English

FUJITSU Software BS2000

SESAM/SQL-Server V9.1 Database Operation

User Guide

June 2019

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
bs2000services@ts.fujitsu.com senden.

Certified documentation according to DIN EN ISO 9001:2015

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2015.

Copyright and Trademarks

Copyright © 2018 Fujitsu Technology Solutions GmbH.

All rights reserved.
Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

Table of Contents

Database Operation **11**
1 Preface **12**
1.1 Objectives and target groups of this manual **13**
1.2 Summary of contents **14**
1.3 Notational conventions **15**
2 Working with the SESAM/SQL DBH **17**
2.1 Starting SESAM/SQL programs via start commands **19**
2.1.1 Overview of the start commands 20
2.1.2 Operands of the start commands 22
2.1.3 Tasks of the start commands 24
2.1.4 Specifying the correction status with the start command 26
2.2 Starting the SESAM/SQL DBH **27**
2.2.1 Requirements for starting 28
2.2.2 Starting the independent DBH 29
2.2.2.1 Command sequence for starting the independent DBH in batch mode . . 30
2.2.2.2 Entering the DBH start parameters for the independent DBH 31
2.2.2.3 Example of a procedure for starting the independent DBH 35
2.2.3 Starting the linked-in DBH 37
2.2.4 Starting several SESAM/SQL DBHs 39
2.2.5 SESAM/SQL share modules 40
2.2.6 Dynamic reconfiguration of the DBH session 42
2.3 Terminating the SESAM/SQL DBH **44**
2.3.1 Terminating the independent DBH 45
2.3.2 Terminating the linked-in DBH 46
2.4 Restarting a DBH session **47**
2.4.1 Restart requirements 48
2.4.2 Controlling the duration of the restart 49
2.4.3 Carrying out an external restart 50
2.5 Job variables **51**
2.5.1 Job variables for controlling requests and programs 52
2.5.2 Monitor job variables for job monitoring 55
2.6 Sending important information of the DBH session by email **56**
3 DBH start statements and options **58**
3.1 Overview and notes on entry **59**
3.1.1 DBH start statements 60
3.1.2 DBH options 62
3.2 Descriptions in alphabetical order **65**

3.2.1 Description format 66
3.2.2 Alphabetical reference section 68
3.2.2.1 ACCOUNTING 69
3.2.2.2 ADD-OLD-TABLE-CATALOG-LIST 71
3.2.2.3 ADD-SQL-DATABASE-CATALOG-LIST 73
3.2.2.4 ADMINISTRATION 77
3.2.2.5 ADMINISTRATOR 78
3.2.2.6 CODED-CHARACTER-SET 80
3.2.2.7 COLUMNS 81
3.2.2.8 CONFIGURATION-NAME 83
3.2.2.9 CPU-RESOURCES 84
3.2.2.10 CURSOR-BUFFER 85
3.2.2.11 DBH-IDENTIFICATION 87
3.2.2.12 DBH-NAME 88
3.2.2.13 DBH-TASKS 89
3.2.2.14 FILE-RESOURCES 90
3.2.2.15 LINKED-IN-ATTRIBUTES 91
3.2.2.16 MEDIA-CATALOG 92
3.2.2.17 MSG-OUTPUT 96
3.2.2.18 OLD-TABLE-CATALOG 99
3.2.2.19 RECOVER-OPTIONS 100
3.2.2.20 REQUEST-CONTROL 103
3.2.2.21 RESTART-CONTROL 105
3.2.2.22 RETRIEVAL-CONTROL 107
3.2.2.23 SECURITY 109
3.2.2.24 SERVICE-TASKS 110
3.2.2.25 SESSION-LOGGING-ID 114
3.2.2.26 SET-DBH-OPTIONS 115
3.2.2.27 SPACES 116
3.2.2.28 SQL-DATABASE-CATALOG 117
3.2.2.29 SQL-SUPPORT 118
3.2.2.30 STORAGE-SIZE 120
3.2.2.31 SUBORDERS 121
3.2.2.32 SYSTEM-DATA-BUFFER 123
3.2.2.33 SYSTEM-LIMITS 124
3.2.2.34 SYSTEM-STRATEGIES 125
3.2.2.35 SYSTEM-THREADS 126
3.2.2.36 THREADS 128
3.2.2.37 TRANSACTION-SECURITY 129
3.2.2.38 TRANSFER-CONTAINER 132
3.2.2.39 USER-DATA-BUFFER 134

3.2.2.40 USERS . . . 135
3.2.2.41 WORK-CONTAINER . . . 136
4 Distributed processing with SESAM/SQL-DCN . . . 139
4.1 Starting the SESDCN distribution component . . . 140
4.1.1 Requirements for starting . . . 141
4.1.2 Starting SESDCN in batch mode and passing control statements . . . 142
4.1.3 Starting SESDCN in interactive mode and passing control statements . . . 143
4.1.4 Passing control statements by means of an input file . . . 144
4.1.5 Starting several SESDCNs . . . 145
4.2 Terminating SESDCN . . . 146
4.3 Restarting an SESDCN session . . . 147
4.3.1 SESDCN restart requirements . . . 148
4.3.2 Restarting SESDCN on a backup computer . . . 149
4.3.3 Synchronization of DBH, SESDCN and UTM restarts . . . 150
4.4 SESDCN control statements . . . 151
4.4.1 Entering SESDCN control statements . . . 152
4.4.2 Syntax and functions of the SESDCN control statements . . . 154
4.4.2.1 SET-DCN-OPTIONS . . . 155
4.4.2.2 ADD-DISTRIBUTION-RULE-LIST . . . 160
4.4.2.3 ADD-NETWORK-LINK-LIST . . . 162
4.4.3 Examples of SESDCN parametrization . . . 164
5 DBH and SESDCN administration . . . 168
5.1 Administration interfaces . . . 169
5.1.1 Administration using INFORM-PROGRAM . . . 170
5.1.1.1 NEXT (administration command) . . . 173
5.1.2 Administration using a CALL DML program . . . 174
5.1.3 Administration using SESADM . . . 176
5.1.3.1 Starting SESADM . . . 179
5.1.3.2 Entering the SESADM statements . . . 180
5.1.3.3 SDF selection screens for SESADM administration . . . 182
5.1.3.4 SESADM control statements . . . 186
5.1.3.5 START-DBH-ADMINISTRATION . . . 187
5.1.3.6 START-DCN-ADMINISTRATION . . . 188
5.1.3.7 END . . . 189
5.1.3.8 Controlling output of the administration statements . . . 190
5.1.3.9 MODIFY-OUTPUT-MODE . . . 191
5.1.3.10 SHOW-LAST-RESULT . . . 192
5.1.3.11 Structured output in S variables . . . 193
5.1.3.12 Result of the administration statements . . . 197
5.1.3.13 SDF standard statement STEP . . . 198
5.2 Administration statements and commands . . . 199

5.2.1 Summary of contents 200
5.2.1.1 DBH administration statements 201
5.2.1.2 SESDCN administration statements 205
5.2.1.3 Administration statements for changing DBH options 206
5.2.2 Equivalent administration statements and commands 208
5.2.3 Descriptions in alphabetical order 214
5.2.3.1 Description format 215
5.2.3.2 Syntax for the identification of users 217
5.2.3.3 Identification of system-internal users 221
5.2.3.4 Syntax for output in S variables 222
5.2.3.5 Alphabetical reference section 223
5.2.3.6 ABORT-LOCK-SEQUENCE 224
5.2.3.7 ADD-DISTRIBUTION-RULE-ENTRY 226
5.2.3.8 ADD-OLD-TABLE-CATALOG-ENTRY 228
5.2.3.9 ADD-SQL-DB-CATALOG-ENTRY 230
5.2.3.10 ASSIGN-SYSLST 234
5.2.3.11 BEGIN-LOCK-SEQUENCE 235
5.2.3.12 CANCEL-STATEMENT 237
5.2.3.13 CHANGE-CATLOG 238
5.2.3.14 CHANGE-DALOG 240
5.2.3.15 CLOSE-SPACE 242
5.2.3.16 COMMIT-PTC-TRANSACTION 244
5.2.3.17 CREATE-DUMP 246
5.2.3.18 CREATE-SERVICE-TASK-DUMP 247
5.2.3.19 END-FOREIGN-COPY 248
5.2.3.20 END-LOCK-SEQUENCE 250
5.2.3.21 HOLD-TRANSACTION-ADMISSION 251
5.2.3.22 HOLD-USER-ADMISSION 253
5.2.3.23 MODIFY-ADMINISTRATION 254
5.2.3.24 MODIFY-CATALOG-ACCESS-RIGHTS 257
5.2.3.25 MODIFY-CATID-LIST 260
5.2.3.26 MODIFY-DISTRIBUTION-RULE-ENTRY 262
5.2.3.27 MODIFY-MAIL-PARAMETERS 263
5.2.3.28 MODIFY-MSG-OUTPUT 264
5.2.3.29 MODIFY-OLD-TABLE-CATALOG-LIMIT 267
5.2.3.30 MODIFY-RECOVER-OPTIONS 268
5.2.3.31 MODIFY-REQUEST-CONTROL 272
5.2.3.32 MODIFY-RESTART-CONTROL 274
5.2.3.33 MODIFY-RETRIEVAL-CONTROL 276
5.2.3.34 MODIFY-SECURITY 278
5.2.3.35 MODIFY-SERVICE-TASKS 279

5.2.3.36 MODIFY-SESSION-LOGGING-ID 283
5.2.3.37 MODIFY-SQL-SORT-LIMIT 285
5.2.3.38 MODIFY-STORAGE-SIZE 286
5.2.3.39 MODIFY-SUBORDER-LIMIT 288
5.2.3.40 MODIFY-TRANSACTION-SECURITY 289
5.2.3.41 PREPARE-FOREIGN-COPY 291
5.2.3.42 RECONFIGURE-DBH-SESSION 294
5.2.3.43 RELEASE-USER-RESOURCES 296
5.2.3.44 RELOAD-DBH-SESSION 298
5.2.3.45 REMOVE-DISTRIBUTION-RULE-ENTRY 300
5.2.3.46 REMOVE-OLD-TABLE-CATALOG-ENTRY 301
5.2.3.47 REMOVE-SQL-DB-CATALOG-ENTRY 303
5.2.3.48 RESUME-TRANSACTION-ADMISSION 305
5.2.3.49 RESUME-USER-ADMISSION 306
5.2.3.50 REUSE-OLD-TABLE-CATALOG-ENTRY 307
5.2.3.51 REUSE-PARTITIONS 309
5.2.3.52 ROLLBACK-PTC-TRANSACTION 311
5.2.3.53 ROLLBACK-TRANSACTION 312
5.2.3.54 SAVE-DBH-OPTIONS 313
5.2.3.55 SET-ACCOUNTING-PARAMETER 315
5.2.3.56 SET-DBH-MSG-TRACE 316
5.2.3.57 SET-DIAGNOSIS-DUMP-PARAMETER 318
5.2.3.58 SET-REQUEST-CONTROL 320
5.2.3.59 SET-SAT-SUPPORT 321
5.2.3.60 SET-SESSION-DIAGNOSIS 322
5.2.3.61 SET-SQL-DB-CATALOG-STATUS 324
5.2.3.62 SET-TUNING-TRACE 326
5.2.3.63 SET-USER-CALL-TRACE 330
5.2.3.64 SET-USER-INACTIVE-TIME 332
5.2.3.65 SET-USER-MSG-TRACE 333
5.2.3.66 SHOW-CALL-DML-SUBORDERS 335
5.2.3.67 SHOW-CATALOG-USERS 337
5.2.3.68 SHOW-CATID-LIST 339
5.2.3.69 SHOW-DBH-MEDIA-CATALOG 341
5.2.3.70 SHOW-DBH-OPTIONS 343
5.2.3.71 SHOW-DISTRIBUTION-RULE-ENTRIES 345
5.2.3.72 SHOW-INACTIVE-SQL-USERS 347
5.2.3.73 SHOW-MAIL-PARAMETERS 349
5.2.3.74 SHOW-OLD-TABLE-CATALOG-ENTRIES 350
5.2.3.75 SHOW-PARTITIONS 353
5.2.3.76 SHOW-SPACE-USERS 355

5.2.3.77 SHOW-SQL-DB-CATALOG-ENTRIES . . . 357
5.2.3.78 SHOW-TRANSACTIONS . . . 360
5.2.3.79 SHOW-USERS . . . 365
5.2.3.80 SHOW-USER-SPACES . . . 369
5.2.3.81 STOP-DBH . . . 371
5.2.3.82 STOP-DCN . . . 372
6 Evaluating request logging with SESCOSP . . . 373
6.1 Calling SESCOSP . . . 375
6.2 SESCOSP input parameters . . . 377
6.2.1 The higher-level parameter CREATE-REPORT . . . 378
6.2.2 The individual parameter INPUT-FILE . . . 379
6.2.3 The individual parameter OUTPUT . . . 380
6.2.4 The individual parameter SELECT . . . 381
6.2.4.1 The SELECT parameter CATALOG-NAMES . . . 382
6.2.4.2 The SELECT parameter USERS . . . 384
6.2.4.3 The SELECT parameter STATEMENTS . . . 386
6.2.4.4 The SELECT parameter TIME-PERIOD . . . 387
6.2.4.5 The SELECT parameter CPU-TIME . . . 388
6.2.4.6 The SELECT parameter ELAPSED-TIME . . . 389
6.2.4.7 The SELECT parameter NUMBER-OF-LOGICAL-IO . . . 390
6.2.5 The individual parameter REPORT-FORMAT . . . 391
6.2.5.1 The REPORT-FORMAT parameter PROTOCOL . . . 392
6.2.5.2 The REPORT-FORMAT parameter TIME-PRECISION . . . 393
6.3 SESCOSP report output . . . 394
6.3.1 Output of general information and the DBH options . . . 395
6.3.2 Variable parts of the report . . . 396
6.3.3 Termination statistics . . . 406
6.3.4 Catalog-space table assignment table . . . 407
7 Outputting operational data with SESMON . . . 408
7.1 Overview of the statistical output of SESMON . . . 410
7.2 Running SESMON . . . 414
7.2.1 Starting SESMON . . . 415
7.2.1.1 Start parameters when running SESMON in interactive mode . . . 416
7.2.1.2 Start parameters when running SESMON in batch mode . . . 417
7.2.2 SESMON administration . . . 421
7.2.3 Automatic termination of SESMON operation . . . 422
7.3 The layout of the SELECTION form . . . 423
7.4 Layout of the forms . . . 425
7.4.1 CONFIGURATION form APPLICATIONS . . . 426
7.4.2 CONFIGURATION form PREFETCH-BUFFERS . . . 428
7.4.3 SESDCN form APPLICATIONS . . . 430

7.4.4 SESDCN form CAPACITY 432
7.4.5 SESDCN form OVERVIEW 435
7.4.6 SESDCN form TRANSACTIONS 437
7.4.7 DBH form I/O 440
7.4.8 DBH form OPTIONS 442
7.4.9 DBH form SERVICE ORDERS 443
7.4.10 DBH form SERVICE TASKS 448
7.4.11 DBH form SQL INFORMATION 450
7.4.12 DBH form STATEMENTS 452
7.4.13 DBH form SYSTEM INFORMATION 454
7.4.14 DBH form TASKS 457
7.4.15 DBH form SYSTEM THREADS 459
7.4.16 DBH form TRANSACTIONS 460
7.5 Layout of output to a file 463
7.5.1 Layout of the output records 464
7.5.1.1 Configuration record 465
7.5.1.2 Option record 467
7.5.1.3 CNF record 468
7.5.1.4 SESDCN record 469
7.5.1.5 DBH record 472
7.5.2 Example of editing the SESMON output file with C 477
7.6 Layout of output to SYSLST 478
7.7 Output of statistical data via SNMP 486
7.8 Outputting statistical data via openSM2 490
8 DA-LOG formatting by SEDI70 492
8.1 SEDI70 call 493
8.2 SEDI70 input parameters 494
8.2.1 The higher-level parameter CREATE-REPORT 495
8.3 SEDI70 report output 502
8.3.1 General information 503
8.3.2 File identification records 504
8.3.3 Variable part of the list 505
9 Error handling 509
9.1 Consistency checks 510
9.1.1 Error weight of consistency checks with message numbers SES6000 and SED9001 511
9.1.2 Error weight of consistency checks with message numbers SEN0001, SEN0003 and SEK0001 517
9.2 Messages 518
9.2.1 Messages in special situations 519
9.3 Setting task switches 522

9.4 End of session after ABEND events 523
9.5 Diagnostics documentation 524
9.5.1 Main-memory dump 525
9.5.2 Diagnostics traces 528
9.5.3 Diagnostic information in the DB-DIAGAREA of openUTM 530
9.5.4 DBH-specific log files 534
9.5.5 Overview of helpful administration statements 535
9.5.6 Diagnostics documents for service tasks 536
9.5.7 EXPLAIN pragma 538
9.5.8 Error files for SQL and utility statements 539
10 Appendix 540
10.1 Specifying the sort sequence for CALL DML 541
10.2 Password protection with SEPA 542
10.2.1 The password-protection concept 543
10.2.2 The SEPA statements 548
10.2.2.1 Statement 1P0 (format 1) 550
10.2.2.2 Statement 1P0 (format 2) 551
10.2.2.3 Statement 1P1 552
10.2.2.4 Statements 1P2 and 1P2N 555
10.2.2.5 Statements 1P3 and 1P3N 557
10.2.2.6 Statement 1P4 559
10.2.2.7 Statement 1P5 560
10.2.2.8 END statement 561
10.2.3 Example of password generation 562
10.3 Layout of the accounting information 563
10.4 Layout of the log records for SAT 566
10.4.1 SESAM object events 567
10.4.2 Auditable information for every SESAM object event 568
10.5 Using FASTPAM for SESAM/SQL 571
10.6 Starting SESAM programs with /START-PROGRAM 574
11 Related publications 576

Database Operation

1 Preface

The functions and architectural features of the SESAM/SQL-Server database system meet all the demands placed on a powerful database server in today's world. These characteristics are reflected in its name: SESAM/SQL-Server.

