

Deutsch



FUJITSU Software BS2000

UDS/SQL V2.8

Taschenbuch

Taschenbuch

Ausgabe März 2016

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

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

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

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

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

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

Copyright und Handelsmarken

Copyright © 2016 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.

Inhalt

1	Einleitung	9
1.1	Konzept der UDS/SQL-Dokumentation	10
1.2	Zielsetzung und Zielgruppen des Handbuchs	14
1.3	Konzept des Handbuchs	15
1.4	Änderungen gegenüber den Vorgänger-Handbüchern	16
1.5	Darstellungsmittel	18
1.5.1	Warnhinweise und Hinweise	18
1.5.2	Nicht-SDF-Darstellungsmittel	18
1.5.3	SDF-Syntaxdarstellung	20
1.5.4	Spaltenkonventionen für Schema-DDL, SSL, Subschema-DDL und COBOL-DML	26
2	Schema-DDL	27
2.1	Aufbau der Schema-DDL	27
2.2	Schema-Eintrag	28
2.3	Realm-Eintrag	28
2.4	Satz-Eintrag	29
2.4.1	Klauseln des Satz-Eintrags	31
2.5	Set-Eintrag	32
2.5.1	Klauseln des Set-Eintrags	33
3	SSL	35
3.1	Aufbau der SSL	35
3.2	Schema-Eintrag	36

3.3	Satz-Eintrag	36
3.3.1	Klauseln des Satz-Eintrags	37
3.4	Set-Eintrag	38
3.4.1	Klauseln des Set-Eintrags	39
4	Subschema-DDL	41
<hr/>		
4.1	Aufbau der Subschema-DDL	41
4.2	IDENTIFICATION DIVISION	42
4.3	AREA SECTION	42
4.4	RECORD SECTION	43
4.5	SET SECTION	44
5	COBOL-DML	45
<hr/>		
5.1	Aufbau eines COBOL-Programms	45
5.2	Anweisungen	47
5.3	Erzeugen eines Anwenderprogramms	53
6	CALL-DML	57
<hr/>		
6.1	Parameterdefinitionen	57
6.2	Format-Tabelle	58
6.3	Der Benutzerinformationsbereich	60
6.4	Übersicht über die CALL-DML-Funktionen	61
6.5	LOOKC	76
6.6	Assembler-Makros der CALL-DML	92
6.7	Erzeugen eines Anwenderprogramms	93

7	Anweisungscodes und Statuscodes der COBOL-DML und CALL-DML	95
7.1	Anweisungscodes	96
7.2	Auswirkungen des Statuscodes	97
7.3	Kombinationen von Anweisungscodes und Statuscodes	98
	Statuscodes bei FIND/FETCH	100
7.4	Statuscodes der DML	102
7.5	Statuscodes der CALL-DML	115
8	DMLTEST	121
8.1	Testen der DML-Funktionen	121
8.2	Schlüsselwortparameter	122
8.3	Schlüsselwörter	123
8.4	Übersicht über die DMLTEST-Kommandos	124
8.5	Übersicht über die Unterschiede zwischen den DMLTEST-DML-Anweisungen und den COBOL-DML-Anweisungen	127
9	Schema-DDL, SSL und Subschema-DDL übersetzen	129
9.1	Kommandofolge zum Übersetzen der Schema-DDL	129
9.2	Kommandofolge zum Übersetzen der SSL	130
9.3	Kommandofolge zum Übersetzen der Subschema-DDL	131
9.4	Anweisungen für den DDL-Compiler bzw. den SSL-Compiler	132
10	Datenbankbetrieb	137
10.1	Datenbankbetrieb mit dem independent DBH	137
10.1.1	UDSADM	151
10.2	Datenbankbetrieb mit dem linked-in DBH	153
10.3	UDSMON	158
10.4	Pubsets für UDS/SQL nutzen	166

10.5	Jobvariablen für UDS/SQL nutzen	167
10.5.1	Pubset-Deklarations-Jobvariable	168
10.5.2	Session-Jobvariable	170
10.5.3	Datenbank-Jobvariable	173
11	Dienstprogramme	177
<hr/>		
11.1	START-Kommandos der UDS/SQL-Programme	177
11.2	BALTER	179
11.3	BCALLSI	183
11.4	BCHANGE	185
11.5	BCHECK	186
11.6	BCREATE	189
11.7	BFORMAT	190
11.8	BGSIA	192
11.9	BGSSIA	194
11.10	BINILOAD	195
11.11	BMEND	202
11.12	BMODTT	206
11.13	BOUTLOAD	208
11.14	BPGSIZE	211
11.15	BPRECORD	213
11.16	BPRIVACY	216
11.17	BPSIA	219
11.18	BPSQLSIA	220
11.19	BRENAME	221
11.20	BREORG	222
11.21	BSTATUS	229
11.22	BTRANS24	232
11.23	ONLINE-PRIVACY	233
11.24	UDS-Online-Utility	236

12 Funktionscodes der DML-Anweisungen 241

Literatur 245

Stichwörter 251

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 auf [Seite 13](#).

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	–	–	–	–

Tabelle 1: Wegweiser durch die Handbücher

(Teil 1 von 3)

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Program- mieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-D

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	-	-	-

Tabelle 1: Wegweiser durch die Handbücher

(Teil 2 von 3)

Inhalt der fünf Haupthandbücher	Aufgaben des Anwenders							
	Entwerfen und Definieren	COBOL/ CALL-DML Programm.	SQL- Program- mieren	Aufbauen und Umstrukt.	Verwalten und Bedienen	Arbeiten mit openUTM	Arbeiten mit IQS	Arbeiten mit UDS-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

(Teil 3 von 3)

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 Taschenbuch ist für den UDS/SQL-Kenner gedacht. Es dient zum Nachschlagen bei der praktischen Arbeit.

1.3 Konzept des Handbuchs

Was enthält dieses Handbuch?

Hier finden Sie in konzentrierter Form eine Zusammenstellung aller wichtigen Syntaxbeschreibungen, Tabellen und Entscheidungshilfen aus den UDS/SQL-Handbüchern:

- [Entwerfen und Definieren](#)
- [Anwendungen programmieren](#)
- [Aufbauen und Umstrukturieren](#)
- [Datenbankbetrieb](#)
- [Sichern, Informieren und Reorganisieren](#)

Wie finden Sie sich im Handbuch zurecht?

Zusätzlich zum Inhaltsverzeichnis können Sie die Antworten auf Ihre Fragen gezielt über die Kolumnentitel nachschlagen.

Wollen Sie Detailinformationen, so finden Sie über die im Taschenbuch stehenden Verweise den Einstieg in die verschiedenen UDS/SQL-Handbücher.

Readme-Datei

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

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

Informationen unter BS2000

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

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

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

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

Ergänzende Produkt-Informationen

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

1.4 Änderungen gegenüber den Vorgänger-Handbüchern

In der folgenden **Tabelle 2** sind die wichtigsten Änderungen der Version UDS/SQL V2.8 gegenüber der Version V2.7 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
UDSMON-Utility: Verbesserungen bei Transaktionszeit und DB COUNTERS		
Für die Ausgabe auf Terminal und Drucker: In der UDS/SQL-Monitor-Maske wird COUNTER, die Einheit für die Ausgabe der AVG TRANSACTION TIME, präzisiert auf Sekunden und Millisekunden für das Monitoring von kurzen Transaktionen.	DBB	11
Neues DISPLAY DBCOUNTERS-Kommando in UDSMON für die Ausgabe von Datenbankzählern.	DBB	11
BSTATUS-Utility: Begrenzung des TABLE STATISTICS FOR OWNER IN SET		
Verbesserte DISPLAY TABLE FOR OWNER-Anweisung, um eine Begrenzung von TABLE STATISTICS FOR OWNER IN SET auf Sätze von bestimmten Ownern oder Satzbereiche zu ermöglichen.	SIR	6
Neue Meldungen des Dienstprogramms BSTATUS	MEL	3
Neue Meldung des Dienstprogramms BPRECORD 2553 falls der Wert 0 als Startwert in RSQ range definiert wird.	MEL	3

Tabelle 2: Änderungen in V2.8 gegenüber V2.7

Thema	Handbuch	Kapitel
Datenbankbetrieb: Die Anzahl der DML-Anweisungen und die Anzahl der Ein- und Ausgaben werden pro Datenbank gezählt.	DBB	4
	MEL	2
BOUTLOAD-Utility: Ausgabe im CSV-Format	AUF	5
	MEL	3
COPY-RECORD-Anweisung: Neuer Operand CSV-OUTPUT	AUF	5
Neues Ausgabeformat CSV	AUF	5
Online-Utility – Probable Position Pointers (PPP) reorganisieren		
Neue DML REORGPPP - PPPs reorganisieren	SIR	8
Neue SDF-Anweisungen: SET-REORGANIZE-PPP-PARAMETERS, SHOW-REORGANIZE-PPP-PARAMETERS	SIR	8
Neue Prozedur-Anweisung REORGPPP	SIR	8
Neue vordefinierte Variablen: REORG-PPP-CURRENT, REORG-PPP-LOCKED, REORG-PPP-PAGES	SIR	8
Neue vordefinierte Standardprozedur *STDREPPP	SIR	8
Neues Beispiel „Reorganisieren von PPPs“	SIR	8
Neue Statuscodes mit Fortschrittshinweisen der Online-Utility REORGPPP und neue Fehlercodes	ANW	10

Tabelle 2: Änderungen in V2.8 gegenüber V2.7



Allgemeine Änderung

Die bisherige Bezeichnung BS2000/OSD-BC des BS2000-Grundausbaus ändert sich und lautet ab Version V10.0: BS2000 OSD/BC.

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
 VORSICHT!	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.	DATABASE- <u>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>fieldname</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>]

Tabelle 3: Zeichen der Metasprache

(Teil 1 von 2)

Sprachelement	Erklärung	Beispiel
... oder ,...	Den unmittelbar vorstehenden Ausdruck können Sie wahlweise mehrmals wiederholen. Die beiden Sprachelemente unterscheiden Wiederholungen mit Leerzeichen oder mit Komma als Trennzeichen.	<i>fieldname,...</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</u> NAME . .
^	Den Punkt müssen Sie angeben, gefolgt von mindestens einem Leerzeichen. Die Unterstreichung geben Sie nicht an.	<u>SET SECTION</u> . 03 <i>fieldname</i> ^
Zwischenraum	Bedeutet, dass Sie mindestens ein Leerzeichen angeben müssen.	<u>USING</u> <u>CALC</u>

Tabelle 3: Zeichen der Metasprache

(Teil 2 von 2)

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 22 und 25).	DATABASE = <dbname>

Tabelle 4: Metasyntax

(Teil 1 von 2)

Kennzeichnung	Bedeutung	Beispiele
<u>Unterstreichung</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.	USER-GROUP-NAME = *KSET-FORMAT(...) *KSET-FORMAT(...) HOST = <host>
	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.	USER-GROUP-NAME = *ALL-EXCEPT(...) *ALL-EXCEPT(...) NAME = *KSET-FORMAT(...) *KSET-FORMAT(...) HOST = <host> ...
,	Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.	,SPACE = <u>STD</u>
list-poss(n):	Aus den list-poss 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

(Teil 2 von 2)

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

Tabelle 5: Datentypen

(Teil 1 von 3)

Datentyp	Zeichenvorrat	Besonderheiten
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

Tabelle 5: Datentypen

(Teil 2 von 3)

Datentyp	Zeichenvorrat	Besonderheiten
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 Eingabeformat: $\left. \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}$ 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.

Tabelle 5: Datentypen

(Teil 3 von 3)

Zusatz	Bedeutung
<i>x..y unit</i>	beim Datentyp integer: Intervallangabe x Mindestwert, der für integer erlaubt ist. x ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf. y Maximalwert, der für integer erlaubt ist. y ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf. <i>unit</i> nur bei integer: zusätzliche Einheiten. Folgende Angaben werden verwendet: <i>Mbyte</i> <i>Kbyte</i> <i>seconds</i>

Tabelle 6: Zusätze zu Datentypen

1.5.4 Spaltenkonventionen für Schema-DDL, SSL, Subschema-DDL und COBOL-DML

ab Spalte 7

kennzeichnen Sie besondere Zeilen von DLL und SSL:

- * Kommentarzeile; den Text in Spalte 8-72 fasst der COBOL-Compiler als Kommentar auf.
- / Seitenvorschub
- Fortsetzungszeile, wenn Eingaben in der vorhergehenden Zeile die Spalte 72 überschreiten würden.

ab Spalte 8

müssen Sie schreiben:

- bei Schema-DDL und SSL:
 - die jeweils erste Klausel eines Eintrags
 - die MEMBER-Klausel
- in der Subschema-DDL:
 - die erste Zeile einer DIVISION
 - die erste Zeile einer SECTION
 - die Stufennummer 01
- in der COBOL-DML:
Klauseln abhängig vom COBOL-Referenzformat

ab Spalte 12

bis max. Spalte 72 einschließlich alle weiteren Klauseln von DDL und SSL

2 Schema-DDL

2.1 Aufbau der Schema-DDL

Schema-Eintrag	{ SCHEMA NAME-Klausel [PRIVACY LOCK-Klausel]]_
Realm-Eintrag	{ AREA NAME-Klausel [TEMPORARY-Klausel]]_
Satz-Eintrag	{ RECORD NAME-Klausel [LOCATION MODE-Klausel] WITHIN-Klausel [SEARCH KEY-Klausel]]_ Satzelementname-Klausel [PICTURE-Klausel] [TYPE-Klausel] [OCCURS-Klausel]]_
Set-Eintrag	{ SET-NAME-Klausel [DYNAMIC-Klausel] ORDER-Klausel OWNER-Klausel]_ [MEMBER-Klausel] [ASCENDING/DESCENDING-KEY-Klausel] [SEARCH KEY-Klausel] [SET OCCURRENCE SELECTION-Klausel]]]_

Bild 1: Aufbau der Schema-DDL

Die Beschreibung der logischen Datenstruktur beginnen Sie immer mit dem Schema-Eintrag und mindestens einem Realm-Eintrag.

Für die Abfolge der Satz-, Set- und der übrigen Realm-Einträge gilt:

- Bevor Sie einen Set definieren, müssen Sie die beiden zugehörigen Satzarten definieren.
- Bevor Sie eine Satzart definieren, müssen Sie alle Realms definieren, die Sie in dessen WITHIN-Klausel nennen.

2.2 Schema-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.1.1)

```
SCHEMA NAME IS schemaname
  [PRIVACY LOCK FOR COPY IS literal-1 [ OR literal-2 ] ]_
```

Sie benennen das Schema und vereinbaren Kennwörter, die das Schema vor dem unbefugten Erstellen eines Subschemas schützen.

2.3 Realm-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.1.2)

```
AREA NAME IS realmname
  [AREA IS TEMPORARY]_
```

Sie benennen einen Realm und erklären ihn ggf. zum Temporären Realm.



In einer Datenbank mit 2048 byte Seitenlänge können Sie maximal 123 Realms definieren.

In einer Datenbank mit 4000 byte oder 8096 byte Seitenlänge können Sie maximal 245 Realms definieren.

Sie dürfen nur *einen* Temporären Realm definieren.

Sie benennen eine Satzart und

- bestimmen die Verteilung ihrer Sätze auf Realms,
- bestimmen ggf. die Reihenfolge der Sätze zur sequenziellen Verarbeitung,
- legen ggf. zusätzliche Zugriffspfade für Direktzugriff über Primär- und Sekundärschlüssel an,
- definieren alle Satzelemente, die der Satzart angehören sollen.



In einer Datenbank mit 2048 byte Seitenlänge können Sie maximal 253 Satzarten definieren.

In einer Datenbank mit 4000 byte oder 8096 byte Seitenlänge können Sie maximal 32 766 Satzarten definieren.

Die Klauseln des Satz-Eintrags sind nachfolgend erläutert.

2.4.1 Klauseln des Satz-Eintrags

Klausel	Bedeutung
<p><u>RECORD</u> NAME IS <i>satzname</i></p>	<p>Sie benennen die Satzart.</p>
<p>[<u>LOCATION</u> MODE IS { { <u>DIRECT</u> } { <u>DIRECT-LONG</u> } { <u>CALC</u>[<i>hashroutine</i>] <u>USING</u> <i>feldname-2</i>,... <u>DUPLICATES ARE</u> [<u>NOT</u>] <u>ALLOWED</u> }]</p> <p style="margin-left: 100px;"> { <i>feldname-1</i> { <u>IN</u> } <i>satzname</i> } { <i>bezeichner</i> } </p>	<p>Sie schaffen ggf. die Möglichkeit, den Database Key eines zu speichernden Satzes selbst zu vergeben und die Reihenfolge bei sequentieller Verarbeitung selbst zu bestimmen.</p> <p>Sie legen einen Primärschlüssel zum Zwecke des Direktzugriffs auf einen bestimmten Satz oder eine Menge von Sätzen mit gleichen Schlüsselwerten an.</p>
<p><u>WITHIN</u> <i>realmname-1</i>[,<i>realmname-2</i>,... <u>AREA-ID</u> IS <i>bezeichner</i>]</p> <p>[<u>SEARCH</u> KEY IS <i>feldname</i>,... <u>USING</u> { { <u>CALC</u>[<i>hashroutine</i>] } { <u>INDEX</u> } }] [<u>NAME</u> IS <i>name</i>]</p> <p>DUPLICATES ARE [<u>NOT</u>] <u>ALLOWED</u>]....</p>	<p>Sie bestimmen die Verteilung der Sätze dieser Satzart auf Realms.</p> <p>Sie legen zusätzliche Direktzugriffspfade über Sekundärschlüssel an und benennen die SEARCH-KEY-Tabelle bzw. den Hashbereich, um ihn mit der SSL anzusprechen zu können.</p>
<p>[<i>stufennummer</i>]<u>satzelementname</u></p>	<p>Sie benennen ein Satzelement und legen fest, ob es zu einer Wiederholungsgruppe gehört.</p> <p>Standardwert ist 1.</p>
<p><u>PICTURE</u> IS { <i>maskenzeichenkette</i> { <u>LX</u>(<i>ganzzahl-1</i>) <u>DEPENDING</u> ON <i>feldname</i> } }</p>	<p>Hiermit definieren Sie ungepackte numerische Felder, alphanumerische Felder fester oder variabler Länge oder nationale Felder.</p>
<p><u>TYPE</u> IS { { <u>FIXED</u> <u>REAL</u> { <u>BINARYC</u> { {15} } { {31} } } { <u>DECIMALC</u> [<i>ganzzahl-1</i>[,<i>ganzzahl-2</i>]] } } { <u>CHARACTERC</u> [<i>ganzzahl-3</i>[<u>DEPENDING</u> ON <i>feldname</i>]] } { <u>DATABASE-KEY</u> } { <u>DATABASE-KEY-LONG</u> } }</p>	<p>Hiermit definieren Sie</p> <ul style="list-style-type: none"> - gepackte numerische Felder, - binäre Felder, - alphanumerische Felder fester oder variabler Länge oder - Database-Key-Felder.
<p>[<u>OCCURS</u> <i>ganzzahl</i> <u>TIMES</u>]</p>	<p>Sie definieren den Wiederholungsfaktor für einen Vektor oder eine Wiederholungsgruppe.</p>

Tabelle 7: Klauseln des Satz-Eintrags der Schema-DDL

2.5 Set-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.1.4)

SET NAME IS *setname*

[SET IS DYNAMIC]

ORDER IS {
LAST
FIRST
NEXT
PRIOR
IMMATERIAL
SORTED[INDEXED[NAME IS *name*]]
 BY {DATABASE-KEY
DEFINED KEYS DUPLICATES ARE NOT] ALLOWED }

OWNER IS { *satzname*
SYSTEM } ±

[MEMBER IS *satzname* { MANDATORY } { AUTOMATIC }
 { OPTIONAL } { MANUAL }]

[{ ASCENDING }
 { DESCENDING }] KEY IS *feldname-1,...*

[SEARCH KEY IS *feldname-2,...* USING { CALC[*hashroutine*]
INDEX }] [NAME IS *name*]

DUPLICATES ARE [NOT] ALLOWED...

[SET OCCURRENCE SELECTION IS

{ CURRENT OF SET
 THRU { LOCATION MODE OF OWNER[ALIAS FOR { *feldname-3*
bezeichner-1 }]] ±
 IS *bezeichner-2*... }

Sie benennen einen Set und

- erklären den Set ggf. zum dynamischen Set,
- legen die Reihenfolge der Membersätze innerhalb der Set-Occurrences für eine sequenzielle Verarbeitung fest,
- legen ggf. zusätzliche Zugriffspfade über Primär- und Sekundärschlüssel an,
- bestimmen eine Satzart zur Ownersatzart des Set,
- bestimmen ggf. eine Satzart zur Membersatzart des Set und definieren die Art der Set-Mitgliedschaft der Membersätze und
- bestimmen ggf. die Auswahlmethode für die Set-Occurrences des Set.



Pro Datenbank können Sie maximal 32 766 Sets definieren.

Pro Satzart, die Owner eines Sets ist, dürfen Sie in diesen Sets in Summe maximal 255 Tabellen erzeugen; eine Tabelle wird angelegt, wenn der Set-Mode Pointer-Array oder List oder Chain sorted indexed ist, ebenso für jeden Sekundärschlüssel in diesen Sets.

Unabhängig davon dürfen Sie maximal 255 Sekundärschlüssel auf Satzartebene pro Satzart sowie auf Setebene pro singulärem Set definieren, wobei Hashverfahren nicht mitzählen.

Die Klauseln des Set-Eintrags sind nachfolgend erläutert.

2.5.1 Klauseln des Set-Eintrags

Klausel	Bedeutung
<code>SET NAME IS <i>setname</i></code>	Sie benennen den Set.
<code>[SET IS <u>DYNAMIC</u>]</code>	Sie erklären den Set zum dynamischen Set.

Tabelle 8: Klauseln des Set-Eintrags (Schema-DDL)

(Teil 1 von 2)

Klausel	Bedeutung
<pre>ORDER IS { LAST FIRST NEXT PRIOR IMMATERIAL SORTED[INDEXED[NAME IS name]] BY { DATABASE-KEY [DEFINED KEYS DUPLICATES ARE[NOT] ALLOWED] }</pre>	<p>Sie legen</p> <ul style="list-style-type: none"> – die Reihenfolge der Sätze innerhalb der Set-Occurrences für eine sequenzielle Verarbeitung fest. – einen zusätzlichen Direktzugriffspfad über den Primärschlüssel an.
<pre>OWNER IS { satzname SYSTEM }</pre>	<p>Sie bestimmen eine von Ihnen definierte Satzart oder die symbolische Satzart SYSTEM zur Ownersatzart des Set.</p>
<pre>MEMBER IS satzname { MANDATORY } { AUTOMATIC OPTIONAL } { MANUAL }</pre>	<p>Die gesamte Beschreibung der Membersatzart entfällt im dynamischen Set. Sonst bestimmen Sie hiermit eine Satzart zur Membersatzart und definieren die Art der Set-Mitgliedschaft der Membersätze.</p>
<pre>{ ASCENDING } { DESCENDING } KEY IS feldname,...]</pre>	<p>Hiermit definieren Sie ein Feld oder eine Kombination von Feldern der Membersatzart zum Sortierschlüssel, nach dessen Werten die Membersätze innerhalb der Set-Occurrence auf- oder absteigend sortiert werden.</p>
<pre>[SEARCH KEY IS feldname,... USING { CALC[hashroutine] INDEX } [NAME] IS name DUPLICATES ARE[NOT] ALLOWED]...</pre>	<p>Sie legen zusätzliche Direktzugriffspfade über Sekundärschlüssel an und benennen die SEARCH-Key-Tabelle bzw. den Hashbereich, um ihn mit der SSL ansprechen zu können.</p>
<pre>[SET OCCURRENCE SELECTION IS CURRENT OF SET THRU { LOCATION MODE OF OWNER[ALIAS FOR { feldname bezeichner-1 }] IS bezeichner-2]...]</pre>	<p>Diese Klausel müssen Sie genau dann angeben, wenn der Set kein SYSTEM-Set ist. Sie bestimmen damit die Auswahlmethode für die Set-Occurrences des Set.</p>

Tabelle 8: Klauseln des Set-Eintrags (Schema-DDL)

(Teil 2 von 2)

3 SSL

3.1 Aufbau der SSL

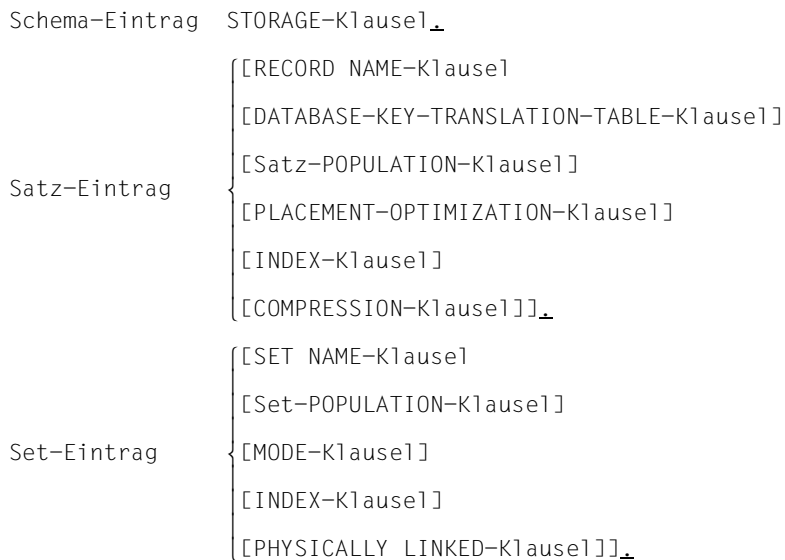


Bild 2: Aufbau der SSL

Die Beschreibung der physischen Speicherstruktur ist wahlfrei. Wenn Sie darauf verzichten, benutzt UDS/SQL voreingestellte Standardwerte, die bei nachfolgenden Erläuterungen der einzelnen Syntaxelemente jeweils angegeben sind.

Im anderen Fall beginnen Sie die Beschreibung immer mit der STORAGE-Klausel. Die Abfolge der Satz- und Set-Beschreibungen ist beliebig.

Alle Namen, die Sie in der Speicherstrukturbeschreibung benutzen, müssen Sie zuvor mit der Schema-DDL definiert haben.

3.2 Schema-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.2.1)

STORAGE STRUCTURE OF SCHEMA *schemaname*.

Sie geben den Namen des Schemas an, auf das sich die Beschreibung der Speicherstruktur beziehen soll.

3.3 Satz-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.2.2)

RECORD NAME IS *satzname*

[DATABASE-KEY-TRANSLATION-TABLE[IS *ganzzahl-1*][WITHIN *realmname-1*]]

[POPULATION IS {*ganzzahl-2* WITHIN *realmname-2*}, ...]

[PLACEMENT OPTIMIZATION FOR SET *setname*]

[INDEX NAME IS *name*

[PLACING IS WITHIN *realmname-3*]

[TYPE IS {DATABASE-KEY-LIST
REPEATED-KEY
[DYNAMIC REORGANIZATION SPANS *ganzzahl-3* PAGES]}]]...

[COMPRESSION FOR ALL ITEMS].

Im Satz-Eintrag nennen Sie den Namen der Satzart, auf die sich die Beschreibung der Speicherstruktur bezieht und machen Angaben

- zur Größe und physischen Lage der DBTT und zur Größe der Hashbereiche für Satz-SEARCH-Keys,
- zur Anzahl der Sätze der Satzart bzw. zur Größe des Hashbereiches für den Primärschlüssel innerhalb bestimmter Realms,
- zur physischen Lage der Sätze innerhalb eines Realms, falls die Satzart Member in einem Set ist,

- zur physischen Lage, zum Typ und zum Reorganisationsaufwand der Satz-SEARCH-Key-Tabellen bzw. zur physischen Lage der Hashbereiche für Satz-SEARCH-Keys und
- zur komprimierten Speicherung.

Die Klauseln des Satz-Eintrages sind nachfolgend erläutert.

3.3.1 Klauseln des Satz-Eintrags

Klausel	Bedeutung
<code>RECORD NAME IS <i>satzname</i></code>	Sie geben den Namen der Satzart an, auf die sich der Satz-Eintrag bezieht.
<code>[DATABASE-KEY-TRANSLATION-TABLE[IS <i>ganzzahl</i>] [WITHIN <i>realmname</i>]]</code>	Sie geben die Größe und Lage der DBTT und gleichzeitig die Größe der Hashbereiche für Satz-SEARCH-Keys an.
<code>[POPULATION IS {<i>ganzzahl</i> WITHIN <i>realmname</i>},...]</code>	Hier geben Sie die Größe und die Lage von Hashbereichen für den Primärschlüssel (LOCATION MODE IS CALC) an. Die Angabe dient UDS/SQL außerdem zur Abschätzung der Realm-Größen.
<code>[PLACEMENT OPTIMIZATION FOR SET <i>setname</i>]</code>	Hiermit erreichen Sie eine Bündelung der Sätze bei dem Ownersatz des Set <i>setname</i> .
<code>[INDEX NAME IS <i>name</i> [PLACING IS WITHIN <i>realmname</i>] [TYPE IS { DATABASE-KEY-LIST REPEATED-KEY [DYNAMIC REORGANIZATION SPANS <i>ganzzahl</i> PAGES]}]]...]</code>	Sie nennen den Namen der Satz-SEARCH-Key-Tabelle bzw. des Hashbereichs auf den sich die Beschreibung beziehen soll und bestimmen <ul style="list-style-type: none"> – für eine Satz-SEARCH-Key-Tabelle die physische Lage, den Typ und den Reorganisationsaufwand, – für einen Hashbereich die physische Lage.
<code>[COMPRESSION FOR ALL ITEMS]</code>	Sie veranlassen UDS/SQL, die Sätze komprimiert zu speichern, wenn sie an der DML-Schnittstelle in komprimierter Form zur Verfügung gestellt werden.

Tabelle 9: Klauseln des Satz-Eintrags (SSL)

3.4 Set-Eintrag

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.2.3)

SET NAME IS *setname*

[POPULATION IS *ganzzahl-1*-[INCREASE IS *ganzzahl-2*]]

[MODE IS {
 {CHAIN[LINKED TO PRIOR]
 {POINTER-ARRAY } {ATTACHED TO OWNER
 {DETACHED[WITHIN *realmname-1*]} }
 {LIST } { [WITH PHYSICAL LINK]} }]

[DYNAMIC REORGANIZATION SPANS *ganzzahl-3* PAGES]

[INDEX NAME IS *name*

[PLACING IS {
 {ATTACHED TO OWNER
 {DETACHED[WITHIN *realmname-2*]} }]

[TYPE IS {
 {DATABASE-KEY-LIST
 {REPEATED-KEY }]]
 { [DYNAMIC REORGANIZATION SPANS *ganzzahl-4* PAGES] }]]...

[MEMBER IS PHYSICALLY LINKED TO OWNER]_.

Im Set-Eintrag nennen Sie den Namen des Set, auf den sich die Beschreibung der Speicherstruktur bezieht und machen Angaben

- zum durchschnittlichen Umfang der Set-Occurrences,
- zur Verknüpfung der Sätze innerhalb der Set-Occurrences,
- zur Platzierung und zum Reorganisationsaufwand von Adresslisten, Listen, Sort-Key-Tabellen und Set-SEARCH-Key-Tabellen,
- zum Typ von Set-SEARCH-Key-Tabellen,
- für einen zusätzlichen Zeiger vom Member auf seinen Owner.

Die Klauseln des Set-Eintrags sind nachfolgend erläutert.

3.4.1 Klauseln des Set-Eintrags

Klausel	Bedeutung
<code>SET NAME IS <i>setname</i></code>	Hier geben Sie den Namen des Set an, auf den sich der Set-Eintrag bezieht.
<code>[POPULATION IS <i>ganzzahl</i>-1[INCREASE IS <i>ganzzahl</i>-2]]</code>	Sie geben die durchschnittliche Anzahl von Membersätzen der Set-Occurrences an, die Sie beim Umladen der Datenbank bzw. bei späteren Erweiterungen der Set-Occurrences erwarten.
$ \left[\text{MODE IS} \left\{ \begin{array}{l} \text{CHAIN[LINKED TO PRIOR]} \\ \text{POINTER-ARRAY} \\ \text{LIST} \end{array} \right\} \left\{ \begin{array}{l} \text{ATTACHED TO OWNER} \\ \text{DETACHED[WITHIN } \textit{realmname} \text{]} \\ \text{[WITH PHYSICAL LINK]} \end{array} \right\} \right] $ <code>[DYNAMIC REORGANIZATION SPANS <i>ganzzahl</i> PAGES]</code>	Sie bestimmen die Verknüpfung der Sätze innerhalb der Set-Occurrences. Falls Sie eine Adressliste, Liste oder Sort-Key-Tabelle anlegen, bestimmen Sie auch den Reorganisationsaufwand für diese Tabellen.
<code>[INDEX NAME IS <i>name</i></code> $ \left[\text{PLACING IS} \left\{ \begin{array}{l} \text{ATTACHED TO OWNER} \\ \text{DETACHED[WITHIN } \textit{realmname} \text{]} \end{array} \right\} \right] $ $ \left[\text{TYPE IS} \left\{ \begin{array}{l} \text{DATABASE-KEY-LIST} \\ \text{REPEATED-KEY} \\ \text{[DYNAMIC REORGANIZATION SPANS } \textit{ganzzahl} \text{ PAGES]} \end{array} \right\} \right] $	Wenn Sie mit der Schema-DDL eine SEARCH-Key- oder eine Sort-Key-Tabelle dieses Set benannt haben, können Sie die Tabelle hier ansprechen, um die Lage, den Typ und den Reorganisationsaufwand für die Tabelle festzulegen. Wenn Sie einen SEARCH-Key zur Speicherung in einen Hashbereich definiert haben, können Sie den Realm bestimmen, in dem der Hashbereich liegen soll. Standardwerte sind: – für PLACING: DETACHED – für TYPE: REPEATED-KEY
<code>MEMBER IS PHYSICALLY LINKED TO OWNER</code>	Hiermit legen Sie in jedem Membersatz dieses Set einen zusätzlichen Zeiger zu ihrem zugehörigen Owner an: Die SCD eines Membersatzes enthält zusätzlich den Probable Position Pointer (PPP) des zugehörigen Ownersatzes.

Tabelle 10: Klauseln des Set-Eintrags (SSL)

4 Subschema-DDL

4.1 Aufbau der Subschema-DDL

```
IDENTIFICATION DIVISION_
SUB-SCHEMA NAME-Klausel
[PRIVACY LOCK-Klausel]
[PRIVACY KEY-Klausel]_
DATA DIVISION_
AREA SECTION_
COPY-Klausel_
RECORD SECTION_
[ COPY-Klausel_]

{ [Satzname-Klausel]_
  Satzelementname-Klausel
  [GROUP-USAGE-Klausel]
  [PICTURE-Klausel]
  [USAGE-Klausel]
  [OCCURS-Klausel]_
  [Bedingungsname-Klausel]
  [VALUE-Klausel_] ]

[SET SECTION_
COPY-Klausel_]
```

Satz-Eintrag

Bild 3: Aufbau der Subschema-DDL

Die im Bild gezeigte Abfolge von Klauseln müssen Sie mit folgenden Ausnahmen einhalten:

- Die Reihenfolge von PICTURE- und USAGE-Klausel ist beliebig.
- Eine COPY-Klausel in der RECORD SECTION darf auch auf einen Satz-Eintrag folgen.

Namen, die Sie in der Definition des Subschemas benutzen, müssen Sie zuvor mit der Schema-DDL vereinbart haben.
Die Ausnahme sind Namen für Datengruppen und Bedingungen, die Sie im Subschema neu definieren.

4.2 IDENTIFICATION DIVISION

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.3.1)

```
IDENTIFICATION DIVISION.
  SUB-SCHEMA NAME IS subschema OF SCHEMA NAME schemaname
  [PRIVACY LOCK FOR COMPILE IS literal-1 [OR literal-2]]
  [PRIVACY KEY FOR COPY IS literal-3].
```

Hiermit benennen Sie das Subschema und

- geben an, aus welchem Schema Sie das Subschema entnehmen wollen,
- vereinbaren Kennwörter zum Schutz gegen unbefugte Übersetzung eines DML-Programms mit diesem Subschema und
- nennen ggf. eines der Kennwörter, die das Schema vor unbefugter Entnahme eines Subschemas schützen.

