die erstellt werden, während ein Realm von einem Dienstprogramm automatisch erweitert wird, sind unbrauchbar, weil BMEND beim Nachfahren das Erreichen eines konsistenten DB-Zustands nicht korrekt ermitteln kann. Die Online-Sicherungsfähigkeit der Datenbank wird deshalb von UDS während eines solchen Dienstprogrammlaufs vorübergehend zurückgenommen. Dadurch werden Online-Sicherungen mit HSMS/ARCHIVE in diesem Zeitraum verhindert. Bei Verwendung des COPY-FILE ist es in der Verantwortung des Benutzers, Online-Sicherungen während des Ablaufs dieser Dienstprogramme zu unterlassen.

7 Datenbank sichern und wiederherstellen im Fehlerfall

Dieses Kapitel erläutert, was Sie tun können, um sich vor Datenverlust zu schützen, und was Sie tun müssen, wenn ein Fehler aufgetreten ist.

7.1 Das Sicherungskonzept von UDS/SQL

Eine der wichtigsten Aufgaben eines Datenbanksystems besteht darin, jederzeit die Konsistenz der Daten, d.h. sowohl die physische Speicherkonsistenz als auch die logische Richtigkeit des Datenbestandes zu gewährleisten. Insbesondere muss das System durch entsprechende Maßnahmen in der Lage sein, den Verlust und die Verfälschung von Daten zu verhindern. Nach eventuell auftretenden Fehlern beim Datenbankbetrieb muss das Wiederherstellen eines konsistenten Zustandes schnell möglich sein.

Alle Maßnahmen, die auf die Erhaltung und Wiederherstellung der Konsistenz der Daten ausgerichtet sind, sind Teile eines integrierten Sicherungskonzepts. Die Maßnahmen lassen sich aufteilen in die *Transaktionssicherung* und die *Langzeitsicherung*.

Während die Transaktionssicherung die Konsistenz der Daten im laufenden Betrieb erhalten soll, ist es Aufgabe der Langzeitsicherung, die Konsistenz der Daten über die gesamte Lebensdauer der Datenbank hinweg zu erhalten und evtl. wiederherzustellen.

Welche Sicherung Sie bei welcher Fehlermöglichkeit nutzen, wird im Folgenden kurz dargestellt.

Fehler auf externen Speichern

Ein externes Speichergerät (Platte) lässt sich nicht mehr bearbeiten.

- Bei Schreibfehler und Lesefehler werden offene Transaktionen mit der Transaktionssicherung zurückgesetzt und der Datenbankbetrieb eingeschränkt fortgesetzt bis zum Austausch des defekten Gerätes mit seinen Daten.
- Bei Hardwarefehler gehen Sie auf einen Sicherungsstand Ihrer Langzeitsicherung zurück (und rekonstruieren evtl. einen jüngeren Stand), ehe Sie den Datenbankbetrieb auf diesem Datenbestand wieder aufnehmen.

Systemfehler

Durch Ausfall des Systems sind Hauptspeicherinhalte verloren gegangen, externe Speicher sind jedoch weiterhin intakt.

Mit der Transaktionssicherung werden offene Transaktionen zurückgesetzt und danach der Datenbankbetrieb normal fortgesetzt.

Fehler im Anwenderprogramm

Abhängig von dem Fehler auf dem Datenbestand gehen Sie auf einen Sicherungsstand Ihrer Langzeitsicherung zurück (und rekonstruieren evtl. einen jüngeren Stand), oder es wird die betroffene offene Transaktion durch die Transaktionssicherung zurückgesetzt.

Wesentliche Merkmale des UDS/SQL-Sicherungskonzeptes sind:

- Fehler rechtzeitig zu erkennen
- Fehler auf möglichst geringe Teileinheiten zu begrenzen
- Fehler schnell und flexibel zu behandeln und die Datenbank auch im laufenden Betrieb zu reparieren

7.2 Transaktionssicherung

Die Transaktionssicherung behandelt folgende Fehlerarten:

- Fehler im Anwenderprogramm

Dazu gehören alle Situationen, bei denen die aktuelle Transaktion des Anwenderprogramms zurückgesetzt werden muss, z.B. Absturz des Anwenderprogramms, Deadlock, Rücksetzen der Transaktion durch die DML-Anweisung FINISH WITH CANCEL.

- Fehler im System

Hier sind alle Fehler gemeint, die im DBH auftreten, z.B. UDS/SQL-Systemabsturz, BS2000-Systemabsturz, Ressourcenmangel. Davon ausgenommen sind Fehler auf Dateien. Diese Fehler werden durch die Langzeitsicherung behandelt.

Es wird im Weiteren vorausgesetzt, dass Sie Ihre Datenbankänderungen über die ALOG- als auch die RLOG-Datei gesichert haben.

7.2.1 Fehlerarten und ihre Behandlung

- Fehler im Anwenderprogramm
- Fehler im System

7.2.1.1 Fehler im Anwenderprogramm

Diese Fehler werden dadurch behandelt, dass die davon betroffene offene Transaktion zurückgesetzt wird. Das Zurücksetzen der Transaktion geschieht mit Hilfe von Before-Images (Before-Images beschreiben den Zustand von Daten vor ihrer Änderung). Die Before-Images werden zunächst im Hauptspeicher gehalten und bei Bedarf auf die RLOG-Datei geschrieben (siehe [Abschnitt „RLOG-Datei“](#)).

7.2.1.2 Fehler im System

Bei einem Fehler im System muss die Fehlerbehandlung Folgendes leisten:

- Sie muss alle offenen Transaktionen zurücksetzen können. Dies geschieht wie bei einem Fehler im Anwenderprogramm mit Hilfe der Before-Images.
- Sie muss die Ergebnisse aller abgeschlossenen Transaktionen, die sich noch nicht auf der Datenbank befinden, in die Datenbank einbringen. Dies geschieht mit Hilfe von After-Images (After-Images beschreiben den Zustand von Daten nach ihrer Änderung). Dazu werden die After-Images jeder Transaktion auf die RLOG-Datei geschrieben. Das Schreiben der After-Images erfolgt spätestens beim Transaktionsende.

Warmstart

Nach einem Absturz des DBH sind die betroffenen Datenbanken zunächst inkonsistent, sofern sie während der Session verändert wurden. Werden diese Datenbanken einer Konfiguration zugeschaltet, so führt UDS/SQL als Erstes für diese Datenbanken einen Warmstart durch. Dabei werden abgeschlossene Transaktionen nachgefahren, und offene Transaktionen zurückgesetzt. Anschließend werden die Datenbanken als konsistent markiert.

Wenn es sich um eine openUTM-Anwendung handelt, so findet eine Synchronisation mit openUTM bezüglich der Zustände der einzelnen Transaktionen statt. Für diese Synchronisation wird die Status-Datei benötigt (siehe [Abschnitt „Status-Datei“](#)).

Wenn es sich um eine UDS-D-Anwendung handelt, so ist zusätzlich eine Synchronisation globaler Transaktionen erforderlich. Globale Transaktionen sind Transaktionen, für die in mehr als einem DBH DML-Anweisungen verarbeitet wurden. Auch für diese Synchronisation wird die Statusdatei benötigt.

Nach einem DBH-Absturz dürfen Sie die betreffenden Datenbanken bis zum Warmstart nicht verändern. Von dieser Regel gibt es folgende Ausnahmen:

- Beim oder nach dem Absturz wurde die Datenbank zerstört. Dann müssen Sie eine Sicherung einspielen, und mit dem Dienstprogramm BMEND diese Sicherung bis zum Absturzzeitpunkt nachfahren (Anweisung UPDATE-DATABASE, DEADLINE = BREAK-POINT).
- Nach dem Absturz wurden nur DMS-Katalogeinträge der betroffenen Dateien, aber nicht deren Inhalt geändert (z. B. nach Dateireorganisation durch das Rechenzentrum, Auslagerung und Wiedereinlagerung durch HSMS). In diesem Fall kann es sein, dass der DBH den Warmstart verweigert. Dann müssen Sie die abgestürzte Datenbank mit dem Dienstprogramm BMEND bis zum Absturzzeitpunkt nachfahren (Anweisung UPDATE-DATABASE, DEADLINE = BREAK-POINT).

Session-Wiederanlauf

UDS/SQL vermerkt sich die aktuelle Konfiguration in der SLF (siehe [Abschnitt „Session-Log-File \(SLF\)“](#)). Wenn der DBH nach einem Absturz mit dem gleichen Konfigurationsnamen gestartet wird, nimmt der DBH die DBH-Ladeparameter aus der SLF. Die DBH-Ladeparameter, die Sie beim Starten des DBH angegeben haben, werden dann nicht berücksichtigt.

Dieser Vorgang heißt Session-Wiederanlauf.

Wenn Sie keinen Session-Wiederanlauf wollen, müssen Sie vor dem Start des DBH die SLF löschen.

7.2.2 Handhabung der benötigten Dateien

- RLOG-Datei
- Session-Log-File (SLF)
- Status-Datei

7.2.2.1 RLOG-Datei

Die RLOG-Datei wird für folgende Fälle benötigt:

- Zurücksetzen von offenen Transaktionen im laufenden Betrieb oder beim Warmstart
- Nachfahren von abgeschlossenen Transaktionen beim Warmstart

Die RLOG-Datei enthält deshalb die Before-Images und After-Images jeder Änderung. Die RLOG-Datei wird vom DBH bei Beginn einer Session angelegt. Sie besteht aus zwei physischen Dateien mit den Namen

konfname.RLOG.zeitstempel.1

konfname.RLOG.zeitstempel.2

Sie können über DBH-Ladeparameter oder DAL-Kommando bestimmen, auf welchen Platten die RLOG-Datei angelegt werden soll. Aus Sicherheitsgründen empfiehlt es sich, die RLOG-Dateien doppelt zu führen. Die Doppelführung kann entweder mit DRV erfolgen, oder Sie können die RLOG-Datei durch UDS/SQL doppelt führen lassen. In diesem Fall legt der DBH noch zwei weitere Dateien mit folgenden Namen an:

konfname.RLOG.zeitstempel.1.SAVE

konfname.RLOG.zeitstempel.2.SAVE

Die RLOG-Datei wird vom DBH gelöscht, wenn die zugehörige Session fehlerfrei beendet wurde. Nach einem Absturz wird die zugehörige RLOG-Datei erst dann gelöscht, wenn für alle Datenbanken, die vom Absturz betroffen waren, ein Warmstart durchgeführt wurde.

Die RLOG-Datei wird vom DBH mit einem internen Kennwort geschützt.

Datenträger für die RLOG-Datei auswählen

Bei der Auswahl der Datenträger für die RLOG-Datei sind die Aspekte Sicherheit und Performance besonders wichtig.

Sicherheit

Aus Sicherheitsgründen sollte die RLOG-Datei nicht auf Platten gelegt werden, auf denen sich auch Datenbankdateien befinden.

Beim Doppelführen der RLOG-Datei durch UDS/SQL wird kontrolliert, ob Sie zwei unabhängige Datenträger als Ablageort angegeben haben. Falls nicht, wird das Einrichten der RLOG-Dateien verweigert.

Bei doppelt angelegten RLOG-Dateien auf SF-Pubsets, die im Globalspeicher gepuffert sind, wird außerdem geprüft, ob sie in unterschiedlichen Units gepuffert sind. Falls nicht, wird das Einrichten der RLOG-Dateien verweigert.

Für RLOG-Dateien auf SM-Pubsets kann diese Prüfung von UDS/SQL nicht durchgeführt werden, weil die entsprechende Information für SM-Pubsets nicht ermittelbar ist. UDS/SQL kann somit nicht mehr vollständig das Vorhandensein von "single points of failure" verhindern.

Da die RLOG-Dateien für die Konsistenz der Daten im laufenden Betrieb eine sehr große Bedeutung haben, wird jede RLOG-Datei mit einem eigenen Kennwort geschützt. Im Allgemeinen löscht der DBH RLOG-Dateien, wenn sie nicht mehr benötigt werden. Sofern in Ausnahmefällen ein Löschen einer RLOG-Datei aus anderen Gründen nötig wird, können Sie dazu das Kennwort mit dem Dienstprogramm RLOGPASS der Liefereinheit UDS-SQL-T absetzen. Mit diesem Kennwort soll nicht nur der Inhalt der Dateien vor unberechtigtem Zugriff geschützt werden, sondern es soll auch ein versehentliches Löschen verhindert werden.

Performance

Wenn bei einer Anwendung die RLOG-Datei zu einem Engpass wird, können Sie folgende Maßnahmen ergreifen:

- RLOG-Datei auf einer Platte anlegen, auf die sonst keine Zugriffe erfolgen. Die RLOG-Datei sollten Sie so anlegen, daß das Schreiben möglichst schnell und ohne Beeinflussung durch andere Aktivitäten auf dem System erfolgen kann.
- Bei noch höherer Durchsatzanforderung kann man die RLOG-Datei auf ein schnelleres Gerät legen (z.B SSD, siehe Handbuch „[Einführung in die Systembetreuung](#)“). Eine weitere Möglichkeit ist, die RLOG-Datei auf gemeinschaftliche Platte zu legen, die über den Globalspeicher (GS) gepuffert wird.

**VORSICHT!**

Die RLOG-Dateien dürfen nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

7.2.2.2 Session-Log-File (SLF)

In der SLF legt der DBH die DBH-Ladeparameter der aktuellen Konfiguration ab. Die SLF wird vom DBH bei Session-Beginn angelegt. Bei einem Session-Wiederanlauf entnimmt der DBH aus der SLF die Ladeparameter der abgestürzten Session. Die DBH-Ladeparameter, die Sie beim DBH-Start angegeben haben, werden dabei ignoriert. Wenn Sie keinen Session-Wiederanlauf wollen, müssen Sie vorher die SLF löschen. Diese Datei ist mit dem Standard-Kennwort C'UDS'BLANK' geschützt. Mit diesem Kennwort soll nicht der Inhalt der session-übergreifend genutzten Datei vor unberechtigtem Zugriff geschützt werden, sondern es soll ein versehentliches Löschen verhindert werden.

**VORSICHT!**

Die SLF darf nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

7.2.2.3 Status-Datei

Die Status-Datei enthält Informationen, die u.a. für die Synchronisation mit openUTM und für die Synchronisation globaler Transaktionen von UDS-D benötigt werden. Die Status-Datei wird in aufeinander folgenden Sessions fortgeschrieben. Sie kann gelöscht werden, wenn alle in einer Konfiguration aktiven openUTM-Anwendungen bzw. über UDS-D angeschlossenen Anwendungen ordnungsgemäß beendet wurden und für diese Anwendungen kein Warmstart mehr durchgeführt werden muss.

Die Status-Datei wird immer doppelt geführt. Die entsprechenden Dateinamen sind

konfname.DBSTAT

konfname.DBSTAT.SAVE

Die Status-Dateien müssen Sie anlegen. Diese Dateien werden vom DBH mit dem Standard-Kennwort C' UDS'BLANK" geschützt.

Wenn Sie die Status-Dateien löschen wollen, müssen Sie in dem DELETE-FILE-Kommando den Operanden OPTION=*DESTROY-ALL angeben.

! **VORSICHT!**

Die Status-Dateien dürfen nicht in flüchtigen Medien schreib- oder schreiblesegepuffert genutzt werden (siehe [Abschnitt „DAB-Caching für UDS/SQL nutzen“](#)).

7.3 Langzeitsicherung

Die Langzeitsicherung umfasst alle Maßnahmen, die dazu dienen, den Bestand der Daten über die gesamte Lebensdauer der Datenbank zu erhalten. Dazu gehören die vorbeugende Sicherung einer Datenbank, das Führen von ALOG-Dateien und die Rekonstruktion einer Datenbank.

Bei der vorbeugenden Sicherung wird in regelmäßigen Abständen der gesamte Datenbestand einer Datenbank gesichert, sodass Sie z.B. bei Ausfall einer DB-Platte auf diesen älteren Sicherungsstand zurückgehen können und nicht sämtliche Daten verloren haben.

Um die Änderungen der Daten zwischen zwei derartigen Sicherungszeitpunkten nicht zu verlieren, führen Sie im laufenden Betrieb ALOG-Dateien (Archiv-Log-Dateien). In ihnen werden die geänderten Daten nach ihrer Änderung (After-Images) gesammelt. Wenn bei Änderungen auf der Datenbank keine ALOG-Datei geführt wird, so besteht für diesen Zeitraum eine „Logging-Lücke“.

Mit Hilfe der ALOG-Datei(en) können Sie konsistente Stände einer Datenbank rekonstruieren, die z.B. zwischen zwei Sicherungszeitpunkten der gesamten Datenbank liegen. Dazu werden mit dem UDS/SQL-Dienstprogramm BMEND (Anweisung UPDATE-DATABASE) auf einem Sicherungsstand der Datenbank die ALOG-Datei bzw. ALOG-Dateien bis zu dem gewünschten konsistenten Stand nachgefahren.

7.3.1 Datenbanksicherung erstellen

Für die Sicherung einer Datenbank, d.h. aller UDS/SQL-Datenbankdateien, benötigen Sie kein datenbankspezifisches Sicherungsprogramm. Sie erstellen die Datenbanksicherung mit den im BS2000 zur Verfügung stehenden Möglichkeiten, üblicherweise mit den BS2000-Dienstprogrammen HSMS bzw. ARCHIVE (siehe dazu das Handbuch „[HSMS / HSMS-SV \(BS2000\) Band 1: Funktionen, Verwaltung und Installation](#)“ bzw. „[ARCHIVE \(BS2000\)](#)“).

Die Zeitpunkte sowie die Archivierung einer solchen Datenbank-“Vollsicherung“ bestimmen Sie auf Grund Ihrer organisatorischen Bedingungen und Gegebenheiten selbst. Außerhalb der periodischen Sicherung empfiehlt es sich, eine Sicherung zu erstellen, nachdem z.B. eine Logging-Lücke aufgetreten ist oder die Datenbank umstrukturiert wurde.

7.3.1.1 Konsistenzpunkt

Der Datenbestand einer Datenbank ist zu bestimmten Zeitpunkten (genannt *Konsistenzpunkt*) konsistent, d.h. alle ändernden Transaktionen sind beendet und ihre Änderungen befinden sich im Datenbestand.

Ein Konsistenzpunkt wird in folgenden Fällen erreicht:

- beim normalen Ende einer DBH-Session oder eines ändernden Dienstprogramms
- nach erfolgreichem Session-Wiederanlauf
- beim Wechsel der ALOG-Datei (z.B. durch Überlauf)
- mit dem DAL-Kommando CHECKPOINT (nur bei independent DBH möglich)
- beim Behandeln bestimmter Ereignisse durch den DBH (z.B. Realm abschalten)

Mit dem Erreichen des Konsistenzpunktes kann gleichzeitig ein Wechsel der ALOG-Datei stattfinden. Der Konsistenzpunkt wird dann festgeschrieben. Die Sicherung einer Datenbank können Sie mit dem Dienstprogramm BMEND und der abgeschlossenen ALOG-Datei jeweils bis zu diesem Konsistenzpunkt aktualisieren.

Das Verlassen des Konsistenzpunktes beim Verändern einer Datenbank bezeichnet zugleich den Log-Intervall-Beginn auf einer leeren ALOG-Datei. Das Log-Intervall-Ende wird mit dem Erreichen des Konsistenzpunktes gesetzt. Wird die ALOG-Datei nun nicht gewechselt und mit den Änderungen bis zum nächsten Konsistenzpunkt fortgeschrieben, so vergrößert sich das Log-Intervall der ALOG-Datei um diese Zeitspanne. Das Log-Intervall-Ende wird entsprechend aktualisiert.

7.3.1.2 Sicherung einer konsistenten Datenbank

Die Sicherung der gesamten Datenbank (ohne ALOG-Dateien) nehmen Sie vor, wenn die Datenbank in einem konsistenten Zustand ist („Konsistenzpunkt“ erreicht hat) und auf ihr kein Datenbankbetrieb stattfindet, sie also nicht prozessiert wird. Falls Sie im Fehlerfall auf diese Sicherung zurückgreifen müssen, so ist zuerst diese Sicherung z. B wieder mit HSMS bzw. ARCHIVE einzuspielen, ehe Sie den Datenbankbetrieb auf diesem Sicherungsstand wieder aufnehmen können.

Für die Sicherung der Datenbank stehen Ihnen alle Möglichkeiten zur Verfügung, die z.B. HSMS bzw. ARCHIVE bieten (Sicherung auf Platte, Band oder Magnetbandkassette).

7.3.1.3 Online-Sicherung einer Datenbank

Mit HSMS (Operanden `SAVE-OPTIONS=*PARAMETERS(SAVE-ONLINE-FILES=YES)`) bzw. ARCHIVE (Operanden `PARAM OLS=YES`) ist es auch möglich, die UDS/SQL-Datenbankdateien zu sichern, während die Datenbank gleichzeitig vom DBH prozessiert und verändert wird.

Voraussetzungen für die Online-Sicherung einer Datenbank

- Sie müssen die „Online-Sicherungsfähigkeit“ der Datenbank vor dem Start des Datenbankbetriebs mit dem Dienstprogramm BMEND (Anweisung `ENABLE-ONLINE-COPY`) für die Datenbank vereinbart haben.
- Das AFIM-Logging (After-Image-Logging) der Datenbank muss eingeschaltet sein, denn nur die Online-Sicherung zusammen mit der(den) während des Sicherns erzeugten ALOG-Datei(en) bilden einen konsistenten Stand der Datenbank. Die Online-Sicherung selbst ist in der Regel inkonsistent, außer es wird durch Administrationsmaßnahmen dafür gesorgt, dass die Datenbank zu Beginn des Sicherungsvorgangs konsistent ist und während der gesamten Dauer der Sicherung keine Änderungen stattfinden. Um den konsistenten Stand einer Online-Sicherung herzustellen, spielen Sie zuerst die Sicherung auf Platte wieder ein, ehe Sie mit dem Dienstprogramm BMEND die dazugehörige(n) ALOG-Datei(en) darauf nachfahren (Anweisung `UPDATE-DATABASE`).

Erstellen von Online-Sicherungen

Zum Erstellen von Online-Sicherungen von Datenbanken bzw. einzelnen Realms sollten Sie grundsätzlich die entsprechenden Anweisungen von HSMS bzw. ARCHIVE verwenden.

! VORSICHT!

Bei Verwendung des COPY-FILE zur Erzeugung einer Online-Kopie wird vom System nicht geprüft, ob für die betreffende Datenbank die "Online Sicherungsfähigkeit" eingeschaltet ist. Sie müssen daher in diesem Fall selbst sicherstellen, dass das AFIM-Logging eingeschaltet ist, damit später die Online-Kopie durch Nachfahren konsistent gemacht werden kann.

Beim Starten eines ändernden UDS-Dienstprogrammes müssen Sie darauf achten, dass alle evtl. vorher begonnenen Online-Sicherungsvorgänge für die zu bearbeitende Datenbank abgeschlossen sind, weil die Online-Kopie sonst möglicherweise unbrauchbar wird.

Die Erzeugung von Online-Kopien während des Ablaufs ändernder Dienstprogramme ist generell nicht zulässig.

Online-Sicherungsfähigkeit einer Datenbankkopie

Die Online-Sicherungsfähigkeit einer Datenbank ist sowohl in den UDS-Verwaltungsdaten als auch in den DVS-Katalogeinträgen der Datenbank-Dateien vermerkt. Beim Kopieren einer Datenbank, sei es mit COPY-FILE oder mit HSMS/ARCHIVE, gehen die Angaben zur Online-Sicherungsfähigkeit im DVS-Katalog abhängig von den verwendeten Parametern evtl. verloren. Daher müssen Sie vor dem Einsatz einer Datenbankkopie, bei deren Original die Online-Sicherungsfähigkeit eingeschaltet war, für die Konsistenz dieser Eigenschaft sorgen, z.B. durch einen erneuten Lauf des Dienstprogramms BMEND mit der Anweisung `ENABLE-ONLINE-COPY`.

7.3.1.4 Schattendatenbank

Eine Schattendatenbank ist eine konsistente Kopie der Datenbank, bei der alle Datenbankdateinamen den Suffix *copyname* enthalten.

Eine Schattendatenbank können Sie parallel zur prozessierten Originaldatenbank

- lesen, um z.B. Auskünfte oder Statistiken zu erstellen, ohne den laufenden Betrieb zu behindern
- mit den After-Images der abgeschlossenen ALOG-Datei(en) der Originaldatenbank ständig aktualisieren, sodass Sie im Fehlerfall auf einen relativ aktuellen Datenbestand zurückgreifen können.

Die Schattendatenbank erzeugen Sie, indem Sie die Dateien der Originaldatenbank (ohne die ALOG-Dateien) kopieren oder eine eingespielte Sicherung umbenennen. Die Dateinamen der Datenbankdateien einer Schattendatenbank erhalten einen frei wählbaren Suffix, genannt *copyname*, sodass allgemein für die Schattendatenbank gilt

dbdateiname.copyname

dbdateiname

Name der Datenbankdatei (z.B. *dbname.realmname*)

Die Angabe *:catid:\$userid.dbname.realmname.copyname* darf maximal 54 Zeichen sein.

copyname

Suffix („Name“) der Schattendatenbank. *copyname* darf maximal sieben Zeichen lang sein.

Falls die Online-Sicherungsfähigkeit bei der Originaldatenbank eingestellt war, müssen Sie die Hinweise des Abschnittes „Online-Sicherungsfähigkeit einer Datenbankkopie“ im [Abschnitt „Online-Sicherung einer Datenbank“](#) beachten.

Die Schattendatenbank hat **keine** eigenen ALOG-Dateien. Zum Aktualisieren werden die ALOG-Dateien der Originaldatenbank verwendet.

i Änderungen des DBDIR für Schattendatenbanken werden nicht in der ALOG-Datei festgehalten. Wenn Sie den DBDIR der Schattendatenbank verwaltungsmäßig ändern (z.B. durch die Anweisungen ENABLE, DISABLE, ADD, REMOVE) und dann diesen DBDIR wieder zum Original machen, sind Sie für dessen Zustand selbst verantwortlich (zu-/abgeschaltete Realms, ein-/ausgeschaltete Online-Sicherungsfähigkeit).

Innerhalb einer Multi-DB-Konfiguration müssen die Subschemaamen eindeutig sein, d.h. Sie können Original- und Schattendatenbank derselben Datenbank **nicht** in einer Konfiguration verwenden. Original- und Schattendatenbank verschiedener Datenbanken können Sie dagegen in einer Konfiguration angeben.

Die Schattendatenbank wird wie eine Originaldatenbank ohne AFIM-Logging mit RETRIEVAL behandelt. Anwenderprogramme können nicht zwischen Original- und Schattendatenbank unterscheiden, sie sprechen sie über dieselben Subschemaamen an.

Verwenden Sie einen älteren Stand einer Schattendatenbank, so müssen eventuell alte Anwendermodule, SSITAB-Module usw. vorhanden sein.

Parallel zu lesenden Anwenderprogrammen können auch lesende UDS/SQL-Dienstprogramme (z.B. BCHECK, kopierender BOUTLOAD, BPRECORD, BPSIA, BPSQLSIA und BSTATUS) auf der Schattendatenbank arbeiten. Von den ändernden Dienstprogrammen ist für die Schattendatenbank nur BMEND zugelassen. Abgeschaltete Realms können von BMEND auch parallel zum Schattendatenbankbetrieb nachgefahren werden.

7.3.2 ALOG-Dateien führen

Datenbanksicherungen enthalten den Datenbestand zum Zeitpunkt der Sicherung. Um im Fehlerfall einen möglichst aktuellen Stand der Daten rekonstruieren zu können, müssen die Änderungen zwischen den Sicherungszeitpunkten erfasst werden. UDS/SQL protokolliert die Änderungen (After-Images) von Daten in der ALOG-Datei. Die ALOG-Datei ist datenbankspezifisch und hat den Namen

dbname.A.alog-seq-nr

dbname

Name der Datenbank

alog-seq-nr

neunstellige Nummer, Folgenummer der ALOG-Datei

Die Protokollierung der Änderungen in die ALOG-Datei müssen Sie mit dem Dienstprogramm BMEND (Anweisungen START-LOG/STOP-LOG) vor dem Datenbankbetrieb vereinbaren. Falls noch nicht geschehen, wird bei START-LOG gleichzeitig eine ALOG-Datei angelegt und initialisiert.

Sie können die ALOG-Datei selbst anlegen mit dem Kommando CREATE-FILE ...PROTECTION=*PAR(...,USER-ACCESS=*ALL-USERS,...). Dies ist dann notwendig, wenn der DBH **nicht** in der Datenbankkennung gestartet wird, da der DBH nicht kennungsübergreifend ALOG-Dateien anlegen kann. Die ALOG-Dateien werden stets unter der Datenbankkennung auf Platte erwartet.

Bei eingeschaltetem AFIM-Logging weist der DBH solange Änderungstransaktionen ab, bis eine ALOG-Datei zur Verfügung steht.

ALOG-Datei-Wechsel

Auf einen Sicherungsstand einer Datenbank fahren Sie mit dem Dienstprogramm BMEND abgeschlossene ALOG-Dateien nach, um einen aktualisierten, konsistenten Sicherungsstand zu erzeugen.

Eine ALOG-Datei wird nur bei einem ALOG-Datei-Wechsel ordnungsgemäß abgeschlossen. Wenn AFIM-Logging eingeschaltet ist, wird die ALOG-Datei in folgenden Fällen gewechselt:

- Der DBH-Ladeparameter PP STDCKPT=YES ist angegeben und der DBH wird normal beendet, nachdem Sie Änderungen durchgeführt haben.
- Der DBH-Ladeparameter PP STDCKPT=YES ist angegeben und der Session-Wiederanlauf ist erfolgreich abgelaufen.
- Der DBH-Ladeparameter PP STDCKPT=YES ist angegeben und beim DBH-Start ist eine ALOG-Datei vorhanden, die bereits ein nicht leeres Logintervall enthält.
- Bei der Ausführung des DAL-Kommandos CHECKPOINT
- Bei ALOG-Datei-Überlauf
- Ein änderndes Dienstprogramm wird beendet.
- Ein änderndes Dienstprogramm wird gestartet und eine ALOG-Datei ist vorhanden, die bereits ein nicht leeres Logintervall enthält
- Wenn die BMEND-Anweisung STOP-LOG angegeben ist und eine ALOG-Datei vorhanden ist, die bereits ein nicht leeres Logintervall enthält.

i Ein ALOG-Datei-Wechsel - und somit auch ein Abschließen der ALOG-Datei - erfolgt nicht beim Ausschließen einer Datenbank aus einer laufenden Session (DROP DB) und beim Beenden einer DBH-Session im Fall PP STDCKPT=NO.

Unabhängig davon, ob AFIM-Logging eingeschaltet ist, wird mit der BMEND-Anweisung START-LOG die ALOG-Datei in folgenden Fällen gewechselt:

- die aktuelle ALOG-Datei steht nicht zur Verfügung
- die aktuelle ALOG-Datei enthält bereits ein nicht leeres Logintervall
- die aktuelle ALOG-Datei wurde inkonsistent beendet

Beim ALOG-Datei-Wechsel wird die intern verwaltete Folgennummer der aktuellen ALOG-Datei um 1 erhöht.

7.3.3 Datenbank rekonstruieren

Sie müssen eine Datenbank oder Teile von ihr (ein oder mehrere Realms) rekonstruieren, wenn diese Datei bzw. Dateien z.B. in der Originaldatenbank ausgefallen, zerstört oder nicht mehr zugreifbar ist bzw. sind. Die Rekonstruktion besteht aus zwei Schritten:

- Zurücksetzen auf einen Sicherungsstand
- Aktualisieren des Sicherungsstandes mit ALOG-Dateien

Die Zeitdauer der Rekonstruktion hängt davon ab, wie aktuell Sie Ihren Sicherungsstand halten und wie umfangreich Ihre nachzufahrenden, geänderten Daten sind.

Zurücksetzen auf einen Sicherungsstand

Die Basis für die Rekonstruktion der Datenbank bzw. Datenbankteile ist die Sicherung der Datenbank. Die Sicherung kann auf unterschiedlichem Wege erzeugt worden sein:

- Sicherung mit HSMS bzw. ARCHIVE
- Online-Sicherung mit HSMS bzw. ARCHIVE
- Kopieren der Datenbank
- Erzeugen einer Schattendatenbank
- Aktualisieren eines Sicherungsstandes

Zum Zurücksetzen, d.h. dem Erzeugen des Originals muss z.B. die Sicherung wieder eingespielt (HSMS bzw. ARCHIVE), eventuell noch konsistent gemacht werden (Online-Sicherung), oder zurückkopiert bzw. nur umbenannt werden (Schattendatenbank).

Wenn Sie die gesamte Datenbank auf einen konsistenten Stand zurücksetzen, so können Sie auf diesem Sicherungsstand den Datenbankbetrieb sofort wieder aufnehmen. Jedoch sind damit alle angefallenen Änderungen ab diesem Zeitpunkt rückgängig gemacht worden. Die ALOG-Dateien ab diesem Zeitpunkt werden überschrieben. Falls Sie die ALOG-Dateien noch benötigen, müssen Sie sie vorher sichern.

Wenn Sie nur einen Realm bzw. mehrere Realms zurücksetzen, so müssen Sie diese Teile mit ALOG-Dateien aktualisieren, ehe Sie den Datenbankbetrieb wieder aufnehmen können.

Aktualisieren des Sicherungsstandes

Mit dem Dienstprogramm BMEND ist es möglich, die gesamte Datenbank, einzelne oder ausgewählte Realms mit einer ALOG-Datei bzw. mehreren ALOG-Dateien zu aktualisieren. BMEND können Sie für die Originaldatenbank, die Schattendatenbank und abgeschaltete Realms verwenden. Die ALOG-Dateien fahren Sie mit der BMEND-Anweisung UPDATE-DATABASE nach.

Die BMEND-Anweisung SHOW-LOG-INFORMATION liefert Ihnen die Angaben, mit welchen ALOG-Dateien die einzelnen Realms zu aktualisieren sind, bis Sie einen konsistenten Stand der gesamten Datenbank erreicht haben, d.h. diese ALOG-Dateien müssen Sie für das Nachfahren zur Verfügung stellen.

Desweiteren erhalten Sie Informationen darüber, ob in der Sequenz der ALOG-Dateien eine Logging-Lücke existiert oder eine ALOG-Datei inkonsistent beendet wurde. In derartigen Fällen ist eine Aktualisierung nur bis zu diesem Zeitpunkt oder erst mit einer Sicherung nach diesem Zeitpunkt möglich.

Nachfahren abgeschlossener ALOG-Dateien

Aktualisieren heißt, die geänderten Daten von der ALOG-Datei in die Datenbank (Original- oder Schattendatenbank) zu übernehmen. Es wird eine abgeschlossene, d.h. gewechselte ALOG-Datei vollständig

nachgefahren. Bis zu welcher ALOG-Datei nachzufahren ist, geben Sie mit dem DEADLINE-Parameter der BMEND-Anweisung UPDATE-DATABASE an. BMEND ermittelt sich die erste ALOG-Datei selbst, mit der das Nachfahren eines Realms oder der Datenbank zu beginnen ist. Diese wie auch die nachfolgenden ALOG-Dateien bis zur Deadline müssen Sie zur Verfügung stellen.

Das Nachfahren abgeschlossener ALOG-Dateien können Sie auch über eine Jobvariable (LINK-NAME = JVBMENT) innerhalb einer Prozedur steuern. Mit der BMEND-Anweisung SHOW-LOG-INFORMATION wird die Jobvariable mit Informationen zum Zustand des DBDIR und des Log Pools versorgt, u.a. mit dem Namen der zuerst nachzufahrenden ALOG-Datei. Nach dem erfolgreichen Nachfahren einer ALOG-Datei wird die Jobvariable mit der nächsten ALOG-Datei aktualisiert, falls Sie die vorgegebene Deadline noch nicht erreicht haben. Weitere Informationen finden Sie im Handbuch „[Sichern, Informieren und Reorganisieren](#)“ unter BMEND.

Nachfahren der aktuellen ALOG-Datei

In verschiedenen Fehlerfällen ist es nicht ausreichend, nur bis zum letzten konsistenten Stand einer Datenbank (entsprechend letzte abgeschlossene ALOG-Datei) zu aktualisieren. Um z.B. einen ausgefallenen Realm aus einer abgebrochenen Session zu rekonstruieren, müssen auch die Änderungen der aktuellen, in diesem Falle auch inkonsistenten ALOG-Datei übernommen werden. Das Nachfahren der aktuellen ALOG-Datei bewirken Sie mit der Angabe DEADLINE=BREAK-POINT in der BMEND-Anweisung UPDATE-DATABASE.

Falls die letzte nachgefahrte ALOG-Datei inkonsistent ist, müssen Sie anschließend einen Warmstart durchführen.

Das Nachfahren der aktuellen ALOG-Datei ist nur auf der Originaldatenbank zulässig.

Aktualisieren einer Schattendatenbank

Beim Aktualisieren der Schattendatenbank werden die abgeschlossenen ALOG-Dateien der Originaldatenbank verwendet. Sobald ein ALOG-Datei-Wechsel auf der Originaldatenbank stattgefunden hat, können Sie diese abgeschlossene ALOG-Datei auf der Schattendatenbank nachfahren. Auf diese Weise halten Sie den Datenbestand der Schattendatenbank sehr aktuell, sodass in einem Fehlerfall sich die Rekonstruktionszeit fast nur auf das Nachfahren der aktuellen ALOG-Datei begrenzt.

Sie können die Schattendatenbank parallel zum Betrieb auf der Originaldatenbank aktualisieren. Sie können nur abgeschlossene ALOG-Dateien nachfahren.

Aktualisieren eines abgeschalteten Realms

Im Falle eines Hardware-Fehlers auf einem Datenbank-Realm wird dieser Realm vom Datenbankbetrieb ausgeschlossen. Dabei wird in der Datenbank ein Konsistenzpunkt gesetzt und außerdem die ALOG-Datei gewechselt. Bei der Rekonstruktion des Realms können Sie parallel zum Datenbankbetrieb auf der Originaldatenbank einen Sicherungsstand des Realms einspielen und mit den zugehörigen ALOG-Dateien aktualisieren. Den rekonstruierten, konsistenten Realm können Sie dann wieder an den Datenbankbetrieb anschließen.

ALOG-Datei-Ausfall beim Warmstart

Für den Ausnahmefall, dass Ihnen die ALOG-Datei durch Hardware-Fehler während des Warmstarts ausfällt, können die offenen Transaktionen auf der ALOG-Datei nicht mehr zurückgesetzt werden. In diesem Fall können Sie mit der BMEND-Anweisung KILL-LOG ein Abschalten des AFIM-Logging erzwingen. Dadurch wird ein zweiter Warmstart **ohne** AFIM-Logging mit Hilfe der RLOG-Datei ermöglicht.

Es entsteht jedoch eine Logging-Lücke. Um für eine zukünftige Rekonstruktion der Datenbank eine Basis zu haben, sollten Sie sich nach dem Warmstart (Konsistenzpunkt der Datenbank) und dem Einschalten des AFIM-Loggings (BMEND-Anweisung START-LOG) wieder eine Sicherung der gesamten Datenbank erstellen.

7.4 Auswahl des Sicherungskonzeptes

Die Auswahl des für Sie geeigneten Sicherungskonzeptes wird sich an verschiedenen Kriterien und deren Gewichtung orientieren. Im Folgenden werden einige Konzepte vorgestellt unter Beachtung folgender Kriterien:

- Aktualität der Daten (z.B. Verluste bereits geänderter Daten)
- Aufwand für die Sicherung (z.B. Platz-, Gerätebedarf, Kosten der Sicherungserstellung)
- Verfügbarkeit (Rekonstruktionszeit nach einem Fehlerfall)

Die Reihenfolge ist gemäß der zunehmenden Verfügbarkeit der Datenbank angeordnet.

Keine Sicherung

Der Datenbankbetrieb wird ohne jegliche Sicherung und AFIM-Logging durchgeführt.

Aktualität

Nach einem Fehlerfall sind sämtliche Daten - Ausgangsbestand wie auch gerade geänderte - verloren.

Aufwand

Keiner.

Verfügbarkeit

Keine. Die Datenbank muss neu generiert und geladen werden.

Periodische Sicherung der Datenbank

Die gesamte Datenbank wird periodisch gesichert. Der Datenbankbetrieb findet ohne AFIM-Logging statt (z.B. wegen Performancesteigerung). Im Fehlerfall wird auf die letzte Datenbanksicherung zurückgesetzt.

Aktualität

Die geänderten Daten seit der letzten Sicherung sind verloren.

Aufwand

Sicherung der Datenbank und deren Archivierung. Falls im Fehlerfall die geänderten Daten wieder erzeugt werden sollen, z.B. durch Wiederholen der Änderungstransaktionen nach der Sicherung, ist zusätzlich eine eigene Sicherung der Änderungstransaktionen in ihrer zeitlich richtigen Reihenfolge und deren erneute Durchführung nötig.

Verfügbarkeit

Die Verfügbarkeit der Datenbank im Fehlerfall ist sehr gering. Sie wird bestimmt von der Zeit, um auf die letzte Sicherung zurückzusetzen und eventuell der Zeit, um Daten wiederherzustellen (Wiederholen der gesicherten Transaktionen).

Während der Datenbanksicherung ist kein Datenbankbetrieb möglich.

Periodische Sicherung und AFIM-Logging

Die Datenbank wird periodisch gesichert. Während des Datenbankbetriebs werden die Änderungen in der ALOG-Datei bzw. den ALOG-Dateien erfasst.

Aktualität

Kein Datenverlust, die verschiedenen Stände zwischen den Sicherungen müssen jedoch rekonstruiert werden.

Aufwand

- Erstellen der Sicherung (während der Sicherung ist kein Datenbankbetrieb möglich).
- Archivierung der gesicherten Datenbank sowie der ALOG-Dateien.
- Platz- und Gerätebedarf für das AFIM-Logging.
- Etwas länger dauernde Transaktionen wegen I/O für das AFIM-Logging.

Verfügbarkeit

Die Verfügbarkeit der Datenbank im Fehlerfall ist gering.

- Es wird Zeit benötigt, um auf die Sicherung zurückzusetzen.
- Es wird Zeit benötigt, um den Sicherungsstand mit ALOG-Dateien zu aktualisieren. Die Zeit hängt von der Zahl und dem Umfang der nachzufahrenden ALOG-Dateien ab.
- Während der Datenbanksicherung ist kein Datenbankbetrieb möglich.

Online-Sicherung und AFIM-Logging

Während des Datenbankbetriebs wird zu den Sicherungszeitpunkten eine Online-Sicherung erstellt. Der Datenbankbetrieb darf nur mit AFIM-Logging durchgeführt werden.

Aktualität

Kein Datenverlust, mögliche Zwischenstände müssen aber rekonstruiert werden.

Aufwand

- Platz- und Gerätebedarf für die Online-Sicherung (abhängig von der Größe der Datenbank).
- Archivierung der gesicherten Datenbank sowie der ALOG-Dateien.
- Platz- und Gerätebedarf für das AFIM-Logging.
- Etwas länger dauernde Transaktionen wegen AFIM-Logging.

Verfügbarkeit

Die Verfügbarkeit der Datenbank wird weiter verbessert, da die Sicherung während des Datenbankbetriebs erfolgen kann.

Im Fehlerfall ist die Verfügbarkeit abhängig von der

- Zeitdauer, um die Online-Sicherung wieder einzuspielen

- Zeit, um evtl. den Sicherungsstand mit ALOG-Dateien konsistent zu machen und zu aktualisieren.

Schattendatenbank und AFIM-Logging

Parallel zur Originaldatenbank wird eine Schattendatenbank geführt. Bei einem ALOG-Datei-Wechsel (z.B. durch das DAL-Kommando CHECKPOINT) wird die abgeschlossene ALOG-Datei sofort in der Schattendatenbank nachgefahren.

Aktualität

Hoher Aktualitätsgrad der Sicherung auch im Fehlerfall, es müssen nur noch die Daten der aktuellen ALOG-Datei nachgefahren werden.

Aufwand

- Erstellen der Schattendatenbank
- Platz- und Gerätebedarf für die Schattendatenbank (ist vor allem bei großen Datenbanken zu beachten).
- Platz- und Gerätebedarf für das AFIM-Logging.
- Etwas länger dauernde Transaktionen wegen AFIM-Logging.
- Sofortiges Aktualisieren der Schattendatenbank, nachdem die ALOG-Datei gewechselt wurde.

Verfügbarkeit

Die Verfügbarkeit der Datenbank ist auch im Fehlerfall sehr hoch.

Während des normalen Datenbankbetriebs kann die Schattendatenbank erstellt - via Online-Sicherung - und aktualisiert werden.

Im Fehlerfall kann sofort nach dem Ersetzen der defekten DB-Teile (z.B. durch Umbenennen der Schattendatenbank-Teile) und dem Nachfahren der aktuellen ALOG-Datei der Datenbankbetrieb fortgesetzt werden.

7.5 Beispiele

Beispiel 1

Die Datenbank VERSAND wird mit HSMS online gesichert. Voraussetzung für eine Online-Sicherung ist, dass das AFIM-Logging eingeschaltet ist und die Datenbank mit der BMEND-Anweisung ENABLE-ONLINE-COPY online-sicherungsfähig gemacht wurde. (Hinweis: Zusätzlich zur im Beispiel gezeigten Online-Sicherung sind zur Wiederherstellung einer Datenbank die passenden ALOG-Dateien erforderlich.)

```
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,      VERSION=02.9A00  
/START-UDS-BMEND
```

```
***** START          BMEND          (UDS/SQL V2.9 0000 )      2017-06-28   11:26:28
```

```
//ALLOCATE-BUFFER-POOL BUFFER-SIZE=*STD  
//OPEN-DATABASE DATABASE-NAME=VERSAND
```

```
***** DATABASE ORIGINAL WITHOUT AFIM LOGGING  
          FUNCTION ENABLE NOT AVAILABLE  
          FUNCTION KILL NOT AVAILABLE  
***** CONSISTENT   DATABASE DIRECTORY
```

```
//START-LOG DEFAULT-SUPPORT=*PUBLIC(CATID=*OWN) ,RESERVE-SUPPORT=*PUBLIC(CATID=ABN2) ,  
SPACE=*STD,RESET-LOG-POOL=*NO
```

```
***** LOGGING WILL BE ACTIVATED  
          FUNCTION ENABLE AVAILABLE FROM NOW ON
```

```
//ENABLE-ONLINE-COPY  
//END
```

```
***** BEGIN          FUNCTION START LOGGING AT 11:26:29  
ALOG FILE CREATED ACCORDING TO DEFAULT-SUPPORT  
***** NORMAL      END FUNCTION START LOGGING AT 11:26:29  
***** BEGIN          FUNCTION ENABLE ONLINE COPY AT 11:26:29  
***** ONLINE COPY FOR DATABASE $XXXXXXXXX.VERSAND ALLOWED  
***** NORMAL      END FUNCTION ENABLE ONLINE COPY AT 11:26:29  
  
***** DIAGNOSTIC SUMMARY OF BMEND  
  
          NO WARNINGS  
          NO ERRORS  
          NO SYSTEM-ERRORS  
  
***** END OF DIAGNOSTIC SUMMARY  
***** NR OF DATABASE ACCESSES   :           147  
***** NORMAL END      BMEND          (UDS/SQL V2.9 0000 )      2017-06-28   11:26:29
```

```

/START-HSMS
//BACKUP-FILES      FILE-NAMES=( -
//      VERSAND.COSSD -
// ,      VERSAND.DBCOM -
// ,      VERSAND.DBDIR -
// ,      VERSAND.HASHLIB -
// ,      VERSAND.ARTIKELRLM -
// ,      VERSAND.AUFTRAGSRLM -
// ,      VERSAND.BESTELLRLM -
// ,      VERSAND.HAUSHALT -
// ,      VERSAND.KLEIDUNG -
// ,      VERSAND.LEBENSMITTEL -
// ,      VERSAND.SCHREIBWAREN -
// ,      VERSAND.SPIELE-HOBBY -
// ,      VERSAND.SPORT -
// ),-
//  SAVE-OPTIONS=*PAR(SAVE-ONLINE-FILES=*YES), -
//  ARCHIVE-NAME=UDS.ARC, -
//  OPERATION-CONT=*PAR(OUT=UDS.ARC.SYSLST.BACKUP, -
//                        REPORT=*FULL,WAIT-FOR-COMPLETION=*YES)
//END

```

```

REQUEST-ENVIRONMENT=SF
REQUEST-NAME=BCF#4TS0 REQUEST-DATE=2017-06-28 09:27:32 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED WITHOUT
ERROR

```

STATEMENT LISTING:

```

BACKUP-FILES  FILE-NAMES=( VERSAND.COSSD ,  VERSAND.DBCOM ,  VERSAND.DBDIR ,  VERSAND.
HASHLIB ,
VERSAND.ARTIKELRLM ,  VERSAND.AUFTRAGSRLM ,  VERSAND.BESTELLRLM ,VERSAND.HAUSHALT ,  VERSAND.
KLEIDUNG ,
VERSAND.LEBENSMITTEL ,  VERSAND.SCHREIBWAREN ,  VERSAND.SPIELE-HOBBY ,  VERSAND.SPORT ) ,
SAVE-OPTIONS=  *PAR(SAVE-ONLINE-FILES=*YES),  ARCHIVE-NAME=UDS.ARC,
OPERATION-CONT=*PAR(OUT=UDS.ARC.SYSLST.BACKUP,  REPORT=*FULL,WAIT-FOR-COMPLETION=*YES)

```

```

ENVIRONMENT                : SF
ARCHIVE-NAME                : $XXXXXXXXX.UDS.ARC
SAVE-FILE ATTRIBUTES       : NEW
TO-STORAGE                  : S2-STORAGE-LEVEL
DEVICE-TYPE                 : TAPE-C4
RETENTION-PERIOD           : 30

SAVE-VERSION ATTRIBUTES
SAVE-VERSION-NAME          :

SELECT-FILES PARAMETER
MODIFIED-FILES PARTIAL-FILE-SAVE : NO
MAX-BACKUP-CLASS           : STD

```

```

***   BACKUP - FILES           HSMS V09.0           FULL           REPORT   *** 2017-06-28
09:27:41   PAGE      2
      REQUEST-ENVIRONMENT=SF
      REQUEST-NAME=BCF#4TS0 REQUEST-DATE=2017-06-28 09:27:32 USER-ID=XXXXXXXX REQUEST-
STATE=COMPLETED   WITHOUT
ERROR
    
```

```

% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.150628.092733', VERSION '09.0A09'
% ARC0033 ARCHIVE SUBTASK TSN '4TS1' GENERATED
% ARC0825 SUBTASK '0' HAS TRANSFERRED '1475' PAM PAGES FOR '13' FILES AND '0' JVS WITH 256K-
BLOCKING IN '0'
SECONDS
    
```

```

      SAVE FILE IDENTIFIER - S.150628.092733           SAVE-VERSION-DATE=15-06-28 SAVE-VERSION-
TIME=09:27:33
    
```

```

      SUBSAVE
      NUMBER           VSNS
      0                QE0146
    
```

```

      SAVE FILE IDENTIFIER - S.150628.092733           SAVE-VERSION-DATE=15-06-28 SAVE-VERSION-
TIME=09:27:33
    
```

```

***   CATALOG - SQL2           USER - XXXXXXXX   ***
      FILE/JOB VARIABLE NAME           LASTPG/   SAVE   INPUT DEV SUB   OUTPUT
      VERS           SIZE   TYPE           VSN   TYP SAVE   VSN(S)
    
```

FILE/JOB VARIABLE NAME	VERS	SIZE	TYPE	INPUT VSN	DEV TYP	SUB SAVE	OUTPUT VSN(S)
VERSAND.ARTIKELRLM	1	124	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.AUFTRAGSRLM	1	76	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.BESTELLRLM	1	120	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.COSSD	1	12	FULL	SQL2.3	D	0	QE0146
VERSAND.DBCOM	1	500	FULL	SQL2.4	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.DBDIR	1	200	FULL	SQL2.4	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.HASHLIB	1	9	FULL	SQL2.6	D	0	QE0146
VERSAND.HAUSHALT	1	48	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.KLEIDUNG	1	108	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.LEBENSMITTEL	1	36	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.SCHREIBWAREN	1	48	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.SPIELE-HOBBY	1	88	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							
VERSAND.SPORT	1	92	FULL	SQL2.3	D	0	QE0146
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'							

```

      NUMBER OF FILES=           13 GLOBAL SIZE=           1461 START= 2017-06-28 09:27:32 END=
2017-06-28 09:27:41
    
```

*** E N D O F HSMS V09.0 FULL REPORT *** 2017-06-28
09:27:41 ***

Beispiel 2

Die Online-Sicherung der Datenbank VERSAND wird von HSMS eingespielt. Der Zustand der Sicherung (Konsistenz, Log-Intervall) wird mit BMEND abgefragt. Die Datenbank wird anschließend unter Verwendung der passenden ALOG-Dateien bis zum letztmöglichen Konsistenzpunkt mit BMEND nachgefahren.

```

/START-HSMS
//RESTORE-FILES -
// FILE-NAMES=(VERSAND.),-
// ARCHIVE-NAME=UDS.ARC,-
// REPLACE-FILES=YES,-
// SELECT-SAVE-VERSIONS=*LATEST,-
// OPERATION-CONT=*PAR(OUT=UDS.ARC.SYSLST.RESTORE, -
// REPORT=*FULL,WAIT-FOR-COMPLETION=*YES)
//STEP
//END

```

```

REQUEST-ENVIRONMENT=SF
REQUEST-NAME=RSF#4TSZ REQUEST-DATE=2017-06-28 09:27:49 USER-ID=SYSHSMS REQUEST-
STATE=COMPLETED WITHOUT
ERROR

STATEMENT LISTING:

RESTORE-FILES FILE-NAMES=(VERSAND.), ARCHIVE-NAME=UDS.ARC, REPLACE-FILES=YES,
SELECT-SAVE-
VERSIONS=*LATEST,
OPERATION-CONT=*PAR(OUT=UDS.ARC.SYSLST.RESTORE, REPORT=*FULL,WAIT-FOR-
COMPLETION=*YES)

% ARC0002 STATEMENT ACCEPTED. ARCHIVE SEQUENCE NUMBER 'A.150628.092749', VERSION '09.0A09'
% ARC0033 ARCHIVE SUBTASK TSN '4TS2' GENERATED
% ARC0815 SUBTASK '0' HAS TRANSFERRED '1475' PAM PAGES FOR '13' FILES AND '0' JVS IN '0'
SECONDS

SAVE FILE IDENTIFIER - S.150628.092733

SUBSAVE
NUMBER VSNS

0 QE0146

*** CATALOG - SQL2 USER - XXXXXXXX ***
FILE/JOB VARIABLE NAME LASTPG/ SAVE VERSION SAVE INPUT SUB
OUTPUT
VERSAND.ARTIKELRLM 1 124 150628.092733 FULL QE0146 0
SQL2.3
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.AUFTRAGSRLM 1 76 150628.092733 FULL QE0146 0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.BESTELLRLM 1 120 150628.092733 FULL QE0146 0

```

```

SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.COSSD                1      12  150628.092733  FULL  QE0146  0
SQL2.4
VERSAND.DBCOM                1      500 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.DBDIR               1      200 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.HASHLIB            1        9 150628.092733  FULL  QE0146  0
SQL2.6
VERSAND.HAUSHALT           1      48 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.KLEIDUNG           1     108 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.LEBENSMITTEL       1      36 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.SCHREIBWAREN      1      48 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.SPIELE-HOBBY       1      88 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'
VERSAND.SPORT              1      92 150628.092733  FULL  QE0146  0
SQL2.4
% ARC0059 FILE HAS BEEN SAVED 'ON LINE'

***      E N D      O F                HSMS V09.0      FULL                REPORT *** 2017-06-28  09:
27:53      ***
    
```

```

/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.9B00
/START-UDS-BMEND
    
```

```

***** START          BMEND          (UDS/SQL V2.9 0000 )    2017-06-28    11:26:35
    
```

```
//SHOW-LOG-INFORMATION REALM-NAME=*ALL,LOG-FILE=4,OUTPUT=*SYSOUT
```

```

***** DATABASE ORIGINAL WITHOUT AFIM LOGGING
***** FUNCTION ENABLE NOT AVAILABLE
***** FUNCTION KILL NOT AVAILABLE
***** CONSISTENT DATABASE DIRECTORY
    
```

```
//END
```

```

***** BEGIN          FUNCTION SHOW LOG INFORMATION AT 11:26:35
***** LOG INFORMATION FOR DATABASE $XXXXXXXX.VERSAND
***** LOG INTERVAL OF SPECIFIED REALMS

REALM-NAME          !      ALOG SEQ NR      !          !
                   ! BEGIN      ! END          ! CONSISTENT !
-----+-----+-----+-----+-----+
    
```

```

DATABASE-DIRECTORY          !          5 !          5 !        YES        !
DATABASE-COMPILER-REALM     !          5 !          5 !        YES        !
AUFTRAGSRLM                 !          5 !          5 !        YES        !
BESTELLRLM                  !          5 !          5 !        YES        !
KLEIDUNG                    !          5 !          5 !        YES        !
HAUSHALT                    !          5 !          5 !        YES        !
SPORT                       !          5 !          5 !        YES        !
LEBENSMITTEL                !          5 !          5 !        YES        !
SPIELE-HOBBY                !          5 !          5 !        YES        !
SCHREIBWAREN                !          5 !          5 !        YES        !
ARTIKELRLM                  !          5 !          5 !        YES        !

```

***** LOG MODE : NO LOGGING

***** SUPPORTS OF ACTUAL LOG FILE :

***** INFORMATION ABOUT LOG HISTORY :

```

ALOG SEQ NR!          LOG INTERVAL          ! AFIM !BACKOUT! LOGGING !
      *-----*-----*
      !   BEGIN   !   END   !   !   BFIM !   GAP   !
-----*-----*-----*-----*-----*
      4 !20170628112638!20170628112639!   *   !   !   ?   !
      3 !20170628112637!20170628112638!   *   !   !   !   !
      2 !20170628112636!20170628112637!   *   !   !   !   !
      1 !20170628112635!20170628112636!   *   !   !   !   !

```

***** ONLINE COPIES BY ARCHIVE ARE NOT ALLOWED

***** NORMAL END FUNCTION SHOW LOG INFORMATION AT 11:26:35

***** DIAGNOSTIC SUMMARY OF BMEND

NO WARNINGS

NO ERRORS

NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY

***** NR OF DATABASE ACCESSES : 49

***** NORMAL END BMEND (UDS/SQL V2.9 0000) 2017-06-28 11:26:35

/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND.DBDIR

/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.9B00

/START-UDS-BMEND

...

```

***** START          BMEND          (UDS/SQL V2.9 0000 )          2017-06-28  11:26:35

```

//OPEN-DATABASE DATABASE-NAME=VERSAND

//UPDATE-DATABASE REALM-NAME=*ALL,DEADLINE=STD

SYSTEM_BREAK OCCURRED IN REALM DATABASE-DIRECTORY

***** INCONSISTENT DATABASE DIRECTORY

SYSTEM_BREAK OCCURRED IN REALM BESTELLRLM

SYSTEM_BREAK OCCURRED IN REALM LEBENSMITTEL

SYSTEM_BREAK OCCURRED IN REALM ARTIKELRLM

***** INCONSISTENT DATABASE DIRECTORY

FUNCTION ADD NOT AVAILABLE

FUNCTION REMOVE NOT AVAILABLE

FUNCTION START NOT AVAILABLE

FUNCTION STOP NOT AVAILABLE

***** STD UPDATE FOR ALL REALMS SPECIFIED

```
FUNCTION ADD AVAILABLE FROM NOW ON
FUNCTION REMOVE AVAILABLE FROM NOW ON
FUNCTION START AVAILABLE FROM NOW ON
FUNCTION STOP AVAILABLE FROM NOW ON
FUNCTION KILL NOT AVAILABLE
```

//END

```
***** BEGIN          FUNCTION UPDATE DATABASE AT 09:27:54
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000002' STARTED
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000002' FINISHED
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000003' STARTED
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000003' FINISHED
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000004' STARTED
***** MENDING WITH ALOG FILE '$XXXXXXXXX.VERSAND.A.000000004' FINISHED
***** ALOG FILE '$XXXXXXXXX.VERSAND.A.000000005' NOT USED FOR MENDING
***** NORMAL      END FUNCTION UPDATE DATABASE AT 09:27:55

***** DIAGNOSTIC SUMMARY OF BMEND

                NO WARNINGS
                NO ERRORS
                NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES   :           306
***** NORMAL END      BMEND      (UDS/SQL  V2.9  0000 )           2017-06-28   11:26:35
```

Beispiel 3

Von der Datenbank KUNDEN wird eine Schattendatenbank KUNDEN.KOP erzeugt.

/SHOW-FILE-ATTRIBUTES FILE-NAME=KUNDEN.

```

768 :SQL2:$XXXXXXXXX.KUNDEN.A.000000001
192 :SQL2:$XXXXXXXXX.KUNDEN.A.000000002
120 :SQL2:$XXXXXXXXX.KUNDEN.COSSD
501 :SQL2:$XXXXXXXXX.KUNDEN.DBCOM
201 :SQL2:$XXXXXXXXX.KUNDEN.DBDIR
 24 :SQL2:$XXXXXXXXX.KUNDEN.DBSTAT
 24 :SQL2:$XXXXXXXXX.KUNDEN.DBSTAT.SAVE
252 :SQL2:$XXXXXXXXX.KUNDEN.FINANZRLM
 12 :SQL2:$XXXXXXXXX.KUNDEN.HASHLIB
252 :SQL2:$XXXXXXXXX.KUNDEN.KUNDENRLM
 24 :SQL2:$XXXXXXXXX.KUNDEN.SLF
SUM PUBLIC:      11 FILES RES=      2370 FRE=      351 REL=      339 PAGES

```

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.COSSD, TO-FILE=KUNDEN.COSSD.KOP

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.DBCOM, TO-FILE=KUNDEN.DBCOM.KOP

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.DBDIR, TO-FILE=KUNDEN.DBDIR.KOP

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.HASHLIB, TO-FILE=KUNDEN.HASHLIB.KOP

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.KUNDENRLM, TO-FILE=KUNDEN.KUNDENRLM.KOP

/COPY-FILE FROM-FILE=:SQL2:KUNDEN.FINANZRLM, TO-FILE=KUNDEN.FINANZRLM.KOP

/SHOW-FILE-ATTRIBUTES FILE-NAME=KUNDEN.*.KOP

```

120 :SQL2:$XXXXXXXXX.KUNDEN.COSSD.KOP
501 :SQL2:$XXXXXXXXX.KUNDEN.DBCOM.KOP
201 :SQL2:$XXXXXXXXX.KUNDEN.DBDIR.KOP
252 :SQL2:$XXXXXXXXX.KUNDEN.FINANZRLM.KOP
 12 :SQL2:$XXXXXXXXX.KUNDEN.HASHLIB.KOP
252 :SQL2:$XXXXXXXXX.KUNDEN.KUNDENRLM.KOP
:SQL2: PUBLIC:      6 FILES RES=      1338 FRE=      119 REL=      117 PAGES

```

Beispiel 4

Der defekte Realm KUNDEN.FINANZRLM wird durch die Realm-Sicherung KUNDEN.FINANZRLM.KOP ersetzt. Der Zustand (Konsistenz, Log-Intervall) des Realms wird mit BMEND abgefragt. Anschließend wird der Realm bis zum letztmöglichen Konsistenzpunkt mit BMEND nachgefahren.

```
/DELETE-FILE :SQL2:KUNDEN.FINANZRLM
/COPY-FILE FROM-FILE=KUNDEN.FINANZRLM.KOP ,TO-FILE=:SQL2:KUNDEN.FINANZRLM
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.9B00
/START-UDS-BMEND
```

...

```
***** START          BMEND          (UDS/SQL V2.9 0000 )    2017-06-28  09:26:39
```

```
//OPEN-DATABASE DATABASE-NAME=KUNDEN
```

```
DIFFERENCE OF BACKUP DATA BETWEEN REALM FINANZRLM AND CONSISTENCY RECORD
***** INCONSISTENT DATABASE DIRECTORY
        FUNCTION START NOT AVAILABLE
        FUNCTION STOP NOT AVAILABLE
```

```
//SHOW-LOG-INFORMATION REALM-NAME=*ALL,OUTPUT=SYSLST
//END
```

```
***** BEGIN          FUNCTION SHOW LOG INFORMATION AT 09:26:40

        ***** LOG INFORMATION FOR DATABASE $XXXXXXXX.KUNDEN

        ***** LOG INTERVAL OF SPECIFIED REALMS

REALM-NAME          !      ALOG SEQ NR      !      !
                   ! BEGIN      ! END      ! CONSISTENT !
-----+-----+-----+-----+
DATABASE-DIRECTORY !      3 !      3 !      YES      !
DATABASE-COMPILER-REALM !      2 !      2 !      YES      !
KUNDENRLM          !      2 !      2 !      YES      !
FINANZRLM          !      1 !      1 !      YES      !

        ***** TO MAKE THE SPECIFIED REALMS CONSISTENT, THE FOLLOWING LOG FILES ARE
        NECESSARY :
        FROM ALOG SEQ NR          1 TO ALOG SEQ NR          3
        OR FROM LOG INTERVAL BEGIN 20170628092636 TO LOG INTERVAL END
        20170628092639

        ***** LOG MODE : AFIM LOGGING

        ***** SUPPORTS OF ACTUAL LOG FILE :

        DEFAULT SUPPORT :PVS ID = DEFAULT PVS
        RESERVE SUPPORT :PVS ID = DEFAULT PVS
***** INFORMATION ABOUT LOG HISTORY :

ALOG SEQ NR!          LOG INTERVAL          ! AFIM !BACKOUT! LOGGING !
```

```

*-----*-----*          !      !      !
!      BEGIN      !      END      !      !      BFIM !      GAP      !
*-----*-----*-----*-----*-----*
3      !20170628092639!20170628092639!      *      !      !      !
2      !20170628092636!20170628092639!      *      !      !      !
1      !20170628092635!20170628092636!      *      !      !      !
      ***** ONLINE COPIES BY ARCHIVE ARE NOT ALLOWED
***** NORMAL      END FUNCTION SHOW LOG INFORMATION AT 09:26:40

***** DIAGNOSTIC SUMMARY OF BMEND

      NO WARNINGS
      NO ERRORS
      NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES      :          20
***** NORMAL END      BMEND      (UDS/SQL      V2.9      0000 )      2017-06-28      09:26:40

```

/START-UDS-BMEND

```

...
***** START      BMEND      (UDS/SQL      V2.9      0000 )      2017-06-28      09:
26:40
//OPEN-DATABASE DATABASE-NAME=KUNDEN
DIFFERENCE OF BACKUP DATA BETWEEN REALM FINANZRLM AND CONSISTENCY RECORD
***** INCONSISTENT DATABASE DIRECTORY
      FUNCTION START NOT AVAILABLE
      FUNCTION STOP NOT AVAILABLE
//UPDATE-DATABASE REALM-NAME=FINANZRLM,DEADLINE=*STD
//END
***** BEGIN      FUNCTION UPDATE DATABASE AT 09:26:40
***** ALOG FILE '$XXXXXXXX.KUNDEN.A.000000001' NOT USED FOR MENDING
***** MENDING WITH ALOG FILE '$XXXXXXXX.KUNDEN.A.000000002' STARTED
***** MENDING WITH ALOG FILE '$XXXXXXXX.KUNDEN.A.000000002' FINISHED
***** ALOG FILE '$XXXXXXXX.KUNDEN.A.000000003' NOT USED FOR MENDING
***** NORMAL      END FUNCTION UPDATE DATABASE AT 09:26:40

***** DIAGNOSTIC SUMMARY OF BMEND

      NO WARNINGS
      NO ERRORS
      NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY
***** NR OF DATABASE ACCESSES      :          41
***** NORMAL END      BMEND      (UDS/SQL      V2.9      0000 )      2017-06-28      09:
26:40

```

Beispiel 5

Die Schattendatenbank KUNDEN.KOP soll aktualisiert werden. Der Zustand der Schattendatenbank (Konsistenz, Log-Intervall) im Vergleich zur Originaldatenbank wird mit BMEND abgefragt. Anschließend wird die Schattendatenbank bis zum letztmöglichen Konsistenzpunkt mit BMEND nachgefahren.

```

/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,      VERSION=02.9B00
/START-UDS-BMEND

...
***** START          BMEND          (UDS/SQL   V2.9   0000 )      2017-06-28   09:
26:40
//OPEN-DATABASE DATABASE-NAME=KUNDEN,COPY-NAME=KOP
***** PROCESSING OF DATABASE COPY
          FUNCTION START NOT AVAILABLE
          FUNCTION STOP NOT AVAILABLE
          FUNCTION KILL NOT AVAILABLE
***** CONSISTENT      DATABASE DIRECTORY
//SHOW-LOG-INFORMATION REALM-NAME=*ALL,OUTPUT=*SYSLST
//END

***** BEGIN          FUNCTION SHOW LOG INFORMATION AT 09:26:40
***** LOG INFORMATION FOR DATABASE $XXXXXXXX.KUNDEN.KOP
          ***** LOG INTERVAL OF SPECIFIED REALMS

REALM-NAME          !      ALOG SEQ NR          !      !
                   !  BEGIN      !  END          !  CONSISTENT !
-----+-----+-----+-----+
DATABASE-DIRECTORY !          2 !          2 !    YES      !
DATABASE-COMPILER-REALM !        1 !          1 !    YES      !
KUNDENRLM          !        1 !          1 !    YES      !
FINANZRML         !        1 !          1 !    YES      !
***** INFORMATION ABOUT LOG HISTORY :

ALOG SEQ NR!      LOG INTERVAL          ! AFIM !BACKOUT! LOGGING !
          *-----*-----*          !      !      !
          !  BEGIN      !  END          !      ! BFIM !  GAP      !
-----*-----*-----*-----*-----*
          2 !20170628092636!20170628092639!  * !      !  ?      !
          1 !20170628092635!20170628092636!  * !      !      !

          ***** ONLINE COPIES BY ARCHIVE ARE NOT ALLOWED

***** NORMAL      END FUNCTION SHOW LOG INFORMATION AT 09:26:40

***** DIAGNOSTIC SUMMARY OF BMEND
          NO WARNINGS
          NO ERRORS
          NO SYSTEM-ERRORS
***** END OF DIAGNOSTIC SUMMARY

```

```
***** NR OF DATABASE ACCESSES :          20
***** NORMAL END   BMEND           (UDS/SQL V2.9 0000 )    2017-06-28   09:26:40
```

/START-UDS-BMEND

...

```
***** START           BMEND           (UDS/SQL V2.9 0000 )    2017-06-28   09:26:40
```

//OPEN-DATABASE DATABASE-NAME=KUNDEN

```
***** CONSISTENT   DATABASE DIRECTORY
***** FUNCTION KILL NOT AVAILABLE
```

//SHOW-LOG-INFORMATION REALM-NAME=*ALL,OUTPUT=*SYSLST

//END

```
***** BEGIN           FUNCTION SHOW LOG INFORMATION AT 09:26:40

***** LOG INFORMATION FOR DATABASE $XXXXXXXX.KUNDEN

***** LOG INTERVAL OF SPECIFIED REALMS
                !      ALOG SEQ NR      !      !
REALM-NAME      ! BEGIN      ! END      ! CONSISTENT !
-----+-----+-----+-----+
DATABASE-DIRECTORY      !      3 !      3 !      YES      !
DATABASE-COMPILER-REALM !      2 !      2 !      YES      !
KUNDENRLM              !      2 !      2 !      YES      !
FINANZRLM              !      2 !      2 !      YES      !

***** LOG MODE : AFIM LOGGING

***** SUPPORTS OF ACTUAL LOG FILE :

DEFAULT SUPPORT :PVS ID = DEFAULT PVS
RESERVE SUPPORT :PVS ID = DEFAULT PVS

***** INFORMATION ABOUT LOG HISTORY :

ALOG SEQ NR!      LOG INTERVAL      ! AFIM !BACKOUT! LOGGING !
      *-----*-----*      !      !      !
      ! BEGIN      ! END      !      ! BFIM !      GAP      !
-----*-----*-----*-----*-----*
      3 !20170628092639!20170628092639! * !      !      !
      2 !20170628092636!20170628092639! * !      !      !
      1 !20170628092635!20170628092636! * !      !      !

***** ONLINE COPIES BY ARCHIVE ARE NOT ALLOWED

***** NORMAL   END FUNCTION SHOW LOG INFORMATION AT 09:26:40
***** NR OF DATABASE ACCESSES :          20
***** NORMAL END   BMEND           (UDS/SQL V2.9 0000 )    2017-06-28   09:26:40
```

/START-UDS-BMEND

...

```
***** START           BMEND           (UDS/SQL V2.9 0000 )    2017-06-28   09:
```

26:40

//OPEN-DATABASE DATABASE-NAME=KUNDEN,COPY-NAME=KOP

***** PROCESSING OF DATABASE COPY

FUNCTION START NOT AVAILABLE

FUNCTION STOP NOT AVAILABLE

FUNCTION KILL NOT AVAILABLE

***** CONSISTENT DATABASE DIRECTORY

//UPDATE-DATABASE REALM-NAME=*ALL,DEADLINE=*STD

//END

***** BEGIN FUNCTION UPDATE DATABASE AT 09:26:40

***** MENDING WITH ALOG FILE '\$XXXXXXXX.KUNDEN.A.00000002' STARTED

***** MENDING WITH ALOG FILE '\$XXXXXXXX.KUNDEN.A.00000002' FINISHED

***** ALOG FILE '\$XXXXXXXX.KUNDEN.A.00000003' NOT USED FOR MENDING

***** NORMAL END FUNCTION UPDATE DATABASE AT 09:26:41

***** DIAGNOSTIC SUMMARY OF BMEND

NO WARNINGS

NO ERRORS

NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY

***** NR OF DATABASE ACCESSES : 147

***** NORMAL END BMEND (UDS/SQL V2.9 0000) 2017-06-28

09:26:41

8 Leistungsoptimierung

Für das Leistungsverhalten von UDS/SQL ist es wesentlich, dass die Datenbanken für die Anwendungen gut strukturiert sind. UDS/SQL bietet vielfältige Möglichkeiten sowohl zur logischen als auch zur physischen Organisation der Daten und damit zur Optimierung der Zugriffe auf die Daten, siehe Handbuch „[Entwerfen und Definieren](#)“.