SESAM/SQL-Server is available in a standard edition for single-task operation and in an enterprise edition for multitask operation.

For the sake of simplicity, we shall use the name SESAM/SQL throughout this manual to refer to SESAM/SQL-Server.

The following introductory descriptions are contained centrally in the "Core manual":

- Brief product description
- Structure of the SESAM/SQL server documentation
- Demonstration database
- Readme file
- Changes since the last editions of the manuals

1.1 Objectives and target groups of this manual

This manual describes the means that can be used to control and monitor database operation. It is therefore aimed at system administrators responsible for starting and terminating database operation and for overseeing the current session. It is also aimed at programmers who create SESAM/SQL applications with the linked-in version of the Data Base Handler (DBH).

Users should know the basic principles, concepts and interrelationships of the SESAM/SQL database system, as described in the “ Core manual”. In particular, they should be familiar with the contents of the chapter entitled “Database operation” in this manual. Knowledge of the fundamentals of the BS2000 operating system and the Universal Transaction Monitor, openUTM, is also advantageous.

1.2 Summary of contents

This manual covers the following topics:

- starting and terminating the Data Base Handler (DBH)
- the DBH start statements and options, by means of which you parameterize the DBH when starting the session
- starting, parameterizing and terminating the SESDCN distribution component
- the options available for administering the DBH and the SESDCN distribution component during the session
- the utilities available for database operation
- error recovery

Where the linked-in DBH differs from the independent DBH, the differences are described at the appropriate points in the text.

In addition to the parts describing database operation, the manual also contains important alphabetically arranged reference sections:

- chapter 3 contains all the DBH start statements and options,
- chapter 4 contains all the SESDCN control statements,
- chapter 5 contains all the administration statements for the DBH and SESDCN.

All the SESAM/SQL messages, including those that concern database operation, are described in the “ Messages” manual.

1.3 Notational conventions

Because of the frequency with which the server names are used, the following abbreviations are used to make things simpler and more straightforward:

- **BS2000 servers** for the servers with /390 architecture and the servers with x86 architecture. These servers are operated with the corresponding BS2000 operating system.
- Servers with /390 architecture (**/390 servers** for short) for the Server Unit /390 of the FUJITSU Servers of the BS2000 SE Series and the Business Servers of the S Series
- Servers with x86 architecture (**x86 servers** for short) for the Server Unit x86 of the FUJITSU Servers of the BS2000 SE Series and the Business Servers of the SQ Series (x86-64 architecture)

The strings `<date>`, `<time>` and `<ver>` in examples indicate, for instance, the current displays for date, time and version when the examples are otherwise independent of date, time and version.

Different sets of notational conventions are used in this manual:

- SDF notational conventions for describing the syntax of commands and statements in SDF format.
  You will find descriptions of the SDF syntax in the “Commands” manual.
- conventions for emphasizing parts of the main text of the manual, and conventions for describing the syntax of commands and statements that are not in SDF format, both of which are referred to here as non-SDF notational conventions

Non-SDF notational conventions are the following:

UPPERCASE	Keywords
underscored	Default values
bold	Used for emphasis in running text; Text to be entered in examples
italics	Variables in syntax definitions and running text
`Fixed-space font`	Program text in examples; Text to be entered in examples of statement sequences
[]	Optional specifications in syntax definitions. The brackets are metacharacters and must not be entered in the statement.
{||}	Alternative specifications in syntax definitions. The available operands are separated by a horizontal bar.
()	Mandatory specifications in syntax definitions. Parentheses enclose the operand and form part of it. They must be entered in the statement.
`variable : : =`	Introduces a definition of the variables occurring in the preceding syntax definition.

... or . .	In syntax definitions, an ellipsis means that you can repeat the preceding specification any number of times. In examples, the ellipsis means that the rest of the statement is of no significance to the example. The ellipsis is a metacharacter and must not be entered in a statement.
	Indicates notes that are of particular importance.
	Indicates warnings.

2 Working with the SESAM/SQL DBH

The SESAM/SQL Data Base Handler (DBH) is the central component for the control, execution and monitoring of database operation. The DBH:

- analyzes the statements from the application programs, ascertains the best access method in each case, and executes the statements
- returns the results of database operations to the application program
- monitors all database activities
- controls the service tasks for utility functions
- logs important information on database activities in DBH-specific log files
- reports errors

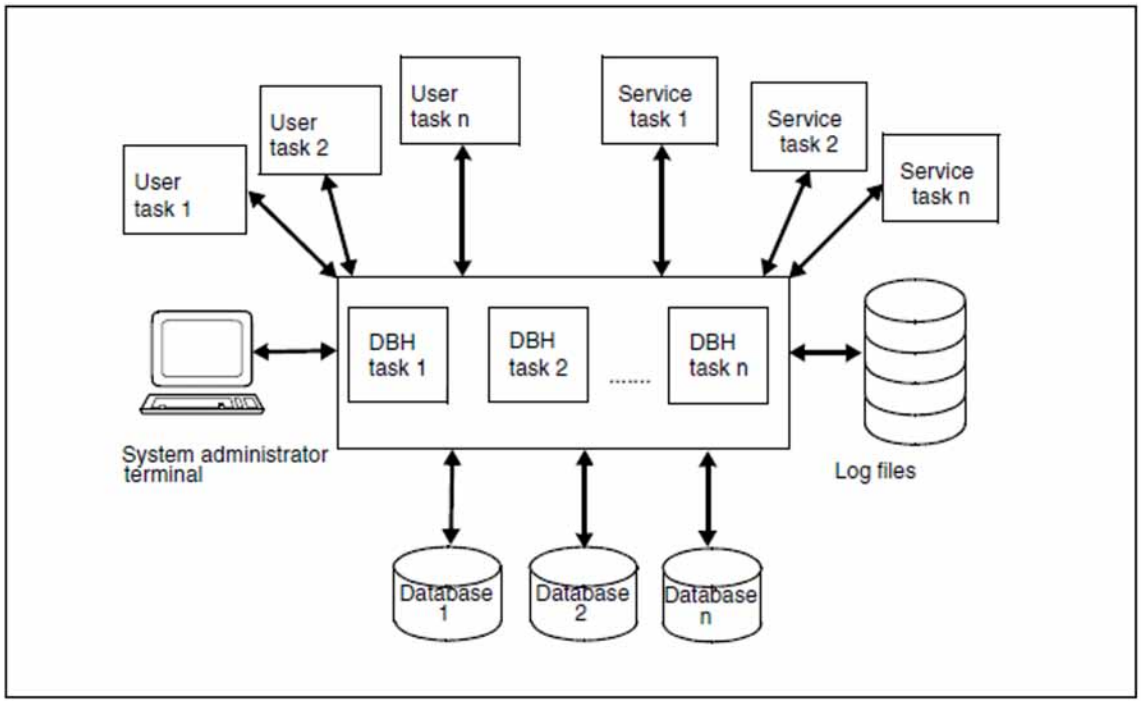


Figure 1: Tasks of the SESAM/SQL DBH

The “ Core manual” (see the chapter entitled “Database operation”) explains the fundamentals of the DBH, its functions and interfaces. You will also find an explanation in the chapter entitled “Backup concept” of how the DBH supports the security concept of SESAM/SQL.

This chapter covers the following topics:

- Start commands
- Starting the SESAM/SQL DBH and entering the start parameters
- Terminating the SESAM/SQL DBH
- DBH restart
- Job variables

- Maximum values for working with the SESAM/SQL DBH

2.1 Starting SESAM/SQL programs via start commands

All SESAM/SQL programs are started using SESAM start commands.

It is assumed that you have installed SESAM/SQL and CRTE (CRTE=**C**ommon **R**un **T**ime **E**nvironment) with IMON or you have stored the CRTE and SESAM libraries under the standard filenames described on "Tasks of the start commands". Further needed files are discussed in the chapter "requirements for starting".

> For reasons of compatibility, the earlier command START-PROGRAM may also be used. You can find information on this in the section “Starting SESAM programs with /START-PROGRAM”.

With SESAM/SQL-Server you can use different correction statuses in parallel. For specific information on this, refer to "Specifying the correction status with the start command".

2.1.1 Overview of the start commands

The start commands for all the SESAM/SQL programs are defined in the SDF syntax file shipped with the SESAM /SQL server. They are described in the respective descriptions for starting the programs.

Start command and aliases	Program
START-SESAM-DBH SESAM-DBH START-SESAM SESAM	SESAM
START-SESAM-DCN SESAM-DCN START-SESDCN SESDCN	SESDCN distribution component
START-SESAM-ADMINISTRATION SESAM-ADMINISTRATION START-SESADM SESADM	SESADM administration program
START-SESAM-PERF-MONITOR SESAM-PERF-MONITOR START-SESMON SESMON	SESMON
START-SESAM-TUNING-TRACE-EVAL SESAM-TUNING-TRACE-EVAL START-SESCOSP SESCOSP	SESCOSP
START-SESAM-RETRIEVAL-DIALOGUE SESAM-RETRIEVAL-DIALOGUE START-SEDI61 SEDI61	SEDI61
START-SESAM-CDML-DIALOGUE SESAM-CDML-DIALOGUE START-SEDI63 SEDI63	SEDI63
START-SESAM-LOG-FILE-EVAL SESAM-LOG-FILE-EVAL START-SEDI70 SEDI70	SEDI70
START-SESAM-UTILITY-MONITOR SESAM-UTILITY-MONITOR START-SESUTI SESUTI	Utility monitor SESUTI

START-SESLK-UTILITY-MONITOR SESLK-UTILITY-MONITOR START-SESUTIL SESUTIL	Utility monitor SESUTIL
START-SESAM-CDML-AUTHORIZATION SESAM-CDML-AUTHORIZATION START-SEPA SEPA	SEPA
START-SESLK-RETRIEVAL-DIALOGUE SESLK-RETRIEVAL-DIALOGUE START-SEDI61L SEDI61L	SEDI61L
START-SESLK-CDML-DIALOGUE SESLK-CDML-DIALOGUE START-SEDI63L SEDI63L	SEDI63L

2.1.2 Operands of the start commands

The essential operands of the start commands are described below, using START-SESAM-DBH as an example:

START-SESAM-DBH

VERSION = **<u>*STD</u>** / <product-version 6..10>

,**MONJV** = **<u>*NONE</u>** / <filename 1..54 without-gen-vers>

,**CPU-LIM**IT = **<u>*JOB-REST</u>** / <integer 1..32767>

,**RESID**ENT-**PAGE**S = **<u>*PAR</u>**AMETERS (...)

***PAR**AMETERS(...)

| **MIN**IMUM = **<u>*STD</u>** / <integer 0..32767 *4Kbyte*>

| ,**MAX**IMUM = **<u>*STD</u>** / <integer 0..32767 *4Kbyte*>

Operands

VERSION =

Specifies the program version to be started.

VERSION = <u>*STD</u>
No explicit product version specification. The product version is selected as follows:

1. The version specified with the command SELECT-PRODUCT-VERSION.
2. The highest version of the program that was installed using IMON.
3. The highest version of the program that exists in the SESAM module library.

VERSION = <product-version 6..10>
Selected version in the form: ['] [V] [m] m.naso ['] (e.g. 9.1A00).

MONJV = <u>*NONE</u> / <filename 1..54 without-gen-vers>
Name of the JV that is to monitor the program.
The system then sets the JV to the appropriate values during the program run:

$R	Program is running
$T	Program terminated normally
$A	Program terminated abnormally

This operand is only available to users with the JV software product (see also the “ Job variables” manual).

CPU-LIMIT = <u>*JOB-REST</u> / <integer 1..32767>

Maximum CPU- time in seconds that the program may utilize during its run. This operand corresponds to the CPU-LIMIT operand of the START-PROGRAM command. The time monitored depends on the operand RESOURCES=*PARAMETERS(CPU-LIMIT=...) in the SET-LOGON-PARAMETERS command.

CPU-LIMIT = <u>*JOB-REST</u>
Default setting:
Assigns the remaining CPU time available to the task to the program run as the maximum CPU time.

CPU-LIMIT = <integer 1..32767>
Explicit specification of the CPU time in seconds.

RESIDENT-PAGES = *PARAMETERS(...)
Number of resident memory pages required for the program run, see section “Using FASTPAM for SESAM/SQL”.

MINIMUM = *STD / <integer 0..32767 *4Kbyte*>
Minimum number of resident memory pages required.

MAXIMUM = *STD / <integer 0..32767 *4Kbyte*>
Maximum number of resident memory pages required.

2.1.3 Tasks of the start commands

The start commands have the following tasks:

- Ascertaining and assigning the system files
- Determining the operating mode

Ascertaining and assigning the system files

The start commands ascertain the system files required to start the program and assign the files by means of corresponding link names.

The names of the required system files are ascertained using IMON on the basis of the following algorithm:

- SESAM/SQL searches for the files of the selected version.
- CRTE searches for the most recent version.

If this is unsuccessful, the default names of the files are used (see next table).

If you have stored the CRTE and SESAM/SQL libraries with the default names specified in the table, you can also use the start commands without the need to install SESAM/SQL and CRTE with IMON.

The following list shows the files affected and their default names:

File	Link name	Standard name
CRTE library	BLSLIBxx	$.SYSLNK.CRTE for /390 servers; $.SKULNK.CRTE for x86 servers
SESAM module library	SESAMOML	$.SYSLNK.SESAM-SQL.<ver> for /390 servers; $.SKULNK.SESAM-SQL.<ver> for x86 servers
Procedure for CONNECT-SESAM-CONFIGURATION command		$.SYSSPR.SESAM-SQL.<ver>.RUN-CFG
Procedure for the start commands		$.SYSSPR.SESAM-SQL.<ver>.RUN-STA
FHS format library for SESMON	MAPLIB	$.SYSFHS.SESAM-SQL.<ver>.MON.E
FHS format library for the utility monitor (German/English)	MAPLIB	$.SYSFHS.SESAM-SQL.<ver>.UTI.D
	MAPLIB	$.SYSFHS.SESAM-SQL.<ver>.UTI.E
Help text file for the utility monitor (German/English)	SEEHELP	$.SYSMAN.SESAM-SQL.<ver>.UTI.D
	SEEHELP	$.SYSMAN.SESAM-SQL.<ver>.UTI.E

Table 2: Default names of the system files

The procedures for all commands are available in compiled form and can be run in a BS2000 environment.

Determining the operating mode

The programs are started in the interactive mode when the started via the start commands. If the user wants to run a program in the batch mode, then the user must call the start command in an ENTER procedure.

2.1.4 Specifying the correction status with the start command

The IMON installation monitor permits parallel administration of different versions of a software product on a single computer system. This means, you can also use multiple SESAM/SQL versions in parallel.

Points to note when starting a SESAM/SQL program

When you start a SESAM/SQL program by means of the start commands, you can specify the correction status with which the program is started by means of the VERSION parameter (see "Operands of the start commands"). If you do not specify this at startup, the most recent version of the LLM in the selected module library is loaded by default.

Points to note about the SESAM DBH in 24-hour operation

If a SESAM DBH is running in 24-hour operation, and if a new correction status is loaded into the SESAM module library during operation, the running DBH continues to work with the old correction status. When the DBH starts a new service task after the new correction delivery status is loaded, this service task is started with the appropriate correction status for the DBH.