4.3 AREA SECTION

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.3.2)

```
DATA DIVISION.
AREA SECTION.
  { COPY ALL AREAS. }
  { { COPY realmname, ... } ... }
```

Hiermit übernehmen Sie alle Realms oder eine Auswahl von Realms aus dem Schema in das Subschema.

4.5 SET SECTION

(siehe Handbuch „[Entwerfen und Definieren](#)“, Abschnitt 9.3.4)

SET SECTION.

```
{COPY ALL SETS.
 {COPY setname, ...} ... }
```

Hiermit übernehmen Sie alle Sets oder eine Auswahl von Sets aus dem Schema.

5 COBOL-DML

5.1 Aufbau eines COBOL-Programms

IDENTIFICATION DIVISION.
PROGRAM-ID.
[PRIVACY.]
Übliche COBOL-Informationen
ENVIRONMENT DIVISION,
Übliche COBOL-Informationen
DATA DIVISION.
Übliche COBOL-Sections
SUB-SCHEMA-SECTION.
DB-Eintrag
PROCEDURE DIVISION.
Übliche COBOL-Informationen

Erweiterungen des ANSCOBOL-Sprachumfangs sind die DML-Anweisungen:

ACCEPT, CONNECT, DISCONNECT, ERASE, FETCH, FIND, FINISH, FREE, GET, IF,
KEEP, MODIFY, READY, SET, STORE, USE.

IDENTIFICATION DIVISION

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.2, 7.2)

Ist das Subschema mit einer PRIVACY LOCK FOR COMPILE-Klausel geschützt, müssen Sie im COBOL-DML-Programm den entsprechenden PRIVACY KEY angeben.

```
[PRIVACY. PRIVACY KEY FOR COMPILE IS literal.]
```

DATA DIVISION

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.2, 7.3)

Der Datenteil eines COBOL-Programms, das mit der COBOL-DML arbeitet, muss ein Kapitel SUB-SCHEMA SECTION enthalten.

Dieses Kapitel muss das letzte des Datenteils sein. Es enthält die DB-Klausel.

Die DB-Klausel benennt ein Subschema und sorgt dafür, dass alle Bereiche, die für den Informationsaustausch mit dem DBH notwendig sind, in der UWA (User Work Area) reserviert werden.

```
DATA DIVISION.
```

```
  Übliche COBOL-Sections.
```

```
SUB-SCHEMA SECTION.
```

```
  DB subschemaname WITHIN schemaname.
```

PROCEDURE DIVISION

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.2, 7.4)

Im Prozedurteil geben Sie COBOL- und DML-Anweisungen an. In den DECLARATIVES innerhalb des Prozedurteils wird mit der für DML erweiterten USE-Anweisung auf Datenbanksonderzustände reagiert.

5.2 Anweisungen

(siehe Handbuch „Anwendungen programmieren“, Abschnitt 7.4.1)

Anweisung	Funktion
<p>Format 1:</p> <pre>ACCEPT <i>feldname-1</i> FROM { <i>satzname</i> <i>setname</i> <i>realmname</i> }] CURRENCY</pre> <p>Format 2:</p> <pre>ACCEPT <i>feldname-2</i> FROM { <i>satzname</i> <i>setname</i> <i>feldname-3</i> }] REALM-NAME</pre>	<p>überträgt den Data Base Key-Wert des CRR, CRS, CRA bzw. CRU in das Feld <i>feldname-1</i></p> <p>überträgt den Namen des Realm, in dem der CRR, CRS, CRU bzw. der zu einem angegebenen Data Base Key-Wert gehörige Satz gespeichert ist</p>
<pre>CONNECT [<i>satzname</i>] TO { <i>setname-1</i>, ... ALL } [RETAINING CURRENCY FOR { <i>setname-2</i>, ... SETS }]</pre>	hängt den CRU in Set-Occurrences ein
<p>Format 1:</p> <pre>DISCONNECT [<i>satzname</i>] FROM { <i>setname</i>, ... ALL }</pre> <p>Format 2:</p> <pre>DISCONNECT ALL FROM <i>setname</i>, ...</pre>	<p>löst den CRU aus Set-Occurrences</p> <p>entfernt alle Sätze aus dynamischen Sets</p>
<pre>ERASE <i>satzname</i> [{ PERMANENT SELECTIVE } MEMBERS] ALL]</pre>	löscht den CRU ggf. mit zugehörigen Membersätzen aus der Datenbank

Tabelle 11: Anweisungen der COBOL-DML

(Teil 1 von 6)

Anweisung	Funktion
<pre> { FIND } { FETCH } } satzauswahlausdruck[RETAINING CURRENCY FOR { MULTIPLE [REALM] [{ SETS { setname,... }] [RECORD]] } </pre>	<p>wählt einen oder mehrere Sätze aus der Datenbank aus, abhängig vom Format des Satzauswahlausdrucks, macht den ausgewählten Satz zum CRU und, wenn keine entsprechende RETAINING-Angabe gemacht wird, zum</p> <ul style="list-style-type: none"> - CRR, - CRS in allen Sets, in denen er Owner oder Member ist, - CRA des Realm, in dem er gespeichert ist. <p>FETCH überträgt den ausgewählten Satz zusätzlich in die UWA des Anwenderprogramms.</p>
<p>Formate des Satzauswahlausdrucks:</p> <p>Format 1:</p> <pre> [satzname] DATABASE-KEY IS feldname [OR { PRIOR { NEXT }] </pre> <p>Format 2:</p> <pre> { ANY { DUPLICATE } } satzname </pre> <p>Format 3:</p> <pre> DUPLICATE WITHIN { satzname { setname } [USING satzelementname,...] </pre>	<p>Zugriff über den Data Base Key</p> <p>Zugriff über einen CALC-Key (Hashverfahren)</p> <p>Zugriff auf einen Satz, der in bestimmten Feldinhalten mit dem CRR bzw. CRS übereinstimmt oder Zugriff auf einen Satz, der einem vorhergehend abgearbeiteten Suchausdruck (FIND/FETCH-7) genügt</p>

Tabelle 11: Anweisungen der COBOL-DML

(Teil 2 von 6)

Anweisung	Funktion
<p>Format 4:</p> <pre> { LAST FIRST NEXT PRIOR ganzzahl feldname } { satzname } [WITHIN { setname } { RECORD } { realmname }] </pre> <p>Format 5:</p> <pre> CURRENT [satzname] [WITHIN { setname } { realmname }] </pre> <p>Format 6:</p> <pre> OWNER WITHIN setname </pre>	<p>Zugriff auf den letzten oder ersten Satz, auf den Nachfolger oder Vorgänger des CRR, CRS bzw. CRA oder auf einen Satz, dessen Position einem anzugebenden Zahlenwert entspricht innerhalb einer Auswahlmenge. Die Auswahlmenge kann eine Satzart, eine Set-Occurrence, ein Realm oder die Durchschnittsmenge einer Satzart mit einem Realm sein.</p> <p>Zugriff auf den CRR, CRS, CRA bzw. CRU</p> <p>Zugriff auf den Ownersatz eines CRS</p>

Tabelle 11: Anweisungen der COBOL-DML

(Teil 3 von 6)

Anweisung	Funktion
<p>Format 7:</p> <p><i>satzenname</i> [<u>WITHIN</u> <i>setname-1</i> [<u>CURRENT</u>]]</p> <p> $\left\{ \begin{array}{l} \left[\text{USING } \textit{satzelementname-1}, \dots \left[\text{OR } \left\{ \begin{array}{l} \text{PRIOR} \\ \text{NEXT} \end{array} \right\} \right] \right. \\ \left[\text{USING } \textit{suchausdruck} \right] \\ \left[\text{RESULT IN } \textit{setname-2} \right] \\ \left[\text{LIMITED BY } \textit{setname-3} \right] \\ \left[\text{TALLYING } \textit{feldname-1} \right] \\ \\ \left[\text{SORTED} \left\{ \begin{array}{l} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{BY} \\ \text{ON} \end{array} \right\} \right] \\ \\ \textit{satzelementname-2} [[,] \textit{satzelementname-3}] \dots \\ \\ \left[[.] \left\{ \begin{array}{l} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} \text{BY} \\ \text{ON} \end{array} \right\} \right] \\ \\ \textit{satzelementname-4} [[,] \\ \textit{satzelementname-5}] \dots] \dots \end{array} \right\}$ </p> <p><i>suchausdruck</i> ::= $\left\{ \begin{array}{l} \textit{komplex-1} \left[\text{AND } \textit{komplex-2} \right] \\ \textit{komplex-2} \end{array} \right\}$</p> <p><i>komplex-1</i> ::= [<u>NOT</u>] <i>bedingung-1</i></p> <p>$\left[\left\{ \begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right\} \left[\text{NOT} \right] \textit{bedingung-1} \right] \dots$</p> <p><i>komplex-2</i> ::= <i>bedingung-2</i> [<u>AND</u> <i>bedingung-2</i>] ...</p> <p><i>bedingung-1</i> ::= <i>satzelementname-6</i> [<u>WITH MASK</u> <i>maske</i>]</p> <p> $\text{IS } \left[\text{NOT} \right] \left\{ \begin{array}{l} \text{EQUAL} \\ = \\ \text{GREATER THAN} \\ > \\ \text{LESS THAN} \\ < \end{array} \right\} \left\{ \begin{array}{l} \textit{feldname-2} \\ \textit{literal-1} \end{array} \right\}$ </p> <p><i>bedingung-2</i> ::= <i>satzelementname-7</i> IS <u>NEXT</u></p> <p> $\left[\text{NOT} \right] \left\{ \begin{array}{l} \text{GREATER THAN} \\ > \\ \text{LESS THAN} \\ < \end{array} \right\} \left\{ \begin{array}{l} \textit{feldname-3} \\ \textit{literal-2} \end{array} \right\}$ </p>	<p>Zugriff auf Sätze über beliebige Felder, ggf. Zählen und Zwischenspeichern der Treffersätze und Suchen mit Maske</p>

Tabelle 11: Anweisungen der COBOL-DML

(Teil 4 von 6)

Anweisung	Funktion
<u>FINISH</u> [WITH <u>CANCEL</u>]	beendet eine Transaktion und gibt gesperrte Realms und Seiten frei.
<u>FREE</u> [<u>ALL</u>]	beendet die Wirkung des KEEP-Status.
<u>GET</u> { <i>satzname</i> { <i>satzelementname</i> ,... } }	stellt den CRU oder einzelne Felder des CRU im Satzbereich der UWA zur Verfügung.
<p>Format 1:</p> <p><u>IF</u> [<u>NOI</u>] [<i>setname</i>] { <u>OWNER</u> { <u>MEMBER</u> { <u>TENANT</u> } } }</p> <p>{ <i>anweisung-1</i> } [<u>ELSE</u> { <i>anweisung-2</i> }]₊ { <u>NEXT SENTENCE</u> } [<u>NEXT SENTENCE</u>]</p> <p>Format 2:</p> <p><u>IF</u> <i>setname</i> IS [<u>NOI</u>] <u>EMPTY</u></p> <p>{ <i>anweisung-1</i> } [<u>ELSE</u> { <i>anweisung-2</i> }]₊ { <u>NEXT SENTENCE</u> } [<u>NEXT SENTENCE</u>]</p>	prüft im Programm Set-Mitgliedschaften.
<u>KEEP</u>	schützt den CRU vor dem Zugriff durch andere Transaktionen bis zu einer FREE-Anweisung oder dem Ende der Transaktion.
<p><u>MODIFY</u> { <i>satzname</i> { <i>satzelementname</i>,... } }</p> <p>[{ <u>INCLUDING</u> } { <u>ALL</u> { <i>setname-1</i>,... } } MEMBERSHIP]</p> <p>[<u>ONLY</u>]</p> <p>[<u>RETAINING CURRENCY FOR</u> { <u>SETS</u> { <i>setname-2</i>,... } }]</p>	ändert Feldinhalte des CRU oder hängt ihn innerhalb eines Set in eine andere Set-Occurrence um.
<p><u>READY</u> [<i>realmname</i>,...]</p> <p>[<u>USAGE-MODE</u> IS [{ <u>EXCLUSIVE</u> }] { <u>RETRIEVAL</u> }] { <u>PROTECTED</u> }] { <u>UPDATE</u> }]</p>	eröffnet eine Transaktion oder eine Verarbeitungskette.
<u>SET</u> <i>feldname-1</i> ,... <u>TO</u> <i>feldname-2</i>	überträgt den Inhalt eines Data Base Key-Feldes in ein oder mehrere Data Base Key-Felder.

Tabelle 11: Anweisungen der COBOL-DML

Anweisung	Funktion
<pre>STORE <i>satzname</i> [RETAINING CURRENCY FOR { MULTIPLE { [REALM] [SETS { <i>setname</i>, ... }] [RECORD] }]]</pre>	<p>überträgt einen Satz aus der UWA als neuen Satz in die Datenbank.</p> <p>fügt den neuen Satz in alle Sets ein, für die seine Satzart im Schema als AUTOMATIC Member definiert ist.</p> <p>richtet eine neue Set-Occurrence für jeden Set ein, für den die Satzart im Schema als Ownersatzart definiert ist.</p>
<pre>USE FOR DATABASE-EXCEPTION [ON { OTHER { <i>literal</i>, ... }]]</pre>	<p>definiert Befehlsfolgen, die durchlaufen werden, wenn eine DML-Anweisung mit einem Datenbanksonderzustand endet.</p>

Tabelle 11: Anweisungen der COBOL-DML

(Teil 6 von 6)

5.3 Erzeugen eines Anwenderprogramms

COBOL-Programm übersetzen

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.2)

- Zuweisung über LINK-NAME=UDSCOSSD
Dieses Verfahren wird nur vom COBOL2000-Compiler ab Version V1.4 unterstützt.
Dem COBOL-Compiler wird die COSSD-Datei explizit zugewiesen mit dem Kommando

```
/ADD-FILE-LINK      LINK-NAME=UDSCOSSD, -
/                   FILE-NAME=[:catid:][$userid.]dbname.COSSD
```

Dabei sind `:catid:` und `$userid` die Katalogkennung und Benutzerkennung, unter der die COSSD-Datei katalogisiert ist. Ohne die Angabe `:catid:` bzw. `$userid` wird der Dateiname nach den Standardregeln des BS2000 komplettiert.

- Zuweisung über LINK-NAME=DATABASE
Dieses Verfahren wird von allen COBOL2000- und COBOL85-Compilern unterstützt.

Dem COBOL-Compiler wird der Datenbankname mitgeteilt mit dem Kommando

```
/SET-FILE-LINK      LINK-NAME=DATABASE, -
/                   FILE-NAME=[:catid:][$userid.]dbname
```

Die Angabe einer `:catid:` beim Kommando SET-FILE-LINK wird ignoriert. Der COBOL-Compiler sucht dann eine COSSD-Datei mit dem Namen `dbname.COSSD` in allen Katalogen, die lokal von derjenigen Benutzerkennung aus zugreifbar sind, die beim Kommando SET-FILE-LINK explizit angegeben wurde oder vom BS2000 ergänzt wurde.

Binden

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.4.2)

```
/START-BINDER
//START-LLM-CREATION INTERNAL-NAME=modul
//INCLUDE-MODULES MODULE-CONTAINER=*LIB(LIBRARY=bibliothek-1
, ELEMENT=element)
//INCLUDE-MODULES MODULE-CONTAINER=*LIB(LIBRARY=udssyslnklib
, ELEMENT=UDSLNKx) _____ (1)
//RESOLVE-BY-AUTOLINK LIBRARY=crtesyslnk
//SAVE-LLM MODULE-CONTAINER=*LIB(LIBRARY=bibliothek-2, ELEMENT=modul)
//END
```

- (1) UDSLNKI: independent DBH
UDSLNKL: linked-in DBH
UDSLNKA: freie DBH-Auswahl

COBOL-Programm starten

Sie können folgende DBH-Varianten benutzen:

bei	independent DBH	linked-in DBH
Mono-DB	X	X
Multi-DB	X	X
openUTM	X	-

Tabelle 12: DBH-Varianten

- Starten eines Anwenderprogramms mit independent DBH

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.4.3)

```
[/MODIFY-JOB-SWITCHES ON=28]
```

```
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version
[/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D,VERSION=version]
```

```
/SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME={
    dbname
    }
    {
    konfigurationsname
    }
```

```
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIBRARY=bibliothek-2,ELEMENT=modul)
,DBL-PAR=(ERROR-PROC(NAME-COLLISION=*STD))
```

```
[Anwenderprogramm-Parameter]
```

- Starten eines Anwenderprogramms mit linked-in DBH

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.4.3 und Handbuch „[Datenbankbetrieb](#)“, Abschnitt 2.3.1)

```
[/MODIFY-JOB-SWITCHES OFF=28]

/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version

/SET-FILE-LINK LINK-NAME=DATABASE,FILE-NAME={ dbname
                                              }
                                              { konfigurationsname }

[/ADD-FILE-LINK LINK-NAME=$UDSKONF,FILE-NAME=UDSTRTAB-bibliothek]

/CREATE-FILE FILE-NAME=konfname.DBSTAT,SUPPRESS-ERR=*FILE-EXISTING
/CREATE-FILE FILE-NAME=konfname.DBSTAT.SAVE,SUPPRESS-ERR=*FILE-EXISTING

[/ADD-FILE-LINK LINK-NAME=PPFILE,FILE-NAME=DBH-parameterdatei]

[weitere DBH-spezifische Kommandos]

/START-EXECUTABLE-PROGRAM FROM-FILE=(LIBRARY=bibliothek-2,ELEMENT=modul)
,DBL-PAR=(ERROR-PROC(NAME-COLLISION=*STD))
[Anwenderprogramm-Parameter]
```

6 CALL-DML

6.1 Parameterdefinitionen

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 8.2)

Die CALL-DML kennt folgende Parameter:

1. Funktionsname: FCOD (function code)
2. Funktionswahl: FOPT (function option)
3. Zusatzwahl: SOPT (special option)
4. Benutzerinformation: UINF (user information)
5. Satzname: RECN (record name)
6. Setname: SETN (set name)
7. Realm-Name: RLMN (realm name)
8. Feldname: ITMN (item name)
9. Satzbereich: RECA (record area)
10. Spezialparameter-1: SPP1 (special parameter 1)
11. Spezialparameter-2: SPP2 (special parameter 2)
12. Spezialparameter-3: SPP3 (special parameter 3)

6.2 Format-Tabelle

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 8.2.2)

Parameter	Inhalt	Länge bei Variante		Format
		(CALL8)	(CALL30)	
1. FCOD	-	6		Schlüsselwort
2. FOPT	-	6		Schlüsselwort
3. SOPT	RET, VAR	6/≥2		formatgebunden/formatfrei, siehe Handbuch „ Anwendungen programmieren “, Abschnitt 8.2.3.
	RET, RES, LMS, TAL, SOA/SOD, NOW	12/≥2		
4. UINF	DBH-Kommunikation	126		siehe Handbuch „ Anwendungen programmieren “, Abschnitt 8.2.4.
5. RECN	Satzname	8	30	Einzelnamenformat
	Satzname oder	8	30	Einzelnamenformat oder Leerzeichen
	RECORD	8□□□□	30	RECORD.../RECORD.....
6. SETN	Setname	8	30	Einzelnamenformat
	1-25 Setnamen	10-226	32-776	Namensleistenformat
7. RLMN	Realm-Name	8	30	Einzelnamenformat
	Feldname bei der Zusatzwahl SOA oder SOD	8	30	Einzelnamenformat
	1-25 Realm-Namen	10-226	32-776	Namensleistenformat
	1-25 Feldnamen bei der Zusatzwahl SOA oder SOD	10-226	32-776	Namensleistenformat
8. ITMN	Feldname	8	30	Einzelnamenformat
	1-25 Feldnamen	10-226	32-776	Namensleistenformat
	Suchausdruck	-		siehe Handbuch „ Anwendungen programmieren “, Abschnitt 8.3.2.
9. RECA ¹	vollständiger Satz	-		Subschemaformat
	Auswahl von Feldern ohne Angabe von VAR	-		Feldpositionen im Satzbereich wie im Subschemaformat
	Auswahl von Feldern mit Angabe von VAR	-		Satzformat entspricht der Feldnamenleiste, Länge = Summe der Feldlängen

Tabelle 13: Formate der CALL-DML

(Teil 1 von 2)

Parameter	Inhalt	Länge bei Variante		Format
		(CALL8)	(CALL30)	
10. SPP1	RETAINING ohne Setnamen	9		siehe Handbuch „ Anwendungen programmieren “, Abschnitt 8.2.5.
	RETAINING mit Setnamen	17-235	39-785	
	Subschema (READYC)	30		Subschemaname in voller Länge mit Leerzeichen ergänzt
11. SPP2	Ganzzahl (FIND4/FTCH4)	4		binäre Ganzzahl $\neq 0$
	Ergebnissetname (FIND7A/FTCH7A)	8	30	Einzelnamenformat
	implizit def. Daten: DB-Key der LOCATION MODE-Klausel und/oder Realm-Name der WITHIN- Klausel (STORE1/STOR1L, STORE2/STOR2L, FIND2/FTCH2)	42		siehe Handbuch „ Anwendungen programmieren “, Abschnitt 8.2.6.
12. SPP3	Begrenzungsetname (FIND7A/FTCH7A)	8	30	Einzelnamenformat

Tabelle 13: Formate der CALL-DML

(Teil 2 von 2)

1 Wenn der Satzbereich zur Rückgabe von Daten verwendet wird, wird er in der Regel auch dann überschrieben, wenn die DML-Funktion FTCH oder GET mit Fehler abgebrochen wurde (Status $\neq 000$). Wenn der Statuscode jedoch mit C, P oder S beginnt, bleibt der Satzbereich unverändert.

6.3 Der Benutzerinformationsbereich

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 8.2.4, Abschnitt 8.2.5)

Inhalt	Eingabe	Ausgabe	Länge	Distanz	Typ
System Communication Locations					
- Realm-Name		X	30	0	Character
- Satzname		X	30	30	Character
- Setname		X	30	60	Character
Ergebnisfelder					
- Anweisungscode		X	2	90	Character
- Statuscode		X	3	92	Character
Leerfeld			1	95	binär
DATABASE-KEY	X	X	4	96	binär
Zähler		X	4	100	binär
Leerfeld			7	104	binär
DB-Kennzeichen	X	X	1	111	binär
DATABASE-KEY-LONG	X	X	8	112	binär
Benutzerinformationseende	X		6	120	Character: USINF*/ UINF1*

Tabelle 14: Benutzerinformationsbereich

6.4 Übersicht über die CALL-DML-Funktionen

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 8.3.1)

Die grauen Hinterlegungen in der Übersicht bedeuten:

Wenn Sie die CALL-DML-Funktion ausgeführt haben, sind hier Informationen abgelegt.

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. RECN	6. SETN	7. RLMN	8. ITMN	9. RECA
ACCPTC	DB-KEY DBKREC DBKRLM DBKSET RLMNAM RLMREC RLMSET RLMDBK		DB-Key DB-Key DB-Key DB-Key Realmname Realmname Realmname DB-Key, Realmname	- Satzname - - Satzname - -	- - Setname - Setname -	- - Realmname - - -		
ACCPTL	DB-KEY DBKREC DBKRLM DBKSET RLMNAM RLMREC RLMSET RLMDBK		DB-Key DB-Key DB-Key DB-Key Realmname Realmname Realmname DB-Key, Realmname	- Satzname - - Satzname - -	- - Setname - Setname -	- - Realmname - - -		
CONNEC	TO-ALL TO-SET	} [RET]		[Satzname] [Satzname]	- Setname...			
DISCON	FRMALL FRMSET ALLFRM			[Satzname] [Satzname] -	- Setname... Setname...			
ERASEC	CORUNT PERMAN SELTIV ALLMEM			} Satzname }				
FINISC	ALLRLM ALLCAN							
FREEC	ALLREC CORUNT							

10. SPP1	11. SPP2	12. SPP3	Funktion
			<p>überträgt</p> <ul style="list-style-type: none"> – den DATABASE-KEY-Wert des CRR, CRS, CRA bzw. CRU in den Benutzerinformationsbereich (siehe Handbuch „Anwendungen programmieren“, Abschnitt 8.2.4). – den Namen des Realm, in dem der CRR, CRS, CRU bzw. der zu einem angegebenen Database-Key-Wert gehörige Satz gespeichert ist.
			<p>überträgt</p> <ul style="list-style-type: none"> – den DATABASE-KEY-LONG-Wert des CRR, CRS, CRA bzw. CRU in den Benutzerinformationsbereich (siehe Handbuch „Anwendungen programmieren“, Abschnitt 8.2.4). – den Namen des Realm, in dem der CRR, CRS, CRU bzw. der zu einem angegebenen Database-Key-Wert gehörige Satz gespeichert ist.
<p>bei RET:</p> <pre>{ SET STNSetName.. STESetname.. }</pre>			hängt den CRU in Set-Occurrences ein
			<ul style="list-style-type: none"> – löst den CRU aus Set-Occurrences – entfernt alle Sätze aus dynamischen Sets
			löscht den CRU ggf. mit zugehörigen Membersätzen aus der Datenbank
			beendet eine Transaktion und gibt gesperrte Realms und Seiten frei
			beendet die Wirkung des KEEP-Status

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. RECN	6. SETN	7. RLMN	8. ITMN	9. RECA
FIND1	[DBKPRI] [DBKNXT]	[RET] [NOW]	DB-Key	[Satzname]				
FIND1L	[DBKPRI] [DBKNXT]	[RET] [NOW]	DB-Key	[Satzname]				
FIND2	ANYREC ANYIMP DUPLIC	} [RET] [NOW]		} Satzname				Feldinhalt Feldinhalt -
FIND3	SETNAM SETITM RECNAM RECITM	} [RET] [NOW]		- } Satzname	Setname Setname - -		- Feldname... - Feldname...	
FIND4	SETNXT SETPRI SETFST SETLST SETSPC	} [RET] [NOW]		} {Satzname} [RECORD...]	} Setname			
	RLMNXT RLMPRI RLMFST RLMLST RLMSPC	} [RET] [NOW]		} {Satzname} [RECORD...]		} Realm- name		
	RECNXT RECPRI RECFST RECLST RECSPC	} [RET] [NOW]		} Satzname				
FIND5	CORUNT RECNAM RECSET SETNAM RECRLM RLMNAM	} [RET] [NOW]		- Satzname Satzname - Satzname -	- - Setname Setname - -	- - - - Realmname Realmname		
FIND6		[RET] [NOW]			Setname			

10. SPP1	11. SPP2	12. SPP3	Funktion
Bei RET: { { MULTIPLE [RLM][REC] { SET STNSet- name... STESet- name... } } }			Zugriff über einen Database-Key-Wert vom Typ DATABASE-KEY
			Zugriff über einen Database-Key-Wert vom Typ DATABASE-KEY-LONG
	impl. def. Datenbereich		Zugriff über einen Calc-Key (Hashverfahren)
			Zugriff auf einen Satz, der in bestimmten Feldinhalten mit dem CRR bzw. CRS übereinstimmt oder Zugriff auf einen Satz, der einem vorhergehend abgearbeiteten Suchausdruck (FIND7A/FTCH7A) genügt.
		pos. Ganzzahl	Zugriff auf den letzten oder ersten Satz, auf den Nachfolger oder Vorgänger des CRR, CRS bzw. CRA oder auf einen Satz, dessen Position einem anzugebenden Zahlenwert entspricht innerhalb einer Auswahlmenge.
		pos. Ganzzahl	Die Auswahlmenge kann eine Satzart, eine Set-Occurence, ein Realm oder die Durchschnittsmenge einer Satzart mit einem Realm sein.
		pos. Ganzzahl	
			Zugriff auf den CRR, CRS, CRA bzw. CRU
			Zugriff auf den Ownersatz eines CRS

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. REC N	6. SET N	7. RLMN	8. ITMN	9. RECA
FIND7A	REFST	[RET] [NOW] [RES] [LMS] [TAL] {SOA} [SOD]	bei TAL: Satz- anzahl	Satzname	-	Feldname...		
	SELFST				Setname	Feldname...		
	CURFST	Setname	Feldname...					
	RECSEX	[RET] [NOW] [RES] [LMS] [TAL] {SOA} [SOD]	Satzname	-	Feldname...	Such- ausdruck		
	CURSEX			Setname	Feldname...			
	RECITM	[RET] [NOW]	Satzname	-		Feld- name...	Feld- inhalt	
	RECITN			Setname				
	SELITM			Setname				
	SELITN							
	SELITP							
	CURITM							
	CURITN							
	CURITP							

10. SPP1	11. SPP2	12. SPP3	Funktion
<pre> { MULTIPLE [RLM][REC] { SET STNSet- name... } [STESet- name...] } </pre>	<p>bei RES:</p> <p>Ergebnis- setname</p>	<p>bei LMS:</p> <p>Begren- zungs- setname</p>	<p>Zugriff auf Sätze über beliebige Felder, ggf. Zählen und Zwischenspeichern der Treffer-sätze und Suchen mit Maske</p>

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. RECN	6. SETN	7. RLMN	8. ITMN	9. RECA
FTCH1	[DBKPRI] [DBKNXT]	[RET] [NOW]	DB-Key	[Satzname]				Satzinhalt
FTCH1L	[DBKPRI] [DBKNXT]	[RET] [NOW]	DB-Key	[Satzname]				Satzinhalt
FTCH2	ANYREC ANYIMP DUPLIC	} [RET] [NOW]		} Satzname				Feldinhalt Satzinhalt Feldinhalt Satzinhalt Satzinhalt
FTCH3	SETNAM SETITM RECNAM RECITM	} [RET] [NOW]		} Satzname	Setname Setname - -		- Feldname... Feldname...	} Satz- inhalt
FTCH4	SETNXT SETPRI SETFST SETLST SETSPC	} [RET] [NOW]		} {Satzname} [RECORD...]	} Setname			
	RLMNXT RLMPRI RLMFST RLMLST RLMSPC	} [RET] [NOW]		} {Satzname} [RECORD...]		} Realm- name		
	REC NXT REC PRI REC FST REC LST REC SPC	} [RET] [NOW]		} Satzname				
FTCH5	CORUNT REC NAM REC SET SET NAM REC RLM RLM NAM	} [RET] [NOW]		- Satzname Satzname - Satzname -	- - Setname Setname - -	- - - - Realmname Realmname		
FTCH6		[RET] [NOW]			Setname			

10. SPP1	11. SPP2	12. SPP3	Funktion
<p> { [MULTIPLE] [RLM][REC] { SET STNSet- name... STESet- name... } } </p>			Zugriff über einen Database-Key-Wert vom Typ DATABASE-KEY
			Zugriff über einen Database-Key-Wert vom Typ DATABASE-KEY-LONG
	- impl. def. Datenbereich -		Zugriff über einen Calc-Key (Hashverfahren)
			Zugriff auf einen Satz, der in bestimmten Feldinhalten mit dem CRR bzw. CRS übereinstimmt oder Zugriff auf einen Satz, der einem vorhergehend abgearbeiteten Suchausdruck (FIND7A/FTCH7A) genügt
	- - - pos. Ganzzahl		Zugriff auf den letzten oder ersten Satz, auf den Nachfolger oder Vorgänger des CRR, CRS bzw. CRA oder auf einen Satz, dessen Position einem anzugebenden Zahlenwert entspricht innerhalb einer Auswahlmenge.
	- - - pos. Ganzzahl		Die Auswahlmenge kann eine Satzart, eine Set-Occurrence, ein Realm oder die Durchschnittsmenge einer Satzart mit einem Realm sein.
	- - - pos. Ganzzahl		
			Zugriff auf den CRR, CRS, CRA bzw. CRU
			Zugriff auf den Ownersatz eines CRS

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. REC N	6. SET N	7. RLMN	8. ITMN	9. RECA
FTCH7A	REFST	[RET] [NOW] [RES] [LMS] [TAL] [SOA] [SOD]	bei TAL: Satz- anzahl	Satzname	-	Feldname...		Satz- inhalt
	SELFST				Setname	Feldname...		
	CURFST	Setname	Feldname...					
	RECSEX	[RET] [NOW] [RES] [LMS] [TAL] [SOA] [SOD]		Satzname	-	Feldname...	Such- ausdruck	
	CURSEX				Setname	Feldname...		
	RECITM	[RET] [NOW]		Satzname	-		Feld- name-1...	Feld- inhalt/ Satz- inhalt
	RECITN							
	SELITM							
	SELITN							
	SELITP							
	CURITM							
	CURITN							
	CURITP							

10. SPP1	11. SPP2	12. SPP3	Funktion
<p> { { { [MULTIPLE [RLM][REC]] { SET STNSet- name... STESet- name... }] } } } ~ </p>	<p>bei RES:</p> <p>Ergebnis- setname</p>	<p>bei LMS:</p> <p>Begren- zungs- setname</p>	<p>Zugriff auf Sätze über beliebige Felder, ggf. Zählen und Zwischenspeichern der Treffersätze und Suchen mit Maske</p>

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. RECN	6. SETN	7. RLMN	8. ITMN	9. RECA
GETC	CORUNT ITMNAM	- [VAR]		[Satzname] Satzname			- Feld- name-1...	Satzinhalt Feldinhalt
IFC	OWNALL OWNSET MEMALL MEMSET TENALL TENSET EMPTY5				- Setname - Setname - Setname Setname			
KEEPC								
MODIF1	CORUNT INCALL ONLALL INCSET ONLSET	} [RET] }		} Satzname }	- - - Setname... Setname...			Satzinhalt Satzinhalt - Satzinhalt -
MODIF2	CORUNT INCALL INCSET	} [RET] } [VAR]		} Satzname }	- - Setname		} Feld- name...	} Feld- inhalt
READYC	ALLRTR ALLLUPD ALLPRT ALLPUP ALLERT ALLEUP RLMRTR RLMUPD RLMPRT RLMPUP RLMERT RLMEUP	} [NOW] }				- - - - - - } Realm- name...		

10. SPP1	11. SPP2	12. SPP3	Funktion
			stellt den CRU oder einzelne Felder des CRU im Satzbereich zur Verfügung
			prüft Set-Mitgliedschaften
			schützt den CRU vor dem Zugriff durch andere Transaktionen bis zu einer FREE-Anweisung oder dem Ende der Transaktion
} bei RET: { SET { STNSetname.. } { SETSetname.. } }			ändert Satzinhalt oder Feldinhalte des CRU und/oder hängt ihn innerhalb eines Set in eine andere Set-Occurrence um
} Subschema- name	} wird nicht mehr benötigt		eröffnet eine Transaktion oder eine Verarbeitungskette

1. FCOD	2. FOPT	3. SOPT	4. UINF	5. RECN	6. SETN	7. RLMN	8. ITMN	9. RECA
STORE1	RECNAM IMPDAT	} [RET]		} Satzname				} Satz- inhalt
STOR1L	RECNAM IMPDAT	} [RET]		} Satzname				} Satz- inhalt
STORE2	ITMNAM IMPDAT	} [RET] [VAR]		} Satzname			} Feld- name...	} Feld- inhalt
STOR2L	ITMNAM IMPDAT	} [RET] [VAR]		} Satzname			} Feld- name...	} Feld- inhalt

10. SPP1	11. SPP2	12. SPP3	Funktion
<pre> bei RET: { MULTIPLE [RLM][REC] { SET STNSet- name... STESet- name... } } </pre>	<pre> - impl. def. Datenbereich - </pre>		<ul style="list-style-type: none"> - überträgt einen Satz oder einzelne Fel- der oder komprimierte Sätze aus der UWA als neuen Satz in die Datenbank - fügt den neuen Satz in alle Sets ein, für die seine Satzart im Schema als AUTOMATIC Member definiert ist. - richtet eine neue Set-Occurrence für je- den Set ein, für den die Satzart im Sche- ma als Ownersatzart definiert ist <p>Wenn Sie bei Funktionswahl IMPDAT einen Database-Key-Wert mit einer REC-REF > 254 und/oder einer RSQ > 2²⁴-1 angeben wollen, müssen Sie STOR1L bzw. STOR2L verwenden.</p>
	<pre> - impl. def. Datenbereich - </pre>		

6.5 LOOKC

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 8.5.1)

Generelle Beschreibung für alle LOOKC-Aufrufe

Bedeutung	Inhalt	Länge	Distanz	Typ
Name des Elements		30 ¹	0	Character
Reference 1		2	30	binär
Reference 2		2	32	binär
Datentyp – ohne Angabe – Realm – Satzart – Set – Feld – Key	(Byte) 00 01 02 03 04 05	1	34	binär
Mehrdeutigkeit – Name ist eindeutig – Name ist mehrfach vorhanden	(Bit) 00 40	1	35	binär
Ergebnis – in Ordnung – nicht gefunden – kein weiteres Element gefunden	(Bit) 00 01/02/08 04	1	36	binär
Reserve		1	37	

Tabelle 15: Generelle Beschreibung im LOOKC-Block

1 auch bei Variante (CALL8)

Spezielle Beschreibung

Die 18 byte lange spezielle Beschreibung ist unterschiedlich aufgebaut, je nachdem, ob Angaben zu einem Realm, Satz, Set, Feld oder Schlüssel gemacht werden.