- Durch Vermeidung von unnötigen Ein- und Ausgaben beim Zugriff auf die Daten in den Datenbanken kann die Leistung wesentlich gesteigert werden. Im [Abschnitt „I/O-Verhalten optimieren“](#) wird beschrieben, wie Sie das Ein-/Ausgabeverhalten analysieren bzw. beeinflussen können.
- Durch die parallele Mehrfachnutzung von Programmen, die als Subsysteme geladen sind, kann der für die UDS/SQL-Konfiguration benötigte Arbeitsspeicher reduziert werden. Im [Abschnitt „Arbeitsspeichernutzung durch Subsystemfunktionalität von UDS/SQL optimieren“](#) wird beschrieben, welche Subsysteme für UDS/SQL erzeugt werden können.
- Bei Mehrprozessoranlagen wird das Leistungsverhalten des independent DBH stark durch die dynamische Verteilung der UDS/SQL-Tasks auf die Prozessoren beeinflusst. Im [Abschnitt „Prozessornutzung mit independent DBH optimieren“](#) wird beschrieben, wie Sie die Prozessornutzung analysieren bzw. beeinflussen können.

8.1 I/O-Verhalten optimieren

Bei größeren Datenbankanwendungen kann das I/O-Verhalten eine entscheidende Rolle für die Performance der Anwendung spielen. Im Folgenden wird beschrieben, bei welchen UDS/SQL-Dateien es zu I/O-Engpässen kommen kann, und wie diese beseitigt werden können.

Session-Log-File (SLF)

Bei der SLF kann kein I/O-Engpass auftreten.

Status-Datei

Bei der Status-Datei kann kein I/O-Engpass auftreten.

Realms

Wenn ein I/O-Engpass bei Realms auftritt, gibt es mehrere Lösungsmöglichkeiten:

- **Datenbankpuffer vergrößern**

Durch eine Vergrößerung des Datenbankpuffers lässt sich die Zahl der I/Os reduzieren, sodass der Engpass beseitigt werden kann. Allerdings müssen Sie dabei die Größe des Hauptspeichers berücksichtigen, damit die Paging-Rate nicht zu hoch wird.

Wichtig ist auch, dass der richtige Datenbankpuffer (2-Kbyte, 4-Kbyte bzw. 8-Kbyte) vergrößert wird. Sie müssen also vorher feststellen, welche Seitengröße die entsprechenden Datenbanken haben.

Mit dem UDS/SQL-Monitor können Sie ermitteln, ob die Vergrößerung des Datenbankpuffers eine Verringerung der physischen I/Os bewirkt hat.
- **Eigenen Datenbankpuffer für eine Datenbank anlegen**

Wenn eine bestimmte Datenbank besonders performancekritisch ist, können Sie für diese Datenbank einen eigenen Datenbankpuffer anlegen. Der Vorteil ist, dass Seiten dieser Datenbank nicht durch I/Os auf andere Datenbanken verdrängt werden können. Der Nachteil ist, dass dieser Datenbankpuffer nicht für I/Os auf andere Datenbanken genutzt werden kann.
- **Realms auf unterschiedliche Platten verteilen**

Wenn eine Vergrößerung des Datenbankpuffers nicht zum Erfolg führt, können Sie durch Verteilung der Realms auf unterschiedliche Platten die Zahl der I/Os auf eine Platte reduzieren und damit den I/O-Engpass beseitigen.
- **Einen Platten-Cache bzw. eine Solid State Disk (SSD) einsetzen**

Wenn der I/O-Engpass sich weder durch eine Vergrößerung des Datenbankpuffers noch durch die Verteilung der Realms auf unterschiedliche Platten beseitigen lässt, so können Sie zur Beschleunigung der I/Os einen Platten-Cache oder, bei noch höheren Anforderungen, einen externen Schnellspeicher einsetzen.
- **Anwendung ändern**

Manchmal tritt ein I/O-Engpass dadurch auf, dass in der Anwendung die Datenbankaufrufe in einer ungünstigen Folge durchgeführt werden. Eine diesbezügliche Änderung kann zu einer starken Reduzierung der I/Os führen. Dies erfordert jedoch eine detaillierte Analyse der Anwendung.

RLOG-Datei

Jede Änderungstransaktion erfordert einen Schreibvorgang auf die RLOG-Datei. Obwohl mit einem I/O mehrere Transaktionen beendet werden können (Group Commit), kann bei Anwendungen mit sehr vielen ändernden Transaktionen die RLOG-Datei zum Engpass werden. In diesem Fall gibt es folgende Lösungsmöglichkeiten:

- RLOG-Datei auf eigene Platte legen

Die RLOG-Datei wird sequenziell geschrieben. Die RLOG-Datei sollten Sie so anlegen, daß das Schreiben möglichst schnell und ohne Beeinflussung durch andere Aktivitäten auf dem System erfolgen kann.

- Einen Platten-Cache bzw. eine SSD einsetzen

Wenn der Engpass sich mit der vorhergenannten Maßnahme nicht beseitigen lässt, so ist der Einsatz eines Platten-Cache bzw. eines externen Schnellspeichers zu empfehlen.

- Einen Globalspeicher (GS) einsetzen

Wenn eine SSD auch nicht hilft den Engpass zu beseitigen, so kann das Pubset, auf dem die RLOG-Datei liegt, über einen Globalspeicher gepuffert werden. Wird die RLOG-Datei doppelt geführt, so müssen zwei Availability-Units des Globalspeichers zur Verfügung stehen, um die Ausfallunabhängigkeit der beiden RLOG-Dateien zu gewährleisten.

ALOG-Datei

Die ALOG-Datei ist im Allgemeinen nicht performancekritisch, da nur dann auf die ALOG-Datei geschrieben wird, wenn der ALOG-Puffer voll ist.

Wenn die ALOG-Datei trotzdem zu einem Engpass werden sollte, empfiehlt es sich, sie auf eine Platte zu legen, auf die keine anderen Zugriffe erfolgen. Die ALOG-Datei sollten Sie dann so anlegen, daß das Schreiben möglichst schnell und ohne Beeinflussung durch andere Aktivitäten auf dem System erfolgen kann.

Temporäre Realms

Temporäre Realms sind im Allgemeinen nicht performancekritisch.

Wenn sie es trotzdem sein sollten, hilft eine Vergrößerung des 4-Kbyte-Systempuffers (Temporäre Realms haben immer eine Seitengröße von 4-Kbyte). Wenn dies auch nicht ausreicht, müssen Sie die Temporären Realms auf mehrere Platten verteilen.

8.2 Arbeitsspeichernutzung durch Subsystemfunktionalität von UDS/SQL optimieren

Zur optimalen Unterstützung des Arbeitsspeichers wird UDS/SQL als Subsystem ausgeliefert. Es werden Objektmoduldateien für die Subsysteme UDS-SQL und UDS-D zur Verfügung gestellt.

Das Subsystem UDS-SQL enthält das sharefähige Coding, das zum lokalen Betrieb von UDS/SQL notwendig ist. Es kann im Allgemeinen in Verbindung mit linked-in und independent Anwendungen genutzt werden.

Das Subsystem UDS-D enthält das sharefähige Coding, das bei zusätzlicher Nutzung von UDS-D benötigt wird.

Die Subsysteme kann der BS2000-Systemverwalter mit den Mitteln der Subsystemverwaltung (siehe Handbuch „[Verwaltung von Subsystemen in BS2000](#)“) in einen bestehenden oder in einen zusätzlichen Subsystemkatalog einbringen.

Wenn UDS-SQL bzw. UDS-D mit IMON installiert wurden, ist die Benutzerkennung, unter der die Subsysteme installiert wurden, dem System bekannt.

Wenn UDS-SQL bzw. UDS-D nicht mit IMON installiert wurden, werden die Subsysteme unter der Standardbenutzerkennung (in der Regel \$TSOS) gesucht. Sollten sie unter einer anderen Benutzerkennung installiert sein, muss der BS2000-Systemverwalter die Bibliothek SYSLNK.UDS-SQL.029 bzw. SYSLNK.UDS-D.029 in die Standardbenutzerkennung kopieren oder den Parameter INSTALLATION-USERID (siehe unten) auf die Installationskennung ändern.

Der BS2000-Systemverwalter muss die Subsystemkonfiguration um das Subsystem erweitern (DSSM-Kommando ADD-SUBSYSTEM). Ab diesem Zeitpunkt ist eine Zuweisung über das BS2000-Kommando SELECT-PRODUCT-VERSION auf dieses Subsystem möglich.

Mit dem BS2000-Kommando MODIFY-SUBSYSTEM-PARAMETER kann der BS2000-Systemverwalter die Eigenschaften des Subsystems verändern. Es sind allerdings nur Änderungen an folgenden Operanden sinnvoll:

INSTALLATION-USERID

Wenn UDS-SQL mit IMON installiert wurde, wird die Installationskennung genutzt.

Wenn UDS-SQL nicht mit IMON installiert wurde, wird die Standardbenutzerkennung (in der Regel \$TSOS) genutzt. In diesem Fall müssen die LLM-Module des Subsystems in der Bibliothek, aus der das Subsystem geladen werden soll (Parameter LIBRARY), in der Standardbenutzerkennung verfügbar sein.

Der BS2000-Systemverwalter kann mit dem Parameter INSTALLATION-USERID aber auch eine andere Benutzerkennung einstellen, aus der das Subsystem geladen werden soll.

LIBRARY

Standardwert: SYSLNK.UDS-SQL.*nnn* (*nnn*=Version)

Bezeichnet die Bibliothek, aus der das Subsystem geladen werden soll.

MEMORY-CLASS

Standardwert: *SYSTEM-GLOBAL

Dadurch wird das Subsystem in den Klasse-3-/Klasse-4-Speicher geladen. Der Einsatz von Klasse-5-Speicher (LOCAL-PRIVILEGED) lohnt sich nur in speziellen Ausnahmesituationen, siehe Handbuch „[Verwaltung von Subsystemen in BS2000](#)“, Abschnitt Speicher- und Taskkonzept.

SUBSYSTEM-ACCESS

Standardwert: *HIGH

Damit wird das Subsystem oberhalb 16 Mbyte geladen und kann von allen XS-fähigen Anwendungen und den UDS/SQL-Tasks genutzt werden.

Wenn nicht-XS-fähige Anwendungen das Subsystem nutzen sollen, kann der BS2000-Systemverwalter das Subsystem auch unterhalb 16 Mbyte laden (SUBSYSTEM-ACCESS=*LOW, siehe aber einschränkende Hinweise im Handbuch „[Verwaltung von Subsystemen in BS2000](#)“, Abschnitt Speicher- und Taskkonzept).

i In Anwenderprogrammen wird zunächst in einigen Fällen versucht, die UDS/SQL-Verbindungsmodule unterhalb 16 Mbyte zu laden. Sofern das Subsystem oberhalb geladen ist (Parameter SUBSYSTEM-ACCESS=*HIGH), kann dies zu folgender Fehlermeldung je nachgeladenem Verbindungsmodul führen:

```
BLS0061 AMODE VON SYMBOL '....' UNGUELTIG IN DIESER UMGEBUNG.  
LADEVORGANG ABGEBROCHEN.
```

UDS/SQL lädt den UDS/SQL-Verbindungsmodul erneut mit modifizierten Parametern. Sofern keine weiteren Fehler gemeldet werden, können diese Meldungen des Binder-Lader-Systems ignoriert werden.

Das Subsystem wird vom BS2000-Systemverwalter mit dem DSSM-Kommando START-SUBSYSTEM aktiviert. Wenn mehrere Subsystemversionen parallel aktiv sein sollen, ist im START-SUBSYSTEM-Kommando folgende Angabe erforderlich:

```
VERSION-PARALLELISM=*COEXISTENCE-MODE
```

8.3 Prozessornutzung mit independent DBH optimieren

Eine wesentliche Zielsetzung beim Einsatz von UDS/SQL ist, stets die vorhandenen Prozessoren effizient auszunutzen. Zum einen profitiert der UDS/SQL-Nutzer direkt durch kürzere Antwortzeiten und erhöhten Durchsatz, zum andern profitieren alle anderen Anwendungen und Programme auf der Anlage, da mit dem Betriebsmittel CPU ökonomisch umgegangen wird.

Die UDS/SQL Servertasks des independent DBH übernehmen die Aufträge der Anwendertasks ohne jede feste Kopplung. Es gibt einen gemeinsamen Auftragspool, aus dem alle Server sich gleichermaßen Aufträge entnehmen - und wieder zurückstellen, wenn der Auftrag nur zum Teil bearbeitet werden konnte und nun z.B. auf das Ende eines I/O warten muss. Die Weiterverarbeitung kann von jedem Server übernommen werden. Daher ist keinerlei Laststeuerung und -verteilung durch den Datenbankadministrator notwendig.

Jede Servertask läuft im „Multithreading“-Verfahren, d.h. sobald ein Auftrag nicht mehr weiterbearbeitet werden kann, unterbricht die Servertask die Bearbeitung, stellt diesen Auftrag zurück und bearbeitet einen anderen Auftrag. Durch diese Arbeitsweise wird eine hohe Prozessorlokalität der Servertasks und dadurch ein sehr gutes Leistungsverhalten erreicht.

8.3.1 Ladeparameter, die die Prozessornutzung beeinflussen

Mit den folgenden DBH Ladeparametern können Sie die Prozessornutzung beeinflussen:

- PP DEACT
- PP RESULT-DELAY
- PP SCHEDULING
- PP SERVERTASK
- PP WAIT

Ladeparameter PP DEACT

Mit dem Ladeparameter DEACT können Sie steuern, wie das Betriebssystem die Servertasks in Situationen hoher Last behandeln soll.

Bei der Einstellung DEACT=NO werden die Servertasks auch bei hoher Auslastung der Anlage, hoher verbrauchter Systemleistung und in Wartezuständen weniger von anderen Tasks im System beeinflusst und laufen dadurch performanter. Allerdings werden dadurch andere Anwendungen eventuell benachteiligt.

Ladeparameter PP RESULT-DELAY

Mit dem Ladeparameter PP RESULT-DELAY können Sie die Auftragsantworten an die Anwendertasks bündeln. Dadurch kann in Hochlastsituationen der Durchsatz verbessert werden.

In Verbindung mit der Einstellung PP SCHEDULING=ASYMMETRIC wird die tatsächlich durchgeführte Bündelung der Auftragsantworten jedoch eingeschränkt.

Ladeparameter PP SCHEDULING

Mit dem Ladeparameter PP SCHEDULING können Sie das interne Verhalten mehrerer Servertasks bei Lastschwankungen im Hochlastbetrieb beeinflussen.

Die Einstellung PP SCHEDULING=SYMMETRIC bewirkt, dass alle Servertasks gleichberechtigt Aufträge bearbeiten. Sie ist günstig bei gut ausgelasteten Servertasks oder wenn den Servertasks keine hohe BS2000-Priorität gegeben werden kann.

Diese Einstellung ist Standardwert, da die gleichberechtigte Abarbeitung der Aufträge durch die Servertasks bei dynamisch sich ändernden Lasten im Mittel das beste Leistungsverhalten erzeugt.

Durch die Einstellung PP SCHEDULING=ASYMMETRIC wird eine ungleiche Auslastung der Servertasks erzielt. Dadurch wird eine bessere Ausnutzung des Prozessor-Cache und ein besseres Zusammenspiel mit dem Task-Scheduling des Betriebssystems in bestimmten Lastsituationen erreicht, insbesondere dann, wenn die Servertasks nicht voll ausgelastet werden können, eine Servertask weniger aber zu wenig wäre.

Ladeparameter PP SERVERTASK

Für eine effiziente Nutzung der Prozessoren ist die richtige Einstellung des Ladeparameters PP SERVERTASK wesentlich. Um eine optimale Wirkung der Einstellung zu erzielen, müssen Sie die BS2000-Prioritäten angemessen berücksichtigen (siehe [Abschnitt „Leistungsbezogene BS2000-Einstellungen“](#)).

Auf Grund des Multithreading-Verfahrens reicht im Allgemeinen auf einem Monoprozessor bei Nutzung des asynchronen I/O (PP IO=ASYN) eine einzige Servertask aus (PP SERVERTASK=1). Erst auf Multiprozessoranlagen lohnt sich der Einsatz weiterer Servertasks. Die Obergrenze für den Ladeparameter PP SERVERTASK ist im Normalfall (PP IO=ASYN) die Anzahl der Prozessoren.

Für besondere Einsatzbedingungen wird allerdings auch ein höherer Wert akzeptiert, z.B. wenn der Ladeparameter PP IO=SYNC angegeben ist.

Die sinnvolle Einstellung des Ladeparameters PP SERVERTASK können Sie im Normalfall (PP IO=ASYNCR) bei hoher Anwenderlast durch eine einfache Messung ermitteln. In der Messkonfiguration sollte die Einstellung des Ladeparameters PP SERVERTASK an der möglichen Obergrenze orientiert werden. In der Messung wird der CPU-Verbrauch der Servertasks, die am Programmnamen UDSSUB erkennbar sind, ermittelt. Die CPU-Verbrauchsmessung können Sie sehr einfach mit dem BS2000-Kommando SHOW-USER-STATUS zu Beginn und am Ende von Messphasen vornehmen.

Wenn Sie den CPU-Verbrauch der Tasks mit dem Programm UDSSUB (erkennbar bei SHOW-USER-STATUS INFORMATION=PROGRAM) addieren und diesen Wert in Relation zum CPU-Verbrauch aller im System aktiven Prozesse setzen, so erhalten Sie eine gute Näherung an die anteilige CPU-Entnahme der Servertasks in der jeweiligen Messphase.

Um den „richtigen“ Ladeparameter PP SERVERTASK zu ermitteln, multiplizieren Sie den Maximalwert über alle Messphasen mit der Anzahl Prozessoren der Anlage.

Wenn z.B. die Servertasks zusammen 41% CPU auf einem Quadroprozessor entnehmen, sind 1,64 (=0,41 x 4) Servertasks notwendig. PP SERVERTASK=2 ist dann die richtige Einstellung.

Wenn eine Konfiguration starke Lastschwankungen aufweist, kann die Einstellung PP SERVERTASK je nach Interessenlage der Spitzenlast (d.h. beste Antwortzeiten) oder einer Durchschnittslast (d.h. optimale CPU-Nutzung im Normalbetrieb) angepasst werden.

Eine zu hohe Einstellung des Ladeparameters PP SERVERTASK erzeugt deutlich mehr Taskkommunikation, d.h. höheren CPU-Verbrauch und behindert möglicherweise andere Anwendungen.

Eine zu niedrige Einstellung reduziert im Allgemeinen die Durchsatzrate in der UDS/SQL-Konfiguration.

Ladeparameter PP WAIT

Mit dem Ladeparameter PP WAIT können Sie eine aktive Wartestrategie einstellen.

Normalerweise wird in Situationen niedriger Last auf neu eintreffende Aufträge über die Ereignissteuerung gewartet. Damit wird sichergestellt, dass die CPU durch andere Anwendungen genutzt werden kann, wenn von den Servertasks keine Aufträge mehr bearbeitet werden können.

Wenn durch die Anwendung sichergestellt werden kann, dass solche Unterlastsituationen sehr selten und sehr kurz sind, kann durch aktives Warten der CPU-Aufwand für die Ereignissteuerung im Normal- und Hochlastfall reduziert werden.

i Die Ladeparameter PP RESULT-DELAY und PP WAIT sind für Hochlastkonfigurationen konzipiert und sollten nur genutzt werden, wenn die bei der Beschreibung der Ladeparameter (siehe "[Auftragsantworten an die Anwendertasks bündeln \(RESULT-DELAY\)](#)" und "[Warte-Modus der Servertasks einstellen \(WAIT\)](#)") genannten Voraussetzungen auch erfüllt sind. Ansonsten können erhebliche Beeinträchtigungen des Betriebs eintreten.

8.3.2 Leistungsbezogene BS2000-Einstellungen

Für eine effiziente CPU-Nutzung einer UDS/SQL-Konfiguration sind geeignete BS2000-Einstellungen wesentlich. Im Folgenden werden solche Einstellungen beschrieben.

Taskprioritäten

BS2000 bietet die Möglichkeit über Prioritäten das Verhalten des Systems in Voll- und Überlastsituationen zu beeinflussen, siehe Handbuch „[Einführung in die Systembetreuung](#)“.

Um ein über einen längeren Zeitraum garantiertes Verhalten der Servertasks zu erhalten, wird es empfohlen, feste Prioritäten für Hochlastanforderungen zu nutzen. Hierdurch wird eine dauerhafte Priorisierung der Servertasks erreicht, was allerdings mit deutlichen Benachteiligungen anderer Tasks der gleichen Kategorie verbunden sein kann.

Einige Einstellungen von Ladeparametern sind nur in Verbindung mit festen Prioritäten sinnvoll.

Im [Abschnitt „Beispiel Hauptlastkonfiguration“](#) wird die Wirkung der Einstellungen erläutert.

Task Attribut TP

UDS/SQL schaltet jede Servertask auf das BS2000-Task Attribut TP um. Dadurch wird eine kontinuierliche Nutzung der Prozessoren durch die Servertasks ermöglicht. Allerdings muss die Jobklasse, die die Servertasks nutzen, der Taskkategorie TP zugeordnet sein (siehe Jobklassenzuteilung im "[DBH-Startkommandos](#)").

Mit dem BS2000-Kommando SHOW-JOB-CLASS können Sie kontrollieren, ob die Taskkategorie TP für die Jobklasse zugewiesen ist. Die Jobklassenzuteilung können Sie mit dem BS2000-Kommando SHOW-USER-STATUS,INFORMATION=*JOB prüfen.

BS2000-Subsystem TANGRAM

Der BS2000-Anwender kann das leistungssteigernde Subsystem TANGRAM (Task and Group Affinity Management) für Multiprozessoren einsetzen, siehe Handbuch „[Einführung in die Systembetreuung](#)“. Der Datenbankadministrator muss keine spezielle UDS/SQL-Einstellung vornehmen, UDS/SQL meldet von sich aus alle Servertasks zu einer Taskgruppe bei TANGRAM an.

TANGRAM wird insbesondere Leistungsverbesserungen für solche UDS/SQL-Konfigurationen ermöglichen, für die entweder wegen stark schwankender Lasten der Ladeparameter PP SERVERTASK mittel bzw. wegen Ladeparameter PP IO=SYNC groß eingestellt werden muss oder die wegen anderer wichtiger Anwendungen nicht priorisiert werden können.

Dagegen ist bei UDS/SQL-Konfigurationen, die mit sonstigen Mitteln (festen BS2000-Prioritäten, UDS/SQL Ladeparametern) optimal eingestellt sind, keine wesentliche Verbesserung zu erwarten. Der Einsatz von TANGRAM kann aber auch in diesen Fällen zur Optimierung von anderen Mehrtasksystemen (z.B. openUTM) ohne schädliche Nebenwirkung auf die UDS/SQL-Konfiguration erfolgen.

8.3.3 Optimierte Taskkommunikation

Die ereignisgesteuerte Taskkommunikation für den Auftragstransport zwischen Anwender- und Servertask wird intern durch UDS/SQL optimiert. Der Datenbankadministrator muss in der Regel keine direkte Maßnahmen ergreifen, um diese Kommunikation zu steuern.

Die Einstellungen im BS2000 und die Ladeparameter beeinflussen die Kommunikation teilweise aber indirekt. Um diese Wirkung kontrollieren zu können, wird im Folgenden die Kommunikation beschrieben und es wird erläutert, welche Monitorzähler Sie zur Bewertung heranziehen können.

Beschreibung der Taskkommunikation

Die Auftragsübergabe geschieht über einen Common Memory Pool, den Communication Pool (CUP), der vom UDS /SQL entsprechend den Angaben beim Ladeparameter PP CUPSIZE angelegt wird.

Die Taskkommunikation dient dem Warten und Wecken der Servertasks und Anwendertasks und ist über die Ereignissteuerung des BS2000 realisiert.

Das folgende Bild stellt die Taskkommunikation pro Auftrag dar.

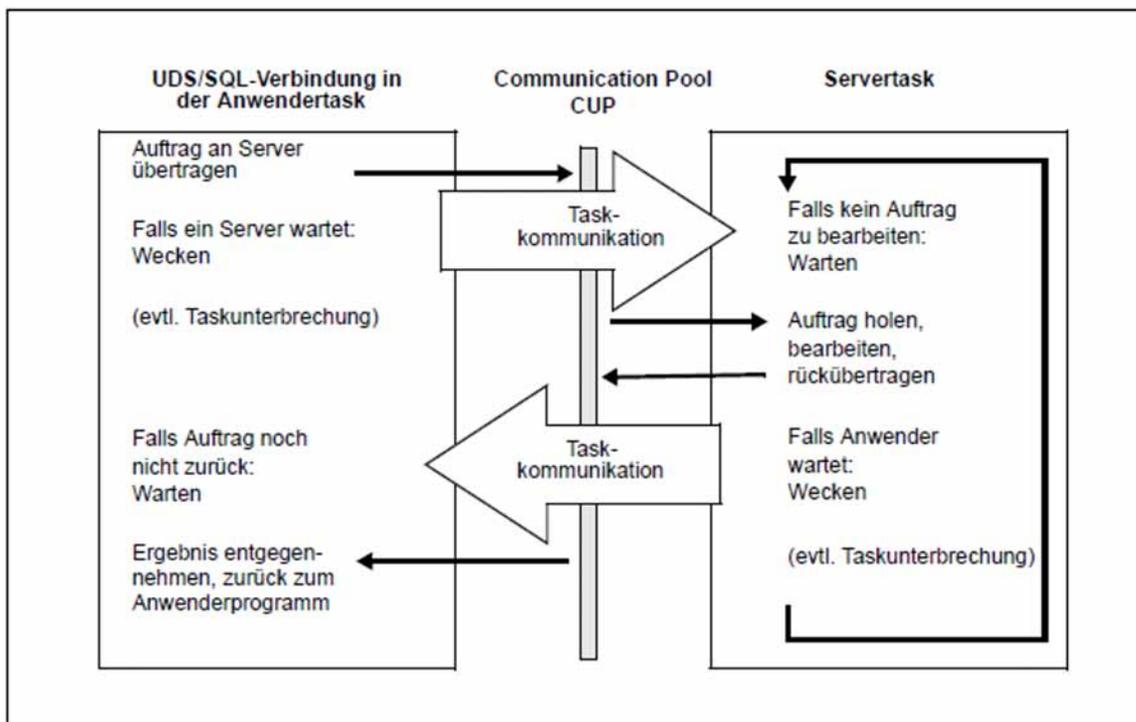


Bild 9: Taskkommunikation pro Auftrag

Ziel der Optimierung der Taskkommunikation ist das Vermeiden von unnötigen Warte- und Weckaufträgen an BS2000 und damit weniger CPU-Verbrauch. Das Wecken einer Servertask zur Bearbeitung einer DML erfolgt nur, wenn diese nicht bereits aktiviert ist („Hinweg-Optimierung“). Beim Zurücksenden des Ergebnisses an das Anwenderprogramm wird dieses nur geweckt, wenn es bereits einen Warteauftrag an das BS2000 übergeben hat („Rückweg-Optimierung“). Dabei werden Eigenschaften des BS2000-Schedulings ausgenutzt.

8.3.4 Interpretation der Kommunikationszähler des UDS/SQL-Monitors

Voraussetzung für eine erfolgreiche Hinweg-Optimierung der Kommunikation ist eine stetige Auftragslast für die Servertasks, d.h. es sollten immer so viele Aufträge für die Servertasks im Auftragspool vorhanden sein, dass diese ständig beschäftigt sind. In diesem Falle kann auf das Wecken der Servertasks verzichtet werden.

Andererseits ist es auch möglich, den Rückweg zum Anwender zu optimieren. Wenn auf Grund des BS2000-Schedulings mit der Übergabe des Auftrags an die Servertask die Anwendertask unterbrochen wurde und diese damit noch nicht wartet, kann das Wecken der Anwendertask eingespart werden.

In welchem Umfang die Optimierungen wirken, können Sie mit folgenden Monitorzählern in der Maske COUNTER beobachten:

- Der Zähler „DML CALLS“ zählt die insgesamt an die Servertasks übergebenen Aufträge.
- Der Zähler „ITC US -> ST“ enthält die Anzahl der ausgeführten Wecksignale von der UDS/SQL-Verbindung zu den Servertasks.
- Der Zähler „ITC ST -> US“ enthält die Anzahl der ausgeführten Wecksignale von den Servertasks zu der UDS/SQL-Verbindung.

Die vollständigen Zählerbeschreibungen können Sie dem Abschnitt „Beschreibung der UDS/SQL-Monitormaske: COUNTER“ im [Abschnitt „Beschreibung der UDS/SQL-Monitor-Ausgabe auf Datensichtstation“](#) entnehmen.

Wenn jeder der beiden Zähler „ITC US -> ST“ und „ITC ST -> US“ fast gleich dem Zähler „DML CALLS“ ist, konnte keine wesentliche Optimierung der Taskkommunikation zustandekommen. Mögliche Ursachen können sein:

- zu geringe Last
- zu hohe Zahl von Servertasks (Ladeparameter PP SERVERTASK)
- falsch eingestellte BS2000-Prioritäten

Wenn der Zähler „ITC US -> ST“ fast gleich dem Auftragszähler „DML CALLS“ der gleichen Monitormaske ist, bedeutet dies, dass die Servertasks häufig gewartet haben, sonst hätten sie nicht geweckt werden müssen. Dies ist ein Indiz dafür, dass sich auf dem Prozessor oft eine Anwender- und eine Servertask abwechselten, wobei jeweils nur ein Auftrag bearbeitet wurde. Die Hinweg-Optimierung hat nicht gewirkt.

Wenn der Zähler „ITC ST -> US“ fast gleich dem Auftragszähler „DML CALLS“ der gleichen Monitormaske ist, bedeutet dies, dass die Anwendertasks sehr häufig schon im Wartezustand waren, wenn der DML-Auftrag in den Servertasks fertig bearbeitet war. Die Rückweg-Optimierung hat nicht gewirkt.

Beide Zähler zusammengezählt können aber nie unter die Anzahl „DML CALLS“ sinken, da pro Auftrag mindestens ein Taskkommunikationspartner immer warten muss.

Bei der Interpretation der Zähler sollten Sie beachten, dass der Durchsatz an Transaktionen das entscheidende Kriterium für die Optimierung darstellt. In den Kommunikationszählern können nicht alle Effekte, die sich im BS2000-Scheduling ergeben, erfasst werden; manchmal geht eine Verbesserung des Durchsatzes mit einer geringfügigen Verschlechterung der Kommunikationsoptimierung einher.

8.3.5 Beispiel Hauptlastkonfiguration

In dem folgenden Beispiel soll die Wirkung der Parameter erläutert werden. Es wird davon ausgegangen, dass auf einem Rechner eine UDS/SQL-Konfiguration als Hauptlast betrieben wird, für die bestimmte Systemleistungen garantiert werden sollen.

Das ideale Systemverhalten aus Sicht der Taskkommunikation ist, wenn sich einige Aufträge ansammeln, die dann die Servertasks unterbrechungsfrei abarbeiten. Parallel zur Abarbeitung können auch weitere Aufträge von anderen Prozessoren eintreffen, wenn weniger Servertasks vorhanden sind als Prozessoren.

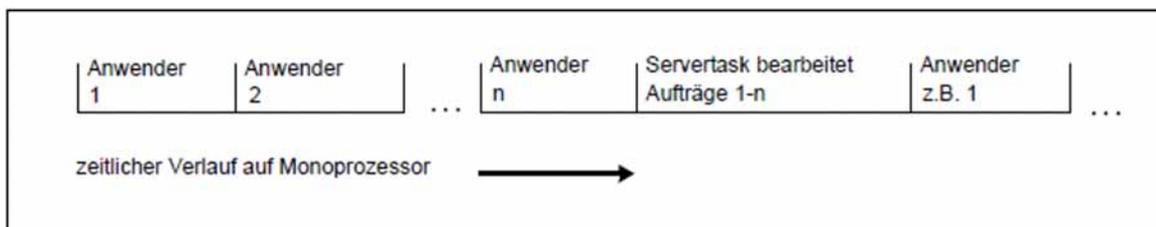
Durch die richtige Einstellung der BS2000-Priorität kann dieser Effekt erreicht werden. Dabei wird ein BS2000-Schedulingprinzip ausgenutzt. Es wird nicht in jedem Fall eine höher priorisierte Task, die auf eine Prozessorzuteilung wartet, eine laufende, schwächer priorisierte verdrängen. Nur wenn der Prioritätsunterschied groß ist, wird dies geschehen.

Wenn die Anwendertasks eine geringfügig höhere Priorität als die Servertasks haben, werden zuerst viele Anwendertasks ihre Aufträge in den Auftragspool stellen, obwohl eine Servertask von der ersten Anwendertask bereits geweckt wurde. Die Servertask bekommt erst den Prozessor zugeteilt, wenn alle Anwendertasks im Warten auf die Auftragsbearbeitung sind.

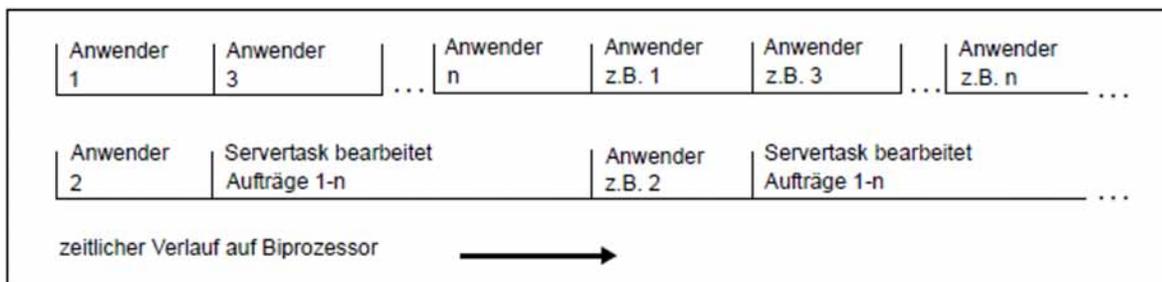
Nun kommt die Servertask auf den Prozessor und arbeitet nacheinander alle anstehenden Aufträge ab, obwohl bereits nach dem ersten Auftrag die erste Anwendertask geweckt wird.

Dieser Effekt stellt sich aber nur bei Benutzung von festen Prioritäten sowohl für Server- als auch für Anwendertasks (im Bereich 127 bis 60) dauerhaft ein. Bei Verwendung von variablen Prioritäten (im Bereich 255 bis 128) werden die internen Prioritätswerte dynamisch vom Betriebssystem verändert, sodass das Echtzeitverhalten des Systems nicht mehr entsprechend vorhergesagt werden kann.

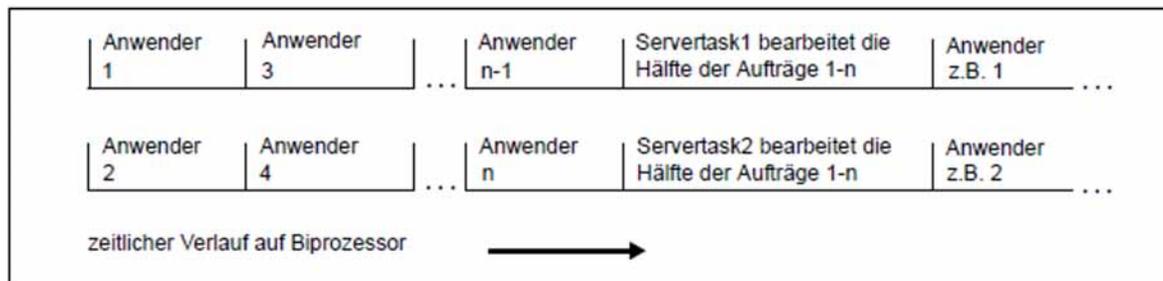
Bei Einstellung der BS2000-Priorität der Mastertask und dadurch der Servertasks von 120 (siehe RUN-PRIO bei ENTER-Jobs auf "DBH-Startkommandos") und der Anwendertasks von 119 ergibt sich auf einem Monoprozessor bei Ladeparametereinstellung PP SERVERTASK=1 folgende günstige Situation:



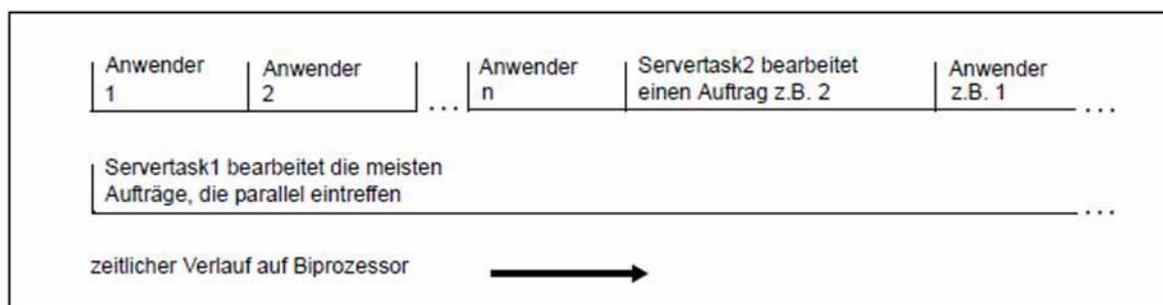
Auf einem Biprozessor mit der Ladeparametereinstellung PP SERVERTASK=1 ergibt sich folgende günstige Situation, wenn der CPU-Anteil der Anwenderprogramme größer ist als der der Servertasks.



Auf einem Biprozessor mit der Ladeparametereinstellung PP SERVERTASK=2 ergibt sich folgende günstige Situation bei n Anwendern:



Damit auf dem Biprozessor mit zwei Servertasks eine ähnliche Situation entsteht wie bei nur einer Servertask, wenn der CPU-Anteil der Servertasks größer ist als der der Anwendertasks, sollte der Ladeparameter PP SCHEDULING=ASYMMETRIC eingestellt werden. In diesem Fall versucht eine Servertask kontinuierlich Aufträge zu bearbeiten; die zweite Servertask bearbeitet jedoch immer nur einen Auftrag.



Auf einem Quadroprozessor ergibt sich ein entsprechendes Bild wie auf dem Biprozessor mit mehr als einer Servertask.

Die Wirkung des Ladeparameters PP SCHEDULING=ASYMMETRIC ist damit die folgende: Dadurch, dass eine Servertask immer nur einen Auftrag ausführt, laufen alle anderen Servertasks auf den parallelen Prozessoren entsprechend länger.

Weitere Tasks mit niedrigerer BS2000-Priorität auf dem Rechner füllen die Prozessorzeit, wenn ein typischer, kurzzeitiger Lasteinbruch auftritt oder die Datenbankkonfiguration nur einen Teil der CPU-Zeit benötigt.

Weitere Tasks mit höherer BS2000-Priorität, wie z.B. Systemtasks, bringen die oben beschriebenen Abläufe nicht wesentlich aus dem Gleichgewicht, vorausgesetzt solche Tasks beanspruchen nicht einen Prozessor dauerhaft.

9 Nutzung der BS2000-Funktionalität

In diesem Kapitel wird beschrieben, wie die BS2000-Funktionalität beim Einsatz von UDS/SQL genutzt werden kann.

9.1 Pubsets für UDS/SQL nutzen

Die Pubset-Verwaltung des BS2000 (siehe Handbuch „[Einführung in die Systembetreuung](#)“) bietet Ihnen die Möglichkeit, den Ort der Speicherung von Dateien auf gemeinschaftlichen Platten zu bestimmen. Dabei können sowohl Single-Feature-Pubsets (SF-Pubsets) als auch System-Managed-Pubsets (SM-Pubsets) genutzt werden.

Folgende **Vorteile** kann der Datenbankadministrator nutzen, wenn er die Möglichkeiten der Pubset-Verwaltung des BS2000 für UDS/SQL ausschöpft:

- Er kann mit dem CREATE-FILE-Kommando UDS/SQL-Dateien einrichten und sie gezielt auf verschiedenen Public Volume Sets platzieren
- Er kann unter Nutzung von physikalischer Allokierung Dateien gezielt auf Volume Sets bzw. auf einzelne Volumes platzieren. Dazu muss der BS2000-Systemverwalter den SF-Pubset mit dem entsprechenden Attribut zur Verfügung stellen bzw. der Benutzerkennung das Recht auf physikalische Allokierung pubset-spezifisch einräumen. Details hierzu siehe Handbücher „[Einführung in die Systembetreuung](#)“ und „[Einführung in das DVS](#)“.
- Er kann durch exklusive Nutzung von Public Volume Sets, Volume Sets bzw. Volumes für Datenbankdateien sicherstellen, daß der bei dynamischer Erweiterung bzw. Online-Erweiterung benötigte Plattenplatz auch tatsächlich zur Verfügung steht.
- Er kann möglichen Schaden eingrenzen, wenn er UDS/SQL-Dateien auf verschiedene Public Volume Sets legt, sodass bei Ausfall eines Public Volume Set Dateien auf anderen Public Volume Sets noch benutzbar sind
- Er kann durch Ablage von Datenbankdateien und RLOG-Dateien in einem Public Volume Set moderne Hardwaretechnologie (Clones, Snaps) nutzen, um jederzeit sehr performant Datenbankkopien im laufenden Betrieb ziehen zu können. Aus Sicherheitsgründen sollte in diesen Fällen aber unter allen Umständen die RLOG-Datei auf einem anderen Volume Set doppelt geführt werden.
- Er kann durch Einschränkung der UDS/SQL-Pubset-Umgebung (siehe [Abschnitt „Pubset-Deklarations-Jobvariable“](#)) verhindern, dass die Mehrdeutigkeit von Dateinamen störende Auswirkungen auf den UDS/SQL-Betrieb hat.
- Er kann entsprechend den Vorschlägen zur Erhöhung der Sicherheit die Originaldateien und Sicherungskopien (.SAVE) auf verschiedene Public Volume Sets legen.
- Er kann Performance-Verbesserungen erzielen durch sinnvolle Verteilung der UDS/SQL-Dateien auf verschiedene Public Volume Sets.

Folgendes muss der Datenbankadministrator beachten:

UDS/SQL behandelt bereits bestehende Datenbankdateien und Systemdateien anders als neu einzurichtende Dateien.

Datenbankdateien sind: Realms, ALOG-Datei, HASHLIB, COSSD

Systemdateien sind: SLF, Temporäre Realms, RLOG-Dateien, DB-Status-Datei.

Bestehende Datenbank- und Systemdateien

Für solche Dateien fordert UDS/SQL, dass es im lokalen Rechner innerhalb ihrer Benutzerkennung über alle relevanten Public Volume Sets keine zweite Datei gleichen Namens gibt. Die für den Betrieb von UDS/SQL relevanten Public Volume Sets können durch die Pubset-Deklaration (siehe [Abschnitt „Pubset-Deklarations-Jobvariable“](#)) für einen Programmlauf spezifiziert werden. Wenn keine Spezifikation angegeben ist, gilt die Eindeutigkeitsforderung für alle lokal verfügbaren Public Volume Sets.

Auf Grund dieser Eindeutigkeitsregel ist UDS/SQL in der Lage, die Katalogkennung *:catid.* jeder bestehenden Datenbank- und Systemdatei selbst zu ermitteln und dann intern zu benutzen. Deshalb ist für solche Dateien eine

explizite Angabe der Katalogkennung an der Benutzerschnittstelle nicht erforderlich und wird von UDS/SQL teilweise sogar verboten, um Widersprüche zwischen angegebener und tatsächlicher Katalogkennung einer Datei zu vermeiden.

Auf diese Weise kann der Datenbankadministrator die Lage der wichtigsten UDS/SQL-Dateien flexibel bestimmen, ohne dass er UDS/SQL die Lage der einzelnen Dateien explizit mitteilen muss.

Bestehende sonstige Dateien

Für alle übrigen Dateien z.B. Ladeparameter-Dateien, DISTABLE-Datei, UDS/SQL-ENTER-Dateien, usw. kann im Allgemeinen die Katalogkennung zusammen mit der Benutzerkennung bei der Zuweisung angegeben werden. Hilfsdateien der Dienstprogramme (SCRTCH- und SORTWK-Dateien) werden aber gemäß Pubset-Deklaration auf Eindeutigkeit geprüft; insofern kann auch eine angegebene Katalogkennung ignoriert werden. Bereits eingerichtete Dateien, für die der Datenbankadministrator keine spezifische Zuweisung vorgenommen hat, müssen ansonsten in der **Standard-Katalogkennung** liegen (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-USER-Kommando).

Grundsätzlich ist es in folgenden Fällen immer erlaubt, die Katalogkennung anzugeben:

- Beim Einrichten von Dateien mit dem CREATE-FILE-Kommando,
- beim Zuweisen von Dateien für UDS/SQL mit dem ADD-FILE-LINK-Kommando und LINK-NAME und
- bei der Bekanntgabe des Konfigurationsnamens mit dem SET-FILE-LINK-Kommando und dem Operanden LINK-NAME.

Durch UDS/SQL einzurichtende Dateien

Nicht vorhandene Dateien werden, sofern sie nicht explizit vom Anwender eingerichtet werden müssen, bei Bedarf durch UDS/SQL in der **Standard-Katalogkennung** der Datenbank- bzw. Konfigurationskennung angelegt (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, ADD-USER-Kommando).

Ausnahme

RLOG-Datei:

Katalogkennung wird mit PP LOG, PP LOG-2, PP RESERVE, MODIFY LOG, MODIFY-LOG-2 oder MODIFY RESERVE festgelegt.

ALOG-Datei:

Katalogkennung wird mit der BMEND-Anweisung START-LOG festgelegt.

Wenn die Standard-Katalogkennung im Benutzerkatalog geändert werden soll, ist zu beachten, dass alle Dateien, die nicht in den verfügbaren Pubsets gesucht werden („sonstige Dateien“), aus der alten in die neue Standard-Katalogkennung transportiert werden müssen, damit sie für UDS/SQL auffindbar sind (siehe die Kommando-Handbücher zu „[BS2000 OSD/BC](#)“, MODIFY-USER-ATTRIBUTES-Kommando).

UDS/SQL spricht auch bei Einsatz von Mehrfachrechnersystemen (MSCF) und Remote File Access (RFA) nur lokal verfügbare Public Volume Sets an.

UDS/SQL eröffnet jede Datenbank- und Systemdatei erst nach Ausführung des FSTAT-Makros (Kommando SHOW-FILE-ATTRIBUTES) über alle Public Volume Sets, die in der aktuell gültigen UDS/SQL-Pubset-Deklaration zugewiesen sind. Dabei wird nicht gewartet, bis Public Volume Sets, die sich im Zustand QUIET befinden, diesen Zustand wieder verlassen; insofern werden Dateien auf diesen Public Volume Sets in der Folgeverarbeitung nicht berücksichtigt, auch wenn sie in der UDS/SQL-Pubset-Deklaration enthalten sind.

Je mehr Public Volume Sets mit Datenbank- und Systemdateien belegt sind, die zu einer Datenbank gehören, umso größer ist der interne Verwaltungsaufwand. Um Performance-Einbußen zu vermeiden, sollten Sie daher nicht mehr Public Volume Sets in die Pubset-Deklaration einbeziehen als nötig.

Bei Anweisungen der Dienstprogramme sind Einschränkungen bei der Zuweisung von Dateien bzgl. der Katalogkennung möglich. Diese sind bei den Schnittstellen der einzelnen Dienstprogramme im jeweiligen Handbuch beschrieben.

Die folgende Tabelle zeigt, an welchen Schnittstellen zwischen Anwender und DBH die Angabe einer Katalogkennung *:catid:* erlaubt bzw. verboten ist:

Benutzerschnittstelle	Funktion
<code>/SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME=[:catid:][\$admuserid.]konfigurationsname</code>	Konfigurationsnamen festlegen
<code>/ADD-FILE-LINK LINK-NAME=DATABASE ¹ ,FILE-NAME=[:catid:][\$dbuserid.]dbname[.DBDIR]</code>	Datenbank für Dienstprogrammmlauf festlegen
<code>PP DBNAME=[\$dbuserid.]dbname[.copyname], ... ADD DB=[\$dbuserid.]dbname[.copyname], ...</code>	Namen und ggf. die Kennung einer Datenbank angeben
<code>PP DISTABLE=[:catid:][\$userid.]dateiname ADD DIS,FILE=[:catid:][\$userid.]dateiname SAVE DIS,FILE=[:catid:][\$userid.]dateiname</code>	Namen einer DISTABLE-Datei angeben
<code>/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME=[:catid:][\$userid.]dateiname /ASSIGN-SYSDTA TO=[:catid:][\$userid.]dateiname</code>	Ladeparameter-Datei zuweisen

Tabelle 20: Übersicht über die Angabemöglichkeiten der Katalogkennung an UDS/SQL-Benutzerschnittstellen

¹ In diesem Fall ist die Angabe einer *:catid:* erlaubt, damit für das ADD-FILE-LINK-Kommando ein existierender DBDIR ausgenutzt werden kann.

Erklärung:

- userid* Benutzerkennung
- dbuserid* Datenbankkennung
- admuserid* Konfigurationskennung

9.2 Jobvariablen für UDS/SQL nutzen

UDS-SQL unterstützt folgende Typen von Jobvariablen:

Typ	Name	Bedeutung	Siehe
Pubset-Deklaration	LINK-NAME= UDSPS01	Festlegung der von DBH und den Dienstprogrammen nutzbaren Pubsets	"Pubset-Deklarations-Jobvariable"
Session	JV-NAME= UDSSQL.DBH. <i>session</i>	Informationen über den Zustand der DBH-Session zur automatisierten Steuerung von Anwendungen und Administration	"Session-Jobvariable"
Datenbank	JV-NAME= UDSSQL.DB. <i>datenbankname</i>	Informationen über den Zustand einer Datenbank zur automatisierten Steuerung von Anwendungen und Administration	"Datenbank-Jobvariable"
BMEND	LINK-NAME= JVBMEND	Ausgabedaten von BMEND zur automatisierten Steuerung von Sicherung und Recovery	"BMEND-Jobvariable" und Handbuch „Sichern, Informieren und Reorganisieren"

Tabelle 21: Von UDS/SQL genutzte Jobvariablen

Voraussetzung für das Arbeiten mit Jobvariablen ist, dass das Subsystem „Jobvariablen (JV)“ installiert ist. Wenn es nicht verfügbar ist, unterbleibt die Versorgung der Jobvariablen im Allgemeinen ohne Meldung.

9.2.1 Pubset-Deklarations-Jobvariable

DBH und Dienstprogramme prüfen in der Regel, ob die genutzten Dateien - insbesondere die Datenbankdateien - über alle angeschlossenen Pubsets eindeutig sind. Mit einer UDS/SQL-Pubset-Deklaration können Sie diese Prüfung auf ausgewählte Pubsets beschränken.

Durch die Einschränkung der Eindeigkeitsanforderung für Dateinamen, erhalten Sie eine größere Flexibilität bei der Organisation mehrerer UDS/SQL-Anwendungen auf einem Rechner. Rückwirkungen auf den Betrieb der UDS/SQL-Anwendungen durch Änderung der Speicherorganisation - etwa bei der aktiven Nutzung moderner Speichertechnologien auf Basis von Clones und Snaps - können vermieden werden.

Die UDS/SQL-Pubset-Deklaration stellen Sie optional in einer Pubset-Deklarations-Jobvariable bereit, auf die der DBH und die Dienstprogramme über den vorgegebenen Jobvariablen-Linknamen UDSPS01 zugreifen.

Falls beim Programmbeginn keine UDS/SQL-Pubset-Deklaration vorliegt, werden bestehende DB-Dateien, Systemdateien und Arbeitsdateien der Dienstprogramme in den Dateikatalogen jedes lokal verfügbaren Pubsets gesucht, bei dem die Benutzerkennung des ablaufenden Programmes als zugriffsberechtigt eingetragen ist. Dieses Standardverhalten entspricht einer UDS/SQL-Pubset-Deklaration "*". Auch bei Nichtverfügbarkeit des Produktes "Jobvariablen" wird als UDS/SQL-Pubset-Deklaration "*" angenommen.

i Wenn während der Initialisierung des DBH oder beim Start eines Dienstprogramms eine fehlerhafte UDS/SQL-Pubset-Deklaration zugewiesen ist oder die Zuweisung der Jobvariable fehlerhaft ist, wird die Session bzw. das Dienstprogramm beendet.

Syntax der UDS/SQL-Pubset-Deklaration

Die UDS/SQL-Pubset-Deklaration in der Pubset-Deklarations-Jobvariablen besteht aus einer Folge von sogenannten Catid-Gruppen, die jeweils eine FSTAT-konforme Katalogkennungsangabe darstellen und voneinander durch ein oder mehrere Leerzeichen getrennt sind.

Die Syntax der UDS/SQL-Pubset-Deklaration ist somit wie folgt:

catid-group[catid-group]...

Angabe einer oder mehrerer Catid-Gruppen (1- bis 4-stellige Katalogkennungen ohne „:“), die die in folgender [Tabelle 22](#) aufgeführten Platzhalter (Wildcards) enthalten dürfen:

*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.
/	Ersetzt genau ein beliebiges Zeichen.
<s _x :s _y >	<p>Ersetzt eine Zeichenfolge, für die gilt:</p> <ul style="list-style-type: none"> • sie ist mindestens so lang wie die kürzeste Zeichenfolge (s_x oder s_y) • sie ist höchstens so lang wie die längste Zeichenfolge (s_x oder s_y) • sie liegt in der alphabetischen Sortierung zwischen s_x und s_y; Zahlen werden hinter Buchstaben sortiert (A...Z, 0...9) • s_x darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht • s_y darf auch die leere Zeichenfolge sein, die an dieser Stelle für die Zeichenfolge mit der höchst möglichen Codierung steht (enthält nur die Zeichen X'FF')
<s ₁ ,...>	Ersetzt alle Zeichenfolgen, auf die eine der mit s angegebenen Zeichenkombinationen zutrifft. s kann auch die leere Zeichenfolge sein. Jede Zeichenfolge s kann auch eine Bereichsangabe „s _x :s _y “ sein.
-s	Ersetzt alle Zeichenfolgen, die der angegebenen Zeichenfolge s nicht entsprechen. Das Minuszeichen darf nur am Beginn der Zeichenfolge stehen. Diese Angabe ist nicht mit anderen Angaben in einer UDS/SQL-Pubset-Deklaration kombinierbar.

Tabelle 22: Platzhalter (Wildcards) für Katalogkennungen in der UDS/SQL-Pubset-Deklaration

Maximal dürfen 100 Catid-Gruppen angegeben werden.

Eine Catid-Gruppe darf maximal 26 Zeichen lang sein.

Kleinbuchstaben werden wie die entsprechenden Großbuchstaben behandelt.

Die Angabe nicht existenter oder nicht verfügbarer Katalogkennungen ist erlaubt.

Katalogkennungen dürfen auch mehrfach angegeben werden.

Beispiele

Angabe	Steht für folgende Katalogkennungen
A001	A001
X/C	XAC, XBC, XCC, ..., XZC, X0C, ..., X9C (Alle 3-stelligen Katalogkennungen, die mit X beginnen und mit C enden)
5*	5, 5A, ..., 59, 5AA, ..., 599, 5AAA, 5999 (Alle 1- bis 4-stelligen Katalogkennungen die mit 5 beginnen)
<C015:C025>	C015, C016, ..., C024, C025 (Alle 4-stelligen Katalogkennungen im Bereich von C015 bis C025)
<BE:DC>	BE, BF, BG, ..., B9, CA, ..., C9, DA, ..., DC (Alle 2-stelligen Katalogkennungen im Bereich von BE bis DC)
<D015:D025,F015:F045>	D015, D016, ..., D024, D025, F015, F016, ..., F044, F045 (Liste von Bereichen: Alle 4-stelligen Katalogkennungen von D015 bis D025 und von F015 bis F045)
<A:D;BE:DC>	A, B, C, D, BE, BF, BG, ..., B9, CA, ..., C9, DA, ..., DC (Liste von Bereichen: Alle 1stelligen Katalogkennungen von A bis D und alle 2-stelligen von BE bis DC)
-5*	Alle (1- bis 4-stelligen) Katalogkennungen, die nicht mit 5 beginnen (Ausschlussbedingung).

Tabelle 23: Beispiele für Katalogkennungen in der UDS/SQL-Pubset-Deklaration

Kommandofolge für die Definition und Zuweisung einer UDS/SQL-Pubset-Deklaration:

```
/CREATE-JV JV-NAME=UDS-PUB-DECL
/MODIFY-JV JV-CONTENTS=UDS-PUB-DECL, -
/ SET-VALUE='A001 B001 <C015:C025> <D015:D025,F015:F045> 5*'
/SET-JV-LINK LINK-NAME=UDSPS01,JV-NAME=UDS-PUB-DECL
```

Kommandofolge für die Definition und Zuweisung einer UDS/SQL-Pubset-Deklaration mit Ausschlussbedingung:

```
/CREATE-JV JV-NAME=UDS-PUB-DECL
/MODIFY-JV JV-CONTENTS=UDS-PUB-DECL, SET-VALUE=' -5*'
/SET-JV-LINK LINK-NAME=UDSPS01,JV-NAME=UDS-PUB-DECL
```

Der Default Public Volume Set der Ablaufkennung wird von UDS/SQL immer implizit berücksichtigt und braucht daher nicht in die UDS/SQL-Pubset-Deklaration aufgenommen zu werden. Es ist nicht möglich, den Default Public Volume Set der Ablaufkennung von der Verwendung auszuschließen.

Die Zuweisung einer UDS/SQL-Pubset-Deklaration, die nur Leerzeichen enthält, ist zulässig; es wird nur der Default Public Volume Set der Ablaufkennung berücksichtigt.

Bei Verwendung von Dateien auf Privatplatte müssen alle Pubsets in der UDS/SQL-Pubset-Deklaration enthalten sein, deren Kataloge für den Betrieb benötigte Dateien auf Privatplatte verwalten.

In die Eindeutigkeitsprüfung werden Datenbankdateien (Realms, ALOG-Dateien, HASHLIB, COSSD), Systemdateien (SLF, Temporaere Realms, RLOG-Dateien, DB-Status-Datei) und die Arbeitsdateien der Dienstprogramme einbezogen.

In die Eindeutigkeitsprüfung werden **nicht** einbezogen die Ladeparameterdateien des DBH, die Bibliotheksdateien für zu ladende oder nachzuladende UDS/SQL-Programmteile und Datenmodule (SSITAB, PLEX) sowie die Verteiltabellendateien fuer UDS-D.

Bei Ladeparameter PP DISTABLE, bei DAL &ADD DISTRIBUTION sowie bei DAL &SAVE DISTRIBUTION können Sie die Katalogkennung für die jeweilige Datei ohne Rückwirkung auf die UDS/SQL-Pubset-Deklaration angeben.

In der Initialisierung wird die UDS/SQL-Pubset-Deklaration vom DBH bzw. vom jeweiligen Dienstprogramm geprüft. Ebenso prüft der DBH eine mittels DAL NEW PUBSETS neu zugewiesene UDS/SQL-Pubset-Deklaration vollständig. Fehlerhafte Zuweisungen bei Session- oder Dienstprogrammbeginn führen zum Abbruch. Bei fehlerhafter Änderung der UDS/SQL-Pubset-Deklaration mittels DAL wird diese abgewiesen und die bestehende UDS/SQL-Pubset-Deklaration weiter genutzt.

Folgende Ursachen für fehlerhafte Zuweisungen können auftreten:

- Eine nicht vorhandene oder nicht zugreifbare Jobvariable wird zugewiesen (DBH-Meldung UDS0752, Dienstprogramm-Meldungen 0048 und 0049).
- Die Catid-Gruppen sind syntaktisch fehlerhaft angegeben (DBH-Meldung UDS0748, Dienstprogramm-Meldung 0045).
- Bei der syntaktischen Prüfung einer Catid-Gruppe wird ein Fehler festgestellt (DBH-Meldung UDS0749 mit Insert CMDWCC).
- Die Nutzung einer Catid-Gruppe beim Katalogzugriff während der syntaktischen Prüfung führt zu einem Fehler (DBH-Meldung UDS0749 mit Insert FSTAT, Dienstprogramm-Meldung 0055).

Die UDS/SQL-Pubset-Deklaration muss mit den Datenträgerzuweisungen für RLOG- und ALOG-Dateien verträglich sein. Der DBH prüft diese Verträglichkeit. Aktionen, die insofern zu unverträglicher Datenträgerzuweisung führen, werden abgewiesen.

9.2.1.1 Nutzung der UDS/SQL-Pubset-Deklaration im DBH

Während der Initialisierung des DBH wird eine vorliegende UDS/SQL-Pubset-Deklaration geprüft. Die Einstellungen einer fehlerfreien UDS/SQL-Pubset-Deklaration werden für die weitere Verwendung vermerkt.

Wenn während der Initialisierung des DBH ein Fehler bei der Bearbeitung einer zugewiesenen Pubset-Deklarations-Jobvariablen auftritt, wird die Session beendet, da die offensichtlich beabsichtigte Nutzung der Jobvariablen nicht erfolgen kann.

Generell, aber insbesondere bei im Hintergrund ablaufendem DBH, wird empfohlen, eine Pubset-Deklarations-Jobvariable in der Startprozedur zuzuweisen, da dies später aus einer anderen Task nicht mehr möglich ist. Wenn Sie keine Beschränkung der zu berücksichtigenden Pubsets wünschen, können Sie die Default-Einstellung "*" in die Pubset-Deklarations-Jobvariable eintragen.