If one or more subsystems of SESAM/SQL are loaded when a new correction status is loaded, the SESAM subsystems do not automatically load the new correction status.

You can also use correction statuses of SESAM/SQL in parallel in 24-hour operation and change to the new correction status without interrupting database operation, see the section “Parallel use of multiple correction statuses”.

Points to note when starting the performance monitor

If you want to start the SESAM performance monitor in this situation, note that it has to be started with the correction status of the SESAM DBH. In other words, you have to specify the correction status of the SESAM DBH to be monitored for the VERSION parameter of the start command for the performance monitor (e.g. VERSION=9.1A00).

As a result, some DBHs in a configuration may be working with the new correction status while others are still working with the old one. If you want all the DBHs in a configuration to be monitored by the performance monitor in this situation, you have to start a separate performance monitor for each correction status.

2.2 Starting the SESAM/SQL DBH

The SESAM/SQL DBH is a link and load module (LLM) called SESAM. When you start the SESAM LLM using the START-SESAM-DBH command or the START-PROGRAM command, this initializes the DBH session.

It makes it easier to use the SESAM/SQL Data Base Handler if you use IMON to install the product. If you do not do this, the files specified in the table “Default names of the system files” must exist with the default names specified in the table.

You parametrize the DBH at startup using DBH start statements and options.

> ¡ The section “Specifying the correction status with the start command” describes the points to note when starting the SESAM DBH with different correction statuses.

This section describes how to start the DBH and pass the start parameters. The syntax and functions of the various start parameters are described in detail in a separate chapter (see chapter “DBH start statements and options”).

The following topics are dealt with in this section:

- Requirements for starting
- Starting the independent DBH
- Starting the linked-in DBH
- Starting multiple DBHs
- SESAM/SQL share modules
- Dynamic reconfiguration of the DBH session

> ¡ The following always applies: Output sent to SYSOUT and SYSLST is not upwardly compatible. Output layout may vary depending on the version used.

64-bit load variant of the SESAM/SQL DBH

The 64-bit load variant of the SESAM/SQL DBH is loaded automatically on all current BS2000 servers with SESAM /SQL.
This can be recognized from the DBH start message on the insert “(64-Bit VERSION)” for /390 servers and “(X86-64-VERSION)” for x86 servers.

The 64-bit variant permits more powerful handling of the input/output load through a higher maximum value for the buffer for system-access data, see "SYSTEM-DATA-BUFFER".

2.2.1 Requirements for starting

Before starting the SESAM/SQL DBH with START-SESAM-DBH or START-PROGRAM, you assign files to the DBH and specify options for it by means of the following commands:

```
[/MODIFY-MSG-FILE-ASSIGNMENT ADD-FILE=$.SYSMES.SESAM-SQL.<ver>]          1.
[/MODIFY-SDF-OPTIONS-
         SYNTAX-FILE=*ADD(NAME=$.SYSSDF.SESAM-SQL.<ver>.USER)]           2.
[/MODIFY-TEST-OPTIONS DUMP=YES]                                          3.
[/ASSIGN-SYSDTA TO-FILE = *SYSCMD]                                       4.
[/ADD-FILE-LINK LINK-NAME=SESAMCID,FILE-NAME=conf_file]                  5.
[/ADD-FILE-LINK LINK-NAME=SESMAIL,FILE-NAME=mail_file]                   6.
```

1. To be able to use the SESAM/SQL messages, you need to activate the SESAM/SQL message file if this file is not available system-wide.
2. SESAM/SQL uses the SDF (**S**ystem **D**ialog **F**acility) dialog interface of BS2000 for entering the DBH start statement and options, among other things. The SESAM-SDF sub system syntax file must therefore be activated. You must activate the SDF user syntax file and may need to change the settings for the SDF options.
3. You use this command to output each memory dump and issue a corresponding message to SYSOUT in order to obtain the diagnostic documentation required in the event of a malfunction.
4. Assigns the SYSDTA system file to the SYSCMD procedure file. This assignment is not necessary if you enter the DBH start statements and options in an S procedure (structured SDF procedure, see the SDF-P manual “Programming in the Command Language”).
5. You use this command to assign the DBH a CATID list by means of a link name. *conf_file* must have the access type SAM and can contain a maximum of 50 CATIDs (see "MODIFY-CATID-LIST"). The CATID that you assign by means of the ADD-FILE-LINK command becomes effective at startup of the DBH.
6. You use this command to assign the DBH a MAIL parameter file by means of a link name. Its structure and its effect are described in the section “Sending important information of the DBH session by email”.

The SESAM/SQL DBH requests virtual address space when started. If there is not enough virtual address space available, loading is aborted and a corresponding error message appears. In this case, the BS2000 system administrator must increase the virtual address space for the user ID. If this is not possible, then you can try to reduce the amount of memory required by limiting the values of some DBH options (see the higher-order DBH option STORAGE-SIZE).

The requirements for the DBH restart are described in the section “Restart requirements” .

2.2.2 Starting the independent DBH

The independent DBH is implemented as a separate program system and runs in BS2000 with one or more DBH tasks. It is to be started in batch mode in order to avoid interruptions.

It is not recommended to start the DBH using several tasks in the same dialog as not all DBH tasks run in the same job class.

> SESAM/SQL only established communication with the applications after all DBH tasks are ready for operation. If this state is not achieved after a waiting period of 30 minutes, then the initialization is aborted with an appropriate error message. The start task outputs a message every 5 minutes until this time.

The independent DBH attempts to reserve the TP task category. For this reason the TP category should be authorized for the start user ID of the DBH as well as for the job class in which the DBH is started.

2.2.2.1 Command sequence for starting the independent DBH in batch mode

You start the DBH as a batch job by means of the command

```
/ENTER-PROCEDURE NAME= filename ,PROC-PAR=NO,LISTING=YES,LOGGING=YES
```

where *filename* is the name of an SDF-P procedure created by the user and in which the user enters the command sequence for starting the DBH:

```
/START-SESAM-DBH      1.
[//start_parameter]   2.
[//END]               3.
```

1. Starts the DBH using the start command START-SESAM-DBH (see section “Starting SESAM/SQL programs via start commands”).
2. You can enter the start parameters of the DBH here (see chapter “DBH start statements and options”). You do this when you do not want to accept the default values and do not assign the start parameters via an input file (see "Entering the DBH start parameters for the independent DBH").
3. The last END statement concludes entry of the DBH start parameters. This statement is necessary only if you do not enter the DBH start parameters via an input file.

Before the actual start command in the start procedure you can assign files to the DBH and change its options using the commands described on "Requirements for starting". You will find an example of a procedure on "Example of a procedure for starting the independent DBH".

2.2.2.2 Entering the DBH start parameters for the independent DBH

There are two ways to enter the DBH start parameters:

- You can pass them in the sequence of commands for starting the DBH.
- You can also pass them via an input file.

Parameter input is logged to SYSLST. You can use the LOGGING parameter of the MODIFY-SDF-OPTIONS statement to specify the extent of the logging to be carried out (see the BS2000 manual “ Commands”). It is advisable to select LOGGING=ACCEPTED-FORM, so that only the start parameters you enter are logged, not the default values.

DBH start parameters in the sequence of commands for starting the DBH

You can enter the start parameters in the sequence of commands for starting the DBH in the procedure.

To do this, you must assign the SYSDTA system file to SYSCMD before the START-SESAM-DBH command.

The start parameters must come immediately after the start command START-PROGRAM.

When entering the DBH start parameters, you must follow the rules on "Entering the DBH start parameters for the independent DBH".

Passing the DBH start parameters via an input file

Before you start the DBH by means of the START-SESAM-DBH command, you have to assign the input file. There are several ways to do this.

- You can assign the input file as a DBH configuration file using the link name SESCONF or via the SYSDTA system file (see "Entering the DBH start parameters for the independent DBH")

  > i A DBH configuration file with all DBH options and their current values can be created conveniently in the active DBH session using the administration statement SAVE-DBH-OPTIONS, see "SAVE-DBH-OPTIONS".

- You can assign the input file as a global configuration file using the CONNECT-SESAM-CONFIGURATION command (see "Entering the DBH start parameters for the independent DBH")

It is possible in interactive mode as well as batch mode to pass the DBH start parameters to the DBH by means of an input file (see "Starting the linked-in DBH").

Assigning a DBH configuration file

You assign the DBH configuration file using the link name SESCONF with the following command:

```
/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=configuration_file
```

You assign the DBH configuration file via the SYSDTA system file with the following command:

```
/ASSIGN-SYSDTA TO-FILE=configuration_file
```

Assigning a global configuration file

This global configuration file can contain configuration parameters for several components of the SESAM/SQL system (see "Starting the linked-in DBH"). The file must have the following structure:

```
CONFIGURATION-LINK = linkname-1                                          1.
        dbh-option | dcn-option | dbcon-option                               2.
CONFIGURATION-LINK = linkname-2                                          3.
        dbh-option | dcn-option | dbcon-option
CONFIGURATION-LINK = linkname-n
        dbh-option | dcn-option | dbcon-option
.
.
```

1. This string must begin in column 1.
2. The DBH options are not checked until the various tasks are executed. They can be separated by “//REMARK” comment lines.
3. The link names must be unique in the file.

Example of a global configuration file

```
//REMARK LOAD OPTIONS FOR THE DBH ********************************
CONFIGURATION-LINK=SESDBB1
//SET-DBH-OPTIONS-
//    DBH-IDENTIFICATION=*PARAMETERS(-
//        CONFIGURATION-NAME=Z-
//       ,DBH-NAME=X-
//    )-
//   ,ADMINISTRATION=*PARAMETERS(-
//        ACCOUNTING=*PARAMETERS(-
.
.
.
//REMARK K_2_DBH_NEW_1 ***************************************
//ADD-SQL-DATABASE-CATALOG-LIST-
//    ENTRY-1=*CATALOG(-
//        CATALOG-NAME=AUFTRAGKUNDEN-
//       ,USER-ID=KENN1-
//       )-
.
.
.
//REMARK LOAD OPTIONS FOR APPLICATION PROGRAMS ******************
CONFIGURATION-LINK=SESDBB2
REMARK CONFIGURATION DATA **************************************
CNF=Z
NAM=X
PUF=64000
TOTAL-USERS=00128
REMARK A DB PROCESS IS ASSIGNED SPECIFICALLY TO A UTM PROCESS *****
UTMVG=JA
VT
```

You can assign this file only with the following command:

```
CONNECT-SESAM-CONFIGURATION

 TO-FILE = <filename 1..54 without-gen-vers> / *LIBRARY-ELEMENT(...)

   *LIBRARY-ELEMENT(...)

     |  LIBRARY = <filename 1..54>

     |  ,ELEMENT = <composed-name 1..64 with-under>(...)

     |     |  VERSION = *HIGHEST-EXISTING / <composed-name 1..24 with-under>

     |  ,TYPE = S / <alphanum-name 1..8>

 ,CONFIGURATION-LINK = <alphanum-name 1..8>
```

Operands

TO-FILE = <filename 1..54>

You specify the name of the global configuration file here.

TO-FILE = *LIBRARY-ELEMENT(...)
If the configuration file is in a library, you make the appropriate entries here.

***LIBRARY-ELEMENT(...)**

LIBRARY = <filename 1..54 without-gen-vers>
You specify the library name here.

ELEMENT = <composed-name 1..64 with-under>(...)
You specify the element name here.

VERSION = *HIGHEST-EXISTING / <composed-name 1..24 with-under>
You specify a version of the element here.

TYPE = S / <alphanum-name 1..8>
You specify the type of the element here.

CONFIGURATION-LINK = <alphanum-name 1..8>
You specify here the link name to which the options for the DBH to be started are assigned in the global configuration file.

The options for the selected configuration are stored locally in a configuration file with the following name:

*t*SESCONF.*tsn*

t Temporary identifier: # or @
or @ If no temporary identifier is assigned at system startup, a permanent SESCONF.*tsn* file is assigned.

Notes on entering control statements

When entering the DBH start parameters, you must observe the following rules regardless of whether you enter them in a procedure or an input file:

- Each line must begin with the escape symbol (//).
- Each start statement must begin on a new line.
- If a start statement and its parameters take up several lines, each line except the last must be concluded with the continuation character (-).
- The continuation character (-) is not permitted at the end of a line if there is a new start statement or an END statement on the next line.
- The parameters must be entered in a fixed order (see chapter “DBH start statements and options”).
- The SET-DBH-OPTIONS statement must be followed by either another start statement or the END statement.
- The DBH start statements ADD-SQL-DATABASE-CATALOG-LIST and ADD-OLD-TABLE-CATALOG-LIST must both be followed by the END statement (see section “DBH start statements”).
- If ADD-SQL-DATABASE-CATALOG-LIST is not followed by the ADD-OLD-TABLE-CATALOG-LIST start statement, you must enter an END statement instead. This END statement can only be omitted when the DBH is started with the default setting for the DBH option OLD-TABLE-CATALOG, which is OLD-TABLE-CATALOG =0 (see "OLD-TABLE-CATALOG").

If one of the DBH start parameters is incorrect, the DBH aborts and you receive an error message.

These rules apply in interactive mode as well as batch mode.

2.2.2.3 Example of a procedure for starting the independent DBH

The following example shows a procedure for starting the independent DBH. A DBH with the DBH name x and the configuration name z is started as a batch job under the identifier $ID1. During the DBH session, SQL and CALL DML statements are to be executed. The DBH start parameters are entered in an S procedure called STARTP.SESDBH.ENTER.

Example

The command for initializing the batch job is:

```
/ENTER-PROCEDURE  NAME=STARTP.SESDBH.ENTER,-
                        PROC-PAR=NO,LISTING=YES,LOGGING=YES
```

The S procedure called STARTP.SESDBH.ENTER has the following structure:

```
/SET-PROCEDURE-OPTIONS
/ASSIGN-SYSLST TO-FILE=LST.SESDBH.E.DBH.IND
/MODIFY-MSG-ATTRIBUTES TASK-LANGUAGE=D
/MODIFY-TEST-OPTIONS DUMP=YES
/START-SESAM-DBH
//SET-DBH-OPTIONS-
//    DBH-IDENTIFICATION=*PARAMETERS(-
//        CONFIGURATION-NAME=Z -
//        ,DBH-NAME=X -
//    )-
//   ,ADMINISTRATION=*PARAMETERS(-
//       ADMINISTRATOR=*ANY(-
//            PASSWORD=C'XXX'-
//        )-
//    )-
//   ,SYSTEM-LIMITS=*PARAMETERS(-
//       SQL-DATABASE-CATALOG=10-
//      ,OLD-TABLE-CATALOG=5 -
//    )-
//   ,SQL-SUPPORT=*YES(-
//       INTERNAL-SORT-LIMIT=5000 -
//       ,INACTIVITY-TIME=1 -
//    )
// ADD-SQL-DATABASE-CATALOG-LIST -
//       ENTRY-1=*CATALOG(CATALOG-NAME=AUFTRAGKUNDEN) -
//       ,ENTRY-2=*CATALOG(CATALOG-NAME=CALLFIRMA)
//END
// ADD-OLD-TABLE-CATALOG-LIST -
//       ENTRY-1=*CALL-DML-TABLE(CALL-DML-TABLE-NAME=VERTRIEB -
//              ,CATALOG-NAME=CALLFIRMA -
//              ,SCHEMA-NAME=VERTRIEBSCH -
//              ,TABLE-NAME=VERTRIEB)-
//       ,ENTRY-2=*CALL-DML-TABLE(CALL-DML-TABLE-NAME=FIRMA -
//              ,CATALOG-NAME=CALLFIRMA -
//              ,SCHEMA-NAME=FIRMASCH -
//              ,TABLE-NAME=FIRMA)
```

```
//END
/ASSIGN-SYSLST TO-FILE=*PRIMARY
/EXIT-PROCEDURE
```

2.2.3 Starting the linked-in DBH

SESAM/SQL-LINK is a chargeable add-on product of the SESAM/SQL database system. It consists of the following components:

- linked-in DBH
- the SEDI61L utility (which outputs responses to CALL DML retrieval statements to a SAM file)
- the SEDI63L utility (which checks CALL DML statements).

You can use SESAM/SQL-LINK whenever a single application program is to have access to one or more databases. The linked-in DBH then processes the requests of this application program exclusively. It is linked to the application program directly; the program and the linked-in DBH belong to the same task.

SESAM/SQL-LINK is not available for x86 servers.
SESAM/SQL-LINK is only used in its 31-bit variant on all the /390 servers.

Application program and linked-in DBH

The linked-in DBH is loaded and started by the connection module linked to the application program.