Spezielle Beschreibung zu Realm im LOOKC-Block

Bedeutung	Inhalt	Länge	Distanz	Typ
Filler		4	0	binär
Realm-Zustand – nicht temporär – temporär	(Bit) ungleich 80 80	1	4	binär
Reserve		13	5	

Tabelle 16: Spezielle Beschreibung zu Realm im LOOKC-Block

Spezielle Beschreibung zu Satzart im LOOKC-Block

Bedeutung	Inhalt	Länge	Distanz	Typ
lange Angabe zur Lage im Satzbereich (Displacement innerhalb des COBOL-BIB)		4	0	binär
Länge der Satzart		2	4	binär
LOCATION MODE – DIRECT – CALC DUPLICATES – CALC NO DUPLICATES – kein LOCATION MODE	(Byte) 00 01 02 03	1	6	binär
Details ¹ – Satzart komprimiert – Satzart mit SEARCH-Key – Satzart in verschiedenen Realms	(Bit) 80 20 10	1	7	binär
kurze Angabe zur Lage im Satzbereich (Displacement innerhalb des COBOL-BIB)		2	8	binär
Reserve		8	10	

Tabelle 17: Spezielle Beschreibung zu Satzart im LOOKC-Block

1 Hier sind Kombinationen möglich

Spezielle Beschreibung zu Set im LOOKC-Block

Bedeutung	Inhalt	Länge	Distanz	Typ
Ownersatzverweis (lang)		2	0	binär
Membersatzverweis (lang)		2	2	binär
Ownersatzverweis (kurz)		1	4	binär
Membersatzverweis (kurz)		1	5	binär
Set-Order – SORTED – FIRST – LAST – NEXT – PRIOR – SORTED INDEXED	(Byte) 00 11 22 44 78 80	1	6	binär
CONNECT-Typ – AUTOMATIC – MANUAL	(Byte) 00 01	1	7	binär
DISCONNECT-Typ – MANDATORY – OPTIONAL	(Byte) 00 01	1	8	binär
SET SELECTION – THRU OWNER – THRU CURRENT	(Byte) 00 01	1	9	binär
Spezielle Typen ¹ – Singulärer Set – DYNAMIC SET – impliziter Set	(Bit) 80 40 20	1	10	binär
Reserve		7	11	

Tabelle 18: Spezielle Beschreibung zu Set im LOOKC-Block

1 Hier sind Kombinationen möglich.

Falls die entsprechenden Bits nicht gesetzt sind: kein Singulärer Set / DYNAMIC SET / impliziter Set

Spezielle Beschreibung zu Feld im LOOKC-Block

Bedeutung	Inhalt	Länge	Distanz	Typ
Filler		2	0	binär
Verbindung Schlüsselfeld-Set: Setverweis (lang) auf den Set, der zum bei Distanz 14 referenzierten Schlüssel gehört.		2	2	binär
Stufennummer		1	4	binär

Tabelle 19: Spezielle Beschreibung zu Feld im LOOKC-Block

(Teil 1 von 2)

Bedeutung	Inhalt	Länge	Distanz	Typ
nächste Stufennummer		1	5	binär
Feldtyp – Database Key – dezimal gepackt – binär – Zeichen – entpackt mit Vorzeichen – entpackt ohne Vorzeichen – Datengruppe – national – nationale Datengruppe	(Byte) 00 01 02 04 05 06 0F 14 1F	1	6	binär
Skalenfaktor Bit 0 = Vorzeichen Bit 1-7 = Wert		1	7	binär
Feldlänge		2	8	binär
Anzahl der Occurrences		2	10	binär
LOCATION MODE IS CALC-Anzeige – einzelner Schlüssel – zusammengesetzter Schlüssel – kein Schlüssel	(Bit) 40 20 00	1	12	binär
SEARCH KEY...USING CALC /SEARCH KEY...USING INDEX- / Sort-Key -Anzeige ¹ – kein Schlüssel – einzelner Schlüssel – zusammengesetzter Schlüssel – Feld in mehreren Schlüsseln verwendet – Wiederholungsgruppenfeld – Feld ist eine Datengruppe, die einen zusammengesetzten Schlüssel bildet	(Bit) 80 40 20 10 01 08	1	13	binär
Verbindung Schlüsselfeld-Set – Verweis auf den ersten Schlüssel, in dem das Feld Schlüsselfeld ist – Setverweis (kurz) auf den zu diesem Schlüssel gehörenden Set		2 1	14 16	binär binär
Zusatzanzeige – Die Zifferanzahl ist gerade (nur relevant für Feldtyp = dezimal gepackt)	(Bit) 80	1	17	binär

Tabelle 19: Spezielle Beschreibung zu Feld im LOOKC-Block

(Teil 2 von 2)

1 Hier sind Kombinationen möglich

Spezielle Beschreibung zu Schlüssel im LOOKC-Block

Bedeutung	Inhalt	Länge	Distanz	Typ
Satzverweis (lang)		2	0	binär
Filler		2	2	binär
Schlüssellänge		1	4	binär
Schlüsseldetails ¹	(Bit)	1	5	binär
– DUPLICATES NOT ALLOWED	80			
– DUPLICATES ALLOWED	00			
– DESCENDING	40			
– ASCENDING	00			
– Indextabelle für Schlüssel	20			
– impliziter Set	10			
– expliziter Set	00			
Anzahl der Felder, die den Schlüssel bilden		1	6	binär
Satzverweis (kurz)		1	7	binär
Position des 1. Schlüsselfeldes im Satz		2	8	binär
Feldlänge		2	10	binär
Feldtyp (vgl. Feldbeschreibung)		1	12	binär
Reserve		5	13	binär

Tabelle 20: Spezielle Beschreibung zu Schlüssel im LOOKC-Block

1 Hier sind Kombinationen möglich

LOOKC-Tabellen

Folgende Abkürzungen werden in der Übersicht verwendet:

I	Input
O	Output
I,O	Output verschieden von Input
AR	Realmreference (binär; Länge: 1 byte)
RR	Recordreference (binär; die Länge der kurzen Referenz beträgt 1 byte, die Länge der langen Referenz beträgt 2 byte).
SR	Setreference (binär; die Länge der kurzen Referenz beträgt 1 byte, die Länge der langen Referenz beträgt 2 byte)
IR	Position des Feldes im Satz (binär; Länge: 2 byte)
K	Keyreference (binär; Länge: 2 byte)

Im Folgenden sind die einzelnen LOOKC-Tabellen jeweils auf zwei gegenüberliegenden Seiten dargestellt.

Bedeutung der Funktion	Funktionswahl	Zusatzwahl	Satzbereich (LOOKC-Block)		
			generelle Beschreibung (38 Bytes)		
			externer Name	Reference 1	Reference 2
	Zeichen	formatfrei	Zeichen	binär	binär
	6		30	2	2

LOOKC nach einem Namen

angegebener Name (des angegebenen Typs)	SPCNAM	()	I	I,O ^{1) 3)}	O ²⁾
erster Name	FSTNAM	()	O	O ¹⁾	O ²⁾
nächster Name	NXTNAM	()	I,O	O ¹⁾	O ²⁾
alle Namen, beginnend mit dem angegebenen	ALLNAM	(SPC)	I,O	I,O ^{1) 3)}	O ²⁾
alle Namen, beginnend mit dem ersten	ALLNAM	(FST)	O	O ¹⁾	O ²⁾
alle Namen, nächster Antwortteil	ALLNAM	(NXA)	O	O ¹⁾	O ²⁾
Owner-/Member-Satznamen eines Sets	OM-NAM	(SET)	O	I(SR)O(RR)	O (Null)
FROM-TO-Namen	FRTNAM	()	I,O	I,O ^{1) 3)}	O ²⁾
FROM-TO-Namen, nächster Antwortteil	FRTNAM	(NXA)	O	O ¹⁾	O ²⁾
Liste der angegebenen Namen	LISNAM	()	I	I,O ^{1) 3)}	O ²⁾

LOOKC nach einem Realm

Liste aller Realms mit dem angegebenen Satztyp	ALLRLM	(REC)	O	O (AR)	
Liste aller Realms mit dem angegebenen Satzart, nächster Antwortteil	ALLRLM	(RECNXA)	O	O (AR)	

- 1) AR bei Datentyp=Realm
RR bei Datentyp=Satzart
SR bei Datentyp=Set
RR bei Datentyp=Feld
RR bei Datentyp=Schlüssel
- 2) NULL bei Datentyp=Realm
NULL bei Datentyp=Satzart
NULL bei Datentyp=Set
IR bei Datentyp=Feld
IR bei Datentyp=Schlüssel
- 3) Input nur bei Datentyp = Feld

				Spezielle Beschreibung (18 Bytes)	Spezial-Parameter-1	Spezial-Parameter-2
Datentyp	gleiche Namen	Ergebnis	Reserve		Zahl der LOOKC-Blöcke	Satz-reference
binär	binär	binär	binär		binär	binär
1	1	1	1		2	4

I,O		O		Nur bei Datentyp=Feld oder Schlüssel; spezielle Feldbeschreibung wird übergeben.		
I,O		O				
I,O		O				
I,O		O			I	
I,O		O			I	
O		O			I	
					I (=2)	
I,O		O			I	
O		O			I	
I,O		O			I	

O		O		O	I	I (RR)
O		O		O	I	

Bedeutung der Funktion	Funktionswahl	Zusatzwahl	Satzbereich (LOOKC-Block)		
			generelle Beschreibung (38 Bytes)		
			externer Name	Reference 1	Reference 2
	Zeichen	formatfrei	Zeichen	binär	binär
	6		30	2	2

LOOKC nach einer Satzart

angegebene Satzarten	SPCREC	()	O	I (RR)	
erste Satzart	FSTREC	()	O	O (RR)	
nächste Satzart	NXTREC	()	O	I,O (RR)	
Ownersatzart des angegebenen Set	OWNREC	(SET)	O	O (RR)	I (SR)
Membersatzart des angegebenen Set	MEMREC	(SET)	O	O (RR)	I (SR)
Liste der angegebenen Satzarten	LISREC	()	O	I (RR)	

LOOKC nach einem Set

angegebener Set	SPCSET	()	O	I (SR)	
erster Set	FSTSET	()	O	O (SR)	
nächster Set	NXTSET	()	O	I,O (SR)	
erster Set der Ownersatzart	FSTSET	(OWNREC)	O	O (SR)	I (RR)
nächster Set der Ownersatzart	NXTSET	(OWNREC)	O	I,O (SR)	I (RR)
erster Set der Membersatzart	FSTSET	(MEMREC)	O	O (SR)	I (RR)
nächster Set der Membersatzart	NXTSET	(MEMREC)	O	I,O (SR)	I (RR)
alle Sets, in denen die angegebene Satzart Owner ist	ALLOWN	(REC)	O	O (SR)	I (RR)
alle Sets, in denen die angegebene Satzart Owner ist, nächster Antwortteil	ALLOWN	(RECXA)	O	O (SR)	
alle Sets, in denen die angegebene Satzart Member ist	ALLMEM	(REC)	O	O (SR)	I (RR)
alle Sets, in denen die angegebene Satzart Member ist, nächster Antwortteil	ALLMEM	(RECXA)	O	O (SR)	
Liste der angegebenen Sets	LISSET	()	O	I (SR)	

				Spezielle Beschreibung (18 Bytes)	Spezial- Parameter-1	Spezial- Parameter-2
Datentyp	gleiche Namen	Ergebnis	Reserve		Zahl der LOOKC-Blöcke	Satz- reference
binär	binär	binär	binär		binär	binär
1	1	1	1		2	4

<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		
<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		
<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		
<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		
<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		
<input type="radio"/>		<input type="radio"/>		<input type="radio"/>		

		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		
		<input type="radio"/>		<input type="radio"/>		

Bedeutung der Funktion	Funktionswahl	Zusatzwahl	Satzbereich (LOOKC-Block)		
			generelle Beschreibung (38 Bytes)		
			externer Name	Reference 1	Reference 2
	Zeichen	formatfrei	Zeichen	binär	binär
	6		30	2	2

LOOKC nach einem Feld

angegebenes Feld der angegebenen Satzart	SPCITM	(REC)	O	I (RR)	I (IR)
erstes Feld der angegebenen Satzart	FSTITM	(REC)	O	I (RR)	O (IR)
nächstes Feld der angegebenen Satzart	NXTITM	(REC)	O	I (RR)	I,O (IR)
alle Felder der angegebenen Satzart	ALLITM	(REC)	O	I (RR)	O (IR)
alle Felder der angegebenen Satzart im nächsten Antwortteil	ALLITM	(RECNXA)	O	O (RR)	O (IR)
alle Felder der angegebenen Zusammenstellung	ALLITM	(AGG)	O	I (RR)	I,O (IR)
alle Felder der angegebenen Zusammenstellung, nächster Antwortteil	ALLITM	(AGGNXA)	O	O (RR)	O (IR)
FROM-TO-Felder	FRTITM	()	O	I (RR)	I,O (IR)
FROM-TO-Felder, nächster Antwortteil	FRTITM	(NXA)	O	O (RR)	O (IR)
Liste der angegebenen Felder	LISITM	()	O	I (RR)	I (IR)

							Spezial-Parameter-1	Spezial-Parameter-2
				Spezielle Beschreibung (18 Bytes)				
Datentyp	gleiche Namen	Ergebnis	Reserve	Filler	Stufennummer	Restliche Beschreibung	Zahl der LOOKC-BI.	Satzreference
binär	binär	binär	binär	binär	binär		binär	binär
1	1	1	1	4	1	13	2	4

0		0	0	0	1,0 ¹	0		
0		0	0	0	0	0		
0		0	0	0	1,0	0		
0		0	0	0	0	0		
0		0	0	0	0	0		
0		0	0	0	1,0	0		
0		0	0	0	0	0		
0		0	0	0	1,0	0		
0		0	0	0	0	0		
0		0	0	0	1,0	0		

1 Falls die angegebene Stufennummer (Input) nicht vorhanden ist, liefert UDS/SQL die nächste gefundene Stufennummer als Output.

Bedeutung der Funktion	Funktionswahl	Zusatzwahl	Satzbereich (LOOKC-Block)		
			generelle Beschreibung (38 Bytes)		
			externer Name	Reference 1	Reference 2
	Zeichen	formatfrei	Zeichen	binär	binär
	6		30	2	2

LOOKC nach einem Schlüssel

erster Schlüssel des angegebenen Set	FSTKEY	(SET)		I (SR)	O (K)
nächster Schlüssel des angegebenen Set	NXTKEY	(SET)		I (SR)	I,O (K)
erster Schlüssel im angegebenen Set, in dem angegebenes Feld vorhanden ist	FSTKEY	(ITMSET)		I (SR)	O (K)
nächster Schlüssel des Feldes im angegebenen Set	NXTKEY	(ITMSET)		I (SR)	I,O (K)
erster Schlüssel des Feldes	FSTKEY	(ITM)		O (SR)	O (K)
nächster Schlüssel des Feldes	NXTKEY	(ITM)		I,O (SR)	I,O (K)
alle Schlüssel des angegebenen Set	ALLKEY	(SET)		I (SR)	O (K)
alle Schlüssel des angegebenen Set; nächster Antwortteil	ALLKEY	(SETNXA)		O (SR)	O (K)
alle Schlüssel des angegebenen Feldes	ALLKEY	(ITM)		O (SR)	O (K)
alle Schlüssel des angegebenen Feldes im angegebenen Set	ALLKEY	(ITMSET)		I (SR)	O (K)
alle Schlüssel des angegebenen Feldes (im angegebenen Set); nächster Antwortteil	ALLKEY	(ITMNXA)		O (SR)	O (K)

									Spezial-Parameter-1	Spezial-Parameter-2
				Spezielle Beschreibung (18 Bytes)						
Datentyp	gleiche Namen	Ergebnis	Reserve	Satzreference (lang)	vgl. Key-Beschreibung	Satzreference (kurz)	Feldreference	vgl. Key-Beschreibung	Zahl der LOOKC-Blöcke	Satzreference
binär	binär	binär	binär	binär		binär	binär		binär	binär
1	1	1	1	2	5	1	2	8	2	4

O		O		O (RR)	O	O (RR)	O (IR)	O		
O		O		O (RR)	O	O (RR)	O (IR)	O		
O		O		I (RR)	O	I (RR)	I (IR)	O		
O		O		I (RR)	O	I (RR)	I (IR)	O		
O		O		I (RR)	O	I (RR)	I (IR)	O		
O		O		I (RR)	O	I (RR)	I (IR)	O		
O		O		O (RR)	O	O (RR)	O (IR)	O	I	
O		O		O (RR)	O	O (RR)	O (IR)	O	I	
O		O		I (RR)	O	I (RR)	I (IR)	O	I	
O		O		I (RR)	O	I (RR)	I (IR)	O	I	
O		O		O (RR)	O	O (RR)	O (IR)	O	I	

Bedeutung der Funktion	Funktionswahl	Zusatzwahl	Satzbereich (LOOKC-Block)		
			generelle Beschreibung (38 Bytes)		
			externer Name	Reference 1	Reference 2
	Zeichen	formatfrei	Zeichen	binär	binär
	6		30	2	2

LOOKC nach den Feldern eines Schlüssels

erstes Feld des angegebenen Schlüssels	FSTITM	(KEY)	O	O (RR), I (SR)	I (K), O (IR)
nächstes Feld des Schlüssels	NXTITM	(KEY)	O	O (RR), I (SR)	I (K), O (IR)
alle Felder des Schlüssels	ALLITM	(KEY)	O	O (RR), I (SR)	I (K), O (IR)
alle Felder des angegebenen Schlüssels; nächster Antwortteil	ALLITM	(KEYNXA)	O	O (RR)	O (IR)

										Spezial-Parameter-1	Spezial-Parameter-2
Spezielle Beschreibung (18 Bytes) bei der											
				Eingabe						Ausgabe	
Datentyp	gleiche Namen	Ergebnis	Reserve	Satzref. (lang)	vgl. Key-Beschreibung	Satzref. (kurz)	Feldreference	vgl. Keybeschreibung	vgl. Feldbeschreibung	Zahl der LOOKC-Blöcke	Satzreference
binär	binär	binär	binär	binär		binär	binär			binär	binär
1	1	1	1	2	5	1	2	8	18	2	4

O		O							O		
O		O		I (RR)		I (RR)	I (IR)		O		
O		O							O	I	
O		O							O	I	

6.6 Assembler-Makros der CALL-DML

(siehe Handbuch „Anwendungen programmieren“, Abschnitt 8.4)

Zur Unterstützung von UDS/SQL-Anwendern, die von Assembler-Programmen aus mit CALL-DML arbeiten, stehen für die Variante (CALL8) folgende Makros zur Verfügung:

Makro	Funktion		Anwendung
	statisch zur Übersetzungszeit	dynamisch zur Ablaufzeit	
DSCAL	Implizite Parameterleiste (im Befehlscode) generieren und vorbesetzen	<ul style="list-style-type: none"> – ggf. implizite Parameterleiste vervollständigen – CALL-DML-Aufruf mit der impliziten Parameterleiste durchführen 	CALL-DML-Aufruf mit impliziter Parameterleiste und komfortablem Parametermechanismus Alternative: DSCAP + DSCDF
DSCAP	-	<ul style="list-style-type: none"> – explizit definierte Parameterleiste übernehmen – ggf. explizit definierte Parameterleiste füllen – CALL-DML-Aufruf mit der expliziten Parameterleiste durchführen 	CALL-DML-Aufruf mit expliziter Parameterleiste und komfortablem Parametermechanismus (wahlfrei)
DSCDF	Parameterleiste (explizit) in den aktuellen CSECT bzw. DSECT generieren und vorbesetzen	-	CALL-DML-Parameterleiste (explizit) generieren
DSCPA	Benutzerinformationsbereich und/oder vordefinierte CALL-DML-Parameterkonstanten generieren	-	Unterstützung der ASSEMBLER-Programmierung an der CALL-DML-Daten-Schnittstelle

Tabelle 21: CALL-DML-Assembler-Makros

`[name] DSCAL fcod,fopt,...,spp3`

`[name] DSCAP [fcod,fopt,...,spp3][,PARAM=param]`

`[name] DSCDF fcod,fopt,...,spp3[,SUFFIX=x]`

`[name] DSCPA [option]`

6.7 Erzeugen eines Anwenderprogramms

Übersetzen des Subschemas mit BCALLSI

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.5)

Übersetzen des Anwenderprogramms

CALL-DML-Anwendung in COBOL:

(siehe [Abschnitt „Erzeugen eines Anwenderprogramms“ auf Seite 53](#))

```
/START-COBOL2000-COMPILER -  
/SOURCE=cobolsource, -  
/COMPILER-ACTION=MODULE-GENERATION(MODULE-FORMAT=LLM), -  
/MODULE-OUTPUT=*LIBRARY(LIBRARY=bibliothek-1,ELEMENT=element)
```

CALL-DML-Anwendung in anderen Programmiersprachen:

(siehe Thema „Übersetzen“ im jeweiligen Compiler-Benutzerhandbuch)

Binden

CALL-DML-Anwendung in COBOL:

(siehe [Abschnitt „Erzeugen eines Anwenderprogramms“ auf Seite 53](#))

CALL-DML-Anwendung in anderen Programmiersprachen:

analog zum Binden von COBOL-Anwendungen; zum Binden der sprachspezifischen Laufzeitsysteme siehe Beispiele im jeweiligen Compiler-Benutzerhandbuch

Starten

- Starten eines Anwenderprogramms mit independent DBH

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.4.3)

```
[/MODIFY-JOB-SWITCHES ON=28]
```

```
  /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version
```

```
[/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D,VERSION=version]
```

```
  /SET-FILE-LINK LINK-NAME=DATABASE, FILE-NAME={ dbname
                                                }
                                                { konfigurationsname }
```

```
  /ADD-FILE-LINK LINK-NAME=$UDSSSI, FILE-NAME=SSITAB-bibliothek-1
```

```
[/ADD-FILE-LINK LINK-NAME=BLSLIBnn, FILE-NAME=SSITAB-bibliothek-nn]
```

```
[/ADD-FILE-LINK LINK-NAME=$UDSPLEX, FILE-NAME=PLITAB-bibliothek-1]
```

```
[/ADD-FILE-LINK LINK-NAME=BLSLIBnn, FILE-NAME=PLITAB-bibliothek-nn]
```

```
  /START-EXECUTABLE-PROGRAM FROM-FILE=(LIBRARY=bibliothek-2,ELEMENT=modul)
    ,DBL-PAR=(ERROR-PROC(NAME-COLLISION=*STD))
```

```
[Anwenderprogramm-Parameter]
```

7 Anweisungscode und Statuscode der COBOL-DML und CALL-DML

7.1 Anweisungscodes

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.6.2)

Bei COBOL-DML finden Sie den Anweisungscode im Sonderregister DATABASE-STATUS. Bei CALL-DML wird das Ergebnisfeld des Parameters Benutzerinformation in den ersten zwei Stellen mit dem Anweisungscode belegt.

Anweisungscode	Anweisungen der	
	COBOL-DML	CALL-DML
01	CONNECT	CONNEC
02	DISCONNECT	DISCON
03	ERASE	ERASEC
04	FIND/FETCH	{ FIND1/FTCH1, FIND1L/FTCH1L FIND2/FTCH2 FIND3/FTCH3 FIND4/FTCH4 FIND5/FTCH5 FIND6/FTCH6 FIND7A/FTCH7A
05	FINISH	FINISC
06	FREE	FREEC
07	GET	GETC
08	IF	IFC
09	KEEP	KEEPC
10	MODIFY	{ MODIF1 MODIF2
12	READY	READYC
14	STORE	{ STORE1, STOR1L STORE2, STOR2L
15	ACCEPT	ACCPTC, ACCPTL
16	SET	–
25	–	LOOKC
00	Bei allen vom Verbindungsmodul vergebenen Sonderzuständen	

Tabelle 22: Zuordnung der Anweisungscodes zu den Funktionen

Die UDS-Online-Utility (siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“) verwendet für ihre spezifischen DMLs den Anweisungscode 13.

7.2 Auswirkungen des Statuscodes

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 6.6.2)

Die folgende Tabelle zeigt Ihnen das Verhalten von UDS/SQL bei den verschiedenen Statuscodes:

	Statuscodes					
	000	001	018, 113, 122, 218	≠ 000 bei READY	200	alle anderen
DML-Anweisung erfolgreich gewesen	ja	ja	nein	nein	vorläufig	nein
UDS/SQL bricht die Transaktion ab	nein ¹⁾	nein ¹⁾	ja	ja	nein	nein ¹⁾
der Inhalt des Satz- bereichs der UWA ist ...	entspre- chend der Anweisung	der nächste Satz	undefiniert, weil die Transaktion abgebro- chen wurde		unverän- dert	unverän- dert

Tabelle 23: Folgen der einzelnen Statuscodes

¹⁾ außer Sie führen einen FINISH WITH CANCEL durch

7.3 Kombinationen von Anweisungs-codes und Status-codes

(siehe Handbuch „Anwendungen programmieren“, Abschnitt 6.6.2)

Statuscode	Anweisungscode																	
	00	01	02	03	04	05	06	07	08	09	10	12	13	14	15	16	25	
001 010					siehe Tabelle 25 auf Seite 100								X					
011 012 013 018 020		X	X	X				X	X	X	X	X	X	X	X	X		X
021 022 023 024 027 028 029		X									X		X		X			
031 032 033		X	X	X				X	X		X				X			
042 043 044		X	X	X						X		X	X	X	X	X		
051		X									X				X			
071 072				X														
081 082 083		X	X								X							
091 092 093 099		X	X	X					X		X		X	X	X	X		
101 102 103							X										X	X
113	X	X	X	X			X		X	X		X	X	X	X	X		X

Tabelle 24: Kombination von Anweisungs-codes und Status-codes

(Teil 1 von 3)

Statuscode	Anweisungscode																	
	00	01	02	03	04	05	06	07	08	09	10	12	13	14	15	16	25	
122	X	X	X	X	siehe Tabelle 25 auf Seite 100	X	X	X	X	X	X	X	X	X	X		X	
123												X	X					
124												X	X					
131	X												X	X				
132													X	X				
134	X	X	X	X			X	X	X	X	X		X	X	X	X		X
136	X	X	X	X				X	X	X	X		X	X	X	X		X
137	X												X	X				
141													X	X				
142													X	X				
144	X	X	X	X				X	X	X	X		X	X	X	X		
145													X	X				
146	X	X	X	X				X	X	X	X		X	X	X	X		
151	X												X	X				
152	X												X	X				
154	X												X	X				
155	X												X	X				
161													X	X				
162													X	X				
163													X	X				
164												X	X					
165												X	X					
166												X	X					
167												X	X					
183																		
184																		
191																		
192																		
193																		
194																		
195																		
197																		
198																		
200	X					X												
201	X					X												
218	X	X	X	X			X	X	X	X	X	X		X	X		X	

Tabelle 24: Kombination von AnweisungsCodes und Statuscodes

(Teil 2 von 3)

Statuscode	Anweisungscode																	
	00	01	02	03	04	05	06	07	08	09	10	12	13	14	15	16	25	
781					siehe Tabelle 25 auf Seite 100								X				X	
782														X				X
783														X				X
784														X				X
785														X				X
786														X				X
789														X				X
802		X										X			X			
804															X			
805		X	X	X						X	X	X	X		X			
888										X				X				
898										X				X				
899							X			X				X				
901		X	X	X				X	X		X	X	X	X				
950											X	X						
954											X	X						

Tabelle 24: Kombination von Anweisungs-codes und Statuscodes

(Teil 3 von 3)

Statuscodes bei FIND/FETCH

Statuscode	Formate des FIND/FETCH-Satzauswahlausdrucks						
	1	2	3	4	5	6	7
001							X
018	X	X	X	X	X	X	X
020	X	X	X	X	X	X	X
021		X	X	X			
023							X
024	X	X		X			X
027			X				X
028	X						
029				X	X		
031		X	X	X	X	X	X
032					X		
033					X		
042	X		X	X			X
043		X					X
071		X	X	X	X		

Tabelle 25: Kombination von Anweisungscode 04 und Statuscodes

(Teil 1 von 2)

Statuscode	Formate des FIND/FETCH-Satzauswahlausdrucks						
	1	2	3	4	5	6	7
04...							
091	X	X	X	X	X	X	X
101				X		X	X
102	X						
103	X	X	X	X	X	X	X
113	X	X	X	X	X	X	X
122	X	X	X	X	X	X	X
134	X	X	X	X	X	X	X
136	X	X	X	X	X	X	X
144	X	X	X	X	X	X	X
146	X	X	X	X	X	X	X
183							X
184			X				X
191							X
192							X
193			X	X			X
194							X
195							X
197			X				
198			X				
218	X	X	X	X	X	X	X
805							X
901	X	X	X	X	X	X	X

Tabelle 25: Kombination von Anweisungscode 04 und Statuscodes

(Teil 2 von 2)

7.4 Statuscodes der DML

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 10.1)

Statuscode als Hinweis

- 001 Bei einem FIND/FETCH Format 1 oder 7 mit OR PRIOR/OR NEXT-Angabe: Es wurde kein Satz gefunden, der mit den vorgegebenen Werten übereinstimmt. Der nächste Satz in der Sortierfolge wurde zur Verfügung gestellt.

Statuscodes mit Fortschrittshinweisen der Online-Utility

- 010 RELOCATE DML: Quell- und Zielpegel sind gleich. Die Verlagerung ist beendet.
REORGPPP DML: Ende der Realms erreicht. Die Reorganisation ist beendet.
- 011 RELOCATE DML: Quell- und Zielpegel sind 0 bei INITIALIZE=*NO.
REORGPPP DML: Die derzeitige Seitenzahl ist 0 bei INITIALIZE=*NO.
Bei einer versuchten Fortsetzung von Verlagerungen mit INITIALIZE=*NO wird festgestellt, dass keine Informationen mehr vorliegen, z.B. weil die Datenbank zwischenzeitlich ausgehängt worden ist oder ein neuer Sessionabschnitt begonnen wurde.
- 012 RELOCATE DML: Beim Lesen einer Quellseite ist ein Sperrkonflikt mit einer parallelen Transaktion aufgetreten.
REORGPPP DML: Beim Lesen einer Seite ist ein Sperrkonflikt mit einer parallelen Transaktion aufgetreten.
- 013 Beim Lesen einer Zielseite ist ein Sperrkonflikt mit einer parallelen Transaktion aufgetreten.

Statuscodes zur Datenkonsistenz

- 018 Deadlock-Zustand (gegenseitiges Sperren mehrerer Transaktionen auf UDS/SQL-Betriebsmitteln);
FINISH WITH CANCEL wird ausgeführt. Es ist sinnvoll, die Transaktionen zu wiederholen (begrenzt).
Für UDS-D:
Im UDS/SQL-Betrieb ohne openUTM erfolgt die globale Deadlock-Erkennung über eine Zeitüberwachung (PP DEADTIME) von Wartesituationen. Nach Ablauf dieses Zeitlimits wird der Statuscode 018 angezeigt, auch wenn u. U. kein wirklicher Deadlock vorliegt.

- 020 FIND/FETCH (nur CALL-DML)
Eine Seite, auf die zugegriffen werden soll, ist von einer anderen Transaktion gesperrt.

Statuscodes zur Satz-Wiedergewinnung

- 021 Das Ende einer Satzart, eines Sets oder Realms wurde erreicht.
FIND/FETCH Formate 2 (DUPLICATE) und 3 (USING):
Es kann kein Satz mit gleichen Werten wie der entsprechende CRR bzw. CRS gefunden werden.
FIND/FETCH Format 3 (ohne USING):
Das Ende der Treffermenge wurde erreicht.
FIND/FETCH Format 4:
Kein nächster (NEXT) oder vorhergehender (PRIOR) Satz kann gefunden werden
oder
ganzzahl bzw. *name* enthält einen Wert, der keinen Satz innerhalb des Realms/der Satzart/der Set-Occurrence adressiert.
- 022 Die Transaktion versucht einen Realm zu eröffnen, der für UPDATE und RETRIEVAL gesperrt ist. Mögliche Ursachen für diese Sperre sind auf Datenbankebene:
- Die Datenbank wurde vom Datenbankadministrator via DAL-ACCESS-Kommando gesperrt.
 - Das DBDIR der Datenbank ist gesperrt (siehe „Realm-Ebene“).
- Realm-Ebene:
- Der Realm wurde im Rahmen einer Datenbankrestrukturierung aus der Datenbank gestrichen.
 - Der Realm wurde vom Datenbankadministrator oder von der UDS/SQL-Fehlerbehandlung abgeschaltet.
 - Der Realm wurde vom Datenbankadministrator via DAL-ACCESS-Kommando gesperrt.
- 023 Nur bei SET OCCURRENCE SELECTION IS THRU LOCATION MODE OF OWNER: Es kann keine den Setauswahlkriterien genügende Set-Occurrence gefunden werden.

- 024 Es kann kein dem Satzauswahlausdruck genügender Satz gefunden werden.
- FIND/FETCH Format 1:
Der Database Key liefert aus einem der folgenden Gründe keinen Treffer:
- Seine Satzartnummer entspricht nicht der explizit angegebenen Satzart.
 - Sein Wert liegt zwar innerhalb der Grenzen seiner DBTT, es existiert aber in der Datenbank kein zugehöriger Satz.
- FIND/FETCH Formate 2 (ANY) und 7:
Es kann kein Satz gefunden werden, der den initiierten Datenelementen bzw. dem Suchausdruck entspricht.
- FIND/FETCH Format 4:
Es kann kein Satz innerhalb der angegebenen Satzart, des Realms oder der Set-Occurrence gefunden werden.
- 027 Die Subskribierung des angegebenen Feldnamens liegt nicht innerhalb des durch die OCCURS-Klausel im Subschema festgelegten Bereiches.
- 028 Der angegebene Database Key enthält eine ungültige Satzartnummer oder eine Satzfolgennummer, die außerhalb der Grenzen seiner DBTT liegt.
- 029 FIND/FETCH Format 4 und 5:
Der Current des Realms bzw. Sets hat nicht die in der Anweisung angegebene Satzart.

Statuscodes zu Currency-Indikatoren

- 031 Der Current des Realms, des Sets oder der Satzart ist nicht bekannt.
- FIND/FETCH Format 3:
Der Current of Set ist Owner und nicht Member des angegebenen Sets oder der angegebene Setname unterscheidet sich von dem im vorausgegangenen FIND7 angegebenen Setnamen.
- FIND/FETCH Format 6 und Format 7:
Der Owner wurde gelöscht.
- IF Format 2:
Der CRS wurde gelöscht oder aus dem angegebenen Set ausgehängt.
- 032 Der Current of Rununit ist nicht bekannt oder wurde gelöscht.
- 033 Der Current of Rununit hat nicht die in der Anweisung angegebene Satzart.