Dadurch können Sie für einen aktiven Independent-DBH die UDS/SQL-Pubset-Deklaration dynamisch verändern: Tragen Sie die gewünschte UDS/SQL-Pubset-Deklaration in die zugewiesene Pubset-Deklarations-Jobvariable ein und weisen die aktualisierte UDS/SQL-Pubset-Deklaration anschließend mit DAL NEW PUBSETS zu.

Beachten Sie bei der Definition einer UDS/SQL-Pubset-Deklaration, dass sämtliche für die Anwendung erforderlichen Pubsets erfasst sind, damit während des Betriebes keine Störung auftritt. Dies gilt insbesondere für die mit Ladeparametern (LOG, LOG-2 oder RESERVE) oder DAL-Kommandos (LOG, LOG-2 oder RESERVE) gemachten Angaben zur Lage der Loggingdateien.

Die Auswirkungen von widersprüchlichen Angaben hierzu wird durch folgende Maßnahmen begrenzt:

- Der DBH wird nicht hochgefahren, falls eine der RLOG-Logging-Angaben PP LOG, PP LOG-2 oder PP RESERVE außerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt.
- Eine neue Festlegung für die RLOG-Logging-Angaben mittels DAL MODIFY LOG, MODIFY LOG-2, oder MODIFY RESERVE wird abgewiesen, falls die neue Angabe außerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration oder außerhalb des Pubset-Raumes einer bereits mit DAL NEW PUBSETS bestimmten, aber noch nicht durch PERFORM wirksam gewordenen neuen UDS/SQL-Pubset-Deklaration liegt.
- Eine Datenbank kann nicht zugeschaltet werden, falls eine der ALOG-Logging-Angaben, die mit den Parametern DEFAULT-SUPPORT oder RESERVE-SUPPORT der BMEND-Anweisungen START-LOG festgelegt wurden, außerhalb des Pubset-Raumes der aktuellen UDS/SQL-Pubset-Deklaration liegt.
- Die Veränderung der aktuellen UDS/SQL-Pubset-Deklaration mittels DAL NEW PUBSETS wird abgewiesen, falls die neue UDS/SQL-Pubset-Deklaration nicht die Pubsets umfasst, die für LOG, LOG-2, RESERVE der RLOG-Logging, sowie für DEFAULT-SUPPORT und RESERVE-SUPPORT der ALOG-Logging jeder zugeschalteten Datenbank vorgegeben sind.

Die Einstellungen einer fehlerfreien UDS/SQL-Pubset-Deklaration werden protokolliert. Sie können sich die aktuelle UDS/SQL-Pubset-Deklaration mit DAL DISPLAY PUBSETS anzeigen lassen.

9.2.1.2 Nutzung der UDS/SQL-Pubset-Deklaration in den Dienstprogrammen

Beim Beginn eines Dienstprogramms wird eine vorliegende UDS/SQL-Pubset-Deklaration geprüft. Die Einstellungen einer fehlerfreien UDS/SQL-Pubset-Deklaration werden für die weitere Verwendung vermerkt, sie werden aber nicht protokolliert. Eine beim Dienstprogrammbeginn vorliegende fehlerhafte Jobvariablen-Zuweisung oder Zuweisung einer fehlerhaften UDS/SQL-Pubset-Deklaration führt zum Abbruch des Dienstprogramms.

Bei der Festlegung der DEFAULT-SUPPORT und RESERVE-SUPPORT der ALOG-Logging mit der BMEND-Anweisung START-LOG erfolgt keine Prüfung gegen die zugewiesene UDS/SQL-Pubset-Deklaration, da die Festlegung evtl. für eine andere Einsatzumgebung getroffen wird.

9.2.1.3 Sonstige Nutzung der UDS/SQL-Pubset-Deklaration

Für den COBOL-Compiler, der bei der Kompilierung eines UDS/SQL-Anwenderprogramms mit COBOL-DML die COSSD-Datei liest, gilt eine UDS/SQL-Pubset-Deklaration nicht. Die COSSD-Datei kann dem COBOL-Compiler mit dem Kommando ADD-FILE-LINK mit LINK-NAME=UDSCOSSD explizit inklusive Katalogkennung zugewiesen werden (siehe Handbuch „[Anwendungen programmieren](#)“, Kapitel „Besonderheiten der COBOL-DML“).

Es wird keine gesonderte Behandlung von Dateinamen aus einem Alias-Katalog (ACS) durchgeführt. Insbesondere sollten Sie keine Aliasnamen mit Katalog- und/oder Benutzerkennung zulassen (COMPLETE-ALIAS-NAMES=*ALLOWED), da diese den beabsichtigten Wirkungen der UDS/SQL-Pubset-Deklaration widersprechen (siehe [Abschnitt „ACS \(Alias Catalog Service\)“](#)).

9.2.2 Session-Jobvariable

Der DBH versorgt eine Session-Jobvariable zur Verbesserung der automatischen Administration. Diese Jobvariable können Sie nutzen, um Benutzeraufträge und Programme zu steuern.

Folgende Jobvariable wird vom UDS/SQL-DBH versorgt:

UDSSQL.DBH.*session*

session

ist der variable Teil des Namens der Jobvariable. Als Standardwert werden die ersten acht Zeichen von *konfigurationsname* verwendet. Der Datenbankadministrator kann aber einen abweichenden Namen für *session* vor dem DBH-Start in der Mastertask bzw. in der linked-in-Task über das folgende Kommando festlegen:

```
/SET-FILE-LINK LINK-NAME=UDSDBHJV,FILE-NAME=session
```

Falls die Jobvariable UDSSQL.DBH.*session* beim DBH-Start nicht existiert, wird sie mit den folgenden Eigenschaften angelegt:

```
ACCESS=*WRITE
USER-ACCESS=*ALL-USERS
BASIC-ACL=*NONE (einfache Zugriffskontrollliste)
```

Wenn die Jobvariable mit anderen Eigenschaften genutzt werden soll, müssen Sie sie vor dem DBH-Start bzw. vor dem Dienstprogrammlauf mit den gewünschten Eigenschaften anlegen. Die Zugriffskontrolle bei der Nutzung der Jobvariablen ist mit Standardmitteln (z.B. Basic-ACL, Guards) realisierbar.

Vor der Verwendung einer Jobvariablen in einer Fremdkennung müssen Sie diese entweder selbst anlegen oder durch geeignete Massnahmen (z.B. Miteigentümerschaft in einer fremden Kennung) dafür sorgen, dass sie dynamisch angelegt werden kann.



Bitte beachten Sie auch die „Hinweise zur Verwendung der Jobvariablen“.

Die Jobvariable UDSSQL.DBH.*session* wird folgendermaßen versorgt:

Spalte	Inhalt	Bedeutung
1-30	UDSDBH_STARTING / UDSDBH_ACTIVE / UDSDBH_CLOSE_INITIATED / UDSDBH_NORMAL_END / UDS- DBH_ABNORMAL_END	Zustand (1)
31-40 41-48	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	DBH-Start (2) Datum Uhrzeit
49-58 59-66	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	DBH-Ende (3) Datum Uhrzeit

67-68	01	Versionskennzeichen des Layouts der Session-Jobvariablen
69-76	<i>sessionname</i>	Konfigurationsname der DBH-Session
77-84	<i>host</i>	Host der DBH-Session
85-92	<i>nnnnnnnn</i> / 'BLANK'	Session-Abschnittsnummer (4)
93		Zustand der DBH-Betriebsunterbrechung:
	B /	– aktuelle Betriebsunterbrechung
	'BLANK'	– aktuell keine Betriebsunterbrechung
94-96		Art der aktuellen oder letzten DBH-Betriebsunterbrechung (5)
	DAL /	– wegen DALs (PERFORM)
	DBH /	– wegen systeminterner Aktivitäten
	'BLANK'	– noch keine Betriebsunterbrechung aufgetreten
97		Zuschalten von Datenbank oder Realm bei aktueller oder letzter Betriebsunterbrechung (5)
	A /	– ja
	– /	– nein
	'BLANK'	– noch keine Betriebsunterbrechung aufgetreten
98		Abschalten von Datenbank oder Realm bei aktueller oder letzter Betriebsunterbrechung (5)
	D /	– ja
	– /	– nein
	'BLANK'	– noch keine Betriebsunterbrechung aufgetreten
99		Aktivität bzgl. ALOG-Dateien bei aktueller oder letzter Betriebsunterbrechung (5)
	A /	– ja
	– /	– nein
	'BLANK'	– noch keine Betriebsunterbrechung aufgetreten
100		Aktivität bzgl. RLOG-Dateien bei aktueller oder letzter Betriebsunterbrechung (5)
	R /	– ja
	– /	– nein
	'BLANK'	– noch keine Betriebsunterbrechung aufgetreten

101-112	'BLANK'	reserviert
113-122 123-130	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	Beginn der Vorbereitungsphase der aktuellen oder letzten Betriebsunterbrechung (5) Datum Uhrzeit
131-138	<i>hh.mm.ss</i>	Beginn der Durchführungsphase der aktuellen oder letzten Betriebsunterbrechung (5) Uhrzeit
139-146	<i>hh.mm.ss</i>	Ende der letzten Betriebsunterbrechung (5) Uhrzeit
147-162	'BLANK'	reserviert
163-172 173-180	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	Letzte Änderung in Jobvariable Datum Uhrzeit

Tabelle 24: Aufbau der Session-Jobvariable für UDS/SQL

Anmerkungen

(1) *Zustand*

Inhalt	wann gesetzt?
UDSDBH_STARTING	bei DBH-Start
UDSDBH_ACTIVE	nach erfolgreicher Initialisierung (z.B. System Ready-Meldung des independent DBH)
UDSDBH_CLOSE_INITIATED	wenn keine Aufträge mehr im DBH zugelassen werden, weil das DBH-Ende von der UDS/SQL-Administration oder aus internen Gründen vom DBH eingeleitet wurde
UDSDBH_NORMAL_END / UDSDBH_ABNORMAL_END	bei DBH-Ende

- (2) *DBH-Start* wird bei DBH-Start mit 0 initialisiert und nach erfolgreicher Initialisierung (z.B. System Ready-Meldung des independent DBH) mit der aktuellen Uhrzeit versorgt.
- (3) *DBH-Ende* wird bei DBH-Start mit 0 initialisiert und bei DBH-Ende mit der aktuellen Uhrzeit versorgt.
- (4) *Session-Abschnittsnummer* wird im bei DBH-Start mit Leerzeichen initialisiert und während der Initialisierung mit einem Wert versorgt, der den Session-Abschnitt eindeutig identifiziert und bis zum Ende des Session-Abschnitts unverändert bleibt.

Mit der Session-Abschnittsnummer kann eine Zuordnung von Datenbank-Jobvariablen zu einer aktuellen Session geprüft werden: Nur wenn die Session Abschnittsnummern in der Datenbank-Jobvariablen und in der Session-Jobvariablen übereinstimmen, sind die Inhalte der Datenbank-Jobvariablen bezogen auf die aktuelle Session gültig.

- (5) Anzeigen und Zeitpunkte der verschiedenen Phasen einer Betriebsunterbrechung beziehen sich immer auf die in *Zustand der DBH-Betriebsunterbrechung* (Byte 93) angezeigte aktuelle oder letzte Betriebsunterbrechung.

Interne Betriebsunterbrechungen in der Start- oder Beendigungs-Phase des DBH werden nicht angezeigt.

9.2.3 Datenbank-Jobvariable

Der DBH und die Dienstprogramme DDL-Compiler, SSL-Compiler, BGSIA, BGSSIA, BPRIVACY, BMEND, BREORG, BCHANGE, BRENAMER und BALTER versorgen eine Datenbank-Jobvariable zur Verbesserung der automatischen Administration. Diese Jobvariable können Sie nutzen, um Benutzeraufträge und Programme zu steuern.

Der Name der Datenbank-Jobvariablen setzt sich wie folgt zusammen:

UDSSQL.DB.*datenbankname*[.*copyname*][.*nr*]

datenbankname

ist der Name der Datenbank (max. 17 Zeichen)

copyname

ist der Copyname einer im Modus SHARED-RETRIEVAL im DBH zugeschalteten Datenbank

nr

ist die einstellige Nummer (1..9) der Datenbank bei Datenbanken, die in verschiedenen Sessions parallel im Modus SHARED-RETRIEVAL genutzt werden.

Dienstprogramme nutzen nur eine Jobvariable ohne *nr*.

Die Datenbank-Jobvariable liegt immer in der Kennung und im Pubset des DBDIR der entsprechenden Datenbank.

Wenn eine Datenbank-Jobvariable zum Zeitpunkt der Initialisierung nicht existiert, wird sie mit den folgenden Eigenschaften angelegt:

```
ACCESS=*WRITE
USER-ACCESS=*ALL-USERS
BASIC-ACL=*NONE (einfache Zugriffskontrollliste)
```

Wenn die Jobvariable mit anderen Eigenschaften genutzt werden soll, müssen Sie sie vor dem DBH-Start bzw. vor dem Dienstprogrammablauf mit den gewünschten Eigenschaften anlegen. Die Zugriffskontrolle bei der Nutzung der Jobvariablen ist mit Standardmitteln (z.B. Basic-ACL, Guards) realisierbar.

Vor der Verwendung einer Jobvariablen in einer Fremdkennung müssen Sie diese entweder selbst anlegen oder durch geeignete Massnahmen (z.B. Miteigentümerschaft in einer fremden Kennung) dafür sorgen, dass sie dynamisch angelegt werden kann.

i Bitte beachten Sie auch die „[Hinweise zur Verwendung der Jobvariablen](#)“.

Besonderheiten bei Datenbanken im Modus SHARED-RETRIEVAL

Beim Öffnen einer Datenbank im Modus SHARED-RETRIEVAL prüft der DBH zunächst, ob bereits passende Jobvariablen mit Nummerierung existieren. Wenn eine dieser Jobvariablen den aktuellen Name der Konfiguration enthält, wird sie verwendet. Andernfalls wählt der DBH einen noch nicht genutzten Jobvariablennamen aus, legt die entsprechende Jobvariable an und trägt den aktuellen Konfigurationsnamen ein. Leere oder noch nicht mit Konfigurationsnamen versorgte Jobvariablen werden nicht genutzt, weil bei paralleler Nutzung der Datenbanken in mehreren DBH Sessions sichergestellt werden muss, dass durch jede DBH Session eine andere Datenbank-Jobvariable genutzt wird. Eine entsprechende Jobvariable ohne Nummer wird in dieses Verfahren einbezogen und bei der freien Auswahl bevorzugt belegt.

Datenbank-Jobvariablen im SHARED-RETRIEVAL Einsatz werden beim Zuschalten der Datenbank immer vollständig initialisiert.

Es wird empfohlen, bei echt parallelem SHARED-RETRIEVAL-Einsatz die Datenbank-Jobvariablen vor der Nutzung einzurichten und den Konfigurationsnamen der DBH-Session (Spalte 61-68) zu versorgen.

Besondere Regeln bei der Nummerierung der Jobvariablen sind nicht zu beachten. Falls schon alle der 10 möglichen Jobvariablen genutzt werden und damit anderen Konfigurationen zugeordnet sind, kann keine Jobvariable für die betroffene Datenbank angelegt werden. Es wird eine entsprechende Fehlermeldung durch den DBH ausgegeben.

Struktur der Datenbank-Jobvariable

Die Datenbank-Jobvariable UDSSQL.DB.*datenbankname*.[*copyname*][*.n*] wird folgendermaßen versorgt:

Spalte	Inhalt	Bedeutung
1-2	01	Versionskennzeichen des Layouts der Session-Jobvariablen
3-19	<i>cccccccccccccccc</i>	Name der Datenbank (1)
20-26	<i>cccccc</i> / 'BLANK'	Copyname (2)
27-32	<i>cccccc</i>	DB-Layout-Version
33	C / I / 'BLANK'	Konsistenz (3)
34-40	'BLANK'	reserviert
41-50	UPDATE / RETR/ WARMSTART/ OPEN / DROP / CLOSE / ERROR	Verarbeitungszustand (4): <ul style="list-style-type: none"> • Zuschaltmodus beim DBH • Zuschaltmodus bei einem Dienstprogramm • Ursache der letzten Beendigung bei DBH oder einem Dienstprogramm
51	'BLANK' / A	Aktivierung ALOG (5)
52	'BLANK' / O	Kennzeichen für Online-Kopierfähigkeit (6)
53-60	DBH/ LKIN-DBH / UTIL-DBH / <i>utility</i>	DBH oder Dienstprogramm (7) Aktueller oder letzter Bearbeiter der Datenbank <ul style="list-style-type: none"> • independent DBH • linked-in DBH • Dienstprogramm mit linked-in-DBH (DDL-/SSL-Compiler, BGSIA, BGSSIA, BPRIVACY) • Name eines ändernden Dienstprogramms
61-68	<i>cccccccc</i> / 'BLANK'	Konfigurationsname der DBH-Session (8)
69-72	<i>cccc</i> / 'BLANK'	Standardkatalog der DBH Kennung (9)
73-81	<i>\$cccccccc</i> / 'BLANK'	Kennung der DBH-Session (mit \$) (9)

82-89	<i>nnnnnnnn</i> / 'BLANK'	Sessionabschnittsnummer (10)
90-99 100-107	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	Verarbeitungsbeginn (11) Datum Uhrzeit
108-117 118-125	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	Verarbeitungsende (12) Datum Uhrzeit
126-135 136-143	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	letzter ALOG-Wechsel (13) Datum Uhrzeit
144-152	<i>nnnnnnnnnn</i> / 'BLANK'	ALOG-Folgenummer (14)
153-162	<i>nnnnnnnnnn</i> / 'BLANK'	Größe der ALOG-Datei (15)
163-167	<i>nnnnn</i> / 'BLANK'	Erweiterungen der ALOG-Datei (16)
168-182	'BLANK'	reserviert
183-192 193-200	<i>jjjj-mm-tt</i> <i>hh.mm.ss</i>	Letzte Änderung in Jobvariable Datum Uhrzeit

Tabelle 25: Aufbau der Datenbank-Jobvariable für UDS/SQL

Anmerkungen

- (1) *Datenbankname* ist der auch im Jobvariablen-Namen enthaltene Name der Datenbank.
- (2) *Copyname* wird nur bei Schattendatenbank versorgt.
- (3) *Konsistenz* wird nur vom DBH versorgt und zeigt an, ob die Datenbank konsistent ("C") oder inkonsistent ("I") ist. In diesem Sinne ist eine Datenbank dann inkonsistent, wenn der DBH gerade ändernd auf der Datenbank aktiv ist und evtl. noch nicht alle Änderungen in die Datenbankdateien zurückgeschrieben sind.

Die Datenbank kann aber auch inkonsistent sein, wenn eine Bearbeitung durch den DBH abnormal beendet worden ist. Durch einen Warmstart der Datenbank kann diese im Allgemeinen wieder in einen konsistenten Stand überführt werden.

- (4) *Verarbeitungszustand* zeigt den Zuschaltmodus bzw. die Ursache der letzten Beendigung an. Temporäre Einschränkungen des Zugriffs (z.B. wegen DAL ACCESS) oder Einschränkungen der Betriebsart (weil z. B. weil die RLOG-Datei nicht genutzt werden kann) werden nicht angezeigt.

ERROR wird gesetzt, wenn die Datenbank kontrolliert inkonsistent abgeschaltet oder ihre Bearbeitung wegen sonstiger Fehler beendet wird. Im letzteren Fall ist es durchaus möglich, dass die Datenbank weiterhin konsistent ist.

In Einzelfällen ist es jedoch möglich, dass die DBH-Session abnormal beendet wird, ohne dass der Verarbeitungszustand aktualisiert werden kann. Dann ist im Allgemeinen die Datenbank weiterhin inkonsistent und der Verarbeitungszustand bleibt UPDATE. Bei einem anschließenden Warmstart durch den DBH wird der Verarbeitungszustand WARMSTART gesetzt und beim Ende des Warmstarts der für

die Bearbeitung gewünschte Zustand (z.B. UPDATE) eingetragen. Ändernde Dienstprogramme versorgen die Zustände OPEN, CLOSE und evtl. ERROR.

In manchen Fehlersituationen ist bei den Dienstprogrammen eine kontrollierte Programmbeendigung nicht mehr möglich. In diesem Fall kann auch der Zustand ERROR nicht mehr korrekt gesetzt werden. Die Jobvariable verbleibt dann auch nach Beendigung des Dienstprogramms im Zustand OPEN.

Außerdem ist es möglich, dass in der letzten Phase der Beendigung eines Dienstprogrammes noch ein Fehler auftritt, nachdem die Jobvariable bereits mit dem Zustand CLOSE versorgt ist. Dann wird die Jobvariable nach Möglichkeit noch mit dem Zustand ERROR versorgt.

- (5) *Aktivierung AL OG* zeigt an, ob für die Datenbank A LOGGING eingeschaltet ist ("A") oder nicht (Leerzeichen). Die aktuelle Verarbeitung im DBH kann ohne A LOGGING erfolgen, wenn z.B. die Datenbank im Modus SHARED-RETRIEVAL zugeschaltet ist.
- (6) *Kennzeichen für Online-Kopierfähigkeit*
"O" zeigt an, dass für die Datenbank aktuell eine Online-Kopie erstellt werden kann, d.h. für die Datenbank ist die Online-Kopierfähigkeit eingeschaltet (BMEND ENABLE-ONLINE-COPY) und die Datenbank ist an einen DBH angeschlossen.
In allen anderen Fällen enthält das Feld Leerzeichen.
Diese Eigenschaft sollte grundsätzlich geprüft werden, bevor mit COPY-FILE eine inkonsistente Kopie parallel zu einer DBH-Session gezogen wird.
Das Feld wird **nur vom DBH** versorgt und beim Abschalten der Datenbank gelöscht. Es wird insbesondere **nicht** von BMEND gesetzt, wenn gerade die Online-Kopierfähigkeit eingeschaltet wurde.
- (7) *DBH oder Dienstprogramm* wird beim Zuschalten der Datenbank durch den DBH bzw. von dem Dienstprogramm versorgt und bleibt dann unverändert.
- (8) *Konfigurationsname der DBH-Session* zeigt den aktuellen bzw. den letzten Namen der DBH Konfiguration an. Das Feld wird beim Zuschalten der Datenbank versorgt und bleibt dann unverändert.
Dienstprogramme, die nicht den linked-in DBH nutzen, tragen Leerzeichen ein. Im Modus SHARED-RETRIEVAL sollte eine Datenbank-Jobvariable vor dem Zuschalten der Datenbank zu einer DBH Session angelegt und mit dem Konfigurationsnamen versorgt werden.
- (9) *Standardkatalog der DBH Kennung* und *Kennung der DBH-Session* enthalten die Werte der entsprechenden DBH Session. Diese Werte können zum eindeutigen Zugriff auf die Session-Jobvariable genutzt werden.

Dienstprogramme, die nicht den Linked-in-DBH nutzen, tragen Leerzeichen ein.
- (10) *Sessionabschnittsnummer* wird beim Zuschalten der Datenbank mit dem aktuellen Wert der bearbeitenden Session versorgt und beim Abschalten durch den DBH mit Leerzeichen gelöscht.

Dienstprogramme, die nicht den Linked-in-DBH nutzen, tragen bereits beim Zuschalten Leerzeichen ein.
- (11) *Verarbeitungsbeginn* wird beim Zuschalten (DBH) bzw. Öffnen (Dienstprogramm) der Datenbank versorgt und bleibt dann unverändert.
- (12) *Verarbeitungsende* wird beim Abschalten (DBH) bzw. Schließen (Dienstprogramm) der Datenbank versorgt. Beim abnormalen Ende von Dienstprogrammen hängt die Versorgung davon ab,

ob die Jobvariable insgesamt aktualisiert wird. Während des Zustandes UPDATE bzw. RETR enthält das Feld Leerzeichen.

- (13) *Letzter ALOG-Wechsel* zeigt den Zeitpunkt des letzten ALOG-Wechsels bzw. der (Re-)Aktivierung des ALOGGINGS an. Das Datum bleibt auch dann erhalten, wenn das ALOGGING ausgeschaltet wird.
- (14) *ALOG-Sequenz-Number* wird von DBH oder Dienstprogrammen beim ALOG-Wechsel versorgt, falls ALOGGING eingeschaltet ist.
- (15) *Größe ALOG-Datei* zeigt die Größe des genutzten Teils der ALOG-Datei in PAM-Seiten an. Diese Größe kann geringfügig von der Dateigröße aus DVS-Sicht abweichen. Diese Abweichung kann entweder durch eine Rundung wegen der genutzten Größe der Allokierungseinheit (3, 4 oder 32 PAM-Seiten) entstehen, oder weil durch Verdoppelung der Sekundärzuweisung (vgl. Klasse-2-Systemparameter DMMAXSC bzw. Parameter MAXIMAL-ALLOCATION bei ADD-/MODIFY-MASTER-CATALOG-ENTRY) eine aktuelle Dateierweiterung größer ist als die von UDS/SQL angeforderte Erweiterung.

Das Feld wird nur vom DBH mit einem aktuellen Wert versorgt. Dienstprogramme, die nicht den Linked-in-DBH nutzen, tragen Leerzeichen ein.

- (16) *Erweiterungen der ALOG-Datei* zeigt die Anzahl der Extents der ALOG-Datei an. Das Feld wird **nur durch den DBH** mit einem aktuellen Wert gefüllt. Die für die Änderung der Datenbank zuständigen Dienstprogramme (BRENAME, BCHANGE, BALTER, BREORG, BMEND) tragen Blanks ein.

Ausgabebeispiel:

```
%01MDV29B6          004.00C      UPDATE    A DBH      MDV29B6 IUDS$UDSDEV
024AD708EF2018-01-2512:36:10      2018-01-2512:34:520000000010000000
19200001          2018-01-2512:36:10
```

Wobei:

- ALOG-Folgenummer = 000000001
- Größe der ALOG-Datei = 0000000192
- Erweiterungen der ALOG-Datei = 00001

9.2.4 Hinweise zur Verwendung der Jobvariablen

Versorgung

Die Jobvariablen werden von UDS/SQL grundsätzlich in ihrer gesamten Länge genutzt. Freie Bereiche sind für zukünftige Erweiterungen reserviert. Extern durchgeführte Änderungen werden somit überschrieben und sollten grundsätzlich unterbleiben, da sie zu nicht definierten Inhalten der Jobvariablen führen können.

Auswertung

Bei der automatisierten Überwachung von Einzelzuständen oder Änderungen des Inhalts in einem beliebigen Feld ist zu beachten, dass Änderungen sehr schnell hintereinander erfolgen können. Dadurch können unter Umständen nicht alle Zwischenzustände abgefangen werden.

Der Inhalt von Session- und Datenbank-Jobvariablen sollte nicht herangezogen werden, um zu prüfen, ob die bearbeitenden Programme aktiv sind. Diese Information ist aus entsprechenden Monitorjobvariablen zu ermitteln.

Störungen

Temporäre Störungen bei der Versorgung von Session- oder Datenbank-Jobvariablen werden nur einmal nach ihrem ersten Auftreten gemeldet.

Wenn eine Jobvariable vom DBH nicht versorgt werden kann, weil eine behebbare Störung vorliegt, wird eine Meldung UDS0700 mit dem Insert `JOBV_SYSTEM_ERROR` und Return-Informationen (SEC-RC) ausgegeben. Die Session wird fortgesetzt.

Mit `HELP-MSG MSG-ID=JVSxxxx` (xxxx=die letzten vier Zeichen von SEC-RC) können Sie im Allgemeinen die Ursache der Störung ermitteln und ggf. beseitigen.

Sofern eine dauerhafte Störung des Jobvariablen-Subsystems erkannt wird, wird eine Meldung UDS0700 mit dem Insert `JOBV_PERMANENT_ERROR` ausgegeben. Die Session wird fortgesetzt.

Wenn eine Störung in einer laufenden Session beseitigt ist, wird die Session Jobvariable ab der nächsten Versorgung und die Datenbank-Jobvariable ab dem nächsten Zuschalten der Datenbank ohne zusätzliche Meldung mit den aktuell gültigen Werten bereitgestellt.

9.2.5 BMEND-Jobvariable

Das Dienstprogramm BMEND versorgt optional eine Jobvariable, wenn sicherungsrelevante Informationen erzeugt werden (Anweisung SHOW-LOG-INFORMATION) bzw. wenn AFIMs in eine Datenbank eingespielt werden (Anweisung UPDATE-DATABASE). Damit können Sicherungsbetrieb und Recovery automatisiert werden. Die Einzelheiten können Sie der Beschreibung des Dienstprogrammes BMEND im Handbuch „[Sichern, Informieren und Reorganisieren](#)“ entnehmen.

9.3 Mehrere UDS/SQL-Versionen parallel nutzen

Verschiedene Versionen von UDS/SQL können in einem BS2000-System parallel genutzt werden. UDS/SQL nutzt die 7-stellige Versionsbezeichnung. Damit können auch Korrekturversionen einer Hauptversion parallel betrieben werden; hierzu müssen die Korrekturversionen nur in verschiedenen Benutzerkennungen installiert werden. Dies kann zum Beispiel genutzt werden, wenn eine Korrekturversion installiert und getestet werden soll, während der Produktivbetrieb mit der Vorgängerversion weiterläuft.

Mehrere UDS/SQL-Subsysteme verschiedener (Korrektur-)Versionen können mit DSSM parallel geladen werden.

Wenn mehrere UDS/SQL-Versionen installiert bzw. als Subsysteme geladen sind und keine Version speziell ausgewählt wird, wird vom BS2000 die Verbindung zu einer Version standardmäßig hergestellt.

Mit IMON-GPN können Sie in jeder Task explizit angeben, welche UDS/SQL-Version genutzt werden soll, indem Sie in der Start-Prozedur der Mastertask bzw. des Anwendungsprogramms das folgende BS2000-Kommando absetzen:

```
SELECT-PRODUCT-VERSION PRODUCT-NAME=produkt,VERSION=version,SCOPE=*TASK
```

Die Versionsnummer sollte immer inklusive Freigabe- und Korrekturstand angegeben werden (z.B. 02.9B00).

Die Zuweisung der Version wirkt auf Versionen, die im Software Configuration Inventory (SCI) eingetragen sind, auf parallel geladene Subsysteme oder auf parallel installierte Startkommandos von UDS/SQL.

Die jeweils mit SELECT-PRODUCT-VERSION angegebene Versionsnummer muss nicht in jeder Anwenderprozedur fest vergeben werden, sondern kann variabel gestaltet und über einen zentral geführten Wert gesetzt werden (z.B. für eine Konfiguration in einer Jobvariablen der zugehörigen DBH-Kennung oder in einer strukturierten Variablen).

Die Information, welche Versionen von UDS/SQL verfügbar sind und wo die Produktdateien installiert sind, erhalten Sie mit dem Kommando

```
SHOW-INSTALLATION-PATH INSTALLATION-UNIT=UDS-SQL(VERSION=*ALL)
```

Bei ENTER-Jobs können Sie das Kommando SELECT-PRODUCT-VERSION in einer ENTER-Datei absetzen.

Wenn IMON-GPN eingesetzt ist, wird die in der Mastertask gesetzte Version auch in den Servertasks und in der UDS-D-Task genutzt.

9.3.1 Beispiel: Parallele Nutzung von zwei UDS/SQL-Versionen

Die Korrekturversion UDS-SQL V2.9B10 soll getestet werden, während der Produktivbetrieb mit der Version UDS-SQL V2.9B00 gleichzeitig weiterläuft. Über Jobvariablen soll es möglich sein, die jeweils gewünschte Version beim DBH-Start zu laden. Dabei werden parallele Subsysteme und Startkommandos genutzt.

Jobvariablen erstellen

Der Datenbankadministrator kann Jobvariablen erstellen, die die Werte für die jeweiligen Subsystemversionen enthalten sollen. Die Jobvariable JV.VERSAND soll den Wert 02.9B00, die Jobvariable JV.TEST den Wert 02.9B10 bekommen. Anschließend wird der Inhalt der Jobvariablen überprüft.

```
/CREATE-JV JV=JV.VERSAND
/MODIFY-JV JV=JV.VERSAND,SET-VALUE=C'02.9B00'
/CREATE-JV JV=JV.TEST
/MODIFY-JV JV=JV.TEST,SET-VALUE=C'02.9B10'
/SHOW-JV JV=JV.VERSAND
% 02.9B00
/SHOW-JV JV=JV.TEST
% 02.9B10
```

DBH-Startprozeduren erstellen

In den DBH-Startprozeduren DBHSTART.TEST und DBHSTART.VERSAND soll für die gewünschte Version der entsprechende Jobvariablenname eingesetzt werden.

DBHSTART.TEST

```
/SET-LOGON-PARAMETERS
/ASSIGN-SYSDTA TO=*SYSCMD
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=TEST
/CREATE-FILE FILE-NAME=TEST.DBSTAT,SUPPRESS-ERRORS=*FILE-EXISTING
/CREATE-FILE FILE-NAME=TEST.DBSTAT.SAVE,SUPPRESS-ERRORS=*FILE-EXISTING
/SET-JV-LINK LINK-NAME=VERS,JV-NAME=JV.TEST ----- (1)
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=&(*VERS) ----- (2)
/START-UDS-DBH
/ASSIGN-SYSDTA TO=*PRIMARY
/EXIT-JOB
```

DBHSTART.VERSAND

```
/SET-LOGON-PARAMETERS
/ASSIGN-SYSDTA TO=*SYSCMD
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND
/CREATE-FILE FILE-NAME=VERSAND.DBSTAT,SUPPRESS-ERRORS=*FILE-EXISTING
/CREATE-FILE FILE-NAME=VERSAND.DBSTAT.SAVE,SUPPRESS-ERRORS=*FILE-EXISTING
/SET-JV-LINK LINK-NAME=VERS,JV-NAME=JV.VERSAND ----- (1)
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=&(*VERS) ----- (2)
/START-UDS-DBH
/ASSIGN-SYSDTA TO=*PRIMARY
/EXIT-JOB
```

1. Der gewünschte Jobvariablenname wird mit dem Kettungsnamen VERS verknüpft.
2. Für den Kettungsnamen VERS wird der Inhalt der vereinbarten Jobvariablen eingesetzt.

Zusätzlich zu den DBH-Startprozeduren müssen auch die Startprozeduren für Anwenderprogramme angepasst werden.

Startprozeduren für Anwenderprogramme erstellen

In den Startprozeduren der Anwenderprogramme DMLTEST.TEST und DMLTEST.VERSAND soll für die gewünschte Version der entsprechende Jobvariablenname eingesetzt werden.

DMLTEST.TEST

```

/BEGIN-PROCEDURE
/SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=TEST
/ASSIGN-SYSDTA TO=*SYSCMD
/SET-JV-LINK LINK-NAME=VERS, JV-NAME=JV.TEST
/ADD-FILE-LINK LINK-NAME=$UDSSSI, FILE-NAME=LMS.SSITAB
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=& (*VERS)
/START-UDS-DMLTEST
DBH IND
LANG COB
DISPLAY RCODE, COND=RCODE NE C'00000'
PROT ON
DISPLAY RECA, L=80
SUBSCHEMA IS ADMIN
READY
SHOW SPP1
SHOW SUBS
E
SYS
/MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=*TIME (TIMEOUT=*STD)
/ASSIGN-SYSDTA TO=*PRIMARY
/RESUME-PROGRAM
/END-PROCEDURE

```

DMLTEST.VERSAND

```

/BEGIN-PROCEDURE
/SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=VERSAND
/ADD-FILE-LINK LINK-NAME=$UDSSSI, FILE-NAME=LMS.SSITAB
/ASSIGN-SYSDTA TO=*SYSCMD
/SET-JV-LINK LINK-NAME=VERS, JV-NAME=JV.VERSAND
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=& (*VERS)
/START-UDS-DMLTEST
DBH IND
LANG COB
DISPLAY RCODE, COND=RCODE NE C'00000'
PROT ON
DISPLAY RECA, L=80
SUBSCHEMA IS ADMIN
READY
SHOW SPP1
SHOW SUBS
E
SYS
/MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=*TIME (TIMEOUT=*STD)
/ASSIGN-SYSDTA TO=*PRIMARY
/RESUME-PROGRAM
/END-PROCEDURE

```

Die Bedeutung der Prozedurparameter entnehmen Sie bitte den Erklärungen zu den DBH-Startprozeduren, siehe ["DBH-Startprozeduren erstellen"](#).

Wenn die Testversion später als Produktivversion eingesetzt werden soll, bleiben die Startprozeduren unverändert.

Sie müssen lediglich den Inhalt der Jobvariablen

JV.VERSAND wie folgt verändern:

```
/MODIFY-JV JV=JV.VERSAND,SET-VALUE=C'02.9B10'
```

9.4 DAB-Caching für UDS/SQL nutzen

Software-Caching (DAB) ist im Datenbankbetrieb nutzbar. Für die Datensicherheit ist es dabei unbedingt notwendig, dass Schreib- oder Schreib-Lesepufferung in flüchtigen Cache-Medien (Hauptspeicher, Erweiterungsspeicher) vermieden wird.

Dies gilt sowohl für die Datenbankdateien als auch für Session- und Loggingdateien. Das Sicherungskonzept von UDS/SQL (siehe [Abschnitt „Das Sicherungskonzept von UDS/SQL“](#)) geht davon aus, dass ein logisch abgeschlossener Schreibvorgang auch physisch abgeschlossen ist und somit auch einen Systemabsturz z.B. wegen Stromausfall überlebt.

Eine ausführliche Beschreibung des DAB-Caching finden Sie im Handbuch „[DAB \(BS2000\)](#)“.

9.5 Datenspiegelung für UDS/SQL nutzen

In BS2000 können unterschiedliche Techniken zur transparenten Datenspiegelung genutzt werden, z.B. DRV (Dual Recording by Volume), SRDF (Symmetrix Remote Data Facility) oder TimeFinder.

Der UDS/SQL-Anwender kann solche transparenten Datenspiegelungen nutzen. Die Ausfallsicherheit der entsprechenden Dateien ist dann erheblich erhöht, da die Dateien doppelt geführt werden. Beim Ausfall einer Einheit kann der Betrieb auf der Spiegelplatte unterbrechungsfrei weitergeführt werden.

Nach einem Systemabsturz kann es jedoch sein, dass die beiden Einheiten einen unterschiedlichen Stand aufweisen. Diese müssen dann zunächst egalisiert werden.

Bei Nutzung von DRV gibt es beispielsweise in BS2000 die Möglichkeit, nach einem Absturz die DRV-Platten automatisch zu egalisieren. Die Egalisierung kann aber zu sehr langen Rekonstruktionszeiten führen, da das Betriebssystem die gesamte Platte lesen und schreiben muss. Der DBH und die Dienstprogramme führen deshalb eine gezielte Egalisierung derjenigen Datenbankbereiche durch, die einen unterschiedlichen Stand aufweisen. Damit ist eine evtl. aufwändigere Egalisierung durch das Betriebssystem nicht erforderlich.

9.6 HSMS für UDS/SQL nutzen

Mit HSMS (Hierarchisches Speicher Management System) können Online-Sicherungen von UDS/SQL-Datenbankdateien gezogen werden (siehe [Abschnitt „Online-Sicherung einer Datenbank“](#)).

Daneben bietet HSMS auch die Möglichkeit, Dateien, auf die längere Zeit nicht zugegriffen wurde, automatisch auf andere Datenträger auszulagern, siehe Handbuch „[HSMS / HSMS-SV \(BS2000\)](#)“. Wenn auf eine ausgelagerte Datei zugegriffen wird, so wird sie von HSMS automatisch zurückgeholt.

Grundsätzlich können auch UDS/SQL-Dateien mit HSMS ausgelagert werden.

i Wenn eine Datenbank als Online-Sicherung oder wegen eines DBH-Absturzes inkonsistent ist und sie vor dem Warmstart von HSMS ausgelagert wird, so kann es sein, dass der DBH nach der Wiedereinlagerung den Warmstart verweigert.

In diesem Fall müssen Sie die Datenbank vor dem Warmstart mit dem Dienstprogramm BMEND bis zum Absturzzeitpunkt nachfahren (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, BMEND-Anweisung UPDATE-DATABASE).

9.7 FASTPAM für UDS/SQL nutzen

UDS/SQL verwendet im independent DBH die Zugriffsmethode FASTPAM für alle Dateien, mit folgenden Ausnahmen:

- Realms im 2-Kbyte-Seitenformat
- Datenbankdateien, die auf Shared oder Protected Private Disks (SPD bzw. PPD) liegen
- Datenbankdateien, die auf Multiple Public Volume Sets (MPVS) liegen.

Diese Dateien werden weiterhin von UDS/SQL mit der Zugriffsmethode UPAM bearbeitet. Der linked-in DBH verwendet nur die Zugriffsmethode UPAM.

Der Zugriff mit FASTPAM ist performanter als der Zugriff mit UPAM, weil bei FASTPAM bestimmte Pfade vorbereitet und Bereiche resident angelegt werden.

Um Bereiche resident anlegen zu können, braucht die BS2000-Benutzerkennung eine FASTPAM-Berechtigung. Ob die BS2000-Benutzerkennung eine entsprechende Berechtigung besitzt, erfahren Sie über das BS2000-Kommando SHOW-USER-ATTRIBUTES. Das Feld DMS-TUNING-RESOURCES muss den Wert *EXCLUSIVE haben.

FASTPAM kann zwar benutzt werden, ohne dass die BS2000-Benutzerkennung die FASTPAM-Berechtigung besitzt, aber die Bereiche werden dann nicht resident gehalten. Dies gilt ebenfalls, wenn die BS2000-Benutzerkennung eine FASTPAM-Berechtigung besitzt, aber wegen zu wenig freiem Hauptspeicher nicht alle I/O-Bereiche resident gehalten werden können.

Wenn das Laden von UDS/SQL trotz eines ausreichenden Wertes für RESIDENT PAGES in der BS2000-Benutzerkennung abgewiesen wird, muss der BS2000-Systemverwalter eventuell die maximale erlaubte Anzahl an residenten Hauptspeicherseiten entsprechend erhöhen (BS2000-Kommando MODIFY-SYSTEM-BIAS). Den eingestellten Maximalwert (CORE) können Sie mit dem BS2000-Kommando SHOW-SYSTEM-STATUS über den Parameter INFORMATION=*SYSTEM-PARAMETERS abfragen.

Die benötigte Anzahl residenter Seiten geben Sie beim Start des DBH (START-UDS-DBH bzw. START-EXECUTABLE-PROGRAM-Kommando) über den Operanden RESIDENT-PAGES an. Sie müssen einen Minimalwert und einen Maximalwert angeben.

Wenn Sie den Minimalwert = 0 angeben, läuft UDS/SQL auf jedem Fall an, wenn auch gegebenenfalls mit Performance-Einbußen. Wenn Sie den Minimalwert > 0 angeben, läuft UDS/SQL nur an, wenn die Minimal-Anforderung befriedigt werden kann.

Für die Berechnung des Maximalwertes sind insbesondere die Größe der 4-Kbyte und 8-Kbyte Buffer Pools relevant. Bei einer Zahl von n Datenbanken erhält man den ungefähren Wert

$$300 \text{ Kbyte} + n \cdot 100 \text{ Kbyte} + 1,1 \cdot (\text{Summe der Größen der 4-Kbyte und 8-Kbyte Buffer Pools})$$

Da der 4-Kbyte System Buffer Pool mindestens 1,3 Mbyte groß ist, ergibt sich (bei einer zugeschalteten Datenbank) ein Mindestbedarf von 1,83 Mbyte = 458 residente 4-Kbyte-Seiten.

Wenn der Returncode UDS0600 mit dem Insert USER LIMIT EXCEEDED beim DBH-Start ausgegeben wird, dann hat der DBH nicht genügend residente Seiten erhalten. Die Ursache dafür kann sein, dass der beim Start angegebene Maximalwert für RESIDENT-PAGES zu klein war, aber auch dass das System nicht genügend Seiten zur Verfügung stellen konnte.

Weitere Informationen zur Zugriffsmethode FASTPAM finden Sie im Handbuch „[Einführung in das DVS](#)“.

9.8 Vollen Adressraum für UDS/SQL nutzen

Für UDS/SQL kann der volle zur Verfügung stehende Adressraum (XS) genutzt werden.

Die UDS/SQL-Tasks des independent DBH laufen grundsätzlich im Adressierungsmodus (AMODE) 31 ab.

UDS/SQL-Coding in Anwendertasks (z.B. UDS/SQL-Verbindung) und im linked-in DBH wird im AMODE des Anwendungsprogramms ausgeführt, das den DML-Auftrag abgeschickt hat. Im linked-in DBH darf der AMODE in einem Anwendungsprogrammablauf für die DML-Aufträge nicht dynamisch geändert werden.

Anwender-Coding, das direkt während der DML-Bearbeitung angesprungen wird (z.B. Fehlerbehandlungsroutine DSCEXT der CALL-DML), muss im AMODE des DML-Auftrages ablauffähig sein.

Es empfiehlt sich, grundsätzlich die Anwenderprogramme XS-fähig zu machen und im AMODE 31 ablaufen zu lassen. Dadurch können Sie Einschränkungen des Adressraums bei komplexem und weiter wachsendem Funktionsumfang der Anwendungen vermeiden.

9.9 Zeitermittlung über das Subsystem GET-TIME

UDS/SQL nutzt das BS2000-Subsystem GET-TIME zur Ermittlung von Zeitinformationen.

GET-TIME muss während der BS2000-Systemeinleitung gestartet werden, um eine UDS/SQL-Konfiguration starten zu können. Die GET-TIME-Parameter für Zeitzone, Zeitsprung, Sommer- und Winterzeit sowie deren Umstelldaten müssen korrekt versorgt werden, damit eine UDS/SQL-Konfiguration fehlerfrei abgewickelt werden kann. Die Umstelldaten müssen für die Vergangenheit so weit zurückreichen, wie die UDS/SQL-Datenbestände, die während der BS2000-Session bearbeitet werden sollen.

Falls UDS/SQL-Coding im Adressierungsmodus AMODE 24 durchlaufen wird, muss der BS2000-Systemverwalter darauf achten, dass das GET-TIME Subsystem unterhalb der 16 Mbyte-Grenze geladen wird. Hierfür wird eine besondere Ausprägung der Deklarationsdatei für das GET-TIME Subsystem verwendet.

Wenn UDS/SQL als DSSM-Subsystem eingesetzt wird, muss der BS2000-Systemverwalter bei der Generierung des UDS/SQL Subsystems die Abhängigkeit zum Subsystem GET-TIME in der SCCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES mit dem Parameter REFERENCED-SUBSYSTEM=GET-TIME explizit angeben. Siehe Handbuch „[Verwaltung von Subsystemen in BS2000](#)“.

9.10 Kennwörter für UDS/SQL-Dateien vergeben

UDS/SQL schützt die vom DBH automatisch eingerichteten Dateien SLF, ALOG-Datei, RLOG-Datei und Temporären Realms mit einem Kennwort.

Ein solches Kennwort wird in einigen Fällen nicht dazu benutzt, um den Inhalt der Dateien vor unberechtigtem Zugriff zu schützen, sondern um das versehentliche Löschen sessionübergreifend genutzter Dateien zu verhindern. Diese Dateien, die mit dem nicht änderbaren Kennwort C'UDS'BLANK" geschützt werden, enthalten nur DB-Verwaltungsdaten.

Die SLF wird immer mit dem Standard-Kennwort: C'UDS'BLANK" geschützt. Die ALOG-Datei und die temporären Dateien werden ebenfalls mit dem Standard-Kennwort C'UDS'BLANK" geschützt, falls im DBH Ladeparameter PP CATPASS nichts anderes angegeben worden ist.

Außerdem hat der Datenbankadministrator die Möglichkeit, UDS/SQL-Dateien selbst mit Kennwörtern zu belegen. Dazu benutzt er das BS2000-Kommando MODIFY-FILE-ATTRIBUTES (siehe die Kommando-Handbücher zu [BS2000 OSD/BC](#)). Eine Ausnahme bildet die RLOG-Datei. Jede RLOG-Datei wird immer mit einem spezifischen UDS/SQL-internen Kennwort geschützt.

Da die RLOG-Dateien im Allgemeinen vom DBH wieder gelöscht werden, wenn sie nicht mehr benötigt werden, müssen Sie das Kennwort nicht verwalten. Sofern aber in Ausnahmefällen das Löschen einer RLOG-Datei aus anderen Gründen nötig wird, können Sie dazu das Kennwort mit dem Dienstprogramm RLOGPASS der Liefereinheit UDS-SQL-T absetzen.

Die Kennwörter für UDS/SQL-Dateien müssen entweder mit dem Ladeparameter PP PASSWORD oder dem DAL-Kommando ADD PW bekannt gegeben werden.

Wenn Sie Kennwörter für UDS/SQL-Dateien vergeben, sollten Sie folgende Punkte beachten:

- Für alle UDS/SQL-Dateien sollte zumindest ein Lese-Kennwort vergeben werden, um den Schutz gegenüber UDS/SQL-fremden Zugriffssystemen zu gewährleisten.
- Ein Schreib-Kennwort ist nur dann sinnvoll, wenn UDS/SQL-fremder Lesezugriff gezielt zugelassen werden soll, ohne den Schreibschutz zu verlieren.
- Die Kennwort-Vergabe sollte ökonomisch sein, nicht jede Datei braucht ein eigenes Kennwort.
- Das Kennwort C'UDS'BLANK" und über PP CATPASS, PP PASSWORD bzw. das DAL-Kommando ADD PW angegebene Lese-Kennwörter sollten nicht für andere Dateien verwendet werden, da UDS diese Kennwörter intern absetzt und somit in Sondersituationen ein Zugriff auf Dateien möglich ist, für die dies aus Anwendersicht nicht erwünscht ist.

Die Datei COSSD nimmt eine Sonderstellung ein, weil sie für COBOL-Übersetzungen (COBOL85 oder COBOL2000) benötigt wird.

Wenn hier ein Schutzbedürfnis besteht, muss der Datenbankadministrator entweder

- dem Anwender eine Übersetzungsprozedur zur Verfügung stellen, die das COSSD-Kennwort enthält und selbst durch ein Lese- und ein Ausführungs-Kennwort geschützt ist. Dem Anwender darf dann nur das Ausführungs-Kennwort bekannt sein.

oder

- bei extremen Sicherheitsanforderungen muss der Datenbankadministrator Übersetzungen von COBOL-DML-Quellprogrammen im Auftrag des Anwenders selbst durchführen.

Wenn Sie Kennwörter angeben für die Datei, die die DBH-Ladeparameter enthält, ist darauf zu achten, dass UDS/SQL immer zugreifen kann. UDS/SQL muss auf folgende Dateien schreibend zugreifen:

SLF, temporäre Dateien und DB-Status-Datei.

Auf Realms und ALOG-Dateien muss UDS/SQL schreibend zugreifen können, außer wenn deren Datenbanken mit SHARED-RETRIEVAL betrieben werden.

Organisatorische Maßnahmen

Wenn Sie für UDS/SQL-Dateien Kennwörter vergeben, ist es sinnvoll, folgende organisatorische Maßnahmen zu treffen:

- Kennwortverschlüsselung durch das BS2000 bei der Systemgenerierung nutzen (siehe die Kommando-Handbücher zu [BS2000 OSD/BC](#), ADD-PASSWORD-Kommando).
- LOGON-Kennwort für die Konfigurationskennung vergeben.
- Dateien, die Kennwörter enthalten z.B. CALL-PROCEDURE-Prozedurdateien, ENTER-Dateien, Dateien mit DBH-Ladeparametern
 - mit Kennwörtern schützen
 - nur löschen mit DESTROY-Angabe im DELETE-FILE-Kommando.

i Es muss bei der Systemgenerierung dafür gesorgt werden, dass UDS/SQL nicht wegen der Kennwort-Konventionen abnormal beendet wird (siehe Handbuch „[Einführung in die Systembetreuung](#)“ CL2-Option):

Systemparameter PWACTIVE:

Der Wert muss mindestens um 1 größer sein als die Anzahl der Kennwörter, die durch PP PASSWORD und ADD PW gleichzeitig angeschlossen sein können.

Systemparameter PWENTERD:

Der Wert muss um ein Vielfaches größer sein als die Anzahl der Kennwörter. Werden während einer Session oft Realms oder Datenbanken angeschlossen oder ausgeschlossen, wird der Maximalwert empfohlen (Maximalwert= Standard).

UDS/SQL setzt vor jedem Anschließen oder Ausschließen von Dateien alle bekannten Kennwörter ab oder nimmt sie zurück.

Systemparameter PWERRORS:

Der Wert sollte um ein Vielfaches größer sein als die Anzahl von UDS/SQL-Dateien, die durch dasselbe Kennwort geschützt sind. Andernfalls können bereits wenige Versehen bei der Kennwort-Eingabe bewirken, dass UDS/SQL-Tasks vom BS2000 abnormal beendet werden (Maximalwert=Standard).

9.11 Accounting im UDS/SQL-openUTM-Betrieb

UDS/SQL unterstützt die Accounting-Schnittstelle, die openUTM den Datenbanken anbietet. Es werden CPU-Verbrauch und Anzahl I/Os pro openUTM-Auftrag erfasst, solange die openUTM-Anwendung Accounting-Informationen anfordert.

Die Accounting-Erfassung verbraucht selbst CPU-Zeit. Die Erfassungsphase sollte daher den Erfordernissen der Anwendung angepasst werden. Einzelne Verbrauchswerte sind stark abhängig vom aktuellen Zustand der UDS/SQL-Session.

UDS/SQL führt Teilaufgaben auch asynchron aus, sodass die an openUTM übergebenen Werte in der Summe nicht den gesamten Verbrauch umfassen.

Der Verbrauch für Mastertask und Administratortask wird nicht berücksichtigt. Auch bleiben Initialisierungs- und Terminierungsphase der UDS/SQL-Session unberücksichtigt.

Die Anzahl I/Os wird am Ende einer openUTM-Transaktion ermittelt. I/Os, die erst nach Ende einer openUTM-Transaktion anfallen, werden nicht mehr gezählt.

Bei UDS-D wird nur die lokale Konfiguration erfasst. Ein CPU-Verbrauchswert wird von UDS-D demnach nicht über Rechengrenzen hinaus aufsummiert.

9.12 UDS/SQL-Benutzer und -Benutzergruppen

In UDS/SQL-openUTM-Anwendungen wird die korrekte Identifizierung des Benutzers durch openUTM vorgenommen. openUTM reicht die entsprechenden Informationen (openUTM-Benutzerkennung, openUTM-Hostname, openUTM-Anwendungsname, KSET-Name) an UDS/SQL weiter. UDS/SQL bildet aus diesen Informationen den UDS/SQL-Benutzernamen und den UDS/SQL-Benutzergruppennamen nach den Formeln:

```
UDS/SQL-Benutzergruppenname = openUTM-Hostname +
                             openUTM-Anwendungsname +
                             KSET-Name

UDS/SQL-Benutzername       = openUTM-Hostname +
                             openUTM-Anwendungsname +
                             openUTM-Benutzerkennung
```

Benutzername und Benutzergruppenname werden intern teilweise mit Unterstrich ergänzt.

Der UDS/SQL-Benutzergruppenname wird für die Rechteprüfung genutzt, der UDS/SQL-Benutzername ist in dieser Version ohne Bedeutung. In UDS/SQL-openUTM-Anwendungen erübrigt sich die Zuordnung der UDS/SQL-Benutzer zu UDS/SQL-Benutzergruppen. Wenn die Zuordnung von Benutzern zu gegebenen Benutzergruppen sich ändern sollte, müssen die Änderungen nur noch vom openUTM-Administrator durchgeführt werden.

Mit dem Ladeparameter PP PRIVACY-CHECK kann der UDS/SQL-Datenbankadministrator die Privacy-Prüfung verändern oder ausschalten (siehe "[Behandlung der Privacy-Prüfung steuern \(PRIVACY-CHECK\)](#)").

Um die gemeinsame Organisation von UDS/SQL und openUTM besser zu beleuchten, wird anschließend das openUTM-Konzept für Zugriffsrechte kurz erläutert.

9.12.1 Zugriffskontrolle über openUTM

In openUTM-Anwendungen realisieren Teilprogramme die Services, die von Terminal-Benutzern oder Client-Programmen angefordert werden. Diese Teilprogramme enthalten die openUTM-Aufrufe sowie ggf. Datenbankzugriffe. Innerhalb einer Anwendung können die Zugriffsrechte differenziert festgelegt werden.

openUTM bietet dazu zwei Zugriffskontrollverfahren, die gleiche Differenzierungsmöglichkeiten bieten, sich aber in der Sichtweise auf die Objekte der Anwendung unterscheiden:

- das benutzerorientierte Lock-/Keycode-Konzept und
- das rollenorientierte Access-List-Konzept

Lock-/Keycode-Konzept

Mit dem Lock-/Keycode-Konzept können Sie erreichen, dass nur besonders autorisierte Benutzer oder Client-Programme bestimmte Services der UTM-Anwendung verwenden dürfen. Sie können auch vereinbaren, dass die Anmeldung unter einer Benutzerkennung nur über bestimmte LTERM-Partner (Anschlusspunkte) möglich ist oder dass bestimmte Services nur über spezielle LTERM-Partner gestartet werden können.

Die zu schützenden Objekte - das sind zum Beispiel LTERM-Partner und den Services zugeordnete Transaktionscodes - können mit einem Lockcode versehen werden.

Für Benutzerkennungen und LTERM-Partner werden Keycodes definiert. Wenn ein Keycode mit dem Lockcode eines gesicherten Objekts übereinstimmt, ist der Zugriff auf dieses Objekt erlaubt.

In der Regel hat eine Benutzerkennung oder ein LTERM-Partner Zugriff auf mehrere Services und verfügt deshalb über mehrere Keycodes. Die einzelnen Keycodes sind daher jeweils zu Keysets zusammengefasst.

Das Lock-/Keycode-Konzept hat folgende Effekte:

- Die Anmeldung eines Terminals oder eines Client-Programms ist nur möglich, wenn der angegebenen Benutzerkennung ein Keycode zugeordnet ist, der mit dem Lockcode des zugeordneten LTERM-Partners übereinstimmt.
- Ein Terminal-Benutzer oder ein Client-Programm kann einen Service nur dann aufrufen, wenn **sowohl** das Keyset der jeweiligen Benutzerkennung **als auch** das des LTERM-Partners einen Keycode enthalten, der mit dem Lockcode des Transaktionscodes übereinstimmt.

Access-List-Konzept

Bei der Nutzung des Access-List-Konzepts werden Benutzer nach Rollen oder Funktionen im Unternehmen zusammengefasst (Pförtner, Sachbearbeiter, Personalbearbeiter, Abteilungsleiter, Administrator, Controller, Geschäftsführer, ...), wobei ein Benutzer natürlich mehrere Rollen haben kann. Jede Rolle wird auf einen Keycode abgebildet.

- Jedem Benutzer einer UTM-Anwendung ordnet der Administrator eine oder mehrere Rollen zu (z.B. Sachbearbeiter, Abteilungsleiter, ...).
- Für die zu schützenden Objekte - Services und TAC-Queues - wird dann anhand einer Access-List festgelegt, welche Benutzergruppen (Sachbearbeiter, Controller...) Zugriff haben.
- Haben Sie zum Beispiel *Personalbearbeiter* als Rolle 2 definiert, und *Geschäftsführer* als Rolle 1, so können Sie festlegen, dass nur diese Benutzergruppen Zugriff auf den Service *Persona*/haben sollen, indem Sie dem Service eine Access-List zuweisen, die die Codes 1 und 2 enthält.
- Den betreffenden Benutzern wiederum wird ein Keyset zugewiesen, das alle Rollen (Keycodes) enthält, die der Benutzer wahrnehmen darf.

Wollen Sie außerdem festlegen, dass der Zugriff auf sicherheitsrelevante Daten nur über bestimmte LTERM-Partner erfolgen darf, weisen Sie den LTERM-Partnern ebenfalls geeignete Keysets zu.

Der Zugriff auf einen Service setzt dann voraus, dass sowohl für den Benutzer als auch für den LTERM-Partner, über den der Benutzer angemeldet ist, mindestens jeweils eine Rolle definiert ist, die in der Access-List des Services enthalten ist.

Auswirkung auf UDS/SQL

Unabhängig von dem verwendeten Zugriffskontrollverfahren bilden alle openUTM-Benutzer, die den gleichen Keyset (KSET) bei openUTM haben, eine Benutzergruppe bezüglich der Datenbank-Rechte. Für jede dieser Benutzergruppen, die mit der Datenbank über UTM-Transaktionscodes arbeiten, müssen die Rechte in den UDS/SQL Datenbanken mit dem Dienstprogramm BPRIVACY definiert werden.

Weitere Informationen zu den Zugriffskontrollen in openUTM entnehmen Sie den Handbüchern zu openUTM.

9.12.2 UDS/SQL-Benutzergruppen definieren und Zugriffsrechte vergeben

Bevor der openUTM-Administrator eine UDS/SQL-openUTM-Anwendung generiert, muss der UDS/SQL-Datenbankadministrator die UDS/SQL-Benutzergruppe und deren Rechte definieren. Dies geschieht mit der ADD-USER-GROUP-Anweisung des Dienstprogramms BPRIVACY.

Beispiel

Auf die Datenbank VERSAND sollen drei Benutzergruppen Zugriff haben.
Die „Benutzergruppe 1“ darf openUTM-Administrationsaufgaben ausführen.
Die „Benutzergruppe 2“ darf schreibend und lesend auf die Datenbank zugreifen.
Die „Benutzergruppe 3“ darf lesend auf die Datenbank zugreifen.

Der openUTM-Administrator generiert die UDS/SQL-openUTM-Anwendung „UTMAWVER“ auf dem Rechner „UTMHOST1“ und macht folgende Vereinbarungen für die Benutzergruppen:

Benutzergruppe 1

„UTMUSER1“ mit dem Kennwort „KENNW001“ darf die Transaktionscodes aufrufen, die die Administrationskommandos definieren (Voraussetzung: das Administrationsprogramm KDCADM muss mit KDCDEF generiert sein). Diesem Benutzer ist das KSET „KSETADM“ zugeordnet. In diesen Transaktionscodes erfolgt kein Zugriff auf die Datenbank.

```
TAC KDCAPPL,ADMIN=YES,PROGRAM=KDCADM,LOCK=1
. . .
    weitere Transaktionscodes für die Administration
. . .
TAC KDCUSER,ADMIN=YES,PROGRAM=KDCADM,LOCK=1
USER UTMUSER1,PASS=(C'KENNW001',DARK),KSET=KSETADM,PERMIT=ADMIN,STATUS=ON
KSET KSETADM,KEYS=1
```

Benutzergruppe 2

„UTMUSER2“ und „UTMUSER3“ mit den Kennwörtern „KENNW002“ bzw. „KENNW003“ dürfen die Transaktionscodes „UPDATE“, „NEUAUF“ und „SUCHEN“ aufrufen. Diesen Benutzern ist das KSET „KSETUPD“ zugeordnet.

```
TAC UPDATE,ADMIN=NO,LOCK=2,PROGRAM=AENDERN
TAC NEUAUF,ADMIN=NO,LOCK=3,PROGRAM=AUFNAHME
TAC SUCHEN,ADMIN=NO,LOCK=4,PROGRAM=FRAGEN
USER UTMUSER2,PASS=(C'KENNW002',DARK),KSET=KSETUPD,STATUS=ON
USER UTMUSER3,PASS=(C'KENNW003',DARK),KSET=KSETUPD,STATUS=ON
KSET KSETUPD,KEYS={2,3,4}
```

Benutzergruppe 3

„UTMUSER4“, „UTMUSER5“ und „UTMUSER6“ mit den Kennwörtern „KENNW004“, „KENNW005“ bzw. „KENNW006“ dürfen den Transaktionscode „SUCHEN“ aufrufen. Diesen Benutzern ist das KSET „KSETRTR“ zugeordnet.

```
TAC SUCHEN,ADMIN=NO,LOCK=4,PROGRAM=FRAGEN
USER UTMUSER4,PASS=(C'KENNW004',DARK),KSET=KSETRTR,STATUS=ON
USER UTMUSER5,PASS=(C'KENNW005',DARK),KSET=KSETRTR,STATUS=ON
USER UTMUSER6,PASS=(C'KENNW006',DARK),KSET=KSETRTR,STATUS=ON
KSET KSETRTR,KEYS=4
```

Wenn die oben genannten Benutzer Datenbankaufrufe absetzen, ergeben sich folgende UDS/SQL-Benutzergruppennamen:

```
UTMHOST1UTMAWVERKSETUPD_
UTMHOST1UTMAWVERKSETRTR_
```

Damit die Aufträge durchgeführt werden können, muss der UDS/SQL-Datenbankadministrator diese UDS/SQL-Benutzergruppen und deren Rechte **vorher** wie im Folgenden definiert haben:

```
/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=VERSAND.DBDIR
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,    VERSION=02.9B00
/START-UDS-BPRIVACY

...
***** START          BPRIVACY          (UDS/SQL V2.9 0000 )      2017-06-28  11:26:28
% UDS0215 UDS STARTET UDS/SQL V2.9 (LINKED-IN), DATE=2017-06-28 (ILL2038,11:26:28
/OYA2)
% UDS0746 UDS-PUBSET-DEKLARATION (CURRENT) FOLGT (ILL1746,11:26:28/OYA2)
OYA2: UDS-PUBSET-JV:   :SQL2:$XXXXXXXXX.PUBSDECL.ALL
OYA2: PUBSETS:        *
OYA2: DEFAULT PUBSET: SQL2
OYA2: -----
% UDS0722 UDS AUFTRAG ADD RLOG 150628092628 WIRD AUSGEFUEHRT (ILL1283,11:26:28
/OYA2)
% UDS0356 UDS DURCHFUEHRUNG DER AUFTRAEGE FUER VERSAND TERMINATED (ILL1309,11:
26:28/OYA2)
//ADD-USER-GROUP USER-GROUP-NAME=*KSET-FORMAT (HOST=UTMHOST1,APPLICATION=UTMAWVER,
KSET=KSETADM),
  OBJECT=( *REALM(NAME=*ALL,RIGHT=ALL), *RECORD(NAME=*ALL,RIGHT=ALL), *SET(NAME=*ALL,
RIGHT=ALL) )
//ADD-USER-GROUP USER-GROUP-NAME=*KSET-FORMAT (HOST=UTMHOST1,APPLICATION=UTMAWVER,
KSET=KSETUPD),
  OBJECT=( *REALM(NAME=*ALL,RIGHT=ALL), *RECORD(NAME=*ALL,RIGHT=ALL), *SET
(NAME=*ALL,RIGHT=ALL) )
//ADD-USER-GROUP USER-GROUP-NAME=*KSET-FORMAT (HOST=UTMHOST1,APPLICATION=UTMAWVER,
KSET=KSETRTR),
  OBJECT=( *REALM(NAME=*ALL,RIGHT=RETRIEVAL), *RECORD(NAME=*ALL,RIGHT=RETRIEVAL),
*SET(NAME=*ALL,RIGHT=RETRIEVAL) )
//END
% UDS0758 ANZAHL DER DML-ANWEISUNGEN UND I/O-ZAEHLER PRO DATENBANK (ILL1758,11:
26:28/OYA2)
OYA2: DATABASE NAME          DMLS    LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
OYA2: -----
OYA2: VERSAND                13      112      59         42         20
% UDS0213 UDS NORMAL BEENDET MIT *****13 DML-STATEMENTS 2017-06-28
(ILLY033,11:26:28/OYA2)

***** DIAGNOSTIC SUMMARY OF BPRIVACY
```

NO WARNINGS
NO ERRORS
NO SYSTEM-ERRORS

***** END OF DIAGNOSTIC SUMMARY

***** NORMAL END BPRIVACY (UDS/SQL V2.9 0000) 2017-06-28 11:26:28

Für weitere Informationen siehe Ladeparameter PP PRIVACY-CHECK auf "[Behandlung der Privacy-Prüfung steuern \(PRIVACY-CHECK\)](#)" und Handbuch „[Aufbauen und Umstrukturieren](#)“, BPRIVACY.