Application programs that work with the linked-in DBH must also be linked to the SESLINK connection module (see the “Core manual”). When the first statement is executed, the SESLINK connection module automatically and dynamically loads the modules of the linked-in DBH from the SESAM/SQL module library, provided the module library was assigned appropriately before the application program started.

To ensure that all SESAM/SQL external references are resolved, the application program and SESLINK should be linked statically. SESLINK is already integrated in the SEDI61L and SEDI63L utilities.

The “ESQL-COBOL for SESAM/SQL-Server” and “CALL-DM Applications” manuals describe how an application program is linked and started.

Passing the start parameters

Users who want to use values other than the defaults can parameterize the linked-in DBH by means of the DBH configuration file (see the “ Core manual”). The DBH option LINKED-IN-ATTRIBUTES is used here to make settings specifically for the linked-in DBH.

The configuration file for the DBH start parameters is a SAM file with a freely definable name in which the user enters the DBH start statements and options. The same rules apply to the entry of the DBH start statements and options as apply to the entry of the DBH start parameters in an input file of the independent DBH (see "Entering the DBH start parameters for the independent DBH").

> i A DBH configuration file with all DBH options and their current values can be created conveniently in the active DBH session using the administration statement SAVE-DBH-OPTIONS, see "SAVE-DBH-OPTIONS".

Before the application program is started, the configuration file of the DBH must be assigned with one of the following commands:

- With the BS2000 command

  `ADD-FILE-LINK LINK-NAME = SESCONF, FILE-NAME = `*`configuration_file`*

- With the SESAM/SQL command (see "Entering the DBH start parameters for the independent DBH")
  `CONNECT-SESAM-CONFIGURATION TO-FILE = `*`global configuration_file`*

When the linked-in DBH starts, the start parameters are passed to it.

If one of the configuration-file parameters to be evaluated contains syntax errors, the file is closed and the associated application program receives an error message.

If a configuration file is not assigned before the application is started, the linked-in DBH outputs a system prompt (//). The user can then enter the DBH start statements and options directly on screen.

2.2.4 Starting several SESAM/SQL DBHs

You can start more than one DBH on a single computer, regardless of whether they are independent or linked-in. You can manage parts of the DBH as share modules.

> i It is important to differentiate between the ability to start several DBHs from a multitasking operation. The first option refers to starting several databases and then having processes from different DBHs run. The second option provides the ability to use several DBH tasks running in parallel for task runs from one and the same database. See the “Core manual” for more detailed information.

Assigning and naming a DBH

To avoid conflicts, you should group related applications into configurations, thus separating them from those in other configurations (see the “Core manual”). You place a DBH in a configuration by assigning it the configuration name (see the description of the DBH option CONFIGURATION-NAME on "CONFIGURATION-NAME"). The configuration name must be unique on the computer.

You use the DBH option DBH-NAME (see "DBH-NAME") to assign the DBH name. The DBH name must be unique in the configuration. If it is not, name conflicts can occur between DBH-specific files that include the DBH name and configuration name in their names (the TA-LOG files, WA-LOG file and cursor files).

2.2.5 SESAM/SQL share modules

To reduce loading times or main-memory requirements, you can make use of the possibility to load SESAM LLMs via DSSM as subsystems.

Loading SESAM-LLMs as subsystems

SESAM delivers several LLMs that can be loaded by the dynamic subsystem management facility (DSSM) of BS2000. Hence several tasks in BS2000 can use these modules shared.

Some communication modules and parts of SESDCN can be loaded into class 4 memory as subsystems under DSSM with the following command:

```
/START-SUBSYSTEM
   SUBSYSTEM-NAME=<SESAM_subsystem_name>,
   VERSION=<SESAM_product-version mandatory-man-corr>,
   VERSION-PARALLELISM=*COEXISTENCE-MODE
```

Specifying `VERSION-PARALLELISM=*COEXISTENCE-MODE` (coexistence permitted) enables you later to start multiple correction statuses of SESAM/SQL in parallel, thus facilitating the migration to a new correction status during ongoing operation. If you do not want this, you can use the default value `VERSION-PARALLELISM=*NONE`. The `VERSION-PARALLELISM=*EXCHANGE-MODE` operand is not supported by SESAM/SQL.

Example

```
/START-SUBSYSTEM SUBSYSTEM-NAME=SESKOM,VERSION=9.1A00,-
/                VERSION-PARALLELISM=*COEXISTENCE-MODE
```

DSSM controls the dynamic loading, initialization, suspension, continuation and termination of the subsystems (see the “Subsystem Management (DSSM/SSCM)” manual).The START-SUBSYSTEM command is described in the “Commands” manual.

Parallel use of multiple correction statuses

With SESAM/SQL-Server, several correction statuses can be used in parallel. If one or more subsystems of SESAM /SQL are already loaded when a new correction delivery status is loaded, the SESAM subsystems do not automatically load the new correction delivery status.

If you have started the systems with coexistence permitted as described in the section “Loading LLMs as subsystems”, you can start the subsystems with the new correction statuses parallel to the running subsystems using START-SUBSYSTEM without impairing database operation, e.g.

```
/START-SUBSYSTEM SUBSYSTEM-NAME=SESKOM,VERSION=9.1A10,-
/                VERSION-PARALLELISM=*COEXISTENCE-MODE
```

Subsequently all newly started SESAM/SQL tasks use the new subsystems. When all SESAM/SQL tasks which were not yet connected with the “old” subsystems have been terminated, the subsystems with the old correction status can be terminated.

As an alternative to the process described above you can terminate all tasks which are attached to the SESAM subsystems. You must then also stop the subsystems with the

`/STOP-SUBSYSTEM SUBSYSTEM-NAME=<SESAM_subsystem_name>` command.

Database operation will then be interrupted. The SESAM subsystems must then be restarted using the command:

```
/START-SUBSYSTEM
    SUBSYSTEM-NAME=<SESAM_subsystem_name>,
    VERSION=<SESAM_product-version mandatory-man-corr>,
    VERSION-PARALLELISM=*NONE/*COEXISTENCE-MODE
```

SESAM/SQL can then restart the tasks. These tasks now load the new correction status (see also the section “Specifying the correction status with the start command”).

Lists of the SESAM/SQL share modules

The tables below list the available SESAM/SQL share modules. The column entitled “DSSM subsystem” indicates the subsystems into which the LLMs are grouped when loaded with DSSM.

In the LLM names, “*xx*” is a placeholder for the digits in the version number of SESAM/SQL.

LLM name	Meaning	DSSM subsystem
SECDN*xx*	SESDCN share module	SESKOMMD

Table 3: Share module of SESDCN, the distribution component for distributed processing with SESAM/SQL-DCN

LLM name	Meaning	DSSM subsystem
SECLAxx	Communication LLM for non-distributed processing	SESKOMOD
SECVAxx	Communication LLM for distributed processing	SESKOMMD
SERTSxx	SQL runtime system	SESSQLRT

Table 4: communication share modules

LLM name	Task types that use the share module	DSSM subsystem
SEZCSxx	DBH, SESDCN and independent communication in the application program	SESKOM for /390 servers
SEJSVxx	DBH and independent communication in the application program	
SECCSxx	Independent communication in the application program	SESKOM for x86 servers
SECSVxx	Independent communication in the application program	

Table 5: Share modules used by several task types

2.2.6 Dynamic reconfiguration of the DBH session

SESAM/SQL supports two administration commands are available for increasing the availability of the independent DBH, for importing a correction version of the DBH while DBH operation is in progress, and for dynamic reconfiguration of the DBH session:

- RECONFIGURE-DBH-SESSION
- RELOAD-DBH-SESSION

Both administration statements are executed without interrupting the DBH session. From the user viewpoint no DBH failure occurs.

The following DBH options can be changed in the ongoing DBH session with RECONFIGURE-DBH-SESSION:

- DBH option COLUMNS
- DBH option CURSOR-BUFFER, parameters BUFFER-SIZE and FRAME-SIZE
- DBH option SPACES
- DBH option SQL-SUPPORT, parameters CURSORS and PLANS
- DBH option SYSTEM-DATA-BUFFER
- DBH option TRANSACTION-SECURITY, parameter MAX-ISOLATION-LEVEL
- DBH option USER-DATA-BUFFER

The DBH modules are reloaded with RELOAD-DBH-SESSION. In this case the following DBH options can be changed in the ongoing DBH session:

- DBH option DBH-TASKS
- DBH option SYSTEM-THREADS, parameter WRITE-THREADS
- DBH option THREADS
- DBH option USERS

To reload the DBH modules SESAM/SQL must be installed with IMON. SESAM/SQL determines the current installation path and uses it to load the most up-to-date correction status of the DBH automatically. The new correction status of the DBH can have been installed in parallel to the version used to date, see section “Parallel use of multiple correction statuses”.

When a new correction status of the DBH is loaded, SESAM performance monitors which monitor this DBH must be restarted.

When the DBH session is dynamically reconfigured, the DBH’s memory is restructured or re-created. This also causes the counters which the SESMON performance monitor evaluates for statistical purposes to be reset to “0”.

Executing administration statements

To execute one of these statements, a transaction-free status of the DBH session is first required. Open transactions are aborted or reset by the DBH.

> Before entering a statement, use the administration statement SHOW-TRANSACTIONS to check whether transactions are still active, and if necessary use the administration statement HOLD-TRANSACTION-ADMISSION to prevent any further transactions.

In the following cases the DBH cannot abort transactions. In these cases the administration statement is aborted with a corresponding message.

- A transaction is in the processing phase Prepare To Commit (PTC)
- A utility statement is being processed
- A lock sequence has occurred

The start of DBH reconfiguration is reported in message `SES6090` (independently of the DHB option MSG-OUTPUT) on the console or terminal and to SYSLST.

When an error occurs, if for example the DBH cannot be reconfigured because of a storage shortage, reconfiguration is aborted. If an external restart is initiated after reconfiguration has been aborted, the DBH is started with the DBH options used to date.

As with a start or restart, the operability of the DBH is displayed with the message `SES0060 SESAM SYSTEM READY.`

2.3 Terminating the SESAM/SQL DBH

This section describes how to terminate the SESAM/SQL DBH and thus the DBH session, depending on the operating mode.

2.3.1 Terminating the independent DBH

You terminate the independent DBH by means of the SESADM administration program with the STOP-DBH administration statement or the STOP-DBH administration command (see "STOP-DBH").

In the case of CALL DML applications, you can also terminate the DBH by means of a CALL DML administration statement issued from a TIAM application or a UTM program unit. However, to do this, you must be authorized by means of the system user ID of the system administrator entered in the DBH option ADMINISTRATOR to use the CALL interface for the purpose of administration (see "ADMINISTRATOR").

In a SESAM/SQL UTM application, you should always terminate the DBH with the following administration statement (or the corresponding administration command):

```
STOP-DBH UTM-SESSION-INFO=KEEP
```

The parameter UTM-SESSION-INFO=KEEP (default value) causes the log files of the DBH to remain active and the restart information they contain to be preserved.

2.3.2 Terminating the linked-in DBH

The linked-in DBH can be terminated only by means of the STOP-DBH administration command.

In the case of the linked-in DBH, the administration command does not take effect until the next statement is issued to the DBH in the application program and control is thus passed to the DBH.

If no further statements are issued during the program run, the linked-in DBH is terminated when the application program is terminated.

2.4 Restarting a DBH session

The SESAM/SQL DBH carries out a restart when operation resumes after interruptions caused by malfunctions (see the “ Core manual”).

During a restart, all logically inconsistent databases are opened and their consistency is restored. The DBH obtains information on the processing status of the database transactions from the transaction log files (TA-LOG files) and the restart log file (WA-LOG file).

Depending on whether or not the current DBH session is interrupted, the DBH carries out an external or internal restart:

- It carries out an external restart when it is started after an aborted session.
- It carries out an internal restart without interrupting operation after internal errors of lesser importance.

Both the independent DBH and the linked-in DBH can carry out a restart.

In a UTM application, the SESAM/SQL DBH and openUTM carry out a synchronized restart (see the “Core manual ”).

> You can also obtain information on the progress of the DBH session restart in the DBH form SYSTEM INFORMATION (see "DBH form SYSTEM INFORMATION"). During a DBH session information on the number of logging blocks accrued for a restart is output in this form.

2.4.1 Restart requirements

Before the DBH can carry out a restart, the following requirements must be met:

- The DBH must be running with transaction management (default mode).
- The TA-LOG files (transaction log files) must be available.
  Its default name is:
  - SESAM*cn*.TA-LOG1 and SESAM*cn*.TA-LOG2 for the independent DBH and
  - SESLK*cn*.TA-LOG1 and SESLK*cn*.TA-LOG2 for the linked-in DBH.
- The WA-LOG file (restart log file) must be available.
  Its default name is:
  - SESAM*cn*.WA-LOG for the independent DBH and
  - SESLK*cn*.WA-LOG for the linked-in DBH.

The TA-LOG files and WA-LOG file include in their names the configuration name (*c*) and the DBH name (*n*) defined by means of the DBH option DBH-IDENTIFICATION when the aborted session was started.

If the TA-LOG files and WA-LOG file are not available, the DBH cannot carry out the restart. The database administrator then has to use the appropriate utility statements to repair logically inconsistent databases (see the “SQL Reference Manual Part 2: Utilities”).

2.4.2 Controlling the duration of the restart

The duration of the restart essentially depends on the duration of the physical repair and the logical resetting of the open transactions. The aim is to reduce the duration of the restart to a minimum. The following sections explain the options provided to you for this purpose by the DBH.

Reducing the duration of physical repair

In physical repair, the afterimage blocks are written from the TA-LOG files to the databases.

The more often the afterimage blocks are saved during normal operation, the fewer data blocks have to be written in the event of a restart. This reduces the time required before normal operation can be resumed after an interruption. You can increase the frequency with which the afterimage blocks are written during the session by means of the BUFFER-LIMIT and TALOG-LIMIT parameters of the DBH option RESTART-CONTROL (see "RESTART-CONTROL").

However, too much saving during normal operation can affect performance. You should therefore use the performance monitor to keep track of the I/O rates and make changes, if necessary, to the BUFFER-LIMIT and TALOG-LIMIT parameters during operation using the MODIFY-RESTART-CONTROL administration statement (see "MODIFY-RESTART-CONTROL").

Delaying logical resetting in normal operation

In a logical reset, the transactions that were open at the last consistency point are reset on the basis of the logical beforeimages.

When LOGICAL-ROLLBACK = *DELAYED is set with the DBH option RESTART-CONTROL (see "RESTART-CONTROL"), this reset can be delayed until normal operation begins after a restart. The reset operations then run parallel to the normal transactions and requests of users and are synchronized accordingly as in normal operation.

Delaying of the logical reset is suppressed in the following cases, even if the option described in the previous paragraph is used:

- In the event of a restart because of a bottleneck affecting the transaction log files, since the removal of the bottleneck is not to be expected before the open transactions are reset.
- In the event of a restart because of a bottleneck affecting a DA log file. In this situation, it is only possible to read the databases. The open transactions therefore have to be reset here as well before a restart.
- In the event of a restart because of a defective catalog space. In this case, too, the transactions have to be reset immediately, since the repaired data and index spaces are physically closed at the end of the restart on account of the missing catalog space.

2.4.3 Carrying out an external restart

After an aborted session you start the DBH in the usual way. If required you can also start the DBH on a different system from that of the aborted session. A switchover of the server architecture (/390 server <---> x86 server) is also possible here.

However, when passing the DBH start parameters, note the following:

- The configuration name and DBH name of the aborted and restarted sessions must be the same (see the DBH options “CONFIGURATION-NAME” and “DBH-NAME”). An external restart is only carried out if you start a DBH with the same configuration name and DBH name as the aborted session. Only in this case can the DBH evaluate the restart information in the corresponding TA-LOG files and WA-LOG file.
- If you do not want to use the default values for one or more parameters, you have to enter the DBH option MEDIA-CATALOG (see "MEDIA-CATALOG").
- You must enter the DBH options CURSOR-BUFFER, SYSTEM-DATA-BUFFER and USER-DATA-BUFFER (see "CURSOR-BUFFER"ff) if you wish to use a value other than the default value for one or more of these options.

  > **i** In the event of an error (see "Requirements for starting") the BS2000 system administrator must enlarge the ID’s virtual address space. If this is not possible, you can attempt to reduce the storage requirements using restrictions in some DBH options (see the higher-order DBH option STORAGE-SIZE, "STORAGE-SIZE").

- The DBH adopts the other DBH start parameters from the WA-LOG file. Any additional start parameters you enter are ignored. The LOGICAL-ROLLBACK = IMMEDIATELY / DELAYED parameter of the RESTART-CONTROL administration statement is an exception to this because. Here you can also change its value subsequently.