Statuscodes zur Namensgebung

- 042 Satzart, Set oder Realm sind nicht im aufgerufenen Subschema definiert oder ein Feld, das Teil eines ASC-, DESC-, CALC-Keys ist, ist nicht im Subschema definiert oder nach einer Subschemaänderung wurde das Anwenderprogramm nicht neu übersetzt (COBOL-DML) oder der BCALLSI-Lauf vergessen (CALL-DML) oder Fehler an der BIB-Schnittstelle (siehe Statuscode 103) oder bei einer Online-Utility wurde ein Realm angegeben, in dem keine Aktivitäten zulässig sind.
- 043 STORE und FIND/FETCH Format 2:
Das AREA-ID-Datenelement enthält den Namen eines Realm, der nicht in der DDL-WITHIN-Klausel angegeben ist oder nicht zum aufgerufenen Subschema gehört oder bei SET OCCURRENCE SELECTION IS THRU LOCATION MODE OF OWNER und Ownersatzart = LOCATION MODE IS CALC: Das AREA-ID-Datenfeld des Ownersatzes enthält den Namen eines Realm, der nicht in der DDL-WITHIN-Klausel angegeben ist oder nicht zum aufgerufenen Subschema gehört.
- 044 IF:
Die Angabe eines dynamischen Sets ist nicht erlaubt.

Statuscodes zur Eindeutigkeit von Schlüsseln

- 051 Doppeltes Auftreten von Schlüsselwerten in der Datenbank. Das bedeutet, dass die Ausführung einer DML-Anweisung einer DUPLICATES ARE NOT ALLOWED-Angabe in einer ORDER IS SORTED BY DEFINED KEYS-Klausel oder SEARCH-KEY-Klausel eines Sets, in dem der betroffene Satz Member ist, oder der LOCATION MODE IS CALC-Klausel oder SEARCH-KEY-Klausel des betroffenen Satzes widersprechen würde.

Statuscodes zu Satzeigenschaften

- 071 FIND/FETCH Format 2 (DUPLICATE), 3, 4 und 5:
Der Aufsetzpunkt der DML-Anweisung (CRR, CRA bzw. CRS) wurde gelöscht oder aus der aktuellen Set-Occurrence ausgehängt. Im Falle der Abarbeitung einer Treffermenge (FIND3 ohne USING) führen Änderungsoperationen der eigenen Transaktion nicht zum Verlust des Aufsetzpunktes, nur Änderungen fremder Transaktionen.

- 072 ERASE:
Der betroffene Satz ist Owner einer nicht leeren Set-Occurrence und kann daher mit der gewählten ERASE-Variante nicht gelöscht werden.

Statuscodes zur Set-Mitgliedschaft

- 081 CONNECT (setname):
Der CRU ist in einem der spezifizierten Sets bereits Member oder einer der spezifizierten Sets ist kein Member-Set des CRU.
CONNECT (ALL):
Der CRU ist in allen seinen Member-Sets bereits Member.
MODIFY (setname):
Einer der spezifizierten Sets ist kein Member-Set des CRU.
MODIFY (ALL):
Der CRU ist in keinem seiner Member-Sets Member.
- 082 DISCONNECT (setname):
Der CRU ist in einem der spezifizierten Sets MANDATORY Member oder einer der spezifizierten Sets ist kein Member-Set des CRU
DISCONNECT (FROM ALL):
Kein Member-Set des CRU ist OPTIONAL.
- 083 DISCONNECT (setname) und MODIFY (setname):
Der CRU ist in einem der spezifizierten Sets kein Member
DISCONNECT (FROM ALL):
Mindestens ein Member-Set des CRU ist OPTIONAL, aber in keinem dieser OPTIONAL Sets ist der CRU Member.

Statuscodes zum READY-Zustand

- 091 Ein Realm ist nicht im READY-Zustand (d.h. ein Realm wurde beim READY nicht explizit angegeben oder ist nicht Teil des aktuellen Subschemas) oder
bei einem ERASE PERMANENT/SELECTIVE/ALL wurden beim READY Realm-Namen explizit angegeben oder
die DBTT einer zu verlagernden Satzart bei der Online-Utility liegt in einem nicht eröffneten Realm.
- 092 In einer RETRIEVAL-Verarbeitungskette ist keine DML-Anweisung mit Update-Funktion zulässig
oder bei einem ERASE PERMANENT/SELECTIVE/ALL wurde die Verarbeitungskette nicht mit EXCLUSIVE UPDATE eröffnet
oder im Falle des P-Parameters PP TA-ACCESS=SHARED wird versucht, eine Verarbeitungskette in den Benutzungsarten PROTECTED oder EXCLUSIVE zu eröffnen.

- 093 Der Database Handler lässt die Verarbeitungskette nicht zu, da die betreffende Datenbank innerhalb der Transaktion bereits eröffnet ist („Second-READY“ innerhalb einer Verarbeitungskette).
- 099 (nur CALL-DML oder Online-Utility)
Beim Eröffnen einer Transaktion ist ein Realm von einer anderen Transaktion gesperrt.

Statuscodes für fehlerhafte DML-Anweisungen

- 101 FIND/FETCH Format 4:
Es wurde für den Eintrag *ganzzahl* oder *feldname* der Wert Null verwendet oder
bei der Suche in einer nicht rückwärts verketteten CHAIN wurde ein negativer Wert verwendet.
- FIND/FETCH Format 6:
setname darf keinen singulären Set bezeichnen.
- FIND/FETCH Format 7:
- OR PRIOR oder OR NEXT konnte nicht durchgeführt werden, weil kein sortierter und indizierter Schlüssel vorlag.
 - „WITHIN *setname-1* USING *satzelementname-1,...*“ wurde angegeben.
die Angabe eines dynamischen Set in *setname-1* ist nicht erlaubt.
 - LIMITED BY *dynamischer-set* ... SORTED BY ... wurde angegeben.
Das Sortieren des Durchschnitts einer Treffermenge und eines dynamischen Set ist nicht möglich.
 - LIMITED BY *sortierter-dynamischer-set* ... wurde angegeben.
Die Durchschnittbildung einer Treffermenge mit einem sortierten dynamischen Set ist nicht möglich.
- FINISH:
FINISH-Typ (mit oder ohne CANCEL) nicht identifizierbar.
- 102 SET, ACCEPT (Format 1):
Ein großer Database-Key-Wert (Database-Key-Wert mit einer REC-REF > 254 und/oder einer RSQ > 2²⁴-1) kann nicht in ein Feld des Typs USAGE IS DATABASE-KEY übertragen werden. Es ist ein Subschema zu verwenden, in dem nicht
SUBSCHEMA FORM IS OLD angegeben ist und das ab UDS/SQL V2.0 erzeugt wurde, zusätzlich muss das angegebene Feld vom Typ
USAGE IS DATABASE-KEY-LONG sein.

- 103 Fehler an der BIB-Schnittstelle.
Mögliche Ursachen: Falscher COBOL-Compiler oder falsches COBOL-Laufzeitsystem, Fehler im CALL-DML-Konverter, in IQS, in der Online-Utility, oder in einem Dienstprogramm, welches BIBs erzeugt, oder Fehler im Database Handler.

Statuscode zu System-Fehlern

- 113 Beim Zugriff auf eine Datenbankseite wurde ein schwerwiegender Fehler im Database Handler oder in der Datenbank entdeckt.

Statuscodes zu den UDS/SQL-Betriebsmitteln

- 122 Die Transaktion wurde vom DBH vorzeitig mit CANCEL beendet.
Mögliche Ursachen:
- RLOG-Datei zu klein oder zu oft gesplittet.
 - UDS/SQL-Puffer zu klein, PP BUFFERSIZE=n größer wählen.
 - Rücksetzen dieser Transaktion in einer zwischenzeitlich durchgeführten Deadlockauflösung.
 - Eingriff des Datenbankadministrators via DAL (Kommandos ABORT, PERFORM, CLOSE).
 - Neue Update-Transaktionen während des Schreibens eines Checkpoints.
 - Auftreten eines Datei- oder Programmierfehlers, der durch CANCEL der Transaktion (vorläufig) umgangen werden kann.
 - Fehler in einer DML-Anweisung, die nicht für sich rücksetzbar ist und daher den CANCEL der gesamten Transaktion erfordert. Der Datenbankadministrator wurde verständigt (über eine UDS/SQL-Meldung).
 - für UDS-D:
Das Rücksetzen der Transaktion kann auch an Fehlern bzw. Administratoreingriffen in einer entfernten Konfiguration (z.B. ABORT, CLOSE CALLS, CLOSE RUN-UNITS, %TERM) oder an Fehlern in der Verbindung zur entfernten Konfiguration liegen.

- 123 Die Transaktion versucht einen Realm mit READY-USAGE-MODE UPDATE zu öffnen, der für Änderungen gesperrt ist.
Mögliche Ursachen für diese Sperre sind auf Konfigurationsebene:
- Die aktuelle Session des independent DBH wurde ohne RLOG-Logging gestartet (PP LOG=NO).
 - Das Eröffnen der RLOG-Datei ist misslungen, d.h. das RLOG-Logging ist zurzeit blockiert.
- Datenbankebene:
- Die Datenbank ist als SHARED-RETRIEVAL-Datenbank zugeschaltet.
 - Die Datenbank ist keine Original-Datenbank, sondern eine Schattendatenbank.
 - Das Eröffnen einer neuen ALOG-Datei ist misslungen, d.h. das AFIM-Logging der Datenbank ist zurzeit blockiert.
 - Die Datenbank wurde vom Datenbankadministrator via DAL-ACCESS-Kommando für Änderungen gesperrt.
 - Das DBDIR der Datenbank ist für Änderungen gesperrt (siehe „Realm-Ebene“).
- Realm-Ebene:
- Der Realm wurde vom Datenbankadministrator via DAL-ACCESS-Kommando für Änderungen gesperrt.
 - Die Transaktion versucht, einen Realm einer entfernten Datenbank zu öffnen, obwohl die aktuelle Session (wegen PP LOG=NO oder misslungenem Eröffnen der RLOG-Datei) ohne RLOG-Logging fährt.
Dadurch würde die Basis für das Zwei-Phasen-Ende-Protokoll verteilter Transaktionen fehlen.
- 124 Die Transaktion wurde vom DBH vorzeitig mit CANCEL zurückgesetzt.
- Ursache:
Neue Update-Transaktion oder Update-Verarbeitungskette während des Schreibens eines Checkpoints oder während des RLOG-Datei-Wechsels.
- Dieser Statuscode wird nur gesetzt, wenn für die aktuelle Session der Ladeparameter PP ORDER-DBSTATUS=SPECIAL angegeben wurde. Andernfalls wird unter den oben genannten Bedingungen der Statuscode 122 gesetzt.
- 131 Der Database Handler lässt die Transaktion nicht zu, da die beim Laden des Database Handler durch den Ladeparameter TRANSACTION angegebene Anzahl parallel zulässiger Transaktionen bzw. Anwendertasks erschöpft ist.

- 132 Der Database Handler lässt die Transaktion nicht zu, da die beim Laden des Database Handler durch den Ladeparameter SUBSCHEMA angegebene Anzahl zulässiger Subschemas erschöpft ist.

Statuscodes zur Reihenfolge der DML-Anweisungen

- 134 Der Database Handler lässt eine DML-Anweisung nicht zu, da keine Transaktion offen ist.
- 136 Eine DML-Anweisung wird abgewiesen, die zwar zu einer existierenden Transaktion gehört, sich aber an eine Datenbank wendet (eine DB-Referenz liefert) für die aktuell keine Verarbeitungskette der Transaktion existiert.
- 137 Das Mischen von SQL- und Nicht-SQL-Anweisungen in einer Transaktion ist unzulässig (Ausnahme: Zugriff auf verschiedene UDS/SQL-Konfigurationen über openUTM). Das Mischen von COBOL-DML- und CALL-DML-Anweisungen in einer Verarbeitungskette ist unzulässig.

Statuscodes zum Subschema

- 141 Die Transaktion hat einen ungültigen bzw. unbekanntem Subschema-Namen angegeben oder der Subschema-Name ist in der aktuellen DB-Konfiguration innerhalb der ersten 6 Zeichen nicht eindeutig oder die betreffende Datenbank ist nicht zugeschaltet.

für UDS-D:

Das angesprochene Subschema ist

- nicht in der lokalen Konfiguration enthalten und nicht in der Verteiltabelle angegeben.
- in der Verteiltabelle angegeben, aber in der entsprechenden UDS/SQL-Konfiguration nicht enthalten.
- in der Verteiltabelle angegeben, aber die entsprechende UDS/SQL-Konfiguration ist nicht erreichbar,
 - a) weil der Rechner nicht erreichbar ist
 - b) weil die Konfiguration gar nicht oder nicht mit eingeschalteter Verteilung läuft.
- in der Verteiltabelle angegeben, aber gesperrt bzw. die zugehörige Datenbank oder Konfiguration ist gesperrt.
- in der lokalen Konfiguration nicht enthalten und in der lokalen Konfiguration wurde UDS-D nicht gestartet.

- Die Anzahl der von dieser Transaktion angesprochenen entfernten Datenbanken überschreitet den Wert PP DISDB.
- 142 Die Subschema-Beschreibung im DBDIR (SSIA) ist zerstört. BGSSIA-Lauf wiederholen.
- 144 Die DML-Anweisung spezifiziert ein zur aktuellen READY-Anweisung unterschiedliches Subschema (Subschema-Referenz).
- 145 Das in der READY-Anweisung angesprochene Subschema kann nicht prozessiert werden, weil es nicht zum aktuellen Stand des Schemas passt (Subschema-DDL-Compilierung und/oder BGSSIA-Lauf nach Datenbankrestrukturierung fehlt) oder die READY-Anweisung wird abgewiesen, weil die UDS/SQL-Version nicht zur Datenbank passt:
- Die Datenbank wurde für die jahrhundertgerechte Bearbeitung zweistelliger Jahresfelder eingestellt bzw. diese Einstellung wurde nicht vorschriftsmäßig entfernt. Daher darf sie nur mit einer Version ab UDS/SQL V2.0B30 bearbeitet werden.
 - Ein Subschema enthält nationale Daten (Unicode: UTF-16, PICTURE N, USAGE NATIONAL). Daher darf es nur mit einer Version ab UDS/SQL V2.5 bearbeitet werden.
- 146 COBOL-DML: Das Subschema, mit dem das Modul der aktuellen DML-Anweisung übersetzt wurde, entspricht nicht dem aktuellen Stand der Datenbank.
CALL-DML: Das verwendete SSITAB-Modul entspricht nicht dem aktuellen Stand der Datenbank.

Statuscodes zur DBH-Verfügbarkeit

- 151 Der Database Handler ist noch nicht verfügbar oder wird normal beendet (Beendigung läuft).
- 152 Der Database Handler wurde abnormal beendet.
- 154 In UDS/SQL wurde ein nicht behebbarer Fehler erkannt; das Programm sollte beendet werden (STOP RUN bei COBOL-Programmen). Die Transaktion wurde nicht abgeschlossen.
- 155 Während UDS/SQL eine DML-Anweisung bearbeitet, trifft für die gleiche Transaktion eine weitere DML-Anweisung ein (Entserialisierung).
Mögliche Fehlerursachen:
Asynchrone Aktivitäten des Anwenderprogramms (z.B. DML-Anweisung in STXIT-Routine) oder UDS/SQL-Systemfehler.

Weitere Statuscodes der UDS-Online-Utility

- 161 Eine Transaktion einer Online-Utility ist bereits auf demselben Realm aktiv.
- 162 Eine parallel ablaufende User-Transaktion hat eine Online-Realm-Erweiterung angestoßen und damit die Online-Utility temporär behindert.
- 163 Auf einem temporären Realm ist die Online-Utility nicht erlaubt.
- 164 Für diesen RELOCATE-Type ist USAGE-MODE EXCLUSIVE UPDATE erforderlich
- 165 Der angegebene SET ist keine verteilbare Liste
- 166 Der angegebene Realm ist für die Satzart nicht erlaubt
- 167 Konkurrierende Änderung einer parallelen User-TA. Die Utility-TA wird zurückgesetzt.

Statuscodes zu FIND/FETCH

- 183 Der Suchausdruck überschreitet die maximale Länge.
- 184 Der Temporäre Realm ist nicht vorhanden.
- 191 Sowohl der Objekt-Set als auch der LIMITED-Set sind dynamisch.
- 192 Der LIMITED-Set ist leer.
- 193 FIND/FETCH Format 7:
Der LIMITED-Set enthält eine andere Satzart als der Objekt-Set.
FIND/FETCH Format 4 und 7:
Der Objekt-Set ist dynamisch und enthält eine andere Satzart als die angegebene.
FIND/FETCH Format 3:
Der angegebene Satzname unterscheidet sich von dem im vorangegangenen FIND/FETCH Format 7 angegebenen Satznamen.
- 194 Vergleichswert oder Sortierfeld hat die Länge 0 oder eine für den Feldtyp nicht erlaubte Länge.
- 195 Vergleichswert oder Sortierfeld hat unbekanntes Feldtyp oder der Vergleichswert enthält nicht typverträgliche Daten.
- 197 Kein FIND/FETCH Format 7 vorausgegangen.
- 198 Der CRS des Result-Sets wurde durch eine andere Transaktion aus dem Objekt-Set aus- bzw. in eine andere Occurrence umgehängt.

Statuscodes zur Zusammenarbeit mit openUTM

- 200 FINISH:
Die FINISH-Anweisung wurde akzeptiert; die Ausführung des FINISH wird jedoch bis zum openUTM-Transaktionsende-Aufruf an die DC-Steuerung (PEND) verzögert. Es werden keine DML-Anweisungen mehr angenommen.
- 201 Nach dem verzögerten FINISH wurde noch eine weitere DML-Anweisung abgesetzt. Die DML-Anweisung wird ignoriert.
- 218 Systemübergreifender Deadlock, der sich nur durch Freigeben des openUTM-Anwendertasks auflösen lässt (z.B. durch PEND RS).

Beispiele

- lokaler UDS/SQL-openUTM-Betrieb:
Deadlock zwischen UDS/SQL-Betriebsmitteln (Daten) und openUTM-Betriebsmitteln (Tasks).
- Verteilte Verarbeitung über UDS-D oder openUTM-D:
Deadlock zwischen UDS/SQL-Betriebsmitteln (Daten) und/oder openUTM-Betriebsmitteln (Tasks).

Die Erkennung solcher Deadlocks erfolgt über eine Zeitüberwachung von Wartesituationen (PP DEADTIME). Nach Ablauf dieses Zeitlimits wird der Statuscode 218 angezeigt, auch wenn u.U. kein wirklicher Deadlock vorliegt.

Statuscodes zur LOOK-Funktion

- 781 Element nicht gefunden oder unbekannter Realmname bei der Online-Utility.
- 782 Es existiert kein nächstes Element.
- 783 Ein Element der Liste nicht gefunden.
- 784 Die eingegebene Feldreferenz existiert nicht. Es wurde die Beschreibung mit der nächstniedrigeren Feldreferenz ausgegeben.
- 785 Der Ergebnisvektor einer zusammengesetzten LOOKC-Funktion muss durch eine lückenlose Folge von entsprechenden LOOKC-Anweisungen abgerufen werden.
- 786 Satzart mit diesem Subschema nicht prozessierbar, da sie Daten eines Typs enthält, der dem Anwenderprogramm nicht bekannt ist.
- 789 Das angegebene Subschema existiert nicht.

Statuscodes zur Zuordnung von Speicherplatz oder Database Key

- 802 Der Speicherplatz im Realm ist erschöpft oder eine aktivierte Online-Realm-Erweiterung ist gescheitert. Der betroffene Satz kann nicht gespeichert oder in eine Set-Occurrence eingefügt werden.
- 804 Zur Speicherung eines neuen Satzes ist kein Database Key mehr verfügbar oder eine aktivierte Online-Realm-Erweiterung ist gescheitert.
- 805 Der System-Adressraum des DBH ist erschöpft. Das Tabellenwerk des DBH kann nicht mehr dynamisch erweitert werden. Der Datenbankadministrator wurde verständigt.

Statuscodes zu variablen Feldern und Komprimierung

- 888 Die Länge des variablen Feldes ist größer als im Schema definiert oder negativ.
- 898 STORE/MODIFY Format-2 ist bei variablen Feldern nicht erlaubt.
- 899 STORE:
Die Anzahl der Felder, die gespeichert werden sollen, ist so groß, dass die Größe des komprimierten Satzes größer ist als eine Seite.
- GET:
Eines der gewünschten Felder ist im komprimierten Satz in der Datenbank nicht vorhanden.
- MODIFY Format-1:
Dieses Format ist nicht erlaubt, falls der angesprochene Satz in komprimierter Form vorliegt.
- MODIFY Format-2:
Eines der Felder, das verändert werden soll, ist im komprimierten Satz nicht vorhanden.

Statuscodes bezüglich Zugriffsrechten

- 901 Zugriff auf einen Realm, Record oder Set innerhalb der Benutzergruppe nicht erlaubt, oder die Dienstprogramme ONLINE-PRIVACY bzw. ONLINE-UTILITY versuchen auf eine Datenbank zuzugreifen, die nicht in der Ablaufkennung des Dienstprogramms steht. Es ist nicht möglich, mittels Setzen des P-Parameters PRIVACY-CHECK auf OFF dieses Verhalten der Dienstprogramme zu umgehen.
- 950 Benutzergruppe unbekannt (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, BPRIVACY).
- 954 Für die Benutzergruppe sind keine Zugriffsberechtigungen definiert.

7.5 Statuscodes der CALL-DML

(siehe Handbuch „[Anwendungen programmieren](#)“, Abschnitt 10.1)

DML-Wahleintrag-Fehler:

- C00 Der angegebene Funktionsname ist nicht korrekt.
- C01 Die angegebene Funktionswahl ist bei dem angegebenen Funktionsnamen nicht erlaubt.
- C02 Die angegebene Zusatzwahl ist bei der angegebenen Kombination von Funktionsname und Funktionswahl nicht erlaubt oder sie ist syntaktisch fehlerhaft.

Satzname-Fehler:

- C03 Der angegebene Satzname ist im betreffenden Subschema nicht vorhanden oder nicht eindeutig.
- C04 Ein obligatorischer Satzname wurde nicht angegeben.

SetName-Fehler:

- C05 Der angegebene Setname ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.
- C06 Syntaxfehler in der Setnamenleiste
(zu viele Setnamen; Trennung oder Abschluss der Setnamen fehlerhaft; Setname tritt mehrfach auf)

Realm-Namen-Fehler:

- C07 Der angegebene Realm-Name ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.
- C08 Syntaxfehler in der Realm-Namenleiste
(zu viele Realm-Namen; Trennung oder Abschluss der Realm-Namen fehlerhaft; Realm-Name tritt mehrfach auf)

Feldnamen-Fehler:

- C09 Der angegebene Feldname ist im betreffenden Satz des aktuellen Subschemas nicht vorhanden oder nicht eindeutig.
- C10 Syntaxfehler in der Feldnamenleiste
(zu viele Feldnamen; Trennung oder Abschluss der Feldnamen fehlerhaft)

IF-Ergebnis:

- C11 Die IF-Bedingung trifft nicht zu.
C11 ist nicht als Fehlercode, sondern als Ergebnis der IF-DML-Anweisung anzusehen; 000, wenn Bedingung zutrifft

Suchausdruck-Fehler:

- C20 Der Suchausdruck enthält zu viele Suchbedingungen.
- C21 Eine NXT-Suchbedingung nach einem OR-Operator ist verboten.
- C22 Das Trennzeichen vor und hinter dem Feldnamen bzw. Vergleichsoperator jeder Suchbedingung muss jeweils ein Zwischenraum sein.
- C23 Die Anzahl der Klammern einer NXT-Suchbedingung muss gleich Null sein.
- C24 Die Maske einer Suchbedingung darf nur aus den Zeichen 0 und 1 bestehen und muss mit einem Zwischenraum abgeschlossen sein.
- C25 Eine NXT-Suchbedingung darf nicht innerhalb von Klammern stehen.
- C26 Die Länge der Maske einer Suchbedingung muss gleich der Länge des Feldes sein.
- C27 NXT-Suchbedingungen dürfen nur am Ende eines Suchausdrucks stehen.
- C28 Eine Suchbedingung ist nicht mit `_OR_`, `_AN_` oder `_END` abgeschlossen.
- C29 Die Länge des Wertes in einer Suchbedingung ist inkorrekt.
- C30 Die Anzahl der rechten Klammern einer Suchbedingung ist nicht numerisch.
- C32 In einem Suchausdruck sind mehr linke als rechte Klammern vorhanden.
- C33 In einer NXT-Suchbedingung ist die NEQ-Beziehung verboten.
- C34 Der Beziehungs-Operator in einer Suchbedingung ist nicht korrekt.
- C35 Die Anzahl der linken Klammern einer Suchbedingung ist nicht numerisch.
- C37 In einer Suchbedingung sind zu viele rechte Klammern angegeben.
- C38 Der Beziehungs-Operator in einer Suchbedingung wird nicht von einem Zwischenraum gefolgt.
- C39 Der Feldname einer Suchbedingung ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.
- C40 Der Feldtyp einer Suchbedingung ist abdruckbar numerisch, der zugehörige Vergleichswert nicht.
- C41 Der Feldtyp einer Suchbedingung ist dezimal gepackt, der zugehörige Vergleichswert nicht.

C42 Suchbedingungen sind für diesen Feldtyp nicht erlaubt.

Retaining-Wahleintrag-Fehler:

- C61 Der angegebene Retaining-Wahleintrag (Spezialparameter-1) ist nicht korrekt.
- C62 Ein angegebener Retaining-Setname (Spezialparameter-1) ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.
- C63 Syntaxfehler in der Retaining-Setnamenleiste (zu viele Setnamen; Trennung oder Abschluss der Setnamen fehlerhaft; Setname tritt mehrfach auf)

Andere Fehler:

- C66 Das SSITAB-Modul des Subschemas ist nicht identifizierbar oder der angegebene Subschemaname stimmt nur in den ersten 6 Zeichen mit dem im SSITAB-Modul vermerkten Subschemanamen überein, nicht aber in voller Länge. BCALLSI-Lauf durchführen.
- C72 Die Ganzzahl der Positionsangabe eines FIND4/FTCH4-Aufrufs darf nicht Null sein.

Spezielle FIND7A/FTCH7A-Fehler:

- C74 Der angegebene Name des begrenzenden Sets ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.
- C75 Der angegebene Name des Ergebnis-Sets ist im aktuellen Subschema nicht vorhanden oder nicht eindeutig.

Spezielle LOOKC-Fehler:

- C80 Die Anzahl der LOOKC-Blöcke muss zwischen 1 und 255 (einschließlich) liegen.

Benutzer-Kommunikations-Fehler:

- C90 Der vom Konverter-Modul UDSCDML benötigte Arbeitspuffer kann nicht im notwendigen Umfang zur Verfügung gestellt werden. Gegebenenfalls muss der Communication Pool vergrößert werden (siehe Handbuch „[Datenbankbetrieb](#)“).
- C91 Der Fehlerausgang DSCEXT wurde nicht definiert.
- C94 Das Konverter-Modul UDSCDML ist nicht vorhanden.
- C95 Das von BCALLSI erstellte SSITAB-Modul ist nicht vorhanden, oder konnte nicht in den Speicher geladen werden (z.B. wegen Speicherplatzmangels).

- C98 Es wird versucht, ACCPTL, FIND1L, FTCH1L, STORE1L oder STORE2L mit einem SSITAB-Modul auszuführen, das vor UDS/SQL V2.0 erzeugt wurde, oder mit einem Subschema "FORM IS OLD". Für die Ausführung der genannten Funktionen wird ein SSITAB-Modul einer Version ab UDS/SQL V2.0 benötigt.
- C99 Das SSITAB-Modul ist ungültig oder passt nicht zur Version des CALL-DML-Umsetzers.

Zulässigkeitsprüfung für DML-Anweisungen auf Grund der Subschemastruktur:

- P01 Ein FIND2/FTCH2 mit Wahleintrag ANY... ist nur erlaubt, wenn LOCATION MODE IS CALC spezifiziert ist und alle Keys der Satzart im Subschema vorhanden sind.
- P02 Ein FIND2/FTCH2 mit Wahleintrag DUPLIC ist nur erlaubt, wenn LOCATION MODE IS CALC und DUPLICATES ARE ALLOWED spezifiziert sind und alle Keys der Satzart im Subschema vorhanden sind.
- P03 Bei dem aktuellen FIND3/FTCH3 sind Duplikate nicht erlaubt.
- P04 Ein FIND7A/FTCH7A ist nur erlaubt, wenn die angesprochene Satzart Member im angegebenen Set ist.
- P05 Ein FIND7A/FTCH7A bei SET OCCURRENCE SELECTION IS THRU LOCATION MODE OF OWNER ist
- im Zusammenhang mit LOCATION MODE IS DIRECT nur erlaubt, wenn das betroffene Feld im Subschema vorhanden ist.
 - im Zusammenhang mit LOCATION MODE IS CALC nur erlaubt, wenn alle Keys der Satzart im Subschema vorhanden sind.
- P06 Ein FIND4/FTCH4 bzw. FIND5/FTCH5 ist nur erlaubt, wenn die angegebene Satzart Member des angegebenen Sets ist.
- P07 Ein FIND4/FTCH4 bzw. FIND5/FTCH5 ist nur erlaubt, wenn die angegebene Satzart im angegebenen Realm zulässig ist.
- P08 Ein FIND6/FTCH6 ist nur erlaubt, wenn es sich nicht um einen SYSTEM Set handelt.
- P09 Die spezifizierte Speicherungsform des Sets lässt kein CONNEC bzw. DISCON zu oder bei einem DISCON ALLFRM ist der angegebene Set kein Dynamic Set.
- P10 In der Setnamenleiste eines CONNEC bzw. DISCON sind nur Sets zulässig, die untereinander dieselbe Satzart als Member haben.
- P11 Der Current of Rununit muss bei einem CONNEC bzw. DISCON der Membersatzart des angegebenen Sets angehören.

- P12 Bei einem CONNEC TO-ALL muss mindestens ein Set mit der angesprochenen Satzart im Subschema enthalten sein, die nicht MANDATORY AUTOMATIC ist. Bei einem DISCON FRMALL muss die angesprochene Satzart OPTIONAL Member in mindestens einem Set des Subschemas sein.
- P13 Der angegebene MODIF1/2 ist nicht erlaubt.
- P14 Der angegebene STORE1/2 ist nicht erlaubt.
- P15 Der angegebene ERASEC ist nicht erlaubt.
- P16 Der in der RESULT- und/oder LIMITED-Klausel angegebene Set ist kein dynamischer Set.

8 DMLTEST

8.1 Testen der DML-Funktionen

(siehe Handbuch „[Anwendungen programmieren](#)“, Kapitel 9)

Mit dem Programm DMLTEST können Sie

- einzelne DML-Funktionen im Dialog testen
- Testprozeduren ablaufen lassen
- auf jede beliebige Datenbankkonfiguration zugreifen
- mit KDBS zusammenarbeiten

8.2 Schlüsselwortparameter

In den Formaten der DMLTEST-Kommandos werden außerdem folgende Schlüsselwortparameter verwendet:

Schlüsselwortparameter	Standardwert
$value: [\underline{VAL}]= \left\{ \begin{array}{l} literal \\ [n1] \left\{ \begin{array}{l} C \\ X \\ P \end{array} \right\} [Ln2] 'literal' \\ 'literal' \end{array} \right\}$	-
$distance: [\underline{DIST}]= \left\{ \begin{array}{l} n \\ X'n' \end{array} \right\}$	D=0
$length: [\underline{LNG}]= \left\{ \begin{array}{l} n \\ X'n' \end{array} \right\}$	L=8
$name: [\underline{NAME}]= literal$ maximale Länge : 20 erlaubte Zeichen: 1. Stelle A-Z, ab 2. Stelle A-Z,-,0-9	-
$filename: [\left\{ \begin{array}{l} \underline{OML} \\ \underline{FILE} \end{array} \right\}] literal$	OML= DMLTEST.MODLIB bei PERFORM
$repetition: [\left\{ \begin{array}{l} \underline{REP} \\ \underline{STEP} \end{array} \right\}] \left\{ \begin{array}{l} n \\ X'n' \end{array} \right\}$	4 bei TRACE 1 in allen anderen Fällen
$form: [\underline{FORM}]= \left\{ \begin{array}{l} C \\ X \\ D \end{array} \right\}$	F=C
$condition: [\left\{ \begin{array}{l} \underline{CASE} \\ \underline{COND} \end{array} \right\}] \left\{ \begin{array}{l} \underline{RECORD} \left\{ \begin{array}{l} (n) \\ (X'n') \end{array} \right\} \\ \underline{TIME} \\ \underline{RCODE} \\ definition \end{array} \right\} \left\{ \begin{array}{l} EQ \\ NE \\ LT \\ GT \\ LE \\ GE \end{array} \right\} value$	-

Tabelle 26: Schlüsselwortparameter mit Standardwerten

definition muss in Hochkommata gesetzt werden

- C alphanumerische Darstellung
- X sedezimale Darstellung
- D Dump-Format
- P gepackte Darstellung
- n Ganzzahl
- n1 Multiplikationsfaktor
- n2 Länge des Literals

8.3 Schlüsselwörter

Die Variable *parameter* in den DMLTEST-Kommandos kann durch folgende Werte ersetzt werden (siehe auch [Seite 58](#)):

Stellung in der CALL-Parameterleiste	Schlüsselwort für den Parameter	
	bei CDML	bei KDBS
1	FCOD	OP
2	FOPT	RE
3	SOPT	DB
4	UINF	AR
5	RECN	FS
6	SETN	SI
7	RLMN	KB
8	ITMN	KE
9	RECA	RT
10	SPP1	ST
11	SPP2	FSI ²
12	SPP3	–
	SEX ¹	–
	SUBS	–

Tabelle 27: Werte bei CDML und KDBS

- 1 Redefinition von ITMN: Diesen Parameter müssen Sie verwenden, wenn Sie die Aufbereitung eines Suchausdrucks von DMLTEST-Syntax auf CALL-DML-Syntax wünschen.
- 2 Redefinition von SI: Diesen Parameter müssen Sie verwenden, wenn Sie die Aufbereitung eines Suchausdrucks von DMLTEST-Syntax auf KDBS-Syntax wünschen.

Sie können beliebig viele Kommandos hintereinander schreiben. Das Trennzeichen ist „;“.