9.13 ACS (Alias Catalog Service)

Bei der Nutzung von ACS in Datenbank-Tasks sollten Aliasnamen grundsätzlich keine Katalog- oder Benutzerkennung enthalten. Für die eigentlichen Datenbankdateien ist die Nutzung von ACS nicht erlaubt.

ACS ist ausschließlich innerhalb einer Task wirksam. Dadurch kann mit ACS in verschiedenen Tasks der gleiche Aliasname für verschiedene Dateien definiert werden.

Beispiel

Task	Aliasname	Dateiname
TASK1	ALIAS	DATEI-1
TASK2	ALIAS	DATEI-2
TASK3	ALIAS	DATEI-3

Da UDS/SQL taskübergreifend arbeitet, könnte dies zu Inkonsistenzen in UDS/SQL-Datenbanken führen.

UDS/SQL prüft beim Eröffnen jeder Datenbankdatei soweit möglich, ob es sich bei dem entsprechenden Dateinamen um einen Aliasnamen handelt. Falls ja, wird eine Fehlermeldung ausgegeben und die entsprechende Aktion abgebrochen. Insbesondere bei der Verwendung von Aliasnamen mit Katalog- und/oder Benutzerkennung (COMPLETE-ALIAS-NAMES=*ALLOWED) kann diese Prüfung nicht vollständig garantiert werden.

9.14 Unicode-Konzept in UDS/SQL

Dem Unicode-Zeichensatz kommt im Zuge der zunehmenden Internationalisierung auch in BS2000 und seinen Anwendungen eine entscheidende Bedeutung zu. Nähere Informationen dazu finden Sie im Handbuch „[Unicode im BS2000](#)“.

Unicode fasst alle weltweit bekannten Textzeichen in einem einzigen Zeichensatz zusammen. Zudem ist Unicode unabhängig von unterschiedlichen Herstellern, Systemen und Ländern.

Die Unicode-Unterstützung in BS2000 ist eingebettet in das bereits vorhandene Konzept der codierten Zeichensätze (Coded Character Set, CCS), siehe Handbuch „[XHCS \(BS2000\)](#)“.

Voraussetzung für die Verwendung von Unicode ist eine dementsprechende BS2000-Systemumgebung, wobei derzeit für UDS/SQL die Produkte COBOL2000 und CRTE relevant sind, siehe die Freigabemitteilung zu UDS/SQL.

Zur Unterstützung von Unicode gibt es den Datentyp NATIONAL. Dieser entspricht in COBOL den nationalen Feldern (PICTURE N(n), USAGE IS NATIONAL). COBOL und UDS/SQL unterstützen die Unicode-Variante UTF-16.

Näheres dazu finden Sie im Handbuch „[Entwerfen und Definieren](#)“ und in den Handbüchern zu „[COBOL2000 \(BS2000\)](#)“.

9.14.1 Einsatz von Unicode (UTF-16) in bestehenden UDS/SQL-Datenbanken

Für die (teilweise) Umstellung von UDS/SQL-Datenbanken auf Unicode (UTF-16) haben Sie u.a. folgende Möglichkeiten:

- Hinzufügen nationaler Felder
- Hinzufügen einer neuen Satzart
- Ersetzen bestehender Felder

Hinzufügen nationaler Felder

Um nationale Felder hinzuzufügen, können Sie folgendermaßen vorgehen:

- Hinzufügen der Datenfelder in der Schema- und Subschema-DDL (siehe Handbuch „[Entwerfen und Definieren](#)“). Definition nationaler Felder (Unicode, PICTURE N(n) USAGE IS NATIONAL) mit derselben Anzahl an Zeichen wie bestehende alphanumerische Felder in derselben Satzart. Beachten Sie bitte, dass sich die Länge in Byte dabei verdoppelt, da ein Unicode-Zeichen in UDS/SQL zwei Byte belegt. Zusätzlich können Sie auch Indikatorfelder zur Kennzeichnung der Relevanz der neuen Felder einfügen.
- Umstrukturieren der Datenbank mit BCHANGE/BALTER (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“). BALTER initialisiert das neue Feld mit (nationalen) Leerzeichen.
- In den neu zu speichernden Sätzen versorgen Sie das neue Feld mit nationalen Daten. Das alte alphanumerische Feld können Sie als nicht relevant oder leer kennzeichnen oder es zusätzlich mit den bisherigen alphanumerischen Daten versorgen.

Hinzufügen einer neuen Satzart

Anstatt nationale Felder in bestehenden Satzarten hinzuzufügen, können Sie auch folgende Vorgehensweise wählen:

- Definition einer neuen Satzart
- Definitionen der neuen zu einer bestehenden Satzart gehörenden nationalen Felder in der **neuen** Satzart
- Verknüpfung der neuen Satzart als Membersatzart mit der bestehenden Satzart.

Die Vorteile dieser Vorgehensweise gegenüber dem Hinzufügen neuer Felder sind:

- Bestehende Satzarten sind nicht betroffen und müssen daher nicht geändert werden
- Das Umstrukturieren der Datenbank mit BCHANGE/BALTER geht sehr schnell

Nachteil ist:

- Es sind zusätzliche DML-Anweisungen für die Member erforderlich. Dadurch erhöht sich der Programmieraufwand und verschlechtert sich die Performance der Anwendung.

Ersetzung bestehender Felder

Um bestehende Felder durch nationale Felder zu ersetzen, können Sie folgende Schritte durchführen:

- Änderung der betroffenen Datenfelder in der Schema- und Subschema-DDL (siehe Handbuch „[Entwerfen und Definieren](#)“). Ersetzen eines bestehenden alphanumerischen Feldes durch ein nationales Feld (Unicode, PICTURE N(n)

USAGE IS NATIONAL) mit derselben Anzahl an Zeichen.

Beachten Sie bitte, dass sich die Länge in Byte dabei verdoppelt, da ein Unicode-Zeichen in UDS/SQL zwei Byte belegt.

- Entladen der betroffenen Satzarten mit BOUTLOAD (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“)
- Umstrukturieren der Datenbank mit BCHANGE/BALTER (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“)
- Ggf. Umsetzen der BOUTLOAD-Ausgabedateien von EBCDIC nach Unicode mit BS2000-Produkten mit XHCS-Anschluss (z.B. PERCON oder EDT)
- Laden der entladenen Satzarten mit BINILOAD (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“)

10 Der SQL-Vorgang

In einer SQL-Anwendung werden SQL-spezifische Verwaltungsdaten über Transaktionsgrenzen hinweg aufbewahrt. Eine solche Verwaltungseinheit wird als Vorgang bezeichnet.

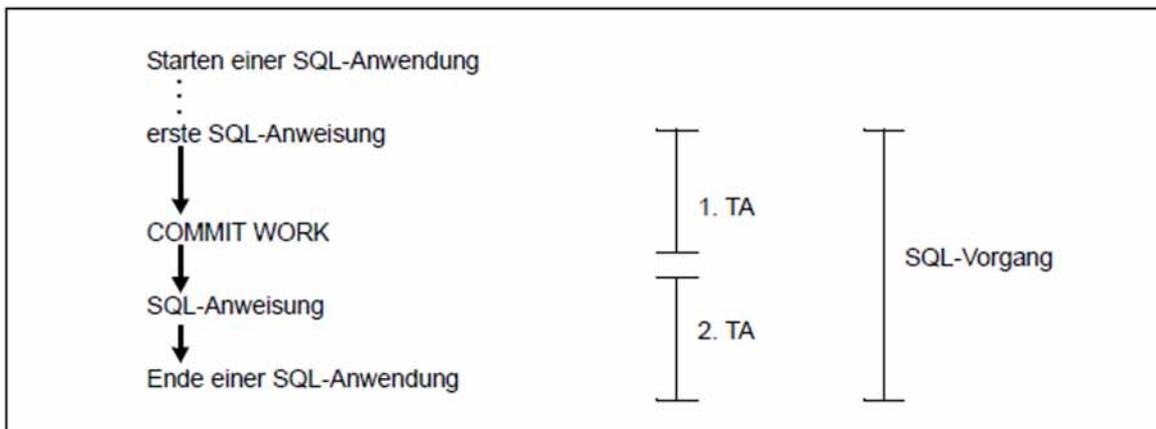


Bild 10: SQL-Vorgang

Der SQL-Vorgang kann verschiedene Zustände haben:

- aktiver Vorgang;
es ist eine Transaktion der SQL-Anwendung offen
- inaktiver Vorgang;
es ist gerade keine SQL-Transaktion offen.

Der DBH merkt sich während eines Vorgangs die Deklarationen von DECLARE CURSOR und CREATE TEMPORARY VIEW.

Es kann vorkommen, dass ein sehr lange nicht aktivierter Vorgang vom DBH oder vom Datenbankadministrator beendet wird. In diesem Fall wird bei der nächsten SQL-Anweisung ein neuer Vorgang eröffnet, wobei das SQL-Laufzeitsystem vorgangsrelevante Daten wieder an den DBH übergibt.

Siehe auch „DBH Ladeparameter PP SQL“ und „PP SQL-LIMIT“ auf ["Anzahl gleichzeitig aktiver SQL-Vorgänge festlegen \(SQL\)"](#), sowie das „DAL-Kommando DISPLAY SQL“ auf ["Informationen über einen SQL-Vorgang ausgeben \(DISPLAY SQL\)"](#).

11 DB-Betriebs- und Statuswerte ausgeben mit UDSMON

Der UDS/SQL-Monitor UDSMON gibt Werte zum Datenbankbetrieb mit UDS/SQL und UDS-D sowie zum Status von Transaktionen aus.

UDSMON arbeitet nur mit dem independent DBH.

Sie können sich folgende Informationen ausgeben lassen:

- momentane Leistung des Datenbankbetriebs
 - Anzahl der Transaktionen pro Session und Intervall
 - Anzahl der DML-Anweisungen pro Session und Intervall
 - Anzahl der SQL-Anweisungen pro Session und Intervall
 - durchschnittliche Transaktionszeiten pro Session und Intervall
 - Anzahl der Eingaben und Ausgaben pro Session und Intervall
- augenblicklicher Status der aktiven Transaktionen
- Ausnutzung systeminterner Optimierungswege
 - Inter Task Communication (ITC)
 - Probable Position Pointer (PPP)
- Ausnutzung systeminterner Betriebsmittel
 - Anzahl der offenen SQL-Vorgänge
 - Anzahl der aktiven Transaktionen
 - Anzahl der freien Servertasks
 - Anzahl und Größe der UDS/SQL Common Pools
- Einzelheiten über Datenbankbetrieb mit UDS-D
 - Kommunikation mit entfernten Partnerkonfigurationen
 - momentane Anforderungen an entfernte Konfigurationen
 - momentane Anforderungen von entfernten Konfigurationen
 - Verbindungen zu den entfernten Partnerkonfigurationen

Werte zum UDS-D-Betrieb erhalten Sie

- in UDS-D-spezifischen Feldern der COUNTER- und STATUS-Maske
- in der UDS-D-spezifischen Maske TRANSACTION
- in der UDS-D-spezifischen Maske CONNECT

Mit diesen Informationen können Sie Blockaden, z.B. durch „Langläufer“, sowie Abweichungen vom üblichen Systemverhalten frühzeitig erkennen.

Welche Werte ermittelt UDSMON nicht?

UDSMON gibt keine Werte aus, die ermittelt werden können mit Hilfe der Systemprogramme, wie z.B.

- SM2:
CPU-Auslastung, Eingabe- und Ausgabebelastung für Platten, Kanalbelastung, Paging-Rate
- COSMOS:
CPU-Zeiten, Wartezeiten, Ereignisse (Events)

mit Hilfe von UDS/SQL-Dienstprogrammen, wie z.B.

- BSTATUS

mit den Systemkommandos:

```
/SHOW-USER-STATUS ;  
/SHOW-FILE-ATTRIBUTES FILE-NAME=dateiname [ ,SELECT=*ALL]
```

11.1 Beschreibung der Funktionen

Um die Datenbankbetriebswerte zur Systemauslastung zu ermitteln, liest UDSMON in den internen UDS/SQL-Systemtabellen.

UDSMON gibt einerseits Gesamtzählerstände aus, d.h. die Summe der Ereignisse, die während der Session stattgefunden haben. Andererseits gibt UDSMON die „Rate“ aus, d.h. die Anzahl der Ereignisse, die innerhalb eines Zeitintervalls stattgefunden haben.

Beispiel

Sie wollen die DML-„Rate“ während des Datenbankbetriebs verfolgen und geben als Zeitintervall 10 Sekunden an. Wenn Sie UDSMON um 12 Uhr starten, können Sie folgende Werte erhalten:

Intervall	DML-Zähler	DML-Rate
12.00.00 Start	80	0
12.00.00 - 12.00.10	90	10
12.00.10 - 12.00.20	120	30
12.00.20 - 12.00.30	140	20
.	.	.
.	.	.

Um die Statuswerte der Transaktionen zu ermitteln, liest UDSMON in den UDS/SQL-Verwaltungstabellen.

Die Statuswerte, die ausgegeben werden, beziehen sich jeweils auf das Ende des vorgegebenen Zeitintervalls.

Ausgabemedien

Die ermittelten Werte werden ausgegeben

- über Datensichtstation: formatiert
- über Drucker: formatiert
- in eine Datei: nicht formatiert
- über openSM2/INSPECTOR: grafisch aufbereitet

Sie können die Zeitintervalle je Medium unabhängig voneinander vorgeben und ändern. Sie können jedes Ausgabemedium beliebig anschalten oder abschalten. Sie können damit z.B. veranlassen, dass der UDS/SQL-Monitor nur in der Zeit mitprotokolliert, wenn „Spitzenlastbetrieb“ herrscht.

Externer Zählerüberlauf

Bei der Konvertierung der internen Binärzähler in abdruckbare Dezimalwerte kann es vorkommen, dass das Ausgabefeld zu klein ist, um den Zähler aufzunehmen.

In diesem Fall wird der Zähler linksbündig in das Ausgabefeld übertragen und je nach der Anzahl der rechts abgeschnittenen Ziffern eine Anzeige in der Spalte unmittelbar vor dem Feld gesetzt.

- D für dekadisch, d.h. die erste Stelle von rechts fehlt
Die Anzeige ist mit 10 zu multiplizieren
- C für centesimal, d.h. rechts wurden zwei Stellen abgeschnitten
Die Anzeige ist mit 100 zu multiplizieren

- V für millesimal, d.h. rechts wurden drei Stellen abgeschnitten
Die Anzeige ist mit 1000 zu multiplizieren
- ? falls das Ausgabefeld um mehr als drei Stellen zu klein war oder ein Fehler bei der Konvertierung auftrat.

Interner Zählerüberlauf

Bei der Aufsummierung von Zählern aus Tabellen, die mehrfach existieren (z.B. pro Datenbank oder paralleler Transaktion), kann es zu Überläufen in den internen Summenfeldern kommen. Der Monitor meldet dies mit der Fehlermeldung:

```
UDS0534 UDSMON INTERNER UEBERLAUF IN ZAEHLER countername
```

Zusätzlich erscheint in sämtlichen Monitormasken:

```
ATTENTION!! - INTERNAL OVERFLOW IN COUNTER countername --> MESSAGE UDS0534
```

countername

Kurzbeschreibung des Maskenfeldnamens des betroffenen Zählers.

Die Meldung bleibt in den Masken stehen, bis Sie das nächste INFORM-PROGRAM- bzw. SEND-MSG-Kommando eingeben.

11.2 Systemumgebung

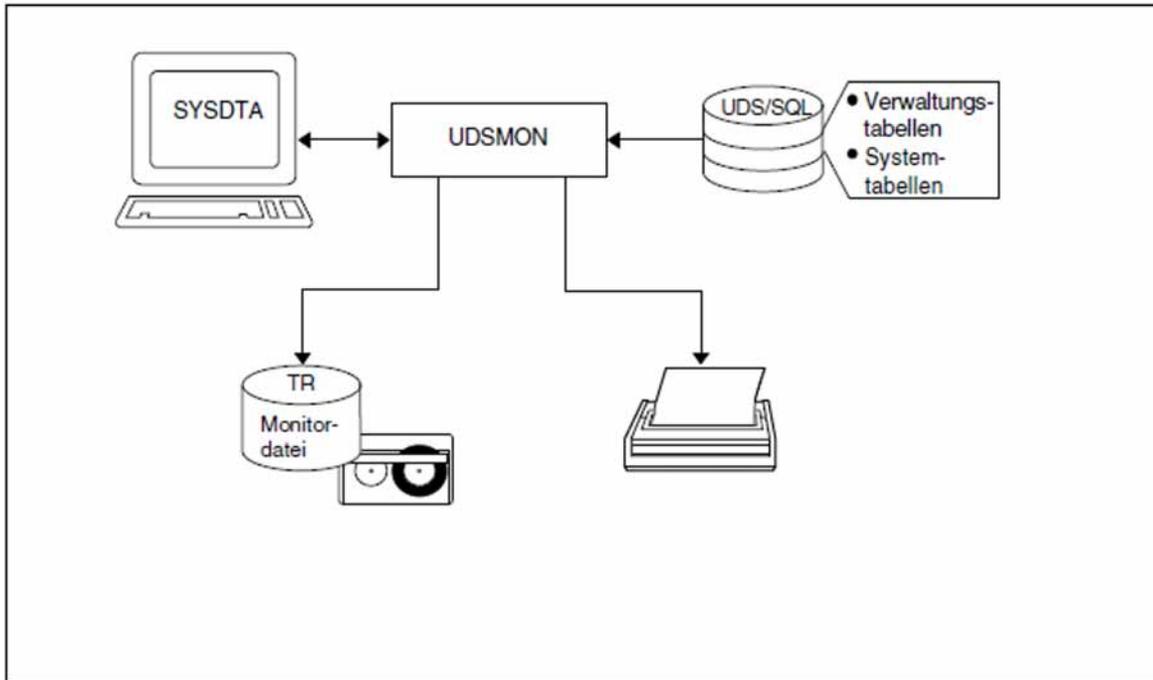


Bild 11: Systemumgebung von UDSMON

Der UDSMON ist ein eigenständiges, von UDS/SQL entkoppeltes Programm.

Starten des UDS/SQL-Monitors

UDSMON wird gestartet mit:

```
/START-UDS-MONITOR
```

Der UDS/SQL-Monitor muss in derselben Kennung geladen werden wie die Mastertask und die Servertask(s). Weitere Hinweise zum Starten von UDS-Dienstprogrammen finden Sie im Handbuch „[Aufbauen und Umstrukturieren](#)“.

UDSMON greift auf die UDS/SQL Verwaltungstabellen und Systemtabellen zu. Er arbeitet parallel zum Datenbankbetrieb und gibt die ermittelten Werte über Datensichtstation, über Drucker oder auf Datei aus. Bevor der Monitor mit dem Sammeln der Daten beginnt, geben Sie Ausgabegeräte, Masken und Zeitintervalle in Anweisungen an.

Wenn Sie während des UDSMON-Betriebs ein neues oder ein anderes Ausgabemedium angeben wollen, können Sie mit [EM],[DUE] unterbrechen und anschließend das entsprechende INFORM-PROGRAM-Kommando eingeben.

Einschränkung:

Wenn sich der UDS/SQL-Monitor an den Common Pool anhängt, wird die Version überprüft. Sind die Versionen von UDS/SQL und UDSMON verschieden, beendet der Monitor sich mit einer entsprechenden Meldung.

Beenden des UDS/SQL-Monitors

Den UDS/SQL-Monitor können Sie jederzeit beenden durch das Kommando

```
/INFORM-PROGRAM MSG='END' [ ,JOB-ID=*OWN]
```

oder er wird automatisch beendet, wenn die UDS/SQL-Session beendet wird, bzw. nach Ablauf der in der RUNTIME-Anweisung angegebenen Zeit.

Zuweisen der Monitordatei

Wenn Sie die Monitordatei explizit einrichten wollen, müssen Sie das entsprechende CREATE-FILE-Kommando geben:

```
/CREATE-FILE FILE-NAME=monitordatei
  [ ,SUPPORT=*PUBLIC-DISK (SPACE=*RELATIVE (PRIMARY-ALLOCATION=primär ,SECONDARY-
ALLOCATION=sekundär ) )
```

oder

```
  ,SUPPORT=*PRIVATE-DISK (VOLUME=priv-vsn,DEVICE-TYPE=gerät [ ,SPACE=... ] ) ]
/ADD-FILE-LINK LINK-NAME=TR,FILE-NAME=monitordatei,ACCESS-METHOD=*SAM
```

monitordatei

beliebiger Name der Monitordatei

TR Dateikettungsname der Monitordatei

Wenn Sie nichts zuweisen, legt UDSMON die Monitordatei automatisch auf gemeinschaftlicher Platte unter folgendem Namen an:

TR. *tsn*. *datum*

TR	festgelegter Dateiname (TR = TRACE)
<i>tsn</i>	zugewiesene Prozessfolgennummer (vierstellig)
<i>datum</i>	Tagesdatum in folgender Form: <i>jjjmmtt</i>

Eine bereits vorhandene Monitordatei wird fortgeschrieben.

11.3 Anweisungen und Kommandos für UDSMON

Es gibt UDSMON-Anweisungen zum Starten und UDSMON-Kommandos während des Betriebs.

UDSMON-Anweisungen zum Starten

Anweisung	Bedeutung
<u>CONFNAME</u> = <i>konfname</i>	Zuweisen der Konfiguration
<u>HELP</u>	Auflisten aller UDSMON-Anweisungen
<u>MASK</u> ={S C ALL CS DC DT D}	Zuweisen der Ausgabemaske(n) UDSMON gibt aus im eingestellten Zeitintervall bei: S: die STATUS-Maske C: die COUNTER-Maske ALL: alle Masken abwechselnd CS : COUNTER-Maske und STATUS-Maske abwechselnd DC: die CONNECT-Maske DT: die TRANSACTION-Maske D: die TRANSACTION-Maske und die CONNECT-Maske abwechselnd Standardwert : S
<u>MEDIUM</u> ={T, <i>n</i> L, <i>n</i> [, D] F, <i>n</i> [, D] S, <i>n</i> }	Zuweisen von Ausgabegerät und Zeitintervall T : Ausgabe auf Datensichtstation L : Ausgabe auf Drucker F : Ausgabe in eine Datei S : Übergabe an SM2 Standardwert : T im Dialogbetrieb L im Stapelbetrieb <i>n</i> : Zeitintervall für die Abfragefrequenz in Sekunden (<i>n</i> =5..999) Standardwert : 10 D : gibt an, dass der UDS-D-Betrieb ausgewertet werden soll. Bei Ausgabe in eine Datei wird ein zusätzlicher Kennungssatz und im eingestellten Zeitintervall ein zusätzlicher Datensatz geschrieben. Die Ausgabe auf Drucker wird um die Informationen der Maske DC (CONNECT) und DT (TRANSACTION) erweitert.
<u>MEDIUM</u> ={DEPRECATED EXPENSIVE MESSAGES DBCOUNTERS }	Ausgabe in Zwischendatei (vgl. entsprechendes Kommando während des Betriebes, siehe Tabelle 27) Das Kommando wird vorgemerkt und beim Start ausgeführt.
<u>REBASE-COUNTER</u> =NEW	Anfordern eines logischen Nullpunkts ausschließlich für die Sessionzähler der Countermaske

<u>RUNTIME</u> = <i>n</i>	Angabe nach welcher Zeit sich UDSMON automatisch beenden soll. Die Anweisung kann vor allem im Stapelbetrieb genutzt werden, z.B. um nach vorgegebener Zeit die Ausgaben auf Drucker zu erhalten. <i>n</i> : Laufzeit in Sekunden (<i>n</i> = 60..86400) Die Zeitangabe sollte größer sein als die Zeitintervalle der zugewiesenen Ausgabemedien.
<u>START</u>	UDSMON starten Das Sammeln der Daten beginnt.
<u>END</u>	UDSMON beenden Es werden keine Daten gesammelt.

Tabelle 26: UDSMON-Anweisungen zum Starten

UDSMON-Kommandos während des Betriebs

Den Betrieb von UDSMON können Sie unterbrechen, um die angegebenen Geräte, Masken oder Zeitintervalle auflisten zu lassen oder zu ändern.

Die Unterbrechung müssen Sie einleiten mit:

[EM], [DUE]

Anschließend können Sie das Kommando
INFORM-PROGRAM MSG = ... [, JOB-ID=*OWN'] eingeben.

Kommando	Bedeutung
INFORM-PROGRAM MSG= 'ADD MEDIUM= { T, <i>n</i> L, <i>n</i> [, D] F, <i>n</i> [, D] S, <i>n</i> }	Zuweisen von Ausgabegerät und Zeitintervall T: Ausgabe auf Datensichtstation L: Ausgabe auf Drucker F: Ausgabe in eine Datei S : Übergabe an SM2 Standardwert : T im Dialogbetrieb L im Stapelbetrieb <i>n</i> : Zeitintervall für die Abfragefrequenz in Sekunden (<i>n</i> =5..999) Standardwert : 10 D: gibt an, dass der UDS-D-Betrieb ausgewertet werden soll. Bei Ausgabe in eine Datei wird ein zusätzlicher Kennungssatz und im eingestellten Zeitintervall ein zusätzlicher Datensatz geschrieben. Die Ausgabe auf Drucker wird um die Informationen der Maske DC (CONNECT) und DT (TRANSACTION) erweitert.

INFORM-PROGRAM MSG= ' <u>DISPLAY</u> '	Auflisten der bisher eingestellten Geräte und Zeitintervalle
INFORM-PROGRAM MSG= ' <u>DISPLAY CP-SIZE</u> '	Auflisten der Anzahl und der Größe der Common Pools
INFORM-PROGRAM MSG= ' <u>DISPLAY DB-IO-COUNTER</u> '	Auflisten aller angefallenen Datenbank-I/Os seit DBH-Start sowie im letzten Zeitintervall. Die Datenbank-I/Os sind in Kategorien (2-Kbyte-, 4-Kbyte-, 8-Kbyte-, Exklusiv-I/Os) und in logische bzw. physische Lese-/Schreib-I/Os pro Kategorie unterteilt
INFORM-PROGRAM MSG= ' <u>DISPLAY DEPRECATED</u> '	Auflistung von Anwendungen mit Eigenschaften, die ein potentielles Risiko beinhalten (eingeschränkte Adressraumnutzung, veraltete Nachladetechnik, fehlende Subschemavalidierung) Die Auflistung wird in eine Zwischendatei geschrieben (siehe Abschnitt „DISPLAY-Ausgabe des UDS-Monitors in Zwischendateien“) und mit einem SHOW-FILE-Kommando vom Monitor angezeigt.
INFORM-PROGRAM MSG= ' <u>DISPLAY EXPENSIVE</u> '	Auflistung von Anwendungen mit der größten Ressourcennutzung bzgl. Transaktionsdauer, Anzahl der DML pro Transaktion, Anzahl der logischen Ein-/Ausgaben je DML und je Transaktion getrennt für UTM- und TIAM-Anwendungen Die Auflistung wird in eine Zwischendatei geschrieben (siehe Abschnitt „DISPLAY-Ausgabe des UDS-Monitors in Zwischendateien“) und mit einem SHOW-FILE-Kommando vom Monitor angezeigt.
INFORM-PROGRAM MSG= ' <u>DISPLAY MESSAGES</u> '	Aufbereitete Auflistung von gepufferten DBH-Meldungen Die Auflistung wird in eine Zwischendatei geschrieben (siehe Abschnitt „DISPLAY-Ausgabe des UDS-Monitors in Zwischendateien“) und mit einem SHOW-FILE-Kommando vom Monitor angezeigt. Bei aufeinanderfolgenden Kommandos in einem Monitorlauf mit expliziter Zuweisung der Zwischendatei werden Meldungen ergänzt, die seit dem vorigen DISPLAY MESSAGES Kommando neu hinzugekommen sind. Im Puffer überschriebene Meldungen, die somit nicht mehr aufgelistet werden können, werden mit einem gesonderten Hinweis ausgewiesen. Der Datumswechsel wird angezeigt.
INFORM-PROGRAM MSG= ' <u>DISPLAY DBCOUNTERS</u> '	Die Anzahl der DML und der Ein-/Ausgaben pro Datenbank ab dem Start des DBH und während die Datenbank online ist.
INFORM-PROGRAM MSG= ' <u>DISPLAY REX</u> '	Die Anzahl aller mit READY EXCLUSIVE oder READY PROTECTED eröffneten Transaktionen wird angegeben

<p>INFORM-PROGRAM MSG= 'DISPLAY SATURATION'</p>	<p>Auflistung der höchsten Sättigung von zentralen Systemressourcen in der bisherigen DBH-Session (z.B. Anzahl paralleler Transaktionen, Nutzung nicht erweiterbarer Memory-Pools)</p>
<p>INFORM-PROGRAM MSG= 'DISPLAY TASKS'</p>	<p>Auflisten aller mit UDS/SQLverbundenen Anwendertasks und der Anzahl der bisher in dieser Anwendertask bearbeiteten DMLs. Bei den bearbeiteten DMLs werden im Gegensatz zu vergleichbaren Werten der Countermaske aus Diagnosegründen auch Aufträge des COBOL-Laufzeitsystems mitgezählt, die in UTM-Anwendungen die korrekte Transaktionsbearbeitung bei Taskwechsel sicherstellen.</p> <p>Für asynchron beendete Anwendungen, z.B. aufgrund von Zeitüberschreitungen in UTM-Anwendungen, wird zusätzlich ausgegeben:</p> <ul style="list-style-type: none"> • ob diese Beendigung erfolgreich abgewickelt wurde und die UDS-internen Ressourcen wiederverwendbar sind (RELEASED) • ob die Beendigung noch nicht vollständig abgewickelt ist (PENDING) oder • ob wegen eines schwerwiegenden Fehlers die entsprechende UDS-interne Ressource auf Dauer gesperrt ist (BLOCKED).
<p>INFORM-PROGRAM MSG= 'END'</p>	<p>Beenden des Monitors</p>
<p>INFORM-PROGRAM MSG= 'FINISH MEDIUM= {T, n L, n[, D] F, n[, D] S, n}</p>	<p>Beenden der Ausgabe auf ein Ausgabegerät</p> <p>T : Ausgabe auf Datensichtstation L : Ausgabe auf Drucker F : Ausgabe in eine Datei S : Übergabe an SM2 D: Es wird nur die Ausgabe zum UDS-D-Betrieb beendet. Bei Ausgabe auf Drucker entfallen die Informationen der Maske DC (CONNECT) und DT (TRANSACTION). Bei Ausgabe in eine Datei entfällt der UDS-D-Datensatz.</p>
<p>INFORM-PROGRAM MSG= 'HELP'</p>	<p>Auflisten der Eingabemöglichkeiten</p>

<p>INFORM-PROGRAM</p> <p>MSG= ' <u>MASK</u>={S C ALL CS DC DT D}</p>	<p>Zuweisen der Ausgabemaske(n)</p> <p>UDSMON gibt aus im eingestellten Zeitintervall bei:</p> <p>S: die STATUS-Maske C: die COUNTER-Maske ALL: alle Masken abwechselnd CS: COUNTER-Maske und STATUS-Maske abwechselnd DC: die CONNECT-Maske DT: die TRANSACTION-Maske D: die TRANSACTION-Maske und die CONNECT- Maske abwechselnd Standardwert : S</p>
<p>INFORM-PROGRAM</p> <p>MSG= ' <u>REBASE-COUNTER=NEW</u> '</p>	<p>Rebasierung der Zähler der COUNTER-Maske, d.h. Anfordern eines logischen Nullpunkts ausschließlich für die Sessionzähler der COUNTER-Maske</p> <p>Beim nächsten Sammeln der Zähler werden die aktuellen Stände der COUNTER-Maske abgespeichert ("Rebasierung" oder "Snapshot" der aktuellen Zählerstände) und ab dann werden alle Zählerstände in Bezug auf diesen gespeicherten Stand (logischer neuer Nullpunkt) relativiert.</p> <p>Die Rebasierung kann genutzt werden, um bei sehr langen DBH-Sessions</p> <ul style="list-style-type: none"> • vergleichbare Sessionzähler etwa im Tages- oder Wochenrhythmus auszugeben • die Wahrnehmung der Veränderung von Zählerständen zu erleichtern • Probleme der vergleichenden Interpretation bei einzelnen überlaufenden Zählern zu vermeiden. <p>Eine Rebasierung bezieht sich immer auf eine einzelne Monitorinstanz. Eine Rückkehr zu einer vorigen Rebasierung ist nicht möglich.</p> <p>Die Werte einer Rebasierung bleiben gültig, solange die entsprechende DBH-Session bzw. der UDS-Monitor kontinuierlich weiterläuft.</p> <p>Folgende Ereignisse lassen eine Rebasierung ungültig werden:</p> <ul style="list-style-type: none"> • Neustart der DBH-Session Mit einem Neustart der DBH-Session beginnt der DBH wieder bei 0 mit seinen Zählern. • Neustart des UDS-Monitors Da sich die Daten der Rebasierung lokal im Speicher des Monitors befinden, gehen diese bei einem Neustart des Monitors verloren.

INFORM-PROGRAM**MSG= ' REBASE-COUNTER=OFF '**

Die für die Relativierung genutzten Zählerstände der COUNTER-Maske werden auf 0 gesetzt. Damit werden die Sessionzähler in der COUNTER-Maske wieder als Absolutwerte bezogen auf den Beginn der DBH-Session ausgegeben.

Tabelle 27: UDSMON-Kommandos während des Betriebs

11.4 DISPLAY-Ausgabe des UDS-Monitors in Zwischendateien

Bei den DISPLAY-Kommandos DISPLAY DEPRECATED, DISPLAY EXPENSIVE, DISPLAY MESSAGES und DISPLAY DBCOUNTERS werden die Ergebnisse in eine SAM-Zwischendatei geschrieben, die mittels eines intern vom UDS-Monitor abgesetzten SHOW-FILE-Kommandos auf dem Bildschirm angezeigt wird.

Mit der Anzeige der Ergebnisse befinden Sie sich in der Ausgabe des SHOW-FILE-Kommandos und können mit den komfortablen Blätter- und Suchfunktionen von SHOW-FILE die Ergebnisse auswerten. Um das DISPLAY-Kommando abzuschließen, geben Sie END ein. Damit wird das SHOW-FILE-Kommando verlassen und UDSMON kehrt in seinen Maskenmodus zurück.

Link-Namen der Zwischendateien:

Die verwendete Zwischendatei kann eine permanente oder temporäre Datei sein und wird über folgende Linknamen angesprochen:

\$UDSDD bei DISPLAY DEPRECATED
 \$UDSDE bei DISPLAY EXPENSIVE
 \$UDSDM bei DISPLAY MESSAGES
 \$UDSDC bei DISPLAY DBCOUNTERS

Ist keine Datei über diese Linknamen zugewiesen, so legt UDSMON eine temporäre Datei mit den folgenden Namen an:

#UDSMON.DEPRECATED
 #UDSMON.EXPENSIVE
 #UDSMON.DBH.MESSAGES
 #UDSMON.DBCOUNTERS

Dabei ist # das Temporärkennzeichen und kann durch den BS2000-Systemparameter TEMPFILE in @ geändert werden

Nur im Fall, dass die Verwendung von temporären Dateien nicht möglich ist (BS2000-Systemparameter TEMPFILE='NO'), legt UDSMON eine permanente Datei mit den folgenden Namen an:

UDSMON.DEPRECATED.*tsn.datum*
 UDSMON.EXPENSIVE.*tsn.datum*
 UDSMON.DBH.MESSAGES.*tsn.datum*
 UDSMON.DBCOUNTERS.*tsn.datum*

Verwendung der Zwischendateien:

UDSMON prüft bei jedem Kommando DISPLAY DEPRECATED, DISPLAY EXPENSIVE, DISPLAY MESSAGES und DISPLAY DBCOUNTERS, ob eine mit dem jeweiligen Linknamen zugewiesene Zwischendatei bereits vorhanden ist. Ist eine solche Datei nicht vorhanden, so legt UDSMON eine Zwischendatei neu an und spricht sie temporär über den jeweiligen Linknamen an.

UDSMON eröffnet die Zwischendateien bei einem Kommando DISPLAY EXPENSIVE, DISPLAY MESSAGES oder DISPLAY DBCOUNTERS im EXTEND-Modus, wenn eine Zwischendatei mit Linknamen bereits zugewiesen ist. In diesem Fall werden die Dateien fortgeschrieben. Bei mehreren Kommandos DISPLAY MESSAGES in einem Monitorlauf werden nur die DBH-Meldungen ergänzt, die seit dem vorigen Kommando hinzugekommen sind. Wenn bei den Kommandos DISPLAY EXPENSIVE, DISPLAY MESSAGES oder DISPLAY DBCOUNTERS keine Zwischendatei mit Linknamen zugewiesen ist oder ein neuer Dateiname erkannt wird, wird die Zwischendatei mit OUTPUT-Modus eröffnet.

Beim Kommando DISPLAY DEPRECATED wird die Zwischendatei im OUTPUT-Modus eröffnet und somit bei jedem neuen DISPLAY-Kommando überschrieben.

Die Zwischendateien sind nur während der Bearbeitung eines DISPLAY-Kommandos geöffnet. Dadurch haben Sie in der Zwischenzeit bis zu einem erneuten DISPLAY-Kommando vielfältige Eingriffsmöglichkeiten, z.B. können Sie die Datei umbenennen, sie kopieren oder auch löschen, falls bei fortgeschriebenen Dateien die vorige Ausgabe nicht mehr benötigt wird.

DISPLAY-Kommandos, die das Ergebnis in eine Zwischendatei schreiben, können auch schon als START-Anweisungen eingegeben werden. Diese Kommandos werden dann vorgemerkt und direkt nach dem START-Kommando ausgeführt. Dabei wird aber kein interner SHOW-FILE-Aufruf nach dem Schreiben der Zwischendatei ausgeführt.

Beispiel für eine Ausgabe in eine Zwischendatei

```
=== UDSMON: DISPLAY DBCOUNTERS =====
DB-CONFNAME : FPADB          DATE: 2014-05-30    TIME: 10:44:07

DATABASE NAME          DML  LOG READ  PHYS READ  LOG WRITE  PHYS WRITE
FPADB                  5    40        23         14         8
DBSTAT                 0    26        20         2          2

=== UDSMON: END OF DISPLAY DBCOUNTERS =====
```

11.5 Kommandofolge zum Starten und Bedienen von UDSMON

```
01  [/CREATE-FILE FILE-NAME=monitordatei
      [,SUPPORT=*PUBLIC-DISK(SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär,SECONDARY-
ALLOCATION=sekundär)) oder
      ,SUPPORT=*PRIVATE-DISK(VOLUME=priv-vsn,DEVICE-TYPE=gerät[,SPACE=...])]
      [/ADD-FILE-LINK LINK-NAME=TR,FILE-NAME=monitordatei,ACCESS-METHOD=*SAM]

02  /START-UDS-MONITOR

03  "udsmon-anweisungen zum starten"

04  [EM], [DUE]

05  "udsmon-kommandos während des betriebs"

06  [EM], [DUE]

07  /INFORM-PROGRAM MSG='END'
```

Erklärungen

- 01 Wenn Sie die Monitordatei explizit einrichten wollen, müssen Sie diese per CREATE-FILE-Kommando anlegen. Hier ist die Angabe der *:catid* gemäß den BS2000-Konventionen zulässig (siehe [Abschnitt „Pubsets für UDS/SQL nutzen“](#)).
- 02 Starten des Monitors.
- 04/06 Während des Monitor-Betriebs leiten Sie mit [EM], [DUE] eine Unterbrechung ein.
- 07 Beenden des Monitors.

Nach Programmunterbrechung, z.B. durch [K2], müssen Sie mit dem Kommando INFORM-PROGRAM fortfahren.

Eine fehlerhafte UDSMON-Anweisung im Stapelbetrieb führt zur Beendigung des Monitor-Starts. Auf SYSOUT wird eine Fehlermeldung ausgegeben.

11.6 Beispiel

Die Datenbankbetriebswerte der Konfiguration BEISPIEL sollen verfolgt werden. Die Ausgabemaske soll die Statusangaben enthalten. Das erste Ausgabegerät soll ein Drucker sein. Die Abfragefrequenz beim Drucker soll 60 Sekunden betragen. Das zweite Ausgabegerät soll eine Datensichtstation sein. Die Abfragefrequenz für die Datensichtstation soll 30 Sekunden betragen. Der Monitor soll sich nach 30 Minuten beenden.

```
/SET-JV-LINK LINK-NAME=VERS ,JV-NAME=JV.MANUAL.VERSION
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=&( *VERS ) ,SCOPE=*TASK
/START-UDS-MONITOR

. . .
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:25:51/66B2)
*CONFNAME=BEISPIEL
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:26:09/66B2)
*MASK=S
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:26:21/66B2)
*MEDIUM=L,60
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:26:35/66B2)
*MEDIUM=T,30
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:26:49/66B2)
*RUNTIME=1800
% UDS0501 UDSSMON START-PARAMETER ODER 'HELP' EINGEBEN (MO81446,13:27:03/66B2)
*START
```

11.7 Beschreibung der UDS/SQL-Monitor-Ausgabe auf Datensichtstation

Im Folgenden sind die Masken beschrieben, die UDSMON auf Datensichtstation ausgibt: die UDS/SQL-Monitor-Masken STATUS und COUNTER, und die UDS-D-Monitor-Masken TRANSACTION und CONNECT.

Beschreibung der UDS/SQL-Monitormaske: STATUS

```

UDS/SQL-MONITOR 2.9 * DB-CONFNAME: BEISPIEL                DATE: 2017-06-18
STATUS              * INTERV. SEC: 030 TSKCON: 002 TAACT: 0002 TIME: 13:49:46
*****
      TA      DML      SQLTA SQLDML RDMLLO RDMLI LRDDDB PRDDB LWRDB PWRDB PRDRL PWRRL
IV:  000000 0000000 00000 000000 00000 00000 00000 00000 00000 00000 00000 00000
PROGRAM TSN  RUN-ID   MO ST DB-NAME  CALLS   DML  RDML  LRDDDB PRDDB LWRDB PWRDB
IQS      3VKT 00000004 SR IO VERSAND  FTC4R   0006 0000  00012 00005 00000 00000
IQS      3VMT 00000005 SU US KUNDEN   READY  0001 0000  00000 00000 00000 00000

```

LTG

TAST

In den beiden Überschriftszeilen gibt der UDS/SQL-Monitor Informationen zur Datenbank-Konfiguration, zum Tagesdatum, zur Ausgabefrequenz und zur Uhrzeit aus:

DB-CONFNAME : *konfname*

Name der DB-Konfiguration

DATE : *jjj-mm-tt*

Tagesdatum, angegeben mit Jahr, Monat, Tag

INTERV. SEC: *nnn*

Zeitintervall für die Maskenausgabe in Sekunden

TSKCON:

Anzahl der mit UDS/SQL verbundenen Anwendertasks

TAACT :

Anzahl der aktiven Transaktionen am Ende eines Intervalls

TIME : *hh.mm.ss*

Uhrzeit, angegeben mit Stunde, Minute, Sekunde

In den folgenden Zeilen finden Sie Informationen zu den internen Zählerständen des letzten Ausgabeintervalls sowie zu den momentan aktiven Transaktionen.

In dieser Maske werden Informationen zu maximal 16 aktiven Transaktionen ausgegeben, nach der Transaktionskennung aufsteigend sortiert. Wenn mehr Transaktionen aktiv sein sollten, werden die 16 ältesten ausgegeben.

IV (Intervall):

TA

Anzahl der Transaktionen

DML

Anzahl der CODASYL-Anweisungen und der SQL-Anweisungen

SQLTA

Anzahl der SQL-Transaktionen

SQLDML

Anzahl der SQL-Anweisungen

Bei UDS-D:

RDMLO

Anzahl der DML-Anweisungen, die zur Verarbeitung an entfernte Partnerkonfigurationen geschickt wurden

Bei UDS-D:

RDMLI

Anzahl der DML-Anweisungen, die von entfernten Partnerkonfigurationen zur Verarbeitung eingegangen sind

LRDDB

Anzahl der logischen Lese-Aufrufe (READ) auf die Datenbanken

PRDDB

Anzahl der physischen Lese-Aufrufe (READ) auf die Datenbanken

LWRDB

Anzahl der logischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PWRDB

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PRDRL

Anzahl der physischen Lese-Aufrufe (READ) auf die RLOG-Datei

PWRRL

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die RLOG-Datei; bei doppelter RLOG-Sicherung wird nur einmal gezählt

Bezogen auf das (die) gestartete(n) Programm(e) erhalten Sie pro aktiver Transaktion folgende Informationen:

PROGRAM

programe

Bei Teilnehmerbetrieb: Name des Anwenderprogramms

Bei Teilhaberbetrieb: Datenbankschlüssel (openUTM)

Bei UDS-D:

Ist *programe* ein '*' nachgestellt, so bezeichnet *programe* eine sekundäre Teiltransaktion, d.h. das Anwenderprogramm wurde in einer anderen Konfiguration gestartet.

TSN

tsn

Bei Teilnehmerbetrieb: Prozessfolgennummer, unter der das Programm *programe* läuft;

Bei Teilhaberbetrieb: Prozessfolgennummer der zuletzt in der Transaktion genutzten openUTM-Task.

RUN-ID

transaktionskennung

Interne, von UDS/SQL vergebene Transaktionskennung. Die Transaktionen werden innerhalb eines Session-Abschnitts fortlaufend nummeriert.

MO

Eröffnungsmodus der Transaktion

SR	SHARED RETRIEVAL
SU	SHARED UPDATE
PR	PROTECTED RETRIEVAL
PU	PROTECTED UPDATE
ER	EXCLUSIVE RETRIEVAL
EU	EXCLUSIVE UPDATE

ST

gegenwärtiger Zustand der Transaktion

AR=AREA

Die Transaktion befindet sich bei der Ausführung einer READY-Anweisung in einer Realm-Warteschlange; sie kann zurzeit auf einen angeforderten Realm nicht zugreifen

CA= IN CANC

Bearbeitung des CANCEL oder

CA=TO CANC

CANCEL-Auftrag für die Transaktion liegt vor

CC=CONNEC

Die Kontrolle über die Transaktion ist im Verbindungsmodul

DH=DBH

DML wird vom DBH verarbeitet

IO=INPUT/OUTPUT

Die Transaktion wartet auf das Ende einer asynchronen Eingabe oder Ausgabe

LO=LOCK

Die Transaktion wartet auf Freigabe einer Seitensperre

SM=SEMAPHOR

Die Transaktion wartet auf die Freigabe einer Codesperre

SP=STOP

Die Transaktion befindet sich im STOP-Zustand oder

SP=TOSTOP

Es liegt ein STOP-Auftrag vor

ST=SERVERTASK

Die Transaktion wartet auf eine freie Servertask

US=USER

DML beim Anwender

DB-NAME

dbname

Name der Datenbank, auf der die aktuelle DML gerade arbeitet

CALLS

Funktionskurzbeschreibung für DML- und SQL-Anweisungen (siehe DAL-Kommando „DISPLAY SQL“ im ["Informationen über einen SQL-Vorgang ausgeben \(DISPLAY SQL\)"](#) und [Abschnitt „Funktionscodes der DML-Anweisungen“](#)).

DML

Laufende Nummer der DML-Anweisung der Transaktion:

Bei UDS-D:

- *prognose* *: laufende DML-Nummer einer sekundären Teiltransaktion (Anwenderprogramm in einer anderen Konfiguration)
- *prognose* ohne *: laufende DML-Nummer einer rein lokalen Transaktion oder wenn Zähler RDML belegt, laufende DML-Nummer einer primären Teiltransaktion

RDML Bei UDS-D:

Laufende Nummer der entfernt zu bearbeitenden DML-Anweisung, die an eine andere Konfiguration geschickt wurde (remote DML).

LRDDB

Anzahl der logischen Lese-Aufrufe (READ) auf die Datenbanken

PRDDB

Anzahl der physischen Lese-Aufrufe (READ) auf die Datenbanken

LWRDB

Anzahl der logischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PWRDB

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die Datenbanken

Beschreibung der UDS/SQL-Monitormaske: COUNTER

```

UDS/SQL-MONITOR 2.9 *   DB-CONFNAME : BEISPIEL           DATE       : 2017-06-18
COUNTER                *   MACHINE TYPE:  S190-40         TIME        : 13:13:16
*****                *   MAIN MEMORY  : 001232 MB       INTERVAL: 030 SEC

      SESSION      INTERVAL      SESSION      INTERVAL
SQL CONVERSATIONS 0000000000 00000000 LOGICAL READ DB 0000000140 00000010
TRANSACTIONS      0000000005 00000000 PHYSICAL READ DB 0000000074 00000005
SQL TRANSACTIONS 0000000000 00000000 PHYSICAL READ RLOG 0000000000 00000000
UPDATE TRANSACT. 0000000000 00000000 LOGICAL WRITE DB 0000000000 00000000
REMOTE UPD. TRANS. 0000000000 00000000 PHYSICAL WRITE DB 0000000000 00000000
DML CALLS         0000000049 00000005 PHYSICAL WRITE RLOG 0000000004 00000000
SQL CALLS         0000000000 00000000 PHYSICAL WRITE ALOG 0000000000 00000000
RDML OUT          0000000000 00000000
RDML IN           0000000000 00000000 DATA DEADLOCKS 0000000000 00000000
GLOBAL DEADLOCKS 0000000000 00000000
ITC ST --> US     0000000049 00000005 LOCK WAITS      0000000000 00000000
ITC US --> ST     0000000049 00000005
REQUEST ST --> ST 0000000022 00000005 PPP OK          100.0%   ---.-%
ITC ST--> ST     0000000000 00000000 AVG TRANSACTION TIME 000.700s 000.300s

PP TRANSACTION    DEF 0004 ACT 0002 PP SQL          DEF 0004 ACT 0000
PP SEC. TRANSACT. DEF 0001 ACT 0000 PP SERVERTASK   DEF 0001 FREE 0001
SEC. TRANSACT. OUT          ACT 0000 PP 2KB/4KB/8KB-BUFFERSIZE 0001/0001/0000

-----
LTG                                     TAST
    
```

Die drei Überschriftszeilen enthalten Informationen zur Datenbank-Konfiguration, zu Tagesdatum und Uhrzeit, zur Ausgabefrequenz sowie Angaben zur verwendeten Hardware.

DB-CONFNAME : *konfname*

Name der DB-Konfiguration

DATE : *jjj-mm-tt*

Tagesdatum, angegeben mit Jahr, Monat, Tag

MACHINE TYPE : *XXXXXXXX*

Maximal achtstellige Anlagenbezeichnung, z.B. „S190-40“

TIME : *hh.mm.ss*

Uhrzeit, angegeben mit Stunde, Minute, Sekunde

MAIN MEMORY : *nnn.n* MB

Größe des nutzbaren Hauptspeichers in Mbyte

INTERVAL : *nnn*

Zeitintervall für die Maskenausgabe in Sekunden

In den folgenden Zeilen finden Sie Informationen zu den internen Zählerständen seit Beginn der Session und während des letzten Ausgabeintervalls:

SQL CONVERSATIONS

Anzahl bearbeiteter SQL-Vorgänge

TRANSACTIONS

Anzahl der CODASYL- und SQL-Transaktionen

SQL TRANSACTIONS

Anzahl der SQL-Transaktionen

UPDATE TRANSACT.

Anzahl der CODASYL- und SQL-Änderungstransaktionen

REMOTE UPD. TRANS.

Anzahl der Transaktionen mit Änderungen in einer entfernten Konfiguration

DML CALLS

Anzahl der CODASYL-Anweisungen und der SQL-Anweisungen

SQL CALLS

Anzahl der SQL-Anweisungen

Bei UDS-D:

RDML OUT

Anzahl der DML-Anweisungen, die an entfernte Partnerkonfigurationen zur Verarbeitung geschickt wurden (remote DML)

Bei UDS-D:

RDML IN

Anzahl der DML-Anweisungen, die von entfernten Partnerkonfigurationen zur Verarbeitung eingegangen sind

ITC ST --> US

Anzahl der ausgeführten Inter-Task-Kommunikationen (ITC) von der Servertask (ST) zum Benutzer (US)

ITC US --> ST

Anzahl der ausgeführten Inter-Task-Kommunikationen (ITC) vom Benutzer (US) zu der Servertask (ST)

REQUESTS ST --> ST

Summe aller Aufträge von Servertask (ST) zu Servertask

ITC ST --> ST

Anzahl der ausgeführten Inter-Task-Kommunikationen (ITC) von Servertask (ST) zu Servertask

LOGICAL READ DB

Anzahl der logischen Lese-Aufrufe auf die Datenbanken (READ)

PHYSICAL READ DB

Anzahl der physischen Lese-Aufrufe auf die Datenbanken (READ)

PHYSICAL READ RLOG

Anzahl der physischen Lese-Aufrufe auf die RLOG-Datei

LOGICAL WRITE DB

Anzahl der logischen Schreib-Aufrufe auf die Datenbanken (WRITE)

PHYSICAL WRITE DB

Anzahl der physischen Schreib-Aufrufe auf die Datenbanken (WRITE)

PHYSICAL WRITE RLOG

Anzahl der physischen Schreib-Aufrufe auf die RLOG-Datei; bei doppelter RLOG-Sicherung wird nur einmal gezählt

PHYSICAL WRITE ALOG

Anzahl der physischen Schreib-Aufrufe auf die ALOG-Datei

DATA DEADLOCKS

Anzahl der Datendeadlocks

GLOBAL DEADLOCKS

Anzahl der Taskdeadlocks im openUTM-Betrieb und der konfigurationsübergreifenden Deadlocks (bei Einsatz von UDS-D).

LOCK WAITS

Anzahl der durch „Sperrern“ aufgetretenen Wartezustände

PPP OK

Anzahl der erfolgreichen Zugriffsversuche über Probable Position Pointer (PPP) in Prozent.

AVG TRANSACTION TIME

Durchschnittliche Transaktionszeiten in Sekunden, z.B. 0.099 sec

In den folgenden Zeilen finden Sie Informationen zu den Ladeparametern des DBH, wie sie beim DBH-Start definiert wurden (DEF) und wie stark sie ausgelastet sind (FREE/ACT):

PP TRANSACTION DEF

Max. mögliche Anzahl von gleichzeitig aktiven Transaktionen
= Wert bei PP TRANSACTION

Bei UDS-D:
primäre und sekundäre Teiltransaktionen sowie lokale Transaktionen

PP TRANSACTION ACT

Anzahl der aktiven Transaktionen

Bei UDS-D:
primäre und sekundäre Teiltransaktionen sowie lokale Transaktionen, die gerade von diesem DBH bearbeitet werden.

PP SEC. TRANSACT. DEF

Bei UDS-D:
Max. mögliche Anzahl der sekundären Teiltransaktionen, die von diesem DBH bearbeitet werden können (Wert m bei PP TRANSACTION).

PP SEC. TRANSACT. ACT

Bei UDS-D:
Anzahl der aktiven sekundären Teiltransaktionen (STT), die gerade von diesem DBH bearbeitet werden.

SEC. TRANSACT. OUT ACT

Bei UDS-D:
Anzahl der aktiven sekundären Teiltransaktionen (STT), wenn das Anwenderprogramm sich in der eigenen Konfiguration befindet.

PP SQL DEF

Anzahl der mit PP SQL definierten SQL-Vorgänge

PP SQL ACT

Anzahl der offenen SQL-Vorgänge

PP SERVERTASK DEF

Anzahl der mit PP SERVERTASK definierten Servertasks

PP SERVERTASK FREE

Anzahl der freien Servertasks

PP 2KB/4KB/8KB-BUFFER-SIZE

Größe der System Buffer Pools in Mbyte; bei PP 2KB-BUFFER-SIZE, PP 4KB-BUFFER-SIZE resp. PP 8KB-BUFFER-SIZE angegeben.

Beschreibung der UDS-D-Monitormaske TRANSACTION

```

UDS/SQL-MONITOR 2.9 * DB-CONFNAME: BEISPIEL                DATE: 2017-06-18
UDS-D TRANSACTION * INTERVAL : 030 SEC          PARTNER: 0002    TIME: 14:03:27
*****
IV:  TA      DML      SQLTA  SQLDML      LRDDB PRDDB  LWRDB PWRDB  PRDRL PWRRL
     000045  0000612  00000  000000      01024 00000  00260 00000  00000 00040

SUM:          STT-OUT          STT-IN          RDML-OUT          RDML-IN
          SESSION INTERV  SESSION INTERV  SESSION INTERV  SESSION INTERV
          00000390 000022  00000620 000022  00005455 000306  00008646 000302

CONFNAM
BEISP02  00000390 000022  00000620 000022  00005455 000306  00008646 000302
BEISP01  00000000 000000  00000000 000000  00000000 000000  00000000 000000

-----
LTG                                          TAST

```

In den beiden Überschriftszeilen gibt der UDS/SQL-Monitor Informationen aus zur Datenbank-Konfiguration, zum Tagesdatum, zur Ausgabefrequenz, zur Anzahl der Partnerkonfigurationen und zur Uhrzeit:

DB-CONFNAME: *konfname*

Name der eigenen DB-Konfiguration

DATE: *jjj-mm-tt*

Tagesdatum, angegeben mit Jahr, Monat, Tag

INTERVAL: *nnn*

Zeitintervall für die Maskenausgabe in Sekunden

PARTNER

Anzahl der dieser Konfiguration bekannten Partnerkonfigurationen

TIME: *hh.mm.ss*

Uhrzeit, angegeben mit Stunde, Minute, Sekunde

In den folgenden Zeilen finden Sie Informationen zu den internen Zählerständen des letzten Ausgabeintervalls.

Es werden Informationen zu maximal 13 Partnern in dieser Maske ausgegeben.

Zuerst werden die Partner aufgelistet, bei bzw. von denen im letzten Ausgabeintervall eine DML-Bearbeitung stattfand, dann die übrigen bekannten Partner.

IV (Intervall):

TA

Anzahl der Transaktionen

DML

Anzahl der CODASYL-Anweisungen und der SQL-Anweisungen

SQLTA

Anzahl der SQL-Transaktionen

SQLDML

Anzahl der SQL-Anweisungen

LRDDB

Anzahl der logischen Lese-Aufrufe (READ) auf die Datenbanken

PRDDB

Anzahl der physischen Lese-Aufrufe (READ) auf die Datenbanken

LWRDB

Anzahl der logischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PWRDB

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PRDRL

Anzahl der physischen Lese-Aufrufe (READ) auf die RLOG-Datei

PWRRL

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die RLOG-Datei; bei doppelter RLOG-Sicherung wird nur einmal gezählt.

In den folgenden Zeilen finden Sie Informationen, die sich auf die Summe aller Partnerkonfigurationen beziehen. Es werden die Zählerstände seit Beginn der Session und während des letzten Intervalls ausgegeben.

STT-OUT

Anzahl aller sekundären Teiltransaktionen, die an entfernte Partnerkonfigurationen zur Verarbeitung geschickt wurden.

STT-IN

Anzahl aller sekundären Teiltransaktionen, die von entfernten Partnerkonfigurationen zur Verarbeitung geschickt wurden.

RDML-OUT

Anzahl der DML-Anweisungen, die an entfernte Partnerkonfigurationen zur Bearbeitung geschickt wurden (remote DML).

RDML-IN

Anzahl der DML-Anweisungen, die von entfernten Partnerkonfigurationen zur Bearbeitung eingegangen sind.

In den folgenden Zeilen finden Sie Informationen über die angeschlossenen Partnerkonfigurationen:

CONFNAM

Name der Partnerkonfiguration

STT-OUT

Anzahl aller sekundären Teiltransaktionen, die an diese Partnerkonfiguration zur Verarbeitung geschickt wurden.

STT-IN

Anzahl aller sekundären Teiltransaktionen, die von dieser Partnerkonfiguration zur Bearbeitung erhalten wurden.

RDML-OUT

Anzahl der DML-Anweisungen, die an diese Partnerkonfiguration geschickt wurden (remote DML).

RDML-IN

Anzahl der DML-Anweisungen, die von dieser Partnerkonfiguration zur Bearbeitung erhalten wurden.

Beschreibung der UDS-D-Monitormaske CONNECT

```

UDS/SQL-MONITOR 2.9 * DB-CONFNAME: BEISPIEL          DATE: 2017-06-18
UDS-D CONNECT * INTERVAL : 030 SEC PARTNER: 0002    TIME: 12:13:24
*****
IV:  TA      DML      SQLTA  SQLDML  RDML0  RDMLI  LRDD0  PRDD0  LWRDB  PWRDB  PRDRL  PWRRL
     000047  0000657  00000  000000  00333  00322  01103  00000  00286  00000  00000  00044

SUM:          CONNECT-OUT          CONNECT-IN          DISCON          GLOB-DEADLOCKS
          REQUEST REJECT ACTIVE  REQUEST REJECT ACTIVE  SESS          SESS
          0000005 000000 000002  0000007 000000 000000  000000  0000000000000000

CONFNAM
BEISP02 0000005 000000 000002  0000007 000000 000000  000000
BEISP01 0000000 000000 000000  0000000 000000 000000  000000

-----
LTG                                          TAST

```

In den beiden Überschriftszeilen gibt der UDS/SQL-Monitor Informationen aus zur Datenbank-Konfiguration, zum Tagesdatum, zur Ausgabefrequenz, zur Anzahl der Partnerkonfigurationen und zur Uhrzeit:

DB-CONFNAME: *konfname*

Name der eigenen DB-Konfiguration

DATE: *jjj-mm-tt*

Tagesdatum, angegeben mit Jahr-Monat-Tag

INTERVAL: *nmn*

Zeitintervall für die Maskenausgabe in Sekunden

PARTNER

Anzahl der dieser Konfiguration bekannten Partnerkonfigurationen

TIME: *hh.mm.ss*

Uhrzeit, angegeben mit Stunde, Minute, Sekunde

In den folgenden Zeilen finden Sie Informationen zu den internen Zählerständen des letzten Ausgabeintervalls.

Es werden Informationen zu maximal 13 Partnern in dieser Maske ausgegeben.

Es werden zuerst die Partner aufgelistet, zu denen momentan aktive Verbindungen bestehen, dann die übrigen bekannten Partner.

IV (Intervall):

TA

Anzahl der Transaktionen

DML

Anzahl der CODASYL-Anweisungen und der SQL-Anweisungen

SQLTA

Anzahl der SQL-Transaktionen

SQLDML

Anzahl der SQL-Anweisungen

RDMLO

Anzahl der DML-Anweisungen, die zur Verarbeitung an entfernte Konfigurationen geschickt wurden.

RDMLI

Anzahl der DML-Anweisungen, die von entfernten Anwenderprogrammen zur Verarbeitung eingegangen sind.

LRDDB

Anzahl der logischen Lese-Aufrufe (READ) auf die Datenbanken

PRDDB

Anzahl der physischen Lese-Aufrufe (READ) auf die Datenbanken

LWRDB

Anzahl der logischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PWRDB

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die Datenbanken

PRDRL

Anzahl der physischen Lese-Aufrufe (READ) auf die RLOG-Datei

PWRRL

Anzahl der physischen Schreib-Aufrufe (WRITE) auf die RLOG-Datei; bei doppelter RLOG-Sicherung wird nur einmal gezählt.

In den folgenden Zeilen finden Sie Informationen, die sich auf die Summe aller Partnerkonfigurationen beziehen. Es werden die Zählerstände seit DBH-Start ausgegeben.

CONNECT-OUT

REQUEST

Gesamtzahl der Verbindungsanforderungen, die während der Session an entfernte Partnerkonfigurationen gestellt wurden.

CONNECT-OUT**REJECT**

Gesamtzahl der von entfernten Partnerkonfigurationen abgewiesenen Verbindungsanforderungen.

CONNECT-OUT**ACTIVE**

Gesamtzahl der momentan bestehenden Verbindungen zu entfernten Partnerkonfigurationen, die von der eigenen Konfiguration angefordert wurden.

CONNECT-IN**REQUEST**

Gesamtzahl der Verbindungsanforderungen, die während der Session von entfernten Partnerkonfigurationen an die eigene Konfiguration gestellt wurden.

CONNECT-IN**REJECT**

Gesamtzahl der von der eigenen Konfiguration abgewiesenen Verbindungsanforderungen entfernter Partnerkonfigurationen.

CONNECT-IN**ACTIVE**

Gesamtzahl der momentan bestehenden Verbindungen zur eigenen Konfiguration, die von entfernten Partnerkonfigurationen angefordert wurden.

DISCON SESS

Gesamtzahl der Verbindungen zu entfernten Partnerkonfigurationen, die während der Session von Anwenderprogrammen der lokalen Konfiguration wegen Fehler gelöst wurden. Unterbrechungsursachen:

- CHECKTIME-Überprüfung des Anwenders
- von DCAM gemeldete Fehler

GLOB-DEADLOCKS SESS

Alle vom Lockmanager der eigenen Konfiguration erkannten Taskdeadlocks, d.h. im openUTM-Betrieb auftretenden Taskdeadlocks und konfigurationsübergreifenden Deadlocks bei Einsatz von UDS-D. Ein konfigurationsübergreifender Deadlock wird dann angenommen, wenn sich konfigurationsübergreifende Wartebeziehungen zwischen verteilten Transaktionen während einer bestimmten Zeit nicht verändern.

Diese Wartezeit ist durch PP DEADTIME einstellbar.

In den folgenden Zeilen finden Sie Informationen über die angeschlossenen Partnerkonfigurationen. Es werden die Zählerstände seit DBH-Start ausgegeben.

CONFNAM

Name der Partnerkonfiguration

CONNECT-OUT

REQUEST

Gesamtzahl der Verbindungsanforderungen, die während der Session an diese Partnerkonfiguration gestellt wurden.

CONNECT-OUT

REJECT

Gesamtzahl der von dieser Partnerkonfiguration während der Session abgewiesenen Verbindungsanforderungen.

CONNECT-OUT

ACTIVE

Gesamtzahl der momentan zu diesem Partner bestehenden Verbindungen.

Pro Anwenderprogramm und von der UDS-D-Task der eigenen Konfiguration ist eine Verbindung zu dieser Partnerkonfiguration möglich. Die Verbindung „eigene UDS-D-Task - fremde UDS-D-Task“ wird nur bei CONNECT-IN ACTIVE gezählt.

CONNECT-IN

REQUEST

Gesamtzahl der Verbindungsanforderungen, die während der Session von dieser Partnerkonfiguration gestellt wurden.

CONNECT-IN

REJECT

Gesamtzahl der von der eigenen Konfiguration abgewiesenen Verbindungsanforderungen dieser Partnerkonfiguration.

CONNECT-IN

ACTIVE

Gesamtzahl der momentan bestehenden Verbindungen von dieser Partnerkonfiguration zur eigenen Konfiguration.

Pro Anwenderprogramm und vom entfernten UDS-D ist eine Verbindung von dieser Partnerkonfiguration möglich.

DISCON SESS

Gesamtzahl der Verbindungen, die während der Session wegen Fehler zu dieser Partnerkonfiguration gelöst wurden.

11.8 Beschreibung der UDS/SQL-Monitor-Ausgabe auf Drucker

Bei der Listenausgabe auf Drucker werden die Zähler der COUNTER-Maske und STATUS-Maske zusammengefasst.

Zusätzlich können aus den TRANSACTION- und CONNECT-Monitormasken Informationen zu UDS-D ausgegeben werden. Diese Ausgabe wird durch das UDSMON-Kommando MEDIUM=L,*n*,D (D=distributed) gesteuert.