If one of the relevant databases or a space is not available, you receive a message to this effect. The DBH expects one of the following responses:

R	to attempt to open it again
I	to skip the database or space and resume processing
H	to abort the restart

Even when databases or spaces cannot be made available to the DBH, it still tries to complete the restart successfully. These databases or spaces can then be repaired by means of the RECOVER utility statement (see the “ SQL Reference Manual Part 2: Utilities”).

During the restart, the DBH does not process any requests; administration statements or commands cannot be executed during this phase.

Once the restart has been successfully completed, the databases remain open and application programs can again work with them.

2.5 Job variables

Where necessary SESAM-DBH operates with the following types of job variables:

- Job variables for controlling requests and programs
- Monitor job variables for job monitoring

2.5.1 Job variables for controlling requests and programs

SESAM/SQL allows job variables to be used in a DBH session to control requests and programs. The software product JV (BS2000) Job Variables must be installed for this purpose (see the “ JV (BS2000)” manual). If JV is not available, the creation and definition of job variables are refused without warning.

To indicate the readiness of the DBH for communication, you can use job variables to document the following statuses:

- the beginning of the DBH session
- the end of the DBH session
- errors during the DBH session
- current status of the DBH session
- WA-LOG and TA-LOG status
- creation of a replication.
  If the replication is located in a BS2000 user ID that is not the same as the DBH user ID, an attempt is made to create the job variable also in the replication ID or use a job variable already created under this name. SESAM /SQL can only create the job variable in the replication ID when the DBH user ID is entered as a co-owner for the job variable. If the job variable cannot be created in the replication ID, or if it is not created there, it is created in the DBH user ID.

The overview given below indicates the times at which the SESAM/SQL DBH sets the job variables.

The DBH always creates job variables as shared job variables (USER-ACCESS=*ALL-USERS). If other attributes are to be used, the job variables must be suitably set up manually by the database administrator beforehand.

Time	**Job variable name**	**Job-variable contents**
Between the start and end of the DBH session	SESAM.SESDBH.*cn* *c* means: configuration name (see "CONFIGURATION-NAME") *n* means: DBH name (see "DBH-NAME")	Date *YYYY-MM-DD* Time *hh:mm:ss* DBH-STARTED DBH_NORMAL_END| DBH_ABNORMAL_END| DBH_ENDED_WITH_WARNINGS TA-LOG ACTIVE | NOT ACTIVE System error: CC XX/YY/ZZ | IW xxx Termination: IW xx Message: yyyyyyy Where: CC XX/YY/ZZ: Consistency Check with number, weight and subnumber IW xxx: The interrupt weight yyyyyyyy: The message code (see the “ Messages” manual)
The first update in the DBH session, or when the session is ended correctly	SICHERUNGSINFORMATION.*cn*	Status of the files SESAM*cn*.TA-LOG1 SESAM*cn*.TA-LOG2 SESAM*cn*.WA-LOG SESLK*cn*.TA-LOG1 SESLK*cn*.TA-LOG2

		SESLK*cn*.WA-LOG Status of the transaction management files: o.k. Restart UTM information
When creating a replication	*user-id*.SESAM. *replication*.NEXT-REPL-LOG	The oldest CATLOG and DALOG file required for the first REFRESH REPLICATION (see the “ Core manual”)

Table 6: Job variables

> The job variable SESAM.SESDBH. *cn* does not contain “System error: IW xx” any more for STXIT events of class ABEND and TERM, rather it contains “Termination: IW xx” (xx: STXIT interrupt weight) because these STXIT events are not system errors. You will find the meanings of the STXIT interrupt weights in the manual “ Executive Macros” in the description of the STXIT macro. If an internal transaction restart (e. g. due to a logging bottleneck) cannot be executed and the session must be terminated, then the job variable SESAM.SESDBH. *cn* contains “Message: yyyyyyy” (yyyyyyy: message code) where the corresponding message contains the cause of the failed restart.

If necessary, you can use predefined macros to change names and explanatory texts in the job variables. The predefined macros are in the SEZTXT.ASS source file which is shipped as part of the SIPANY.SESAM-SQL.<ver>. SPEZ library.

The SEZTXT.ASS source file contains the following macros:

`JOBV` *`name`*`[,[`*`len`*`][,MONJV]]`

This macro declares a job variable called *`name`*.

`name` The name of the job variable (maximum length: 54 characters)

`len` The length of the job-variable name's suffix

`MONJV` This defines the job variable as a monitoring job variable. In this case the text is not stored until position 128 (i.e. after the system part) in the job variable.

TEXT *text*

This macro generates contents in the form of a text for the job variable called *name*.

`text` A text string consisting of any characters (C'...') or hexadecimal characters (X'...') and with a maximum length of 54 characters. Several text strings can refer to a single job variable.

INSRT *len*

This macro generates an insert with a length of *`length`* for the job variable called *name*. As a result, a text of the specified length is subsequently inserted dynamically. The macro can be specified several times for a single job variable.

`len` The length of the insert (maximum length: 256 characters)

You can change the name of the job variable in the JOBV macro, and you can change the name of the text string in the TEXT macro. The entries in the INSRT macro cannot be changed. Note that the total length of all texts and inserts for a job variable cannot be more than 256 characters. After each change, you have to assemble the source file and enter it in the SESAM/SQL module library under the module name SEZTXT.ASS. You will find the source module in the SESAM/SQL library with special elements.

2.5.2 Monitor job variables for job monitoring

SESAM-DBH supplies an existing monitoring job variable at the same time, with the job variable SESAM.SESDBH.*cn*.

The content of the monitoring job variable is described in the manual “ JV (BS2000)”, section entitled “Request /program monitoring with JVs”.

SESAM/SQL supplies the following fields:

- “Name of the application to be monitored” (Bytes 53-60) with “SESDBH*cn*”
- “Application-specific information” (bytes 71-128) is supplied with the content of the job variable SESAM.SESDBH.*cn* without “date” and “time” (see the table “Job variables”).

2.6 Sending important information of the DBH session by email

You can also transmit important information of the DBH session automatically by email.

To do this, in the MAIL parameter file you enter the recipient, sender and relevant message numbers for the email outputs of the DBH.

You define the type and scope of the DBH email output using DBH options and administration statements.

Prerequisites for email output of the DBH

The mail service of the software product interNet Services must be available.

The email is actually sent by the DBH with the mail transmitter (SEND-MAIL interface) of the software product interNet Services.

MAIL parameter file

The MAIL parameter file is a BS2000 file which the user creates and assigns to the DBH via the link name SESMAIL at startup, see "Requirements for starting".

If you assign the MAIL parameter file to a DBH before it is started, the parameters set there become effective when the DBH is started. Otherwise you can (initially) not use the email output of the DBH.

The MAIL parameter file can be modified while the DBH session is active. However, the (modified) parameters become effective only when the administration statement MODIFY-MAIL-PARAMETERS (see "MODIFY-MAIL-PARAMETERS") is entered. The administration statement SHOW-MAIL-PARAMETERS (see "SHOW-MAIL-PARAMETERS") enables you to display the current MAIL parameters.

Structure of the MAIL parameter file

The MAIL parameter file must be a SAM file and can, for example, be created using the file editor EDT (see the “EDT (BS2000)” manual).

The specifications in the parameter file must satisfy the following rules:

- Each line contains precisely one parameter specification.
- Lines beginning with # are ignored (comment lines).
- All other lines must begin with the keyword “ADDRESS=”, “SENDER=” or “MSG=”. Blanks make no sense in these lines: they terminate analysis of the input.
- The `ADDRESS=<email-address>` line is used to specify the email address to which the emails of the DBH are sent. You can also specify more than one email address, each separated by a semicolon. The `<email-address>` can also be enclosed in single quotes. SESAM/SQL does not check the validity of an email address. Nor does SESAM/SQL check whether the email transmission was successful. Answers to the email go to the email address entered under “SENDER=”.
  If the “ADDRESS=” line is specified more than once, only the last “ADDRESS=” line specified is used.
- The `SENDER=<email-address>` line is used to specify an email address which is entered as the sender of the email. The `<email-address>` can also be enclosed in single quotes. SESAM/SQL does not check the validity of an email address. If no sender address is specified, the email has no sender.

Specifying an unsuitable sender can lead to problems with the recipient's mail server or to unwanted processing by SPAM filters.

- The MSG=<msg-no> lines are used to specify message numbers whose occurrence is to result in an email notification being sent (depending on the IMMEDIATE=*YES/*NO option).

  When message numbers are specified, right-justified wildcard characters * (any character) may be used to form message groups (e.g. all bottleneck messages can be entered with MSG=SES60**).

  Each message number or message group must be specified in a separate line. Up to 100 lines with message numbers or message groups can be specified. A message number can be specified multiple times, also within message groups with wildcards. However, when it occurs the message is only sent once.

Example of a MAIL parameter file

```
##############################################
# MAIL parameter file for the SESAM/SQL DBH
##############################################
ADDRESS=NAME1@COMPANY.COM
SENDER=NAME2@COMPANY.COM
# Display start message
MSG=SES0060
# Display of bottleneck messages
MSG=SES60**
```

Options and statements for controlling email output of the DBH

With the MODIFY-OUTPUT-MODE statement (see "MODIFY-OUTPUT-MODE") you can define additional output by email as the output destination for all the following administration statements.

You set the scope of the DBH's email outputs with the DBH options MSG-OUTPUT (see "MSG-OUTPUT") and SERVICE-TASKS (see "SERVICE-TASKS"). The administration statements MODIFY-MSG-OUTPUT (see "MODIFY-MSG-OUTPUT") and MODIFY-SERVICE-TASKS (see "MODIFY-SERVICE-TASKS") allow you to modify the scope of the DBH's email outputs.

3 DBH start statements and options

This chapter describes the syntax and functions of the DBH start parameters and includes notes on how to enter them.

The chapter is divided into the following sections:

- Overview and notes on entry
- Descriptions in alphabetical order

The different ways of passing the DBH start statements and options when starting the DBH are described in section “Starting the SESAM/SQL DBH”.

3.1 Overview and notes on entry

The purpose of DBH start statements and options is to parametrize the DBH. You use them to specify the resources, operating rules and limit values of the current DBH session.

You enter DBH start statements and options in SDF format. You can enter them in a prompted or nonprompted dialog, from procedure files or in batch mode. You will find a detailed description of the SDF dialog interface (**S**ystem **D**ialog **F**acility) in the “ SDF Dialog Interface”.
The DBH start statements and options are supplied in the user syntax file SYSSDF.SESAM-SQL.<ver>.USER.

In this section, the DBH start statements and options are arranged in different tables, depending on their contents. The section also contains information on entering the start parameters.

3.1.1 DBH start statements

The DBH start statements initiate the parametrization of the DBH.

The following table lists the DBH start statements in the order in which they are entered:

DBH start statement	Function
SET-DBH-OPTIONS	Causes the DBH options to be read in.
ADD-SQL-DATABASE-CATALOG-LIST	Inserts entries in the SQL database catalog.
ADD-OLD-TABLE-CATALOG-LIST	Inserts entries in the CALL DML table catalog.

Table 7: DBH start statements

You must enter the SET-DBH-OPTIONS statement first. It is followed by the DBH options you want to use.
The other two start statements are optional, since you can also make entries in the SQL or CALL DML table catalog using administration statements (see section “ADD-OLD-TABLE-CATALOG-ENTRY” and section “ADD-SQL-DB-CATALOG-ENTRY”).
However, if you want to insert entries in the CALL DML table catalog by means of the ADD-OLD-TABLE-CATALOG-LIST start statement when starting the DBH, you must first issue the ADD-SQL-DATABASE-CATALOG-LIST start statement.

END statement

When entering the DBH start statements, you must always use the END statement to conclude parameter input.

The END statement is required at the following points during parameter input:

- after the SET-DBH-OPTIONS statement, if this statement concludes parameter input
- after the ADD-SQL-DATABASE-CATALOG-LIST start statement
- after or instead of the ADD-OLD-TABLE-CATALOG-LIST start statement; you can only omit this END statement when the DBH is started with the default setting for the lowerlevel DBH option OLD-TABLE-CATALOG, which is OLD-TABLE-CATALOG=0 (see "OLD-TABLE-CATALOG").

Example 1

The END statement concludes parameter input after the SET-DBH-OPTIONS statement.

```
//SET-DBH-OPTIONS ...
//END
```

Example 2

The first END statement concludes the ADD-SQL-DATABASE-CATALOG-LIST start statement and the second concludes all parameter input.

```
//SET-DBH-OPTIONS ...
//ADD-SQL-DATABASE-CATALOG-LIST ...
```

```
//END
//ADD-OLD-TABLE-CATALOG-LIST ...
//END
```

Example 3

The first END statement concludes the ADD-SQL-DATABASE-CATALOG-LIST start statement and the second concludes all parameter input. The ADD-OLD-TABLE-CATALOG-LIST start statement is not included.

```
//SET-DBH-OPTIONS ...
//ADD-SQL-DATABASE-CATALOG-LIST ...
//END
//END
```

3.1.2 DBH options

DBH options parametrize the DBH and define the essential characteristics of a session. You can modify most of the DBH options during ongoing operation (with the exception of CONFIGURATON-NAME and DBH-NAME), see section “Administration statements for changing DBH options”.
You can save the current DBH options to a file and use them again in the next DBH session, see "SAVE-DBH-OPTIONS".

DBH options are subdivided into higher- and lower-level DBH options.

- Higher-level DBH options, such as ADMINISTRATION or STORAGE-SIZE, relate to a particular subject area.
- Lower-level DBH options each deal with part of the subject area of their higher-level option.

All the higher-level DBH options are assigned one or more lower-level options.

You enter the DBH options after entering the DBH start statement SET-DBH-OPTIONS.

There are default settings for all DBH options. Therefore, you only need to enter those options for which you want to specify values other than the defaults. However, before you can specify your own values for one or more lower-level options, you must first enter the associated higher-level option.

The following overview shows the DBH options that are available.

Overview of the DBH options

Higher-level option	Lower-level option	Brief description
DBH-IDENTIFICATION		**Identifies the DBH**
	CONFIGURATION-NAME	Defines the configuration name
	DBH-NAME	Defines the DBH name
ADMINISTRATION		**Administers the DBH**
	ACCOUNTING	Activates accounting
	ADMINISTRATOR	Assigns administration authorization
	MSG-OUTPUT	Controls the outputs of the DBH
	SECURITY	Changes security settings
CPU-RESOURCES		**Controls the CPU utilization**
	DBH-TASKS	Specifies the number of the DBH tasks
	SERVICE-TASKS	Specifies the number of tasks for service requests
FILE-RESOURCES		**Specifies settings for files**
	MEDIA-CATALOG	Creates a media catalog
	SESSION-LOGGING-ID	Identifies session-specific files

LINKED-IN-ATTRIBUTES	CODED-CHARACTER-SET	**Specifies attributes of the linked-in DBH** Specifies a coded character set which is used by the user program.
RECOVER-OPTIONS		**Specifies settings for RECOVER or REFRESH runs**
	SYSTEM-DATA-BUFFER	Dimensions the system-data buffer
	USER-DATA-BUFFER	Dimensions the user-data buffer
	MEDIA-CATALOG	Creates the media catalog
STORAGE-SIZE		**Creates buffer and container size**
	CURSOR-BUFFER	Dimensions the cursor buffer
	SYSTEM-DATA-BUFFER	Dimensions the system data buffer
	TRANSFER-CONTAINER	Dimensions the transfer container
	USER-DATA-BUFFER	Dimensions the user data buffer
	WORK-CONTAINER	Dimensions the work container
SYSTEM-LIMITS		**Specifies limit values**
	COLUMNS	Enlarges the area for retrieval statements
	OLD-TABLE-CATALOG	Specifies the maximum number of entries permitted in the CALL DML table catalog
	SPACES	Specifies the maximum number of simultaneously accessible spaces
	SQL-DATABASE-CATALOG	Specifies the maximum number of entries permitted in the SQL database catalog
	SQL-SUPPORT	Specifies limit values for the SQL interface
	SUBORDERS	Makes SQL scans or logical files available
	SYSTEM-THREADS	Specifies the number of concurrent system threads
	THREADS	Specifies the maximum number of concurrent threads
	USERS	Specifies the maximum number of concurrent users
		Defines the processing strategy

SYSTEM STRATEGIES	REQUEST-CONTROL	Activates priority control
	RESTART-CONTROL	Controls the duration of availability in the event of a restart
	RETRIEVAL-CONTROL	Influences the search strategy of the DBH
	TRANSACTION-SECURITY	Activates transaction management

3.2 Descriptions in alphabetical order

This section contains descriptions of the DBH start statements and options in alphabetical order.