Kommando	Funktion																								
LANGUAGE { <table style="display: inline-table; vertical-align: middle;"> <tr><td>CDML</td></tr> <tr><td>CDML30</td></tr> <tr><td>COBOL</td></tr> <tr><td>COBOL30</td></tr> <tr><td>KDBS</td></tr> <tr><td>KKDS</td></tr> <tr><td>KLDS</td></tr> </table>	CDML	CDML30	COBOL	COBOL30	KDBS	KKDS	KLDS	die Datenbehandlungssprache auswählen Standardwert: COBOL																	
CDML																									
CDML30																									
COBOL																									
COBOL30																									
KDBS																									
KKDS																									
KLDS																									
LEAVE[<i>condition</i>]	einen Prozeduraufruf abbrechen																								
LIST } <table style="display: inline-table; vertical-align: middle;"> <tr><td>{</td><td>CMD</td><td>}</td></tr> <tr><td></td><td>DCL</td><td>}</td></tr> <tr><td></td><td>DEF</td><td>}</td></tr> <tr><td></td><td>PROC</td><td>}</td></tr> <tr><td></td><td colspan="2">kommandoname</td></tr> <tr><td></td><td colspan="2">declaration</td></tr> <tr><td></td><td colspan="2">definition</td></tr> <tr><td></td><td colspan="2">procname</td></tr> </table> [, <i>name</i>]	{	CMD	}		DCL	}		DEF	}		PROC	}		kommandoname			declaration			definition			procname		die angegebene Information auf SYSOUT ausgeben
{	CMD	}																							
	DCL	}																							
	DEF	}																							
	PROC	}																							
	kommandoname																								
	declaration																								
	definition																								
	procname																								
MOVE { <table style="display: inline-table; vertical-align: middle;"> <tr><td>RECORD</td></tr> <tr><td>RCODE</td></tr> <tr><td>parameter</td></tr> <tr><td>definition</td></tr> <tr><td>declaration</td></tr> </table> } , <i>value</i> [, <i>distance</i>] [, <i>condition</i>]	RECORD	RCODE	parameter	definition	declaration	die angegebenen Bereiche mit Werten versorgen																			
RECORD																									
RCODE																									
parameter																									
definition																									
declaration																									
NEXT	auf Unterbrechungen reagieren oder das aktuelle Kommando abbrechen																								
PERFORM <i>name</i> [, <i>filename</i>] [, <i>condition</i>]	ein Modul aufrufen																								
PRINT { <table style="display: inline-table; vertical-align: middle;"> <tr><td>RECORD</td></tr> <tr><td>RCODE</td></tr> <tr><td>TALLY</td></tr> <tr><td>TIME</td></tr> </table> } [, <i>distance</i>] [, <i>length</i>] [, <i>form</i>] [, <i>condition</i>]	RECORD	RCODE	TALLY	TIME	den angegebenen Bereich oder Wert nach jeder DML-Anweisung auf SYSLST ausgeben																				
RECORD																									
RCODE																									
TALLY																									
TIME																									
PROC <i>procname</i>	eine Prozedurdefinition eröffnen																								
PROFF } [{ <table style="display: inline-table; vertical-align: middle;"> <tr><td>RECORD</td></tr> <tr><td>RCODE</td></tr> <tr><td>TIME</td></tr> </table> }] [, <i>condition</i>]	RECORD	RCODE	TIME	PRINT-Funktion ausschalten																					
RECORD																									
RCODE																									
TIME																									
PROT [{ <table style="display: inline-table; vertical-align: middle;"> <tr><td>ON</td></tr> <tr><td>OFF</td></tr> <tr><td>OUT</td></tr> </table> }] [, <i>condition</i>]	ON	OFF	OUT	Protokollfunktion ein- oder ausschalten Standardwert: PROT ON																					
ON																									
OFF																									
OUT																									
{ <table style="display: inline-table; vertical-align: middle;"> <tr><td>REMARK</td></tr> <tr><td>*</td></tr> </table> } <i>literal</i>	REMARK	*	Kommentarzeilen einfügen																						
REMARK																									
*																									

Tabelle 28: Übersicht über die DMLTEST-Kommandos

(Teil 2 von 3)

8.5 Übersicht über die Unterschiede zwischen den DMLTEST-DML-Anweisungen und den COBOL-DML-Anweisungen

(siehe Handbuch „Anwendungen programmieren“, Abschnitt 9.3.1)

Anweisung	COBOL-DML	DMLTEST-DML
ACCEPT	Format 1: <i>fieldname-1</i>	DB-KEY, DB-KEY-LONG
	Format 2: <i>fieldname-2</i> <i>fieldname-3</i>	AREA-ID DB-KEY, DB-KEY-LONG
FIND/FETCH	Format 1: <i>fieldname</i> OR PRIOR/NEXT	DB-KEY, DB-KEY-LONG Diese Klausel können Sie in DMLTEST in der Form SET FOPT,DBKPRI/DBKNXT verwenden, nachdem Sie die FIND1/FTCH1- bzw. FIND1L/FTCH1L-Anweisung in COBOL-DML-Syntax (ohne EXECUTE) eingegeben haben.
	Format 2:	Wenn die Satzart in mehreren Realms gespeichert werden kann, müssen Sie IMP verwenden: ... <i>satzname</i> [<u>IMP</u>]
	Format 3:	Sie können USING erweitern: ...[$\left. \begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right\} \textit{satzname}$]
	Format 4: Format 7: <i>fieldname-1</i> OR PRIOR/NEXT	- TALLY Diese Klausel können Sie in DMLTEST in der Form SET FOPT,...ITP/...ITN verwenden, nachdem Sie die FIND7A/FTCH7A-Anweisung in COBOL-DML-Syntax (ohne EXECUTE) eingegeben haben.
GET		Sie können die Anweisung erweitern: ...[$\left. \begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right\} \textit{satzname}$]
IF	Format 1: Format 2:	} NEXT SENTENCE, ELSE und NOT dürfen Sie nicht verwenden
MODIFY		Die Anweisung hat folgenden Zusatz: ... $\left. \begin{array}{c} \text{IN} \\ \text{OF} \end{array} \right\} \textit{satzname}$
STORE		Wenn die Satzart in mehreren Realms gespeichert werden kann oder der Database-Key-Wert durch den Anwender vergeben wird (DDL-Klausel LOC MODE), müssen Sie den Zusatz IMP bzw.IMP-LONG verwenden: ... <i>satzname</i> [<u>IMP</u>]... bzw.... <i>satzname</i> [<u>IMP-LONG</u>]..

Tabelle 29: Unterschiede zwischen COBOL-DML- und DMLTEST-DML-Anweisungen

9 Schema-DDL, SSL und Subschema-DDL übersetzen

9.1 Kommandofolge zum Übersetzen der Schema-DDL

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.2.2)

```
01 /CREATE-FILE FILE-NAME=dbname.COSSD ...
02 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
03 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
04 /CREATE-FILE FILE-NAME=dbname.DBSTAT, SUPPRESS-ERRORS=*FILE-EXISTING
    /CREATE-FILE FILE-NAME=dbname.DBSTAT.SAVE, SUPPRESS-ERRORS=*FILE-EXISTING
05 /START-UDS-DDL
06 ddl-compiler-anweisungen
07 END
```

06 Die einzelnen Anweisungen können, durch Kommas oder Leerzeichen getrennt, in einer Zeile eingegeben werden.

9.2 Kommandofolge zum Übersetzen der SSL

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.2.3)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
03 /START-UDS-SSL
04 ssl-compiler-anweisungen
05 END
```

04 Die einzelnen Anweisungen können, durch Kommas oder Leerzeichen getrennt, in einer Zeile eingegeben werden.

9.3 Kommandofolge zum Übersetzen der Subschema-DDL

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.4.1)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
03 /START-UDS-DDL
04 sddl-compiler-anweisungen
05 END
```

04 Die einzelnen Anweisungen können durch Kommas oder Leerzeichen getrennt in einer Zeile eingegeben werden.

9.4 Anweisungen für den DDL-Compiler bzw. den SSL-Compiler

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.2.2)

Anweisung	Com- piler	Stan- dard- wert	Bedeutung
<code>PARLIST IS</code> $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	DDL SDDL SSL	NO	wahlweise; YES alle Anweisungen werden auf SYSLST aufgelistet NO die Anweisungen werden nicht aufgelistet
<code>SORCLIST IS</code> $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	DDL SDDL SSL	YES	wahlweise; YES auf SYSLST wird ein Protokoll ausgedruckt, u.U. mit Fehlermeldungen NO kein Protokoll wird ausgedruckt
<code>SOURCE IS</code> $\left\{ \begin{array}{l} 'dateiname' \\ 'bib(element)' \end{array} \right\}$	DDL SDDL SSL	-	nicht erforderlich, wenn die Eingabe im Dialog erfolgt oder die Eingabedatei SYSDTA zugewiesen ist - in diesem Fall ist zu beachten, dass zunächst alle Anweisungen (mindestens END) und anschließend die eigentliche Source eingegeben wird. weist dem Compiler die Datei zu, die die Schema-DDL/Subschema-DDL/SSL enthält. An Stelle von <i>'dateiname'</i> kann auch ein Element einer Programmbibliothek angegeben werden (siehe Handbuch „ LMS (BS2000) “, Programmbibliotheken). <i>bib</i> : Name der Programmbibliothek <i>element</i> : Name des Elements SYSDTA wird auf die Eingabedatei umgeschaltet. Nach Beendigung des Compiler-Laufs wird SYSDTA wieder auf SYSCMD zurückgesetzt. Die Anweisungen „SOURCE IS“ und „DELETE SCHEMA“ bzw. „DELETE SUBSCHEMA“ dürfen Sie nicht innerhalb desselben DDL-Compiler-Laufs verwenden.

Tabelle 30: Anweisungen zum Übersetzen der Schema-DDL/Subschema-DDL/SSL

(Teil 1 von 5)

Anweisung	Com- piler	Stan- dard- wert	Bedeutung
<u>SUBSCHEMA</u> FORM IS <u>OLD</u>	SDDL	-	<p>wahlweise; diese Anweisung wird nur noch für Subschemata benötigt, die von KDBS-Anwendungen genutzt werden; sie ist nur zulässig in Verbindung mit der Anweisung „SOURCE IS <i>dateiname</i>“ und wird bei der Schemaübersetzung ignoriert.</p> <p>Die Anweisung „SUBSCHEMA FORM IS OLD“ bewirkt, dass das transformierte Subschemata und die Prüftabelle (CHECK-TABLE) in einem internen Format in das COSSD eingetragen wird, das bis einschließlich UDS/SQL V1.2 das Standardformat war („alte“ Form; alle Referenznummern sind 1 byte lang).</p> <p>Ein Subschemata kann nur dann in eine Form gemäß UDS/SQL V1.2 übersetzt werden, wenn die folgenden Bedingungen erfüllt sind:</p> <ul style="list-style-type: none"> – Kein Feld des Subschematas hat den Typ DATABASE-KEY-LONG. – Kein Feld des Subschematas hat den Typ NATIONAL. – Keine Satzart des Subschematas ist länger als 2020 byte. – Alle Satzart- und Setnummern des Schematas sind ≤ 254. <p>Andernfalls beendet sich der DDL-Compiler mit Syntaxfehler und das Subschemata wird nicht in DBCOM und COSSD eingetragen.</p>
<u>DELETE</u> SCHEMA ' <i>schemaname</i> '	DDL	-	<p>wahlweise; das angegebene Schema wird wieder gelöscht; sinnvoll, wenn nach der Umstrukturierung mit BALTER die DDL zwar richtig läuft, aber die SSL-Übersetzung Fehler anzeigt, die eigentlich auf die DDL zurückzuführen sind</p> <p><i>schemaname</i>: Name des Schematas</p> <p>Die Anweisungen „SOURCE IS“ und „DELETE SCHEMA“ dürfen Sie nicht innerhalb desselben DDL-Compiler-Laufs verwenden.</p>

Tabelle 30: Anweisungen zum Übersetzen der Schema-DDL/Subschemata-DDL/SSL

(Teil 2 von 5)

Anweisung	Com- piler	Stan- dard- wert	Bedeutung
<p><u>DELETE</u> [<u>ONLY</u>]SUBSCHEMA</p> <p>'<i>subschemaname</i>' $\left. \begin{array}{l} \{ \text{OF} \\ \vdots \} \right\}$</p> <p>SCHEMA '<i>schemaname</i>'</p>	SDDL	-	<p>wahlweise; das angegebene Subschema wird gelöscht. Das Subschema, das übersetzt wird, und das Subschema, das Sie durch die DELETE-Anweisung benennen, kann denselben Namen haben, da es vor dem Übersetzungslauf gelöscht wird.</p> <p>ONLY wird der Parameter weggelassen, <u>muss</u> nach der DELETE-Anweisung eine SOURCE-Anweisung folgen.</p> <p>Ist der Parameter angegeben, wird eine SOURCE-Anweisung ignoriert.</p> <p><i>subschemaname</i>: Name des Subschemas <i>schemaname</i>: Name des Schemas</p> <p>beide Namen müssen in Hochkommata eingeschlossen werden</p>
<p><u>DISPLAY IS</u> $\left. \begin{array}{l} \{ \text{YES} \\ \text{NO} \} \right\}$</p>	DDL SDDL SSL	NO	<p>wahlweise; YES die verschiedenen Informationen über Satzarten, Sets usw. aus dem DBCOM werden in Klartext ausgegeben. NO Werte aus dem DBCOM werden nicht ausgegeben</p>

Tabelle 30: Anweisungen zum Übersetzen der Schema-DDL/Subschema-DDL/SSL

(Teil 3 von 5)

Anweisung	Com- piler	Stan- dard- wert	Bedeutung
<u>CREATE</u> COSSD ' <i>schemaname</i> '	DDL SDDL	-	nachträglich das COSSD einrichten; wenn dies bei der Schemaübersetzung verges- sen wurde oder der DDL-Compiler wegen eines Fehlers beim Einrichten des COSSD abnormal beendet wurde, so kann es bis zur Subschema- Übersetzung in einem eigenen Lauf nachgeholt werden; <i>schemaname</i> muss in Hochkommata einge- schlossen werden. Das COSSD muss vor dem Compilerlauf per CREATE-FILE-Kommando eingerichtet wer- den. Hinweis: Wenn Sie gleichzeitig den Parameter SOURCE IS ... angeben, findet keine Überset- zung statt.
<u>COMPARE</u> <u>SUBSCHEMAS</u>	SDDL	-	gilt nur nach einer Umstrukturierung mit BALTER; die Subschemas des alten Schemas werden auf Verträglichkeit zum neuen Schema geprüft; dazu liest der DDL-Compiler die Subschemas aus dem alten COSSD nach einem BALTER- Lauf heraus. Ist das alte Subschema kompatibel zum neuen Schema, wird das Subschema in den neuen DBCOM und das neue COSSD eingetragen.
<u>DIAGNOSTIC</u> IS $\left. \begin{array}{l} \{ \text{YES} \} \\ \{ \text{NO} \} \end{array} \right\}$	SDDL	NO	nur sinnvoll zusammen mit COMPARE; YES Unverträglichkeiten der zum neuen Schema in- kompatiblen Subschemas werden diagnosti- ziert und in Form von Fehlermeldungen aufge- listet NO es werden keine Fehlermeldungen ausgegeben
<u>QUOTE</u> IS $\left. \begin{array}{l} \{ \text{SINGLE} \} \\ \{ \text{DOUBLE} \} \end{array} \right\}$	DDL SDDL	DOU- BLE	wahlweise; SINGLE Literele in der Schema-DDL/Subschema-DDL werden in Apostrophen eingeschlossen DOUBLE Literele in der Schema-DDL/Subschema-DDL werden in Anführungszeichen eingeschlossen

Tabelle 30: Anweisungen zum Übersetzen der Schema-DDL/Subschema-DDL/SSL

(Teil 4 von 5)

Anweisung	Com- piler	Stan- dard- wert	Bedeutung
<u>END</u>	DDL SDDL SSL	-	generell erforderlich; schließt die Eingabe der Anweisungen ab

Tabelle 30: Anweisungen zum Übersetzen der Schema-DDL/Subschema-DDL/SSL

(Teil 5 von 5)

10 Datenbankbetrieb

10.1 Datenbankbetrieb mit dem independent DBH

(siehe Handbuch „[Datenbankbetrieb](#)“, Kapitel 3)

Ladeparameter

Parameter	Standardwert	Bedeutung
PP <u>2KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 2-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
PP <u>4KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 4-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
PP <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
PP <u>ADM</u> = $\left\{ \begin{array}{l} \text{REMOTE} \\ \text{LOCAL} \end{array} \right\}$	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
PP <u>ADMPASS</u> = <i>admkeyword</i>	-	Kennwort für die Administration über DCAM festlegen
PP <u>BCAM-PREFIX</u> = <i>prefix</i>	SUD\$	Präfix festlegen für Namen von Anwendertasks, die auf virtuellen Hosts ablaufen

Tabelle 31: Ladeparameter des independent DBH

(Teil 1 von 6)

Parameter	Standardwert	Bedeutung
<u>PP</u> <u>CATPASS</u> = $\left\{ \begin{array}{l} \text{STANDARD} \\ \text{kennwort} \end{array} \right\}$	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> = $\left\{ \begin{array}{l} \text{MONO-PROCESSOR} \\ \text{MULTI-PROCESSOR} \end{array} \right\}$	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>]dbname [.copyname] [,SHARED-RETRIEVAL] [, $\left\{ \begin{array}{l} n \\ [n],bufferid \end{array} \right\}$]	-	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> = $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	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>]dateiname	-	Eingabedatei zum Anlegen der Verteilertabelle festlegen

Tabelle 31: Ladeparameter des independent DBH

(Teil 2 von 6)

Parameter	Standardwert	Bedeutung
<u>PP DISTRIBUTION</u> = $\left\{ \begin{array}{l} \underline{NO} \\ \underline{STANDBY} \\ \underline{START} \end{array} \right\}$	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> = $\left\{ \begin{array}{l} \underline{STD} \\ \underline{ALL} \end{array} \right\}$	ALL	Umfang eines Dumps beeinflussen
<u>PP END</u>	-	Eingabe der Ladeparameter beenden
<u>PP IO</u> = $\left\{ \begin{array}{l} \underline{ASYNC} \\ \underline{SYNC} \end{array} \right\}$ ~	ASYNC	I/Os in den Servertasks asynchron oder synchron ausführen
<u>PP LOCK</u> = $\left\{ \begin{array}{l} \underline{STD} \\ \underline{SHARED} \\ \underline{EXCLUSIVE} \end{array} \right\}$	STD	Sperrprotokoll festlegen
<u>PP LOG</u> = $\left\{ \begin{array}{l} \underline{NO} \\ :catid: \\ \underline{PUBLIC} \\ (priv-vsn-1/gerät-1 \\ [,priv-vsn-2/gerät-2 \\ [,priv-vsn-3/ \\ gerät-3]]) \\ (vsn-1[,vsn-2 \\ [,vsn-3]]) \end{array} \right\}$	-	RLOG-Datei führen
<u>PP LOG-2</u> = $\left\{ \begin{array}{l} :catid: \\ \underline{PUBLIC} \\ (priv-vsn-1/gerät-1 \\ [,priv-vsn-2/ \\ gerät-2 \\ [,priv-vsn-3/ \\ gerät-3]]) \\ (vsn-1[,vsn-2 \\ [,vsn-3]]) \end{array} \right\}$	-	RLOG-Duplikatdatei führen
<u>PP LOG-SIZE</u> =($[primär]$ [, $[sekundär]]$)	192,192	Speicherplatzgröße der RLOG-Dateien in Anzahl PAM-Seiten festlegen

Tabelle 31: Ladeparameter des independent DBH

(Teil 3 von 6)

Parameter	Standardwert	Bedeutung
PP <u>MAXDB</u> = <i>n</i>	Summe von PP DBNAME	max. Anzahl der Datenbanken der DB-Konfiguration festlegen; <i>n</i> =1..222
PP <u>MPSE</u> G={ STD 64K	STD	Segmentgröße festlegen
PP <u>ORDER-DBSTATUS</u> ={ STD SPECIAL	STD	Systemverhalten steuern, wenn neue UPDATE-Transaktionen oder -Verar- beitungsketten mit der Ausführung von vorgemerkten Aufträgen kollidie- ren
PP <u>PARLIST</u> ={ YES NO	NO	Verwendete Parameter auflisten
PP <u>PASS</u> WORD={ NONE STANDARD <i>kennwort</i> (<i>kennwort,kenn-</i> <i>wort,...</i>)	STD	Kennwort für Dateien festlegen, das der DBH neben PP CATPASS zum Er- öffnen von Dateien verwenden muss
PP <u>PRIVACY-CHECK</u> ={ STD NO-KSET OFF	STD	Behandlung der Privacy-Prüfung steuern
PP <u>PTCSYN</u> CH= ([{ WAIT ABORT COMMIT }][,][{ WAIT ABORT COMMIT }]]])	(WAIT,WAIT)	Behandlung von Transaktionen im Zu- stand PTC steuern; der erste Wert gilt bei einem Warmstart, der zweite gilt während der laufenden Session, wenn der Zustand der primären Teil- transaktion nicht ermittelbar ist
PP <u>RESERVE</u> ={ NONE : <i>catid</i> : PUBLIC (<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>]])	NONE	Ersatzdatenträger für die RLOG- Dateien festlegen
PP <u>RESULT-DELAY</u> = <i>n</i>	0	Auftragsantworten an die Anwender- tasks bündeln; <i>n</i> =1.. <i>m</i> <i>m</i> =PP TRANSACTION

Tabelle 31: Ladeparameter des independent DBH

(Teil 4 von 6)

Parameter	Standardwert	Bedeutung
PP <u>SCHEDULING</u> = $\left\{ \begin{array}{l} \text{SYMMETRIC} \\ \text{ASYMMETRIC} \end{array} \right\}$	SYMMETRIC	Optimierung der Kommunikation zwischen Anwendertask und Servertask sowie die Abarbeitung anstehender DML-Aufträge in der Servertask steuern
PP <u>SERVERTASK</u> = <i>n</i>	1	Anzahl der Servertasks des independent DBH festlegen <i>n</i> =1..30
PP <u>SIP-SIZE</u> = <i>n</i>	1024 ¹ 128 ²	Größe des SSITAB-Pools in Kbyte festlegen; <i>n</i> =1..16384
PP <u>SQL</u> = <i>n</i>	4	max. Anzahl der gleichzeitig aktiven SQL-Vorgänge festlegen; <i>n</i> =0..9999
PP <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
PP <u>STDCKPT</u> = $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	NO	Standard-Konsistenzpunkte im AFIM-Logging festschreiben beim DBH-Start, DBH-Ende und Session-Wiederanlauf
PP <u>SUBSCHEMA</u> = <i>n</i>	1	max. Anzahl der gleichzeitig innerhalb einer Datenbank verwendbaren Sub-schemata festlegen; <i>n</i> =1..100
PP <u>SUBTRANSACTION</u> = <i>n</i>	0	max. Anzahl der parallel eröffneten Dateikennungen pro Datenbank festlegen (nur KDBS); <i>n</i> =1..254
PP <u>TA-ACCESS</u> = $\left\{ \begin{array}{l} \text{STD} \\ \text{SHARED} \end{array} \right\}$	STD	Benutzungsarten für Transaktionen festlegen

Tabelle 31: Ladeparameter des independent DBH

(Teil 5 von 6)

Parameter	Standardwert	Bedeutung
$PP \text{ TRANSACTION} = \left\{ \begin{array}{l} n \\ ([n][,m]) \end{array} \right\}$	(4,1)	<i>n</i> : maximale Anzahl der gleichzeitig aktiven Transaktionen und Anwender-tasks <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>
$PP \text{ UCON} = C' \left\{ \begin{array}{l} (mn) \\ <x \\ nnnn \end{array} \right\} ' [, \left\{ \begin{array}{l} MSG \\ UDS \end{array} \right\}]$	C' <U'	Bedienstation (UCON) festlegen, auf die die DCAM-Administration protokolliert werden soll
$PP \text{ WAIT} = \left\{ \begin{array}{l} EVENT \\ BUSY \end{array} \right\} = n$	EVENT	Warte-Modus einstellen
$PP \text{ WARMSTART} = \left\{ \begin{array}{l} STD \\ FAST \\ VERY-FAST \end{array} \right\}$	STD	Dauer eines Warmstarts beeinflussen

Tabelle 31: Ladeparameter des independent DBH

(Teil 6 von 6)

1 bei Angabe PP MPSEG=STD

2 bei Angabe PP MPSEG=64K

Kommandos zum Starten der Session

(siehe Handbuch „Datenbankbetrieb“, Abschnitt 2.3.1)

Den independent DBH starten Sie mit folgenden Kommandos:

Kommandos	Bedeutung
/SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME= <i>konfigurationsname</i>	Namen der DB-Konfiguration der Session
[/CREATE-FILE FILE-NAME= <i>konfname.TEMPO.nnn</i> ,SUPPORT=*PRIVATE-DISK(VOLUME= <i>priv-vsn</i> ,DEVICE-TYPE= <i>gerät</i> [,SPACE=...])]]	Temporäre Benutzerrealms auf privater Platte
[/CREATE-FILE FILE-NAME= <i>konfname.SLF</i> ,SUPPORT=*PRIVATE-DISK(VOLUME= <i>priv-vsn</i> ,DEVICE-TYPE= <i>gerät</i> [,SPACE=...])]]	Session-Log-File auf privater Platte
/CREATE-FILE FILE-NAME= <i>konfname.DBSTAT</i> ,SUPPRESS-ERRORS=*FILE-EXISTING /CREATE-FILE FILE-NAME= <i>konfname.DBSTAT.SAVE</i> ,SUPPRESS-ERRORS=*FILE-EXISTING	DB-Status-Dateien
[/ASSIGN-SYSDTA TO= { *LIBRARY-ELEMENT(...) } *SYSCMD]	Eingabequelle für die DBH-Parameter
[/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME= <i>dateiname</i>]	
[/SET-FILE-LINK LINK-NAME=UDSDBHJV ,FILE-NAME= <i>fvname</i>]	Variabler Teil des Namens der Session-Jobvariable
[/SET-JV-LINK LINK-NAME=UDSPS01 ,JV-NAME= <i>fvname</i>]	UDS/SQL-Pubset-Deklaration
[/ADD-FILE-LINK LINK-NAME=\$UDSKONF ,FILE-NAME= <i>UDSTRTAB-nachladebibliothek</i>]	Benutzerspezifisches UDSTRTAB-Modul
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL ,VERSION= <i>nn.nann</i> ,SCOPE=*TASK	UDS/SQL- bzw. UDS-D-Produktauswahl (Produktbibliotheken, Subsysteme)
[/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-D ,VERSION= <i>nn.nann</i> ,SCOPE=*TASK]	
<i>Privatinstallation (zusätzliche Kommandos)</i>	
[/ADD-FILE-LINK LINKNAME=\$UDSLIB ,FILE-NAME= <i>UDS/SQL-modulbibliothek</i>]	
[/ADD-FILE-LINK LINKNAME=\$UDSDLIB ,FILE-NAME= <i>UDS-D-modulbibliothek</i>]	

Tabelle 32: DBH-Startkommandos (independent)

(Teil 1 von 2)

Kommandos	Bedeutung
/START-UDS-DBH [RESIDENT-PAGES=*PAR(MIN=... ,MAX=...)]	Starten des independent DBH
<i>Privatinstallation (alternatives Kommando)</i>	
/START-EXECUTABLE-PROGRAM FROM-FILE= (LIB=UDS/SQL-modulbibliothek ,ELEM=UDSSQL)	

Tabelle 32: DBH-Startkommandos (independent)

(Teil 2 von 2)

Syntax des Kommandos START-UDS-DBH:

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

DAL-Kommandos des independent DBH

(siehe Handbuch „Datenbankbetrieb“, Abschnitt 4.4)

DAL-Kommando	Bedeutung
$\text{ABORT } \left\{ \begin{array}{l} \textit{transaktionskennung} [, \text{OPTION}=\text{PTC}] \\ \text{ALL} \end{array} \right\}$	Angegebene offene Transaktionen zurücksetzen OPTION=PTC: ohne Rücksicht auf konfigurationsübergreifende Konsistenz bzw. UDS/SQL-openUTM-Konsistenz
$\text{ACCESS } \left\{ \begin{array}{l} \text{LOCK} \\ \text{RETRIEVAL} \\ \text{UPDATE} \end{array} \right\},$ $\left\{ \begin{array}{l} \text{DB}=\textit{dbname} \\ \text{RN}=\textit{realmname} [, \text{DB}=\textit{dbname}] \end{array} \right\}$	Zugriffssperren auf Datenbank- und Realm-Ebene behandeln
$\text{ACT } \left\{ \begin{array}{l} \text{DBTT-INCR}, \text{DB}=\textit{dbname} [, \text{RECR}=\textit{recordref}] \\ [, \text{SCAN}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}] \\ \text{INCR}, \text{DB}=\textit{dbname} [, \text{RR}=\textit{realmref}] \\ [, \text{EXT}=(\textit{nr-pages}, \textit{min-pages})] \end{array} \right\}$	Online-Erweiterbarkeit von DBTTs bzw. Realms aktivieren
$\text{ADD } \left\{ \begin{array}{l} \text{DB}=[\$userid.]\textit{dbname} [.\textit{copyname}] \\ [, \text{OPTION}=\left\{ \begin{array}{l} \text{SHARED-RETRIEVAL} \\ \text{OWN-BUFFER-SIZE}=\textit{n} \end{array} \right\}] [, \text{ID}=\textit{bufferid}] \\ \text{RN}=\textit{realmname} [, \text{DB}=\textit{dbname}] \\ \text{PW}=\textit{kennwort} \\ \text{ADM}=\textit{admkenwort} \end{array} \right\}$	Datenbanken, Realms und Kennwörter anschließen
$\&\text{ADD DISTRIBUTION},$ $\left\{ \begin{array}{l} \text{NODE}=\textit{prozessorname}, \\ \text{CONF}=\textit{konfname} [, \text{DB}=\textit{dbname}] \\ \text{DB}=\textit{dbname}, \text{SS}=\textit{subschemaname} \\ \text{FILE}=\textit{dateiname} \end{array} \right\}$	Neue Einträge in die Verteiltabelle bringen
%BIB	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.
$\&\text{CHANGE DISTRIBUTION},$ $\text{NODE}=\textit{prozessorname},$ $\text{CONF}=\textit{konfname}$	Eine Konfiguration einem anderen Verarbeitungsrechner zuordnen

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 1 von 6)

DAL-Kommando	Bedeutung
<u>CHECKPOINT</u> [<u>DB</u> = <i>dbname</i> [, <u>OPTION</u> = <u>EVEN-WITHOUT-ALOG</u>]]	Konsistenzpunkte festschreiben im AFIM-Logging für einzelne Datenbanken der Konfiguration oder die gesamte Konfiguration
<u>CLOSE</u> { <u>RUN-UNITS</u> <u>CALLS</u> <u>ADMINISTRATION</u> }	Laufende Session normal beenden bzw. DCAM-Administration beenden
<u>&CLOSE</u> <u>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>CONTINUE</u>	In der Ausgabe des DISPLAY SQL-DAL-Kommandos „blättern“
<u>DEACT</u> { <u>DBIT-<u>INCR</u></u> , <u>DB</u> = <i>dbname</i> [, <u>RECR</u> = <i>recordref</i>] } { <u>INCR</u> , <u>DB</u> = <i>dbname</i> [, <u>RR</u> = <i>realmref</i>] }	Online-Erweiterbarkeit von DBTTs bzw. Realms deaktivieren

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 2 von 6)

DAL-Kommando	Bedeutung
<pre> 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 }] DBIT-INCR [, DB=dbname [, RECR=recordref]] [, LINES={ n } { ALL }] PUBSETS } </pre>	<p>Datenbanken, Transaktionen, Sub-schemata, 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>

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 3 von 6)

DAL-Kommando	Bedeutung
<code>DISPLAY SQL { ,VG=vg-nr ,OPTION=ALL[,VG=vg-nr] ,OPTION=IDLE[,TIME=z] }</code>	Detaillierte Informationen über einen oder mehrere SQL-Vorgänge ausgeben
<code>%DML</code>	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.
<code>DROP { DB=dbname RN=realmname[,DB=dbname] PW=kennwort ADM=admkenwort }</code>	Datenbanken, Realms und Kennwörter ausschließen
<code>&DROP DISTRIBUTION, { ALL NODE=prozessorname, CONF=konfname[,DB=dbname][,ALL] DB=dbname,SS=subschemaname }</code>	Einträge in der Verteiltabelle löschen
<code>DUMP [{ ALL STD transaktionskennung }]</code>	Speicherabzug vom DBH erstellen
<code>EXTEND DBIT ,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

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 4 von 6)

DAL-Kommando	Bedeutung
MODIFY { LOG } , VALUE = { :catid: PUBLIC (priv-vsn-1/gerät-1 [,priv-vsn-2/gerät-2] [,priv-vsn-3/ gerät-3]) (vsn-1[,vsn-2 [,vsn-3]])	Datenträgerzuweisung für eine neu einzurichtende Original- und Duplikat-RLOG-Datei ändern
MODIFY LOG-SIZE, VALUE=([primär] [, [sekundär]])	Speicherplatzgröße der RLOG-Dateien ändern
MODIFY PTC SYNCH, VALUE=([{WAIT ABORT COMMIT}] [, [{WAIT ABORT COMMIT}]])	Wert des DBH-Ladeparameters PTC SYNCH ändern
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]])	Ersatzdatenträger für RLOG-Dateien ändern
NEW PUBSETS	Neue UDS/SQL-Pubset-Deklaration prüfen und vormerken
NEW RLOG	Neue RLOG-Dateien auswählen
PERFORM { NOCANCEL } { CANCEL }	Aufträge anstoßen aus den Kommandos: DROP, ADD, NEW bzw. CHECK-POINT Standardwert: NOCANCEL
&PWD DISTRIBUTION, CONF=konfname, { PWN=kennwortneu PWO=kennwortalt PWO=kennwortalt, PWN=kennwortneu }	Kennwort vergeben und verändern
REACT INCR, DB=dbname[, RR=realmref]	Online-Realm-Erweiterbarkeit reaktivieren
RESET ORDERS	Vorgemerkte Aufträge zurücknehmen
&SAVE DISTRIBUTION, FILE=dateiname	Verteiltabelle sichern
&START DISTRIBUTION	UDS-D-Betrieb starten
STOP { transaktionskennung } { NEW ALL }	Ausführung der angegebenen Transaktion(en) stoppen
&SYNCHRONIZE DISTRIBUTION	Sekundäre Teiltransaktionen im Zustand PTC beenden

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 5 von 6)


DAL-Kommando	Bedeutung
%TERM	<p>Session auf dem kürzesten Weg abbrechen (Nothalt) und wahlweise einen kompletten Speicherabzug des DBH ausgeben</p> <p> VORSICHT!</p> <p>%TERM sollten Sie nur dann verwenden, wenn die Mastertask wegen eines Fehlers keine anderen DAL-Kommandos mehr akzeptiert.</p>
<pre>&UNLOCK DISTRIBUTION, { NODE=prozessorname CONF=konfname DB=dbname SS=subschemaname }</pre>	<p>Sperren für Einträge in der Verteiltabelle aufheben</p>

Tabelle 33: DAL-Kommandos für den independent DBH

(Teil 6 von 6)

10.1.1 UDSADM

UDSADM starten

(siehe Handbuch „Datenbankbetrieb“, Abschnitt 4.1)

UDSADM wird gestartet mit:

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

Regeln zur Auswahl der Anweisungen

Anweisung	Regeln zur Auswahl
END HELP-DAL-CMD MODIFY-MSG-FORMAT MODIFY-MSG-WAIT-TIME DISCONNECT-CONFIGURATION	Diese Anweisungen sind jederzeit erlaubt.
CONNECT-CONFIGURATION	Diese Anweisung ist nur für eine zuvor mit CONNECT-CONFIGURATION 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 34: Regeln zur Auswahl der Anweisungen

UDSADM-Anweisungen

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 4.1.1)

Die Anweisungsformate des Administrationsprogramm UDSADM entsprechen den Regeln von SDF (System Dialog Facility), siehe Handbücher „[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 seconds>	Wartezeit für DAL-Kommandos einstellen
SET-RECEIVE-MODE	Dauerempfangsprozess aktivieren
SHOW-CONNECTION-ATTRIBUTES	Verbindungsinformation anzeigen
SHOW-OUTSTANDING-MSG	Anstehende Nachrichten ausgeben

Tabelle 35: UDSADM-Anweisungen

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

10.2 Datenbankbetrieb mit dem linked-in DBH

(siehe Handbuch „[Datenbankbetrieb](#)“, Kapitel 3)

Ladeparameter

Parameter	Standardwert	Bedeutung
PP <u>2KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 2-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
PP <u>4KB-BUFFER-SIZE</u> = <i>n</i>	1	Größe des 4-Kbyte System Buffer Pools in Mbyte festlegen <i>n</i> =1..2047
PP <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
PP <u>CATPASS</u> = $\left\{ \begin{array}{l} \text{STANDARD} \\ \text{kennwort} \end{array} \right\}$	STD	Kennwort für vom DBH einzurichtende Dateien wie ALOG-Dateien und Temporäre Realms festlegen
PP <u>CONSOLE</u> = $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	NO	Meldungen des linked-in DBH zusätzlich auf der Operatorkonsole ausgeben
PP <u>DBNAME</u> =[<i>userid.</i>] <i>dbname</i> [<i>.copyname</i>] [, <u>SHARED-RETRIEVAL</u>] [, $\left\{ \begin{array}{l} n \\ [n], \text{bufferid} \end{array} \right\}$]	-	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
PP <u>END</u>	-	Eingabe der Ladeparameter beenden
PP <u>LOG</u> = $\left\{ \begin{array}{l} \text{NO} \\ \text{:catid:} \\ \text{PUBLIC} \\ (\text{priv-vsn-1/gerät-1} \\ [, \text{priv-vsn-2/gerät-2}] \\ [, \text{priv-vsn-3/gerät-3}]) \\ (\text{vsn-1}[, \text{vsn-2} \\ [, \text{vsn-3}]) \end{array} \right\}$	-	RLOG-Datei führen