Die Listenausgabe entspricht inhaltlich im Wesentlichen der Ausgabe auf Datensichtstation (siehe [Abschnitt „Beschreibung der UDS/SQL-Monitor-Ausgabe auf Datensichtstation“](#)). Es existieren jedoch folgende Unterschiede:

- Bei den STATUS-, TRANSACTION- und CONNECT-Monitormasken entfällt die Informationszeile mit den Intervallzählern (IV). Es werden Informationen zu allen aktiven Transaktionen ausgegeben.
- In der STATUS-Maske wird, falls es sich um eine openUTM-Transaktion handelt, in der Spalte TSN im Anschluss an die TSN das Zeichen „*“ ausgegeben.
- Falls eine Transaktion in OSI TP mit „Functional Unit Commit“ global verteilt ist, werden für diese Transaktion in einer zusätzlichen Zeile der Listenausgabe bis zu 64 Byte der Atomic Action Identifier (AAID) im Hexaformat ausgegeben. Bei über UDS-D verteilten Transaktionen erfolgt die Ausgabe nur in der Konfiguration der primären Teiltransaktion, also in der Konfiguration, in der die entsprechende UTM-Anwendung angeschlossen ist. Die AAID wird von einer OSI-TP Anwendung erzeugt, die als Partner einer UTM-Anwendung die Transaktion steuert. Diese steuernde OSI-TP Anwendung kann z.B. UTM (auf BS2000 oder anderen Plattformen), das LU6.2 Gateway oder die generische BeanConnect-Proxy UTM-Anwendung auf Linux sein. In dem entsprechenden Instanz-Trace des Gateways wird die Erzeugung der AAID protokolliert. Damit kann die Transaktion der steuernden Transaktion im Fremdsystem zugeordnet werden.

11.9 Beschreibung der UDS/SQL-Monitor-Ausgabe in eine Datei

Wenn Sie das UDSMON-Kommando MEDIUM=F, r[,D] eingeben, schreibt der UDS/SQL-Monitor immer nach Ablauf des angegebenen Zeitintervalls Datensätze in die Ausgabedatei.

Zu Beginn der UDSMON-Ausgabe und bei Änderung des Zeitintervalls wird ein Kennungssatz in die Datei geschrieben.

Die Ausgabedatei ist eine SAM-Datei mit variabler Satzlänge. Die aktuellen Zählerstände werden in binärer Form ausgegeben.

Bei UDS-D:

Wenn mit UDS-D gearbeitet wird und Sie D (D=distributed) angeben, wird auch der UDS-D-Betrieb ausgewertet.

Die Ausgabedatei ist dann folgendermaßen aufgebaut:

UDS/SQL-Kennungssatz

UDS-D-Kennungssatz

UDS/SQL-Datensatz

UDS-D-Datensatz

...

Der Kennungssatz für UDS-D wird ausgegeben bei Beginn der UDSMON-Ausgabe, bei Änderung des Zeitintervalls, bei Änderung des DBH-Ladeparameters PP PTCSYNCH und bei Zuschalten und Abschalten der Verteiltabelle.

i Der Satzaufbau kann sich bei Versionswechsel ändern.
 Wenn bei UDS/SQL versionsbedingt eine Änderung des Satzaufbaus notwendig wird, so wird im Regelfall der Satz verlängert.
 Sie können die UDS/SQL-Versionsnummer aus dem Feld Nummer 4 entnehmen (siehe „Aufbau des UDS/SQL-Kennungssatzes“).

Satzaufbau der Dateiausgabe

Aufbau des UDS/SQL-Kennungssatzes

Feldnr	Feldinhalt	Distanz [dez]	Länge [byte]
1	Satzlänge	0	2
2	Filler	2	2
3	Satzidentifikation X'0010'	4	2
4	UDS/SQL-Versionsnummer X'0290'=V2.9	6	2
5	Zeit im STCK-Format	8	8
6	Konfigurationsname	16	20
7	Zeitintervall in Sekunden	36	4

8	PP SERVERTASK		40	2
9	PP TRANSACTION		42	2
10	PP 2KB-BUFFER-SIZE		44	2
11	PP MAXDB		46	2
12	PP SUBSCHEMA		48	2
13	PP LOG	X'00' kein RLOG	50	1
		X'01' einfache RLOG		
		X'02' doppelte RLOG		
14	PP CPU	X'00' CPU-MONO	51	1
		X'01' CPU-MULTI		
15	PP DEACT	X'00' DEACTIVE=YES	52	1
		X'01' DEACTIVE=NO		
16	PP SCHEDULING	X'00' SYMMETRIC	53	1
		X'01' ASYMMETRIC		
17	PP IO	X'00' ASYNCHRON	54	1
		X'01' SYNCHRON		
18	Filler		55	1
19	Rechnertyp, z.B C'S150-60'		56	8
20	interne Kennzeichnung von Betriebssystem und Betriebssystemversion		64	4
21	DCAM-Prozessorname (z.B. D016ZE09)		68	8
22	Hauptspeichergröße in Kbyte abdruckbar		76	8
23	Anzahl Prozessoren abdruckbar		84	2
24	bis zu 8 Seriennummern (binär); je CPU 3 Byte Identifikationsnummer (Byte 0-2: Nummer der ersten CPU, Byte 3-5 Nummer der zweiten CPU,...)		86	24
25	PP SQL		110	2
26	Filler (für internen Gebrauch)		112	4
27	PP 4KB-BUFFER-SIZE		116	2
28	PP 8KB-BUFFER-SIZE		118	2
29	Datum (abdruckbar) Format: <i>jjj-mm-tt</i>		120	10
30	Uhrzeit (abdruckbar) Format: <i>hh.mm.ss</i>		130	8

31	<p>Kenngößen der lokalen Zeitzone Format: $shh_1:mm_1-hh_2:mm_2-z$ <i>s.</i> Vorzeichen („+“ oder „-“ für den Zeitunterschied zwischen lokaler Zeitzone und UTC-Zeit (Greenwich-Zeit) $hh_1:mm_1$: Zeitunterschied zwischen lokaler Zeitzone und UTC-Zeit in Stunden (hh_1) und Minuten (mm_1) $hh_2:mm_2$: Zeitverschiebung zwischen Sommer- und Normalzeit in Stunden (hh_2) und Minuten (mm_2) <i>z.</i> Aktuelle Zeitählung der lokalen Zeitzone („W“ für Winterzeit, „S“ für Sommerzeit)</p>	138	14
----	---	-----	----

Tabelle 28: Aufbau des UDS/SQL-Kennungssatzes

Aufbau des UDS-D-Kennungssatzes

Feldnr	Feldinhalt		Distanz [dez]	Länge [byte]
1	Satzlänge		0	2
2	Filler		2	2
3	Satzidentifikation	X'0020'	4	2
4	UDS-D-Versionsnummer	X'0290' = V2.9	6	2
5	Zeit im STCK-Format		8	8
6	Zeitintervall in sec		16	4
7	PP TRANSACTION LOC. bzw. GLOB.		20	2
8	PP TRANSACTION gleichzeitige STTs		22	2
9	PP CHCKTIME in sec		24	2
10	PP DEADTIME in sec		26	2
11	PP DISDB		28	2
12	PP PTCSYNCH-WARM	X'00' WAIT	30	1
		X'01' ABORT		
		X'02' COMMIT		
13	PP PTCSYNCH-SESSION	X'00' WAIT	31	1
		X'01' ABORT		
		X'02' COMMIT		
14	PP DISTABLE	X'00' keine Verteiltabelle	32	1
		X'01' Verteiltabelle		
15	Filler		33	3
16	Datum (abdruckbar) Format: $jjjj-mm-tt$		36	10

17	Uhrzeit (abdruckbar) Format: <i>hh.mm.ss</i>	46	8
18	Kenngößen der lokalen Zeitzone Format: <i>shh₁:mm₁-hh₂:mm₂-z</i> <i>s</i> : Vorzeichen („+“ oder „-“ für den Zeitunterschied zwischen lokaler Zeitzone und UTC-Zeit (Greenwich-Zeit) <i>hh₁:mm₁</i> : Zeitunterschied zwischen lokaler Zeitzone und UTC-Zeit in Stunden (<i>hh₁</i>) und Minuten (<i>mm₁</i>) <i>hh₂:mm₂</i> : Zeitverschiebung zwischen Sommer- und Normalzeit in Stunden (<i>hh₂</i>) und Minuten (<i>mm₂</i>) <i>z</i> : Aktuelle Zeitählung der lokalen Zeitzone („W“ für Winterzeit, „S“ für Sommerzeit)	54	14

Tabelle 29: Aufbau des UDS-D-Kennungssatzes

Aufbau des UDS/SQL-Datensatzes

Feldnr	Feldinhalt	Distanz [dez]	Länge [byte]
1	Satzlänge	0	2
2	Filler	2	2
3	Satzidentifikation X'0011'	4	2
4	Filler	6	2
5	Zeit im STCK-Format	8	8
6	Freie Servertasks	16	4
7	Aktive Transaktionen	20	4
8	Offene SQL-Vorgänge	24	4
9	Anzahl DMLs	28	4
10	Anzahl Transaktionen	32	4
11	Anzahl Änderungs-Transaktionen	36	4
12	Anzahl Daten-Deadlocks	40	4
13	Anzahl globaler Task-Deadlocks	44	4
14	Filler (wird mit 0 vorbelegt)	48	4
15	Anzahl PPPs	52	4
16	Anzahl PPP OK	56	4
17	Anzahl LOCK-CALLs	60	4
18	Anzahl LOCK-Waits	64	4
19	Anzahl Aufträge US --> ST	68	4
20	Anzahl ITC US --> ST	72	4

21	Anzahl Aufträge ST --> ST	76	4
22	Anzahl ITC ST --> ST	80	4
23	Anzahl ITC ST --> US	84	4
24	Anzahl LOG. READ DB	88	4
25	Anzahl LOG. WRITE DB	92	4
26	Anzahl PHYS. READ DB	96	4
27	Anzahl PHYS. WRITE DB	100	4
28	Anzahl LOG. READ RLOG	104	4
29	Anzahl LOG. WRITE RLOG	108	4
30	Anzahl PHYS. READ RLOG	112	4
31	Anzahl PHYS. WRITE RLOG	116	4
32	Anzahl PHYS. READ ALOG	120	4
33	Anzahl PHYS. WRITE ALOG	124	4
34	Anzahl LOG. READ ALOG	128	4
35	Anzahl LOG. WRITE ALOG	132	4
36	Anzahl SEQUENTIAL READ DB	136	4
37	Anzahl ACCEPT CRU ohne ITC	140	4
38	Anzahl SQLDMLs	144	4
39	Anzahl aus SQLDMLs generierter BIBs	148	4
40	Anzahl SQL-Transaktionen	152	4
41	Anzahl Remote-UPDATE-Transaktionen	156	4
42	Anzahl bearbeiteter SQL-Vorgänge	160	4
43	Filler	164	4
44	Summe TA-Zeiten in Sekunden (binär)	168	4
45	Anzahl gemessener Transaktionen	172	4
46	Anzahl LOG. READ 2KB-DB	176	4
47	Anzahl LOG. WRITE 2KB-DB	180	4
48	Anzahl PHYS. READ 2KB-DB	184	4
49	Anzahl PHYS. WRITE 2KB-DB	188	4
50	Anzahl LOG. READ 4KB-DB	192	4
51	Anzahl LOG. WRITE 4KB-DB	196	4
52	Anzahl PHYS. READ 4KB-DB	200	4
53	Anzahl PHYS. WRITE 4KB-DB	204	4

54	Anzahl LOG. READ 8KB-DB	208	4
55	Anzahl LOG. WRITE 8KB-DB	212	4
56	Anzahl PHYS. READ 8KB-DB	216	4
57	Anzahl PHYS. WRITE 8KB-DB	220	4
58	Anzahl LOG. READ EXCL-DB	224	4
59	Anzahl LOG. WRITE EXCL-DB	228	4
60	Anzahl PHYS. READ EXCL-DB	232	4
61	Anzahl PHYS. WRITE EXCL-DB	236	4
62	Filler	240	10
63	Datum (abdruckbar) Format: <i>jjjj-mm-tt</i>	250	10
64	Uhrzeit (abdruckbar) Format: <i>hh.mm.ss</i>	260	8

Tabelle 30: Aufbau des UDS/SQL-Datensatzes

Aufbau des UDS-D-Datensatzes

Feldnr	Feldinhalt	Distanz [dez]	Länge [byte]
1	Satzlänge	0	2
2	Filler	2	2
3	Satzidentifikation X'0021'	4	2
4	Filler	6	2
5	Zeit im STCK-Format	8	8
6	Anzahl STT ACTIVE OUT	16	4
7	Anzahl STT ACTIVE IN	20	4
8	Anzahl CONNECT ACTIVE OUT	24	4
9	Anzahl CONNECT ACTIVE IN	28	4
10	Filler	32	4
11	Filler	36	4
12	Filler	40	4
13	Filler	44	4
14	Anzahl Partner	48	4
15	Anzahl RDML OUT	52	4
16	Anzahl RDML IN	56	4

17	Anzahl STT OUT	60	4
18	Anzahl STT IN	64	4
19	Anzahl CONNECT REQUEST OUT	68	4
20	Anzahl CONNECT REJECT OUT	72	4
21	Anzahl CONNECT REQUEST IN	76	4
22	Anzahl CONNECT REJECT IN	80	4
23	Anzahl DISCONNECT	84	4
24	Anzahl GLOB-DEADLOCKS SESS	88	4
25	Filler	92	6
26	Datum (abdruckbar) Format: <i>jjj-mm-tt</i>	98	10
27	Uhrzeit (abdruckbar) Format: <i>hh.mm.ss</i>	108	8

Tabelle 31: Aufbau des UDS-D-Datensatzes

11.10 Übergabe von Monitorzählern an openSM2

Ab UDS/SQL V2.6 und BS2000 V8 können Sie Zählerstände der Counter-Maske des UDS-Monitors an openSM2 übergeben und diese dort z.B. mit dem INSPECTOR grafisch aufbereitet präsentieren. Einige dieser Zählerstände können auch über den UDS SQL-Report von openSM2 am Terminal ausgegeben werden bzw. in den SM2R1-Reports ausgewertet werden.

Eine ausführliche Beschreibung von openSM2 finden Sie im Handbuch „openSM2 (BS2000)“.

Voraussetzung in UDS/SQL

Beim Start des UDS-Monitors veranlassen Sie die Übergabe von Zählerständen der Counter-Maske an openSM2 mit der folgenden Anweisung:

```
MEDIUM=S,n
```

Während des Betriebes (siehe „UDSMON-Kommandos während des Betriebs“ im [Abschnitt „Anweisungen und Kommandos für UDSMON“](#)) starten Sie die Übergabe der Zählerstände an openSM2 mit dem Kommando

```
INFORM-PROGRAM MSG='ADD MEDIUM=S,n'
```

Sie beenden die Übergabe der Zählerstände mit dem Kommando

```
INFORM-PROGRAM MSG='FINISH MEDIUM=S'
```

Mit *n* wird das Zeitintervall in Sekunden ($5 \leq n \leq 999$) definiert, mit dem der UDS-Monitor die Daten bei SM2 abliefern. Es sollte deutlich niedriger als das in SM2 eingestellte Messintervall gewählt werden, damit innerhalb eines SM2-Messintervalls mehrmals Daten übergeben werden.

Die Übergabe der Zählerstände an openSM2 kann durch eine Monitor-Instanz erfolgen, die auch noch andere Ausgaben tätigt (Ausgaben auf Terminal, in eine Liste oder in einem gesonderten Format in eine Datei). Um eine ungestörte Versorgung zu garantieren, wird aber empfohlen, für die Übergabe der Zählerstände an openSM2 einen eigenständigen Monitorlauf im Hintergrund zu nutzen. Bei Monitor-Instanzen im Vordergrund besteht die Gefahr, dass Unterbrechungen zur Änderung von Ausgabemedien, Masken bzw. Zeitintervallen unerwünschte Auswirkungen auf die Präsentation der Zählerstände bei openSM2 haben.

Voraussetzung in BS2000

Voraussetzung für die Übergabe von Daten an openSM2 ist, dass in SM2 im Sammler mit der folgenden Anweisung die Entgegennahme der Daten aktiviert ist:

```
START-MEASUREMENT-PROGRAM TYPE=*UDS-SQL
```

Wenn diese Schnittstelle nicht aktiviert ist oder mit der Anweisung STOP-MEASUREMENT-PROGRAM TYPE=*UDS-SQL deaktiviert wurde, gibt UDS/SQL einmalig eine entsprechende Fehlermeldung (UDS0536) aus.

Es wird jedoch weiterhin versucht, openSM2 mit den Daten zu versorgen, sodass nach dem Start des Messprogramms UDS-SQL ohne weitere Maßnahmen im UDS-Monitor die Daten an openSM2 übergeben werden.

Für die Nutzung der übergebenen Daten im INSPECTOR ist es notwendig, dass der Agent des INSPECTORs im BS2000 aktiv ist (START-INSPECTOR).

i Die Messwerte werden von UDS/SQL asynchron an openSM2 geliefert und gelten für ein oder mehrere von UDS/SQL festgelegte Intervalle, die nicht exakt mit dem SM2-Intervall übereinstimmen müssen. Hierbei kann es sowohl Unterschiede bei der Dauer der Intervalle als auch zeitliche Verschiebungen zwischen den UDS/SQL- und den SM2-Intervallen geben.
Für die Normierung der Messwerte auf eine Sekunde wird die Dauer des einen oder der mehreren UDS/SQL-Intervalle herangezogen. Die Werte sind also exakt, aber sie passen nur bedingt zum SM2-Intervall.

Darstellung der Counter-Daten mit INSPECTOR

Mit der INSPECTOR-Komponente von openSM2 können Sie die Daten der Counter-Maske an einer Workstation grafisch aufbereiten und zusammen mit anderen Verbrauchswerten des BS2000-Systems anzeigen lassen. In der Report-Gruppe UDS-SQL wählen Sie aus, welche Zählerstände angezeigt werden sollen. Sie können jederzeit die zeitliche Entwicklung der Zählerstände verfolgen, so das Systemverhalten beurteilen und Probleme frühzeitig erkennen.

Insbesondere können Sie die Daten anhand von benutzerdefinierten Regeln überwachen und damit ungewöhnliche Zustände erkennen. In einer Regel sind Bedingungen und Aktionen angegeben. Treffen die Bedingungen zu, so werden die zugehörigen Aktionen ausgeführt, z.B. können SNMP-Traps ausgelöst werden, die über den Performance-Subagenten SM2 zu Benachrichtigungen an eine angeschlossene SNMP-Station führen, oder es können ereignisbezogene E-Mails versandt werden.

Eine Beschreibung zum Einsatz des INSPECTORs sowie eine Liste der dort darstellbaren UDS/SQL-Zähler finden Sie im Handbuch „[openSM2 \(BS2000\)](#)“. Eine Beschreibung des Performance-Subagenten SM2 finden Sie im Handbuch „[SNMP Management \(BS2000\)](#)“.

Ausgabe des UDS SQL-Reports von openSM2

Einige Zählerstände der Counter-Maske können Sie auch über den UDS SQL-Report von openSM2 am BS2000-Terminal ausgeben. Eine ausführliche Beschreibung des UDS SQL-Reports von openSM2 finden Sie im Handbuch „[openSM2 \(BS2000\)](#)“.

12 Allgemeine Funktionen der Dienstprogramme

Dieses Kapitel beschreibt allgemeine Funktionen der Dienstprogramme:

- Dienstprogramme starten
- Prüfung der UDS/SQL-Subset-Deklaration
- Automatische Realm-Erweiterung
- Datenbankinformationen systemneutral ausgeben (Ausgabe im CSV-Format)
- Auftragsschalter setzen und auswerten

12.1 Dienstprogramme starten

Die UDS-SQL-Dienstprogramme werden mit folgenden Kommandos geladen und gestartet:

START-UDS-*program*

VERSION = *STD /<product-version>

,**MONJV = *NONE** / <filename 1..54 without-gen-vers>

,**CPU-LIMIT = *JOB-REST** / <integer 1..32767 *seconds*>

program

Name des Dienstprogramms:

ADM, BALTER, BCALLSI, BCHANGE, BCHECK, BCREATE, BFORMAT, BGSIA, BGSSIA, BINILOAD, BMEND, BMODTT, BOUTLOAD, BPGSIZE, BPRECORD, BPRIVACY, BPSIA, BPSQLSIA, BRENAME, BREORG, BSTATUS, DDL, DMLTEST, MONITOR, ONLINE-PRIVACY oder SSL.

VERSION =

Produktversion des Dienstprogramms, die gestartet werden soll.

VERSION = *STD

Keine explizite Angabe der Produktversion. Die Produktversion wird folgendermaßen ausgewählt:

1. Die mit dem Kommando /SELECT-PRODUCT-VERSION vorgegebene Version.
2. Die höchste mit IMON installierte Version.

VERSION = <product-version>

Explizite Angabe der Produktversion in der Form mm.n[a[kk]].

Es wird empfohlen, die Version immer vollständig anzugeben, z.B. 02.9B00, um den Versionswechsel bei Korrekturlieferungen zu erleichtern.

MONJV =

Angabe einer Monitor-Jobvariablen zur Überwachung des DBH-Laufs.

MONJV = *NONE

Es wird keine Monitor-Jobvariable verwendet.

MONJV = <filename 1..54 without-gen-vers>

Name der zu verwendenden Jobvariablen.

Während des Programmlaufs setzt das System die Jobvariable auf folgende Werte:

\$R Programm läuft

\$T Programm erfolgreich beendet

\$A Programm fehlerhaft beendet

CPU-LIMIT =

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf.

CPU-LIMIT = *JOB-REST

Es soll die verbleibende CPU-Zeit des BS2000-Jobs für die Aufgabe verwendet werden.

CPU-LIMIT = <integer 1..32767 *seconds*>

Es soll nur die angegebene Zeit verwendet werden.

12.2 Zuweisung des Datenbanknamens

Die Dienstprogramme bearbeiten jeweils eine Datenbank. Der Name der Datenbank kann im Allgemeinen über den Linknamen DATABASE übergeben werden. Bei der Zuweisung mit ADD-FILE-LINK bzw. SET-FILE-LINK können Sie als FILE-NAME auch den Dateinamen des DBDIR der betreffenden Datenbank angeben, z.B. um eine Schattendatenbank eindeutig zu identifizieren.

Speziell bei Dienstprogrammen, die den linked-in DBH nutzen, ist der Datenbankname gleichzeitig auch der Konfigurationsname. Dieser Name wird ebenfalls über den Linknamen DATABASE übergeben (siehe Link-Zuweisung DATABASE für den Konfigurationsnamen des DBH ab "[DBH-Startkommandos](#)").

Bei der Nutzung eines Online-Dienstprogrammes (UDS-Online-Utility, ONLINE-PRIVACY) weisen Sie über den Linknamen DATABASE nur den Namen der DBH-Konfiguration zu. Die dabei zu bearbeitende Datenbank wird über eine Anweisung OPEN-DATABASE oder über einen eindeutigen Subschemanamen bestimmt.

12.3 Prüfung der UDS/SQL-Pubset-Deklaration

Beim Beginn eines Dienstprogramms wird eine vorliegende UDS/SQL-Pubset-Deklaration geprüft. Die Einstellungen einer fehlerfreien UDS/SQL-Pubset-Deklaration werden für die weitere Verwendung vermerkt, sie werden aber nicht protokolliert.

- i** Eine beim Dienstprogrammbeginn vorliegende fehlerhafte Jobvariablen-Zuweisung oder die Zuweisung einer fehlerhaften UDS/SQL-Pubset-Deklaration führt zum Abbruch des Dienstprogramms.

Näheres zur UDS/SQL-Deklaration siehe [Abschnitt „Pubset-Deklarations-Jobvariable“](#)).

12.4 Automatische Realm-Erweiterung

Einige Dienstprogramme erweitern bei Bedarf automatisch die Realms der bearbeiteten Datenbank bzw. die DBTTs der Satzarten in DBDIR und DBCOM. Näheres hierzu siehe [Abschnitt „Automatische Realm-Erweiterung durch Dienstprogramme“](#)).

12.5 Datenbankinformationen systemneutral ausgeben

Die informierenden Dienstprogramme BPSIA, BSTATUS, BPRECORD und BOUTLOAD unterstützen die Ausgabe der Daten im sogenannten CSV-Format (**character separated values**). Mit diesem Format wird die Weiterverarbeitung von Daten in anderen Systemumgebungen (z.B. mit Tabellenkalkulationsprogrammen) erleichtert.

Als Separatorzeichen für die Werte wird ein Semikolon „;“ verwendet, da Kommas in der Datenausgabe verwendet werden.

12.5.1 Unterschiede der CSV-Ausgabe zur Ausgabe auf SYSLST

Daten im CSV-Format können optional zusätzlich zur Ausgabe auf SYSLST ausgegeben werden. Die Ausgabe der Daten im CSV-Format orientiert sich inhaltlich an der Ausgabe auf SYSLST. Es existieren jedoch Abweichungen im Format der Datenausgabe, die dadurch begründet sind, dass die Daten nicht auf Papier weiter genutzt werden, sondern in weiterverarbeitenden Programmen:

- Alle Ausgaben im CSV-Format werden bei einem Dienstprogrammlauf in einer Datei abgespeichert.
- In der ersten Spalte wird eine Kennzeichnung bereitgestellt, die eine zeilenweise Auswertung der Daten ermöglicht.
- Bei der Ausgabe auf SYSLST ergibt sich eine Beschränkung durch die Druckbreite von 132 Zeichen je Zeile. Diese Einschränkung ist bei der CSV-Ausgabe nicht gegeben. Text-Informationen, die in der SYSLST-Ausgabe z.B. bei Überschriften in mehrere Zeilen umgebrochen werden müssen, werden im CSV-Format in einer Zeile ausgegeben. Abgekürzte Textteile werden zur besseren Verständlichkeit im CSV-Format ausgeschrieben.
- Auf seitenorientierte Zwischenüberschriften wird in der CSV-Ausgabe verzichtet.
- Sich wiederholende erläuternde Texte aus Teiltabellen werden in zusätzliche Überschriften verlagert.
- Jede CSV-Datei enthält in der Kopfzeile den Namen des Dienstprogramms, die Version des CSV-Ausgabeformats des entsprechenden Dienstprogramms sowie Datum und Uhrzeit der Erstellung der CSV-Ausgabe.

Die Version wird geändert, wenn die CSV-Ausgabe bezüglich der Zuordnung zu Ausgabezellen oder die Semantik von Ausgabezellen verändert ist.

- Zahlendarstellungen, die in der SYSLST-Ausgabe aus Platzgründen in Sonderformaten dargestellt werden (z.B. 25M oder 14K in BSTATUS), werden in der CSV-Ausgabe in der korrekten Dezimalausgabe dargestellt.
- Für erläuternde Textteile in der CSV-Ausgabe wird im Gegensatz zur SYSLST-Ausgabe Groß-Kleinschreibung genutzt. Technische Begriffe von UDS/SQL und in der Datenbank deklarierte Namen werden weiterhin nur groß geschrieben.
- BSTATUS gibt in der CSV-Ausgabe zusätzlich den Datenbanknamen und den Schemanamen aus.
- Satzinhalte und Tabelleneinträge im Hex- und Char-Format werden von BPRECORD in der CSV-Datei kompakt ausgegeben. Im Char-Format können beliebige hexadezimale Zeichen vorkommen. Bei der Übertragung der CSV-Datei in eine andere Systemumgebung können einzelne Zeichen mit Sonderbedeutung dazu führen, dass die Zeilenstruktur der CSV-Ausgabe zerstört wird.

Beispielsweise wird das EBCDIC-Zeichen X'25' bei der Umwandlung in ISO8859-1 (ASCII) zu X'0A' umgewandelt. In Unix- und Windows-Umgebung wird X'0A' als \n und damit als Zeilenvorschub interpretiert. Das EBCDIC-Zeichen X'0D' wird beim Wechsel von EBCDIC nach ASCII nicht verändert. In Windows-Umgebung erzeugt es als \r einen Zeilenvorschub. In Unix-Umgebung wird wieder auf den Beginn der aktuellen Zeile positioniert und es kann somit zu Überschreibungen bei der Ausgabe kommen.

Eine einfache Möglichkeit, solche störenden Effekte zu vermeiden, besteht darin, bereits im BS2000 die betreffenden Zeichen durch ein anderes Zeichen zu ersetzen. Dies kann beispielsweise im EDT mit den folgenden Anweisungen erfolgen:

```
@ON & CA X'25' T '?' und
@ON & CA X'0D' T '?'
```

Wenn das Separatorzeichen in der CHAR-Ausgabe des BPRECORD vorkommt, wird es in der CSV-Ausgabe in X'00' umgewandelt, um die weitere übergeordnete Verarbeitung der Felder nicht zu behindern. Wird das Separatorzeichen in der Weiterverarbeitung benötigt, dann muss die Weiterverarbeitung über die unveränderte HEX-Ausgabe erfolgen.

12.5.2 Steuern der CSV-Ausgabe mit DISPLAY-Anweisungen

Die Ausgabe im CSV-Format steuern Sie mit den DISPLAY-Anweisungen der Dienstprogramme BPSIA, BSTATUS und BPRECORD (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“). Hierfür steht der Parameter IN CSV zur Verfügung:

```
DISPLAY [IN CSV [csv-dateiname]] ...
```

csv-dateiname

Name der Datei, in die die Daten im CSV-Format ausgegeben werden sollen.

Die Ausgabe der Daten im CSV-Format erfolgt optional zusätzlich zur Ausgabe auf SYSLST.

Die Ausgabe im CSV-Format erfolgt in die Datei, die bei der ersten DISPLAY-Anweisung angegeben wurde. Insofern ist die Angabe von *csv-dateiname* bei der ersten DISPLAY-Anweisung mit CSV-Ausgabe zwingend.

Bei weiteren DISPLAY-Anweisungen mit CSV-Ausgabe kann *csv-dateiname* weggelassen werden. Wird bei weiteren DISPLAY-Anweisungen mit CSV-Ausgabe ein anderer *csv-dateiname* angegeben, dann wird eine Warnung ausgegeben. Die Ausgabe erfolgt in die ursprünglich angegebene Datei.

Die CSV-Ausgabe ist prinzipiell für einzelne DISPLAY-Anweisungen ansteuerbar.

Intern werden diese DISPLAY-Anweisungen zur performanten Durchführung eines Dienstprogrammlaufes teilweise umgruppiert und zusammengefasst. Hierdurch kann es bei komplexeren Kombinationen von DISPLAY-Anweisungen sein, dass in der CSV-Ausgabe mehr Daten ausgegeben werden als bei enger Auslegung der abgesetzten DISPLAY-Anweisungen erwartet wird. Dies betrifft bei BSTATUS Kombinationen von DISPLAY TABLE SET und DISPLAY TABLE OWNER. Bei BPRECORD wirkt die Ansteuerung einer CSV-Ausgabe grundsätzlich auf alle Ausgaben eines Typs (PAGE, FPA, DBTT, CALC oder DATA) für einen Realm.

Soll in solchen Fällen nur genau die beabsichtigte Ausgabe im CSV-Format erzeugt werden, können Sie dies durch mehrere isolierte Dienstprogrammläufe realisieren.

12.5.3 CSV-Ausgabe in BOUTLOAD-Utility

Mit folgender Anweisung geben Sie Daten im CSV-Format in BOUTLOAD aus:

```
COPY-RECORD . . . , CSV-OUTPUT = *YES
```

Die Daten werden zusätzlich zur Ausgabe in der Ausgabedatei im CSV-Format ausgegeben.

Es ist nicht zwingend, die CSV-Ausgabedatei zu erzeugen. Sie wird immer von der BOUTLOAD-Utility auf einer öffentlichen Platte erzeugt (siehe auch Handbuch "[Aufbauen und Umstrukturieren](#)").

12.5.4 Nutzungsbeispiele

In der Liefereinheit UDS-SQL-T und auf der mit UDS/SQL ausgelieferten CD „UDS/SQL Tools and Drivers“ finden sie Beispiele zur Nutzung der CSV-Ausgabe.

12.6 Auftragsschalter

In bestimmten Situationen setzen die UDS/SQL-Dienstprogramme Auftragsschalter bzw. werten vom Anwender gesetzte Auftragsschalter aus:

- Schalter 29: Wird von den UDS/SQL-Dienstprogrammen beim Ein-/Ausschalten des Loggings und bei Fehlern auf der Logdatei ausgewertet. Wurde vom Anwender dieser Schalter gesetzt, dann gibt das Dienstprogramm Konsolmeldungen aus.
- Schalter 30: Tritt während des Dienstprogrammlaufs eine WARNING auf, so setzt das Dienstprogramm den Schalter 30. Dieser bleibt nach Dienstprogrammende gesetzt und kann vom Anwender ausgewertet werden.
- Schalter 31: Tritt während des Dienstprogrammlaufs ein ERROR oder SYSTEM ERROR auf, so setzt das Dienstprogramm den Schalter 31. Dieser bleibt nach Dienstprogrammende gesetzt und kann vom Anwender ausgewertet werden.

Weitere Informationen zu den Auftragsschaltern finden Sie im Handbuch „[Sichern, Informieren und Reorganisieren](#)“ bei den Dienstprogrammen BCHECK und BREORG.

13 Einsatz von IQS

Wenn das Dialogsystem IQS eingesetzt wird, so ist der Datenbankadministrator dafür verantwortlich, dass jeder IQS-Anwender einwandfrei mit der Datenbank arbeiten kann.

Dazu müssen Sie Folgendes tun:

- in einer Kennung alle Objektmodule, Lademodule und Prozeduren des Dialogsystems IQS eintragen
- beachten, dass das Unterprogramm-Modul des BS2000-SORT, das Unterprogramm-Modul des EDT und das Modul MFHSROUT in der System-Tasklib verfügbar ist
- den IQS-Anwender beim Beseitigen von Fehlern unterstützen.

13.1 Von IQS verwendete Dateien, Module und Prozeduren

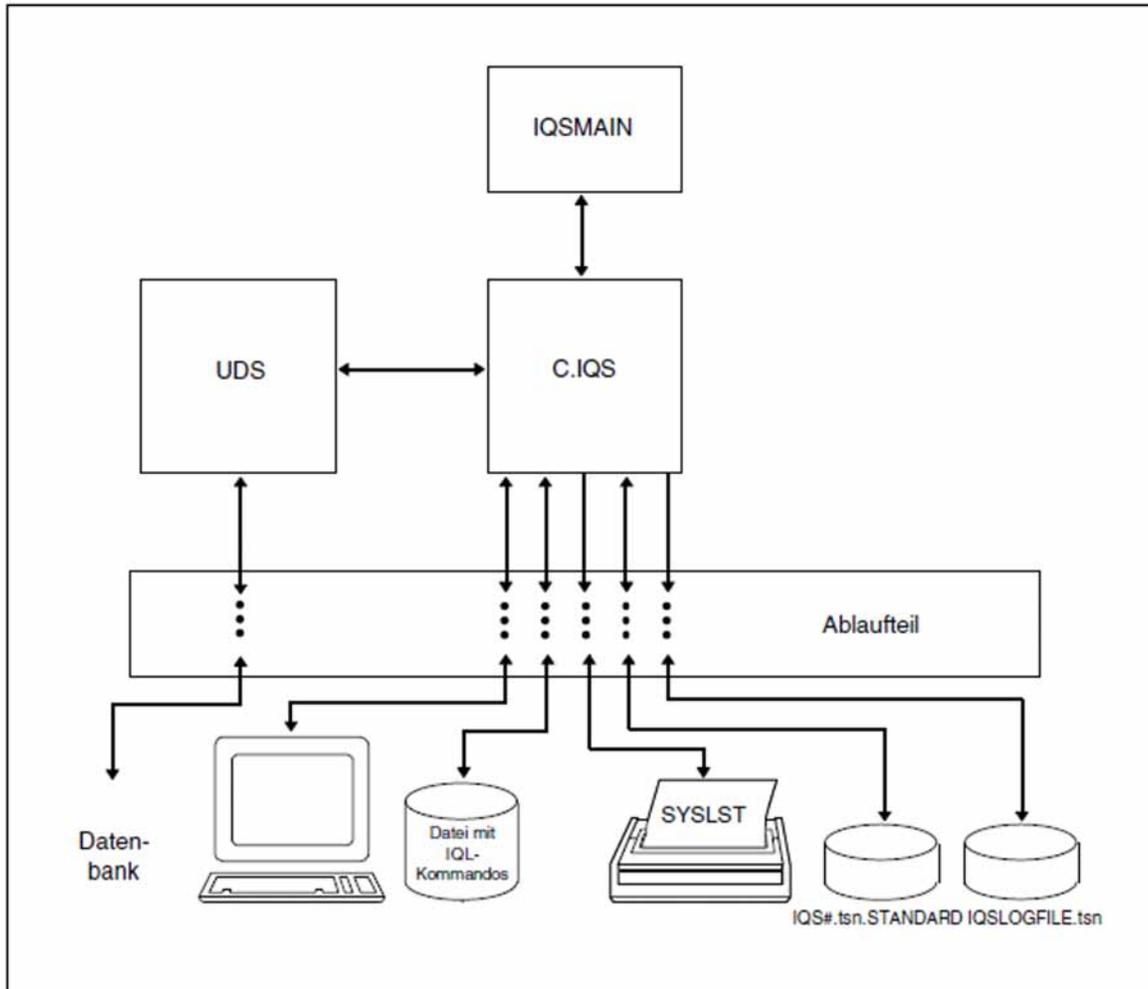


Bild 12: Zusammenwirken von IQS, UDS/SQL und Betriebssystem

C.IQS

IQS-Lademodul für independent DBH

C.L.IQS

IQS-Lademodul für linked-in DBH

IQSMAIN

IQL-Prozessor, ablauffähig als mehrfach benutzbarer Code im Klasse-4- bzw. Klasse-5-Speicher (siehe Handbuch „[Verwaltung von Subsystemen in BS2000](#)“). Er wird vom IQS automatisch aus der Modulbibliothek SYSLNK.IQS.040 nachgeladen. Sein Hauptspeicherbedarf beträgt ca. 188 Kbyte.

IQSGERM

IQSENGL

IQL-Sprachmodule, die die IQL-Meldungen/Fehlermeldungen enthalten. Die Module werden aus der SYSLNK.IQS.040 nachgeladen und können in den Klasse-4- bzw. Klasse-5-Speicher gelegt werden.

IQS#.tsn.STANDARD

Temporäre Arbeitsdatei, die beim FINDE-Kommando automatisch eingerichtet und bei normaler Beendigung der IQS-Sitzung wieder gelöscht wird.

Wird IQS abnormal beendet, muss die Datei mit dem DELETE-FILE-Kommando gelöscht werden.

IQS.LOGFILE.tsn

Protokolldatei, die die Protokolle der IQS-Sitzungen aufnimmt.

SYSPRG.IQS.040

Bibliothek, die die IQS-Lademodule enthält.

SYSLNK.IQS.040

Bindemodulbibliothek, die u.a. den IQL-Prozessor IQSMAIN enthält.

IQS.PRC.START

Startprozedur für IQS. Diese Prozedur steht in der Bibliothek SYSPRG.IQS.040.

IQS lädt die Bindemodule, die es für UDS/SQL benötigt, aus einer Bibliothek nach, die mit dem Link-Namen UDSOML über das Kommando ADD-FILE-LINK zugewiesen wird.

```
/ADD-FILE-LINK LINK-NAME=UDSOML,FILE-NAME= . . . ,
```

Die weiteren Verbindungsmodule des independent DBH bzw. der linked-in-DBH werden wie bei anderen Anwendungsprogrammen nachgeladen (siehe Handbuch „[Anwendungen programmieren](#)“).

13.2 Hilfsmittel zur Fehlerdiagnose

IQS erstellt zur Fehlerdiagnose ein umfangreiches IQS-Protokoll mit BIB-Logging, wenn innerhalb der IQS-Sitzung SETZE PROTOKOLL=BIB verwendet wurde. Dieses Protokoll kann mit dem Auswerteprogramm BWRLOG (siehe Handbuch „[Dialogsystem IQS](#)“, BWRLOG) ausgedruckt werden. (Nur im Fehlerfall verwenden, da erheblicher Papierverbrauch.)

i Bei fehlerhafter Behandlung eines IQS-Kommandos erzeugen Sie bitte folgende Diagnoseunterlagen:

- IQS-Protokoll (SETZE PROTOKOLL=BIB)
- DDL- und SSL-Eingabedatei
- LMS-Inhaltsverzeichnis der SYSLNK.IQS.040

bei abnormaler Beendigung zusätzlich: Dump

14 Einsatz von UDS-D

Dieses Kapitel beschreibt UDS-D, ein Zusatzprodukt zu UDS/SQL. Mit UDS-D ist es möglich, UDS/SQL-Datenbanken auf verschiedenen Rechnern zu bearbeiten. Das Kapitel enthält im Einzelnen:

- eine Beschreibung des Produkts mit seinen Merkmalen sowie Hinweise zu dem Einsatz
- einen Überblick über die Möglichkeiten und über die allgemeine Arbeitsweise von UDS-D
- eine Beschreibung des Verfahrens zur netzweiten Konsistenzsicherung
- die für den Datenbankbetrieb mit verteilten Datenbanken notwendigen Zusatzinformationen.

14.1 Kurzbeschreibung des Produkts

UDS-D als Zusatzprodukt zu UDS/SQL ermöglicht die Verarbeitung und Verwaltung von UDS/SQL-Datenbanken in homogenen BS2000-Rechnernetzen.

UDS-D hebt die starre Bindung von Datenbanken und Anwendungen zum gleichen Rechner auf und mindert dadurch dessen Belastung. Die Daten lassen sich an dem Rechner halten, an dem sie am meisten benützt werden. Andere Rechner können diese Daten bei Bedarf abfragen oder ändern. Beispielsweise können zentrale Dienststellen und Zweigstellen eines Betriebs auf den gesamten Datenbestand der verteilten Datenbanken zugreifen.

UDS-D unterstützt uneingeschränkt die verteilte Transaktionsverarbeitung mit netzweiter Deadlock-Erkennung und -Behandlung bei COBOL- und CALL-DML-Anwendungen.

UDS-D ist bedienungsfreundlich aufgebaut. An den Anwenderschnittstellen ist die Verteilung nicht sichtbar, das bedeutet einerseits, der Programmierer muss nicht wissen, welche Daten auf welchem Rechner gespeichert sind, und andererseits können Anwenderprogramme unverändert an allen Rechnern des Netzes eingesetzt werden.

Daraus ergeben sich folgende Vorteile bei der Anwendung von verteilten Datenbanken mit UDS-D:

- Verschiedene organisatorische Konzepte verwirklichen:
 - Verteilte Datenbanken als Abbild dezentraler Organisationsstrukturen. Die Daten werden dort bearbeitet, wo sie anfallen (Dezentralisierung).
 - Zusammenwachsen von isolierten Anwendungen zu einer rechnerübergreifenden DB- bzw. DB/DC-Anwendung.
- Trotz Dezentralisierung den gesamten Datenbestand im direkten Zugriff haben, sodass z.B. Ad-hoc-Abfragen und -Auswertungen leicht möglich sind.
- Rechnerengpässe durch Hinzunahme eines weiteren Rechners vermeiden.
- Die Verfügbarkeit bei Systemausfällen erhöhen.

UDS/SQL kann - sowohl mit als auch ohne UDS-D - mit dem verteilten Transaktionsmonitor openUTM-D zusammenarbeiten.

Die Verteilungskonzepte von UDS-D und openUTM-D sind unterschiedlich:

UDS-D

- Verteilung von Datenbanken
- Kommunikation pro DML-Anweisung
- Verteilung für das Anwenderprogramm nicht sichtbar
- Multi-DB-Programme kompatibel zu Programmen für verteilte Verarbeitung

openUTM-D

- Verteilung von Anwendungen
- Kommunikation pro Transaktion
- Verteilung für das Anwenderprogramm sichtbar

14.1.1 Merkmale von UDS-D

UDS-D zeichnet sich durch folgende Merkmale aus:

- Zugriff auf mehrere Datenbankkonfigurationen

Ein Anwenderprogramm kann über UDS-D auf Datenbanken mehrerer Mono- oder Multi-DB-Konfigurationen zugreifen.

Die DB-Konfigurationen können auf geografisch getrennte Verarbeitungsrechner verteilt sein, die über ein Netz miteinander verbunden sind.

Für die Art der Verteilung gibt es keine Einschränkungen. Sie richtet sich ausschließlich nach den Bedürfnissen des Anwenders. Grundsätzlich sollte sich jedoch jede Datenbank dort befinden, wo sie am häufigsten bearbeitet wird. Bei einem Zugriff auf Datenbanken entfernter DB-Konfigurationen werden die Daten über das Netz transportiert. In diesem Fall hängen die Antwortzeiten in erster Linie von den Übertragungseinrichtungen im Netz ab. Deshalb können Zugriffe über das Netz erheblich langsamer sein, als Zugriffe auf Datenbanken der lokalen DB-Konfiguration.

- Ortsunabhängigkeit der Anwenderprogramme

Der Anwender braucht die Verteilung der Datenbanken nicht zu kennen. Dem Anwenderprogramm müssen grundsätzlich nur die Namen der zu bearbeitenden Subschemas bekannt sein. Nicht bekannt sein müssen die dazugehörige DB-Konfiguration und der entsprechende Verarbeitungsrechner.

Anwenderprogramme können mit Datenbanken arbeiten, ohne zu wissen, an welche DB-Konfiguration sie angeschlossen sind. Das bedeutet, dass Anwenderprogramme keine verteilungsspezifischen Informationen enthalten.

DML-Anweisungen werden anhand einer Verteiltabelle zu der DB-Konfiguration an dem Verarbeitungsrechner geschickt, wo die angesprochene Datenbank liegt.

- Konfigurationsübergreifende Konsistenz und Deadlockauflösung

UDS-D garantiert Ihnen die netzweite logische Konsistenz aller beteiligten Datenbanken.

Eine Transaktion, die in Datenbanken mindestens einer entfernten DB-Konfiguration geändert hat, wird von UDS-D so beendet, dass die Änderungen entweder auf den Datenbanken aller beteiligten DB-Konfigurationen durchgeführt werden oder auf keiner Datenbank. UDS-D verwendet dazu ein Zwei-Phasen-Ende-Protokoll (Two Phase Commit). Das Transaktionsende ist aufgeteilt in zwei Phasen. In der ersten Phase werden alle Änderungen der Transaktion ausfallsicher protokolliert. Erst nach erfolgreichem Abschluss der ersten Phase werden in der zweiten Phase alle Änderungen auf den Datenbanken festgeschrieben. So ist zu jedem Zeitpunkt des Transaktionsendes sichergestellt, dass entweder alle Änderungen festgeschrieben oder alle Änderungen zurückgesetzt werden können.

Konfigurationsübergreifende Deadlocks werden automatisch erkannt und aufgelöst.

14.2 Hinweise zum Einsatz von UDS-D

Im Folgenden werden Hinweise dazu gegeben, was Sie beim Entwerfen, Programmieren und Umstrukturieren von Datenbanken beachten müssen, die Sie in verteilten Transaktionen über UDS-D-Verbindungen ansprechen wollen.

14.2.1 Entwerfen und Programmieren

Das Zeitverhalten von Zugriffen auf entfernte Konfigurationen ist stark abhängig von der Länge und der Anzahl der zu übertragenden Daten. Daraus ergeben sich folgende Empfehlungen:

- Transaktionen sollten hauptsächlich Zugriffe auf lokale Datenbanken enthalten.
- Die Menge der zu übertragenden Daten verringert sich in COBOL-DML-Programmen, wenn der Programmierer möglichst kleine Subschemas verwendet.
Definitionen wie SUBSCHEMA=SCHEMA sollten Sie nicht verwenden!
- Die Menge der zu übertragenden Daten ist bei CALL-DML-Programmen geringer als bei COBOL-DML-Programmen, da bei CALL-DML nur der aktuelle Satz und nicht die gesamte RECORD-AREA übertragen wird.

Der Statuscode 141 „Ungültige Subschemadefinition“ kann bei verteilten Datenbanken folgende Ursachen haben:

- Das angesprochene Subschema ist
 - nicht in der lokalen Konfiguration enthalten und nicht in der Verteiltabelle angegeben.
 - in der Verteiltabelle angegeben, aber in der entsprechenden Konfiguration nicht enthalten.
 - in der Verteiltabelle angegeben, aber die entsprechende Konfiguration ist nicht erreichbar, weil der Rechner nicht erreichbar ist oder die Konfiguration gar nicht oder nicht mit eingeschaltetem UDS-D-Betrieb läuft.
 - in der Verteiltabelle enthalten, aber gesperrt bzw. die zugehörige Datenbank oder Konfiguration ist gesperrt.
 - in der lokalen Konfiguration nicht enthalten und in der lokalen Konfiguration wurde der UDS-D-Betrieb nicht gestartet.
- Die Anzahl der von dieser Transaktion angesprochenen entfernten Datenbanken überschreitet den Wert von PP DISDB.
- Eine Konfiguration ist auf Grund von Besonderheiten im Netz nur kurzzeitig nicht erreichbar. Daher ist es sinnvoll, die Transaktion zu wiederholen.

14.2.2 Umstrukturieren

Nach einer Umstrukturierung oder Neugenerierung einer Datenbank müssen Sie

- SSITAB-Module zu einem entfernten CALL-DML-Programm transportieren.
- COBOL-DML-Programme übersetzen und binden. Dafür werden Informationen aus dem COSSD gebraucht, der auf einem entfernten Verarbeitungsrechner liegen kann.

14.3 Arbeitsweise von UDS-D

Das folgende Bild zeigt an einem Beispiel, wie die Verteilung von UDS/SQL-Datenbanken durch eine Kopplung über UDS-D erreicht wird.

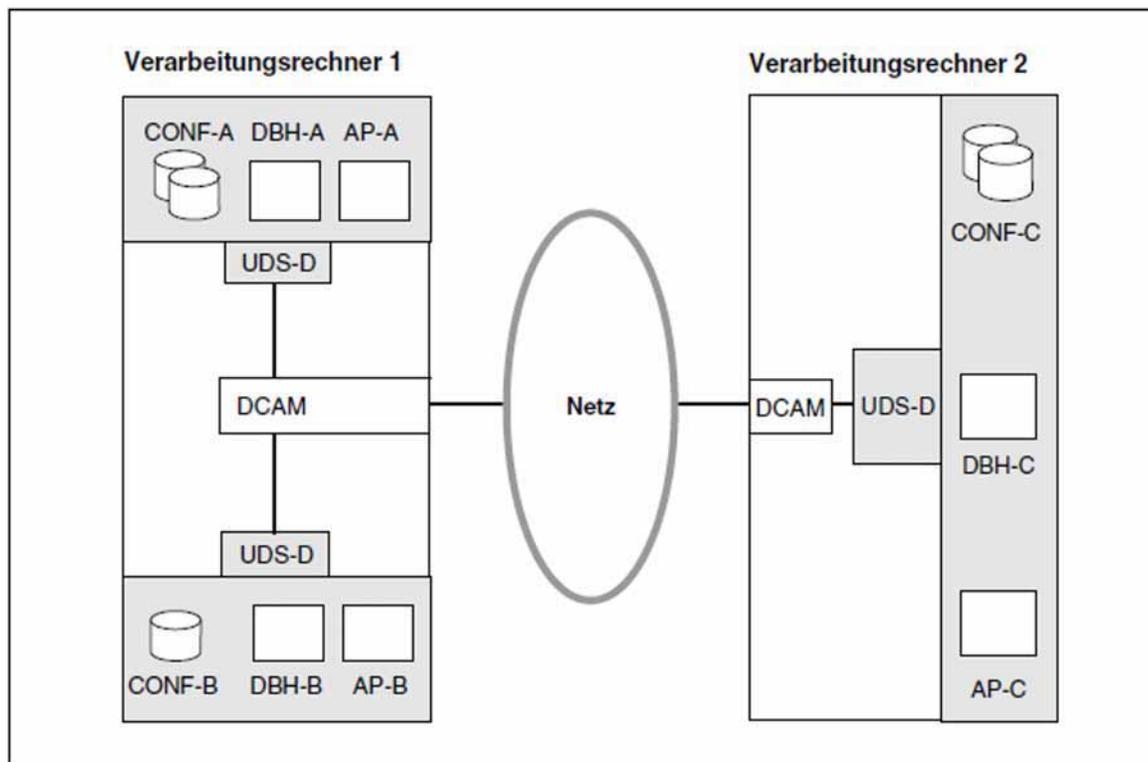


Bild 13: Verteilte Datenbanken mit UDS-D

Ein Anwenderprogramm wird beim Start an eine Konfiguration angeschlossen mit:

```
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIB=...,ELEM=...)
```

Diese Konfiguration wird bezüglich des Anwenderprogramms als **lokal** bezeichnet. Im **Bild 13** ist die Konfiguration **CONF-A** für das Anwenderprogramm **AP-A** lokal, die Konfiguration **CONF-B** für das Anwenderprogramm **AP-B**, usw. Alle anderen Konfigurationen sind **entfernte** Konfigurationen für das Anwenderprogramm. Die Konfigurationen **CONF-B** und **CONF-C** sind für das Anwenderprogramm **AP-A** entfernt. Umgekehrt ist für eine Konfiguration jedes Anwenderprogramm lokal, das an sie angeschlossen wurde. Alle anderen Anwenderprogramme sind für diese Konfiguration entfernt. Das Anwenderprogramm **AP-A** ist für die Konfiguration **CONF-A** lokal, die Anwenderprogramme **AP-B** und **AP-C** sind entfernt.

14.3.1 Das Teiltransaktionskonzept

Das Teiltransaktionskonzept ist eine Erweiterung des Multi-DB-Transaktionskonzepts (siehe Handbuch „Anwendungen programmieren“, Transaktion im Multi-DB-Betrieb).

Eine Verarbeitungskette ist eine Folge von DML-Anweisungen an eine einzige Datenbank innerhalb einer Transaktion.

In einer verteilten Transaktion bilden alle Verarbeitungsketten, die Datenbanken **einer** Konfiguration ansprechen, **eine** Teiltransaktion.

Beispiel

Gegeben sind die Konfigurationen und Datenbanken:

CONF-A	CONF-B
DB-A1	DB-B1
DB-A2	

Eine Transaktion kann folgende Teiltransaktionen und Verarbeitungsketten enthalten:

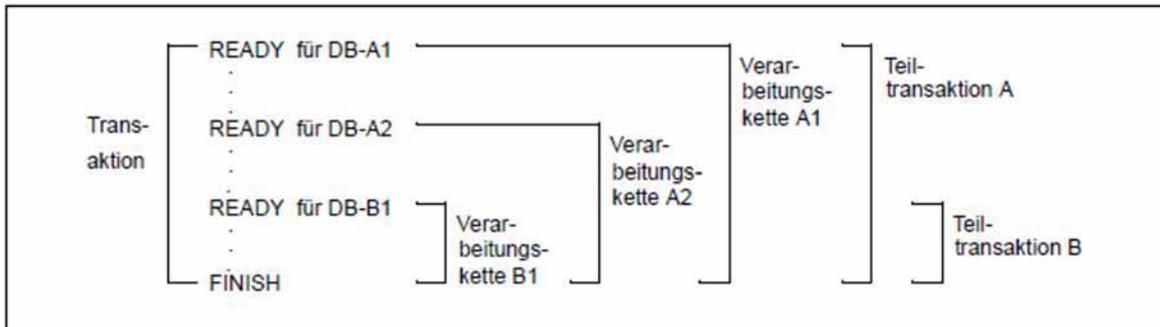


Bild 14: Transaktion, Teiltransaktionen und Verarbeitungsketten bei verteilten Datenbanken mit UDS-D

Im Bild 14 ist eine **verteilte Transaktion** dargestellt. Eine Transaktion heißt dann verteilt, wenn sie auf mindestens eine Datenbank in einer entfernten Konfiguration zugreift.

14.3.2 Primäre und sekundäre Teiltransaktionen

Alle Verarbeitungsketten, die Datenbanken **einer** DB-Konfiguration ansprechen, bilden **eine** Teiltransaktion.

Die Teiltransaktion, die die lokale Konfiguration anspricht, heißt primäre Teiltransaktion, alle übrigen Teiltransaktionen heißen sekundäre Teiltransaktionen (siehe Beispiel).

Das bedeutet: die primäre Teiltransaktion läuft immer in der lokalen Konfiguration ab und sekundäre Teiltransaktionen laufen immer in einer entfernten Konfiguration ab.

Beispiel

Gegeben sind die Konfigurationen und Datenbanken:

CONF-A	CONF-B	CONF-C
DB-A1	DB-B1	DB-C1
DB-A2		

Das Anwenderprogramm, das die folgende Beispieltransaktion ausgelöst hat, ist an die Konfiguration CONF-A angeschlossen. Daher gilt:

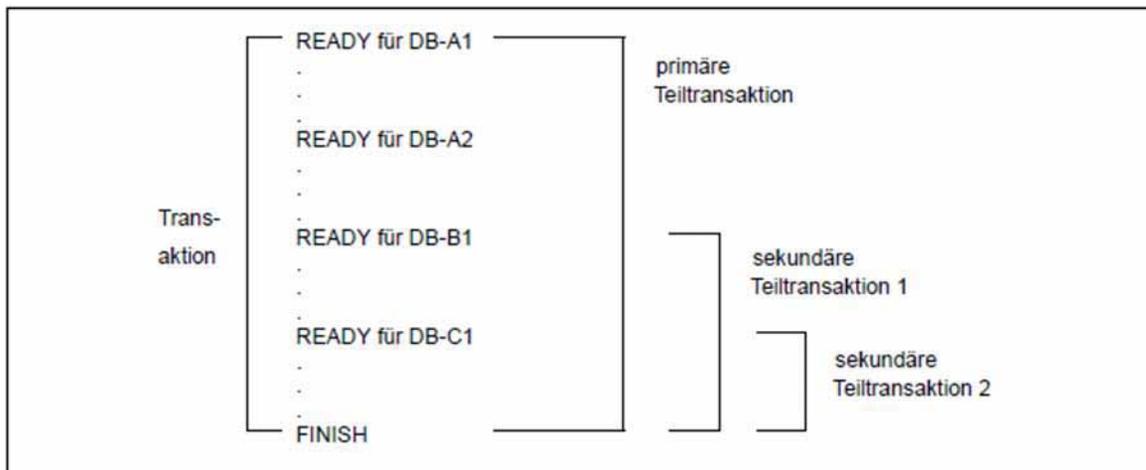


Bild 15: Primäre und sekundäre Teiltransaktionen

i Die erste READY-Anweisung einer Transaktion eröffnet die primäre Teiltransaktion. Falls die erste READY-Anweisung einer Transaktion eine entfernte Datenbank anspricht, erzeugt UDS-D eine so genannte Dummy-Teiltransaktion als primäre Teiltransaktion. Sobald eine READY-Anweisung dieser Transaktion eine lokale Datenbank anspricht, wird diese Dummy-Teiltransaktion zu einer echten primären Teiltransaktion.

14.3.3 Multi-DB-Programme für verteilte Verarbeitung

Multi-DB-Programme können unverändert für verteilte Verarbeitung verwendet werden. Ein Multi-DB-Programm greift auf Datenbanken zu, die zu einer DB-Konfiguration gehören. Mit UDS-D kann ein Multi-DB-Programm auf Datenbanken zugreifen, die zu verschiedenen DB-Konfigurationen gehören. Die DB-Konfigurationen wiederum können auf verschiedenen Verarbeitungsrechnern liegen.

Bild 16 zeigt ein Multi-DB-Programm. Die Verteilung der Datenbanken ist grau dargestellt. Sie hat auf das Multi-DB-Programm keinen Einfluss.

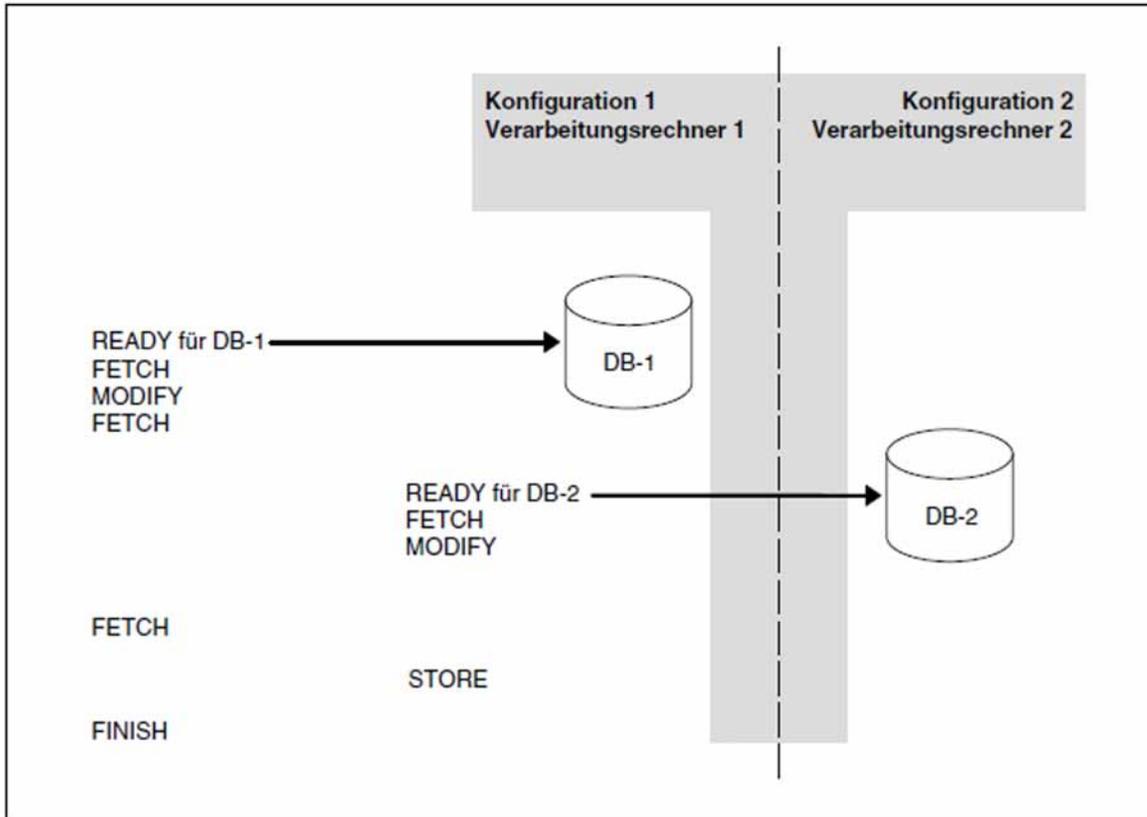


Bild 16: Multi-DB-Programm für verteilte Verarbeitung

14.3.4 Kommunikation zwischen Konfigurationen

In einer Konfiguration ist UDS-D im Wesentlichen realisiert durch

- die UDS-D-Task (UDSCT)
- eine Verteilkomponente in der Anwendertask (Modulname UDSNET).

In jeder Konfiguration, die mit entfernten Konfigurationen kommunizieren soll, muss der UDS-D-Betrieb laufen, d.h. die UDS-D-Task muss in jeder Konfiguration gestartet sein.

Die UDS-D-Task übernimmt die Kommunikationsaufgaben, die nötig sind, um DML-Anweisungen zu bearbeiten, die von entfernten Anwenderprogrammen kommen.

Die Verteilkomponente in der Anwendertask wird bei Einsatz von UDS-D durch UDS/SQL nachgeladen. Sie leitet DML-Anweisungen weiter, die von einem lokalen Anwenderprogramm kommen und in einer entfernten Konfiguration bearbeitet werden sollen. Das geschieht folgendermaßen:

Die UDS-D-Task richtet beim DBH-Start nach Ihren Angaben eine Verteiltabelle ein (siehe [Abschnitt „Die Verteiltabelle“](#)). In der Verteiltabelle ist zugeordnet:

- Subschema: Datenbank
- Datenbank: Konfiguration und
- Konfiguration: Verarbeitungsrechner und Kennwort.

Wenn der UDS-D-Betrieb gestartet ist, informiert sich die Verteilkomponente in der Anwendertask bei jeder READY-Anweisung in der Verteiltabelle, wohin die DML-Anweisungen dieser Verarbeitungskette geschickt werden sollen.

Die Kommunikation zwischen zwei Konfigurationen findet statt durch DCAM-Anwendungen, die über logische Verbindungen miteinander verkehren. Das gilt auch, wenn die Konfigurationen im gleichen Verarbeitungsrechner liegen.

In [Bild 17](#) ist die Kommunikation zwischen zwei Konfigurationen dargestellt, wenn in einer Konfiguration ein Anwenderprogramm verteilt arbeitet.

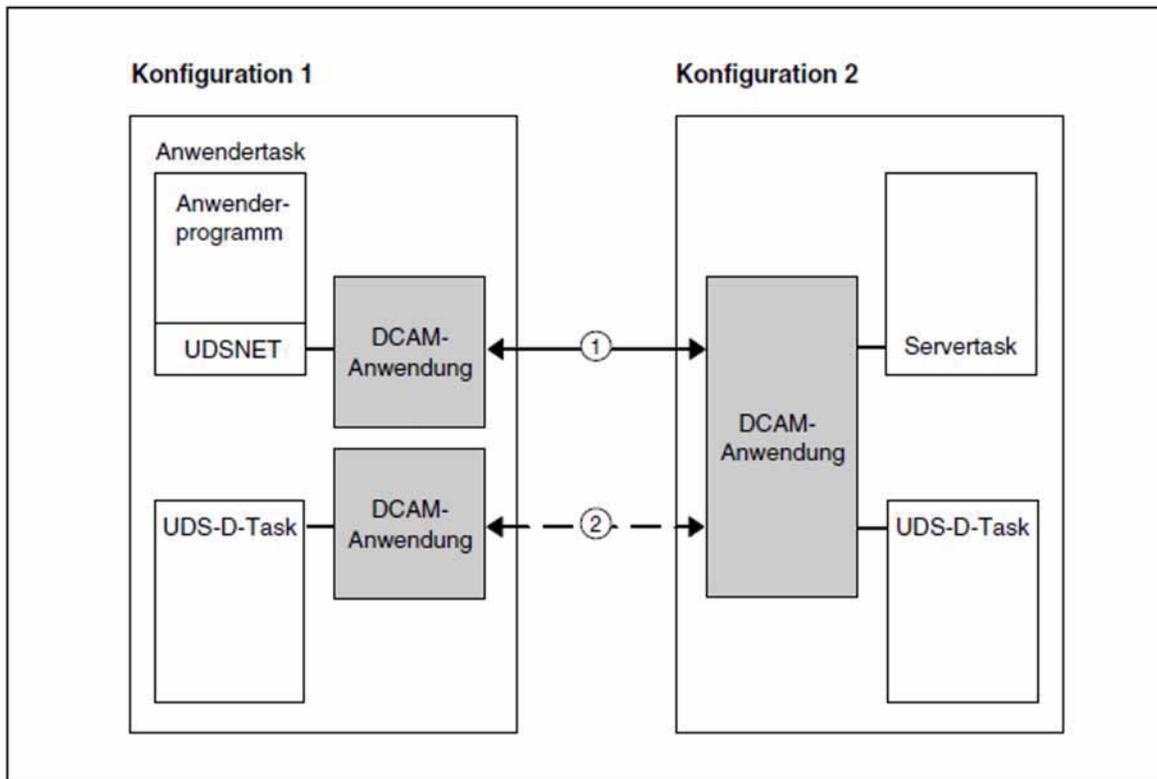


Bild 17: DCAM-Anwendungen und logische Verbindungen bei verteilten Datenbanken mit UDS-D

1. Diese logische Verbindung dient zur Bearbeitung von DML-Anweisungen. Sie wird einmal pro Anwendertask aufgebaut, sobald das Anwenderprogramm eine entfernte Konfiguration zum ersten Mal anspricht.
2. Diese logische Verbindung dient zur Überwachung und zum Warmstart. Sie wird einmal pro Konfiguration bei Bedarf aufgebaut und wieder abgebaut, wenn sie nicht mehr gebraucht wird.