3.2.1 Description format

The descriptions of the different DBH start statements and options are all structured in the same way. There is an entry for each start statement or option and in each case, the name of the statement or option is the entry's heading.

Each entry consists of several parts.

Sections that are not relevant to a specific statement or option are not included. The most important sections are described overleaf.

Name of the start statement or option

Brief description

Definition

This section indicates whether the entry describes a DBH start statement or an option.

In the case of **start statements**, you will see the text “The *n*-th DBH start statement” here. This means that you can immediately identify that this is a description of a start statement and what position this start statement occupies in the input sequence.

In the case of **DBH options**, the text will be either “A higher-level DBH option” or “A lowerlevel DBH option of *higher-level dbh option*”.
This means that you can immediately identify which category the option belongs to and which option (if any) is superordinate to it.

Function

This section provides a detailed description of the statement or option. As well as telling you what the statement or option does, it provides other important information and explanations relating to it.

It is followed by a syntax diagram.

Name of the start statement or option
Operands

Operands

This section contains explanations of the various operands. The operands are dealt with in the order in which they occur in the syntax diagram.

Subheading

Sections containing additional information on the statement or option may follow the operand descriptions.

Special feature of the descriptions of higher-level DBH options

The SDF syntax diagrams in the descriptions of higher-level DBH options differ from those in the other descriptions.

The following example shows the SDF syntax diagram of the higher-level DBH option ADMINISTRATION.

ADMINISTRATION
= *STD / *PARAMETERS(...) *PARAMETERS(...)

```
| ACCOUNTING = *OFF
| ,ADMINISTRATOR = *NONE
| ,MSG-OUTPUT = *STD
| ,SECURITY = *STD
```

Unlike the other syntax diagrams, the line *PARAMETERS(...) with the lines following it is contained here. All the DBH options which are at a lower level than the DBH option ADMINISTRATION are listed as operands here. Only the default operand values are given. The complete description of all permitted operand values is not provided here.

The list of all the associated lower-level DBH options with their default values is intended to facilitate orientation. If you select *PARAMETERS(...) rather than *STD for a higher-level option, you have to take the listed DBH options with all their operands into consideration.

3.2.2 Alphabetical reference section

The syntax of the DBH start statements and options corresponds to the SDF syntax (see the “ Commands” manual).

The DBH start statements and options can be abbreviated in accordance with the SDF conventions.

3.2.2.1 ACCOUNTING

This statement activates accounting.

Definition

A lower-level DBH option of ADMINISTRATION

Function

The DBH option ACCOUNTING activates logging of the accounting of services provided in the different user sessions for BS2000's accounting procedure.

SESAM/SQL generates an accounting record (DBS1 record, see section “Layout of the accounting information”) for each user session.

At the end of the user session, the accounting record is made available in the accounting buffer:

- In the case of SESAM/SQL UTM applications, SESAM/SQL is notified immediately by the openUTM when the conversation is terminated at openUTM. This generates the DBS1 record.
- In the case of TIAM or DCAM CALL DML applications, the conversation is terminated at the DBH by means of the appropriate CALL DML statements and the DBS1 record is generated.
- In the case of TIAM applications the process is terminated at the end of the program run and the DBH is notified. The DBH generates the DBS1 record.

The contents of the accounting buffer are written to the BS2000 accounting file and can be further processed by the RAV accounting procedure.

To make it possible to write to the BS2000 accounting file, the system administrator must ensure that the following requirements are met:

- MAX-ACCOUNT-RECORDS=*NO-LIMITS for the DBH user ID,
- The record type DBS1 must be enabled in BS2000 accounting.

In SESAM/SQL UTM applications, the accounting procedure also supplies the UTM accounting fields (UTMK record).

```
ACCOUNTING

= *OFF / *ON(...)

  *ON(...)

    |  CPU-TIME = *NO / *YES
```

Operands

ACCOUNTING = *OFF

Account logging is inactive initially. However, you can activate it during the DBH session. The SET-ACCOUNTING-PARAMETER administration statement is provided to carry out this function (see "SET-ACCOUNTING-PARAMETER").

ACCOUNTING = *ON(...)
RAV account logging is activated immediately.

CPU-TIME = *NO / *YES

Turns CPU-time counting on/off.

3.2.2.2 ADD-OLD-TABLE-CATALOG-LIST

This statement inserts entries in the CALL DML table catalog.

Definition

Third DBH start statement

Function

You use the DBH start statement ADD-OLD-TABLE-CATALOG-LIST to supply the CALL DML table catalog with entries. You only have to enter this start statement if CALL DML applications are to run during the DBH session.

You enter in the CALL DML table catalog all the CALL DML tables that the DBH can access during the session. You assign each CALL DML table to a database. You also assign the table name used in the CALL DML Open statement to the name of a base table.

The DBH can use the CALL DML table catalog to assign the CALL DML tables addressed in an application to the appropriate databases.

There must be an entry in the SQL database catalog for each database to which a CALL DML table is assigned (see "ADD-SQL-DATABASE-CATALOG-LIST").

You use the DBH option OLD-TABLE-CATALOG to specify the maximum number of entries permitted in the CALL DML table catalog. Its default value is 0, but you can specify a value of up to 254 (see "OLD-TABLE-CATALOG"). If you enter more databases than the maximum number permitted, an error message appears.

You can enter up to 100 CALL DML tables in a single start statement. If you want to enter more than 100 tables, you have to enter the DBH start statement ADD-OLD-TABLE-CATALOG-LIST as many times as is necessary. However, the total number of all entered CALL DML tables must not exceed the limit value specified by the DBH option OLD-TABLE-CATALOG.

You can also use several ADD-OLD-TABLE-CATALOG-LIST statements to enter the required databases in groups.

You conclude parameter input for ADD-OLD-TABLE-CATALOG-LIST with the END statement. This END statement simultaneously concludes input of all DBH start statements.

You can enter fewer CALL DML tables than specified by the DBH option OLD-TABLE-CATALOG. Free entries can be reserved by means of administration. You can use the ADD-OLD-TABLE-CATALOG-ENTRY administration statement (see "ADD-OLD-TABLE-CATALOG-ENTRY") to enter additional CALL DML tables during the session, but these entries apply only to this session.

```
ADD-OLD-TABLE-CATALOG-LIST

ENTRY-1 = *NONE / *CALL-DML-TABLE(...)

   *CALL-DML-TABLE(...)

      |  CALL-DML-TABLE-NAME = <text 1..17>

      |  ,CATALOG-NAME = <filename 1..18_without_all>

      |  ,SCHEMA-NAME = <c-string 1..31_with_lowercase> or <text 1..31>

      |  ,TABLE-NAME = <c-string 1..31_with_lowercase> or <text 1..31>

,ENTRY-2 = *NONE / *CALL-DML-TABLE(...)

.
```

.

,ENTRY-100 = *NONE / *CALL-DML-TABLE(...)

Operands

ENTRY-1 = *NONE

No CALL DML table is entered in the CALL DML table catalog.

ENTRY-1 = *CALL-DML-TABLE(...)
First CALL DML table entered in the CALL DML table catalog

CALL-DML-TABLE-NAME = <text 1..17>
Name of the CALL DML table, as used in the CALL-DML Open statement. You must enter this name in full (17 characters)! It must be unique; i.e. you may enter it only once in a DBH session.

CATALOG-NAME = <filename 1..18_without_all>
Logical name of the database to which the CALL DML table is to be assigned. The logical database name must be entered in the SQL database catalog.

SCHEMA-NAME = <c-string 1..31_with_lowercase> or <text 1..31>
Name of the schema to which the CALL DML table is to be assigned.

TABLE-NAME = <c-string 1..31_with_lowercase> or <text 1..31>
Base table name corresponding to CALL-DML-TABLE-NAME.

ENTRY-2 = *NONE

No further CALL DML table is entered in the CALL DML table catalog at this point.

ENTRY-2 = *CALL-DML-TABLE(...)
Opening line of the second CALL DML table entered in the CALL DML table catalog. The operands of ENTRY-2 are identical to those of ENTRY-1.

You can enter up to 100 CALL DML tables in an ADD-OLD-TABLE-CATALOG-LIST statement. You begin each entry with ENTRY-*x* = ... where the placeholder *x* can have any value from 1 to 99. The placeholder "*x*" must have a different value for each entry. The associated operands in each case are identical to those of ENTRY-1.

ENTRY-100 = *NONE / *CALL-DML-TABLE(...)
Opening line of the 100th CALL DML table entered in the CALL DML table catalog. You cannot enter more than 100 CALL DML tables in one ADD-OLD-TABLE-CATALOG-LIST statement. If you want to include more than 100 CALL DML tables in the session, you have to enter this start statement again.

3.2.2.3 ADD-SQL-DATABASE-CATALOG-LIST

This statement inserts entries in the SQL database catalog.

Definition

Second DBH start statement

Function

You use the DBH start statement ADD-SQL-DATABASE-CATALOG-LIST to supply the SQL database catalog with entries. You enter in the SQL database catalog the databases that the DBH can access during a DBH session. During a session, the DBH can only access databases that are entered in the SQL database catalog.

Each database in the SQL database catalog is identified by its logical name. The logical database name is the name by which an application program addresses a SESAM/SQL database. If there is no SESAM/SQL database with this logical name, an existing database is assigned by means of the physical database name. The physical database name is the name of the physical SESAM/SQL database.

When you make an entry, you assign a BS2000 user ID (DB user ID) to each database, enter the appropriate BS2000 password, assign the physical database name to the logical database name, and regulate access to the database. For each entry, SESAM/SQL inserts the valid status of the database during the current DBH session (see also the section on the SHOW-SQL-DB-CATALOG-ENTRIES administration statement on "SHOW-SQL-DB-CATALOG-ENTRIES").

You use the DBH option SQL-DATABASE-CATALOG (see "SQL-DATABASE-CATALOG") to specify the maximum permitted number of database entries. The default is 1, but you can specify a value of up to 254. If you enter more databases than the maximum number permitted, an error message appears.

You can enter up to 100 databases in a single start statement. If you want to enter more than 100 databases, you have to enter ADD-SQL-DATABASE-CATALOG-LIST as many times as is necessary.

You can also use several ADD-SQL-DATABASE-CATALOG-LIST statements to enter the required databases in groups.

You must always conclude parameter input for the ADD-SQL-DATABASE-CATALOG-LIST statement with the END statement. You can then enter the END statement again. The second END statement concludes input of the DBH start statements. Do not enter the second END statement at this point if you want to use the ADD-OLD-TABLE-CATALOG-LIST start statement to insert entries in the CALL DML table catalog.

You can enter fewer databases than specified by the DBH option SQL-DATABASE-CATALOG. SESAM/SQL reserves unused entries for administration. You can use administration statements to add or delete database entries during the session, but these additions and deletions apply only to this session.

Note that the CREATE CATALOG utility statement requires a free entry in the SQL database catalog.

```
ADD-SQL-DATABASE-CATALOG-LIST

ENTRY-1 = *NONE / *CATALOG(...)

  *CATALOG(...)

    |  CATALOG-NAME = <filename 1..18_without_all>

    |  ,USER-ID = *DBH-USER-ID / <name 1..8>

    |  ,PASSWORD = *NONE / <c-string1..4> / <x-string 1..8> / <integer -2147483648..+2147483647>
```

```
      |  ,PHYSICAL-NAME = *STD / <filename 1..18_without_all>
      |  ,COPY-NUMBER=*NONE / <integer 1..999999>
      |  ,ACCESS = *ALL / *PARAMETERS(...)
      |        *PARAMETERS(...)
      |              |     READ = *YES / *NO
      |              |     WRITE = *YES / *NO
      |              |     CAT-ADMINISTRATION = *YES / *NO
      |              |     REPLICATION = *YES / *NO
,ENTRY-2 = *NONE / *CATALOG(...)
   .
   .
,ENTRY-100 = *NONE / *CATALOG(...)
```

Operands

ENTRY-1 = *NONE

No database is entered in the SQL database catalog at this point.

ENTRY-1 = *CATALOG(...)
First database entered in the SQL database catalog. You must at least enter the CATALOG-NAME operand, otherwise the entry in the SQL database catalog is invalid.

CATALOG-NAME = <filename 1..18_without_all>

Logical name of the database to be entered. The logical database name is the name by which a database is addressed in the application program. You must enter the logical database name, otherwise the entry in the SQL database catalog is invalid. The logical database name must be unique; you cannot assign this name more than once.

USER-ID = *DBH-USER-ID / <name 1..8>
DB user ID of the database.
The default value of USER-ID corresponds to the DBH user ID.

PASSWORD = *NONE / <c-string1..4> / <x-string 1..8> /
<integer -2147483648..+2147483647>
BS2000 password that protects the files in the database from unauthorized access. If the files in the database are protected by a BS2000 password, you must enter it here. The default, *NONE, means that the files in the database are not password protected.

PHYSICAL-NAME =
Physical name of the database
The physical database name assigns the logical database name used by the application program to the name of a physical database. The physical database name must be unique; you cannot assign this name more than once.

PHYSICAL-NAME = *STD
The physical database name corresponds to the logical database name you assigned under CATALOG-NAME.

PHYSICAL-NAME = <filename 1..18_without_all>
You must enter the physical database name when it differs from the logical database name.
The physical database name is derived from the BS2000 file name of a database's catalog space. The . CATALOG suffix is omitted. The BS2000 file name of a database's catalog space is: :*catid*:*user-id*.*name***.** CATALOG

COPY-NUMBER =
Version number of the SESAM backup copy of the specified database. The SESAM backup copy of the catalog space must be complete and must be created with the COPY CATALOG utility statement.

COPY-NUMBER = *NONE
The default value *NONE means that you only work with the original database.

COPY-NUMBER = <1..999999>
Version number of a SESAM backup copy of the specified catalog space. If you specify COPY-NUMBER, read access to the database is entered automatically (see ACCESS = *PARAMETERS(...)).

ACCESS =

Specifies which access functions can be used on the database.

ACCESS = *ALL
Allows unrestricted access to the database. Permits read and write access to user data and metadata. The operand value *ALL only refers to the original database.

ACCESS = *PARAMETERS(...)
Restricts access to the database.
The parameters are arranged in a hierarchical structure. If CAT-ADMINISTRATION is permitted, so, implicitly, are WRITE and READ In the same way, WRITE implicitly permits READ.
At least one parameter must be specified with *YES.

If you specified COPY-NUMBER, the value READ = *YES is set automatically. An implicit or explicit ACCESS = *ALL specification is then ignored.
WRITE=*NO and CAT-ADMINISTRATION=*NO must be specified explicitly when ACCESS=*PARAMETERS() is used to specify the access privileges in detail, and READ=*YES and REPLICATION=*NO must also be set (implicitly or explicitly), otherwise an error message will result.
You are not permitted to specify the parameters WRITE and CAT-ADMINISTRATION for backup copies.

READ = *YES
Permits read access to user data and metadata. The data can be queried but not changed.

> When READ=*YES (and also WRITE=*NO and CAT-ADMINISTRATION=*NO) is specified for a database, accesses to this database are accelerated by means of simplified transaction locks.

READ = *NO
Prevents read and write access to user data and metadata.
You cannot select READ = *NO if you have set *YES (the default) for either CAT-ADMINISTRATION or WRITE.

WRITE = *YES

Permits read and write access to user data. Metadata can be read but not changed.

WRITE = *NO
Prevents write access to user data and metadata.
You cannot select WRITE = *NO if you have set *YES (the default) for CAT-ADMINISTRATION.

CAT-ADMINISTRATION = *YES
Permits read and write access to both user data and metadata.

CAT-ADMINISTRATION = *NO
Prevents write access to the metadata in the catalog space of the database. It therefore prevents the use of the following statements:

- SQL statements for schema definition and management and memory management
- Utility statements

REPLICATION = *YES
Access to a replication of the specified database. You cannot access a normal catalog with this database entry. Only read access and the REFRESH
REPLICATION utility statement are permitted for a replication.

REPLICATION = *NO
No access to a replication.

ENTRY-2 = *NONE / *CATALOG(...)
Opening line of the second database entered in the SQL database catalog. The operands of ENTRY-2 are identical to those of ENTRY-1.
You can enter up to 100 databases in an ADD-SQL-DATABASE-CATALOG-LIST statement. You begin each entry with ENTRY-*x* = ... where the placeholder *x* can have any value from 1 to 99, The placeholder "*x*" must have a different value for each entry. The associated operands in each case are identical to those of ENTRY-1.

ENTRY-100 = *NONE / *CATALOG(...)
Opening line of the 100th database entered in the SQL database catalog.You cannot enter more than 100 databases in one ADD-SQL-DATABASE-CATALOG-LIST statement. If you want to include more than 100 databases in the DBH session, you have to enter this start statement again.