Tabelle 36: Ladeparameter des linked-in DBH

(Teil 1 von 3)

Parameter	Standardwert	Bedeutung
$\text{PP LOG-2} = \left\{ \begin{array}{l} \text{:catid:} \\ \text{PUBLIC} \\ (\text{priv-vsn-1/gerät-1} \\ \text{[,priv-vsn-2/} \\ \text{gerät-2} \\ \text{[,priv-vsn-3/} \\ \text{gerät-3]])} \\ (\text{vsn-1[,vsn-2} \\ \text{[,vsn-3]])} \end{array} \right\}$	-	RLOG-Duplikatdatei führen
$\text{PP LOG-SIZE} = ([\text{primär}] \\ \text{[, [,sekundär]])}$	192,192	Speicherplatzgröße der RLOG-Dateien in Anzahl PAM-Seiten festlegen
$\text{PP MAXDB} = n$	Summe von PP DBNAME	max. Anzahl Datenbanken der DB-Konfiguration festlegen; $n=1..222$
$\text{PP PARLIST} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	NO	Verwendete Ladeparameter auflisten
$\text{PP PASSWORD} = \left\{ \begin{array}{l} \text{NONE} \\ \text{STANDARD} \\ \text{kennwort} \\ (\text{kennwort,} \\ \text{kennwort,..}) \\ \text{kennwort,..}) \end{array} \right\}$	STD	Kennwort für Dateien festlegen, das der DBH neben PP CATPASS zum Eröffnen von Dateien verwenden muss
$\text{PP PRIVACY-CHECK} = \left\{ \begin{array}{l} \text{STD} \\ \text{NO-KSET} \\ \text{OFF} \end{array} \right\}$	STD	Behandlung der Privacy-Prüfung steuern
$\text{PP RESERVE} = \left\{ \begin{array}{l} \text{NONE} \\ \text{:catid:} \\ \text{PUBLIC} \\ (\text{priv-vsn-1/} \\ \text{gerät-1} \\ \text{[,priv-vsn-2/} \\ \text{gerät-2} \\ \text{[,priv-vsn-3/} \\ \text{gerät-3]])} \\ (\text{vsn-1[,vsn-2} \\ \text{[,vsn-3]])} \end{array} \right\}$	NONE	Ersatzdatenträger für die RLOG-Dateien festlegen
$\text{PP STDCKPT} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$	NO	Standard-Konsistenzpunkte im AFIM-Logging festschreiben bei DBH-Start, DBH-Ende und Session-Wiederanlauf
$\text{PP SUBSCHEMA} = n$	1	max. Anzahl der gleichzeitig innerhalb einer Datenbank verwendbaren Subschemas festlegen; $n=1..100$

Tabelle 36: Ladeparameter des linked-in DBH

(Teil 2 von 3)

Parameter	Standardwert	Bedeutung
PP <u>SUBTRANSACTION</u> = <i>n</i>	0	max. Anzahl der parallel eröffneten Dateikennungen pro Datenbank festlegen (nur KDBS); <i>n</i> =1..254
PP <u>TRANSACTION</u> = <u>1</u>	1	Anzahl der gleichzeitig aktiven Transaktionen festlegen
PP <u>WARMSTART</u> = $\left\{ \begin{array}{l} \text{STD} \\ \text{FAST} \\ \text{VERY-FAST} \end{array} \right\}$	STD	Dauer eines Warmstarts beeinflussen

Tabelle 36: Ladeparameter des linked-in DBH

(Teil 3 von 3)

Kommandofolge zum Starten einer linked-in Anwendung(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 2.3.1)

Kommandos	Bedeutung
/SET-FILE-LINK LINK-NAME=DATABASE ,FILE-NAME= <i>konfigurationsname</i>	Namen der DB-Konfiguration der Session
[/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=...])]	Temporäre Benutzerrealms auf privater Platte
[/CREATE-FILE FILE-NAME= <i>konfname</i> .SLF ,SUPPORT=*PRIVATE-DISK(VOLUME= <i>priv-vsn</i> ,DEVICE-TYPE= <i>gerät</i> [,SPACE=...])]	Session-Log-File auf privater Platte
/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	DB-Status-Dateien
[/MODIFY-JOB-SWITCHES OFF=28]	Auswahl des linked-in DBH bei eingebundenem Modul UDSLNKA
[/ASSIGN-SYSDTA TO={*LIBRARY-ELEMENT(...)} *SYSCMD}]	Eingabequelle für die DBH-Parameter
[/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME= <i>dateiname</i>]	
[/SET-FILE-LINK LINK-NAME=UDSDBHJV ,FILE-NAME= <i>jvname</i>]	Variabler Teil des Namens der Session-Jobvariable
[/SET-JV-LINK LINK-NAME=UDSPS01 ,JV-NAME= <i>jvname</i>]	UDS/SQL-Pubset-Deklaration
[/ADD-FILE-LINK LINK-NAME=\$UDSSSI ,FILE-NAME= <i>SSITAB-nachladebibliothek</i>]	SSITAB-Module (CALL-DML)
[/ADD-FILE-LINK LINK-NAME=\$UDSPLEX ,FILE-NAME= <i>PLITAB-nachladebibliothek</i>]	PLITAB-Module (UDSKDBS)
[/ADD-FILE-LINK LINK-NAME=\$UDSKONF ,FILE-NAME= <i>UDSTRTAB-nachladebibliothek</i>]	Benutzerspezifisches UDSTRTAB-Modul

Tabelle 37: DBH-Startkommandos (linked-in)

(Teil 1 von 2)

Kommandos	Bedeutung
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL ,VERSION= <i>nn.nann</i> ,SCOPE=*TASK	UDS/SQL-Produktauswahl (Produktbibliotheken, Subsysteme)
<i>Privatinstallation (zusätzliche Kommandos)</i>	
[/ADD-FILE-LINK LINKNAME=\$UDSLIB ,FILE-NAME=UDS/SQL-modulbibliothek]	
[/ADD-FILE-LINK LINKNAME=\$UDSKLIB ,FILE-NAME=KDBS-modulbibliothek]	Starten der linked-in Anwendung
<i>Privatinstallation (alternatives Kommando)</i>	
/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=...)]	

Tabelle 37: DBH-Startkommandos (linked-in)

(Teil 2 von 2)

DAL-Kommandos des linked-in DBH

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 4.4)

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 38: DAL-Kommandos für den linked-in DBH

10.3 UDSMON

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 11.3)

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> = $\left. \begin{array}{l} S \\ C \\ ALL \\ CS \\ DC \\ DT \\ D \end{array} \right\}$	Zuweisen der Ausgabemaske(n) UDSMON gibt aus im eingestellten Zeitintervall bei: S: STATUS-Maske C: COUNTER-Maske ALL: alle Masken abwechselnd CS : COUNTER-Maske und STATUS-Maske abwechselnd DC: CONNECT-Maske DT: TRANSACTION-Maske D: TRANSACTION-Maske und CONNECT-Maske abwechselnd Standardwert : S
<u>MEDIUM</u> = $\left\{ \begin{array}{l} T, n \\ L, n[, D] \\ F, n[, D] \\ S, n \end{array} \right\}$	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.

Tabelle 39: UDSMON-Anweisungen zum Starten

(Teil 1 von 2)

Anweisung	Bedeutung
<u>DISPLAY</u> { DBCOUNTERS DEPRECATED EXPENSIVE MESSAGES }	Ausgabe in Zwischendatei (vgl. entsprechendes Kommando während des Betriebes, siehe Tabelle 40 auf Seite 160) 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 39: UDSMON-Anweisungen zum Starten

(Teil 2 von 2)

UDSMON-Kommandos während des Betriebs

Kommando	Bedeutung
INFORM-PROGRAM MSG= 'ADD MEDIUM= $\left. \begin{array}{l} T, n \\ L, n[, D] \\ F, n[, D] \\ S, n \end{array} \right\}$ '	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 n : Zeitintervall für die Abfragefrequenz in Sekunden (n=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= 'DISPLAY'	Auflisten der bisher eingestellten Geräte und Zeitintervalle
INFORM-PROGRAM MSG= 'DISPLAY CP-SIZE'	Auflisten der Anzahl und der Größe der Common Pools
INFORM-PROGRAM MSG= 'DISPLAY DB-IO-COUNTER'	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= 'DISPLAY DBCOUNTERS'	Anzeige der Anzahl der DMLs und der Anzahl der I/Os pro Datenbank seit DBH-Start und während die Datenbank online ist.
INFORM-PROGRAM MSG= 'DISPLAY DEPRECATED'	Auflistung von Anwendungen mit Eigenschaften, die ein potentielles Risiko beinhalten (eingeschränkte Adressraumnutzung, veraltete Nachladetechnik, fehlende Sub-schemavalidierung) Die Auflistung wird in eine Zwischendatei geschrieben (siehe Handbuch „Datenbankbetrieb“) und mit einem SHOW-FILE-Kommando vom Monitor angezeigt.

Tabelle 40: UDSMON-Kommandos während des Betriebs

(Teil 1 von 5)

Kommando	Bedeutung
INFORM-PROGRAM MSG= ' <u>DISPLAY</u> <u>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 Handbuch „Datenbankbetrieb“) und mit einem SHOW-FILE-Kommando vom Monitor angezeigt.
INFORM-PROGRAM MSG= ' <u>DISPLAY</u> <u>MESSAGES</u> '	Aufbereitete Auflistung von gepufferten DBH-Meldungen Die Auflistung wird in eine Zwischendatei geschrieben (siehe Handbuch „Datenbankbetrieb“) 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</u> <u>REX</u> '	Die Anzahl aller mit READY EXCLUSIVE oder READY PROTECTED eröffneten Transaktionen wird angegeben
INFORM-PROGRAM MSG= ' <u>DISPLAY</u> <u>SATURATION</u> '	Auflistung der höchsten Sättigung von zentralen Systemressourcen in der bisherigen DBH-Session (z.B. Anzahl paralleler Transaktionen, Nutzung nicht erweiterbarer Memory-Pools)

Tabelle 40: UDSMON-Kommandos während des Betriebs

(Teil 2 von 5)

Kommando	Bedeutung
INFORM-PROGRAM MSG= 'DISPLAY TASKS'	<p>Auflisten aller mit UDS/SQLverbundenen Anwender-tasks und der Anzahl der bisher in dieser Anwendertask bearbeiteten DMLs.</p> <p>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).
INFORM-PROGRAM MSG= 'END'	Beenden des Monitors
INFORM-PROGRAM MSG= 'FINISH MEDIUM= $\left. \begin{array}{l} T \\ L, [D] \\ F, [D] \\ S \end{array} \right\}$	<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>
INFORM-PROGRAM MSG= 'HELP'	Auflisten der Eingabemöglichkeiten

Tabelle 40: UDSMON-Kommandos während des Betriebs

(Teil 3 von 5)

Kommando	Bedeutung
INFORM-PROGRAM MSG= 'MASK= { 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

Tabelle 40: UDSMON-Kommandos während des Betriebs

(Teil 4 von 5)

Kommando	Bedeutung
<p>INFORM-PROGRAM MSG= ' REBASE-COUNTER=NEW '</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.
<p>INFORM-PROGRAM MSG= ' REBASE-COUNTER=OFF '</p>	<p>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.</p>

Tabelle 40: UDSMON-Kommandos während des Betriebs

(Teil 5 von 5)

Kommandofolge zum Starten und Bedienen von UDSMON

(siehe Handbuch „Datenbankbetrieb“, Abschnitt 11.4)

```

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], [DÜ]

05 "udsmon-kommandos während des betriebs"

06 [EM], [DÜ]

07 /INFORM-PROGRAM MSG='END'

```

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.

02 Starten des Monitors.

04/06 Während des Monitor-Betriebs leiten Sie mit [EM], [DÜ] 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.

10.4 Pubsets für UDS/SQL nutzen

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 9.1)

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],...</code> <code>ADD DB=[\$dbuserid.]dbname[.copyname],...</code>	Namen und ggf. die Kennung einer Datenbank angeben
<code>PP DISTABLE=[:catid:][\$userid.]dateiname</code> <code>ADD DIS,FILE=[:catid:][\$userid.]dateiname</code> <code>SAVE DIS,FILE=[:catid:][\$userid.]dateiname</code>	Namen einer DISTABLE-Datei angeben
<code>/ADD-FILE-LINK LINK-NAME=PPFILE ,FILE-NAME=[:catid:][\$userid.]dateiname</code> <code>/ASSIGN-SYSDTA TO=[:catid:][\$userid.]dateiname</code>	Ladeparameter-Datei zuweisen

Tabelle 41: 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.

10.5 Jobvariablen für UDS/SQL nutzen

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 9.2)

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	Seite 168
Session	JV-NAME= UDSSQL.DBH. <i>session</i>	Informationen über den Zustand der DBH-Session zur automatisierten Steuerung von Anwendungen und Administration	Seite 170
Datenbank	JV-NAME= UDSSQL.DBH. <i>datenbankname</i>	Informationen über den Zustand einer Datenbank zur automatisierten Steuerung von Anwendungen und Administration	Seite 173
BMEND	LINK-NAME= JVBMEND	Ausgabedaten von BMEND zur automatisierten Steuerung von Sicherung und Recovery	Handbuch „ Sichern, Informieren und Reorganisieren “

Tabelle 42: 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.

10.5.1 Pubset-Deklarations-Jobvariable

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 9.2.1)

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 (siehe Handbuch „[Einführung in das DVS](#)“, Anhang Musterzeichen) darstellen und voneinander durch ein oder mehrere Leerzeichen getrennt sind.

catid-group[catid-group]...

Angabe einer oder mehrerer Catid-Gruppen (1- bis 4-stellige Katalogkennungen ohne „.“), die die in folgender [Tabelle 43](#) 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 >	Ersetzt eine Zeichenfolge, für die gilt: <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 43: 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 44: 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.

10.5.2 Session-Jobvariable

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 9.2.2)

Der DBH versorgt eine Session-Jobvariable zur Verbesserung der automatischen Administration. Diese Jobvariable können Sie nutzen, um Benutzeraufträge und Programme zu steuern.

Die Jobvariable UDSSQL.DBH.*session* wird folgendermaßen versorgt:

Spalte	Inhalt	Bedeutung
1-30	UDSDBH_STARTING / UDSDBH_ACTIVE / UDSDBH_CLOSE_INITIATED / UDSDBH_NORMAL_END / UDSDBH_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> / <i>␣</i>	Session-Abschnittsnummer (4)
93	B / <i>␣</i>	Zustand der DBH-Betriebsunterbrechung: – aktuelle Betriebsunterbrechung – aktuell keine Betriebsunterbrechung
94-96	DAL / DBH / <i>␣</i>	Art der aktuellen oder letzten DBH-Betriebsunterbrechung (5) – wegen DALs (PERFORM) – wegen systeminterner Aktivitäten – noch keine Betriebsunterbrechung aufgetreten

Tabelle 45: Aufbau der Session-Jobvariable für UDS/SQL

(Teil 1 von 2)

Spalte	Inhalt	Bedeutung
97	A / - / _	Zuschalten von Datenbank oder Realm bei aktueller oder letzter Betriebsunterbrechung (5) - ja - nein - noch keine Betriebsunterbrechung aufgetreten
98	D / - / _	Abschalten von Datenbank oder Realm bei aktueller oder letzter Betriebsunterbrechung (5) - ja - nein - noch keine Betriebsunterbrechung aufgetreten
99	A / - / _	Aktivität bzgl. ALOG-Dateien bei aktueller oder letzter Betriebsunterbrechung (5) - ja - nein - noch keine Betriebsunterbrechung aufgetreten
100	R / - / _	Aktivität bzgl. RLOG-Dateien bei aktueller oder letzter Betriebsunterbrechung (5) - ja - nein - noch keine Betriebsunterbrechung aufgetreten
101-112	_	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	_	reserviert
163-172 173-180	<i>jjjj-mm-tt</i> <i>hh:mm:ss</i>	Letzte Änderung in Jobvariable Datum Uhrzeit

Tabelle 45: Aufbau der Session-Jobvariable für UDS/SQL

(Teil 2 von 2)

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* wrld 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* wrld 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.

10.5.3 Datenbank-Jobvariable

(siehe Handbuch „[Datenbankbetrieb](#)“, Abschnitt 9.2.3)

Der DBH und die Dienstprogramme DDL-Compiler, SSL-Compiler, BGSIA, BGSSIA, BPRIVACY, BMEND, BREORG, BCHANGE, BRENAM und BALTER versorgen eine Datenbank-Jobvariable zur Verbesserung der automatischen Administration. Diese Jobvariable können Sie nutzen, um Benutzeraufträge und Programme zu steuern.

Die Datenbank-Jobvariable UDSSQL.DB.*datenbankname*.[*copyname*][*.nr*] 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> / <i>┘</i>	Copyname (2)
27-32	<i>cccccc</i>	DB-Layout-Version
33	C / I / <i>┘</i>	Konsistenz (3)
34-40	<i>┘</i>	reserviert
41-50	UPDATE / RETR/ WARMSTART/ OPEN / DROP / CLOSE / ERROR	Verarbeitungszustand (4): – Zuschaltmodus beim DBH – Zuschaltmodus bei einem Dienstprogramm – Ursache der letzten Beendigung bei DBH oder einem Dienstprogramm
51	<i>┘</i> / A	Aktivierung ALOG (5)
52	<i>┘</i> / 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 – 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> / <i>┘</i>	Konfigurationsname der DBH-Session (8)
69-72	<i>cccc</i> / <i>┘</i>	Standardkatalog der DBH Kennung (9)
73-81	<i>\$cccccccc</i> / <i>┘</i>	Kennung der DBH-Session (mit \$) (9)
82-89	<i>nnnnnnnn</i> / <i>┘</i>	Sessionabschnittsnummer (10)
90-99 100-107	<i>jjjj-mm-tt</i> <i>hh:mm:ss</i>	Verarbeitungsbeginn (11) Datum Uhrzeit

Tabelle 46: Aufbau der Datenbank-Jobvariable für UDS/SQL

(Teil 1 von 2)

Spalte	Inhalt	Bedeutung
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>nnnnnnnnn</i> / <i>┘</i>	ALOG-Folgenummer (14)
153-162	<i>nnnnnnnnnn</i> / <i>┘</i>	Größe der ALOG-Datei (15)
163-182	<i>┘</i>	reserviert
183-192 193-200	<i>jjjj-mm-tt</i> <i>hh:mm:ss</i>	Letzte Änderung in Jobvariable Datum Uhrzeit

Tabelle 46: Aufbau der Datenbank-Jobvariable für UDS/SQL

(Teil 2 von 2)

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 ALOG* zeigt an, ob für die Datenbank ALOGGING eingeschaltet ist ("A") oder nicht (Leerzeichen). Die aktuelle Verarbeitung im DBH kann ohne ALOGGING 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.

11 Dienstprogramme

11.1 START-Kommandos der UDS/SQL-Programme

Syntax der START-UDS-...-Kommandos

START-UDS-...
VERSION = *STD / <product-version>
,MONJV = *NONE / <filename 1..54 without-gen-vers>
,CPU-LIMIT = *JOB-REST / <integer 1..32767 seconds>
,RESIDENT-PAGES = [*PARAMETERS] (...) <i>Nur für DBH</i>
[*PARAMETERS] (...)
MINIMUM = *STD / <integer 0..32767 4Kbyte>
,MAXIMUM = *STD / <integer 0..32767 4Kbyte>

Die folgende Tabelle zeigt die START-Kommandos und deren Alias-Namen, mit denen Sie die angegebenen UDS/SQL-Programme aufrufen können.

Folgende Voraussetzungen müssen erfüllt sein:

- UDS/SQL muss mit IMON installiert sein und
- die SDF-Systemsyntaxdatei muss aktiviert sein.

Programm	START-Kommando	Alias-Namen
UDSSQL	START-UDS-DBH	UDS, SYSINT
BALTER	START-UDS-BALTER	BALTER
BCALLSI	START-UDS-BCALLSI	BCALLSI
BCHANGE	START-UDS-BCHANGE	BCHANGE
BRENAME	START-UDS-BRENAME	BRENAME
BCHECK	START-UDS-BCHECK	BCHECK
BCREATE	START-UDS-BCREATE	BCREATE
BFORMAT	START-UDS-BFORMAT	BFORMAT
BGSIA	START-UDS-BGSIA	BGSIA
BGSSIA	START-UDS-BGSSIA	BGSSIA
BINILOAD	START-UDS-BINILOAD	BINILOAD
BMEND	START-UDS-BMEND	BMEND, START-UDS-REPAIR
BMODTT	START-UDS-BMODTT	BMODTT
BOUTLOAD	START-UDS-BOUTLOAD	BOUTLOAD, START-UDS-OUTLOAD
BPGSIZE	START-UDS-BPGSIZE	BPGSIZE START-UDS-PAGE-RESIZING
BPRECORD	START-UDS-BPRECORD	BPRECORD
BPRIVACY	START-UDS-BPRIVACY	BPRIVACY START-UDS-AUTHORIZATION
BPSIA	START-UDS-BPSIA	BPSIA
BPSQLSIA	START-UDS-BPSQLSIA	BPSQLSIA, START-UDS-PRINT-SQLSIA
BREORG	START-UDS-BREORG	BREORG, START-UDS-REORGANIZATION
BSTATUS	START-UDS-BSTATUS	BSTATUS
DDL	START-UDS-DDL	DDL
DMLTEST	START-UDS-DMLTEST	DMLTEST
SSL	START-UDS-SSL	SSL
ONLINE-PRIVACY	START-UDS-ONLINE-PRIVACY	OPRIVACY
UDSADM	START-UDS-ADM	UDSADM, START-UDS-ADMINISTRATION
UDSMON	START-UDS-UDSMON	UDSMON
UDS-Online-Utility	START-UDS-ONLINE-UTILITY	ONLUTIL

Tabelle 47: UDS/SQL-Programme über START-Kommandos aufrufen

11.2 BALTER

BALTER wird sowohl in einem Umstrukturierungszyklus genutzt, der durch BCHANGE eingeleitet wird (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 6.11), als auch in einem Umbenennungszyklus, der durch BRENAME eingeleitet wird (siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 7.8).

Anweisungen im Umstrukturierungszyklus

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 6.11.4)

Anweisung	Standardwert	Bedeutung
[<u>SORTCORE</u> IS <i>nnn</i> .]	150	Größe des Sortierpuffers festlegen
<u>EXECUTION</u> IS $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$.	-	Umstrukturierungsphase starten/nicht starten
<u>REPORT</u> IS $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$.	-	Protokollierung anfordern/unterbinden
[<u>FILLING</u> IS <i>nnn</i> PERCENT [IN <u>SET</u> NAME IS $\left\{ \begin{array}{l} \text{setname}, \dots \\ *ALL[\text{EXCEPT } \text{setname}, \dots] \end{array} \right\}]$.]	-	Füllgrad von Tabellen festlegen (Format 1)
[<u>FILLING WITH POPULATION</u> [IN <u>SET</u> NAME IS $\left\{ \begin{array}{l} \text{setname}, \dots \\ *ALL[\text{EXCEPT } \text{setname}, \dots] \end{array} \right\}]$.]	-	Füllgrad von Tabellen festlegen (Format 2)
<u>END</u> .	-	beendet die Eingabe der Anweisungen

Tabelle 48: Anweisungen für BALTER im Umstrukturierungszyklus

Anweisungen im Umbenennungszyklus

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 7.8)

Anweisung	Standardwert	Bedeutung
<u>REPORT</u> IS { <u>YES</u> } { <u>NO</u> } .	-	Protokollierung anfordern/unterbinden
<u>END</u> .	-	beendet die Eingabe der Anweisungen

Tabelle 49: Anweisungen für BALTER im Umbenennungszyklus

Kommandofolge zum Starten von BALTER in einer Umstrukturierungsphase(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 6.11.5)*Analysephase*

```

01  /ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR
02  /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
03  /START-UDS-BALTER
04  EXECUTION IS NO.
05  REPORT IS YES.
06  END.

```

Umstrukturierungsphase

```

01  /ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR
02  [/CREATE-FILE FILE-NAME=arbeitsdatei-1 ...
    /ADD-FILE-LINK LINK-NAME=SCRATCH1,FILE-NAME=arbeitsdatei-1
      ,ACCESS-METHOD=*UPAM]
03  [/CREATE-FILE FILE-NAME=arbeitsdatei-2 ...
    /ADD-FILE-LINK LINK-NAME=SORTWK,FILE-NAME=arbeitsdatei-2
      ,ACCESS-METHOD=*UPAM]
04  /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
05  /START-UDS-BALTER
06  [SORTCORE IS nnn.]
07  [FILLING IS nnn PERCENT
      [ IN SET NAME IS { setname,... } ]].]
08  EXECUTION IS YES.
09  REPORT IS { YES }
              { NO  }.
10  END

```

Arbeitsdateien in einer Umstrukturierung einrichten

(siehe Handbuch „Aufbauen und Umstrukturieren“, Abschnitt 6.11.3)

- SCRTCH1

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich annähernd aus der folgenden Formel:

$$\max(\text{schlüsselänge} \times \text{anzahl sätze}) \times 3 \text{ Bytes}$$

- SORTWK

Das Mengengerüst der zu sortierenden Daten ergibt sich aus der folgenden Formel:

$$\max(\text{satzlänge} \times \text{anzahl sätze}) \text{ Bytes}$$

Richten Sie die beiden Arbeitsdateien nicht selbst ein, so richtet BALTER sie mit folgenden Namen und Größen ein:

UTI.*tsn*.SCRTCH1 (360,360)

UTI.*tsn*.SORTWK (120,120)

11.3 BCALLSI

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.5)

Anweisungen

Anweisungen	Standardwert	Bedeutung
$\left\{ \begin{array}{l} \underline{\text{SCHEMA}} \\ \underline{\text{S}} \end{array} \right\} = \textit{schemaname},$ $\left\{ \begin{array}{l} \underline{\text{SUBSCHEMA}} \\ \underline{\text{SS}} \end{array} \right\} = \textit{subschemaname}$	-	<p>generell erforderlich; weist BCALLSI die Namen des Schemas und des Subschemas zu</p> <p><i>schemaname</i> Name des Schemas, der in der Schema-DDL vergeben wurde</p> <p><i>subschemaname</i> Name des Subschemas, der in der Subschema-DDL vergeben wurde</p>
$\left[\left\{ \begin{array}{l} \underline{\text{MESSAGE}} \\ \underline{\text{M}} \end{array} \right\} = \left\{ \begin{array}{l} \underline{\text{*ALL}} \\ \underline{\text{N[O-AMBIGUITY-8]}} \end{array} \right\} \right]$	*ALL	<p>*ALL Alle Eindeutigkeits-Verletzungen, auch in den ersten 8 Zeichen, werden auf SYSLST einzeln ausgegeben.</p> <p>NO-AMBIGUITY-8 Eindeutigkeits-Verletzungen in den ersten 8 Zeichen eines Namens werden nicht einzeln ausgegeben</p>

Tabelle 50: Anweisungen für BCALLSI

Kommandofolge zum Starten von BCALLSI

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
03 /START-UDS-BCALLSI
04 bcallsi-anweisung
```

SSITAB-Modul in die Modulbibliothek eintragen

```
05 /START-LMS
06 //OPEN-LIB LIB=modlib, MODE=*UPDATE
07 //ADD-ELEMENT FROM-FILE=*OMF, TO-ELEMENT=*LIBRARY-ELEMENT(TYPE=R)
08 //END
```

04 Bei BCALLSI gibt es keine END-Anweisung!

11.4 BCHANGE

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 6.7)

Kommandofolge zum Starten von BCHANGE

Das Dienstprogramm BCHANGE starten Sie in der Kennung, unter der die Datenbank katalogisiert ist, mit folgenden Kommandos:

```
01 [/CREATE-FILE FILE-NAME=dbname.DBCOM.0 ...]
02 [/CREATE-FILE FILE-NAME=dbname.COSSD.0 ...]
03 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
05 /START-UDS-BCHANGE
```

01,02 Siehe Abschnitt „Compilerdatenbank einrichten“, Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.1.1.



BCHANGE hat keine Anweisungen!

11.5 BCHECK

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 3)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 3.4)

Anweisung	Standardwert	Bedeutung
<code>[SORTCORE IS <i>n</i>]</code>	150	Größe des Sortierpuffers festlegen
<code>CHECK [{ [GENERATE]SORTING }] [SUMMING] [AGAINST COPY NAME IS <i>copyname</i>]</code>	SUMMING	Prüfmodus wählen und Umfang der Prüfung festlegen
<code>[TYPE IS { ALL [EXCEPT <i>type-nr-1</i> [, <i>type-nr-2</i>] ...] } 0 { <i>type-nr-1</i> [, <i>type-nr-2</i>] ... }]</code>	ALL	Kriterien für die globale Konsistenzprüfung wählen
<code>SCHEMA NAME IS <i>schemaname</i></code>	-	Schema bezeichnen
<code>REALM NAME IS { ALL [EXCEPT <i>realmname-1</i>, ...] } { <i>realmname-2</i>, ... }</code>	-	Zu prüfende Realms spezifizieren
<code>[RECORD NAME IS { ALL [EXCEPT <i>satzname-1</i>, ...] } { <i>satzname-2</i>, ... } [{ WITH } { WITHOUT } { LOCATION CHECK } { KEYVALUE CHECK }]]</code>	-	Zu prüfende Satzarten spezifizieren

Tabelle 51: Anweisungen für BCHECK

(Teil 1 von 2)

Anweisung	Standardwert	Bedeutung
<pre>[<u>SET</u> NAME IS {<u>ALL</u>[<u>EXCEPT</u> setname-1,...]} {setname-2,...} [{<u>WITH</u> {<u>WITHOUT</u>} <u>INDEX</u> CHECK]]</pre>	WITHOUT INDEX CHECK	Zu prüfende Sets spezifizieren
<pre>[<u>KEY</u> REF IS {<u>ALL</u>[<u>EXCEPT</u> keyref-1,...]} {keyref-2,...} [{<u>WITH</u> {<u>WITHOUT</u>} <u>INDEX</u> CHECK]]</pre>	WITHOUT INDEX CHECK	Zu prüfende SEARCH-Keys spezifizieren

Tabelle 51: Anweisungen für BCHECK

(Teil 2 von 2)

Kommandofolge zum Starten von BCHECK

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 3.5)

Abhängig vom Prüflauf benötigen Sie folgende Kommandos zum Starten von BCHECK:

```
01 [/CREATE-FILE FILE-NAME=arbeitsdatei-1 ...
    /ADD-FILE-LINK LINK-NAME=SCRATCH1, FILE-NAME=arbeitsdatei-1,
    ACCESS-METHOD=*SAM ]

02 [/CREATE-FILE FILE-NAME=arbeitsdatei-2 ...
    /ADD-FILE-LINK LINK-NAME=SORTWK, FILE-NAME=arbeitsdatei-2,
    ACCESS-METHOD=*UPAM ]

03 /ADD-FILE-LINK LINK-NAME=DATABASE,
    FILE-NAME=[ :catid: ][ $userid. ]dbname.DBDIR[ .copyname ]

04 [/ADD-FILE-LINK LINK-NAME=BCHECK,
    FILE-NAME=[ :catid: ]UTI.tsn.zeitstempel.BCHECK ]

05 [/ASSIGN-SYSLST TO=dateiname]

06 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK

07 /START-UDS-BCHECK

08 bcheck-anweisungen

09 END

10 [/ASSIGN-SYSLST TO=*PRIMARY]
```

Arbeitsdateien einrichten

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 3.2)

- SCRTCH1

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich aus der Formel:

$$\textit{seitenanzahl} \times 16 \text{ Bytes}$$

- SORTWK

Das Mengengerüst der zu sortierenden Daten ist bei Totalprüfung mit den beiden folgenden Formeln ermittelbar:

- Formel RSQ-Prüfung:

$$26 \times \textit{anzahl-prüfsätze} \text{ Bytes}$$

- Formel Indexwertprüfung und Schlüsselwertprüfung:

$$\textit{schlüsselänge} \times \textit{anzahl-prüfsätze} \text{ Bytes}$$

11.6 BCREATE

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.2.1)

Kommandofolge zum Starten von BCREATE

```
01 /CREATE-FILE FILE-NAME=dbname.DBDIR ...
02 /CREATE-FILE FILE-NAME=dbname.DBCOM ...
03 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
05 /START-UDS-BCREATE
06 [DATABASE-PAGE-LENGTH IS {2/4/8}KB]
07 END
```

01/02 siehe „DBDIR und DBCOM einrichten“, Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.1.1.

11.7 BFORMAT

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.3)

Anweisungen

Die REALM-Anweisung von BFORMAT benennt die Realms, die formatiert werden sollen. Realms können in mehreren BFORMAT-Läufen formatiert werden. Jeder Realm kann nur einmal formatiert werden.

Erst wenn alle Realms formatiert sind, können Sie den Datenbankaufbau fortsetzen.

Anweisung	Standardwert	Bedeutung
<pre>REALM NAME IS { ALL [EXCEPT] [realmname1,..] realmname1,.. }</pre>	ALL	<p>wahlweise; die angegebenen Realms sollen formatiert/nicht formatiert werden;</p> <p>ALL alle Realms, die in der Schema-DDL definiert wurden, sollen formatiert werden</p> <p>ALL EXCEPT <i>realmname</i> Negativliste, d.h. alle Realms, außer dem (den) benannten Realm(s), sollen formatiert werden</p> <p><i>realmname</i> bezeichnet einen Benutzerrealm</p>
END	-	generell erforderlich; beendet die Eingabe der Anweisungen

Tabelle 52: Anweisungen für BFORMAT



Es ist sinnvoll, Realms einzeln zu formatieren. Bei mehreren Realms betrifft ein Abbruch von BFORMAT auf Grund eines Betriebssystemfehlers ohne Endebehandlung auch bereits erfolgreich formatierte Realms. Der BFORMAT-Lauf muss dann auch für diese Realms wiederholt werden.

Der BFORMAT-Lauf ist sehr schnell, da nur Hashbereiche, FPA- und DBTT-Seiten formatiert werden.

Kommandofolge zum Starten von BFORMAT

```
01 /CREATE-FILE FILE-NAME=dbname.realmname ...
02 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
03 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
04 /START-UDS-BFORMAT
05 [bformat-anweisung]
06 END
```

01 siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.1.2.

05 Wenn Sie die REALM-Anweisung nicht angeben, werden alle Realms formatiert.

11.8 BGSIA

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.2.4)

Anweisungen

Anweisung	Standardwert	Bedeutung
<u>GENERATE</u> SCHEMA <i>schemaname</i>	-	muss angegeben werden; prüfen und erstellen der SIA <i>schemaname</i> : Name des Schemas, der in der Schema-DDL angegeben wurde
<u>RENAME</u> { <u>AREA</u> <u>RECORD</u> <u>SET</u> } { 'name-old' TO 'name-new' } [, ...] .		darf nur im Umbenennungszyklus angegeben werden; Ändert die Namen von Satzarten, Sets und Benutzerrealms <i>name-old</i> : Name der geändert werden soll <i>name-new</i> neuer Name Umbenennungen und Änderungen von Feldern in Satzarten können hier nicht angegeben werden.
<u>DISPLAY</u> [SCHEMA <i>schemaname</i>]	-	wahlweise: ausdrucken der von BGSIA erzeugten SIA <i>schemaname</i> : Name des Schemas, der in der GENERATE-Anweisung angegeben wurde Die Angabe DISPLAY genügt
<u>END</u>	-	generell erforderlich; schließt die Eingabe der Anweisungen ab

Tabelle 53: Anweisungen für BGSIA

Kommandofolge zum Starten von BGSIA