14.4 Konfigurationsübergreifende Konsistenz mit dem Zwei-Phasen-Ende-Protokoll sichern

UDS-D beendet jede verteilte Transaktion so, dass Änderungen entweder auf den Datenbanken aller beteiligten DB-Konfigurationen durchgeführt werden oder auf keiner Datenbank.

Das Anwenderprogramm beendet eine Transaktion mit einer FINISH-Anweisung. Diese FINISH-Anweisung wird automatisch an alle Konfigurationen gesendet, in denen Teiltransaktionen eröffnet wurden.

Damit eine verteilte Transaktion, die in mindestens einer Datenbank einer entfernten DB-Konfiguration geändert hat, auch im Fehlerfall konsistent beendet werden kann, ist das Transaktionsende in zwei Phasen aufgeteilt:

- In der ersten Phase lässt UDS-D die geänderten Seiten jeder Teiltransaktion ausfallsicher in der RLOG-Datei der jeweiligen Konfiguration protokollieren.
Nach erfolgreichem Ablauf der ersten Phase ist sichergestellt, dass alle Teiltransaktionen auf die gleiche Weise abschließen können, entweder mit FINISH oder mit FINISH WITH CANCEL.
- In der zweiten Phase beendet UDS-D alle Teiltransaktionen mit FINISH, wenn die erste Phase in allen Konfigurationen erfolgreich war, oder mit FINISH WITH CANCEL, wenn die erste Phase in mindestens einer Konfiguration nicht erfolgreich war.

Diese Vorgehensweise heißt Zwei-Phasen-Ende-Protokoll (Two Phase Commit).

Bei logischem Verbindungsverlust oder Rechnerausfall während der Beendigungsphase einer verteilten Transaktion sorgt UDS-D dafür, dass die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-openUTM-Konsistenz erhalten bleibt (z.B. bei einem Session-Wiederanlauf oder im Rahmen der Transaktionsüberwachung, siehe [Abschnitt „Überwachen sekundärer Teiltransaktionen, die im Zustand PTC sind“](#)).

Der Ablauf des Zwei-Phasen-Ende-Protokolls hängt davon ab, wie eine Transaktion verteilt ist:

- nur über UDS-D (siehe [Abschnitt „Ablauf des Zwei-Phasen-Ende-Protokolls bei Verteilung über UDS-D“](#)) oder
- über UDS-D und openUTM (siehe [Abschnitt „Ablauf des Zwei-Phasen-Ende-Protokolls in UDS-D bei Verteilung über UDS-D und openUTM“](#)).

Verteilung über UDS-D

Die Verteilkomponente in der Anwendertask steuert das Zwei-Phasen-Ende-Protokoll. Die primäre Teiltransaktion kommt nicht in den Zustand PTC (PTC: Prepared to Commit, siehe [Abschnitt „Zustand PTC“](#)).

Verteilung über UDS-D und openUTM

openUTM steuert das Zwei-Phasen-Ende-Protokoll. Auch die primäre Teiltransaktion kommt in den Zustand PTC.

14.4.1 Ablauf des Zwei-Phasen-Ende-Protokolls bei Verteilung über UDS-D

Die FINISH-Anweisung im Anwenderprogramm leitet das Zwei-Phasen-Ende-Protokoll ein.

Beim Zwei-Phasen-Ende-Protokoll werden Teiltransaktionen mit und ohne UPDATE-Kennzeichnung unterschieden. Jede Teiltransaktion hat zunächst unabhängig vom Eröffnungsmodus (READY) der Datenbank(en) keine UPDATE-Kennzeichnung. Erst nach erfolgreicher Ausführung einer ändernden DML-Anweisung in einer Datenbank erhält eine Teiltransaktion die UPDATE-Kennzeichnung.

Erste Phase des Zwei-Phasen-Ende-Protokolls

1. Die Verteilkomponente in der Anwendertask sendet
 - die FINISH-Anweisung an alle sekundären Teiltransaktionen ohne UPDATE-Kennzeichnung
 - eine PETA-Anweisung an alle sekundären Teiltransaktionen mit UPDATE-Kennzeichnung (PETA: Preliminary End of Transaction)

Die PETA-Anweisung bewirkt, dass in der RLOG-Datei der jeweiligen Konfiguration ausfallsicher gespeichert werden:

- alle Seiten, die von der jeweiligen sekundären Teiltransaktion geändert wurden,
- die Rücksetzinformationen und
- die Namen aller beteiligten Konfigurationen für den Fall, dass ein Warmstart nötig wird.

In den noch folgenden Schritten ist mit „sekundäre Teiltransaktion“ grundsätzlich „sekundäre Teiltransaktion mit UPDATE-Kennzeichnung“ gemeint, da die übrigen sekundären Teiltransaktionen bereits abgeschlossen sind.

2. Die sekundären Teiltransaktionen führen die PETA-Anweisung durch und melden der primären Teiltransaktion, ob die PETA-Anweisung ohne Fehler durchgeführt werden konnte oder nicht.
3. Wenn die primäre Teiltransaktion von jeder sekundären Teiltransaktion eine Meldung erhalten hat, prüft sie, ob die Durchführung der PETA-Anweisung bei allen sekundären Teiltransaktionen erfolgreich war.

Abhängig davon wird 4a. oder 4c. durchgeführt.

i Falls die primäre Teiltransaktion nach Ablauf einer von PP CHCKTIME abhängigen Zeitspanne nicht von allen sekundären Teiltransaktionen eine Meldung erhalten hat, wird die primäre Teiltransaktion mit FINISH WITH CANCEL beendet und die sekundären Teiltransaktionen werden aufgefordert, sich ebenfalls mit FINISH WITH CANCEL zu beenden. Das Anwenderprogramm erhält den Statuscode 122.

Zweite Phase des Zwei-Phasen-Ende-Protokolls

4a.

Wenn alle sekundären Teiltransaktionen die PETA-Anweisung erfolgreich durchgeführt haben, wird die primäre Teiltransaktion mit Festschreiben der Änderungen (FINISH) beendet. Wenn der FINISH der primären Teiltransaktionen gelingt, wird 4b. ausgeführt. Falls der FINISH der primären Teiltransaktion nicht erfolgreich durchgeführt werden kann, werden die sekundären Teiltransaktionen zurückgesetzt.

4b.

Alle sekundären Teiltransaktionen werden aufgefordert, sich mit FINISH zu beenden.

4c.

Wenn nicht alle sekundären Teiltransaktionen die PETA-Anweisung erfolgreich durchführen konnten, wird die primäre Teiltransaktion mit Zurücksetzen der Änderungen beendet (FINISH WITH CANCEL) und es werden die sekundären Teiltransaktionen aufgefordert, sich ebenfalls mit FINISH WITH CANCEL zu beenden.

14.4.2 Ablauf des Zwei-Phasen-Ende-Protokolls in UDS-D bei Verteilung über UDS-D und openUTM

Erste Phase des Zwei-Phasen-Ende-Protokolls

1. openUTM sendet eine PETA-Anweisung an UDS/SQL. Die Verteilkomponente in der Anwendertask sendet
 - eine FINISH-Anweisung an alle sekundären Teiltransaktionen ohne UPDATE-Kennzeichnung und
 - die PETA-Anweisung an alle sekundären Teiltransaktionen mit UPDATE-Kennzeichnung.

Die PETA-Anweisung bewirkt, dass ausfallsicher in der RLOG-Datei der jeweiligen Konfiguration gespeichert werden

- alle Seiten, die von der jeweiligen sekundären Teiltransaktion geändert wurden,
- die Rücksetzinformationen und
- die Namen aller beteiligten Konfigurationen für den Fall, dass ein Warmstart nötig wird.

In den noch folgenden Schritten ist mit „sekundäre Teiltransaktion“ grundsätzlich „sekundäre Teiltransaktion mit UPDATE-Kennzeichnung“ gemeint, da die übrigen sekundären Teiltransaktionen bereits abgeschlossen sind.

2. Die sekundären Teiltransaktionen führen die PETA-Anweisung durch und melden der Verteilkomponente in der Anwendertask, ob die PETA-Anweisung ohne Fehler durchgeführt werden konnte oder nicht.
3. Wenn die Verteilkomponente in der Anwendertask von jeder sekundären Teiltransaktion eine Meldung erhalten hat, prüft sie, ob die Durchführung der PETA-Anweisung bei allen Teiltransaktionen erfolgreich war.

Abhängig davon wird 4a. oder 4c. durchgeführt.

i Falls die Verteilkomponente in der Anwendertask nach Ablauf einer von PP CHCKTIME abhängigen Zeitspanne nicht von allen Teiltransaktionen eine Meldung erhalten hat, werden alle Teiltransaktionen aufgefordert, sich mit FINISH WITH CANCEL zu beenden. openUTM erhält den Returncode „Transaktion zurückgesetzt“.

4a.

Wenn alle Teiltransaktionen die PETA-Anweisung erfolgreich durchgeführt haben, wird auch für die primäre Teiltransaktion eine PETA-Anweisung abgesetzt. Wird diese fehlerfrei durchgeführt, so befindet sich die ganze DB-Transaktion im Zustand PTC (PTC: Prepared to Commit, siehe [Abschnitt „Zustand PTC“](#)). Der Zustand PTC wird an openUTM gemeldet, openUTM erhält die Steuerung und kann die zweite Phase einleiten. Wenn die PETA-Anweisung nicht erfolgreich durchgeführt werden kann, wird 4b. ausgeführt.

Zweite Phase des Zwei-Phasen-Ende-Protokolls

4b.

Alle sekundären Teiltransaktionen werden aufgefordert, sich mit FINISH WITH CANCEL zu beenden. openUTM erhält den Returncode „Transaktion zurückgesetzt“.

4c.

Wenn nicht alle sekundären Teiltransaktionen die PETA-Anweisung erfolgreich durchführen konnten, werden alle Teiltransaktionen - auch die primäre - aufgefordert, sich mit Zurücksetzen der Änderungen zu beenden (FINISH WITH CANCEL). openUTM erhält den Returncode „Transaktion zurückgesetzt“.

5. openUTM stimmt mit anderen Anwendungen ab, wie die Transaktion beendet werden soll und veranlasst UDS/SQL zum Festschreiben oder zum Zurücksetzen der Änderungen.

Die Beendigung der einzelnen Teiltransaktionen wird von der Verteilkomponente in der Anwendertask gesteuert.

14.4.3 Zustand PTC

Eine Teiltransaktion befindet sich im Zustand PTC (Prepared to Commit), wenn sie die PETA-Anweisung erfolgreich durchgeführt, aber noch keinen Auftrag zur Beendigung der Transaktion erhalten hat. Im Zustand PTC ist noch nicht entschieden, ob die gesamte Transaktion - also auch die Teiltransaktion selbst - mit FINISH oder mit FINISH WITH CANCEL beendet wird.

Bei einer Verteilung über UDS-D kommen nur sekundäre Teiltransaktionen mit UPDATE-Kennzeichnung in den Zustand PTC. Bei einer Verteilung über UDS-D und openUTM kommt auch die primäre Teiltransaktion in den Zustand PTC, falls sie selbst oder mindestens eine ihrer sekundären Teiltransaktionen die UPDATE-Kennzeichnung hat.

Die folgenden Bilder zeigen die Aktionen der einzelnen Teiltransaktionen und den Zustand PTC

- wenn das Zwei-Phase-Ende-Protokoll zum FINISH führt bei einer Transaktion, die über UDS-D verteilt ist (siehe Bild 18).
- wenn das Zwei-Phasen-Ende-Protokoll zum Zurücksetzen führt bei einer Transaktion, die über UDS-D verteilt ist (siehe Bild 19).
- wenn das Zwei-Phasen-Ende-Protokoll zum FINISH führt bei einer Transaktion, die über UDS-D und openUTM verteilt ist (siehe Bild 20).
- wenn das Zwei-Phasen-Ende-Protokoll zum Zurücksetzen führt bei einer Transaktion, die über UDS-D und openUTM verteilt ist (siehe Bild 21).

In den Bildern bedeutet:

PTT: primäre Teiltransaktion

STT: sekundäre Teiltransaktion

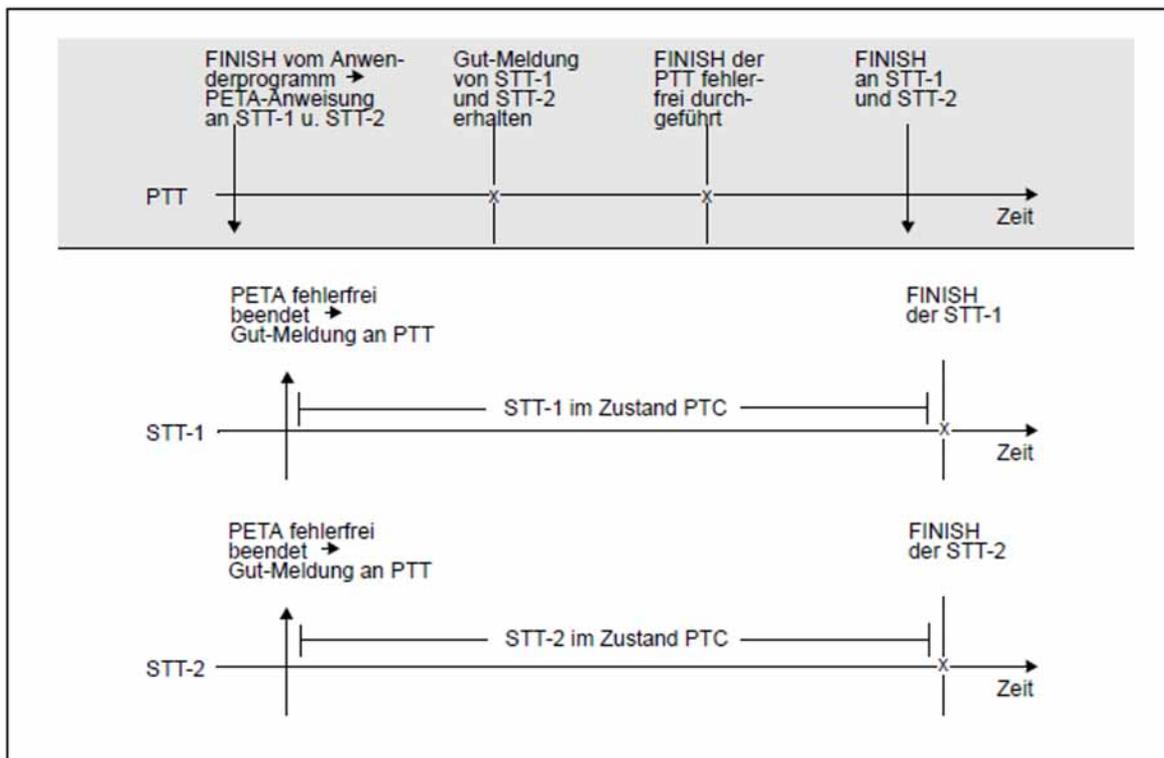


Bild 18: Zeitlicher Ablauf eines Zwei-Phasen-Ende-Protokolls mit FINISH bei UDS-D

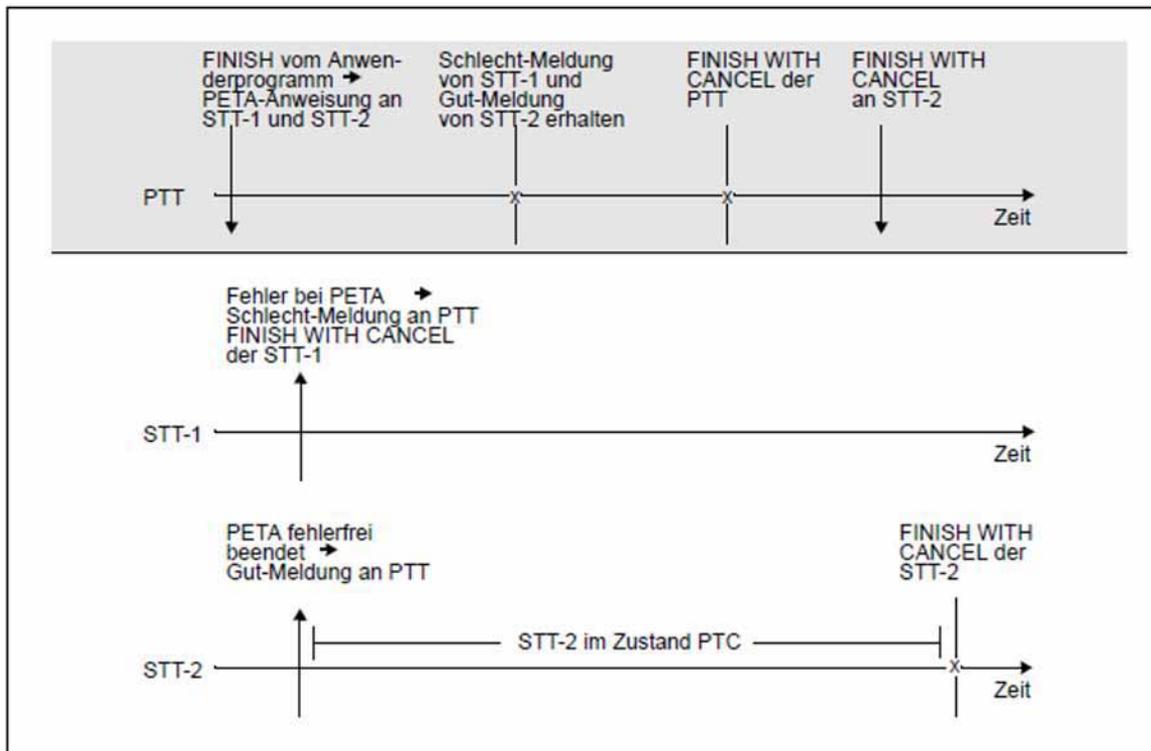


Bild 19: Zeitlicher Ablauf eines gescheiterten Zwei-Phasen-Ende-Protokolls bei UDS-D

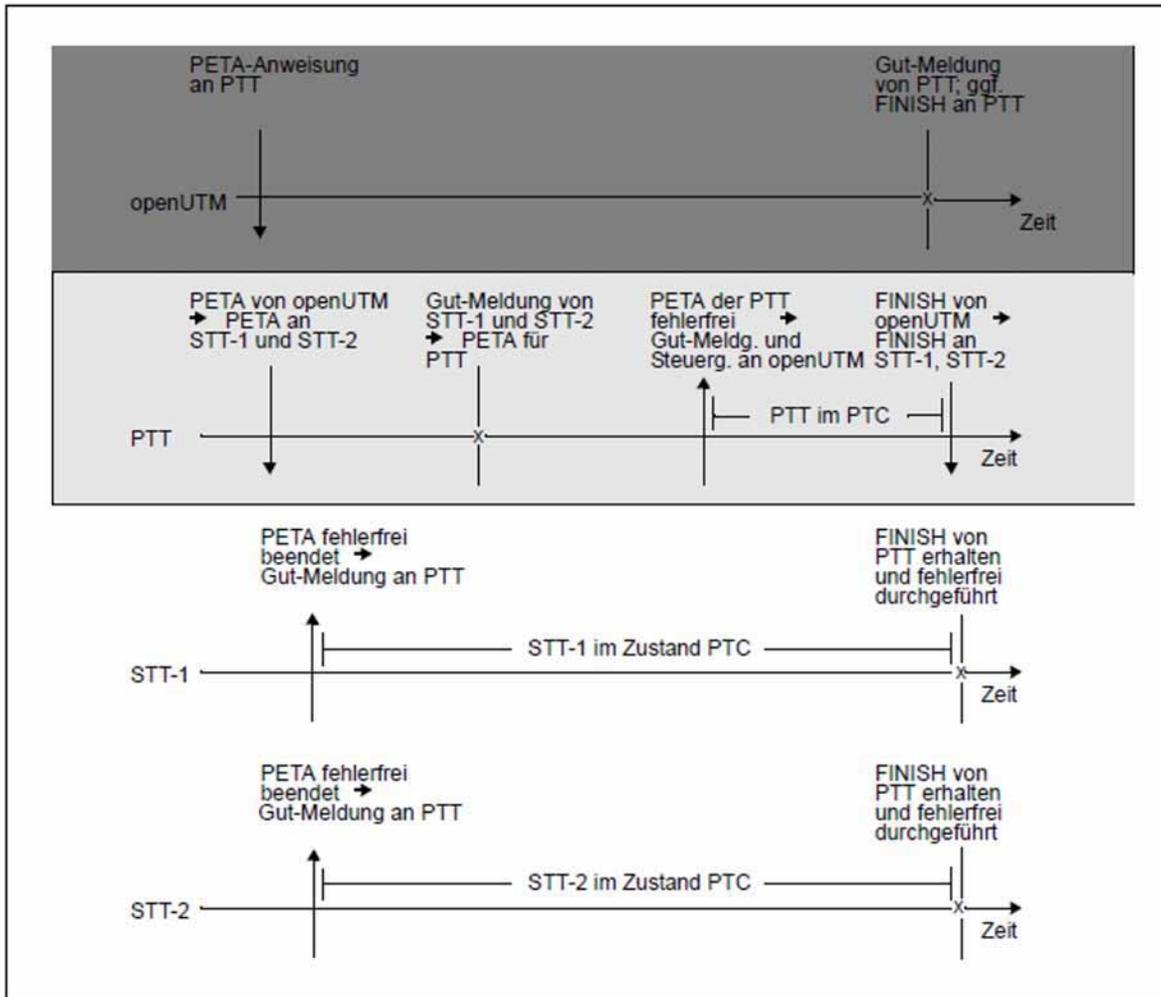


Bild 20: Zeitlicher Ablauf eines Zwei-Phasen-Ende-Protokolls mit FINISH bei Verteilung über UDS-D und openUTM

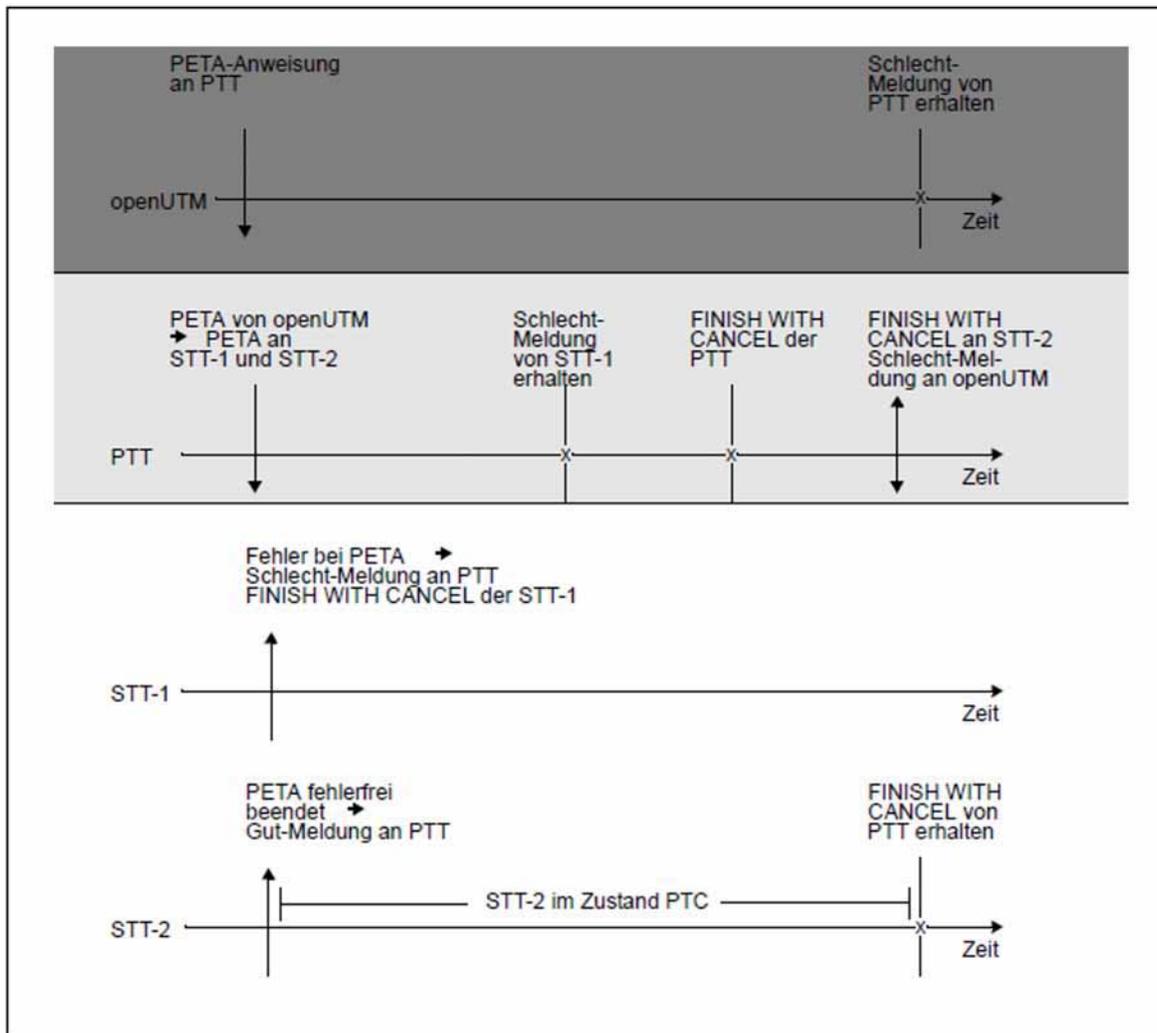


Bild 21: Zeitlicher Ablauf eines gescheiterten Zwei-Phasen-Ende-Protokolls bei Verteilung über UDS-D und openUTM

14.4.4 Zustand PTC beenden

Es ist möglich, dass Transaktionen, die sich im Zustand PTC befinden, z.B. wegen Partner- oder Leitungsausfall nicht beendet werden können. Wenn der Ausfall nicht in vertretbarer Zeit behoben werden kann, bietet UDS-D dem Datenbankadministrator die Möglichkeit, einzugreifen, um den Datenbankbetrieb fortzusetzen. Er kann

- mit dem DBH-Ladeparameter PP PTCSYNCH steuern, ob sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC nach einer von PP CHCKTIME abhängigen Zeitspanne mit FINISH oder mit FINISH WITH CANCEL beendet werden sollen. Der Datenbankadministrator kann den Wert des DBH-Ladeparameters PP PTCSYNCH auch während der Session verändern mit dem DAL-Kommando MODIFY.
- mit den DAL-Kommandos ABORT,OPTION=PTC oder COMMIT sekundäre Teiltransaktionen oder openUTM-Transaktionen „per Hand“ mit FINISH oder mit FINISH WITH CANCEL beenden.

! VORSICHT!

Beim Beenden des Zustands PTC gefährdet der Datenbankadministrator die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz, sobald

- der DBH-Ladeparameter PP PTCSYNCH nicht auf (WAIT,WAIT) steht oder
- er mit dem DAL-Kommando ABORT,OPTION=PTC oder COMMIT eingreift.

Wenn der Datenbankadministrator sekundäre Teiltransaktionen bzw. openUTM-Transaktionen im Zustand PTC beendet, können Änderungen derselben Transaktion in Datenbanken einer DB-Konfiguration eingebracht sein, in Datenbanken einer anderen DB-Konfiguration nicht.

UDS/SQL unterrichtet den Datenbankadministrator darüber, wie die Transaktion verteilt ist, damit er Maßnahmen treffen kann, um die Inkonsistenz einzugrenzen oder zu beheben.

Die Konsistenz innerhalb einer DB-Konfiguration bleibt unabhängig davon gesichert.

14.5 Datenbankbetrieb mit UDS-D

In diesem Abschnitt wird erläutert, was zum reibungslosen Betrieb mit verteilten Datenbanken erforderlich ist. Folgendes wird beschrieben:

- die UDS-D-spezifischen Teile des DBH
- die Abwicklung einer COBOL- bzw. CALL-DML-Anweisung, die ein entferntes Subschema bearbeitet
- der konfigurationsbezogene Kennwortschutz
- die Verteiltabelle
- die zeitgesteuerte Verbindungs- und Transaktionsüberwachung
- das Starten und Beenden des UDS-D-Betriebs
- die Auswirkungen des Zustands PTC
- die Administration mehrerer Konfigurationen
- Hinweise zum Ändern der DB-Konfiguration

14.5.1 DBH beim Einsatz von UDS-D

Nur der independent DBH kann zusammen mit UDS-D verteilte Datenbanken bearbeiten.

Beim Einsatz von UDS-D besteht der independent DBH aus folgenden Tasks:

- der Mastertask
- einer oder mehreren Servertasks
- der UDS-D-Task

Die Beschreibung der **Mastertask** und der **Servertasks** finden Sie im [Abschnitt „Arbeitsweise des independent DBH“](#).

Die **UDS-D-Task** (UDSCT) wird abhängig vom DBH-Ladeparameter PP DISTRIBUTION durch die Mastertask als ENTER-Job geladen.

Die UDS-D-Task richtet UDS-D-spezifische Tabellen und Speicherbereiche ein, z.B. den Distribution Pool mit der Verteiltabelle und den Transfer Pool zur Aufnahme von BIBs. Sie empfängt DML-Anweisungen von entfernten Anwenderprogrammen und reicht sie an eine Servertask weiter.

Die UDS-D-Task überwacht fortlaufend die logischen Verbindungen zu den entfernten Anwenderprogrammen (siehe [Abschnitt „Überwachen der logischen Verbindung zu entfernten Anwenderprogrammen“](#)).

Der **Transfer Pool** hat eine vergleichbare Funktion für DML-Anweisungen von entfernten Anwenderprogrammen wie der **Communication Pool** für DML-Anweisungen von lokalen Anwenderprogrammen. Er enthält

- die Transaktionskennung
- die DML-Anweisung mit ihren Parametern (BIB)
- die zu übertragenden Daten (BIB)

Der **Common Pool** ist ein Speicherbereich, auf den alle Aufträge des DBH zugreifen können. Eine genaue Beschreibung finden Sie im [Abschnitt „Arbeitsweise des independent DBH“](#).

Im **Distribution Pool** liegen die **Verteiltabelle** und UDS-D-spezifische Systemtabellen.

Aus der Verteiltabelle entnimmt die Verteilkomponente in der Anwendertask, ob das zu bearbeitende Subschema lokal oder entfernt ist.

[Bild 22](#) zeigt den independent DBH beim Einsatz von UDS-D. Eingezeichnet sind die wichtigsten Kommunikationswege, wenn ein entferntes Anwenderprogramm mit dem DBH kommuniziert.

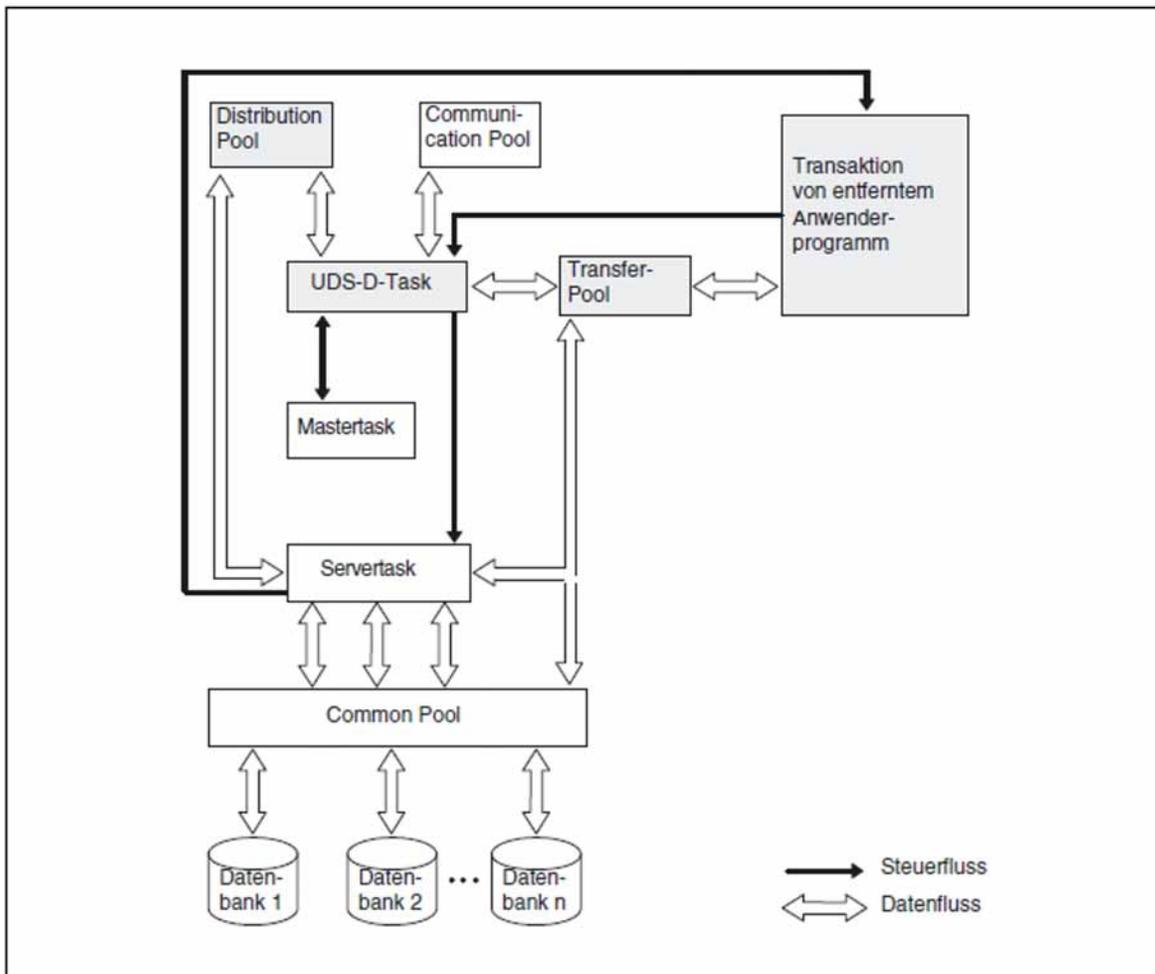


Bild 22: Der independent DBH beim Einsatz von UDS-D

Abwicklung einer COBOL- bzw. CALL-DML-Anweisung an ein entferntes Subschema

Bild 23 stellt dar, wie eine COBOL- bzw. CALL-DML-Anweisung des Anwenderprogramms abgewickelt wird, die ein entferntes Subschema bearbeitet.

Dabei ist zu beachten, dass im Falle einer CALL-DML-Anweisung für die weitere Bearbeitung des Auftrags ein BIB erzeugt wird (siehe Abschnitt „Bearbeiten einer CALL-DML-Anweisung durch den independent DBH“, „Arbeitsweise des independent DBH“). Dieser BIB dient als Eingabe für die Verteilkomponente UDSNET.

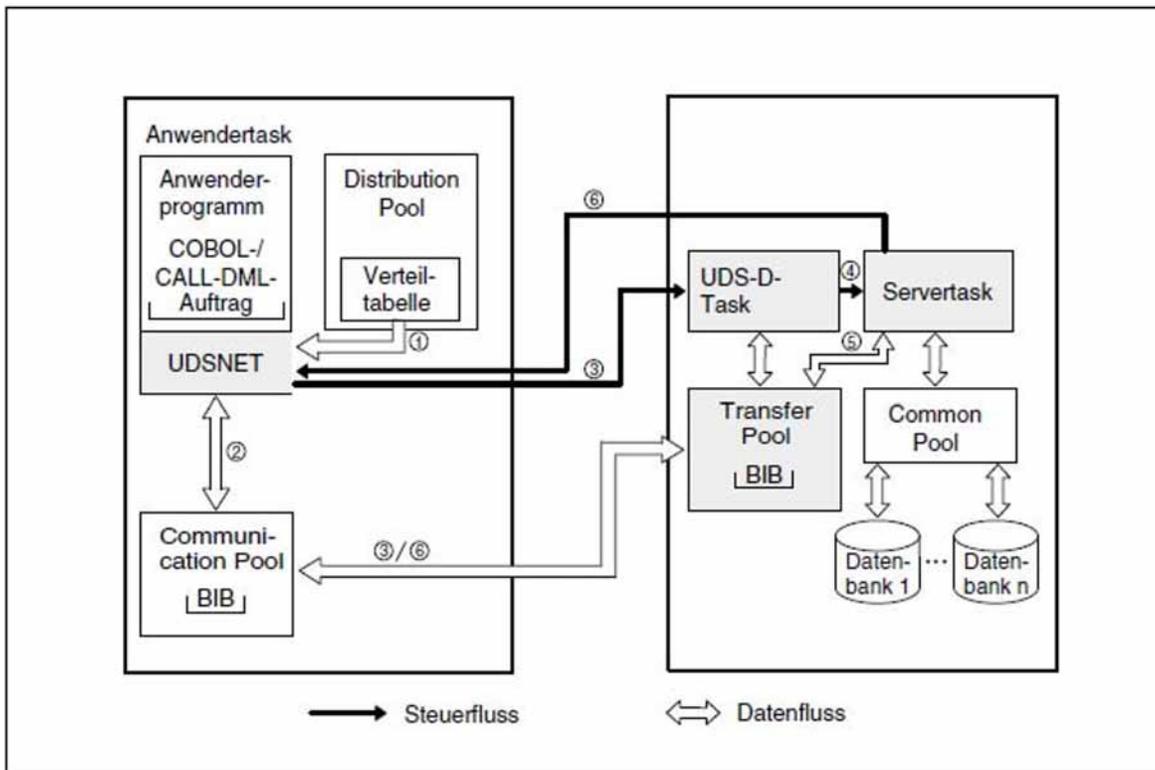


Bild 23: Abwicklung einer COBOL-/CALL-DML-Anweisung, die ein entferntes Subschema bearbeitet

1. Bei der zugehörigen READY-Anweisung ermittelt das Verbindungsmodul die Lage der entfernten Konfiguration über die Verteilertabelle.
2. Die Anwendungstask überträgt den BIB, der ein entferntes Subschema bearbeiten soll, zuerst in den Communication Pool.
3. Aus dem Communication Pool wird der BIB über die logische Verbindung in den Transfer Pool übertragen. Dadurch wird die entfernte UDS-D-Task geweckt. Die Anwendungstask selbst wartet auf das Eintreffen des bearbeiteten BIBs.
4. Der BIB wird einer derzeit nicht arbeitenden Servertask zugewiesen.
5. Die Servertask bearbeitet den BIB.
6. Der BIB wird aus dem Transfer Pool über die logische Verbindung in den Communication Pool übertragen. Dadurch wird der Wartezustand der Anwendungstask beendet.

14.5.2 Konfigurationsbezogener Kennwortschutz

Konzept

Jede Konfiguration kann durch ein konfigurationsbezogenes Kennwort vor unerwünschten Zugriffen aus den Anwendertasks anderer Konfigurationen geschützt werden. Das Kennwort, das der lokalen Konfiguration zugeordnet ist, dient als Lockcode, die den entfernten Konfigurationen zugeordneten Kennwörter als Keycodes für den Verbindungsaufbau (vgl. das openUTM „Zugriffskontrolle über openUTM“).

Bei jeder Aufforderung zum Verbindungsaufbau, die in einer Anwendertask angestoßen wird, gibt UDS-D dem Auftrag das entsprechende Keycode-Kennwort mit. In der entfernten Konfiguration wird dieses beim Eintreffen der Aufforderung in der UDS-D-Task mit dem betreffenden Lockcode verglichen. Wenn Lockcode und Keycode nicht übereinstimmen, wird die Aufforderung zum Verbindungsaufbau von der UDS-D-Task abgewiesen. Der Datenbankauftrag des Anwendungsprogramms, der zur Verbindungsaufforderung führte, wird abnormal beendet; wenn bereits eine Transaktion offen war, wird diese zurückgesetzt.

Kennwörter vergeben

Konfigurationsbezogene Kennwörter können sie in der Eingabedatei für die Verteiltabellen und über das DAL-Kommando &PWD DISTRIBUTION vergeben (siehe [Abschnitt „Die Verteiltabelle“](#) und [Abschnitt „Kennwort vergeben und verändern \(&PWD DISTRIBUTION\)“](#)).

Beispiel

Zwei UDS/SQL-Anwendungen laufen zeitlich überlappend, die eine in einer Testkonfiguration, die andere in einer Produktivkonfiguration. Um den späteren Übergang der Testanwendung in den Produktivbetrieb zu erleichtern, bestehen keine Unterschiede bezüglich Privacy-Informationen, Subschema- und Datenbanknamen zwischen den beiden Konfigurationen.

Dieses Vorgehen wird dadurch unterstützt, dass die Produktivkonfiguration gegenüber der Testkonfiguration mit einem konfigurationsbezogenen Kennwort geschützt wird. Wenn aus Versehen der Konfigurations- und Prozessurname der Produktivanwendung in die Verteiltabelle der Testanwendung eingetragen wird, so kann auf diese Weise dennoch ein Zugriff der Testanwendung auf die Produktiv-Datenbank(en) verhindert werden.

14.5.3 Die Verteiltabelle

Ein Anwenderprogramm kann über UDS-D auf Datenbanken mehrerer DB-Konfigurationen zugreifen. Das Anwenderprogramm kennt nur den Subschemaanamen. Deshalb muss UDS-D bekannt sein, ob sich das Subschema in der lokalen oder in einer entfernten DB-Konfiguration befindet.

Dazu erstellt der Datenbankadministrator für jede DB-Konfiguration, in der verteilte Transaktionen gestartet werden sollen, eine Eingabedatei. Aus der Eingabedatei baut UDS-D die so genannte Verteiltabelle auf. In der Eingabedatei und in der daraus entstehenden Verteiltabelle ist zugeordnet:

- Subschema: Datenbank
- Datenbank: Konfiguration und
- Konfiguration: Verarbeitungsrechner und Kennwort.

Die Verteiltabelle hinterlegt UDS-D im Distribution Pool, damit die Verteilkomponente in der Anwendertask während der Session darauf zugreifen kann. In der Verteiltabelle dürfen auch lokale Datenbanken und lokale Subschemata beschrieben sein. Dadurch ist es möglich, für alle DB-Konfigurationen gleiche Verteiltabellen bereitzustellen. Bei großen Verteiltabellen ist es vorteilhaft, lokale Datenbanken und Subschemata an den Anfang der Eingabedatei zu stellen, damit das Zeitverhalten für lokale Transaktionen günstig bleibt.

Datenbanken und Subschemata, die in der Verteiltabelle nicht beschrieben sind, gelten als lokal.

Auf entfernte Datenbanken und Subschemata, die in der Verteiltabelle nicht beschrieben sind, kann ein lokales Anwenderprogramm nicht zugreifen.

14.5.3.1 Aufbau der Eingabedatei für die Verteiltabelle

Die Eingabedatei ist eine SAM- oder ISAM-Datei. Sie enthält Datenbanksätze und Subschemasätze.

- Datenbanksätze sind wie folgt aufgebaut:

```
DB'BLANK'dbname'BLANK'konfname'BLANK'prozessorname['BLANK'kennwort]
```

- Subschemasätze sind wie folgt aufgebaut:

```
SS'BLANK'subschemaname'BLANK'dbname
```

Jeder Datenbanksatz und Subschemasatz muss in einer neuen Zeile beginnen und darf bis zu 160 Zeichen lang sein. UDS-D ignoriert führende Leerzeichen.

dbname, *konfname*, *prozessorname* und *subschemaname*, die zunächst als **gesperrt** gelten sollen, müssen durch ein **vorangestelltes Minuszeichen** gekennzeichnet werden. Ein Anwenderprogramm kann auf diese Datenbanken und Subschemata erst dann zugreifen, wenn die Sperre mit dem DAL-Kommando &UNLOCK DISTRIBUTION aufgehoben wird.

Sätze, die mit einem Stern „*“ beginnen, gelten als Kommentarsätze und werden beim Aufbau der Verteiltabelle übergangen. Kommentarsätze ermöglichen eine übersichtliche Gestaltung der Eingabedatei.

UDS-D prüft die Syntax und die eindeutige Zuordnung der Namen. Es darf zugeordnet sein:

- *subschemaname* nur zu einem *dbname*
- *dbname* nur zu einem *konfname*/*prozessorname*

Wenn derselbe Name ein zweites Mal auftritt, ignoriert UDS-D den betreffenden Datenbank- bzw. Subschemasatz und gibt eine Meldung aus.

Im Falle einer Multi-DB-Konfiguration existieren mehrere DB-Sätze mit demselben Eintrag für *konfname*/*prozessorname*. Wenn einer solchen Konfiguration ein Kennwort zugeordnet werden soll, so genügt es, das Kennwort bei einem der zugehörigen DB-Sätze anzugeben.

Wenn in verschiedenen DB-Sätzen unterschiedliche Kennwörter zu einer Konfiguration angegeben werden, so gilt das in der Sequenz der DB-Sätze zuerst angegebene Kennwort. Die zur selben Konfiguration gehörigen DB-Sätze mit davon abweichender Kennwortangabe werden mit einer entsprechenden Meldung abgewiesen.

Datenbanksätze

Datenbanksätze geben an, zu welcher DB-Konfiguration eine Datenbank gehört, gegebenenfalls das Kennwort dieser Konfiguration und auf welchem Verarbeitungsrechner die Konfiguration liegt.

Es gelten folgende Namensregeln:

prozessorname

Der Name des Verarbeitungsrechner darf maximal acht Zeichen lang sein.

dbname

Datenbanknamen müssen innerhalb der Verteiltabelle netzweit eindeutig sein. Siehe auch Tabelle „Datentypen“ („SDF-Syntaxdarstellung“).

konfname

Die ersten acht Zeichen von *konfigurationsname*; er muss in den ersten sieben Zeichen eindeutig sein. Das achte Zeichen darf nicht „@“ sein. Nullen am Ende von *konfname* sind nicht signifikante Zeichen, d.h. *konfname* ABC und *konfname* ABC0 werden nicht unterschieden.

kennwort

Das konfigurationsübergreifende Kennwort darf maximal acht Zeichen lang sein.

Subschemasätze

Subschemasätze geben an, zu welcher Datenbank das Subschema gehört.

Es gelten folgende Namensregeln:

*subschema*name

Er muss in den ersten sechs Zeichen innerhalb der Verteiltabelle netzweit eindeutig sein. Siehe auch Tabelle „Datentypen“ („[SDF-Syntaxdarstellung](#)“).

Es dürfen nur Benutzersubschemasätze angegeben werden. PRIVACY-AND-IQF-Subschema und COMPILER-Subschema sind in der Verteiltabelle nicht erlaubt.

14.5.3.2 Eingabedatei einlesen

Der DBH liest den Namen der Eingabedatei für eine Verteiltabelle mit dem DBH-Ladeparameter PP DISTABLE (siehe [Abschnitt „Eingabedatei für die Verteiltabelle festlegen \(DISTABLE\)“](#)).

14.5.3.3 Aufbau der Verteiltabelle

UDS-D baut nach dem Inhalt der Eingabedatei eine Verteiltabelle auf (siehe Bild 24).

Die Verteiltabelle unterscheidet:

- **NODE/CONF-Eintrag**
enthält einen Prozessornamen und die ersten sieben Zeichen eines Konfigurationsnamens gegebenenfalls mit Nullen aufgefüllt und das achte Zeichen „@“
- **DB-Eintrag**
enthält einen Datenbanknamen
- **SS-Eintrag**
enthält die ersten sechs Zeichen eines Subschemanamens

Der SS-Eintrag ist verkettet mit dem dazugehörigen DB-Eintrag und der DB-Eintrag ist verkettet mit dem dazugehörigen NODE/CONF-Eintrag.

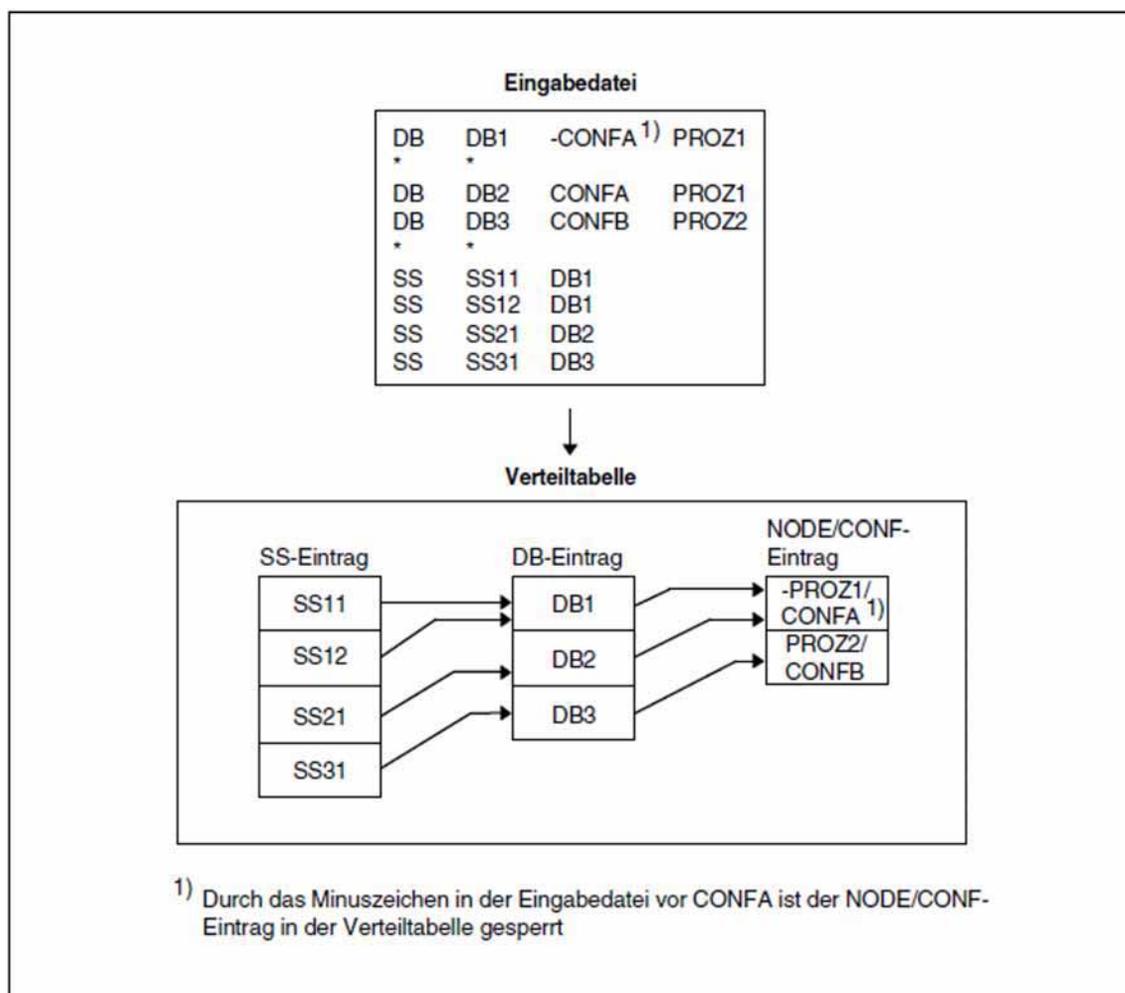


Bild 24: Aufbau der Verteiltabelle aus einer Eingabedatei

Die Pfeile in der Verteiltabelle verweisen vom SS-Eintrag auf den dazugehörigen DB-Eintrag und vom DB-Eintrag auf den dazugehörigen NODE/CONF-Eintrag. Nur wenn die Kette SS-Eintrag - DB-Eintrag - NODE/CONF-Eintrag vollständig ist, kann die Verteilkomponente in der Anwendertask eine DML-Anweisung an das gewünschte Subschema einer entfernten Konfiguration senden. Andernfalls erhält das Anwenderprogramm den Statuscode 141: „Ungültige Subschemadefinition“.

14.5.3.4 Übersicht der DAL-Kommandos für Verteiltabellen

Der Datenbankadministrator kann mit DAL-Kommandos die lokale Verteiltabelle so ändern, dass eine vollständige Kette erhalten bleibt oder eine neue Kette entsteht. Die Beschreibung der DAL-Kommandos finden Sie im [Abschnitt „Die Datenbankadministratorsprache DAL“](#).

DAL-Kommando	Bedeutung
&ADD DISTRIBUTION...	Neue Einträge in die Verteiltabelle bringen
&CHANGE DISTRIBUTION...	Eine Konfiguration einem anderen Verarbeitungsrechner zuordnen, d.h. den NODE/CONF-Eintrag ändern
&DISPLAY DISTRIBUTION...	Verteiltabelle auf Bildschirm ausgeben
&DROP DISTRIBUTION...	Einträge löschen
&LOCK DISTRIBUTION...	Einträge in der Verteiltabelle sperren
&PWD DISTRIBUTION...	Konfigurationsbezogenes Kennwort vergeben oder verändern
&SAVE DISTRIBUTION...	Verteiltabelle in eine SAM- oder ISAM-Datei sichern
&UNLOCK DISTRIBUTION...	Sperren für Einträge aufheben

Tabelle 32: Übersicht über die DAL-Kommandos für Verteiltabellen

i

- Änderungen in Verteiltabellen wirken sich erst bei der nächsten READY-Anweisung aus.
- Der Datenbankadministrator kann nur Einfluss nehmen auf die Verteiltabelle, die zu seinem DBH gehört, es sei denn, er administriert zentral mehrere DB-Konfigurationen (siehe [Abschnitt „Zentral administrieren“](#)).
- Verschiedene DBHs können sich nicht über den Inhalt ihrer Verteiltabellen abstimmen.
- Änderungen von Verteiltabellen haben keinen Einfluss darauf, wo tatsächlich eine DB-Konfiguration liegt.
- Die Ausgabe der Verteiltabelle ist auf 32 KB beschränkt. Ist die Verteiltabelle größer, dann wird sie nicht vollständig ausgegeben und Sie werden in einer Meldung hierüber informiert.

14.5.4 Zeitgesteuertes Überwachen von Verbindungen und Transaktionen

UDS-D überwacht mit einem Timer den Betrieb mit verteilten Transaktionen. Zur Berechnung des Timer-Zeitintervalls wird der DBH-Ladeparameter PP CHCKTIME herangezogen (siehe [Abschnitt „Verbindung und Transaktion überwachen \(CHCKTIME\)“](#)).

UDS-D überwacht

- die logischen Verbindungen zu den entfernten Anwenderprogrammen.
- sekundäre Teiltransaktionen und prüft, ob die dazugehörigen primären Teiltransaktionen noch existieren.
- sekundäre Teiltransaktionen im Zustand PTC.
- primäre Teiltransaktionen, die durch den Datenbankadministrator beendet wurden.

14.5.4.1 Überwachen der logischen Verbindung zu entfernten Anwenderprogrammen

UDS-D verhindert, dass der Partner eine langlaufende Bearbeitung einer DML-Anweisung als Leitungs- oder Partnerausfall interpretiert.

Wenn die UDS-D-Task eine DML-Anweisung von einem entfernten Anwenderprogramm empfängt, sendet sie Zeitquittungen an das entfernte Anwenderprogramm.

Die Zeitquittungen zeigen, dass die DML-Anweisung noch von einer Servertask bearbeitet wird (siehe Bild 25). Die UDS-D-Task sendet die Zeitquittungen an UDSNET, die Verteilungskomponente in der Anwendertask, die die Kommunikationsaufgaben ausführt.

Die Zeitquittungen werden in Zeitintervallen gesendet, die abhängig sind vom DBH-Ladeparameter PP CHCKTIME.

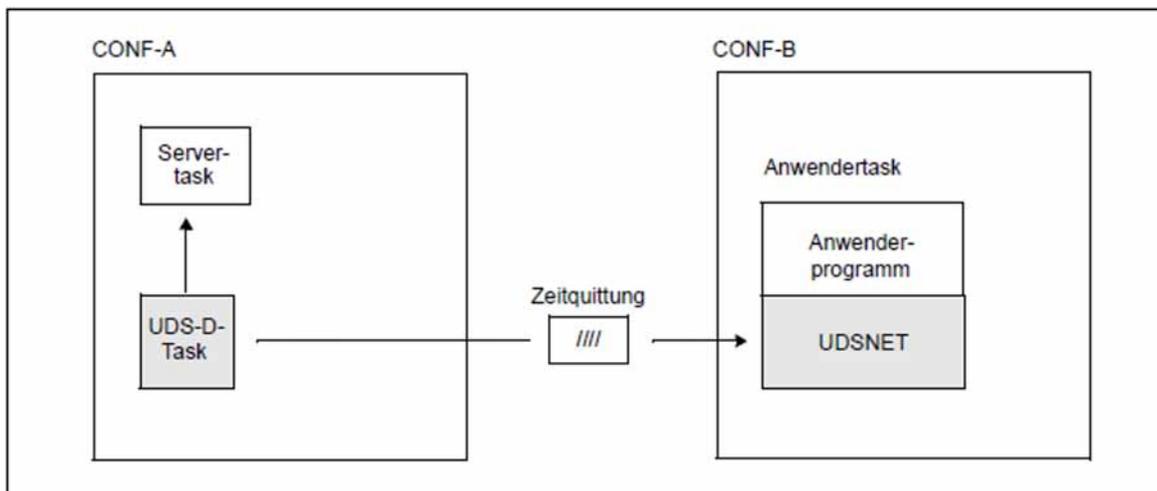


Bild 25: Überwachen der logischen Verbindung zum entfernten Anwenderprogramm

UDSNET wartet eine Zeitspanne, die abhängig ist von PP CHCKTIME, auf eine Nachricht von der entfernten UDS-D-Task.

Wenn es nach Ablauf dieser Zeitspanne keine Nachricht von der entfernten UDS-D-Task erhält und auch keine Nachricht über einen logischen Verbindungsverlust hat, baut UDSNET die Verbindung zu der entfernten UDS-D-Task ab und erneut auf und wiederholt die letzte DML-Anweisung.

Wenn UDSNET wieder keine Antwort erhält, setzt es die Transaktion zurück. Das Anwenderprogramm erhält den Statuscode 122: „Transaktion wurde inzwischen zurückgesetzt“.

14.5.4.2 Überwachen sekundärer Teiltransaktionen, die nicht im Zustand PTC sind

Falls eine sekundäre Teiltransaktion, die nicht im Zustand PTC ist, innerhalb einer von PP CHCKTIME abhängigen Zeitspanne keine Nachricht mehr von ihrer primären Teiltransaktion erhalten hat, fragt die UDS-D-Task bei der UDS-D-Task der Konfiguration der primären Teiltransaktion, ob die zugehörige primäre Teiltransaktion noch existiert. Wenn sie nicht mehr existiert, wird die sekundäre Teiltransaktion zurückgesetzt. Wenn die UDS-D-Task der primären Teiltransaktion nicht erreichbar ist, bleibt die sekundäre Teiltransaktion stehen.

14.5.4.3 Überwachen sekundärer Teiltransaktionen, die im Zustand PTC sind

Sekundäre Teiltransaktionen, die im Zustand PTC sind, warten auf die FINISH- bzw. FINISH WITH CANCEL-Anweisung der zugehörigen primären Teiltransaktion.

Wenn eine sekundäre Teiltransaktion in dieser Phase die logische Verbindung zu ihrer primären Teiltransaktion verliert, wird die automatische PTC-Überwachung, die UDS-D durchführt, besonders wichtig.

Eine sekundäre Teiltransaktion im Zustand PTC wartet eine bestimmte Zeitspanne, die abhängig ist von PP CHCKTIME, auf eine Nachricht der primären Teiltransaktion. Danach versucht die UDS-D-Task der sekundären Teiltransaktion automatisch, den Zustand der zugehörigen primären Teiltransaktion zu ermitteln. Wenn dies gelingt, wird die sekundäre Teiltransaktion wie ihre primäre Teiltransaktion behandelt. Wenn dies nicht gelingt, wird die sekundäre Teiltransaktion gemäß dem Wert des DBH-Ladeparameters PTCSYNCH für die laufende Session behandelt, d.h. wenn PTCSYNCH auf ABORT oder COMMIT steht, wird die sekundäre Teiltransaktion im Zustand PTC mit ABORT oder COMMIT beendet, wenn PTCSYNCH auf WAIT steht, bleibt sie im Zustand PTC.

Mit dem DAL-Kommando &SYNCHRONIZE DISTRIBUTION kann der Datenbankadministrator die Nachfrage bei der Konfiguration der primären Teiltransaktion beschleunigen. Wenn nach &SYNCHRONIZE DISTRIBUTION die Konfiguration der primären Teiltransaktion nicht erreichbar ist, werden unmittelbar nach der Ausführung von &SYNCHRONIZE DISTRIBUTION alle sekundären Teiltransaktionen, für die der Zustand der primären Teiltransaktion nicht ermittelt werden kann, gemäß dem eingestellten PTCSYNCH-Wert für die laufende Session behandelt.

i Wenn der DBH-Ladeparameter PP PTCSYNCH nicht auf (WAIT,WAIT) steht, ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

14.5.4.4 Überwachen primärer Teiltransaktionen, die durch den Datenbankadministrator beendet wurden

Falls der Datenbankadministrator primäre Teiltransaktionen „per Hand“ mit den DAL-Kommandos COMMIT oder ABORT,OPTION=PTC beendet, werden die dazugehörigen sekundären Teiltransaktionen automatisch über den Timer benachrichtigt und beendet wie die primäre Teiltransaktion.

14.5.5 UDS-D-Betrieb starten und beenden

Die folgenden Abschnitte beschreiben, wie Sie den UDS-D-Betrieb starten, beenden und nach Beendigung erneut starten.

14.5.5.1 UDS-D-Betrieb starten

Beim DBH-Start mit dem independent DBH steuern Sie mit dem DBH-Ladeparameter PP DISTRIBUTION, ob der UDS-D-Betrieb für eine Konfiguration gestartet wird oder nicht (siehe [Abschnitt „Die Session“](#)). Wenn der UDS-D-Betrieb gestartet ist, kann diese Konfiguration am UDS-D-Betrieb teilnehmen. Am UDS-D-Betrieb teilnehmen heißt, lokale Anwenderprogramme können verteilte Transaktionen starten, und DML-Anweisungen von entfernten Anwenderprogrammen können bearbeitet werden.

- PP DISTRIBUTION=NO (Standard): In dieser Session nimmt die Konfiguration nicht am UDS-D-Betrieb teil. Es können weder lokale Anwenderprogramme verteilte Transaktionen starten, noch werden sekundäre Teiltransaktionen bearbeitet, die von entfernten Anwenderprogrammen kommen. Das entfernte Anwenderprogramm erhält den Statuscode 141, falls es versucht, auf eine Datenbank dieser Konfiguration zuzugreifen.
- PP DISTRIBUTION=START: Die Konfiguration nimmt ab dem DBH-Start am UDS-D-Betrieb teil.
- PP DISTRIBUTION=STANDBY: Der UDS-D-Betrieb wird vorbereitet. In dieser Session nimmt die Konfiguration erst dann am UDS-D-Betrieb teil, wenn der Datenbankadministrator den UDS-D-Betrieb mit dem DAL-Kommando &START DISTRIBUTION startet.

Wenn Sie den DBH-Ladeparameter PP DISTRIBUTION auf START oder STANDBY gesetzt haben, schreibt die Mastertask bei ihrem Start die ENTER-Datei für die UDS-D-Task UDSCCT mit folgendem Namen in die Kennung, in der Sie den DBH starten:

```
UDS. ENTER. tsn. CT000
```

tsn Auftragsnummer der Mastertask

Bei PP DISTRIBUTION=START wird der ENTER-Job sofort gestartet.

Die ENTER-Datei bleibt bis zum DBH-Ende bestehen, auch wenn der UDS-D-Betrieb vorher beendet wird.

Weitere DBH-Ladeparameter zum Steuern des UDS-D-Betriebs finden Sie im [Abschnitt „DBH-Ladeparameter für UDS-D“](#) und in [Kapitel „Ladeparameter des DBH“](#).

i Bei PP LOG=NO lässt der DBH nicht zu, dass lokale Anwenderprogramme verteilte UPDATE-Transaktionen starten, da ohne Logging kein Session-Wiederanlauf möglich ist. Verteilte RETRIEVAL-Transaktionen sind mit PP LOG=NO erlaubt.

14.5.5.2 UDS-D-Betrieb beenden

Sie beenden den UDS-D-Betrieb, indem Sie

- den DBH beenden oder
- nur den UDS-D-Betrieb beenden.

DBH beenden

Den DBH beenden Sie mit einem der folgenden DAL-Kommandos:

- CLOSE RUN-UNITS (DBH normal beenden)
- CLOSE CALLS (DBH beschleunigt normal beenden mit FINISH WITH CANCEL für alle offenen Transaktionen). UDS/SQL kann den DBH erst dann mit CLOSE RUN-UNITS oder CLOSE CALLS beenden wenn keine Teiltransaktionen mehr im Zustand PTC sind.
Der Datenbankadministrator kann den DBH trotz Teiltransaktionen im Zustand PTC beenden (siehe [Abschnitt „Zustand PTC beenden“](#)).
- %TERM (DBH sofort abbrechen, ohne offene Transaktionen zu beenden)
Teiltransaktionen im Zustand PTC bleiben im Zustand PTC. Bei einem Warmstart werden sie gemäß dem Wert des DBH-Ladeparameters PP PTCSYNCH für Warmstart behandelt (siehe „Auswirkungen des Zustands PTC bei einem Warmstart“ im [Abschnitt „Auswirkungen des Zustands PTC“](#)).

Die Beschreibung der DAL-Kommandos zum Beenden des DBH finden Sie im [Abschnitt „DBH beenden“](#).

UDS-D-Betrieb beenden

Mit dem DAL-Kommando &CLOSE DISTRIBUTION beenden Sie den UDS-D-Betrieb. Sobald alle Teiltransaktionen abgeschlossen sind, beendet UDS/SQL die UDS-D-Task UDSCT. Teiltransaktionen im Zustand PTC, die nach Ablauf einer von PP CHCKTIME abhängigen Zeitspanne nicht beendet werden können, bleiben im Zustand PTC. Wenn die UDS-D-Task beendet ist, lässt UDS/SQL weder entfernte Verarbeitungsketten eines lokalen Anwenderprogramms zu, noch akzeptiert es sekundäre Teiltransaktionen eines entfernten Anwenderprogramms.

Nach der Beendigung des UDS-D-Betriebs können Sie in dieser Session den UDS-D-Betrieb erneut starten (siehe [Abschnitt „UDS-D-Betrieb erneut starten“](#)).

i Bei abnormaler Beendigung der UDS-D-Task in einer Konfiguration können lokale Anwenderprogramme dieser Konfiguration in der Regel weiterhin entfernte Konfigurationen ansprechen.

14.5.5.3 UDS-D-Betrieb erneut starten

Sie können innerhalb einer Session - nach normaler oder abnormaler Beendigung des UDS-D-Betriebs - in der Regel den UDS-D-Betrieb erneut starten mit dem DAL-Kommando `&START DISTRIBUTION`. In diesem Fall werden die vorhandenen UDS-D-Tabellen weiterbenutzt.

Sekundäre Teiltransaktionen, die aus dem vorangegangenen UDS-D-Betrieb noch offen sind, werden folgendermaßen behandelt:

- Sekundäre Teiltransaktionen, die nicht im Zustand PTC sind, werden zurückgesetzt.
- Sekundäre Teiltransaktionen im Zustand PTC werden beendet wie die zugehörige primäre Teiltransaktion. Falls es nicht gelingt, den Zustand der primären Teiltransaktion zu ermitteln, wird die sekundäre Teiltransaktion gemäß dem Wert des DBH-Ladeparameters `PP PTCSYNCH` für die laufende Session behandelt.
 - Bei `PP PTCSYNCH=(...,ABORT)` bzw. `=(...,COMMIT)` wird die sekundäre Teiltransaktion mit `ABORT` bzw. `COMMIT` beendet.
 - Bei `PP PTCSYNCH=(...,WAIT)` bleibt die sekundäre Teiltransaktion im Zustand PTC.