3.2.2.4 ADMINISTRATION

This statement administers the DBH.

Definition

Higher-level DBH option

Function

ADMINISTRATION is the higher-level option for all DBH options that are important for SESAM/SQL system administration.

```
ADMINISTRATION

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | ACCOUNTING = *OFF

    | ,ADMINISTRATOR = *NONE

    | ,MSG-OUTPUT = *STD

    | ,SECURITY = *STD
```

Operands

ADMINISTRATION = *STD

Default values apply to all the operands of ADMINISTRATION. These are described under the relevant lower-level DBH options.

ADMINISTRATION = *PARAMETERS(...)
The various operands of ADMINISTRATION are DBH options. Their operands are described in each case under the relevant DBH option.

3.2.2.5 ADMINISTRATOR

Assigns administrator authorization

Definition

A lower-level DBH option of ADMINISTRATION

Function

You use the DBH option ADMINISTRATOR to define a password that prevents the unauthorized issue of administration statements via the CALL interface and the SESADM administration program. You can also specify that only one user is authorized to issue administration statements, namely the system administrator. To do this, enter the system user identification of the system administrator.

You must also specify the system administrator's system user identification if certain utility statements are to be issued during the DBH session. The system user identification entered here is required at the authorization check for the CREATE CATALOG, CREATE REPLICATION, RECOVER CATALOG, RECOVER CATALOG_SPACE, REFRESH REPLICATION and UNLOAD ... FROM COPY_FILE utility statements (see the “ SQL Reference Manual Part 2: Utilities”).

During the DBH session, the system administrator can modify the entries for ADMINISTRATOR to suit requirements. This is done using the MODIFY-ADMINISTRATION administration statement (see "MODIFY-ADMINISTRATION").

```
ADMINISTRATOR

= *NONE / ANY(...) / *TIME-SHARING-USER(...) / *APPLICATION-USER(...)
    *ANY(...)
        |   PASSWORD = <c-string 3..3> / <x-string 5..6>
    *TIME-SHARING-USER(...)
        |   PASSWORD = <c-string 3..3> / <x-string 5..6>
        |   ,HOST-NAME = <name 1..8>
        |   ,USER-ID = <name 1..8>
    *APPLICATION-USER(...)
        |   PASSWORD = <c-string 3..3> / <x-string 5..6>
        |   ,HOST-NAME = <text 1..8>
        |   ,APPLICATION-NAME = <text 1..8>
        |   ,CUSTOMER-NAME = <text 1..8>
```

Operands

ADMINISTRATOR = *NONE

Administration using the CALL interface and SESADM is not possible. The MODIFY-ADMINISTRATION administration statement is also rejected.

ADMINISTRATOR = *ANY(...)
Users who are not system administrators can also issue administration statements using the CALL interface and SESADM.

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

ADMINISTRATOR = TIME-SHARING-USER(...)
Only the system administrator can issue administration statements using the CALL interface and SESADM. The system administrator is a timesharing user and is identified by the system user identification, i.e. host name and BS2000 user ID.

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

HOST-NAME = <name 1..8>
Name of the computer from which administration is to be carried out

USER-ID = <name 1..8>
BS2000 user ID of the system administrator

ADMINISTRATOR = APPLICATION-USER(...)
Only the system administrator can issue administration statements using the CALL interface. The system administrator is an application user and is identified by the system user identification, i.e. the host name, the name of the UTM application and the name of the user (customer name).

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

HOST-NAME = <text 1..8>
Name of the computer from which administration is to be carried out

APPLICATION-NAME = <text 1..8>
Name of the application from which administration is to be carried out

CUSTOMER-NAME = <text 1..8>
Name of the user.
If the system administrator is working under openUTM, the KDCSIGN name must be specified here. If the system administrator is working under DCAM, the name used by the system administrator at the programming interface must be specified.

3.2.2.6 CODED-CHARACTER-SET

This statement specifies the CCS name of the user program.

Definition

A lower-level DBH option of LINKED-IN-ATTRIBUTES

Function

The DBH option CODED-CHARACTER-SET allows you to specify a coded character set which is used by the user program of a linked-in DBH. The DBH option is ignored in the independent DBH.
In the independent DBH the user program's CCS name is not specified using a DBH option but via the user's configuration file, see the " Core manual".

If a coded character set was specified for the database (CODE-TABLE *ccs_name* clause in the CREATE CATALOG or ALTER CATALOG statement), the CCS name of the database must match the CCS name of the user program when the user program accesses the database. Accesses by user programs using CODED-CHARACTER-SET=*NONE or by user programs from SESAM/SQL < V5.0 are rejected with SQLSTATE.

This check does not take place for the utility statements CREATE/ALTER CATALOG, CREATE/REFRESH REPLICATION and RECOVER CATALOG [SPACE]. If no coded character set was specified for the database, the check does not take place either.

CODED-CHARACTER-SET
= *NONE / <alphanum_name 1..8>

Operands

CODED-CHARACTER-SET = *NONE / <alphanum_name 1..8>

CCS name of the user program as defined in BS2000 (system component XHCS). The default value is *NONE, i.e. no coded character set has been defined for the user program.

3.2.2.7 COLUMNS

This statement enlarges the area for retrieval statements.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

The DBH option COLUMNS allows you to enlarge the area for retrieval statements.

The area for retrieval statements is normally limited to 256 columns. In the case of large retrieval statements that address more than 256 columns, you must increase the size of this area up to 1024 columns.

You can adjust the value of the option during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

COLUMNS
= 256 / <integer 256..1024>

Operands

COLUMNS = 256 / <integer 256..1024>

Area size in columns for retrieval statements.

Calculating the length of a retrieval statement

You base your entry for the COLUMNS option on the length of the longest retrieval statement in an application. This is determined by the number of columns (attributes in the case of CALL DML) or the number of conditions attached to columns. In the case of SQL applications, the number of operators in arithmetic expressions is also included in the calculation.

To calculate the length of a retrieval statement, proceed as follows:

- Count each column whenever it is addressed in the retrieval statement.
- In the case of SQL applications, count also the operators in arithmetic expressions.
- In the case of multiple columns, add the number of occurrences addressed in the retrieval statement.
- If one or more conditions are attached to one and the same column, add the number of conditions for this column.

In the case of CALL DML, the value calculated corresponds to the area specified in the COLUMNS option. In the case of SQL DML, the calculated length of the longest retrieval statement represents a maximum value for the area specified in the COLUMNS option. As a result of internal optimization, the optimal value can be lower than the calculated value.

Space required to save the temporary work file

The space required to store internal statement formats in the temporary work file depends on the COLUMNS option. The space required can be seen in the following list (see also the description of WORK-CONTAINER on "WORK-CONTAINER").

COLUMNS =	Space required for storing average internal statement formats (Kbytes)
up to 260	28

up to 325	32
up to 366	36
up to 408	40
up to 449	44
up to 512	48
up to 553	52
up to 594	56
up to 636	60
up to 1024	64

Table 9: Space required for storing internal statement formats

3.2.2.8 CONFIGURATION-NAME

This statement assigns the DBH to a configuration.

Definition

A lower-level DBH option of DBH-IDENTIFICATION

Function

You use the DBH option CONFIGURATION-NAME to specify the configuration name of the DBH.

By specifying a configuration name, you assign the DBH to a configuration. Each configuration has a unique name on its host system.

Like the DBH name (see the DBH option DBH-NAME on "DBH-NAME"), the configuration name identifies the following DBH-specific files: the cursor files and the files for transaction management. When several SESAM/SQL DBHs are loaded, these files must be uniquely identified by a configuration name and a DBH name in each case.

If the configuration and/or DBH name have the default value *BLANK, the character "'BLANK'" in the file names is replaced by the character "@". For example, if you specify CONFIGURATION-NAME = 7 and DHB-NAME = *BLANK, the file name of the WA-LOG file of the independent DBH is:

```
SESAM7@.WA-LOG
```

If you specify CONFIGURATION-NAME = 7 and DHB-NAME = 8, the file name of the WA-LOG file is:

```
SESAM78.WA-LOG
```

CONFIGURATION-NAME
= *BLANK / <alphanum-name 1..1>

Operands

CONFIGURATION-NAME = *BLANK

The blank is used as the configuration name.

CONFIGURATION-NAME = <alphanum-name 1..1>
Configuration name assigned by the system administrator. The characters #, @ and $ are not permitted as configuration name.

3.2.2.9 CPU-RESOURCES

This statement controls CPU utilization.

Definition

Higher-level DBH option

Function

CPU-RESOURCES is the higher-level option of DBH options with which you can control the use of CPU resources.

```
CPU-RESOURCES
= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | DBH-TASKS = 1
    | ,SERVICE-TASKS = *STD
```

Operands

CPU-RESOURCES = *STD

Default values apply to all the operands of CPU-RESOURCES. These are described in the description of the lower-level DBH option SERVICE-TASKS.

CPU-RESOURCES = *PARAMETERS(...)
CPU-RESOURCES has the DBH options DBH-TASKS and SERVICE-TASKS as its operands.

3.2.2.10 CURSOR-BUFFER

This statement sets the size of the cursor buffer.

Definition

A lower-level DBH option of STORAGE-SIZE

Function

Cursor files are DBH-specific work files that are available for processing retrieval statements. The DBH stores intermediate results of retrieval statements in them. Access to internal cursor files is via the cursor buffer, provided it is large enough.

You can use the DBH option CURSOR-BUFFER to set the size of the cursor buffer so as to minimize the number of accesses to internal cursor files. You specify the size of the cursor buffer in units of 1 Kbyte.

You can adjust the values of the operands during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

```
CURSOR-BUFFER

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | BUFFER-SIZE = *STD / <integer 72..1500000>

    | ,FRAME-SIZE = 4 / <integer 4..32>
```

Operands

CURSOR-BUFFER = *STD

The default values for the size of the cursor buffer are described under *PARAMETERS(...).

CURSOR-BUFFER = *PARAMETERS(...)
Modifies the cursor buffer settings.

BUFFER-SIZE = *STD / <integer 72..1500000>
Sets the size of the cursor buffer in units of 1 Kbyte.
The default value of CURSOR-BUFFER depends on the maximum permitted number of concurrent active threads that you specify using the DBH option THREADS and the number of system threads required. A minimum of 32 Kbytes is required per thread. If you specify a value for BUFFER-SIZE that is smaller than the product of the minimum size per thread and the number of threads (THREADS option plus the number of system threads), then it is rounded up to the minimum size.

FRAME-SIZE = 4 / <integer 4..32>

Sets the size of the cursor buffer frame in units of 1 Kbyte. The default value is 4. Other valid specifications in the range of <integer 4..32> are always rounded off to a number that is a multiple of 4.

> The minimum size of CURSOR-BUFFER per thread is dependent on the value of FRAME-SIZE and must in all cases be eight times the value of FRAME-SIZE. If more is specified by the user, the user specifications apply. However, these are always rounded up to a multiple of the FRAME-SIZE value.

Information on cursor files

The operational statistics of the SESAM/SQL utility SESMON provide useful information on cursor files.
The “SYSTEM INFORMATION” form displays the number of cursor files that are currently open (see "DBH form SYSTEM INFORMATION").
The cursor buffer should be large enough to keep to a minimum the number of times internal cursor files are accessed. The “I/O” form (see "DBH form I/O") helps you set the optimum size for the cursor buffer. It displays the number of logical and physical accesses of the cursor files.

The “ Performance” manual describes the cases in which it is advisable to modify the size of the cursor buffer the next time the DBH starts up.

3.2.2.11 DBH-IDENTIFICATION

This statement names a DBH and assigns it to a configuration.

Definition

Higher-level DBH option

Function

DBH-IDENTIFICATION is the higher-level DBH option for DBH options that allow you to identify the DBH you want to start rather than any other DBH.

```
DBH-IDENTIFICATION

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | CONFIGURATION-NAME = *BLANK

    | ,DBH-NAME = *BLANK
```

Operands

DBH-IDENTIFICATION = *STD

Default values apply to all DBH-IDENTIFICATION operands. These values are described under the relevant lower-level options.

DBH-IDENTIFICATION = *PARAMETERS(...)
The DBH-IDENTIFICATION operands are DBH options. Their operands are described in each case under the relevant DBH option.

3.2.2.12 DBH-NAME

This statement defines the DBH name.

Definition

A lower-level DBH option of DBH-IDENTIFICATION

Function

You use the DBH option DBH-NAME to define the name of the DBH.

The DBH name indicates that this DBH is to be started rather than any of the other DBHs belonging to the same configuration.
There cannot be two DBHs with the same DBH name in a single configuration. You must remember this when you load a number of SESAM/SQL DBHs (independent DBHs and/or linked-in DBHs). It is always advisable to assign a DBH name other than the default, even if the DBH is initially to be loaded only once. This prevents collisions when, for test purposes, for example, another DBH is loaded.
Like the configuration name, the DBH name identifies the cursor files and the files for transaction management. When several SESAM/SQL DBHs are loaded, these files must be uniquely identified by a configuration name and a DBH name in each case (see the section on the DBH option CONFIGURATION-NAME on "CONFIGURATION-NAME").

If the configuration and/or DBH name has the default value *BLANK, then the character "'BLANK'" is replaced by the "@" character in the file name (see the DBH option CONFIGURATION-NAME).

DBH-NAME
= *BLANK / <alphanum-name 1..1>

Operands

DBH-NAME = *BLANK

The blank is used as the DBH name.

DBH-NAME = <alphanum-name 1..1>
DBH name assigned by the system administrator. The characters #, @ and $ are not permitted as DBH name.

3.2.2.13 DBH-TASKS

This statement defines DBH tasks.

Definition

A lower-level DBH option of CPU-RESOURCES

Function

You use the DBH option DBH-TASKS to specify the number of DBH tasks.

After the first DBH task, additional DBH tasks are started via ENTER. All DBH tasks are assigned the same job class.

In a linked-in DBH the option DBH-TASKS is ignored because this kind of DBH generally only consists of one TASK.

If a value greater than 1 was specified and SESAM/SQL is only licensed for one task, an informational message is output stating that the license is only valid for one task and the DBH can therefore only be started with one task.

You can adjust the value of the option with the administration statement RELOAD-DBH-SESSION (see "RELOAD-DBH-SESSION") when the DBH modules are reloaded.

DBH-TASKS
= 1 / <integer 1..16>

Operands

DBH-TASKS = 1 / <integer 1..16>

Specifies the number of DBH tasks in a session. The default is 1.

3.2.2.14 FILE-RESOURCES

This statement specifies settings for files.

Definition

Higher-level DBH option

Function

FILE-RESOURCES is the higher-level option of all DBH options with which you specify settings for session-specific files.

```
FILE-RESOURCES
= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | ,SESSION-LOGGING-ID = *STD
    | ,MEDIA-CATALOG = *STD
```

Operands

FILE-RESOURCES = *STD

Default values apply to all FILE-RESOURCES operands. These values are described under the relevant lower-level DBH options.

FILE-RESOURCES = *PARAMETERS(...)
The FILE-RESOURCES operands are DBH options. Their operands are described in each case under the relevant DBH option.

3.2.2.15 LINKED-IN-ATTRIBUTES

This statement specifies settings for the linked-in DBH.

Definition

Higher-level DBH option

Function

LINKED-IN-ATTRIBUTES is the higher-level option for all DBH options with which you specify settings for the linked-in DBH. These settings are only taken into account by the linked-in DBH. They are ignored in the independent DBH.

```
LINKED-IN-ATTRIBUTES

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    |  ,CODED-CHARACTER-SET = *NONE / <alphanum-name 1..8>
```

Operands

LINKED-IN-ATTRIBUTES = *STD

Default values apply to all LINKED-IN-ATTRIBUTES operands which are described under the relevant lower-level DBH options.

LINKED-IN-ATTRIBUTES = *PARAMETERS(...)
The LINKED-IN-ATTRIBUTES operands are DBH options. Their operands are described in each case under the relevant DBH option.

3.2.2.16 MEDIA-CATALOG

This statement creates a media catalog.

Definition

A lower-level DBH option of FILE-RESOURCES

Function

The media catalog contains storage information for certain DBH-specific files: the transaction log files (TA-LOG files), the restart log file (WA-LOG), the temporary work files and the cursor files. The media catalog defines the volumes on which these files are to be stored. The media catalog does not exist in the form of a file. It is kept in the DBH's memory and is only valid for one DBH session.

The DBH option MEDIA-CATALOG allows you to specify the storage information in the media catalog.

The DBH does not store this information on the files of the media catalog in the transaction log file. The information is therefore not failproof. In the event of an external restart, you can, if required, specify other volumes for media-catalog files.

There are 2 transaction log files (TA-LOG files). The operand TALOG-SUPPORT applies to both of these files.