```
01 /DELETE-SYSTEM-FILE FILE-NAME=*OMF
02 /ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR
03 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
04 /START-UDS-BGSIA
05 bgsia-anweisungen
06 END
```

Modul UDSHASH in die HASHLIB eintragen

```
01 /START-LMS
02 //OPEN-LIB LIB=dbname.HASHLIB,MODE=*UPDATE(STATE=*NEW)
03 //ADD-ELEMENT FROM-FILE=*OMF,TO-ELEMENT=*LIBRARY-ELEMENT(TYPE=R)
04 //END
```

11.9 BGSSIA

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.4.2)

Anweisungen

Anweisung	Standardwert	Bedeutung
<u>GENERATE</u> SUBSCHEMA <i>subschema</i> <u>name</u> <u>OF</u> SCHEMA <i>schemaname</i>	-	wahlweise; <i>subschema</i> <u>name</u> : Name des Subschemas <i>schemaname</i> : Name des Schemas Prüfen, ob bereits eine SSIA für ein bestimmtes Subschema vorhanden ist und Erstellen von: <ul style="list-style-type: none"> – SSIA mit einzelnen Informationen über Realms, Satzarten und Sets – Listen der einzelnen Felder – Listen aller im Subschema auftretenden Namen
<u>DELETE</u> SUBSCHEMA <i>subschema</i> <u>name</u> <u>OF</u> SCHEMA <i>schemaname</i>	-	wahlweise; Löschen einer früher erzeugten SSIA aus dem DBDIR
<u>REGENERATE</u> SUBSCHEMA <i>subschema</i> <u>name</u> <u>OF</u> SCHEMA <i>schemaname</i>	-	wahlweise; Löschen der alten SSIA und neu Erstellen der SSIA (Funktion DELETE und GENERATE zusammengefasst); geeignet zum Korrigieren eines Subschemas
<u>DISPLAY</u> [SUBSCHEMA <i>subschema</i> <u>name</u> <u>OF</u> SCHEMA <i>schemaname</i>]	-	wahlweise; gilt nur in Verbindung mit der GENERATE-Anweisung oder REGENERATE-Anweisung: Ausdrucken der SSIA; die Angabe DISPLAY genügt
<u>END</u>	-	generell erforderlich; beendet die Eingabe der Anweisungen

Tabelle 54: Anweisungen für BGSSIA

Kommandofolge zum Starten von BGSSIA

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
03 /START-UDS-BGSSIA
04 bgssia-anweisungen
05 END
```

11.10 BINILOAD

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.1)

Anweisungen

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.1.4)

Steueranweisungen

steuern den Ablauf des UDS/SQL-Dienstprogramms BINILOAD

Anweisung	Standardwert	Bedeutung
[<u>EXECUTION</u> { <u>WITH</u> <u>WITHOUT</u> } <u>CHECK</u> .]	WITH	Eingaben prüfen/nicht prüfen
[<u>SortCORE</u> IS <i>nnn</i> .]	150	Größe des Hauptspeichers für das Sortier-/Mischprogramm festlegen

Tabelle 55: Steueranweisungen des BINILOAD

Programmanweisungen

bestimmen Schema, Subschema, Eingabedatei und den Füllgrad von Tabellen

Anweisung	Standardwert	Bedeutung
<u>SCHEMA</u> NAME IS <i>schemaname</i> . <u>SUBSCHEMA</u> NAME IS <i>sub schemaname</i> .	-	Name des Schemas und des Subschemas
<u>FILLING</u> IS <i>nnn</i> PERCENT.	-	Füllgrad für Tabellenseiten angeben
<u>USER</u> FILE <u>RECORD</u> <u>LENGTH</u> IS <i>n</i> .	-	Länge der Eingabesätze in byte angeben
<u>USER</u> FILE <u>BUFFER</u> <u>LENGTH</u> IS <i>n</i> .	-	Blocklänge der Eingabedatei; muss ein Vielfaches von 2048 sein
<u>INPUT</u> FILE NAME ' <i>dateiname</i> '.	-	Dateiname der Eingabedatei
<u>INPUT</u> <u>RECORDNUMBER</u> IS <i>n</i> .	-	keine, wird aus Kompatibilitätsgründen geduldet

Tabelle 56: Programmanweisungen des BINILOAD

STORE-Anweisungen

informieren BINILOAD über die Satzart und deren Zusammenhang mit den Eingabesätzen

Anweisung	Standardwert	Bedeutung
<u>STORE RECORD</u> NAME IS <i>satzname</i> .	-	Satzart, die gespeichert werden soll
<u>RECORD-DBKEY</u> IS <u>DISPL</u> IS <i>n</i> , <u>LENGTH</u> IS $\left\{ \begin{array}{l} 4 \\ 8 \end{array} \right\}$.	-	Database-Key-Wert vergeben; – Position und Länge des Data Base Key-Wertes
<u>RECORD-RSQ</u> IS <u>DISPL</u> IS <i>n</i> , <u>LENGTH</u> IS $\left\{ \begin{array}{l} 3 \\ 6 \end{array} \right\}$.	-	Database-Key-Wert vergeben; – Position und Länge der Satzfolgennummer (RSQ) Die zugehörige Satzartnummer (REC-REF) wird von BINILOAD ermittelt.
<u>RECORD-DISPL</u> IS <i>n</i> , $\left\{ \begin{array}{l} \text{DISPL IS } n, \text{LENGTH IS } n \\ \text{VALUE IS 'literal'} \end{array} \right\}$.	-	Datenbanksatz aufbauen zur angegebenen Satzart; – Position und Länge von Feldern dieses Satzes – Zeichenfolge, die in die Datenbanksätze eingefügt werden soll
<u>RECORD-AREA</u> NAME IS <i>realmname</i> .	-	Realm, in den die Sätze geladen werden sollen

Tabelle 57: STORE-Anweisungen des BINILOAD

INSERT-Anweisungen

nennen BINILOAD die Sets, in die die Sätze einzufügen sind

Anweisung	Standardwert	Bedeutung
<u>INSERT INTO SET NAME IS setname.</u>	-	den Set angeben, in den die Sätze als Membersätze eingefügt werden sollen;
<u>SET ORDER</u> { <u>USING DISPL IS n, LENGTH IS n</u> } { <u>VIA USER FILE SEQUENCE</u> }	VIA USER FILE SEQUENCE	Sortierfolge der Sätze innerhalb der Sets mit ORDER IS FIRST, LAST, NEXT, PRIOR, IMMATERIAL; Länge des Sortierfeldes
<u>OWNER CALCKEY IS</u> { <u>DISPL IS n, LENGTH IS n</u> } { <u>VALUE IS 'literal'</u> } <u>AREA NAME IS realmname.</u>	-	Set-Occurrence auswählen durch Auswählen des Owner: – Position und Länge der CALC-Key-Werte in den Sätzen der Eingabedatei, mit denen der Owner ausgewählt werden soll – Zeichenfolge mit CALC-Key – Name des Realm, in dem der Ownersatz gespeichert ist
<u>OWNER SEARCHKEY IS</u> { <u>DISPL IS n, LENGTH IS n</u> } { <u>VALUE IS 'literal'</u> } [<u>VIA SET NAME IS setname,</u>] <u>SEARCHKEY TABLE</u> { <u>COLUMN-NR IS n</u> } { <u>ORDER-NR IS keyref</u> }	-	Set-Occurrence auswählen durch Auswählen des Owner über SEARCH-Key: – Position und Länge der SEARCH-Key-Werte in den Sätzen der Eingabedatei, mit denen der Owner ausgewählt werden soll – Zeichenfolge mit SEARCH-Key-Tabelle – Name des SYSTEM-Set, in dem der Owner Member ist – DBTT-Column-Nr der SEARCH-Key-Tabelle – Schlüsselnummer

Tabelle 58: INSERT-Anweisungen des BINILOAD

(Teil 1 von 2)

Anweisung	Standardwert	Bedeutung
<p><u>OWNER DBKEY</u> IS</p> $\left\{ \begin{array}{l} \text{DISPL IS } n, \text{ LENGTH IS } \left\{ \begin{array}{l} 4 \\ 8 \end{array} \right\} \\ \text{VALUE IS } dbkey \end{array} \right\}.$	-	<p>Set-Occurrence auswählen durch Auswählen des Owner über seinen Database-Key-Wert:</p> <ul style="list-style-type: none"> - Position und Länge des Database-Key-Werts in den Sätzen der Eingabedatei, mit denen der Owner ausgewählt werden soll - Zeichenfolge mit Database-Key-Wert
<p><u>OWNER RSQ</u> IS</p> $\left\{ \begin{array}{l} \text{DISPL IS } n, \text{ LENGTH IS } \left\{ \begin{array}{l} 3 \\ 6 \end{array} \right\} \\ \text{VALUE IS } rsq \end{array} \right\}.$	-	<p>Set-Occurrence auswählen durch Auswählen des Owner über seinen Database-Key-Wert:</p> <ul style="list-style-type: none"> - Position und Länge der Satzfolgennummer (RSQ) in den Sätzen der Eingabedatei, mit denen der Owner ausgewählt werden soll - Zeichenfolge mit Satzfolgennummer (RSQ). <p>Die zugehörige Satzartnummer (REC-REF) wird von BINILOAD ermittelt.</p>
<p><u>OWNER KEY</u> IS <u>DISPL</u> IS <i>n</i>, <u>LENGTH</u> IS 1.</p>	-	<p>Position des Feldes in den Eingabesätzen, das angibt, ob der Satz in den SYSTEM-Sets eingehängt werden soll.</p>

Tabelle 58: INSERT-Anweisungen des BINILOAD

(Teil 2 von 2)

Kommandofolge zum Starten von BINILOAD

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.1.5)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
02 [/CREATE-FILE FILE-NAME=eingabebanddatei,...]
03 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
04 /START-UDS-BINILOAD
05 biniload-anweisungen
06 END
```



Die BINILOAD-Anweisungen werden über SYSDTA eingelesen! Dabei kann es sich auch um eine Datei handeln, die mit BOUTLOAD erzeugt wurde.

Arbeitsdateien einrichten

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.1.6)

Wollen Sie die Arbeitsdateien explizit einrichten, so müssen Sie die entsprechenden CREATE-FILE-Kommandos angeben. Geben Sie einen zu geringen Speicherplatzwert mit SPACE an, so wird er intern durch BINILOAD korrigiert.

```
/CREATE-FILE FILE-NAME=arbeitsdatei-n [,SUPPORT ...]
```

```
/ADD-FILE-LINK LINK-NAME=
{
  SCRTCH1
  SCRTCH2
  SCRTCH3
  SCDnnnnn
  STKnnnnn
  KEYnnnnn
  KSTnnnnn
  SORTWK
  SRT1WK
}
, FILE-NAME=arbeitsdatei-n [, ACCESS-METHOD=
{
  *UPAM
  *SAM
}
]
```

SCRTCH1

(gesamtschlüssellänge + 12) x anzahl der eingabesätze Bytes

SCRTCH2

12 x anzahl der eingabesätze Bytes

SCRTCH3

3 x anzahl der eingabesätze Bytes

SCD*nnnnn*

bei 2048 byte Seitenlänge:

40 x anzahl der eingabesätze Bytes

bei 4000/8096 byte Seitenlänge:

50 x anzahl der eingabesätze Bytes

STKnnnnn

bei 2048 byte Seitenlänge:

$8 + \text{satzlänge}_1) \times \text{anzahl der eingabesätze}$ Bytes

bei 4000/8096 byte Seitenlänge:

$(12 + \text{satzlänge}_1) \times \text{anzahl der eingabesätze}$ Bytes

KEYnnnnn und bei SEARCH-Key

bei 2048 byte Seitenlänge:

$(16 + \text{schlüsselänge}_1) \times \text{anzahl der eingabesätze}$ Bytes

bei 4000/8096 byte Seitenlänge:

$(24 + \text{schlüsselänge}_1) \times \text{anzahl der eingabesätze}$ Bytes

KEYmmmmm und KSTnnnnn bei SORT-Key

bei 2048 byte Seitenlänge:

$(\text{schlüsselänge}_1 + 12 + \text{schlüsselänge}_2) \times$
 $\text{anzahl der eingabesätze}$ Bytes

bei 4000/8096 byte Seitenlänge:

$(\text{schlüsselänge}_1 + 16 + \text{schlüsselänge}_2) \times$
 $\text{anzahl der eingabesätze}$ Bytes

SORTWK und SRT1WK

Das Mengengerüst der zu sortierenden Daten ergibt sich aus der Formel:

$(\text{satzlänge}_2 + \text{SCD} + 12) \times \text{anzahl der eingabesätze}$ Bytes

11.11 BMEND

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 2)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 2.2.3)

Anweisung	Bedeutung
ADD-REALM	Realms zu einer Datenbank zuschalten
REALM-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(30): <realmname> *ALL-EXCEPT(...) NAME = list-poss(30): <realmname>	
ALLOCATE-BUFFER-POOL	Puffergröße festlegen
BUFFER-SIZE = *STD / <integer 1..2000>	
DISABLE-ONLINE-COPY	Online-Sicherungsfähigkeit der Datenbank zurücknehmen
ENABLE-ONLINE-COPY	Datenbank als online-sicherungsfähig markieren
END	Eingabe der Anweisungen beenden
KILL-LOG	Logging bei inkonsistenter Datenbank ausschalten
OPEN-DATABASE	Datenbank eröffnen
DATABASE-NAME = <dbname> ,COPY-NAME = *NONE / <copyname> ,USER-IDENTIFICATION = *OWN / <userid>	

Tabelle 59: Anweisungen für BMEND

(Teil 1 von 4)

Anweisung	Bedeutung
REMOVE-REALM REALM-NAME = *ALL-EXCEPT(...) / list-poss(30): <realmname> *ALL-EXCEPT(...) NAME = list-poss(30): <realmname>	Realms abschalten
SHOW-LOG-INFORMATION REALM-NAME = * <u>ALL</u> / *ALL-EXCEPT(...) / list-poss(30): <realmname> *ALL-EXCEPT(...) NAME = list-poss(30): <realmname> ,LOG-FILE = * <u>STD</u> / NONE / <alog-seq-nr> ,OUTPUT = list-poss: * <u>SYSLST</u> / *SYSOUT	

Tabelle 59: Anweisungen für BMEND

(Teil 2 von 4)

Anweisung	Bedeutung
<pre> START-LOG DEFAULT-SUPPORT = *PUBLIC(...) / *UNCHANGED / list-poss(15): *PRIVATE(...) *PUBLIC(...) CATID = *STD / *OWN / <catid> ,VOLUME-SET = *STD / <catid> ,VOLUME = *STD / list-poss(15): <volume> *PRIVATE(...) VOLUME = list-poss(15): <volume> ,DEVICE = <device> ,RESERVE-SUPPORT = *PUBLIC(...) / *UNCHANGED / list-poss(15): *PRIVATE(...) *PUBLIC(...) CATID = *STD / *OWN / <catid> ,VOLUME-SET = *STD / <catid> ,VOLUME = *STD / list-poss(15): <volume> *PRIVATE(...) VOLUME = list-poss(15): <volume> ,DEVICE = <device> ,SPACE = STD / *RELATIVE(...) / *UNCHANGED *RELATIVE(...) PRIMARY-ALLOCATION = <integer 192..50331645> ,SECONDARY-ALLOCATION = <integer 576..32767> ,USER-ACCESS = *OWNER-ONLY / *ALL-USERS ,RESET-LOG-POOL = *NO / *YES </pre>	<p>Logging für das Datenbank-Original einschalten</p>
<pre> STOP-LOG </pre>	<p>Logging für den Datenbankbetrieb ausschalten</p>
<pre> UNDO </pre>	<p>Anweisungen rückgängig machen</p>

Tabelle 59: Anweisungen für BMEND

(Teil 3 von 4)

Anweisung	Bedeutung
UPDATE-DATABASE REALM-NAME = <u>*ALL</u> / *ALL-EXCEPT(...) / list-poss(30): <realmname> *ALL-EXCEPT(...) NAME = list-poss(30): <realmname> ,DEADLINE = <u>*STD</u> / BREAK-POINT / <alog-seq-nr> / *TIME-STAMP(...) *TIME-STAMP(...) DATE = <date> ,TIME = <time> ,DELETE = <u>*NO</u> / *YES	AFIMs in eine Datenbank einspielen

Tabelle 59: Anweisungen für BMEND

(Teil 4 von 4)

Kommandofolge zum Starten von BMEND

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 2.2.4)

```

01 [/CREATE-JV-LINK JV-NAME=JOBVAR,PROTECTION=*STD]
02 [/SET-JV-LINK LINK-NAME=JVBMEND,JV-NAME=JOBVAR]
03 [/ADD-FILE-LINK LINKNAME=DATABASE,
    FILE-NAME=[ :catid:][ $userid.]dbname.DBDIR]
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
05 /START-UDS-BMEND
06 [//OPEN-DATABASE DATABASE-NAME=dbname
    [, COPYNAME=*NONE/copyname]
    [, USER-IDENTIFICATION=*OWN/userid]
07 bmend-anweisungen
08 //END

```

03, 06 Sie müssen eine der beiden Anweisungen angeben.

11.12 BMODTT

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 10)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 10.2)

Anweisung	Bedeutung
<pre> {KEEP} {DBKEY OF RECORD} {REMOVE} {OF RECORD} {REUSE} {satzname-1[,satzname-2]... {*ALL[EXCEPT satzname-1[,satzname-2]...]} </pre>	<ul style="list-style-type: none"> - KEEP: freigewordene Database Keys sperren - REMOVE: die gesperrten Database Keys einmalig zur Wiederverwendung freigeben - REUSE: Database Keys grundsätzlich zur Wiederverwendung freigeben
<pre> {SET} REUSE-FREE-SPACE OF REALM {RESET} {realmname-1[,realmname-2]... {*ALL[EXCEPT realmname-1[,realmname-2]...]} </pre>	<ul style="list-style-type: none"> - SET: bei der Freiplatzsuche ab der ersten Seite im Realm suchen, die noch freien Platz enthält - RESET: bei der Freiplatzsuche ab der ersten freien Seite suchen, auf die bis zum Ende des Realms keine teilweise gefüllten Seiten mehr folgen, sondern nur noch freie oder voll belegte Seiten

Tabelle 60: Anweisungen des BMODTT

Kommandofolge zum Starten von BMODTT

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 10.3)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,  
    FILE-NAME=[ :catid: ][ $userid. ]dbname.DBDIR[ .copyname ]  
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL , VERSION=version .SCOPE=*TASK  
03 /START-UDS-BMODTT  
04 bmodtt-anweisungen  
05 END
```

03 BMODTT darf nur unter der Kennung des Datenbankadministrators aufgerufen werden.

Das Dienstprogramm BMODTT ist wiederanlauffähig.

11.13 BOUTLOAD

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.2)

Anweisungen

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.2.5)

Anweisung	Bedeutung
COPY-RECORD RECORD-NAME = *ALL / list-poss(20): <recordname> / *ALL-EXCEPT(...) ,REALM-NAME = *ALL / <realmname> ,SET-INFORMATION = YES / NO ,CSV-OUTPUT = *NO / *YES	Alle Sätze der angegebenen Satzarten in die Ausgabedateien kopieren
END	BOUTLOAD beenden
EXPORT-RECORD RECORD-NAME = *ALL / list-poss(20): <recordname> / *ALL-EXCEPT(...) ,SET-INFORMATION = YES / NO	Alle Sätze der angegebenen Satzarten in die Ausgabedateien entladen
OPEN-DATABASE DATABASE-NAME = <dbname> ,COPY-NAME = *NONE / <copyname> ,USER-IDENTIFICATION = *OWN / <userid>	Datenbank zuweisen
REMOVE-RECORD RECORD-NAME = *ALL / list-poss(20): <recordname> / *ALL-EXCEPT(...) 	Alle Sätze der angegebenen Satzarten löschen

Tabelle 61: Anweisungen des BOUTLOAD

Kommandofolge zum Starten von BOUTLOAD

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.2.6)

```
01 [/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR]
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
03 /START-UDS-BOUTLOAD
04 [OPEN-DATABASE DATABASE-NAME=dbname]
05 boutload-Anweisungen
06 END
```

01,04 Sie müssen eine der beiden Zuweisungen für die Datenbank verwenden.

Ausgabedateien einrichten

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.2.2)

Die einzelnen Ausgabedateien für BOUTLOAD können Sie mit folgenden Kommandos einrichten:

```
/CREATE-FILE FILE-NAME=dbname.RECnnnnn[.mmmmm] [,SUPPORT= ...]
/ADD-FILE-LINK LINK-NAME=linkname,FILE-NAME=dbname.RECnnnnn[.mmmmm]
[,BUFFER-LENGTH=xxx][,FILE-SEQUENCE=*NEW]
```

Die auszugebende Datenmenge errechnen Sie so:

Anzahl der Sätze x Satzlänge Bytes

Die Länge der Sätze errechnen Sie so:

- für Sätze mit Setinformation in einer 2-Kbyte-Datenbank:

*Satzlänge = Satzlänge nach SIA-Protokoll - Länge der Systeminformation
+ 4 * (Anzahl der nicht-singulären Sets, in denen der Satz
Member ist + 1)*

*+ 1 * (Anzahl der singulären Sets, in denen der Satz Member
ist, außer bei MANDATORY AUTOMATIC Members)*

- für Sätze mit Setinformation in einer 4-Kbyte-/8-Kbyte-Datenbank:

*Satzlänge = Satzlänge nach SIA-Protokoll - Länge der Systeminformation
+ 8 * (Anzahl der nicht-singulären Sets, in denen der Satz
Member ist + 1)*

*+ 1 * (Anzahl der singulären Sets, in denen der Satz Member
ist, außer bei MANDATORY AUTOMATIC Members)*

- für Sätze ohne Setinformation:

Satzlänge = Satzlänge nach SIA-Protokoll - Länge der Systeminformation

Bei auf Realms verteilten Satzarten kommen zur Satzlänge noch fünf Bytes für die Area-Ref hinzu, wenn die Sätze aus mehreren Realms kopiert oder extrahiert werden.

Die Sätze werden nämlich immer in eine Ausgabedatei pro Satzart kopiert, wichtig für das Auftreten der Area-Reference ist also die Herkunft aus mehr als einem Realm.

Aufbau des Ausgabesatzes

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 5.2.2)

Wenn BOUTLOAD wegen der Anweisung SET-INFORMATION=YES die Set-Information mit ausgegeben hat, wird der Ausgabesatz in folgender Struktur angelegt:

Database Key	Feld	Database Keys aller Owner	Benutzerteil	Area-Ref
--------------	------	---------------------------	--------------	----------

- der Database Key des Satzes
- ein ein Byte langes Feld mit dem Inhalt
X'00' = Member eingefügt
X'FF' = Member nicht eingefügt
(für alle singulären Sets, in denen der Satz Member ist, außer bei MANDATORY AUTOMATIC Members)
- die Database Keys der Owner der nicht singulären Sets, in denen der Satz Member ist
- Ist der Satz nicht im Set eingehängt, wird der Database Key des Owners auf High-Value gesetzt (X'FFFFFFFF' bei einer 2-KB-Datenbank, bzw. X'FFFFFFFFFFFFFFFF' bei einer 4/8-KB-Datenbank)
- Benutzerteil
- die Area-Reference (Realm-Referenz) in der Länge von fünf Byte bei auf Realms verteilten Satzarten, wenn deren Sätze aus mehreren Realms kopiert werden. Die Sätze werden nämlich immer in eine Ausgabedatei pro Satzart kopiert, wichtig für das Auftreten der Area-Reference ist also die Herkunft aus mehr als einem Realm.

Wenn BOUTLOAD Set-Informationen zu den einzelnen Sätzen ausgibt, ist im BOUTLOAD-Protokoll, das die Anweisungen für einen nachfolgenden BINLOAD-Lauf enthält, die Länge der Database-Key-Werte angegeben (Länge „4“ bei einer 2-Kbyte-Datenbank, Länge „8“ bei einer 4-Kbyte-/8-Kbyte-Datenbank).

Ohne Set-Information besteht der Ausgabesatz nur aus dem Benutzerteil.

11.14 BPGSIZE

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 8.2)

Anweisungen

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 8.2.4)

Anweisung	Bedeutung
ALLOCATE-BUFFER-POOL	Puffergröße festlegen
BUFFER-SIZE = <u>STD</u> / <integer 1..2000>	
CONVERT-DATABASE	Umstellung der Datenbank
REALM-NAME = <u>*ALL</u> / <u>*ALL-EXCEPT(...)</u> / list-poss(30): <realmname> <u>*ALL-EXCEPT(...)</u> NAME = list-poss(30): <realmname> , DATABASE-PAGE-LENGTH = <u>*UNCHANGED</u> / 2KB / 4KB / 8KB , TABLE-FILLING = <u>*UNCHANGED</u> / <u>*MAXIMUM</u> / <integer 1..100>	
END	Eingabe der Anweisungen beenden
OPEN-DATABASE	Datenbank eröffnen
DATABASE-NAME = <dbname> , COPY-NAME = <u>*NONE</u> / <copyname> , USER-IDENTIFICATION = <u>*OWN</u> / <userid>	
UNDO	Anweisungen rückgängig machen

Tabelle 62: Anweisungen für BPGSIZE

Kommandofolge zum Starten von BPGSIZE

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 8.2.5)

```
01 [/ADD-FILE-LINK LINK-NAME=DATABASE
    ,FILE-NAME=[ :catid: ][$userid.].DBDIR.[copyname]]
02 [/CREATE-FILE FILE-NAME=[ :catid: ][$userid.]dbname.realmname.NEW
    [,SUPPORT=*PUBLIC-DISK(SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär
    ,SECONDARY-ALLOCATION=576)) oder
    ,SUPPORT=*PRIVATE-DISK(VOLUME=archivnr, -
    DEVICE-TYPE=gerät[ ,SPACE=...])]
03 [... weitere CREATE-FILE-Anweisungen für Dateien der umgestellten Realms]
04 [/CREATE-FILE FILE-NAME= -
    [ :catid: ][$userid.]UTI.BPGSIZE.dbname.realmnummer.satzartnummer
    [,SUPPORT=*PUBLIC-DISK(SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär
    ,SECONDARY-ALLOCATION=sekundär)) oder
    ,SUPPORT=*PRIVATE-DISK(VOLUME=archivnr, -
    DEVICE-TYPE=gerät[ ,SPACE=...])]
05 [... weitere CREATE-FILE-Anweisungen für Arbeitsdateien von BPGSIZE]
06 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
07 /START-UDS-BPGSIZE
08 [//OPEN-DATABASE DATABASE-NAME = ...]
09 //BPGSIZE-Anweisungen
10 //END
```

01,08 Sie müssen genau eine der beiden Anweisungen angeben.

11.15 BPRECORD

(siehe Handbuch „Sichern, Informieren und Reorganisieren“, Kapitel 7)

Anweisungen

(siehe Handbuch „Sichern, Informieren und Reorganisieren“, Abschnitt 7.3)

Anweisung	Standardwert	Bedeutung
<u>SCHEMA</u> NAME IS <i>schemaname</i> .	Benutzer-schema	SCHEMA bezeichnen
<u>REALM</u> NAME IS <i>realmname</i> .	-	Auszugebenden Realm angeben
<u>PRINT</u> [{ <u>WITH</u> } <u>PAGEINFO</u>][{ <u>WITH</u> } <u>PAGEINDEX</u>] [{ <u>WITH</u> } <u>SCD</u>][<u>DBTT</u> { <u>DEC</u> <u>HEX</u> }]; [{ <u>WITHOUT</u> }]	WITHOUT und DEC	Umfang der Ausgaben festlegen
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>PAGE ZERO</u> ;	-	Kennseite ausdrucken
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>FPA OF</u> <i>seitenauswahl</i> ;	-	FPA-Einträge auflisten
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>DBTT OF</u> { <u>ALL RECORDS</u> <u>RECORD</u> <i>satzname</i> FOR <i>rsq-auswahl</i> };	-	DBTT-Einträge auflisten
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>CALC PAGES</u> <i>seitenauswahl</i> { <u>ALL</u> [{ <u>RECORDS</u> }] [<u>CALC SEARCHKEYS</u>] } { <u>ONLY</u> { <u>RECORD</u> <i>satzname</i> [<u>CALC SEARCHKEY</u> <i>keyref</i>] } <i>rsq-auswahl</i> };	-	CALC-Seiten ausdrucken

Tabelle 63: Anweisungen für BPRECORD

(Teil 1 von 2)

Anweisung	Standardwert	Bedeutung
<pre> DISPLAY [IN CSV [csv-dateiname]] DATA PAGES <i>seitenauswahl</i> { ALL [{RECORDS} {TABLES}] ONLY { RECORD <i>satzname</i> TABLES OF {OWNER <i>satzname</i> {SET <i>setname</i> KEY <i>keyref</i> } } } } <i>rsq-auswahl</i> </pre>	-	Datenseiten ausdrucken
END		BPRECORD-Lauf beenden

Tabelle 63: Anweisungen für BPRECORD

(Teil 2 von 2)

Physische Auswahl (Seitenauswahl)

```

seitenauswahl := { ALL PAGES
                   { PAGE { snr-1 [ TO snr-2 ] }, ... }

```

ALL PAGES

alle Seiten der mit der logischen Auswahl festgelegten Menge

PAGE *snr-1*,...

Liste mit Seitennummern

PAGE {*snr-1* TO *snr-2*},...Bereich von Seitennummer *snr-1* bis Seitennummer *snr-2*, usw.

Logische Auswahl (RSQ-Auswahl)

$$rsq\text{-auswahl} := \left\{ \begin{array}{l} \text{ALL RSQS} \\ \text{RSQ } \{rsq-1[\text{ TO } rsq-2] \}, \dots \end{array} \right\}$$

ALL RSQS

alle Satzfolgennummern

RSQ *rsq-1*,...

Liste mit Satzfolgennummern

RSQ *{rsq-1 TO rsq-2}*,...Bereich von Satzfolgennummer *rsq-1* bis Satzfolgennummer *rsq-2*, usw.**Kommandofolge zum Starten von BPRECORD**(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 7.4)

```

01 /ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid: ][ $userid. ]dbname.DBDIR[ .copyname]
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version.SCOPE=*TASK
03 /START-UDS-BPRECORD
04 bprecord-anweisungen
05 END

```

11.16 BPRIVACY

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Kapitel 4; für die Festlegung der Zugriffsrechte im laufenden Datenbankbetrieb siehe „[ONLINE-PRIVACY](#)“ auf Seite 233)

Aufbau der Benutzergruppenangaben

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 4.3)

Konfiguration	Wert			Definition in der ADD-USER-GROUP-Anweisung
	host	appl	grp	
openUTM Anw. ohne KSET	<i>host</i>	<i>appl</i>	<i>kset</i>	*KSET-FORMAT(HOST= <i>host</i> ,APPLICATION= <i>appl</i> ,KSET=*NONE)
openUTM Anw. mit KSET	<i>host</i>	<i>appl</i>	<i>kset</i>	*KSET-FORMAT(HOST= <i>host</i> ,APPLICATION= <i>appl</i> ,KSET= <i>kset</i>)
TIAM	<i>host</i>	'_'	<i>Kenn</i>	*FREE-FORMAT(HOST= <i>host</i> ,USER-ID= <i>Kenn</i>)
linked-in	<i>host</i>	'_'	<i>Kenn</i>	*FREE-FORMAT(HOST= <i>host</i> ,USER-ID= <i>Kenn</i>)

Tabelle 64: Aufbau der Benutzergruppenangaben

host Name des Verarbeitungsrechners, auf dem die UDS/SQL-UTM-Anwendung bzw. das UDS/SQL-Anwenderprogramm läuft.

Hier müssen Sie den Standardnamen des eigenen Prozessors aus DCAM-Sicht angeben. Sofern im TIAM-Fall kein DCAM zur Verfügung steht, geben Sie HOST=LOCAL an.

appl Name der UTM-Anwendung

kset KSET-Name, der der betreffenden UTM-Anwendung zugeordnet ist

Kenn BS2000-Benutzerkennung

Kommandofolge zum Starten von BPRIVACY

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 4.9)

```
01 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
02 [/ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=dbname.DBDIR]
03 /START-UDS-BPRIVACY
04 bprivacy-anweisungen
05 END
```

- 02 Wenn Sie die Datenbank über LINK-NAME=DATABASE zuweisen, dürfen Sie die BPRIVACY-Anweisung OPEN-DATABASE nicht angeben.
Wenn Sie die Datenbank nicht über LINK-NAME=DATABASE zuweisen, müssen Sie die BPRIVACY-Anweisung OPEN-DATABASE angeben.

11.17 BPSIA

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 4)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 4.2)

Anweisung	Bedeutung
<code>[DISPLAY [IN CSV [<i>csv-dateiname</i>]] <u>SCHEMA</u> <i>schemaname</i>]</code>	Schema ausdrucken
<code>[DISPLAY [IN CSV [<i>csv-dateiname</i>]] <u>SUBSCHEMA</u> <i>subschemaname</i>]</code>	Subschema ausdrucken
<code>END</code>	Eingabe der Anweisungen beenden

Tabelle 66: Anweisungen für BPSIA

Beide DISPLAY-Anweisungen sind wahlweise anzugeben. Sie dürfen sie auch mehrfach angeben.