14.5.5.4 Beispiele für UDS-D-Betrieb

Das folgende Ablaufbeispiel zeigt, wie Sie den UDS-D-Betrieb starten, beenden und nach Beendigung erneut starten.

```
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=CONFVERS
/CREATE-FILE FILE-NAME=CONFVERS.DBSTAT
/CREATE-FILE FILE-NAME=CONFVERS.DBSTAT.SAVE
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.9B00
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D, VERSION=02.9B00
/START-UDS-DBH
```

```
...
% BLS0523 ELEMENT 'UDSSQL', VERSION '02.9B00', TYPE 'L' VON BIBLIOTHEK ':SQL2:$XXXXXXXXX.
SYSLNK.UDS-
SQL.029' IN BEARBEITUNG
% BLS0524 LLM 'UDSSQL', VERSION '02.9B00' VON '2017-04-17 23:27:17' GELADEN
```

```
PP ADM=REMOTE
PP DBNAME=KUNDEN
PP LOG=PUBLIC
PP MAXDB=4
PP PARLIST=YES
PP PRIVACY-CHECK=OFF
PP DISDB=3
PP DISTABLE=UDSDBB.VT.CONFKUND
PP DISTRIBUTION=START
PP END
```

```
% UDS0746 UDS-PUBSET-DEKLARATION (CURRENT) FOLGT (OPI6746,11:29:09/0YA9)
% 0YA9: UDS-PUBSET-JV: :SQL2:$XXXXXXXXX.PUBSDECL.PUBS
% 0YA9: PUBSETS: SQL2
% 0YA9: DEFAULT PUBSET: SQL2
% 0YA9: -----
% UDS0347 UDS-ADMINISTRATION APP.NAME = CONFVERS (OPDI239,11:29:09/0YA9)
% UDS0722 UDS AUFTRAG ADD RLOG 150628092908 WIRD AUSGEFUEHRT (OPCC283,11:29:09/0YA9)
0YA8: MAXDB = 4
0YA8: TRANSACTION = ( 4, 1)
0YA8: SUBSCHEMA = 1
0YA8: SERVERTASK = 1
0YA8: 2KB-BUFFER-SIZE= 1
0YA8: 4KB-BUFFER-SIZE= 1
0YA8: 8KB-BUFFER-SIZE= 0
0YA8: CP-SIZE = 1024
0YA8: CUP-SIZE = 1024
0YA8: SIP-SIZE = 1024
0YA8: DIP-SIZE = 1024
0YA8: DISDB = 3
0YA8: CHCKTIME = 60
0YA8: DEADTIME = 60
0YA8: DISTRIBUTION = START
0YA8: PTCSYNCH = (WAIT ,WAIT )
0YA8: LOG = PUBLIC
0YA8: LOG-2 = NO
0YA8: LOG-SIZE = ( 192, 192)
```

```

OYA8: RESERVE           = NONE
OYA8: WARMSTART        = STD
OYA8: DBDCSYNCH       = YES
OYA8: DEACT           = YES
OYA8: STDCKPT         = NO
OYA8: ADM             = REMOTE
OYA8: CPU             = MONO-PROCESSOR
OYA8: DISTABLE        = UDSDBB.VT.CONFKUND
OYA8: SQL             = 4
OYA8: SQL-LIMIT       = 10
OYA8: TASKLIB         = NO
OYA8: DUMP            = ALL
OYA8: MPSEG           = STD
OYA8: UCON            = C'<U ',MSG
OYA8: SECLEVEL        = NO ,NO-AUDIT
OYA8: LOCK            = STD
OYA8: TA-ACCESS       = STD
OYA8: WAIT            = EVENT
OYA8: RESULT-DELAY    = 0
OYA8: SCHEDULING      = SYMMETRIC
OYA8: IO              = ASYNC
OYA8: ORDER-DBSTATUS = STD
OYA8: PRIVACY-CHECK   = OFF
OYA8: BCAM-PREFIX     = SUD$
OYA8: CONFNAME        = $XXXXXXXXX.CONFKUND
. . .

```

//EXECUTE-DAL-CMD CMD=&CLOSE DISTRIBUTION

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &CLOSE DISTRIBUTION (OPOX073,11:30:48
/OYA9)
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (CTCC012,11:30:48/OYBC)
% UDS0807 UDS-D: BEENDEN EINGELEITET (CTCC351,11:30:48/OYBC)
% UDS0809 UDS-D WURDE NORMAL BEENDET (2017-06-28) (CTCC353,11:30:53/OYBC)
. . .

```

//EXECUTE-DAL-CMD CMD=&SAVE DISTRIBUTION,FILE=VT.BEISPIEL.SAVE

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &SAVE DISTRIBUTION,FILE=VT.
BEISPIEL.SAVE
(OPOX073,11:30:48/OYA9)
% UDS0808 UDS-D WIRD GERADE BEENDET (OPCC352,11:30:48/OYA9)
. . .

```

//EXECUTE-DAL-CMD CMD=&START DISTRIBUTION

```

% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &START DISTRIBUTION (OPOX073,11:31:
48/OYA9)
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (OPCC382,11:31:48/OYA9)
% UDS0801 UDS-D GESTARTET V2.9 /2017-06-28 (CTCC345,11:31:48/OYBF)
% UDS0823 UDS-D: EINGABEDATEI FUER VERTEILTABELLE UDSDBB.VT.CONFVERS WURDE GELESEN
(CTCC386,11:31:48/OYBF)
% UDS0805 UDS-D IST BETRIEBSBEREIT (CTCC349,11:31:48/OYBF)
. . .

```

//EXECUTE-DAL-CMD CMD=&SAVE DISTRIBUTION,FILE=VT.BEISPIEL.SAVE

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: &SAVE DISTRIBUTION,FILE=VT.BEISPIEL.  
SAVE  
(OPOX073,11:32:48/0YA9)  
% UDS0832 UDS-D: KOMMANDO AUSGEFUEHRT (CTCC382,11:32:50/0YBF)  
. . .
```

Eingabedatei für die Verteiltabelle VT.BEISPIEL.START

```
* DATENBANKSAETZE CONF-3
DB DB11 CONF3 PROZ1
DB DB12 CONF3 PROZ1
*
* DATENBANKSAETZE CONF-4
DB DB21 CONF4 PROZ2
DB DB22 CONF4 PROZ2
*
* SUBSCHEMASAETZE CONF-3
SS SS111 DB11
SS -SS112 DB11
*
SS SS121 DB12
SS SS122 DB12
*
* SUBSCHEMASAETZE CONF-4
SS SS21 DB21
SS SS22 DB22
```

Im folgenden Ablaufbeispiel wird gezeigt, wie Sie auf mehrere DB-Konfigurationen zugreifen können.

Mit dem Programm DMLTEST wird auf die lokale Konfiguration CONFVERS (Subschema ADMIN) und auf die entfernte Konfiguration CONFKUND (Subschema VERWALTUNG) zugegriffen.

```
/CALL-PROCEDURE P.DMLTEST.IND
/BEGIN-PROC LOG=*ALL,PAR=*YES(PROC-PAR=( &CONFNAME ))
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=&CONFNAME
/ADD-FILE-LINK LINK-NAME=$UDSSSI,FILE-NAME=LMS.SSITAB
```

```
&&CONFNAME=
```

CONFVERS

```
/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=CONFVERS
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=02.9A00
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D, VERSION=02.9A00
/START-UDS-DMLTEST
```

```
...
DBH IND
LANG COB
DISPLAY RCODE,COND=RCODE NE C'00000'
PROT ON
DISPLAY RECA,L=80
SYS
```

SUBSCHEMA ADMIN

```
READY USAGE-MODE UPDATE;E
```

```
RECORD - AREA      :
.....
```

FETCH LAST FARBEN;E

```
RECORD - AREA      :  
25LILA             .....
```

M RECA,26MAGENTA

STORE FARBEN;E

```
RECORD - AREA      :  
26MAGENTA         .....
```

SUBSCHEMA VERWALTUNG

READY USAGE-MODE UPDATE;E

```
RECORD - AREA      :  
26MAGENTA         .....
```

FETCH LAST KUNDE;E

```
RECORD- AREA      :  
YMEIER           .....
```

M RECA, SCHMIDTCHEN

MODIFY K-NAME;E

```
RECORD - AREA      :  
SCHMIDTCHEN.....
```

FETCH LAST KUNDE;E

```
RECORD - AREA      :  
SCHMIDTCHEN.....
```

SUBSCHEMA ADMIN

FETCH LAST FARBEN;E

```
RECORD - AREA      :  
26MAGENTA         .....
```

FINISH;E

```
RECORD - AREA      :  
26MAGENTA         .....
```

HALT

```
DMLTEST NORMAL TERMINATION
```

Eingabedatei für die Verteiltabellen beider Konfigurationen

```

DB  VERSAND  CONFVERS  XXXXXXXX
DB  KUNDEN   CONFKUND  XXXXXXXX
DB  ARTIKEL  CONFART   YYYYYYYY
*
SS  ADMIN    VERSAND
SS  VERWALTUNG  KUNDEN
SS  SUBART   ARTIKEL

```

Konfiguration der primären Teiltransaktion

```
//EXECUTE-DAL-CMD CMD=DISPLAY USERS
```

```

%   UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY USERS (OPOX073,11:30:44
/OYA9)
0YA9: PROGRAM   TSN/TERM      RUNUNIT-ID   STATE      PTC  FC   MR-NR  DLY
0YA9: -----
0YA9: DMLTEST   TSN 0YBE           1    USER      -   134   1    -
0YA9:
0YA9: UDS/SQL V2.9      1 USER    OF CONFVERS
%   UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:30:44/OYA9)

```

```
//EXECUTE-DAL-CMD CMD=DISPLAY 1
```

```

%   UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY 1 (OPOX073,11:30:44/OYA9)
0YA9: PROGRAM-NAME:      DMLTEST
0YA9: TSN:                0YBE
0YA9: RUNUNIT-ID / -STATE:      1 / USER
0YA9: MAINREF-NR / -STATE:      1 / USED
0YA9: FC:                  134
0YA9: PTT-PROC-NAME:        D017ZE07
0YA9: PTT-CONF-NAME:        CONFVERS
0YA9: PTT-RLOG-DATE:        150628092908
0YA9: PTT-RUNUNIT-ID:        1
0YA9: LOCAL-TT-STATE:        PTT
%   UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:30:44
/OYA9)

```

Konfiguration der sekundären Teiltransaktion

//EXECUTE-DAL-CMD CMD=DISPLAY USERS

```
% UDS0347 UDS-ADMINISTRATION LOGON      = UAD@0YA2 (OPDM239,11:30:40/0YA8)
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY USERS (OPOX073,11:30:40/0YA8)
0YA8: PROGRAM      TSN/TERM      RUNUNIT-ID  STATE      PTC  FC   MR-NR DLY
0YA8: -----
0YA8: DMLTEST      TSN 0YBE              1    USER      -   134   1    -
0YA8:
0YA8: UDS/SQL V2.9      1 USER      OF CONFKUND
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:30:40/0YA8)
```

//EXECUTE-DAL-CMD CMD=DISPLAY 1

```
% UDS0220 UDS HAT FOLGENDES KOMMANDO ENTGEGENGENOMMEN: DISPLAY 1 (OPOX073,11:30:40/0YA8)
0YA8: PROGRAM-NAME:      DMLTEST
0YA8: TSN:                0YBE
0YA8: RUNUNIT-ID / -STATE:      1 / USER
0YA8: MAINREF-NR / -STATE:      1 / USED
0YA8: FC:                  134
0YA8: PTT-PROC-NAME:        D017ZE07
0YA8: PTT-CONF-NAME:        CONFVERS
0YA8: PTT-RLOG-DATE:
0YA8: PTT-RUNUNIT-ID:        1
0YA8: LOCAL-TT-STATE:        STT
% UDS0218 UDS HAT DIE BEARBEITUNG DES DAL-KOMMANDOS ABGESCHLOSSEN. (OPCC074,11:30:40/0YA8)
```

14.5.6 Auswirkungen des Zustands PTC

Bei Ablauf des Zwei-Phasen-Ende-Protokolls wird in der RLOG-Datei der jeweiligen Teiltransaktion protokolliert:

- „Transaktion im Zustand PTC“, wenn eine Teiltransaktion dieser Transaktion im Zustand PTC ist
- „Transaktion mit FINISH beendet“ oder „Transaktion mit FINISH WITH CANCEL beendet“, wenn die primäre Teiltransaktion mit FINISH oder mit FINISH WITH CANCEL beendet wurde.

Siehe [Abschnitt „Konfigurationsübergreifende Konsistenz mit dem Zwei-Phasen-Ende-Protokoll sichern“](#).

Da die RLOG-Datei sessionspezifisch geführt wird, ist es möglich, dass nach normalem DBH-Ende in der Konfiguration der primären Teiltransaktion nicht mehr abgefragt werden kann, wie die primäre Teiltransaktion beendet wurde. Deshalb wird die Information, wie die Transaktion beendet wurde, auch in der DB-Status-Datei hinterlegt, die konfigurationsbezogen geführt wird.

Eine kritische Phase des Zwei-Phasen-Ende-Protokolls ist der Zustand PTC. Die Behandlung einer Teiltransaktion im Zustand PTC steuern Sie über den DBH-Ladeparameter PP PTCSYNCH.

i Der DBH-Ladeparameter PP PTCSYNCH sollte immer auf (WAIT,WAIT) stehen. Nur dann ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz sicher! Der Datenbankadministrator sollte den PTCSYNCH-Wert nur dann auf ABORT oder COMMIT ändern, wenn er in der Lage ist, über die konfigurationsübergreifende Konsistenz zu entscheiden (siehe [Abschnitt „Zustand PTC beenden“](#)).

Der Zustand PTC kann sich auswirken:

- während der Session
- bei einem Warmstart
- bei DBH-Ende
- auf das Anwenderprogramm bei logischem Verbindungsverlust

Auswirkungen des Zustands PTC während der Session

Während der Session kann sich der Zustand PTC auswirken

- wenn der UDS-D-Betrieb nach normaler oder abnormaler Beendigung erneut gestartet wird.
- beim Ändern der Datenbank-Konfiguration (siehe [Abschnitt „DB-Konfiguration ändern- Hinweise für UDS-D“](#)). Wenn sich Teiltransaktionen im Zustand PTC befinden, können Sie die betroffenen Datenbanken nicht konsistent aus der DB-Konfiguration ausschließen.
- beim Festschreiben eines Konsistenzpunkts (siehe [Abschnitt „Konsistenzpunkte festschreiben \(CHECKPOINT\)“](#)). Wenn Konsistenzpunkte festgeschrieben werden sollen, muss sichergestellt sein, dass die Verbindung zur Konfiguration der primären Teiltransaktion bzw. zu openUTM vorhanden ist. Denn UDS/SQL schreibt nur Konsistenzpunkte für eine Datenbank, wenn alle Transaktionen, die auf dieser Datenbank geändert haben, beendet sind.

Falls das Festschreiben von Konsistenzpunkten oder das Aus- bzw. Anschließen von Datenbanken bereits eingeleitet ist, aber nicht durchführbar ist wegen Teiltransaktionen im Zustand PTC, dann kann der Datenbankadministrator:

- mit `&SYNCHRONIZE DISTRIBUTION` bei sekundären Teiltransaktionen die Nachfrage veranlassen, wie die primäre Teiltransaktion beendet wurde. Falls UDS/SQL das Ende der primären Teiltransaktion nicht ermitteln kann, bleibt die sekundäre Teiltransaktion im Zustand PTC.
- mit `DISPLAY USERS` die Transaktionskennungen der Teiltransaktionen ermitteln, die im Zustand PTC sind, und mit den folgenden DAL-Kommandos diese Teiltransaktion beenden:

```
ABORT transaktionskennung, OPTION=PTC
```

oder

```
COMMIT transaktionskennung
```

i Wenn der Datenbankadministrator mit `ABORT,OPTION=PTC` oder `COMMIT` eingreift, ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

- mit `%TERM`
die Session abbrechen und sie anschließend ohne die betroffene Datenbank starten.

Auswirkungen des Zustands PTC bei einem Warmstart

Bei einem Warmstart werden alle offenen Transaktionen mit Rollback zurückgesetzt. Wenn sich bei einem Warmstart Teiltransaktionen im Zustand PTC befinden, werden sie gemäß dem DBH-Ladeparameter `PP PTCSYNCH` für Warmstart behandelt.

Wenn dieser bei einer primären Teiltransaktion auf `WAIT` steht, wird die betroffene Datenbank bei einem Warmstart so lange nicht zugeschaltet, bis `openUTM` das Transaktionsende fordert.

Wenn dieser bei einer sekundären Teiltransaktion auf `WAIT` steht, versucht UDS/SQL, die zugehörige primäre Teiltransaktion zu erreichen. Dafür müssen folgende Voraussetzungen erfüllt sein:

- Der UDS-D-Betrieb muss gestartet sein in der Konfiguration der primären und der sekundären Teiltransaktion.
- Die Verteiltabellen müssen den Gegebenheiten im Netz entsprechen.
- Es muss eine Konfiguration mit dem Konfigurationsnamen existieren, unter dem vor dem Session-Abbruch die primäre Teiltransaktion lief.
Die Konfiguration darf aber anders zusammengesetzt sein und auf einem anderen Verarbeitungsrechner liegen, wenn die Verteiltabellen angepasst sind.
- Die DB-Status-Datei muss in der Konfiguration der primären und in der Konfiguration der sekundären Teiltransaktion vorhanden sein.

Wenn der Zustand der zugehörigen primären Teiltransaktion nicht ermittelbar ist, bleibt die sekundäre Teiltransaktion im Zustand PTC, und die betroffene Datenbank wird nicht zugeschaltet.

In [Bild 26](#) ist dargestellt, wie der Warmstart abläuft, wenn sich sekundäre Teiltransaktionen im Zustand PTC befinden.

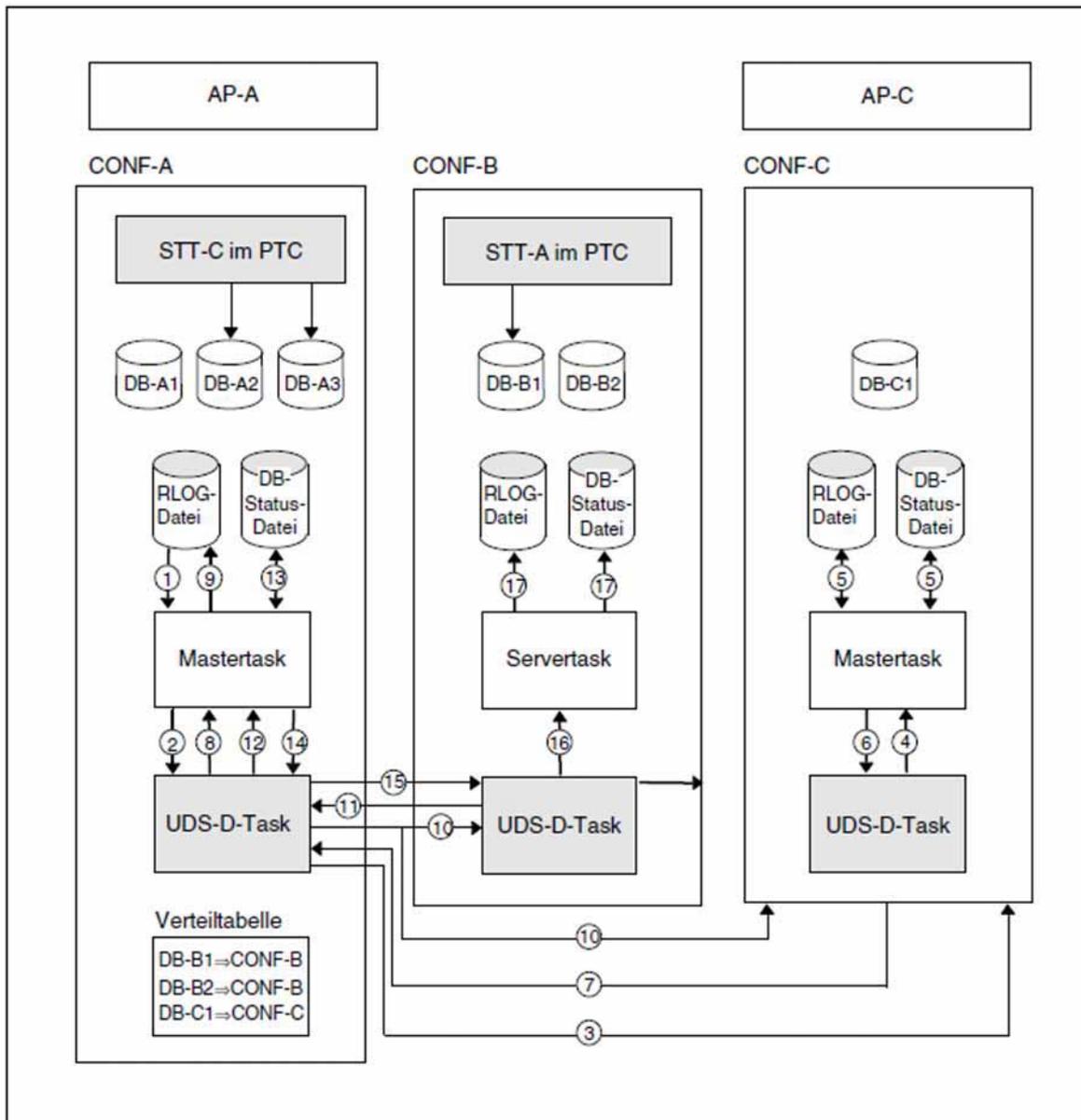


Bild 26: Warmstart mit sekundären Teiltransaktionen im Zustand PTC

Nach einem Session-Abbruch sollen in der Konfiguration CONF-A die Datenbanken DB-A1, DB-A2 und DB-A3 gestartet werden.

Die Datenbanken DB-A2 und DB-A3 müssen warmgestartet werden, da sich noch die sekundäre Teiltransaktion STT-C im Zustand PTC befindet, die das Anwenderprogramm AP-C der Konfiguration CONF-C ausgelöst hat. In der RLOG- und in der DB-Status-Datei von CONF-C ist eingetragen, ob die primäre Teiltransaktion PTT-C mit FINISH oder mit FINISH WITH CANCEL abgeschlossen hat.

1. Beim Warmstart der Konfiguration CONF-A wird in der RLOG-Datei eine Transaktion im Zustand PTC gefunden.
2. Die Mastertask übergibt die in der RLOG-Datei hinterlegte Warmstart-Information an die UDS-D-Task.
3. Die UDS-D-Task ermittelt aus der Warmstart-Information den Verarbeitungsrechner und die Konfiguration der zur STT-C gehörenden primären Teiltransaktion. Er sendet eine Nachricht an die entfernte UDS-D-Task mit der Aufforderung, festzustellen, wie die primäre Teiltransaktion beendet wurde.

4. Diese Nachricht übergibt die entfernte UDS-D-Task an ihre Mastertask.
5. Die Mastertask informiert sich in der DB-Status-Datei, wie die primäre Teiltransaktion beendet wurde. Falls diese nicht aktuell ist, schaut sie in die RLOG-Datei.
6. und 7.
Das Ergebnis wird an die UDS-D-Task weitergereicht und von dort an die UDS-D-Task der sekundären Teiltransaktion.
8. Die UDS-D-Task übergibt das Ergebnis an die Mastertask.
9. Je nach Ende der primären Teiltransaktion wird die sekundäre Teiltransaktion STT-C beendet.
10. Die UDS-D-Task von CONF-A meldet sich bei allen Konfigurationen, die in ihrer Verteiltabelle aufgeführt sind, um mitzuteilen, dass die Konfiguration CONF-A wieder ansprechbar ist.
11. Da in der Konfiguration CONF-B noch die sekundäre Teiltransaktion STT-A im Zustand PTC ist, sendet die UDS-D-Task von CONF-B eine Nachricht an die UDS-D-Task der primären Teiltransaktion, um den Zustand der primären Teiltransaktion zu ermitteln.
12. und 13.
siehe 4. und 5.
14. und 15.
siehe 6. und 7.
16. Abhängig vom Ergebnis sendet die UDS-D-Task den FINISH- oder FINISH WITH CANCEL-Auftrag an eine Servertask.
17. Das Transaktionsende wird in der RLOG-Datei protokolliert.

Wenn es bei einem Warmstart nicht gelingt, eine sekundäre Teiltransaktion im Zustand PTC genauso wie ihre primäre Teiltransaktion zu beenden, bleibt die sekundäre Teiltransaktion im Zustand PTC und die betroffene Datenbank wird beim Warmstart nicht zugeschaltet. Eine Nachricht informiert den Datenbankadministrator über andere beteiligte Teiltransaktionen.

Wenn Datenbanken bei einem Warmstart nicht zuschaltbar sind, weil sich eine Teiltransaktion im Zustand PTC befindet, müssen Sie warten, bis die Konfiguration der primären Teiltransaktion wieder ansprechbar ist bzw. bis openUTM das Transaktionsende fordert oder müssen Sie mit dem DAL-Kommando MODIFY den PTCSYNCH-Wert für Warmstart auf ABORT oder COMMIT setzen.

i Wenn der DBH-Ladeparameter PP PTCSYNCH nicht auf (WAIT, WAIT) steht, ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-/openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

Auswirkungen des Zustands PTC bei DBH-Ende

- DBH-Ende mit CLOSE RUN-UNITS oder CLOSE CALLS
UDS-D wartet, bis sich alle sekundären Teiltransaktionen im Zustand PTC automatisch beenden bzw. bis openUTM das Transaktionsende fordert.
Wenn sich die Transaktionen nicht automatisch beenden, kann der Datenbankadministrator:

- Mit `&SYNCHRONIZE DISTRIBUTION` bei sekundären Teiltransaktionen die Nachfrage veranlassen, wie die primäre Teiltransaktion beendet wurde. Falls UDS-D das Ende der primären Teiltransaktion nicht ermitteln kann, bleibt die sekundäre Teiltransaktion im Zustand PTC.
- Mit `DISPLAY USERS` die Transaktionskennungen der Teiltransaktionen ermitteln, die im Zustand PTC sind, und mit den folgenden DAL-Kommandos die Teiltransaktionen im Zustand PTC beenden:

```
ABORT transaktionskennung,OPTION=PTC
```

oder

```
COMMIT transaktionskennung
```

Wenn keine Teiltransaktionen mehr offen sind, wird der DBH beendet.

i Wenn der Datenbankadministrator Teiltransaktionen mit `ABORT,OPTION=PTC` oder `COMMIT` beendet, ist die konfigurationsübergreifende Konsistenz bzw. die UDS/SQL-openUTM-Konsistenz gefährdet (siehe [Abschnitt „Zustand PTC beenden“](#)).

- Session-Abbruch mit `%TERM`

Teiltransaktionen im Zustand PTC bleiben im Zustand PTC. Was Sie in diesem Fall bei einem Warmstart beachten müssen, finden Sie im [Abschnitt „Auswirkungen des Zustands PTC bei einem Warmstart“](#).

i Datenbanken, die während einer Session inkonsistent wurden, sollten Sie nach einem Session-Abbruch möglichst schnell warmstarten, um eine möglichst problemlose Kommunikation zwischen den UDS/SQL-Konfigurationen zu gewährleisten.

Auswirkungen des Zustands PTC auf das Anwenderprogramm bei logischem Verbindungsverlust

Wenn in der Beendigungsphase einer verteilten Transaktion die logische Verbindung zu einer entfernten Konfiguration abbricht und nicht wieder neu aufgebaut werden kann, hängen die Auswirkungen auf das Anwenderprogramm davon ab, in welcher Phase des Zwei-Phasen-Ende-Protokolls sich die primäre Teiltransaktion befindet.

- Die Anwendertask hat die PETA-Anweisung an die sekundären Teiltransaktionen gesendet, die primäre Teiltransaktion wurde aber selbst noch nicht beendet.
Das Anwenderprogramm bekommt vom Verbindungsmodul den Statuscode 122: „Transaktion wurde inzwischen zurückgesetzt“.
- Das Anwenderprogramm hat FINISH eingeleitet und er wurde fehlerfrei durchgeführt. Das Anwenderprogramm bekommt vom Verbindungsmodul den Statuscode 000: „DML-Anweisung erfolgreich durchgeführt“.
In der Konfiguration der primären Teiltransaktion gilt die Transaktion als beendet, unabhängig davon, ob in einer entfernten Konfiguration noch eine sekundäre Teiltransaktion im Zustand PTC ist oder nicht.
- Falls der FINISH der primären Teiltransaktion fehlerhaft abläuft, bekommt das Anwenderprogramm den Statuscode des fehlerhaften FINISH.

14.5.7 Mehrere Konfigurationen administrieren

Es gibt zwei Möglichkeiten, die an einer Verteilung angeschlossenen DB-Konfigurationen bzw. deren Database Handler zu administrieren:

- Lokale Administration
- Zentrale Administration

Es ist auch möglich, einen Teil der DB-Konfigurationen lokal und einen Teil zentral zu administrieren.

14.5.7.1 Lokal administrieren

Jede DB-Konfiguration wird über einen eigenen Bedienplatz angesprochen über eine der folgenden Schnittstellen:

- über das Administrationsprogramm UDSADM (siehe [Abschnitt „UDS/SQL über UDSADM administrieren“](#)).
- über das BS2000-Kommando `INFORM-PROGRAM MSG= 'dal-kommando' ["kommentar"]` (siehe [Abschnitt „UDS/SQL über INFORM-PROGRAM administrieren“](#)).
- über den DCAM-Anschluss zum Administrieren (siehe [Abschnitt „UDS/SQL über DCAM administrieren“](#)).

Die Datenbankadministratorsprache DAL ist im [Abschnitt „Die Datenbankadministratorsprache DAL“](#) beschrieben. Eine Übersicht der DAL-Kommandos für verteilte Datenbanken finden Sie in [Abschnitt „Datenbankadministratorsprache DAL für UDS-D“](#).

14.5.7.2 Zentral administrieren

Bei zentraler Administration werden **alle** DB-Konfigurationen über **einen** Bedienplatz angesprochen. Von einem Bedienplatz aus kann jeder beteiligte DBH gestartet, administriert und beendet werden.

Eine zentrale Administration mehrerer DB-Konfigurationen ist möglich

- mit UDSADM über OMNIS
- mit dem Softwareprodukt OMNIS ab der OMNIS-Version 4.0.
Die genaue Beschreibung von OMNIS finden Sie in den OMNIS-Handbüchern „[Funktionen und Kommandos](#)“ und „[Administration und Programmierung](#)“.
- über ein Administrationsprogramm, das die DCAM-Schnittstelle benutzt (siehe [Abschnitt „UDS/SQL über DCAM administrieren](#)“).

Zentral administrieren mit OMNIS

OMNIS vermittelt Nachrichten zwischen dem Datenbankadministrator und den DBHs, die der Datenbankadministrator verwalten möchte, unabhängig davon, auf welchem Verarbeitungsrechner die DBHs laufen.

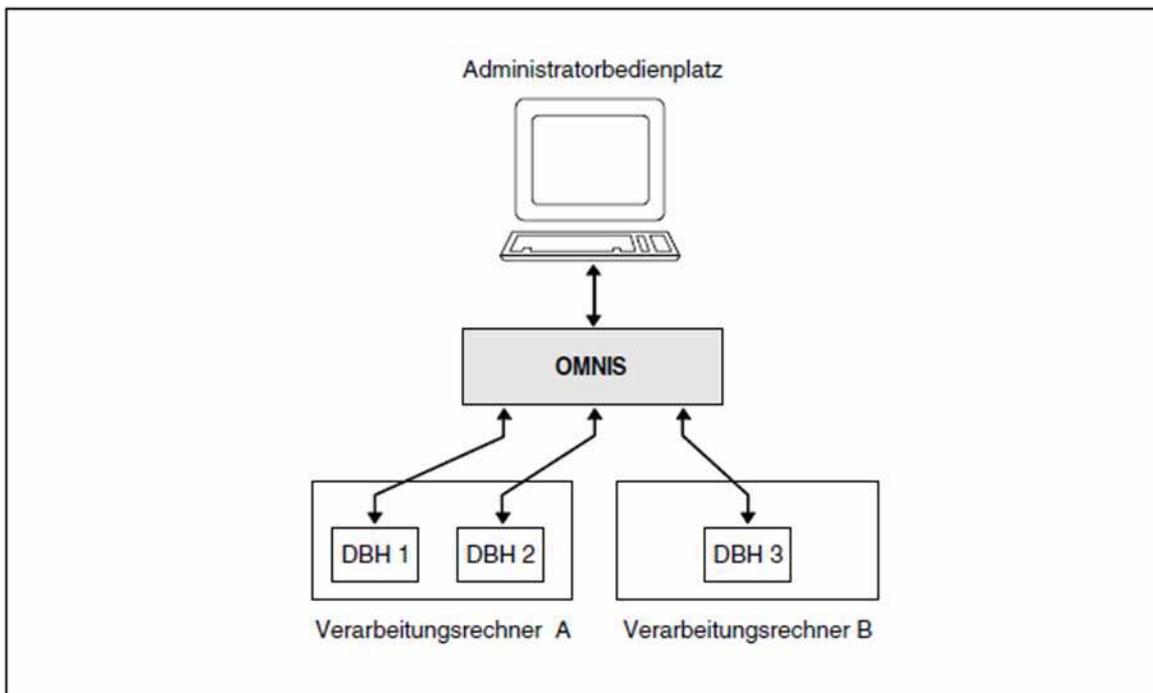


Bild 27: Zentrale Administration mit OMNIS

DBHs starten und administrieren mit OMNIS

Aus Gründen der Betriebssicherheit ist es sinnvoll, UDS/SQL als ENTER-Job zu starten und über UDSADM bzw. die DCAM-Schnittstelle zu administrieren.

Alle DBHs müssen mit dem DBH-Ladeparameter PP ADM=REMOTE gestartet werden (siehe PP ADM auf "[Administrierungsverfahren festlegen \(ADM\)](#)"). Dadurch eröffnet jeder DBH eine DCAM-Anwendung zur Administration mit dem Anwendungsnamen *konfname*.

Der Datenbankadministrator verbindet OMNIS mit diesen DCAM-Anwendungen bzw. startet in mehreren Tasks UDSADM und kann über OMNIS von einem Bedienplatz alle DBHs ansprechen.

Wenn die Verbindung zu OMNIS abbricht, kann der Datenbankadministrator jede DB-Konfiguration lokal administrieren.

Sobald OMNIS wieder verfügbar ist, kann der Datenbankadministrator die zentrale Administration wieder aufnehmen, sofern nicht lokal administriert wird.

i Im OMNIS-Administrationskonzept sind alle Datensichtstationen gleichrangig. Dadurch ist eine Absicherung gegen unerlaubte Eingriffe in die UDS/SQL-Administration nur bedingt möglich.

14.5.8 DB-Konfiguration ändern - Hinweise für UDS-D

Den Abschnitt „DB-Konfiguration während der Session ändern“ finden Sie auf "[DB-Konfiguration festlegen und ändern](#)".

Für UDS-D ist Folgendes zu beachten:

- Wenn Sie DB-Konfigurationen ändern, müssen Sie, falls erforderlich, die Verteiltabellen anpassen.
- Wenn sich sekundäre Teiltransaktionen im Zustand PTC befinden, können Sie die betroffenen Datenbanken während der Session nicht konsistent aus der DB-Konfiguration ausschließen (siehe [Abschnitt „Auswirkungen des Zustands PTC“](#)). Nach Session-Abbruch sollten Sie das Ausschließen vermeiden, bzw. inkonsistente Datenbanken möglichst schnell warmstarten, um eine möglichst problemlose Kommunikation zwischen den DB-Konfigurationen zu gewährleisten.

Sie können im Notfall den Zustand PTC „per Hand“ beenden (siehe [Abschnitt „Zustand PTC beenden“](#)).

14.5.9 Einsatz der UDS/SQL-Konfiguration auf virtuellem Host

Eine UDS/SQL-Konfiguration kann auch auf einem virtuellen Host genutzt werden. Dadurch ist es möglich, die Verteilregeln unabhängig von realen Hostnamen zu bilden. Für den störungsfreien UDS-D-Betrieb müssen die Kommunikationspartner (DBH und Anwendertasks) dabei dem virtuellen Host zugeordnet werden. Sind mehrere UDS/SQL-Konfigurationen auf einem realen Rechner vorhanden, so können diese im UDS-D-Betrieb auf verschiedenen virtuellen Hosts ablaufen.

Hierbei ist zu beachten, dass den jeweiligen Anwendertasks mittels dem DBH-Startparameter PP BCAM-PREFIX UDS-D-weit und für jede UDS/SQL-Konfiguration eindeutige BCAM-Präfixe zugeordnet sind (siehe auch [Abschnitt „Präfix für Anwendertasks in UDS-D-Verteilungen festlegen \(BCAM-PREFIX\)“](#)). Die Zuordnung von UDS/SQL-Konfiguration und Applikation zu einem speziellen virtuellen Host erfolgt durch einen Eintrag in der BCAM-Konfigurationsdatei für Applikationen (Standard-Dateiname ist SYSDAT.BCAM.APPLICATIONS).

Wird die Anwendung im Teilhaberbetrieb (z.B. mit openUTM) betrieben, dann sollte auch der DCAM-Name, mit dem die Anwendung im Teilhaberbetrieb angesprochen werden kann, dem gleichen virtuellen Host zugeordnet werden.

Namen von Anwendertasks

Zur mit UDS-D verteilten UDS/SQL-Kommunikation zwischen Anwendertask und entfernter Konfiguration werden pro Anwendertask sogenannte DCAM-Anwendungen gebildet, deren Name aus einem Präfix und der TSN besteht. Standardmäßig ist der Name SUD\$*tsn*. Dieser Name orientiert sich an den Namenskonventionen im BS2000 (S=SYSTEM, UD=Produktkürzel für UDS/SQL).

Allerdings können Sie das Präfix des Namens auch beim Start einer Session in dem Ladeparameter BCAM-PREFIX des independent DBH vorgeben. Damit ist es möglich, die Anwendertasks unterschiedlicher Konfigurationen anhand des BCAM-Präfix zu unterscheiden und darüber hinaus verschiedenen virtuellen Hosts zuzuordnen (siehe auch [Abschnitt „Präfix für Anwendertasks in UDS-D-Verteilungen festlegen \(BCAM-PREFIX\)“](#)).

i Sofern für den Ladeparameter BCAM-PREFIX ein vom Standardwert verschiedener Wert genutzt wird, wird empfohlen, auf allen über UDS-D beteiligten, entfernten Konfigurationen mindestens UDS-D V2.6 einzusetzen, auch in solchen beteiligten Konfigurationen, die nicht auf virtuellen Hosts ablaufen. Ansonsten kann es in Konfigurationen mit älteren UDS-D-Versionen zu internen Betriebsmittelengpässen kommen, die auch die sonstige DML-Bearbeitung behindern können.

Zuweisen der Kommunikationspartner

Die Zuweisungen der Kommunikationspartner zum virtuellen Host erfolgen in der BCAM-Konfigurationsdatei für Applikationen (Standard-Dateiname ist SYSDAT.BCAM.APPLICATIONS). Näheres hierzu siehe im Handbuch [„BCAM“](#).

Beispiel:

Die lokale UDS/SQL-Konfiguration UKONF1 soll zusammen mit den ihr zugeordneten Applikationen mit dem virtuellen Host VIRT01 verbunden sein. Die Applikationen im Teilhaberbetrieb werden alle mit den Namen APPLIK11 bis APPLIK99 angesprochen.

In dieser Konstellation sollen die DCAM-Applikationen, die die verteilte Kommunikation zwischen Applikation und entfernter Konfiguration übernehmen, das BCAM-Präfix VKO1 erhalten.

Name der UDS/SQL-Konfiguration: UKONF1

Name des virtuellen Hosts: VIRT01

BCAM-Präfix für die Kommunikation zwischen Anwendung und DBH: VKO1 Namen der Applikationen für den Teilhaberbetrieb: APPLIK11, ..., APPLIK99

In der BCAM-Konfigurationsdatei sind dann folgende Einträge vorzunehmen:

```
NEA UKONF1* VIRT01
NEA APPLIK* VIRT01
NEA VKO1* VIRT01
```

Soll jedoch auf ein spezielles BCAM-Präfix für die UDS-D-Kommunikation zwischen Anwendung und entfernter UDS /SQL-Konfiguration verzichtet werden, dann muss statt des letzten oberen Eintrags in der BCAM-Kommunikationsdatei der folgende Eintrag stehen (Angabe des Standard-Präfix):