Kommandofolge zum Starten von BPSIA

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 4.3)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid:][$userid.]dbname.DBDIR[.copyname]
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version,SCOPE=*TASK
03 /START-UDS-BPSIA
04 bpsia-anweisungen
05 END
```

04 Ein Punkt wird als Endekriterium betrachtet. Hinter dem Punkt kann wieder eine Anweisung stehen.

11.18 BPSQLSIA

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 5)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 5.5)

Anweisung	Bedeutung
END	Eingabe beenden
OPEN-DATABASE	Datenbank eröffnen
DATABASE-NAME = <dbname> ,COPY-NAME = <u>*NONE</u> / <copyname> ,USER-IDENTIFICATION = <u>*OWN</u> / <userid>	
PRINT-RELATIONAL-SCHEMAINFO	Subschemata auswählen
SUBSCHEMA-NAME = *ALL / *ALL-EXCEPT(...) / list-poss(20): <subschemaname> *ALL-EXCEPT(...) NAME = list-poss(20): <subschemaname>	

Tabelle 67: Anweisungen für BPSQLSIA

Kommandofolge zum Starten von BPSQLSIA

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 5.6)

```

01 [/ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR[ .copyname ]]
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version.SCOPE=*TASK
03 /START-UDS-BPSQLSIA
04 [//OPEN-DATABASE DATABASE-NAME = ...]
05 //PRINT-Anweisungen
06 //END

```

01, 04 Sie müssen eine der beiden Zuweisungen für die Datenbank verwenden.

11.19 BRENAME

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 7.4)

Kommandofolge zum Starten von BRENAME

Das Dienstprogramm BRENAME starten Sie in der Kennung, unter der die Datenbank katalogisiert ist, mit folgenden Kommandos:

```
01 [/CREATE-FILE FILE-NAME=dbname.DBCOM.0 ...]
02 [/CREATE-FILE FILE-NAME=dbname.COSSD.0 ...]
03 /ADD-FILE-LINK LINK-NAME=DATABASE, FILE-NAME=dbname.DBDIR
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version, SCOPE=*TASK
05 /START-UDS-BRENAME
06 END
```

01,02 Siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 3.1.1.



Die END-Anweisung ist die einzige BRENAME-Anweisung.

11.20 BREORG

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 9)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 9.4)

Anweisung	Bedeutung
ALLOCATE-BUFFERPOOL	Puffergröße festlegen
BUFFER-SIZE = <u>*STD</u> / <integer 1..2000>	
END	Eingabe der Anweisungen beenden
MODIFY-REALM-SIZE	Realm-Größe ändern
REALM-NAME = <realmname> ,REALM-SIZE = <integer 1..16777216> / *RELATIVE(...) / *MINIMUM *RELATIVE(...) DIFFERENCE = <integer -16777216..16777216>	
MODIFY-RECORD-POPULATION	Satzmengengerüst ändern
RECORD-NAME = <recordname> ,RECORD-POPULATION = <integer 1..2147483647> / *RELATIVE(...) / *MINIMUM *RELATIVE(...) DIFFERENCE = <integer -2147483647..2147483647>	
OPEN-DATABASE	Datenbank eröffnen
DATABASE-NAME = <dbname> ,SCHEMA-NAME = <u>*STD</u> / <schemaname> ,USER-IDENTIFICATION = <u>*OWN</u> / <userid>	

Tabelle 68: Anweisungen für BREORG

(Teil 1 von 2)

Anweisung	Bedeutung
<p>REORGANIZE-CALC</p> <p>RECORD-NAME = <recordname></p> <p>,CALC-RECORD = NONE / list-poss(6): *WITHIN-POPULATION(...)</p> <p style="padding-left: 20px;">*WITHIN-POPULATION(...)</p> <p style="padding-left: 40px;">REALM = *<u>ALL</u> / <realmname></p> <p style="padding-left: 40px;">,POPULATION = *UNCHANGED / <integer 1..2147483647></p> <p>,CALC-SEARCHKEY = NONE / list-poss(30): *KEY-POPULATION(...)</p> <p style="padding-left: 20px;">*KEY-POPULATION(...)</p> <p style="padding-left: 40px;">KEY-REF = *<u>ALL</u> / <integer 1..65535></p> <p style="padding-left: 40px;">,POPULATION = *<u>STD</u> / *UNCHANGED / <integer 1..2147483647></p>	<p>CALC-Bereiche reorganisieren</p>
<p>REORGANIZE-POINTERS</p> <p>REALM-NAME = <realname></p>	<p>Alle Probable Position Pointer (PPP) eines Realms reorganisieren</p>
<p>REORGANIZE-SET</p> <p>SET-NAME = <setname></p> <p>,OWNER-SELECTION = *<u>ALL</u> / list-poss(30): <integer 1..2147483647> /</p> <p style="padding-left: 20px;">*RANGE(...)</p> <p style="padding-left: 20px;">*RANGE(...)</p> <p style="padding-left: 40px;">FROM-RSQ = <integer 1..2147483647></p> <p style="padding-left: 40px;">,TO-RSQ = <integer 1..2147483647></p> <p>,KEY-SELECTION = *<u>ALL</u> / list-poss(30): <integer 1..32767></p> <p>,FILLING = *<u>UNCHANGED</u> / <integer 1..100></p>	<p>Tabellen und Set-Konstrukte reorganisieren</p>
<p>SPECIFY-SCHEMA</p> <p>SCHEMA-NAME = *<u>STD</u> / <schemaname></p>	<p>Schema angeben</p>
<p>SPECIFY-SUBSCHEMA</p> <p>SUBSCHEMA-NAME = <subschemaname></p>	<p>Subschema angeben</p>
<p>UNDO</p>	<p>Anweisung rückgängig machen</p>

Tabelle 68: Anweisungen für BREORG

Folgender Übersicht können Sie entnehmen, welche Probable Position Pointer (PPP) und Tabellen sich mit der Funktion REORGANIZE-SET reorganisieren lassen (siehe Handbuch „Entwerfen und Definieren“):

DDL- und SSL-Anweisungen		Probable Position Pointer (PPP)		Tabelle			
		Erläuterungen	aktualisieren möglich	Art	neuaufbauen möglich		
MODE IS CHAIN	ORDER IS FIRST/NEXT/PRIOR SORTED	Ownersatz enthält PPP ¹ des 1. Membersatzes der Kette	Ja	-	-		
		Vorwärtsverkettung der Membersätze mit RSQ und PPP ¹	Ja				
	ORDER IS LAST oder LINKED TO PRIOR	Ownersatz enthält PPP des letzten Membersatzes der Kette	Ja				
	LINKED TO PRIOR	Rückwärtsverkettung der Membersätze mit RSQ und PPP	Ja				
	ORDER IS SORTED INDEXED BY DEFINED KEYS... ASC/DESC KEY IS	jede Tabellenzeile enthält PPP des Membersatzes	Ja			mehrstufige Sort-Key-Tabelle	Ja
	ORDER IS SORTED INDEXED BY DATABASE-KEY	jede Tabellenzeile enthält PPP des Membersatzes	Ja			mehrstufige Sort-Key-Tabelle	Ja
MODE IS POINTER-ARRAY	ORDER IS FIRST/LAST/NEXT/PRIOR	jede Tabellenzeile enthält PPP des Membersatzes	Ja	einstufige Adressliste	Ja		
	ORDER IS SORTED INDEXED BY DEFINED KEYS... ASC/DESC KEY IS...	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige Adressliste	Ja		
	ORDER IS SORTED INDEXED BY DATABASE-KEY oder ORDER IS IMMATERIAL	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige Adressliste	Ja		

Tabelle 69: Übersicht über die Möglichkeit der Funktion REORGANIZE-SET

(Teil 1 von 2)

DDL- und SSL-Anweisungen		Probable Position Pointer (PPP)		Tabelle	
		Erläuterungen	aktualisieren möglich	Art	neue aufbauen möglich
MODE IS LIST	ORDER IS FIRST/LAST/NEXT/PRIOR	kein PPP enthalten	-	einstufige Liste	Ja
	ORDER IS SORTED INDEXED (DB-Key o. ASC/DESC-Key)	kein PPP enthalten	-	mehrstufige Liste	Ja
SEARCH KEY ..USING INDEX	TYPE IS REPEATED-KEY	jede Tabellenzeile enthält PPP des Membersatzes	Ja	mehrstufige SEARCH-Key-Tabelle	Ja
	TYPE IS DATABASE-KEY-LIST	kein PPP enthalten	-	Duplikattabelle	Ja
MEMBER IS PHYSICALLY LINKED TO OWNER		Membersatz enthält Zeiger zum Ownersatz (PPP)	Ja	-	-

Tabelle 69: Übersicht über die Möglichkeit der Funktion REORGANIZE-SET

(Teil 2 von 2)

¹ Diese PPPs kommen bei MODE IS CHAIN generell vor.

Kommandofolge zum Starten von BREORG

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 9.5)

```

01 [/ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid: ] [ $userid. ] dbname.DBDIR]]
02 [/CREATE-FILE FILE-NAME=arbeitsdatei-1 [,SUPPORT=*PUBLIC-DISK
      (SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär,
      SECONDARY-ALLOCATION=sekundär))] /
      ,SUPPORT=*PRIVATE-DISK(VOLUME=archivnr,
      DEVICE-TYPE=gerät [,SPACE=...])]
      /ADD-FILE-LINK LINK-NAME=SCRTCH1,FILE-NAME=arbeitsdatei-1,
      ACCESS-METHOD=*UPAM]
03 [/CREATE-FILE FILE-NAME=arbeitsdatei-2 [,SUPPORT=*PUBLIC-DISK
      (SPACE=*RELATIVE(PRIMARY-ALLOCATION=primär,
      SECONDARY-ALLOCATION=sekundär))] /
      ,SUPPORT=*PRIVATE-DISK(VOLUME=archivnr,
      DEVICE-TYPE=gerät [,SPACE=...])]
      /ADD-FILE-LINK LINK-NAME=SORTWK,FILE-NAME=arbeitsdatei-2,
      ACCESS-METHOD=*UPAM]
04 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version.SCOPE=*TASK
05 /START-UDS-BREORG
06 [//ALLOCATE-BUFFERPOOL BUFFER-SIZE = ...]
07 [//OPEN-DATABASE DATABASE-NAME = ...]
08 [//SPECIFY-SCHEMA SCHEMA-NAME = ...]
09 [//SPECIFY-SUBSCHEMA SUBSCHEMA-NAME = ...]
10... weitere breorg-anweisungen
11 //END

```

01, 07 Sie müssen eine der beiden Anweisungen angeben.

Arbeitsdateien einrichten

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 9.2)

Arbeitsdateien für die REORGANIZE-CALC- und REORGANIZE-SET-Anweisungen

- Dateikettungsname SCRTCH1

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich

- bei der Reorganisation von indirekten Hashbereichen durch

$(12 + \textit{schlüssel\länge}) * \textit{anzahl-einträge}$ Bytes

- bei der Reorganisation von direkten Hashbereichen durch

$8 * \textit{anzahl-einträge}$ Bytes

- bei Reorganisation mehrstufiger Tabellen durch

$12 * \textit{anzahl-einträge}$ Bytes

- Dateikettungsname SORTWK

Das Mengengerüst der zu sortierenden Daten ergibt sich

- bei der Reorganisation von indirekten Hashbereichen durch

$(12 + \textit{schlüssel\länge}) * \textit{anzahl-einträge}$ Bytes

- bei der Reorganisation von direkten Hashbereichen durch

$(\textit{satzlänge} + \textit{schlüssel\länge} + 7) * \textit{anzahl-einträge}$ Bytes

- bei Reorganisation mehrstufiger Tabellen durch

$12 * \textit{anzahl-einträge}$ Bytes

Richten Sie die beiden Arbeitsdateien nicht selbst ein, so richtet BREORG sie mit folgenden Namen und Größen ein:

UTI.tsn.SCRTCH1	(360,360) bei REORGANIZE-SET und REORGANIZE-CALC
UTI.tsn.SORTWK	(120,120)

Arbeitsdateien für die REORGANIZE-POINTERS-Anweisung

- Dateinamen UTI.BREORG.*dbname*.*xxx*.*yyyyy*

dbname Name der Datenbank

xxx Realm-Nummer des angegebenen Realms

yyyyy Nummer der Satzart, deren Probable Position Pointer (PPP) im Realm aktualisiert werden;
yyyyy=0 wird gegebenenfalls für Probable Position Pointer in SYSTEM-Sets genutzt.

Das Mengengerüst der zwischenspeichernden Daten ergibt sich aus der Formel

$$\text{anzahl-ppps} * 11 \text{ Bytes}$$

- Dateinamen UTI.BREORG.*dbname*.*xxx*.00001

dbname Name der Datenbank

xxx Realm-Nummer des angegebenen Realms

Im User-Schema gibt es keine Satzart mit der Satzartnummer 1. In der Arbeitsdatei mit der Satzartnummer 1 werden alle aktualisierten Probable Position Pointer (PPP) sortiert nach ihrer Lage im Realm abgelegt. Die Größe dieser Arbeitsdatei orientiert sich daher an der Summe der benötigten Einzeldateien UTI.BREORG.*dbname*.*xxx*.*yyyyy* (*yyyyy*=0 bzw. *yyyyy*>1).

- Dateikettungsname SRT1WK

Die benötigte Dateigröße entspricht dem Maximum der Größe aller Dateien UTI.BREORG.*dbname*.*xxx*.00001.

11.21 BSTATUS

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 6)

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 6.3)

Anweisung	Bedeutung
<u>SUBSCHEMA</u> IS <i>subschemaname</i>	Subschema bezeichnen
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>REALM</u> STATISTICS FOR $\left. \begin{array}{l} \{ \textit{realmname-1}, \dots \} \\ \{ \underline{ALL} \} \end{array} \right\}$	Realm-Statistik ausdrucken
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>TABLE</u> STATISTICS FOR <u>SET</u> $\left. \begin{array}{l} \{ \textit{setname-1} [\textit{rsq-selection-1}], \dots \} \\ \{ *ALL [\underline{EXCEPT} \textit{setname-1}, \dots] \} \end{array} \right\}$	Set-Statistik ausdrucken
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>TABLE</u> STATISTICS FOR <u>OWNER</u> <u>IN</u> <u>SET</u> $\left. \begin{array}{l} \{ \textit{setname-1}, \dots \} \\ \{ *ALL [\underline{EXCEPT} \textit{setname-1}, \dots] \} \end{array} \right\}$	Owner-Statistik ausdrucken
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>RECORD</u> STATISTICS FOR $\left. \begin{array}{l} \{ \textit{satzname-1}, \dots \} \\ \{ \underline{ALL} \} \end{array} \right\}$	Satzart-Statistik ausdrucken
<u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>CALC</u> KEY STATISTICS FOR $\left. \begin{array}{l} \left. \begin{array}{l} \{ \textit{satzname-1}, \dots \} \\ \{ \underline{ALL} \} \end{array} \right\} \\ \left. \begin{array}{l} \{ \textit{keyref-1}, \dots \} \\ \{ \underline{ALL} \} \end{array} \right\} \end{array} \right\} \text{ IN REALM } \left. \begin{array}{l} \{ \textit{realmname-1}, \dots \} \\ \{ \underline{ALL} \} \end{array} \right\}$	CALC-Key-Statistik ausdrucken

Tabelle 70: Anweisungen für BSTATUS

(Teil 1 von 2)

Anweisung	Bedeutung
<p><u>DISPLAY</u> [IN CSV [<i>csv-dateiname</i>]] <u>RECORDNUMBER</u> STATISTICS FOR</p> $\left. \begin{array}{l} \left\{ \begin{array}{l} \text{REALM} \left\{ \begin{array}{l} \textit{realmname-1, \dots} \\ \text{ALL} \end{array} \right\} \\ \text{RECORD} \left\{ \begin{array}{l} \textit{satzname-1, \dots} \\ \text{ALL} \end{array} \right\} \end{array} \right\}$	<p>Satzanzahl-Statistik ausdrucken</p>
<p><u>END</u></p>	<p>Eingabe der Anweisungen beenden</p>

Tabelle 70: Anweisungen für BSTATUS

(Teil 2 von 2)

Alle DISPLAY-Anweisungen sind wahlweise. Sie können sie in beliebiger Reihenfolge beliebig oft angeben.

Alle Anweisungen von BSTATUS können Sie mit einem Punkt (.) abschließen.

Kommandofolge zum Starten von BSTATUS

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 6.4)

```
01 /ADD-FILE-LINK LINK-NAME=DATABASE,
      FILE-NAME=[ :catid: ][ $userid. ] dbname.DBDIR[ .copyname]
02 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL, VERSION=version. SCOPE=*TASK
03 /START-UDS-BSTATUS
04 SUBSCHEMA IS subschemaname
05 display-anweisungen
06 END
```

Arbeitsdateien einrichten

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 6.2)

- SCRTCH1

Das Mengengerüst der zwischenzuspeichernden Daten ergibt sich aus der Formel:

$$132 * (setanzahl + schlüsselanzahl) \text{ Bytes}$$

Die Primärzuweisung für die Arbeitsdatei-1 sollte sich am Mengengerüst der zwischenzuspeichernden Daten orientieren. Es sollte immer eine angemessene Sekundärzuweisung erfolgen für den Fall, dass der Speicherplatz erweitert werden muss.

- SORTWK

Das Mengengerüst der zu sortierenden Daten ergibt sich aus der Formel :

$$16 * \text{anzahl-sortsätze} \text{ Bytes}$$

Richten Sie die beiden Arbeitsdateien nicht selbst ein, so richtet BSTATUS sie mit folgenden Namen und Größen ein:

UTI.SAMWORK.*tsn.zeitstempel.nnnn* (33,33)

UTI.*tsn*.SORTWK (120,120)

Nach normaler Beendigung des Laufes werden von BSTATUS angelegte Arbeitsdateien mit ihren Dateikettungsnamen wieder gelöscht. Explizit von Ihnen eingerichtete Arbeitsdateien werden nicht gelöscht und die Dateikettungsnamen nicht freigegeben.

11.22 BTRANS24

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Kapitel 9)

Anweisungen

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 9.3):

Anweisung	Funktion
CHECK-DATABASE	Prüflauf anstoßen
TRANSFORM-DATABASE	Datenbank umsetzen
END	Eingabe der Anweisungen beenden und Ausführung starten

Tabelle 71: Anweisungen für BTRANS24

Kommandofolge zum Starten von BTRANS24

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 9.4)

```
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIB=UDS/SQL-T-modulbibliothek,  
ELEM=BTRANS24)
```

```
//BTRANS24-Anweisungen ...
```

```
//END
```

BTRANS24 ist Bestandteil der Liefereinheit UDS-SQL-T und steht standardmäßig in der Bibliothek SIPPRG.UDS-SQL-T.028 zur Verfügung.

11.23 ONLINE-PRIVACY

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Kapitel 4; für die Festlegung der Zugriffsrechte im Offline-Modus siehe „[BPRIVACY](#)“ auf Seite 216)

Aufbau der Benutzergruppenangaben

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 4.3)

Konfiguration	Wert			Definition in der ADD-USER-GROUP-Anweisung
	host	appl	grp	
openUTM Anw. ohne KSET	<i>host</i>	<i>appl</i>	<i>kset</i>	*KSET-FORMAT(HOST= <i>host</i> ,APPLICATION= <i>appl</i> ,KSET=*NONE)
openUTM Anw. mit KSET	<i>host</i>	<i>appl</i>	<i>kset</i>	*KSET-FORMAT(HOST= <i>host</i> ,APPLICATION= <i>appl</i> ,KSET= <i>kset</i>)
TIAM	<i>host</i>	'_'	<i>Kenn</i>	*FREE-FORMAT(HOST= <i>host</i> ,USER-ID= <i>Kenn</i>)
linked-in	<i>host</i>	'_'	<i>Kenn</i>	*FREE-FORMAT(HOST= <i>host</i> ,USER-ID= <i>Kenn</i>)

Tabelle 72: Aufbau der Benutzergruppenangaben

host Name des Verarbeitungsrechners, auf dem die UDS/SQL-UTM-Anwendung bzw. das UDS/SQL-Anwenderprogramm läuft.
Hier müssen Sie den Standardnamen des eigenen Prozessors aus DCAM-Sicht angeben. Sofern im TIAM-Fall kein DCAM zur Verfügung steht, geben Sie HOST=LOCAL an.

appl Name der UTM-Anwendung

kset KSET-Name, der der betreffenden UTM-Anwendung zugeordnet ist

Kenn BS2000-Benutzerkennung

Kommandofolge zum Starten von ONLINE-PRIVACY

(siehe Handbuch „[Aufbauen und Umstrukturieren](#)“, Abschnitt 4.8)

```
01 /SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=version
02 /ADD-FILE-LINK LINK-NAME=DATABASE,FILE-NAME=konfigurationsname
03 /START-UDS-ONLINE-PRIVACY
04 OPEN-DATABASE DATABASE-NAME=dbname
05 übrige online-privacy-anweisungen
06 END
```

- 02 Mit dem Kommando SET-FILE-LINK weisen Sie den Konfigurationsnamen FILE-NAME=*konfigurationsname* über den Linknamen DATABASE zu. Die UDS/SQL-Konfiguration, mit der ONLINE-PRIVACY zusammenarbeiten soll und an die die zu bearbeitende Datenbank angeschlossen ist, müssen Sie mit diesem Kommando bekannt machen.

11.24 UDS-Online-Utility

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Kapitel 8).

Anweisungen

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 8.6, 8.7, 8.8)

SDF-Anweisungen der UDS-Online-Utility:

Anweisung	Bedeutung
DECLARE-PROCEDURE PROCEDURE-NAME = <structured-name 1..20> ,CODE = <c-string 1..1800> / <filename> / *SYSDTA	Prozedur-Deklaration eröffnen
DECLARE-VARIABLE VARIABLE-NAME = <structured-name 1..20> ,TYPE = *STRING(...) / <u>*INTEGER</u> *STRING LENGTH = <integer 1..20> ,INITIAL-VALUE = <integer 1..16777215> / <c-string> / <u>*STD</u> / *NONE	
DELETE-PROCEDURE PROCEDURE-NAME = <structured-name 1..20>	Prozedur löschen
DELETE-VARIABLE VARIABLE-NAME = <structured-name 1..20>	
END	UDS-Online-Utility beenden
REPEAT-PROCEDURE PROCEDURE-NAME = *STDRELOC / *STDFPASCAN / *STDREPPP / <structured-name 1..20> ,CYCLE-LIMIT = <integer 1..16777215> / *MAX	Prozedur ausführen
SET-FPA-SCAN-PARAMETERS SUBSCHEMA-NAME = <structured-name 1..30> ,REALM-NAME = <structured-name 1..30> ,SEARCH-MODE = <u>*REUSE</u> / *NOREUSE	
SET-ONLINE-UTILITY-PARAMETERS DBH = <u>*INDEPENDENT</u> / *LINKED-IN ,CONFIGURATION-NAME = <structured-name 1..17>	Online-Utility- Parameter festlegen

Tabelle 74: SDF-Anweisungen für UDS-Online-Utility

(Teil 1 von 3)

Anweisung	Bedeutung
SET-PREF-REALM-PARAMETERS SUBSCHEMA-NAME= <structured-name 1..30> ,SET-NAME= <structured-name 1..30> ,PREFERRED-REALM-NAME= <structured-name 1..30>	Preferred-Realm für eine verteilbare Liste festlegen
SET-RELOCATE-PARAMETERS SUBSCHEMA-NAME = <structured-name 1..30> ,REALM-NAME = <structured-name 1..30> ,RELOCATE-TYPE = *RECORD-PAGES(...) / *BASE-LEVEL-TABLE-PAGES(...) / *INDEX-LEVEL-TABLE-PAGES(...) / *DISTRIBUTABLE-TABLE-PAGES(...) *RECORD-PAGES INITIALIZE= *ANY / *YES / *NO ,PAGES-PER-DML= <integer 1..16777215> ,SKIP-ABOVE-FILLING= <integer 1..100> ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE *BASE-LEVEL-TABLE-PAGES INITIALIZE= *ANY / *YES / *NO ,PAGES-PER-DML= <integer 1..16777215> ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE *INDEX-LEVEL-TABLE-PAGES INITIALIZE= *ANY / *YES / *NO ,PAGES-PER-DML= <integer 1..16777215> *DISTRIBUTABLE-TABLE-PAGES INITIALIZE= *ANY / *YES / *NO ,PAGES-PER-DML= <integer 1..16777215> ,CLASH-HANDLING= *BREAK-DML / *SKIP-PAGE ,SET-NAME= <structured-name 1..30> ,TARGET-REALM-NAME= <structured-name 1..30>	Eigenschaften einer RELOCATE-DML festlegen
SET-REORGANIZE-PPP-PARAMETERS SUBSCHEMA-NAME = <structured-name 1..30> ,REALM-NAME = <structured-name 1..30> ,INITIALIZE = *ANY / *YES / *NO ,PAGES-PER-DML = <integer 1..16777215> ,CLASH-HANDLING = *BREAK-DML / *SKIP-PAGE	Eigenschaften einer RORGPPP-DML festlegen
SHOW-FPA-SCAN-PARAMETERS	Aktuell gültige Parameter für DML FPASCAN ausgeben

Tabelle 74: SDF-Anweisungen für UDS-Online-Utility

(Teil 2 von 3)

Anweisung	Bedeutung
SHOW-PROCEDURE	Prozedur ausgeben
PROCEDURE-NAME = <structured-name 1..20> / *STDRELOC / *STDFPASCAN / *STDREPPP	
SHOW-PREF-REALM-PARAMETERS	Aktuell gültige Parameter für DML PREFRLM ausgeben
SHOW-RELOCATE-PARAMETERS	Aktuell gültige Parameter für DML RELOCATE ausgeben
SHOW-REORGANIZE-PPP-PARAMETERS	Aktuell gültige Parameter für DML REORGPPP ausgeben
SHOW-VARIABLE	Aktuellen Wert einer Variablen ausgeben
VARIABLE-NAME = <structured-name 1..20>	

Tabelle 74: SDF-Anweisungen für UDS-Online-Utility

(Teil 3 von 3)

Prozedur-Anweisungen der UDS-Online-Utility:

Anweisung	Bedeutung
<u>ADD</u> <i>name</i> , { <i>value</i> <i>variable</i> } [, <i>condition</i>]	Wert zu einer Variablen addieren
<u>BREAK</u> <i>condition</i>	Prozedurfolge sofort beenden
<u>END</u>	Eingabe von Prozedur-Anweisungen beenden
<u>EXIT</u> <i>condition</i>	Prozedurfolge nach aktuellem Durchlauf beenden
<u>FINISH</u> [<u>WITH CANCEL</u>]	Aktuelle Transaktion beenden
<u>EPASCAN</u>	Startseite für Freiplatzsuche festlegen
<u>MOVE</u> <i>name</i> , { <i>value</i> <i>variable</i> } [, <i>condition</i>]	Wert einer Variablen festlegen
<u>PREFRLM</u>	Preferred-Realm für eine verteilbare Liste neu setzen bzw. verändern
<u>READY</u> [<u>EXCLUSIVE</u>] <u>UPDATE</u>	Aktuelle Transaktion der UDS-Online-Utility starten
<u>RELOCATE</u>	Verlagerung durchführen
{ <u>REMARK</u> * }	Kommentar einfügen
<u>REORGPPP</u>	Probable Position Pointers reorganisieren

Tabelle 75: Prozedur-Anweisungen der UDS-Online-Utility

<code>SHOW name[,condition]</code>	Wert einer Variablen ausgeben
<code>WAIT n[,condition]</code>	Wartezeit festlegen

Tabelle 75: Prozedur-Anweisungen der UDS-Online-Utility

Kommandofolge zum Starten der UDS-Online-Utility

(siehe Handbuch „[Sichern, Informieren und Reorganisieren](#)“, Abschnitt 8.6)

```
/SELECT-PRODUCT-VERSION PRODUCT-NAME=UDS-SQL,VERSION=...  
/START-UDS-ONLINE-UTILITY ...
```

Die UDS-Online-Utility muss in der Kennung der zu bearbeitenden Datenbank ablaufen. Eine Verteilung der Aufträge über UDS-D ist nicht möglich.

Der Name, mit dem die UDS-Online-Utility in den Ausgaben von DAL bzw. im UDS-Monitor erscheint ist \$UDSOUTI.

12 Funktionscodes der DML-Anweisungen

(siehe Handbuch „[Datenbankbetrieb](#)“, Kapitel 15)

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

Tabelle 76: Funktionscodes der DML-Anweisungen

(Teil 1 von 3)

Funktionscode	Funktionskurzbeschreibung	Bedeutung
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)

Tabelle 76: Funktionscodes der DML-Anweisungen

(Teil 2 von 3)

Funktions- code	Funktions- kurzbeschreibung	Bedeutung
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 76: Funktionscodes der DML-Anweisungen

(Teil 3 von 3)

¹ Nur bei Zusammenarbeit mit UDS-D

Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.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)
Datenbankbetrieb
Benutzerhandbuch

UDS/SQL (BS2000)
Entwerfen und Definieren
Benutzerhandbuch

UDS/SQL (BS2000)
Meldungen
Benutzerhandbuch

UDS/SQL (BS2000)
Sichern, Informieren und Reorganisieren

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

Stichwörter

A

ADD 124
Adressliste 38, 39
Alias-Namen 177
alog-seq-nr 22
altes Format (Subschema) 133
angeben
 Katalogkennung 166
appl 22
Area 42
AREA NAME-Klausel 28
AREA SECTION 42
ASCENDING KEY-Klausel 34
Assembler-Makros 92
Aufbau
 Ausgabesatz 210
aufrufen, BINDER 177
Ausgabedatei
 einrichten 209
Ausgabesatz
 Aufbau 210
 BOUTLOAD 210
Auswahlmethode für Set-Occurrences 33, 34

B

BALTER 179
BCALLSI 183
BCHANGE 185
 Kommandofolge 185, 221
BCHECK 186
BCREATE 189
Bedingung 43
Bedingungsname-Klausel 43
beenden, BINDER 177

Benutzerinformation 57
 Bereich 60
BFORMAT 190
BGSIA 192
BGSSIA 194
BINILOAD 195
 INSERT (Übersicht) 197
 INSERT-Anweisungen 197
 Kommandofolge 199
 Programmanweisungen 195
 Steueranweisungen 195
 STORE-Anweisungen 196
BMEND 202
BMODTT 206
 Kommandofolge 207
BOUTLOAD 208
 Ausgabesatz 210
 Kommandofolge 208
 Protokoll 210
BPGSIZE 211
BPRECORD 213
 Kommandofolge 215
BPRIVACY 216
 Kommandofolge 218
BPSIA 219
BPSQLSIA 220
BRENAME 221
BREORG 222
 Kommandofolge 226
BSTATUS 229
 Kommandofolge 230
BTRANS24 232

C

c-string 22
catid 22, 165
Catid-Gruppe 168
CHECK- TABLE 133
COBOL-Programm übersetzen 53, 93
COMPARE SUBSCHEMAS 135
CONTINUE 124
COPY-Klausel 42
copyname 22
COSSD-Datei 53
CREATE 135

D

dal-cmd 22
DAL-Kommando
 linked-in DBH (Übersicht) 157
Data Base Key 31
Data Base Key Translation Table 36
 Lage 36
DATA DIVISION 46
DATABASE-KEY-Feld 31
date 22
Datenbank-Jobvariable 173
Datengruppe 43
Datenteil 46
Datentypen 22
 Zusätze 25
DBH 124
 linked-in, DAL-Kommandos (Übersicht) 157
DBH-Varianten 54
dbname 22
DCL 124
DECLARE 124
DEFINE 124
DELETE 124
DELETE SCHEMA 133
DELETE SUBSCHEMA 134
DESCENDING KEY-Klausel 34
device 22
DIAGNOSTIC 135
Direktzugriff 30, 34
DISPLAY 124, 134
DISPOFF 124

DML-Programm
 Starten 54, 94
DMLTEST 121
DO 124
DOFF 124
DSCAL 92
DSCAP 92
DSCDF 92
DSCPA 92

E

EDT 124
einrichten
 Ausgabedatei 209
END 124
END-Anweisung 136
ESCAPE 124
EXECUTE 124
EXECUTION 195

F

FCOD 57, 58, 62
Feld
 binär 31
 gepackt 31
 national 133
Feldname 57
FETCH 100
FILLING 179, 195
FIND 100
FOPT 57, 58, 62
Format
 altes (Subschema) 133
Funktionen
 UDSADM 152
Funktionsname 57
Funktionswahl 57

H

HALT 124
Hashbereich
 benennen 34
 Größe 36
 Lage 37

HELP 124

host 23

I

IDENTIFICATION DIVISION 42, 46

Identifikationsteil 46

independent DBH 54, 94

INPUT FILE 195

INSERT (Übersicht) 197

INSERT SET 197

INSERT-Anweisungen

 BINILOAD 197

integer 23

ITMN 57, 58, 62

J

Jobvariable

 Datenbank 173

 Pubset-Deklaration 168

 Session 170

K

Katalogkennung

 angeben 166

Kennwort 28

Kommandofolge

 BCHANGE 185, 221

 BINILOAD 199

 BMODTT 207

 BOUTLOAD 208

 BPRECORD 215

 BPRIVACY 218

 BREORG 226

 BSTATUS 230

 ONLINE-PRIVACY 235, 240

 SSL übersetzen 130

Kommandofolge UDSMON 165

kset 23

L

Laden und Starten des BINDER 177

LANGUAGE 125

LEAVE 125

linked-in DBH 55

 DAL-Kommandos (Übersicht) 157

LIST 125

Liste 38, 39

 verteilbar, siehe verteilbare Liste

LMS 193

LOOKC

 Block, Generelle Beschreibung 76

 Block, Spezielle Beschreibung 76

 Tabellen 81

LS 125

M

MEMBER-Klausel 34

Membersatzart 33

Metasprache 18

Metasyntax, SDF-Anweisungen 20

MODE-Klausel 39

MOVE 125

N

name 23

NATIONAL 111

nationales Feld 133

NEXT 125

O

ONLINE-PRIVACY 233

 Kommandofolge 235, 240

OWNER CALCKEY 197

OWNER DBKEY 198

OWNER KEY 198

OWNER RSQ 198

OWNER SEARCHKEY 197

OWNER-Klausel 34

Ownersatzart 33

P

Parameter der CALL-DML 57

PARLIST 132

PERFORM 125

POFF 125

PPP (Probable Position Pointer) 224

Preferred-Realm
 verteilbare Liste 237, 239
Primärschlüssel 30, 33, 36
PRINT 125
Probable Position Pointer (PPP) 224
PROC 125
PROFF 125
Programmanweisungen
 BINILOAD 195
PROT 125
Protokoll
 BOUTLOAD 210
Pubset-Deklarations-Jobvariable 168

Q

QUOTE 135

R

Readme-Datei 15
Realm 42
Realm-Eintrag
 DDL 28
Realmname 57
realmname 23
realmref 23
RECA 57, 58, 62
RECN 57, 58, 62
RECORD SECTION 43
RECORD-AREA NAME 196
RECORD-DBKEY 196
RECORD-DISPL 196
RECORD-RSQ 196
recordname 23
recordref 23
REMARK 125
RENAME 192
Reorganisation
 dynamisch 36, 37
RLMN 57, 58, 62
RUN 126

S

Satz 39
Satz-Eintrag
 DDL 29
 SSL 36
 Subschema-DDL 41
Satz-SEARCH-Key-Tabelle 37
Satzart 30, 36, 43
Satzbereich 57
Satzelement 30, 43
Satzname 57
Satzreihenfolge 30, 33
SCHEMA 183
Schema
 Benennung 28
SCHEMA NAME 195
Schema-DDL
 Aufbau 28
Schema-Eintrag
 SSL 35
schemaname 23
Schlüsselwort 18
Schlüsselwörter 123
Schlüsselwortparameter 122
SDF-Anweisungen, Metasyntax 20
SEARCH KEY-Klausel 34
SEARCH-Key 34, 36
SEARCH-Key-Tabelle 38, 39
 benennen 34
Seitenlänge 30
Sekundärschlüssel 30, 33
Session-Jobvariable 170
SET 126
Set 33, 38, 44
 dynamisch 33
SET ORDER 197
SET SECTION 44
Set-Eintrag
 DDL 32
 SSL 38
Set-Mitgliedschaft 33, 34
Set-Occurrence 38
Set-SEARCH-Key-Tabelle 38
SETN 57, 58, 62

- Setname 57
- setname 24
- SHOW 126
- SOPT 57, 58, 62
- SORCLIST 132
- Sort-Key-Tabelle 38, 39
- SORTCORE 195
- SOURCE 132
- Spezialparameter-1 57
- Spezialparameter-2 57
- Spezialparameter-3 57
- SPP1 57, 59, 63
- SPP2 57, 59, 63
- SPP3 57, 59, 63
- SSL 35
 - Aufbau 35
- SSL übersetzen
 - Kommandofolge 130
- START-Kommandos
 - UDS/SQL-Programme 177
- START-UDS-DBH
 - Syntax 144
- Startanweisungen 177
 - Alias-Namen 177
- starten
 - UDSADM 151
- Starten DBH, START-UDS-DBH 144
- Statuscodes
 - UDS-Online-Utility 102, 112
- Steueranweisungen
 - BINILOAD 195
- STOP 124
- STORE RECORD 196
- STORE-Anweisungen
 - BINILOAD 196
- structured-name (Datentyp) 24
- SUBSCHEMA 126, 183
- Subschema 41
 - benennen 42
- SUBSCHEMA FORM IS OLD 133
- SUBSCHEMA NAME 195
- Subschema-DDL 41
 - Aufbau 41
- subschema_name 24
- Syntaxbeschreibung 20
- SYSTEM 126
- T**
- TEMPORARY-Klausel 28
- Testen von DML-Funktionen 121
- time 24
- TRACE 126
- U**
- übersetzen
 - COBOL-Programm 53
- Übersicht DAL-Kommandos
 - linked-in DBH 157
- UDS-Online-Utility 236
 - Statuscodes 102, 112
- UDS/SQL-Programme
 - START-Kommandos 177
- UDSADM
 - Funktionen 152
 - starten 151
- UDSMON
 - Kommandofolge 165
- UINF 57, 58, 62
- Unicode 111, 133
- USER FILE BUFFER LENGTH 195
- USER FILE RECORD LENGTH 195
- userid 24
- UTF-16 111
- V**
- Variable 18
- Vektor 31
- Verknüpfung von Sätzen 39
- verteilbare Liste 112, 237, 239
 - Preferred-Realm 237, 239
- volume 24
- W**
- Wahlwort 18
- WAIT 126
- Wiederholungsfaktor 31
- Wiederholungsgruppe 31

X

x-string [24](#)

Z

Zeiger [38](#)

Zugriff

 direkt [30, 34](#)

Zugriffspfad [30, 33](#)

Zusätze, Datentypen [25](#)

Zusatzwahl [57](#)