```
NEA SUD$* VIRT01
```

14.6 DBH-Ladeparameter für UDS-D

Der independent DBH bearbeitet folgende für UDS-D wichtigen DBH-Ladeparameter:

Parameter	Standardwert	Bedeutung
<u>PP</u> <u>BCAM-PREFIX</u> = <i>prefix</i>	SUD\$	Präfix festlegen für Namen von Anwendertasks, die auf virtuellen Hosts ablaufen
<u>PP</u> <u>CHCKTIME</u> = <i>n</i>	60	Zeitintervall in Sekunden für die Verbindungs- und Transaktionsüberwachung der UDS-D-Task angeben <i>n</i> =60..900
<u>PP</u> <u>DEADTIME</u> = <i>n</i>	60	Zeitangabe in Sekunden, um konfigurationsübergreifende Deadlocks bzw. Deadlocks im Zusammenhang mit openUTM aufzulösen <i>n</i> =5..900
<u>PP</u> <u>DISDB</u> = <i>n</i>	1	maximale Anzahl entfernter Datenbanken, die pro Transaktion angesprochen werden können; <i>n</i> =1..32
<u>PP</u> <u>DISTABLE</u> =[: <i>catid</i> :] [<i>\$userid</i> .] <i>dateiname</i>	-	Eingabedatei zum Anlegen der Verteiltabelle festlegen
<u>PP</u> <u>DISTRIBUTION</u> ={ <u>NO</u> <u>STANDBY</u> <u>START</u> }	NO	Die Teilnahme am UDS-D-Betrieb steuern: NO: Kein UDS-D-Betrieb möglich STANDBY: UDS-D-Betrieb wird vorbereitet und kann später mit &START DISTRIBUTION gestartet werden START: Der UDS-D-Betrieb wird gestartet
<u>PP</u> <u>PTCSYNCH</u> = ([{ <u>WAIT</u> <u>ABORT</u> <u>COMMIT</u> }] [, [{ <u>WAIT</u> <u>ABORT</u> <u>COMMIT</u> }]])	(WAIT,WAIT)	Behandlung von Transaktionen im Zustand PTC steuern; der erste Wert gilt bei einem Warmstart, der zweite gilt während der laufenden Session, wenn der Zustand der primären Teiltransaktion nicht ermittelbar ist

<p><u>PP TRANSACTION</u>={<i>n</i> ([<i>n</i>][, <i>m</i>])}</p>	<p>(4,1)</p>	<p><i>n</i>: maximale Anzahl der gleichzeitig aktiven Transaktionen und Anwendertasks <i>n</i> = 1..225; <i>m</i>: maximale Anzahl sekundärer Teiltransaktionen, die dieser DBH gleichzeitig bearbeiten kann <i>m</i> = 1..<i>n</i> und <i>m</i> <= <i>n</i></p>
--	--------------	---

Tabelle 33: DBH-Ladeparameter für UDS-D

Die DBH-Ladeparameter sind in Kapitel 3 ("[Ladeparameter des DBH](#)") ausführlich beschrieben.

14.7 Datenbankadministratorsprache DAL für UDS-D

Der independent DBH bearbeitet folgende für UDS-D wichtigen DAL-Kommandos.

Alle DAL-Kommandos, die nur UDS-D kennt, beginnen mit dem Sonderzeichen „&“. Diese DAL-Kommandos (außer &START DISTRIBUTION) werden abgewiesen, wenn der UDS-D-Betrieb nicht gestartet ist. Die Beschreibung der „&“-DAL-Kommandos ist in die alphabetische Beschreibung der übrigen UDS-D DAL-Kommandos eingeordnet.

DAL-Kommando	Bedeutung
<u>ABORT</u> { <i>transaktionskennung</i> [, <u>OPTION=PTC</u>] <u>ALL</u> }	Angegebene offene Transaktionen zurücksetzen OPTION=PTC: ohne Rücksicht auf konfigurationsübergreifende Konsistenz bzw. UDS/SQL-/openUTM-Konsistenz
<u>&ADD DISTRIBUTION</u> , { <u>NODE</u> = <i>prozessorname</i> , <u>CONF</u> = <i>konfname</i> [, <u>DB</u> = <i>dbname</i>] <u>DB</u> = <i>dbname</i> , <u>SS</u> = <i>subschemaname</i> <u>FILE</u> = <i>dateiname</i> }	Neue Einträge in die Verteiltabelle bringen
<u>&CHANGE DISTRIBUTION</u> , <u>NODE</u> = <i>prozessorname</i> , <u>CONF</u> = <i>konfname</i>	Eine Konfiguration einem anderen Verarbeitungsrechner zuordnen
<u>&CLOSE DISTRIBUTION</u>	UDS-D-Betrieb beenden
<u>COMMIT</u> <i>transaktionskennung</i>	Transaktion im Zustand PTC beenden mit Festschreiben der Änderungen (FINISH) ohne Rücksicht auf konfigurationsübergreifende Konsistenz bzw. UDS/SQL-/openUTM-Konsistenz
<u>&DISPLAY DISTRIBUTION</u> [, <u>NODE</u> = <i>prozessorname</i>] [, <u>CONF</u> = <i>konfname</i>] [, <u>DB</u> = <i>dbname</i>] [, <u>SS</u> = <i>subschemaname</i>]	Verteiltabelle ausgeben
<u>&DROP DISTRIBUTION</u> , { <u>ALL</u> <u>NODE</u> = <i>prozessorname</i> , <u>CONF</u> = <i>konfname</i> [, <u>DB</u> = <i>dbname</i>] [, <u>ALL</u>] <u>DB</u> = <i>dbname</i> , <u>SS</u> = <i>subschemaname</i> }	Einträge in der Verteiltabelle löschen
<u>&LOCK DISTRIBUTION</u> , { <u>NODE</u> = <i>prozessorname</i> <u>CONF</u> = <i>konfname</i> <u>DB</u> = <i>dbname</i> <u>SS</u> = <i>subschemaname</i> }	Einträge der Verteiltabelle sperren

<code>MODIFY PTCSYNCH, VALUE=([{WAIT ABORT COMMIT}] [, [{WAIT ABORT COMMIT}]])</code>	Wert des DBH-Ladeparameters PTCSYNCH ändern
<code>&PWD DISTRIBUTION, CONF=<i>konfname</i>, { PWN=<i>kennwortneu</i> PWO=<i>kennwortalt</i> PWO=<i>kennwortalt</i>, PWN=<i>kennwortneu</i> }</code>	Kennwort vergeben und verändern
<code>&SAVE DISTRIBUTION, FILE=<i>dateiname</i></code>	Verteiltabelle sichern
<code>&START DISTRIBUTION</code>	UDS-D-Betrieb starten
<code>&SYNCHRONIZE DISTRIBUTION</code>	Sekundäre Teiltransaktionen im Zustand PTC beenden
<code>&UNLOCK DISTRIBUTION, { NODE=<i>prozessorname</i> CONF=<i>konfname</i> DB=<i>dbname</i> SS=<i>subschemaname</i> }</code>	Sperrungen für Einträge in der Verteiltabelle aufheben

Tabelle 34: DAL-Kommandos für UDS-D

Die DAL-Kommandos sind im [Abschnitt „Die Datenbankadministratorsprache DAL“](#) ausführlich beschrieben.

15 Anhang

- Funktionscodes der DML-Anweisungen

15.1 Funktionscodes der DML-Anweisungen

In der folgenden Tabelle werden die Funktionscodes dezimal dargestellt.

Funktionscode	Funktionskurzbeschreibung	Bedeutung
0	START	Start eines Anwenderprogramms
1	FINISH	FINISH
2	ERASUN	ERASE <i>satzname</i>
3	ERASQU	ERASE <i>satzname</i> {PERMANENT/SELECTIVE/ALL}
4	FND1	FIND-1
5	FND5	FIND-5
6	FND4R	FIND-4 ohne WITHIN oder mit WITHIN <i>realmname</i>
7	FND4S	FIND-4 WITHIN <i>setname</i>
8	FND6	FIND-6
9	FND2	FIND-2
10	FND7IS	FIND-7 USING <i>satzelementname</i> ,... (sequenzielle Suche)
11	FND7IK	FIND-7 USING <i>satzelementname</i> ,... (Suche mit Schlüssel)
12	FND3IS	FIND-3 USING <i>satzelementname</i> ,... (sequenzielle Suche)
13	FND3IK	FIND-3 USING <i>satzelementname</i> ,... (Suche mit Schlüssel)
15	CONNEC	CONNECT
16	MODIFY	MODIFY
17	ACCEPT	ACCEPT
18	READY	READY
19	DISCON	DISCONNECT
20	STORE	STORE
21	STOP	Ende eines Anwenderprogramms
22	FREE	FREE
23	KEEP	KEEP
24	IF_	IF
25	FND7SE	FIND-7 ohne USING oder mit USING <i>suchausdruck</i>
26	ONLINE	UDS-Online-Utility

28	FND3SE	FIND-3 ohne USING
30	LONAM	LOOK nach einem Namen
31	LOREC	LOOK nach einem Satz
32	LOSET	LOOK nach einem Set
33	LOITM	LOOK nach einem Feld
34	LOKEY	LOOK nach einem Schlüssel
36	LORLM	LOOK nach einem Realm
37	RDYGLO	Beginn einer verteilten Transaktion ¹
38	PTC	Ende-Einleitung einer verteilten Transaktion ¹
125	RELBAS	RELOCATE RELOCATION-TYPE = *BASE-LEVEL-TABLE-PAGES
126	RELINX	RELOCATE RELOCATION-TYPE = *INDEX-LEVEL-TABLE-PAGES
127	RELDST	RELOCATE RELOCATION-TYPE = *DISTRIBUTABLE-TABLE-PAGES
128	RELREC	RELOCATE RELOCATION-TYPE = *RECORD-PAGES
129	FPASCA	FPASCAN
130	PRERLM	PREFRLM
131	REOPPP	REORGPPP
132	FTC1	FETCH-1
133	FTC5	FETCH-5
134	FTC4R	FETCH-4 ohne WITHIN oder mit WITHIN <i>realmname</i>
135	FTC4S	FETCH-4 mit WITHIN <i>setname</i>
136	FTC6	FETCH-6
137	FTC2	FETCH-2
138	FTC7IS	FETCH-7 USING <i>satzelementname</i> ,... (sequenzielle Suche)
139	FTC7IK	FETCH-7 USING <i>satzelementname</i> ,... (Suche mit Schlüssel)
140	FTC3IS	FETCH-3 USING <i>satzelementname</i> ,... (sequenzielle Suche)
141	FTC3IK	FETCH-3 USING <i>satzelementname</i> ,... (Suche mit Schlüssel)
142	GET	GET
153	FTC7SE	FETCH-7 ohne USING oder mit USING <i>suchausdruck</i>
156	FTC3SE	FETCH-3 ohne USING
158	NLONAM	nächster LOOK nach einem Namen

160	NLOSET	nächster LOOK nach einem Set
161	NLOITM	nächster LOOK nach einem Feld
162	NLOKEY	nächster LOOK nach einem Schlüssel
164	NLORLM	nächster LOOK nach einem Realm

Tabelle 35: Funktionscodes der DML-Anweisungen

¹ Nur bei Zusammenarbeit mit UDS-D

16 Fachwörter

Dieses Fachwortverzeichnis enthält Definitionen wichtiger Begriffe, die in den Handbüchern zu UDS/SQL verwendet werden.

*Kursiv*gedruckte Fachwörter in den Definitionen verweisen auf entsprechende Definitionen für diese Fachwörter.

Ein „siehe“-Verweis für ein Fachwort verweist auf das in den UDS/SQL-Handbüchern hauptsächlich verwendete Fachwort.

A

Act-Key

act-key

(actual key) Aktuelle Adresse einer *Seite*, bestehend aus *Realmnummer* und *Seitennummer*.

Act-Key-0-Seite

act-key-0 page

Erste *Seite* eines *Realm*. Sie enthält allgemeine Informationen über den Realm, z.B.

- Erstellungszeitpunkt des Realm,
- Zeitpunkt der letzten Änderung,
- *interne Versionsnummer* des Realm,
- Unterbrechungsinformationen des Systems (*Systembreak-Informationen*),
- ggf. Kenndaten für den *Warmstart*.

Act-Key-N-Seite

act-key-N page

Kennseite eines *Realm* mit der höchsten *Seitennummer*.
Kopie der *Act-Key-0-Seite*.

Administratortask

administrator task

Task des *independent DBH*. Der *Datenbankadministrator* kann über diese Task den Ablauf des *independent DBH* steuern.

Adresse, physische

address, physical

siehe *Act-Key* oder *Probable Position Pointer (PPP)*

Adressliste

pointer array

Tabelle, die auf die *Membersätze* einer *Set-Occurrence* verweist. Dient dem *sequentiellen* und *direkten Zugriff* auf die *Membersätze*.

AFIM

AFIM

siehe *After-Image*

After-Image

after-image

Geänderter Teil einer *Seite* **nach** einer Änderung des Seiteninhalts.

After-Images schreibt der *DBH* sowohl in die *RLOG-Datei* als auch in die *ALOG-Datei*.

After-Image, ALOG-Datei

after-image, ALOG file

Die After-Images werden in die ALOG-Datei geschrieben, wenn der ALOG-Puffer voll ist. Die After-Images in der ALOG-Datei werden zur Langzeitsicherung, d.h. für lange Zeit benötigt. Sie werden benutzt, um eine Originaldatenbank zu rekonstruieren oder eine *Schattendatenbank* zu aktualisieren.

After-Image, RLOG-Datei

after-image, RLOG file

Die After-Images werden in die RLOG-Datei geschrieben, **bevor** die Änderungen auf der *Datenbank* festgeschrieben werden. Die After-Images in der RLOG-Datei werden nur zum *Warmstart* benötigt und deshalb zyklisch überschrieben.

ALOG-Datei

ALOG file

Datei zur Langzeitsicherung, siehe *After-Image*.

ALOG-Folgenummer

ALOG sequence number

Kennzeichnung im Dateinamen der *ALOG-Dateien* (000000001- 999999999).

Die erste ALOG-Datei einer *Datenbank* trägt immer die Folgenummer 000000001.

Ankersatz

anchor record

Satz, den UDS/SQL automatisch als *Ownersatz* für *SYSTEM-Sets* einrichtet. Er enthält keine mit der *Schema-DDL* definierten *Felder* und es kann auf ihn nicht zugegriffen werden.

Anweisungscode

statement code

Nummer, die im ersten Teil des Feldes *DATABASE-STATUS* hinterlegt wird und die darüber informiert, bei welcher *DML*-Anweisung ein Sonderzustand aufgetreten ist.

Anwenderprogramm (AP)

application program (AP)

Z.B. *COBOL-DML*-Programm, IQS.

Anwendertask

user task

Ausführung eines *Anwenderprogramms* bzw. *openUTM*-Teilprogramms, einschließlich der vom System dazugebundenen Teile.

Anwendung

application

Umsetzung einer Aufgabenstellung in ein *Anwenderprogramm* oder mehrere Anwenderprogramme, die mit UDS/SQL-*Datenbanken* arbeiten.

Area

area

siehe *Realm*

Ascending-Key (ASC-Key)

ascending key (ASC key)

Primärschlüssel eines *Set*. Der Ascending-Key legt die Reihenfolge der *Membersätze* in den *Set-Occurrences* nach aufsteigenden Schlüsselwerten fest.

Auftrag

request

Die Funktionen, die durch die *DAL*-Kommandos ADD DB, ADD RN, DROP DB, DROP RN, NEW RLOG und CHECKPOINT zunächst im *DBH* nur vorgemerkt sind, werden erst durch das *DAL*-Kommando PERFORM zur Durchführung angestoßen.

automatische DBTT-Erweiterung

automatic DBTT extension

Einige Dienstprogramme erweitern die Anzahl möglicher Sätze einer Satzart bei Engpässen automatisch; hierfür ist keine gesonderte Administration erforderlich.

Siehe auch *Online-DBTT-Erweiterung*.

automatische Realm-Erweiterung

automatic realm extension

Einige Dienstprogramme erweitern Realms bei Engpässen automatisch; hierfür ist keine gesonderte Administration erforderlich.

Siehe auch *Online-Realm-Erweiterung*.

B**Base Interface Block**

Base Interface Block

siehe *BIB***Before-Image**

before-image

Teil einer *Seite* vor einer Änderung des Seiteninhalts.

Before-Images schreibt der *DBH* in die *RLOG-Dateien*. Dort werden die Before-Images während des Datenbankbetriebs geschrieben, bevor die Änderungen auf der *Datenbank* festgeschrieben werden. Voraussetzung ist, dass RLOG-Dateien geführt werden.

Benutzerdatenbank

user database

Die *Realms* und Dateien der *Datenbank*, die der Anwender benötigt, um Daten in die Datenbank zu speichern und wiederzugewinnen.

Dies sind:

- das *Database Directory (DBDIR)*
- die *Benutzerrealms*
- die Modulbibliothek für *Hashroutinen (HASHLIB)*.

Benutzerrealm

user realm

Im Realm-Eintrag der *Schema-DDL* definierter *Realm*. Er enthält u.a. die Benutzersätze.

Bezeichner

identifier

Name, den der Datenbankentwerfer für ein *Feld* vergibt, das UDS/SQL automatisch anlegt. Feldtyp und Feldlänge richtet UDS/SQL nach dem vorgegebenen Verwendungszweck des Feldes aus.

BFIM

BFIM

siehe *Before-Image***BIB**

BIB

(Base Interface Block) Standardschnittstelle zwischen UDS/SQL und jedem einzelnen Benutzer; enthält u.a. die *RECORD-AREA* (Benutzersätze wie im *Subschema* definiert).

Buffer Pools

buffer pools

siehe *System Buffer Pools* und *exklusiver Buffer Pool*

C**CALC-Key**

CALC key

Schlüssel, dessen Schlüsselwerte durch eine *Hashroutine* in eine relative *Seitennummer* umgerechnet werden.

CALC-SEARCH-Key

CALC SEARCH key

Sekundärschlüssel, der als *Zugriffspfad* für *direkten Zugriff* über *Hashverfahren* realisiert wird.

CALC-Seite

CALC page

Seite eines *Hashbereichs*.

CALC-Tabelle

CALC table

Tabelle in einer direkten/indirekten *CALC-Seite*, deren Einträge auf die gespeicherten Sätze verweisen.

Sie enthält pro Zeile:

- den *CALC-Key*
- die *Satzfolgennummer*
- die Distanz zum zugehörigen *Seitenindex-Eintrag* (direkte *CALC-Seite*) bzw. den *Probable Position Pointer* (indirekte *CALC-Seite*)

CALL-DML

CALL DML

DML, die von verschiedenen Programmiersprachen (Assembler, COBOL, FORTRAN, PASCAL, PL/1) über die *CALL-Schnittstelle* angesprochen wird.

CHAIN

CHAIN

Speicherungsart für eine *Set-Occurrence*, bei der jeder *Satz* einen Zeiger auf seinen Nachfolger mitführt.

Character Separated Values (CSV)

Character Separated Values

Ausgabeformat, bei dem die Werte durch ein vorgegebenes Zeichen getrennt sind

CHECK-TABLE

CHECK-TABLE

Prüftabelle, die der *DDL*-Compiler bei der *Subschema-DDL*-Übersetzung erstellt und die vom COBOL-Compiler und von *CALL-DML* benutzt wird, um zu prüfen, ob die angegebenen *DML*-Anweisungen im *Anwenderprogramm* zulässig sind.

Sie befindet sich im *COSSD* bzw. im *SSITAB-Modul*.

Clone-Paar, Clone-Pubset, Clone-Session, Clone-Unit

clone pair, clone pubset, clone session, clone unit

Eine Clone-Unit ist die Kopie einer (Original-)Unit (logische Platte im BS2000) zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“). Die Komponente TimeFinder/Clone erstellt diese Kopie wahlweise als komplette Kopie oder als „Snapshot“.

Nach der Aktivierung sind Unit und Clone-Unit voneinander getrennt, Anwendungen können auf beide zugreifen.

Unit und Clone-Unit bilden zusammen ein Clone-Paar. TimeFinder/Clone verwaltet es in einer sogenannten Clone-Session.

Wenn es zu allen Units eines Pubsets Clone-Units gibt, so bilden diese Clone-Units zusammen das Clone-Pubset.

Details zu diesem Thema finden Sie im Handbuch „[Einführung in die Systembetreuung](#)“.

COBOL-DML

COBOL DML

In den COBOL-Sprachumfang integrierte *DML*.

COBOL-Laufzeitsystem

COBOL runtime system

Laufzeitsystem. Mehrfachbenutzbare Routinen, die der COBOL-Compiler (COBOL2000 bzw. COBOL85) zur Ausführung komplexer Anweisungen auswählt.

COBOL Subschema Directory (COSSD)

COBOL Subschema Directory (COSSD)

liefert dem COBOL-Compiler die Subschema-Informationen für die Übersetzung der DB-*Anwenderprogramme*.

Common Memory

common memory

Von mehreren Tasks gemeinsam benutzbarer Speicherbereich. Er besteht bei UDS/SQL immer aus dem *Common Pool* und dem *Communication Pool* und je nach Anwendungsfall aus dem *SSITAB Pool* (siehe *SSITAB-Modul*), wenn die *CALL-DML* verwendet wird. Beim Einsatz von UDS-D, besteht er zusätzlich noch aus dem *Distribution Pool* und dem *Transfer Pool*.

Common Pool

common pool

Kommunikationsbereich des *independent DBH* für die Verständigung der *DBH*-Module untereinander. Er enthält u.a. einen Ein-/Ausgabe-Puffer für *Seiten (Buffer Pools)*.

Communication Pool

communication pool

Kommunikationsbereich des *independent DBH* für *Anwenderprogramme*. Er dient u.a. zur Aufnahme der Base Interface Blocks (*BIB*).

Compilerdatenbank

compiler database

Die *Realms* und Dateien der *Datenbank*, die die UDS/SQL-Compiler benötigen. Dies sind:

- das *Database Directory (DBDIR)*
- der *Database Compiler Realm (DBCOM)*
- das *COBOL Subschema Directory (COSSD)*

COMPILER-SCHEMA

COMPILER-SCHEMA

UDS/SQL-internes *Schema* der *Compilerdatenbank*.

COMPILER-SUBSCHEMA

COMPILER-SUBSCHEMA

UDS/SQL-internes *Subschema* der *Compilerdatenbank*.

Compound Key

compound key

siehe *Schlüssel, zusammengesetzter*

Connectionmodul

connection module

siehe *Verbindungsmodul*

Consistency Record

consistency record

Verwaltungssatz mit Konsistenz-Zeitstempeln im *DBDIR*. Bei einer Änderung in einem *Realm* trägt der *DBH* im Consistency Record und im geänderten Realm Datum und Uhrzeit ein. Beim Anschließen von *Datenbanken* oder Realms an eine *Session* überprüft der DBH anhand dieser Zeitstempel, ob die Realms jeder Datenbank unter dem Konsistenzaspekt zueinander passen.

COSSD

COSSD

siehe *COBOL Subschema Directory*.

CRA

CRA

(Current Record of Area) *Satz*, der in der *Currency-Tabelle* als aktueller Satz eines bestimmten *Realm* (Area) verzeichnet ist.

CRR

CRR

(Current Record of Record) *Satz*, der in der *Currency-Tabelle* als aktueller Satz einer bestimmten *Satzart* (Record) verzeichnet ist.

CRS

CRS

(Current Record of Set) *Satz*, der in der *Currency-Tabelle* als aktueller Satz eines bestimmten *Set* verzeichnet ist.

CRU

CRU

(Current Record of Rununit) *Satz*, der in der *Currency-Tabelle* als aktueller Satz der *Verarbeitungskette* verzeichnet ist.

CSV

CSV

siehe Character Separated Values

Currency-Tabelle

currency table

Die Currency-Tabelle enthält

- die CURRENT-OF-AREA-Tabelle (Tabelle der *CRAs*),
- die CURRENT-OF-RECORD-Tabelle (Tabelle der *CRRs*),
- die CURRENT-OF-SET-Tabelle (Tabelle der *CRSs*).

CURRENT-OF-AREA-Tabelle

CURRENT OF AREA table

siehe *Currency-Tabelle*

CURRENT-OF-RECORD-Tabelle

CURRENT OF RECORD table

siehe *Currency-Tabelle*

CURRENT-OF-SET-Tabelle

CURRENT OF SET table

siehe *Currency-Tabelle*

D**DAL**

DAL

(Database Administrator Language) Datenbankadministratorsprache für Kommandos zum Überwachen und Steuern einer *Session*.

Database Compiler Realm (DBCOM)

database compiler realm (DBCOM)

Speichert Einzelheiten über die *Realms*, *Sätze* und *Sets*, die der Anwender in der *Schema-DDL* und der *Subschema-DDL* definiert hat.

Database Directory (DBDIR)

database directory (DBDIR)

Enthält u.a. die *S/A*, alle *SS/As* und Informationen über die *Zugriffsberechtigungen*.

Database Key

database key

Schlüssel, dessen Schlüsselwerte einen *Satz* in der *Datenbank* eindeutig identifizieren. Er setzt sich aus einer *Satzartnummer* und einer *Satzfolgenummer* zusammen. Die Schlüsselwerte können vom Datenbankprogrammierer vergeben oder von UDS/SQL automatisch erzeugt werden.

Database-Key-Feld

database key item

Feld vom Typ DATABASE-KEY oder DATABASE-KEY-LONG, das für die Aufnahme von *Database-Key*-Werten definiert wird.

Felder vom Typ DATABASE-KEY und Felder vom Typ DATABASE-KEY-LONG unterscheiden sich hinsichtlich der Feldlänge (4 byte / 8 byte) und des Wertebereichs.

DATABASE-KEY-Feld

DATABASE-KEY item

siehe *Database-Key-Feld*

DATABASE-KEY-LONG-Feld

DATABASE-KEY-LONG item

siehe *Database-Key-Feld*

DATABASE-STATUS

DATABASE-STATUS

5 byte langes Feld zur Anzeige des Datenbankzustands. Der Datenbankzustand besteht aus dem *Anweisungscode* und dem *Statuscode*.

Datenbank (DB)

database

Zusammengehörige Datenbestände, die mit Hilfe eines *Datenbanksystems* ausgewertet, bearbeitet und verwaltet werden.

Eine Datenbank wird durch den Datenbanknamen identifiziert.

Eine UDS/SQL-Datenbank besteht aus der *Benutzerdatenbank* und der *Compilerdatenbank*.

Zum Schutz vor Datenverlust kann parallel zur Datenbank (Originaldatenbank) eine *Schattendatenbank* betrieben werden.

Datenbankadministrator

database administrator

Person, die die *Datenbank* im laufenden Betrieb verwaltet und steuert. Der Datenbankadministrator bedient die Dienstprogramme und die Database Administrator Language (*DAL*).

Datenbankkopie

database copy

Kopie einer konsistenten *Datenbank*, die zu einem beliebigen Zeitpunkt erstellt wurde.

Datenbank-Jobvariable

database job variable

Jobvariable, in der UDS/SQL Informationen über den Zustand einer *Datenbank* hinterlegt.

Datenbankseite

database page

siehe *Seite*

Datenbanksystem

database system

Softwaresystem, das alle Aufgaben im Zusammenhang mit Verwaltung und Kontrolle großer Datenbestände unterstützt. Die im Datenbanksystem enthaltenen Verfahren führen zu einer stabilen, redundanzfreien und erweiterbaren Datenorganisation. Sie ermöglichen einer Vielzahl von Anwendern den parallelen Zugriff auf die *Datenbanken* und gewährleisten einen konsistenten Datenbestand.

Datenbankzustand

database status

siehe *DATABASE-STATUS*

Datendeadlock

data deadlock

siehe *Deadlock*

Datengruppe

group item

Benennbare Zusammenfassung von *Satzelementen*.

Datenschutz

data protection (privacy)

Schutz vor unberechtigtem Zugriff auf Daten. Datenschutz wird in UDS/SQL verwirklicht durch das Schema/Subschema-Konzept und die Zugriffsrechtsprüfung. Die *Zugriffsrechte* werden mit dem Dienstprogramm BPRIVACY vergeben.

Datensicherung

data backup

Schutz vor Datenverlust bei Software- oder Hardware-Fehlern.

DBCOM

DBCOM

siehe *Database Compiler Realm*

DBDIR

DBDIR

siehe *Database Directory*

DBH

DBH

(Database Handler) Programm (bzw. Programmgruppe), das den Zugriff auf die *Datenbank(en)* einer *Session* steuert und alle dabei notwendigen Verwaltungsarbeiten übernimmt.

DBH, independent

DBH, independent

siehe *independent DBH*

DBH, Ladeparameter

DBH load parameters

siehe *Ladeparameter (DBH)*.

DBH, linked-in

DBH, linked-in

siehe *linked-in DBH*

DBH-Ende

DBH end

Beenden des *DBH* Programmlaufs. DBH-Ende kann entweder *Session-Ende* oder *Session-Abbruch* sein.

DBH-Start

DBH start

Starten des *DBH* Programmlaufs. DBH-Start kann entweder *Session-Beginn* oder *Session-Wiederanlauf* sein.

DB-Key

DB key

Siehe *Database Key*.

DB-Konfiguration

DB configuration

(database configuration) Die Menge aller *Datenbanken*, die einem *DBH* während einer *Session* momentan zugeschaltet ist. Die DB-Konfiguration kann sich im Laufe einer Session ändern, durch *DAL*-Kommandos oder durch die DBH-Fehlerbehandlung.

Eine DB-Konfiguration kann zu *Session-Beginn* auch leer sein. Mit *DAL*-Kommandos können Datenbanken nach Session-Beginn angeschlossen werden. Mit *DAL*-Kommandos können aber auch während einer Session Datenbanken ausgeschlossen werden.

DB-Status-Datei

DB status file

(database status file) Enthält Informationen über die letzten zurückgesetzten *Transaktionen*. Diese Informationen werden von UTM-S und bei verteilter Verarbeitung mit UDS-D/openUTM-D zum *Session-Wiederanlauf* benötigt.

DBTT

DBTT

(Database Key Translation Table) Tabelle, in der UDS/SQL mit Hilfe eines Database-Key-Wertes die *Seitenadresse (Act-Key)* des zugehörigen *Satzes* und der zugehörigen Tabellen findet. Die DBTT des SSIA-RECORD besteht nur aus der DBTT-Basis. Bei allen anderen Satzarten besteht die DBTT jeweils aus einer Basistabelle (DBTT-Basis) und eventuell einer der mehreren Erweiterungstabellen (DBTT-Extents), welche durch eine Online-DBTT-Erweiterung oder durch BREORG entstehen.

DBTT-Ankerseite

DBTT anchor page

Im Realm der zugehörigen DBTT liegende Seite, in der DBTT-Basis und DBTT-Extents verwaltet werden. Möglicherweise sind mehrere untereinander verkettete DBTT-Ankerseiten zur Verwaltung der DBTT nötig.

DBTT-Basis

DBTT base

siehe *DBTT*

DBTT-Extent

DBTT extent

siehe *DBTT*

DBTT-Seite

DBTT page

Seite, die die *DBTT* oder einen Teil der DBTT einer *Satzart* enthält.

DCAM

DCAM

Teil des Datenkommunikationssystems TRANSDATA

DCAM-Anwendung

DCAM application

Kommunikationsanwendung, die die Kommunikationsmethode *DCAM* benutzt.

Eine DCAM-Anwendung bietet Kommunikationsmöglichkeit zwischen

- einer DCAM-Anwendung und Datensichtstationen.
- DCAM-Anwendungen untereinander im selben oder in verschiedenen Verarbeitungsrechnern, sowie mit *entfernten Konfigurationen*.
- einer DCAM-Anwendung und einer *openUTM*-Anwendung.

DDL

DDL

(Data Description Language) Formale Sprache zur Beschreibung der logischen Datenstruktur.

Deadlock

deadlock

Gegenseitiges Blockieren von *Transaktionen*.

Ein Deadlock kann in folgenden Situationen auftreten:

- Datendeadlock: *Transaktionen* blockieren sich gegenseitig bei *konkurrierenden Zugriffen*
- Taskdeadlock: Eine *Transaktion*, die eine Sperre hält, kann diese nicht freigeben, da keine openUTM-Task frei ist. Diese Deadlock-Situation kann nur bei UDS/SQL-openUTM-Zusammenarbeit auftreten.

Descending-Key (DESC-Key)

descending key (DESC key)

Primärschlüssel eines *Set*. Der Descending-Key legt die Reihenfolge der *Membersätze* in den *Set-Occurrences* nach absteigenden Schlüsselwerten fest.

direkter Hashbereich

direct hash area

siehe *Hashbereich*

direkter Zugriff

direct access

Zugriff auf einen *Satz* über einen Feldinhalt. UDS/SQL unterstützt den direkten Zugriff über den *Database Key* sowie über *Hashverfahren* und *mehrstufige Tabellen*.

Distribution Pool

distribution pool

Kommunikationsbereich des *independent DBH* für die Verständigung von *UDSCT*, *Servertasks*, *Anwendertasks* und *Mastertask* untereinander bezüglich UDS-D-spezifischer Daten. Im Distribution Pool liegen die *Verteiltabelle* und UDS-D-spezifische Systemtabellen.

DML

DML

(Data Manipulation Language) Sprachmittel für den Zugriff auf eine UDS/SQL-*Datenbank*.

Dummy-Teiltransaktion

dummy subtransaction

Ist eine primäre *Teiltransaktion*, die UDS-D erzeugt, wenn die erste *READY*-Anweisung einer *Transaktion* eine *entfernte Datenbank* anspricht.

Die Dummy-Teiltransaktion dient dazu, die Transaktion in der *lokalen Konfiguration* bekannt zu machen, um im Fehlerfall ein Wiederherstellen der *Datenbank* zu ermöglichen.

Duplikat-Kopf

duplicates header

Enthält allgemeine Informationen über eine *Duplikat-Tabelle* bzw. eine *Seite* einer Duplikat-Tabelle:

- die Verkettung zur nächsten und zur vorhergehenden *Überlaufseite*
- die Anzahl freier Bytes in der Seite der Duplikat-Tabelle

Duplikat-Tabelle

duplicates table

Spezielle *SEARCH-Key-Tabelle*, in der ein mehrfach auftretender Schlüsselwert nur einmal gespeichert wird.

Die Duplikat-Tabelle enthält pro Schlüsselwert

- einen Tabellenindex-Eintrag mit dem Schlüsselwert und dem Verweis auf die zugehörige Tabellenzeile
- eine Tabellenzeile (DB-Key-Liste), die auf mehrere Seiten aufgeteilt sein kann, mit den *Satzfolgennummern* der *Sätze*, die diesen Schlüsselwert enthalten

Duplikat-Tabelle, Grundstufe

duplicates table, main level

Main Level bzw. Level 0; enthält einen Tabellenindex-Eintrag und den Beginn der zugehörigen Tabellenzeile (DB-Key-Liste).

dynamischer Set

dynamic set

Set, der zeitlich begrenzt durch die Dauer der *Transaktion*, *Membersätze* von Suchfragen aufnehmen kann.

E

entfernte Datenbank

remote database

Datenbank einer entfernten Konfiguration.

entfernte Konfiguration

remote configuration

DB-Konfigurationen, die dem Anwenderprogramm nicht über /SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname zugeordnet werden, sondern erst bei Ablauf des Anwenderprogramms über die Verteiltabelle. Mit entfernten Konfigurationen verkehrt das Verbindungsmodul des Anwenderprogramms über die DCAM-Anwendungen.

Entfernte Konfigurationen liegen auf dem lokalen oder auf einem entfernten Verarbeitungsrechner.

entfernter Verarbeitungsrechner

remote host

Verarbeitungsrechner, der nicht lokal ist.

entferntes Anwenderprogramm

remote application program

Anwenderprogramm, das bezüglich einer bestimmten Konfiguration nicht lokal ist.

ESTIMATE-REPORT

ESTIMATE-REPORT

Protokollausgabe nach dem BGSIA-Lauf. Dient dazu, die Größe der Benutzerrealms zu schätzen.

Event-Name

event name

Name einer Ereigniskennung.

exklusiver Buffer Pool

exclusive buffer pool

Puffer, der zusätzlich zu den System Buffer Pools ausschließlich für die Pufferung von Seiten der angegebenen Datenbank verwendet wird.

F**Feld**

item

Kleinste benennbare Dateneinheit innerhalb einer *Satzart*. Das Feld ist definiert durch Feldtyp und Feldlänge.

Folgenummer

sequence number

siehe *ALOG-Folgenummer*

FPA

FPA

siehe *Freiplatzverwaltung*.

FPA-Basis

FPA base

siehe *Freiplatzverwaltung*.

FPA-Extent

FPA extent

siehe *Freiplatzverwaltung*.

FPA-Seite

FPA page

Seite der Freiplatzverwaltung.

Freiplatzverwaltung (FPA)

Free Place Administration (FPA)

Freier Platz wird sowohl auf Realm-Ebene (*FPA-Seiten*), als auch auf Seiten- und Tabellenebene verwaltet. Die Freiplatzverwaltung der Seiten erfolgt in einer Basistabelle (FPA-Basis) und eventuell in einer oder mehreren Erweiterungstabellen (FPA-Extent), welche durch eine Online-Realmerweiterung oder durch BREORG entstehen.

Fremdschlüssel

foreign key

Satzelement, dessen Werte mit den *Primärschlüssel*werten einer anderen Tabelle (UDS/SQL-*Satzart*) übereinstimmen. Fremdschlüssel im Sinne von UDS/SQL werden im BPSQLSIA-Protokoll in der Membersatzart einer Set-Beziehung als "REFERENCES owner-satzart" qualifiziert.

Funktionscode (FC)

function code

Verschlüsselung einer *DML*-Anweisung. Wird beim *DAL*-Kommando DISPLAY und bei UDSMON ausgegeben.

H

Hashbereich

hash area

Speicherbereich, in dem UDS/SQL Daten speichert oder wiedergewinnt aufgrund der Umrechnung von Schlüsselwerten in relative *Seitennummern*. Ein Hashbereich kann sowohl die Adressen von *Sätzen* als auch die Sätze selbst enthalten.

In einem *direkten Hashbereich* sind die Sätze selbst gespeichert, während in einem *indirekten Hashbereich* die Adressen der andernorts gespeicherten Sätze enthalten sind.

HASHLIB

HASHLIB

Modulbibliothek zur Aufnahme der *Hashroutinen* einer *Datenbank*.

Hashroutine

hash routine

Modul, das ein *Hashverfahren* ausführt.

Hashverfahren

hashing

Methode, mit der ein Schlüsselwert in eine *Seitenadresse* umgerechnet wird.

I**Identifizierung**

authorization

Erkennung der Benutzergruppe.

impliziter Set

implicit set

SYSTEM-Set, den UDS/SQL bildet, wenn ein *SEARCH-Key* auf Satzartebene definiert wird.

independent DBH

independent DBH

Selbständiges Programmsystem, das den simultanen Zugriff mehrerer Anwender auf eine *Datenbank (Mono-DB-Betrieb)* oder auf mehrere Datenbanken gleichzeitig (*Multi-DB-Betrieb*) ermöglicht. Der independent DBH ist als Taskfamilie konzipiert:

- eine *Mastertask (UDSSQL)*
- eine oder mehrere *Servertasks (UDSSUB)*
- eine *Administratortask (UDSADM)*

INDEX-Search-Key

INDEX search key

Sekundärschlüssel. Er wird als *Zugriffspfad* für *direkten Zugriff* über eine *mehrstufige Tabelle* realisiert.

Indexseite

index page

Seite, in der die höchsten (niedrigsten) Schlüsselwerte der nächstniedrigen Stufe einer indizierten Tabelle gespeichert werden.

Indexstufe

index level

Hierarchiestufe einer *Indexseite*.

indirekter Hashbereich

indirect hash area

siehe *Hashbereich*

Inkonsistenz

inconsistency

Widerspruch zwischen gespeicherten Informationen.

Integrität

integrity

Fehlerfreiheit und Vollständigkeit der gespeicherten Informationen

- Objekt-Integrität (Entity Integrity)
- *referentielle Integrität* (Referential Integrity)
- Benutzer-Integrität (User Integrity)

interne Versionsnummer

internal version number

Jeder *Realm* der *Datenbank*, inklusive *DBDIR* und *DBCOM*, besitzt eine interne Versionsnummer, die die Dienstprogramme (z. B. BREORG, BALTER) bei Veränderungen des Realms um eins erhöhen. Diese interne Versionsnummer steht in der *Act-Key-0-Seite* des Realms und zusätzlich im PHYS VERSION RECORD im DBDIR.

Item

item

siehe *Feld*

K**Katalogkennung**

catalog identifier

Bezeichnung der gemeinschaftlichen Platte (Public Volume Set), in der die BS2000-/UDS/SQL-Dateien gespeichert sind. Die Katalogkennung ist Bestandteil des Datenbank-/Datei-Namens und in Doppelpunkte eingeschlossen:

„:catid:“

KDBS

KDBS

(Compatible Database Interface) Kompatible Datenbankschnittstelle. KDBS ermöglicht, Programme auf Anwendungen von *Datenbanksystemen* verschiedener Hersteller zu übertragen.

Kennwort für die UDS/SQL-Dateien

password for UDS/SQL files

Wort, mit dem die von UDS/SQL eingerichteten Dateien geschützt sind (Standardwert: C'UDS '). Außerdem kann der *Datenbankadministrator* Kennwörter festlegen mit PP CATPASS oder durch MODIFY-FILE-ATTRIBUTES.

Kette

chain

siehe *CHAIN*

Kommunikationspartner

communication partners

Tasks bzw. Datensichtstationen

Komprimierung

compression

Nur belegte *Felder* eines *Satzes* werden gespeichert (siehe *SSL*-Klausel COMPRESSION).

Konfiguration

configuration

siehe *DB-Konfiguration*

Konfigurationskennung

configuration user ID

Kennung, in der der *Datenbankadministrator* den *DBH* startet.

Konfigurationsname

configuration name

Frei wählbarer Name der *Datenbankkonfiguration* einer *Session*. Aus dem Konfigurationsnamen bildet der *DBH*

- den Namen der *Session-Log-File*,
- den Namen der *DB-Status-Datei* und ihrer Sicherungskopie,
- den Namen der *RLOG-Dateien*,
- den Namen der Temporären *Realms*,
- den Namen der Session-Jobvariablen
- die *Event-Namen* des *P1-Eventing*,
- den Namen der *DCAM-Anwendung* für die Administration,
- die Namen für die *Common Pools*,
- die Namen der Dump-Dateien.

konfigurationsübergreifend

interconfiguration

Mindestens eine *entfernte Konfiguration* betreffend.

konfigurationsübergreifende Konsistenz

interconfiguration consistency

Eine *verteilte Transaktion*, die in mindestens einer *entfernten Konfiguration* geändert hat, wird so beendet, dass die Änderungen entweder auf den *Datenbanken* aller beteiligten *DB-Konfigurationen* durchgeführt werden oder auf keiner Datenbank.

Die konfigurationsübergreifende Konsistenz wird sichergestellt durch das *Zwei-Phasen-Ende-Protokoll*.

konfigurationsübergreifender Deadlock

interconfiguration deadlock

Zustand wechselseitiger Blockierungen von *verteilten Transaktionen* bei *konkurrierenden Zugriffen*.

konkurrierender Zugriff

contending access

Gleichzeitiger Zugriff auf eine *Seite* aus verschiedenen *Transaktionen*.

Konsistenz

consistency

Widerspruchsfreiheit der gespeicherten Informationen.

Konsistenz, logische

consistency, logical

Widerspruchsfreiheit der gespeicherten Daten untereinander und in Bezug auf die Realität.

Konsistenz, physische

consistency, physical

Widerspruchsfreiheit der gespeicherten Daten in Bezug auf physisch richtige Speicherung sowie vollständige und richtige *Zugriffspfade* und Beschreibungsinformationen.

Konsistenz, Speicherkonsistenz

consistency, storage

siehe *physische Konsistenz*

Konsistenzfehler

consistency error

Eine Verletzung der *physischen Konsistenz* der gespeicherten Daten.

Konsistenzpunkt

consistency point

(Zeit-)Punkt, an dem die *Datenbank* konsistent ist, d.h. alle ändernden Transaktionen sind beendet und ihre Änderungen wurden im Datenbestand durchgeführt.

Konsistenzpunkt, festgeschriebener

checkpoint

Konsistenzpunkt, bei dem die ALOG-Datei gewechselt wurde und auf den jederzeit mit Hilfe des Dienstprogramms BMEND nachgefahren werden kann

Kopie

copy

siehe *Datenbankkopie*

Kopie aktualisieren

database copy update

Datenbankkopie durch Einspielen der *After-Images* auf einen festgeschriebenen *Konsistenzpunkt* vorsetzen.

L**Ladeparameter (DBH)**

load parameters (DBH)

Parameter, die der *DBH* beim Starten der *Session* anfordert. Die Parameter definieren die wesentlichen Merkmale einer Session.

Linked-in-Control-System

linked-in control system

Komponente von UDS/SQL bei *linked-in DBH*, die Steuerungsaufgaben übernimmt (entspricht dem *Subcontrol-System* bei *independent DBH*).

linked-in DBH

linked-in DBH

Modul, das in das jeweilige DB-*Anwenderprogramm* eingebunden oder nachgeladen wird und die Zugriffe auf eine *Datenbank (Mono-DB-Betrieb)* oder auf mehrere Datenbanken gleichzeitig (*Multi-DB-Betrieb*) steuert.

Liste

list

Tabelle, die die *Membersätze* einer *Set-Occurrence* enthält. Dient zum *sequentiellen* und *direkten Zugriff* auf die Membersätze.

Bei einer verteilbaren Liste können die Datenseiten, die die Membersätze enthalten (Stufe-0-Seiten), über mehrere Realms verteilt sein. Die Seiten, die die übergeordneten Tabellenstufen der verteilbaren Liste enthalten, liegen alle in einem Realm (Tabellenrealm einer verteilbaren Liste).

Logging

logging

Protokollierung über alle Änderungen in der *Datenbank*.

logische Verbindung

logical connection

Zuordnung zweier *Kommunikationspartner*, die es ihnen ermöglicht, Daten auszutauschen. *DCAM-Anwendungen* kommunizieren über logische Verbindungen.

lokale Datenbank

local database

Datenbank einer *lokalen Konfiguration*.

lokale Konfiguration

local configuration

Die *Konfiguration*, die dem *Anwenderprogramm* vor seinem Aufruf mit */SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname* zugewiesen wurde.

Mit der lokalen Konfiguration verkehrt das Anwenderprogramm über den *Communication Pool*. Die lokale Konfiguration liegt immer im Verarbeitungsrechner des Anwenderprogramms.

lokale Transaktion

local transaction

Transaktion, die nur auf die *lokale Konfiguration* zugreift.

lokale Verteiltabelle

local distribution table

Für einen *DBH* ist die *Verteiltabelle* lokal, die in seinem *Distribution Pool* liegt.

lokaler Verarbeitungsrechner

local host

Verarbeitungsrechner, in dem das *Anwenderprogramm* liegt.

lokales Anwenderprogramm

local application program

Ein *Anwenderprogramm* ist bezüglich einer *Konfiguration* lokal, wenn es über `/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname` an sie angeschlossen wurde.

M**Mainreference**

main reference

Die Mainreference dient im *DBH* der Verwaltung der zur Bearbeitung der Aufträge einer Transaktion erforderlichen Ressourcen, einschließlich solcher für die Übertragung der Aufträge vom Anwenderprogramm zum *DBH* und zurück.

Mainrefnummer

mainref number

Nummer, die bei *READY* der *Transaktion* zugewiesen wird. Diese Nummer ist nur zu einem Zeitpunkt eindeutig, nach Ende der Transaktion wird sie wieder einer anderen Transaktion zugewiesen.

Maske

pattern

Bei der Definition von *Feldern* eine symbolische Darstellung aller möglichen Feldinhalte.

Maskenzeichenkette

pattern string

Zeichenfolge, die eine *Maske* definiert.

Mastertask (MT)

master task

Task des *independent DBH*, in der das Modul *UDSSQL* abläuft.
Steuert das Einleiten und Beenden einer *Session* und kommuniziert direkt oder über die *Administratortask* mit dem *Datenbankadministrator*.

mehrstufige Tabelle

multi-level table

SEARCH-KEY-Tabelle, die für jeden *Satz* der zugehörigen *Satzart* bzw. für jeden *Membersatz* der zugehörigen *Set-Occurrence* eine Zeile enthält, die aus dem Schlüsselwert des Satzes und aus dem Zeiger zum Satz besteht. Wird auch als *Indextabelle* bezeichnet.

Member

member

siehe *Membersatz* bzw. *Membersatzart*

Member, AUTOMATIC

member, AUTOMATIC

Ein *Satz* wird beim Speichern eingehängt.

Member, MANDATORY

member, MANDATORY

Ein *Satz* kann nicht ausgehängt werden.

Member, MANUAL

member, MANUAL

Der *Satz* wird beim Speichern nicht automatisch eingehängt.

Member, OPTIONAL

member, OPTIONAL

Der *Satz* kann ausgehängt werden.

Membersatz

member record

Untergeordneter *Satz* in einer *Set-Occurrence*.

Membersatzart

member record type

Untergeordnete *Satzart* in einem *Set*.

Mono-DB-Betrieb

mono-DB operation

Der *DBH* arbeitet mit nur einer *Datenbank* einer *Konfiguration*.

Mono-DB-Konfiguration

mono-DB configuration

Nur eine *Datenbank* ist an einer *Session* beteiligt.

Multi-DB-Betrieb

multi-DB operation

Der *DBH* arbeitet mit mehreren *Datenbanken* einer *Konfiguration*.

Multi-DB-Konfiguration

multi-DB configuration

Mehrere *Datenbanken* sind an einer *Session* beteiligt.

Multi-DB-Programm

multi-DB program

Anwenderprogramm, das auf mehrere *Datenbanken* zugreift. Die *Datenbanken* können zu einer *Mono-* oder *Multi-DB-Konfiguration* oder zu mehreren *Mono-* oder *Multi-DB-Konfigurationen* gehören.

Multithreading-Verfahren

multithreading

Verfahren, durch das der *DBH* die Zentraleinheit (CPU) so intensiv wie möglich nutzen kann. Im Multithreading-Verfahren bearbeitet der DBH parallel mehrere Aufträge unter Verwendung sogenannter Threads. In jedem Thread sind Informationen über den gegenwärtigen Zustand eines bestimmten Auftrags hinterlegt. Muss ein Auftrag auf den Abschluss eines Eingabe/Ausgabe-Vorgangs warten, nutzt der DBH die CPU für die Verarbeitung eines anderen Auftrags.

N**Netz**

network

Alle über TRANSDATA gekoppelten Rechner.

netzweit eindeutig

unique throughout the network

In allen zu einem *Netz* gehörenden Rechnern eindeutig.

O**offene Transaktion**

open transaction

Eine nicht mit FINISH oder mit FINISH WITH CANCEL bzw. COMMIT oder ROLLBACK abgeschlossene *Transaktion*.

OLTP

OLTP

(Online Transaction Processing) Bei einer OLTP-Anwendung greift eine sehr große Anzahl von Benutzern auf die gleichen Programme und Daten zu. Dies geschieht in der Regel unter der Steuerung eines Transaktionsmonitors (TP-Monitor)

Online-DBTT-Erweiterung

online DBTT extension

Erweiterung der Anzahl der möglichen Sätze einer Satzart im laufenden Datenbankbetrieb. Für die Administration der Online-Erweiterbarkeit von DBTTs stehen die DAL-Kommandos ACT DBTT-INCR, DEACT DBTT-INCR, DISPLAY DBTT-INCR und EXTEND DBTT zur Verfügung.
Siehe auch *automatische DBTT-Erweiterung*.

Online-Realm-Erweiterung

online realm extension

Erweiterung von *Benutzerrealms* und *DBDIR* im laufenden Datenbankbetrieb. Für die Administration der Online-Erweiterbarkeit von Realms stehen die DAL-Kommandos ACT INCR, DEACT INCR, DISPLAY INCR, EXTEND REALM und REACT INCR zur Verfügung.
Siehe auch *automatische Realm-Erweiterung*.

Online-Sicherung

online backup

Wenn AFIM-Logging eingeschaltet ist, kann eine Sicherung der *Datenbank* im laufenden Betrieb erstellt werden. Die Online-Sicherungsfähigkeit einer Datenbank wird mit dem Dienstprogramm BMEND festgelegt.

Operatortask (OT)

operator task

siehe *Mastertask*

openUTM

openUTM

(universal transaction monitor) Universeller Transaktionsmonitor. Er ermöglicht die einfache Erstellung und den Betrieb von Transaktionsanwendungen.

Originaldatenbank

original database

Der Begriff Originaldatenbank bezieht sich lediglich auf die Namensgebung der Datenbankdateien (*dbname.dbdatei*), nicht auf den inhaltlichen Stand der Datenbank (siehe auch *Schattendatenbank*).

Owner

owner

siehe *Ownersatz* bzw. *Ownersatzart*

Ownersatz

owner record

Übergeordneter *Satz* in einer *Set-Occurrence*.

Ownersatzart

owner record type

Übergeordnete *Satzart* in einem *Set*.

P**PETA**

PETA

(Preliminary End of Transaction) Anweisung bei UDS-D und openUTM-D, die ein vorläufiges Transaktionsende herbeiführt.

Die PETA-Anweisung gehört zur ersten Phase des *Zwei-Phasen-Ende-Protokolls*, das eine *verteilte Transaktion* beendet.

Die Anweisung PETA speichert ausfallsicher in der *RLOG-Datei* des lokalen *DBH*.

- alle geänderten *Seiten*
- die Rücksetz- und Sperrinformationen
- die Namen aller beteiligten *Konfigurationen*

Diese Informationen werden bei einem eventuellen *Warmstart* benötigt.

POINTER-ARRAY

pointer array

siehe *Adressliste*

PPP

PPP

siehe *Probable Position Pointer (PPP)*.

Prepared to Commit (PTC)

prepared to commit (PTC)

Teil des *Zwei-Phasen-Ende-Protokolls*.

Zustand einer *Teiltransaktion* nach Durchführen der *PETA*-Anweisung und vor Erhalt der Nachricht, ob die gesamte *Transaktion* mit FINISH oder mit FINISH WITH CANCEL beendet wird.

primäre Teiltransaktion (PTT)

primary subtransaction

Teiltransaktion, die in der *lokalen Konfiguration* abläuft.

Die erste *READY*-Anweisung einer *Transaktion* auf eine *lokale Datenbank* eröffnet die primäre Teiltransaktion.

Falls die erste *READY*-Anweisung eine *entfernte Datenbank* anspricht, erzeugt UDS-D eine sogenannte *Dummy-Teiltransaktion* als primäre Teiltransaktion.

Primärschlüssel (DDL)

primary key (DDL)

Der mittels "LOCATION MODE IS CALC" definierte *Schlüssel* einer *Satzart* bzw. der mittels "ORDER IS SORTED [INDEXED]" definierte ordnungsbestimmende *Schlüssel* einer Set-Occurrence. Dient außerdem zum *Direktzugriff* auf einen *Satz* oder eine Menge von Sätzen mit gleichen Schlüsselwerten oder innerhalb eines Suchintervalls.

Primärschlüssel (SQL)

primary key (SQL)

Im weiteren Sinne (SQL) ein *Satzelement*, das einen Datensatz eindeutig identifiziert.

In UDS-SQL der im BPSQLSIA-Protokoll als "PRIMARY KEY" ausgegebene Database Key eines Ownersatzes (siehe auch *Fremdschlüssel*).

Ein einen Datensatz eindeutig identifizierendes *Satzelement* ist im BPSQLSIA-Protokoll als "UNIQUE" ausgewiesen, wenn es sich nicht um den obigen "PRIMARY KEY" handelt.

PRIVACY-AND-IQF-Schema

PRIVACY-AND-IQF SCHEMA

UDS/SQL-internes *Schema* für den Zugriffsschutz.

PRIVACY-AND-IQF-Subschema

PRIVACY-AND-IQF SUBSCHEMA

UDS/SQL-internes *Subschema* für den Zugriffsschutz.

Probable Position Pointer (PPP)

probable position pointer (PPP)

Wahrscheinliche Adresse einer *Seite*, bestehend aus *Realmnummer* und *Seitennummer*. Bei einer Lageänderung von Daten aktualisiert UDS/SQL die zugehörigen Probable Position Pointer (PPP) nicht in jedem Fall.

Prüfsätze

check records

Informationselemente zum Prüfen der Datenbank. Sie haben eine variable Länge von 20 bis 271 byte.

Pubset-Deklaration

pubset declaration

siehe *UDS/SQL-Pubset-Deklaration*

Pubset-Deklarations-Jobvariable

pubset declaration job variable

Jobvariable, in der eine *UDS/SQL-Pubset-Deklaration* vereinbart wird.

P1-Eventing

P1 eventing

Verständigung der Tasks untereinander.

Q

Quellprogramm

source program

In einer Programmiersprache formuliertes, noch nicht in die Maschinensprache übersetztes Programm.

R

READY

READY

Beginn einer *Transaktion* oder *Verarbeitungskette* bei *COBOL-DML*-Programmen.

READYC

READYC

Beginn einer *Transaktion* oder *Verarbeitungskette* bei *CALL-DML*-Programmen.

Realm

realm

Benennbare physische Untereinheit der *Datenbank*. Der Realm entspricht einer Datei. Außer den *Benutzerrealms* für die Daten gibt es die Realms *DBDIR* und *DBCUM*, die UDS/SQL selbst beansprucht.

Realm-Konfiguration

realm configuration

Die *Realms* einer *Datenbank*, die an einer *Session* beteiligt sind.

Realm-Kopie

realm copy

siehe *Datenbankkopie*

Realm-Nummer

realm reference number

Realms einer *Datenbank* werden, bei 1 beginnend, aufsteigend und lückenlos nummeriert. Die Realm-Nummer (Area-Reference) ist Bestandteil der *Seitenadresse*.

RECORD AREA

RECORD AREA

siehe *Satzbereich*

REC-REF

REC-REF

(Record Reference)
 siehe *Satzartennummer*

referentielle Integrität

referential integrity

Integrität der Beziehungen zwischen Tabellen (UDS/SQL-*Satzarten*).

Rekonfiguration

reconfiguration

Neugruppierung von *Datenbanken* in einer *DB-Konfiguration* nach einem *Session-Abbruch*.
Voraussetzung für eine Rekonfiguration ist, dass die *SLF* gelöscht oder inhaltlich entwertet wird.

Returncode

return code

Interner Code eines aufgerufenen Programms an das aufrufende Programm.
Returncode != 0: Fehler aufgetreten.

RLOG-Datei

RLOG file

Datei zur Ablaufsicherung. In die RLOG-Datei schreibt der *DBH* während der *Session* sowohl Daten vor ihrer Änderung (*Before-Images*) als auch Daten nach ihrer Änderung (*After-Images*). Mit Hilfe der *RLOG-Datei* kann der DBH Änderungen nicht abgeschlossener *Transaktionen* zurücksetzen.
Es gibt eine RLOG- Datei pro *Konfiguration*. Die RLOG-Datei besteht aus zwei physischen Dateien.

Rollback

rollback

Rückgängigmachen aller Änderungen einer *Transaktion*.

RSQ

RSQ

siehe *Satzfolgenummer*.

RUNUNIT-ID

RUNUNIT-ID

siehe *Transaktionskennung*

S**Satz**

record

Einzelne Ausprägung einer *Satzart*. Ein Satz besteht aus je einem Feldinhalt aller am Aufbau der Satzart beteiligten *Felder* und ist die kleinste Dateneinheit, die UDS/SQL über einen eindeutigen Identifizierer, den *Database Key*, verwaltet.

Satzadresse

record address

Adresse der *Seite*, in der sich der *Satz* befindet. Siehe *Seitenadresse*.

Satzart

record type

Benennbare Zusammenfassung von *Satzelementen*.

Satzart, lineare

record type, linear

Satzart, die weder *Owner* noch *Member* eines *Set* ist (entspricht Satzarten einer konventionellen Datei).

Satzartnummer

record reference number

Satzarten werden, bei 1 beginnend, aufsteigend und lückenlos numeriert. Die Satzartnummer ist Bestandteil des *Database Key*.

Satzbereich

record area

Vom Benutzer adressierbarer Bereich der *USER-WORK-AREA (UWA)*.

Der Satzbereich enthält die *Satzarten* und die implizit definierten Felder (IMPLICITLY-DEFINED-DATA-NAMES) der Datenbank wie z.B. die AREA-ID-Felder der WITHIN-Klauseln des Schemas. Die Länge des Satzbereichs ist wesentlich durch die in ihm definierten Satzarten bestimmt.

Satzelement

record element

Feld, *Vektor* oder *Datengruppe*.

Satzfolgenummer

record sequence number

Der Datenbankprogrammierer kann die Satzfolgenummer vergeben oder UDS/SQL numeriert die *Sätze* einer *Satzart* selbst, bei 1 beginnend, aufsteigend und lückenlos in der Reihenfolge wie die Sätze gespeichert werden. Die Satzfolgenummer ist Bestandteil des *Database Key*.

Satzhierarchie

record hierarchy

Owner-/Memberbeziehung zwischen *Satzarten*.
Ownersatzart ist übergeordnet
Membersatzart ist untergeordnet.

Satz-SEARCH-Key-Tabelle

record SEARCH KEY table

SEARCH-Key-Tabelle für die Auswahl eines *Satzes* aus einer *Satzart*.

SCD

SCD

(Set Connection Data) Verknüpfungsinformation für die *Sätze* einer *Set-Occurrence*.

Schattendatenbank

backup database

Sicherung sämtlicher Dateien einer *Datenbank* jeweils unter „*dbname.dbdatei.copypname*“.
Die Schattendatenbank kann zu einem beliebigen Zeitpunkt erstellt werden und ist parallel zur Originaldatenbank im Benutzungsmodus RETRIEVAL ablauffähig.
Außerdem können die bereits abgeschlossenen *ALOG-Dateien* auf ihr parallel zur UDS/SQL-*Session* mit BMEND nachgefahren werden.

Schema

schema

Formalisierte Beschreibung der in der *Datenbank* zugelassenen Datenstrukturen. Ein UDS/SQL-Schema wird mit der *Schema-DDL* beschrieben.

Schema-DDL

Schema DDL

Formale Sprache zur Beschreibung eines *Schemas*.

Schlüssel

key

Feld, das der Datenbankprogrammierer für *Direktzugriff* auf *Sätze* benutzt und für das UDS/SQL entsprechend den Angaben im *Schema* einen optimierten *Zugriffspfad* anlegt.

Schlüssel, zusammengesetzter

key, compound

Schlüssel, der aus mehreren *Schlüsselfeldern* besteht.

Schlüsselfeld

key item

Feld, das durch Angaben im *Schema* zum *Schlüssel* erklärt wird.

Schlüsselnummer

key reference number

Schlüssel werden, bei 1 beginnend, aufsteigend und lückenlos numeriert.

Schnittstelle

interface

In der Software: Speicherbereich, den mehrere Programme zum Austausch von Daten untereinander verwenden.

SEARCH-Key

SEARCH KEY

Sekundärschlüssel. Zugriffspfade über Sekundärschlüssel realisiert UDS/SQL über *Hashverfahren* und *mehrstufige Tabellen*.

SEARCH-Key-Tabelle

SEARCH KEY table

Mehrstufige Tabelle, die UDS/SQL als *Zugriffspfad* über einen *Sekundärschlüssel* benutzt.

Seite

page

Physische Untereinheit von *Realms*. Seiten identifiziert UDS/SQL über eindeutige Schlüssel (*Act-Key*). Die Länge einer Seite kann wahlweise 2048 byte, 4000 byte oder 8096 byte betragen. Innerhalb derselben Datenbank müssen alle Seiten gleich lang sein. Seiten der Länge 4000 byte oder 8096 byte sind in einen *Seitencontainer* eingebettet.

Seitenadresse

page address

Bei der Seitenadresse unterscheidet man die aktuelle Adresse einer *Seite*, den *Act-Key*, und die wahrscheinliche Adresse einer Seite, den *Probable Position Pointer (PPP)*.

Seitencontainer

page container

Seiten der Länge 4000 byte oder 8096 byte sind jeweils in einen sogenannten Seitencontainer eingebettet. Der Seitencontainer besteht aus einem 64 byte langen Header, der vor der Seite liegt, und einem 32 byte langen Trailer im Anschluss an die Seite.

Seitenindex-Eintrag

page index entry

Verweist auf die Position eines *Satzes* innerhalb einer *Seite*.

Seitenkopf

page header (page info)

Die ersten 20 byte einer *Seite* (mit Ausnahme der *FPA-Basis-Seiten* und *DBTT-Seiten* der Länge 2048 byte). Sie enthalten

- den *Act-Key* der *Seite* selbst
- die Anzahl der *Seitenindex-Einträge*

- die Länge und Position der in dieser Seite noch freien Bytes
- den Seitentyp (*ACT-Key-0-Seite*, *FPA-Seite*, *DBTT-Seite*, *DBTT-Ankerseite*, allgemeine Datenseite oder *CALC-Seite*)

Seitennummer

page number

In jedem *Realm* sind die *Seiten*, bei 0 beginnend, aufsteigend und lückenlos numeriert. Die Seitennummer ist Bestandteil der *Seitenadresse*.

Seitennummer = PAM-Seitennummer-1 bei Datenbanken mit einer Seitenlänge von 2048 byte

Seitennummer = (PAM-Seitennummer-1) / 2 bei Datenbanken mit einer Seitenlänge von 4000 byte

Seitennummer = (PAM-Seitennummer-1) / 4 bei Datenbanken mit einer Seitenlänge von 8096 byte.

sekundäre Teiltransaktionen

secondary subtransactions

Teiltransaktionen, die *entfernte Konfigurationen* ansprechen.

Sekundärschlüssel

secondary key

Jeder *Schlüssel*, der nicht *Primärschlüssel* ist; dient zum *Direktzugriff* auf einen *Satz* oder eine Menge von Sätzen mit gleichen Schlüsselwerten oder innerhalb eines Suchintervalls.

sequentieller Zugriff

sequential access

Zugriff auf einen *Satz* aufgrund seiner Position innerhalb einer vorgegebenen Satzreihenfolge.

Servertask (ST)

server task

Task des *independent DBH*, in der das Modul *UDSSUB* abläuft. Die Servertask bearbeitet die Anforderungen der *DB-Anwenderprogramme*.

Session

session

Zeitraum zwischen dem Starten und dem normalen Beenden des *DBH* (*independent/linked-in DBH*), in dem mit den *Datenbanken* der *Konfiguration* gearbeitet werden kann. Im allgemeinen Fall besteht eine Session aus einer Folge von *Session-Abschnitten* und *Session-Unterbrechungen*.

Session-Abbruch

session abort

Liegt vor, wenn der *DBH* nach erfolgreichem *Session-Beginn* abnormal beendet wird.

Ursachen für einen Session-Abbruch können sein: Stromausfall, Rechnerausfall, BS2000-Störung, DBH-Fehler, %TERM.

Session-Abschnitt

session section

Beginnt mit dem Starten eines *DBH* entweder bei *Session-Beginn* oder bei *Session-Wiederanlauf* und endet mit dem normalen *Session-Ende* oder mit *Session-Abbruch*.

Session-Abschnittsnummer

session section number

Nummer, die einen Session-Abschnitt eindeutig identifiziert.

Session-Beginn

session start

Liegt vor, wenn ein *DBH* unter einem *Konfigurationsnamen* gestartet wird, für den noch keine *Session-Log-File (SLF)* mit gültigem Inhalt existiert.

Session-Ende

session end

Wird erreicht durch

- *DAL* bei *independent DBH*,
- *TERM* in *DML-Anwenderprogrammen* bei *linked-in DBH*,
- die *DBH-Fehlerbehandlung*.

Während einer *Session-Unterbrechung* kann das Session-Ende auch erreicht werden, indem der Anwender die *SLF* inhaltlich entwertet. Bei inkonsistenten *Datenbanken* kann die *Konsistenz* auch ohne *SLF* mit *Warmstart* wiederhergestellt werden.

Session-Jobvariable

session job variable

Jobvariable, in der UDS/SQL Informationen über eine Session hinterlegt.

Session-Log-File (SLF)

Session Log File (SLF)

Datei, die einer *Session* fest zugeordnet ist und die der *DBH* bei einem eventuellen *Session-Wiederanlauf* benötigt. Sie enthält Informationen über die aktuelle *DB-Konfiguration*, die Menge der aktuellen Dateikennwörter und über die aktuellen Werte der *DBH-Ladeparameter*.

Session-Unterbrechung

session interrupt

Zeitraum zwischen einem *Session-Abbruch* und dem zugehörigen *Session-Wiederanlauf*.

Session-Wiederanlauf

session restart

Start des *DBH* nach einer abgebrochenen *Session* unter gleichem *Konfigurationsnamen* und in der gleichen *Konfigurationskennung*. Mit Hilfe der *SLF* werden die *DBH-Ladeparameter* und die aktuellen Datei-Kennwörter wiederhergestellt, die bei *Session-Abbruch* vorlagen und die *Datenbanken* der damaligen *Konfiguration* werden ggf. mit *Warmstart* angeschlossen.

Set

set

Benennbare Beziehung zwischen zwei *Satzarten*.

Set, dynamischer

set, dynamic

siehe *dynamischer Set*

Set, impliziter

set, implicit

siehe *impliziter Set*

Set, singulärer

set, singular

siehe *SYSTEM-Set*

Set, Standard-

set, standard

siehe *Standard-Set*

Setnummer

set reference number

Sets werden, bei 1 beginnend, aufsteigend und lückenlos numeriert.

Set-Occurrence

set occurrence

Einzelne Ausprägung eines *Set*. Eine Set-Occurrence besteht aus genau einem *Ownersatz* und beliebig vielen ihm untergeordneten *Membersätzen*.

Set-SEARCH-Key-Tabelle

set SEARCH KEY table

SEARCH-Key-Tabelle für die Auswahl eines *Membersatzes* aus einer *Set-Occurrence*.

Shared User Buffer Pool

Shared User buffer pool

Gemeinsamer Puffer mehrerer Datenbanken, der zusätzlich zu den *System Buffer Pools* ausschließlich für die Pufferung von *Seiten* der ihm zugewiesenen *Datenbanken* verwendet wird.

SF-Pubset

SF pubset

siehe *Single-Feature-Pubset*

SIA

SIA

(Schema Information Area) Sie enthält die vollständige Datenbankbeschreibung. Der *DBH* lädt die SIA zum Arbeiten generell in den Hauptspeicher.

SIB

SIB

(SQL Interface Block) Schnittstelle zwischen UDS/SQL und SQL-Anwenderprogramm(en); enthält die SQL-Anweisung mit eventuell vorhandenen Parametern und das Anweisungsergebnis.

Single-Feature-Pubset

single feature pubset

Ein Single-Feature-Pubset (SF-Pubset) besteht aus einer oder mehreren homogenen Platten, die in den wesentlichen Eigenschaften (Plattenformat, Allokierungseinheit) übereinstimmen müssen.

SLF

SLF

siehe *Session-Log-File (SLF)*.

SM-Pubset

SM pubset

siehe *System-Managed-Pubset*

Snap-Paar, Snap-Pubset, Snap-Session, Snap-Unit

snap pair, snap pubset, snap session, snap unit

Eine Snap-Unit ist die Kopie einer (Original-)Unit (logische Platte im BS2000) zu einem bestimmten Zeitpunkt („Point-in-Time-Kopie“). Die Komponente TimeFinder/Snap erstellt diese Kopie als „Snapshot“ nach der „Copy-On-First-Write-Strategie“: Nur wenn Daten geändert werden, werden zuvor die jeweiligen Original-Daten in einen zentralen Speicherbereich (Save-Pool) des Symmetrix-Systems geschrieben. Die Snap-Unit enthält die Verweise (Track-Pointer) auf die Original-Daten. Bei unveränderten Daten zielen die Verweise auf die Unit, bei veränderten auf den Save-Pool. Nach der Aktivierung sind Unit und Snap-Unit voneinander getrennt, Anwendungen können auf beide zugreifen.

Unit und Snap-Unit bilden zusammen ein Snap-Paar. TimeFinder/Snap verwaltet es in einer sogenannten Snap-Session.

Wenn es zu allen Units eines Pubsets Snap-Units gibt, so bilden diese Snap-Units zusammen das Snap-Pubset.

Details zu diesem Thema finden Sie im Handbuch „[Einführung in die Systembetreuung](#)“.

Sort-Key-Tabelle

sort key table

Zusätzlicher *Direktzugriffspfad* mittels des *Primärschlüssels* auf Setebene auf die *Membersätze* einer *Set-Occurrence* bei "MODE IS CHAIN" und "ORDER IS SORTED INDEXED".

spanned record

spanned record

Satz, der länger ist als eine *Seite*. Spanned records gibt es **nur UDS/SQL-intern**. Benutzersatzarten dürfen generell nicht länger sein als

- 2020 byte bei 2048 byte Seitenlänge
- 3968 byte bei 4000 byte Seitenlänge
- 8064 byte bei 8096 byte Seitenlänge

SQL

SQL

(Structured Query Language) SQL ist eine relationale Datenbanksprache, die von der ISO (International Organization for Standardization) standardisiert worden ist.

SQL-DML

SQL-DML

Data Manipulation Language von *SQL*, für die Abfrage und Änderung von Daten.

SQL-Transaktion

SQL transaction

Zusammengehörige Folge von *SQL*-Anweisungen, die UDS/SQL entweder ganz oder gar nicht bearbeitet, um die *Datenbank(en)* von einem konsistenten Zustand in einen anderen konsistenten Zustand zu überführen.

SQL-Vorgang

SQL conversation

siehe *Vorgang*

SSIA

SSIA

(Subschema Information Area) enthält alle Subschema-abhängigen Informationen, die der *Database Handler* benötigt, um für den Anwender auf die *Datenbank* innerhalb der Möglichkeiten des aufgerufenen *Subschemas* zuzugreifen.

Der *DBH* lädt die SSIA, sobald sie bei einem *READY* angesprochen wird, in den Hauptspeicher.

SSIA-RECORD

SSIA-RECORD

UDS/SQL-interne *Satzart*, die im *Database Directory (DBDIR)* liegt. *Sätze* dieser Satzart sind u.a. die Schema Information Area (*SIA*) und die Subschema Information Areas (*SSIA*).

SSITAB-Modul

SSITAB module

Vom Dienstprogramm BCALLSI erzeugtes Modul. Es stellt die Subschema-Informationen für *CALL-DML*-Programme bereit.

SSL

SSL

(Storage Structure Language) Formale Sprache zur Beschreibung der Speicherstruktur.

Standard-Set

standard set

Set, der kein *dynamischer* oder *impliziter Set* oder *SYSTEM-Set* ist.

Statuscode

status code

Nummer, die im zweiten Teil des Feldes *DATABASE-STATUS* hinterlegt wird, und die darüber informiert, welcher Sonderzustand aufgetreten ist.

String

string

Eine Reihe aufeinanderfolgender alphanumerischer Zeichen.

Subcontrol-System

subcontrol system

Komponente des *independent DBH*, die Steuerungsaufgaben übernimmt.

Subschema

subschema

Für eine bestimmte *Anwendung* erforderlicher Teil eines *Schemas*, der für eine Anwendung in begrenztem Umfang neu strukturiert werden kann. Das Subschema wird mit der *Subschema-DDL* beschrieben.

Subschema-DDL

Subschema DDL

Formale Sprache zur Beschreibung eines *Subschemas*.

Subschemamodul

subschema module

Modul, das beim Übersetzen eines *COBOL-DML*-Programms aus der Übersetzung des *Subschemas* entsteht. Es muss in das *Anwenderprogramm* eingebunden werden und enthält die *UWA* sowie die *RECORD AREA*, die gleichzeitig Teil des Base Interface Block (*BIB*) ist. Der Name des Subschemamoduls sind die ersten acht Zeichen des Subschemanamens.

Subschemasatz

subschema record

Satz laut *Subschema-DDL*.

SUB-SCHEMA SECTION

SUB-SCHEMA SECTION

Bei einem COBOL-Programm mit *DML*-Anweisungen: Abschnitt in der DATA DIVISION zur Angabe des Schemanamens und des Subschemanamens.

Subtask (ST)

subtask

siehe *Servertask*.

System Buffer Pools

system buffer pools

Ein-/Ausgabe-Puffer für Datenbankseiten (siehe *Seite*). Sie liegen im *Common Pool (independent DBH)* bzw. *DBH-Arbeitsbereich (linked-in DBH)*. Ihre Größe bestimmen die *DBH-Ladeparameter* 2KB-BUFFER-SIZE, 4KB-BUFFER-SIZE bzw. 8KB-BUFFER-SIZE.

Systembereich

system area

Realm, der nur von UDS/SQL benötigt wird. Zu den Systembereichen einer Datenbank zählt man:

- das *Database Directory (DBDIR)*,
- den *Database Compiler Realm (DBCOM)*,
- das *COBOL Subschema Directory (COSSD)*

Systembreak-Informationen

system break information

Kennzeichen, ob die *Datenbank* konsistente oder inkonsistente Information enthält.

System-Managed-Pubset

system managed pubset

Ein System-Managed-Pubset besteht aus einem oder mehreren Volume-Sets, die wie bei einem *SF-Pubset* eine Zusammenfassung von mehreren homogenen Platten sind; die Homogenität bezieht sich auch hier auf bestimmte physikalische Eigenschaften wie z.B. Plattenformat und Allokierungseinheit.

SYSTEM-Record

SYSTEM record

siehe *Ankersatz*

SYSTEM-Set

SYSTEM set

Set, dessen *Ownersatzart* die symbolische *Satzart* SYSTEM ist.

T**Tabelle, mehrstufige**

table, multi-level

siehe *mehrstufige Tabelle***Tabelle (SQL)**

table (SQL)

Eine Tabelle im *SQL*-Sinn entspricht einer UDS/SQL-*Satzart*.**Tabellenkopf**

table header

Enthält allgemeine Informationen über eine Tabelle bzw. eine *Tabellenseite*.

- die Angabe über den Tabellentyp und die Stufennummer der Tabellenseite,
- die Anzahl der reservierten und der aktuellen Einträge in dieser Tabellenseite,
- die Verkettung mit weiteren Tabellenseiten der gleichen Stufe,
- den Verweis auf die zugehörige Tabellenseite der nächsthöheren Stufe und
- den Verweis auf die Seite mit der letzten Tabelle der Grundstufe (nur bei der Tabelle der höchsten Stufe).

Tabellenseite

table page

Seite, die eine Tabelle oder einen Tabellenteil enthält. Handelt es sich um eine *Tabelle*, die sich nicht über mehrere Seiten erstreckt, oder um die höchste Stufe einer mehrstufigen *Tabelle*, so ist mit „Tabellenseite“ nur das entsprechende Objekt gemeint, nicht die ganze *Seite*.

TANGRAM

TANGRAM

(Task and Group Affinity Management) Subsystem des BS2000; dieses Subsystem plant für Taskgruppen, die bei Multitask-Anwendungen auf größere gemeinsame Datenmengen zugreifen, die Zuordnung zu den Prozessoren.

Task Attribut TP

task attribute TP

Im BS2000 gibt es 4 Task Attribute: SYS, TP, DIALOG und BATCH.

Den Task Attributen sind jeweils spezielle, für das Task-Scheduling wichtige Ablaufparameter zugeordnet.

TP zeichnet sich gegenüber den anderen Task Attributen durch eine, speziell auf die Bedürfnisse des Teilhaberbetriebs optimierte Hauptspeicher-Verwaltung aus.

Taskdeadlock

task deadlock

siehe *Deadlock*

Taskkommunikation

task communication

Verständigung der *DBH*-Module untereinander. Siehe auch *Common Pool*.

Taskpriorität

task priority

Im BS2000 kann die Priorität für eine Task festgelegt werden. Diese Priorität wird bei der Initiierung und Aktivierung der Task berücksichtigt.

Es gibt variable und feste Prioritäten. Variable Prioritäten passen sich an, feste verändern sich nicht. (UDS/SQL-Servertasks sollen mit einer festen Priorität gestartet werden, um eine gleichbleibende Performance zu erreichen).

TCUA

TCUA

(Transaction Currency Area) enthält die Currency-Informationen.

Teiltransaktion

subtransaction

In einer verteilten *Transaktion* bilden alle *Verarbeitungsketten*, die *Datenbanken einer Konfiguration* ansprechen, eine Teiltransaktion.

Transaktion (TA)

transaction

Zusammengehörige Folge von *DML*-Anweisungen, die UDS/SQL entweder ganz oder gar nicht bearbeitet, um die *Datenbank(en)* von einem konsistenten Zustand in einen anderen konsistenten Zustand zu überführen.

Bei UDS-D:

Gesamtheit aller zu einem Zeitpunkt gestarteten *Teiltransaktionen*.

Transaktion normal beenden

transaction, committing a

Eine *Transaktion* mit FINISH beenden, d.h. alle Änderungen festschreiben, die auf den *Datenbanken* gemacht wurden.

Transaktion zurücksetzen

transaction, rolling back a

Eine *Transaktion* mit FINISH WITH CANCEL beenden, d.h. alle Änderungen rückgängig machen, die auf den *Datenbanken* gemacht wurden.

Transaktionskennung

transaction identification (TA-ID)

Vergibt der *DBH* zur Kennzeichnung einer *Transaktion*, kann mit dem *DAL*-Kommando DISPLAY erfragt werden.

Transfer Pool

transfer pool

UDS-D-spezifischer Speicherbereich, in dem der *UDSCT* die *BIBs* von *entfernten Anwenderprogrammen* empfängt.

U

UDSADM

UDSADM

Modul des *independent DBH*. Das Modul läuft in der *Administratortask* ab.

UDSHASH

UDSHASH

Vom Dienstprogramm BGSIA erzeugtes Modul mit den Namen aller *Hashroutinen*, die in der *Schema-DDL* definiert wurden.

UDSNET

UDSNET

Verteilkomponente in der *Anwendertask*.

UDSSQL

UDSSQL

Startmodul des *independent DBH*. Das Modul läuft in der *Mastertask* ab.

UDSSUB

UDSSUB

Startmodul des *independent DBH*. Das Modul läuft in der *Servertask* ab.

UDS-D-Task UDSCD

UDS-D task UDSCD

Task, die UDS/SQL für jede *Konfiguration* startet, damit sie an der verteilten Verarbeitung mit UDS-D teilnehmen kann.

UDS/SQL / openUTM-D-Konsistenz

UDS/SQL / openUTM-D consistency

Eine *Transaktion*, die sowohl *openUTM*-Daten als auch *UDS/SQL-Datenbanken* geändert hat, wird so beendet, dass entweder die *openUTM*-Daten und die *UDS/SQL-Datenbanken* geändert werden, oder keines von beiden.

UDS/SQL-Pubset-Deklaration

UDS/SQL pubset declaration

Vereinbarung in einer *Pubset-Deklarations-Jobvariable* zur Einschränkung der *UDS/SQL-Pubset-Umgebung*. Dadurch wird die Gefahr durch die Mehrdeutigkeit von Dateinamen verringert bzw. vermieden.

Überlaufseite

overflow page

Seite bei *Hashbereichen* und *Duplikat-Tabellen*, die diejenigen Daten aufnimmt, die nicht mehr in die Primärseite passen. Ihr Aufbau entspricht den Seiten des Hashbereichs bzw. der Duplikat-Tabelle.

Umstrukturierung

restructuring

Änderung von *Schema-DDL* oder *SSL* bei *Datenbanken*, in denen bereits Daten gespeichert sind.

USER-WORK-AREA (UWA)

USER-WORK-AREA (UWA)

Übergabebereich zur Kommunikation zwischen *Anwenderprogramm* und *DBH*.

UTM

UTM

siehe openUTM.

UWA

UWA

siehe *USER-WORK-AREA (UWA)*.

V**Vektor**

vector

Feld mit Wiederholungsfaktor. Der Wiederholungsfaktor muss größer als 1 sein.
Er gibt an, wieviel Duplikate des Feldes zu dem Vektor zusammengefasst werden.

Verarbeitungskette

processing chain

Folge von *DML*-Anweisungen an eine *Datenbank* innerhalb einer *Transaktion*.

Verbindungsmodul

connection module

Modul, das in jedes UDS/SQL-*Anwenderprogramm* eingebunden werden muss und die Verbindung zum *DBH* herstellt.

Versionsnummer, interne

version number, internal

siehe *interne Versionsnummer*

Verteiltabelle

distribution table

Tabelle, die UDS-D anhand der zugewiesenen Eingabedatei im *Distribution Pool* aufbaut. Mit Hilfe der Verteiltabelle entscheidet die Verteilkomponente in der *Anwendertask*, ob eine *Verarbeitungskette* lokal oder entfernt bearbeitet werden soll.

In der Verteiltabelle ist zugeordnet:

Subschema - Datenbank

Datenbank - Konfiguration

Konfiguration - Verarbeitungsrechner.

verteilte Datenbanken

distributed database

Ein logisch zusammengehörender Datenbestand, der auf mehrere UDS/SQL-Konfigurationen verteilt ist.

verteilte Transaktion

distributed transaction

Transaktion, die auf mindestens eine *entfernte Konfiguration* zugreift.

Eine Transaktion kann verteilt sein über:

- UDS-D,
- openUTM-D,
- UDS-D und openUTM-D.

Vorgang

conversation

In einer *Anwendung* mit *SQL* werden *SQL*-spezifische Verwaltungsdaten über Transaktionsgrenzen hinweg aufbewahrt. Eine solche Verwaltungseinheit wird als Vorgang bezeichnet.

W**Warmstart (einer DB)**

warm start

Ein Warmstart wird von UDS/SQL durchgeführt, wenn eine inkonsistente *Datenbank* an eine *Session* angeschlossen wird. Ein Warmstart umfasst das Nachfahren der Änderungen abgeschlossener *Transaktionen*, die noch nicht auf der Datenbank festgeschrieben waren, den *Rollback* aller auf der Datenbank offenen Transaktionen und das Konsistentmachen der Datenbank. Für einen Warmstart wird die zugehörige *RLOG-Datei* benötigt und die *DB-Status-Datei*.

Wiederanlauf (von BMEND)

restart of BMEND

Fortsetzung eines abgebrochenen BMEND-Laufs.

Wiederanlauf (einer Session)

restart of a session

siehe *Session-Wiederanlauf*

Wiederholungsgruppe

repeating group

Datengruppe mit Wiederholungsfaktor. Der Wiederholungsfaktor muss größer als 1 sein. Er gibt an, wieviele Duplikate der Datengruppe zu der Wiederholungsgruppe zusammengefasst werden.

Z**Zeitquittung**

time acknowledgment

Nachrichten, die die *UDS-D-Task* zum entfernten *Anwenderprogramm* sendet, um mitzuteilen, dass noch eine *DML*-Anweisung bearbeitet wird.

Zugriff, direkter

access, direct

siehe *direkter Zugriff*

Zugriff, konkurrierender

access, contending

siehe *konkurrierender Zugriff*

Zugriff, sequentieller

access, sequential

siehe *sequentieller Zugriff*

Zugriffsart

access type

Art und Weise des Zugriffs, zum Beispiel Lesen, Ändern usw.

Zugriffsberechtigte

authorized users

Festgelegte Benutzergruppen und deren Benutzer, die auf die *Datenbank* zugreifen dürfen.

Zugriffsberechtigung

access authorization

Recht einer definierten Benutzergruppe in definierter Weise auf die *Datenbank* zuzugreifen. Die Zugriffsberechtigung wird im laufenden Datenbankbetrieb mit dem Dienstprogramm ONLINE-PRIVACY bzw. im Offline-Modus mit dem Dienstprogramm BPRIVACY festgelegt.

Zugriffspfad

access path

Hilfsmittel, um eine bestimmte, durch eine Suchfrage qualifizierte Untermenge aller *Sätze* auffinden zu können, ohne die ganze *Datenbank* sequentiell absuchen zu müssen.

Zugriffsrechte

access rights

Zugriffsrechte werden durch das Dienstprogramm BPRIVACY festgelegt. Sie regeln den Zugriff auf die *Datenbank*.

Zustand PTC

PTC state

siehe *Prepared to Commit*

Zwei-Phasen-Ende-Protokoll

two-phase commit protocol

Verfahren, um eine *verteilte Transaktion*, die in mindestens einer *entfernten Konfiguration* geändert hat, so zu beenden, dass die *konfigurationsübergreifende Konsistenz* bzw. die UDS/SQL-/openUTM-D-Konsistenz gesichert ist.

Das Zwei-Phasen-Ende-Protokoll wird gesteuert:

- von der Verteilkomponente in der *Anwendertask*, wenn die *Transaktion* über UDS-D verteilt ist.
- von openUTM-D, wenn die Transaktion über openUTM-D bzw. über openUTM-D und über UDS-D verteilt ist.

17 Abkürzungen

ACS	Alias Catalog Service
Act-Key	Actual-Key
AFIM	After-Image
AP	Anwenderprogramm, Application Program
ASC	Ascending
BIB	Base Interface Block
BFIM	Before-Image
COBOL	Common Business Oriented Language
CODASYL	Conference on Data System Languages
CRA	Current Record of Area
CRR	Current Record of Record
CRS	Current Record of Set
CRU	Current Record of Rununit
COSSD	COBOL Subschema Directory
DAL	Database Administration Language
DB	Datenbank
DBCUM	Database Compiler Realm
DBDIR	Database Directory
DBH	Database Handler
DB-Key	Database Key
DBTT	Database Key Translation Table
DDL	Data Description Language
DESC	Descending
DML	Data Manipulation Language
DRV	Dual Recording by Volume
DSA	Database System Access
DSSM	Dynamische Verwaltung von Subsystemen
FC	Function Code
FPA	Free Place Administration
GS	Global Store (Globalspeicher)

HSMS	Hierarchisches Speicher Management System
ID	Identification (Kennung)
IMON	Installation Monitor
IQL	Interactive Query Language
IQS	Interactive Query System
KDBS	Kompatible Datenbank-Schnittstelle
KDCS	Kompatible Datenkommunikations-Schnittstelle
LM	Lock Manager
LMS	Library Maintenance System
MPVS	Multiple Public Volume Set
MR-NR	Mainref-Number
MT	Mastertask
OLTP	Online Transaction Processing
openUTM	Universeller Transaktionsmonitor
OT	Operatortask
PETA	Preliminary End of Transaction
PPP	Probable Position Pointer
PTC	Prepared to Commit
PTT	Primäre Teiltransaktion
PVS	Public Volume Set
REC-REF	Record-Reference
RSQ	Record-Sequence-Number (Satzfolgenummer)
SC	Subcontrol
SCD	Set Connection Data
SCI	Software Configuration Inventory
SECOLTP	Secure Online Transaction Processing
SECOS	Security Control System
SET-REF	Set-Reference
SIA	Schema Information Area
SIB	SQL Interface Block
SLF	Session-Log-File
SQL	Structured Query Language

SSD	Solid State Disk
SSIA	Subschema Information Area
SSITAB	Subschema Information Table
SSL	Storage Structure Language
ST	Servertask
STT	Sekundäre Teiltransaktion
TA	Transaction
TA-ID	Transaction-Identification
TANGRAM	Task and Group Affinity Management
TCUA	Transaction Currency Area
UDS/SQL	Universelles Datenbanksystem/Structured Query Language
UWA	User Work Area

18 Literatur



Error while executing colored text function; No macro Fujitsu ParameterColoredText found on this page!
Text box created by Fujitsu colored text function, delete this box if function was executed unintentionally.

Die Handbücher finden Sie im Internet unter <https://bs2manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.

UDS/SQL (BS2000)

Anwendungen programmieren

Benutzerhandbuch

UDS/SQL (BS2000)

Aufbauen und Umstrukturieren

Benutzerhandbuch

UDS/SQL (BS2000)

Entwerfen und Definieren

Benutzerhandbuch

UDS/SQL (BS2000)

Meldungen

Benutzerhandbuch

UDS/SQL (BS2000)

Sichern, Informieren und Reorganisieren

Benutzerhandbuch

UDS/SQL (BS2000)

Taschenbuch

UDS (BS2000)

Dialogsystem IQS

Benutzerhandbuch

UDS-KDBS (BS2000)

Kompatible Datenbankschnittstelle

Benutzerhandbuch

SQL für UDS/SQL

Sprachbeschreibung

BS2000 OSD/BC

Kommandos

Benutzerhandbuch

BS2000 OSD/BC

Einführung in die Systembetreuung

Benutzerhandbuch

BS2000 OSD/BC

Makroaufrufe an den Ablaufteil

Benutzerhandbuch

BS2000 OSD/BC

Einführung in das DVS

Benutzerhandbuch

SDF (BS2000)

Dialogschnittstelle SDF

Benutzerhandbuch

SORT (BS2000)

Benutzerhandbuch

SPACEOPT (BS2000)

Optimierung und Reorganisation von Platten

Benutzerhandbuch

LMS (BS2000)

SDF-Format

Benutzerhandbuch

DSSM/SSCM

Verwaltung von Subsystemen in BS2000

Benutzerhandbuch

ARCHIVE (BS2000)

Benutzerhandbuch

DRV (BS2000)

Dual Recording by Volume

Benutzerhandbuch

HSMS / HSMS-SV (BS2000)

Hierarchisches Speicher Management System

Band 1: Funktionen, Verwaltung und Installation

Benutzerhandbuch

SECOS (BS2000)

Security Control System

Benutzerhandbuch

openNet Server (BS2000)

BCAM

Referenzhandbuch

DCAM (BS2000)

Programmschnittstellen

Beschreibung

DCAM (BS2000)

Makroaufrufe

Benutzerhandbuch

OMNIS/OMNIS-MENU (BS2000)

Funktionen und Kommandos

Benutzerhandbuch

OMNIS/OMNIS-MENU (BS2000)

Administration und Programmierung

Benutzerhandbuch

openUTM

Konzepte und Funktionen

Benutzerhandbuch

openUTM

Anwendungen programmieren mit KDCS für COBOL, C und C++

Benutzerhandbuch

openUTM

Anwendungen generieren

Benutzerhandbuch

openUTM

Anwendungen administrieren

Benutzerhandbuch

openUTM

Einsatz von openUTM-Anwendungen unter BS2000

Benutzerhandbuch

openUTM

Meldungen, Test und Diagnose (BS2000)

Benutzerhandbuch

COBOL2000 (BS2000)

COBOL-Compiler

Sprachbeschreibung

COBOL2000 (BS2000)

COBOL-Compiler

Benutzerhandbuch

COBOL85 (BS2000)

COBOL-Compiler

Beschreibung

COBOL85 (BS2000)

COBOL-Compiler

Benutzerhandbuch

CRTE (BS2000)

Common Runtime Environment

Benutzerhandbuch

DRIVE/WINDOWS (BS2000)

Programmiersystem

Benutzerhandbuch

DRIVE/WINDOWS (BS2000)

Programmiersprache

Sprachbeschreibung

DRIVE/WINDOWS (BS2000)

Lexikon der DRIVE-Anweisungen

Referenzhandbuch

DRIVE/WINDOWS (BS2000/SINIX)

Lexikon der DRIVE-SQL-Anweisungen für UDS

Referenzhandbuch

DAB (BS2000)

Disk Access Buffer

Benutzerhandbuch

XHCS (BS2000)

8-bit-Code- und Unicode-Unterstützung im BS2000

Benutzerhandbuch

Unicode im BS2000

Übersichtshandbuch

BS2000 OSD/BC

Softbooks Deutsch

CD-ROM

openSM2 (BS2000)

Software Monitor

Benutzerhandbuch

SNMP Management (BS2000)

Benutzerhandbuch