> ¡ These specifications only have an effect if the corresponding file does not yet exist.

```
MEDIA-CATALOG
= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | TALOG-SUPPORT = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |   *PUBLIC-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
    |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
    |   *PRIVATE-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
    |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
    |     | ,DEVICE-TYPE = <structured-name 1..8>
    |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
    | ,WALOG-SUPPORT = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |   *PUBLIC-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
    |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
    |   *PRIVATE-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
```

```
|     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
|     | ,DEVICE-TYPE = <structured-name 1..8>
|     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
| ,CURSOR-MEDIA = *STD / *PARAMETERS(...)
|   *PARAMETERS(...)
|     | SUPPORT-1 = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
|     |   *PUBLIC-DISK(...)
|     |     | CAT-ID = *STD / <cat-id>
|     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
|     |   *PRIVATE-DISK(...)
|     |     | CAT-ID = *STD / <cat-id>
|     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
|     |     | ,DEVICE-TYPE = <structured-name 1..8>
|     |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>

|     | , SUPPORT-5 = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
|     |   *PUBLIC-DISK(...)
|     |     | CAT-ID = *STD / <cat-id>
|     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
|     |   *PRIVATE-DISK(...)
|     |     | CAT-ID = *STD / <cat-id>
|     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *STD / <integer 0..32767>
|     |     | ,DEVICE-TYPE = <structured-name 1..8>
|     |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
```

Operands

MEDIA-CATALOG = *STD

All the files in the media catalog are created on a public disk.

MEDIA-CATALOG = *PARAMETERS(...)

TALOG-SUPPORT =
Defines the volume for the TA-LOG files.
There are a total of two TA-LOG files (TA-LOG1, TA-LOG2) that are written to one after the other in rotation.

TALOG-SUPPORT = *PUBLIC-DISK(...)
Causes the TA-LOG files to be created on a public disk.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space.
The specified number of PAM pages is reserved immediately. It should correspond to the expected file size.
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY-ALLOCATION = *STD / <integer 0..32767>
Number of PAM pages for subsequent extensions to files
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

TALOG-SUPPORT = *PRIVATE-DISK(...)

The TA-LOG files are to be created on a private disk. To do this, you must make an entry for DEVICE-TYPE.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space.
The specified number of PAM pages is reserved immediately. It should correspond to the expected file size.
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY ALLOCATION = *STD / <integer 0..32767>
Number of PAM pages for subsequent extensions to files
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

DEVICE-TYPE = <structured-name 1..8>
BS2000 device type to which the required disks are assigned.
If you make an incorrect entry, the DBH aborts the start procedure.

VOLUME = list-poss(6): <alphanum-name 1..6>
VSNs of the disks on which the files are to be created. Up to 6 VSNs are permitted, each of which can be specified only once. VOLUME is not evaluated unless the DEVICE-TYPE operand is specified.

WALOG-SUPPORT =
Defines the volume for the WA-LOG file.
The operands are identical to those in the operand descriptions for TALOG-SUPPORT.

CURSOR-MEDIA =
Defines the volume for cursor files.
Files that have been created are reused when they are unlocked. This applies both to internal cursor files and CALL DML-specific cursor files.

CURSOR-MEDIA = *STD
The DBH creates the cursor files on a public disk.

CURSOR-MEDIA = *PARAMETERS(...)

SUPPORT-*n* =

n = 1...5

You can define the volumes for the two cursor files SESAM *cn*.CURSOR.0001 and SESAM *cn*.CURSOR.0002 (or for the linked-in DBHs SESLK *cn*.CURSOR.0001 and SESLK *cn*.CURSOR.0002) yourself, up to five different specifications being permitted (SUPPORT-1... SUPPORT-5). SESAM/SQL creates a maximum of two cursor files. The values entered for SUPPORT-1 determine which medium the first required cursor file with the suffix “0001” or “0002” will be created on. The values entered for SUPPORT-2 determine which medium the second required cursor file with the suffix “0001” or “0002” will be created on. The values entered for SUPPORT-3 to SUPPORT-5 are evaluated only if the media specified for SUPPORT-1 or SUPPORT-2 are not available or if there is insufficient space. The media are used in the specified order. The operands are identical with those in the operand description for TA-LOG-SUPPORT. After the entry with number 5 has been used, the cycle begins again with 1 if 1 has not yet been attempted for this file.

3.2.2.17 MSG-OUTPUT

Controls the output of the DBH

Definition

A lower-level DBH option of ADMINISTRATION

Function

You can use the DBH option MSG-OUTPUT to request SESAM/SQL DBH outputs to SYSOUT or the console of the BS2000 system administrator or to SYSLST. Specific output can be suppressed.
You can completely suppress OPEN and CLOSE messages for logical files. You can also enter specifications for transferring information by email.

When using the linked-in DBH, it can be a good idea to request that DBH outputs only be written to SYSOUT so that outputs of the application program to SYSLST are not interspersed with DBH outputs.

Consistency checks and the end-of-program message are always output to the console and cannot be suppressed.

You can adjust the values of the operands during the DBH session by means of the MODIFY-MSG-OUTPUT administration statement (see "MODIFY-MSG-OUTPUT").

```
MSG-OUTPUT

= *STD / *ALL / *PARAMETERS(...)

  *PARAMETERS(...)

    | MSG = *ALL / list-poss(2): *SYSOUT / *SYSLST

    | ,OPEN-CLOSE-MSG = *NONE / *ALL / list-poss(2): *SYSOUT / *SYSLST

    | ,MAIL = *NONE / *PARAMETERS(...)

    |       *PARAMETERS(...)

    |             |     COLLECT = *NO / *YES

    |             |     ,IMMEDIATE = *NO / *YES
```

Operands

MSG-OUTPUT = *STD

The DBH output, except for OPEN and CLOSE messages, is sent to SYSOUT or the console and SYSLST. OPEN and CLOSE messages are completely suppressed. This default setting always applies during the initialization of the DBH or until completion of an external restart. DBH output cannot be redirected during this time. This corresponds to specifying MSG-OUTPUT=*PARAMETERS(MSG=*ALL,OPEN-CLOSE-MSG=*NONE).

MSG-OUTPUT = *ALL
The DBH output, except for OPEN and CLOSE messages, is sent to SYSOUT or the console and SYSLST.
This corresponds to specifying MSG-OUTPUT=*PARAMETERS(MSG=*ALL, OPEN-CLOSE-MSG=*ALL).

MSG-OUTPUT = *PARAMETERS(...)

MSG =
Controls the output of messages.

MSG = *ALL
The DBH output, except for OPEN and CLOSE messages, is sent to SYSOUT or the console and SYSLST.

MSG = list-poss(2): *SYSOUT / *SYSLST
The DBH output, except for OPEN and CLOSE messages, is sent to SYSOUT or the console or SYSLST. Output sent to an unspecified destination is suppressed.

OPEN-CLOSE-MSG =
The output of OPEN and CLOSE messages from SES1001, SES1011, SES1201, SES2015 and SES2023 can be suppressed within the framework defined by the MSG= operand.
OPEN and CLOSE messages for databases are not affected by this.

OPEN-CLOSE-MSG = *NONE
This is the default. Suppresses OPEN and CLOSE messages completely.

OPEN-CLOSE-MSG = *ALL
The DBH outputs the OPEN and CLOSE messages as specified by the MSG= operand.

OPEN-CLOSE-MSG = list-poss(2): *SYSOUT / *SYSLST
The DBH outputs OPEN and CLOSE messages only to SYSOUT or SYSLST, provided the operand MSG= permits it.

MAIL =

Controls the output of information by email.

MAIL = *NONE
The information is not sent by email.

MAIL = *PARAMETERS(...)
Controls the volume of information which is sent by email.

COLLECT = *NO
Outputs to SYSLST are not sent by email.

COLLECT = *YES
All outputs to SYSLST are also stored in a temporary file. The temporary file is updated until an administration command MODIFY-MSG-OUTPUT is issued or to the end of the DBH session. Then the temporary file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email"). The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST DBH” is entered under “Subject:”.
The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The temporary file is attached to the email.

IMMEDIATE = *NO
Messages are not sent by email.

IMMEDIATE = *YES
All messages with message numbers which are entered in the MAIL parameter file (see "Sending important information of the DBH session by email") are immediately sent by email to the address which is entered in the MAIL parameter file. The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

The message number is entered under “Subject:”.
The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The text of the email contains the message.

> When running in the batch mode, the DBH output is sent to the console when *SYSOUT is specified.

Example

If OPEN and CLOSE messages are only to be output to *SYSLST, the DBH option could contain the following values:

```
MSG-OUTPUT=*PARAMETERS(MSG=*ALL,OPEN-CLOSE-MSG=*SYSLST)
```

Within the framework defined by MSG=*ALL, it is possible to restrict the output of OPEN and CLOSE messages to *SYSLST.

Output of OPEN and CLOSE messages to *SYSLST is not possible if the DBH option contains the following values:

```
MSG-OUTPUT=*PARAMETERS(MSG=*SYSOUT,OPEN-CLOSE-MSG=*SYSLST)
```

In this case, MSG=*SYSOUT defines a framework that does not permit output to SYSLST.

3.2.2.18 OLD-TABLE-CATALOG

This statement limits the number of entries in the CALL DML table catalog.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

You use the DBH option OLD-TABLE-CATALOG to specify the maximum number of entries permitted in the CALL DML table catalog.

You edit the CALL DML table catalog by means of the DBH start statement ADD-OLD-TABLE-CATALOG-LIST (see "ADD-OLD-TABLE-CATALOG-LIST"). You can enter as many CALL DML tables in the catalog as specified by the DBH option OLD-TABLE-CATALOG.

You can also enter fewer tables. During the DBH session you can use an administration statement to enter more CALL DML tables, as required, up to the maximum number of permitted entries.

You can adjust the values of the option during the DBH session by means of the MODIFY-OLD-TABLE-CATALOG-LIMIT administration statement (see "MODIFY-OLD-TABLE-CATALOG-LIMIT").

OLD-TABLE-CATALOG
= 0 / <integer 0..254>

Operands

OLD-TABLE-CATALOG = 0 / <integer 0..254>

Maximum number of table entries in the CALL DML table catalog. The default is 0, this means that CALL DML tables cannot be edited during the DBH session. However, you can enter a different value and permit up to 254 entries for CALL DML tables.

3.2.2.19 RECOVER-OPTIONS

This statement sets the options for RECOVER and REFRESH runs.

Scope of validity

A higher-level DBH option (the particular occurrences of the lower-level options are also described in this section)

Function

You use the DBH option RECOVER-OPTIONS to set the following options which are used in the case of a RECOVER or REFRESH run for the DBH in the service task:

- The size of the buffer for system-access data
- The size of the buffer for user data
- The storage information for the transaction log files (TA-LOG files)
- The storage information for the restart log file (WA-LOG file)

You can adjust the values of the operands during the DBH session by means of the MODIFY-RECOVER-OPTIONS administration statement (see "MODIFY-RECOVER-OPTIONS").

```
RECOVER-OPTIONS

= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | SYSTEM-DATA-BUFFER = *STD / <integer 80..1000000>
    | ,USER-DATA-BUFFER = *STD / <integer 80..1000000>
    | ,MEDIA-CATALOG = *STD / *PARAMETERS(...)
    |   *PARAMETERS(...)
    |     | TALOG-SUPPORT = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |     |   *PUBLIC-DISK(...)
    |     |     | CAT-ID = *STD / <cat-id>
    |     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     |     | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
    |     |   *PRIVATE-DISK(...)
    |     |     | CAT-ID = *STD / <cat-id>
    |     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     |     | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
    |     |     | ,DEVICE-TYPE = <structured-name 1..8>
    |     |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
    |     | ,WALOG-SUPPORT = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |     |   *PUBLIC-DISK(...)
    |     |     | CAT-ID = *STD / <cat-id>
    |     |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     |     | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
```

```
|    |   *PRIVATE-DISK(...)
|    |     | CAT-ID = *STD / <cat-id>
|    |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|    |     | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
|    |     | ,DEVICE-TYPE = <structured-name 1..8>
|    |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
```

Operands

RECOVER-OPTIONS = *STD

The default values are used for all options (see the description of the operands below).

RECOVER-OPTIONS = *PARAMETERS(...)

Changes the settings for the options concerned.

SYSTEM-DATA-BUFFER = *STD / <integer 80..1000000>

Size of the buffer for system-access data in a RECOVER or REFRESH run in units of 1 Kbyte. The DBH calculates the default value internally according to the value of the session to be recovered.

USER-DATA-BUFFER = *STD / <integer 80..1000000>

Size of the buffer for user data in a RECOVER or REFRESH run in units of 1 Kbyte. The DBH calculates the default value internally according to the value of the session to be recovered.

MEDIA-CATALOG = *STD

The TA-LOG and WA-LOG files are created with default values on a public disk.

MEDIA-CATALOG = *PARAMETERS(...)

Changes the settings for the TA-LOG and WA-LOG files.

TALOG-SUPPORT =

Defines the volume for the TA-LOG files. There are a total of two TA-LOG files (TA-LOG1, TA-LOG2) that are written to one after the other in rotation.

TALOG-SUPPORT = *PUBLIC-DISK(...)

Causes the TA-LOG files to be created on a public disk.

CAT-ID = *STD / <cat-id>

Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>

Number of PAM pages for the initial allocation of memory space. The specified number of PAM pages is reserved immediately. It should correspond to the expected file size. The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY-ALLOCATION = *STD / <integer 1..32767>

Number of PAM pages for subsequent extensions to files. The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

TALOG-SUPPORT = *PRIVATE-DISK(...)

The TA-LOG files are to be created on a private disk. To do this, you must make an entry for DEVICE-TYPE.

CAT-ID = *STD / <cat-id>

Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>

Number of PAM pages for the initial allocation of memory space. The specified number of PAM pages is reserved immediately. It should correspond to the expected file size. The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY ALLOCATION = *STD / <integer 1..32767>

Number of PAM pages for subsequent extensions to files. The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

DEVICE-TYPE = <structured-name 1..8>

BS2000 device type to which the required disks are assigned. If you make an incorrect entry, the DBH aborts the start procedure.

VOLUME = list-poss(6): <alphanum-name 1..6>

VSNs of the disks on which the files are to be created. Up to 6 VSNs are permitted, each of which can be specified only once. VOLUME is not evaluated unless the DEVICE-TYPE operand is specified.

WALOG-SUPPORT =

Defines the volume for the WA-LOG file. The operands are identical to those in the operand descriptions for TALOG-SUPPORT.

3.2.2.20 REQUEST-CONTROL

This statement activates priority control for the independent DBH.

Definition

A lower-level DBH option of SYSTEM-STRATEGIES

Function

You use the DBH option REQUEST-CONTROL to specify whether the independent DBH is to work with priority control. You can also modify the default values for priority control.

Priority control is not effective with the linked-in DBH.

During a DBH session, requests from tasks with different BS2000 priorities usually have to be processed concurrently. The DBH processes the individual requests in accordance with their BS2000 priority in the order in which they arrive. The priority control of the independent DBH allows you to qualify the BS2000 priority for requests and modify it to suit the requirements of the DBH session. To do this, you subdivide the whole range of BS2000 priorities into three priority classes: Queue-1, Queue-2 and Queue-3. By assigning a weight to each priority class, you can influence the speed at which the requests of the different classes are processed.

The assignment of a request to the DBH priority classes is based on the request's BS2000 priority.

The priority classes are defined as follows:

Queue-1: 30 <= BS2000 priority of the request < LOW-LEVEL

Queue-2: LOW-LEVEL <= BS2000 priority of the request < HIGH-LEVEL

Queue-3: HIGH-LEVEL <= BS2000 priority of the request <= 255

The highest priority (30) and the lowest priority (255) represent the limits between which the three priority classes for requests to the DBH are spread. LOW-LEVEL and HIGH-LEVEL values define the limits between Queue-1 and Queue-2 and between Queue-2 and Queue3. Their default values are 180 and 240 and you can modify them using the DBH option REQUEST-CONTROL.

During processing the requests are deactivated internally after N logical block accesses according to their priority class. N is set according to the priority class weighting. This allows requests with a higher weighting to be processed faster than requests with a lower weighting.

```
REQUEST-CONTROL

= *NONE / *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | ,QUEUE-PRIORITY = *PARAMETERS(...)
    |   *PARAMETERS(...)
    |     | QUEUE-1 = 3 / <integer 1..99>
    |     | ,QUEUE-2 = 2 / <integer 1..99>
    |     | ,QUEUE-3 = 1 / <integer 1..99>
    | ,QUEUE-RANGE = *PARAMETERS(...)
    |   *PARAMETERS(...)
```

```
|       | LOW-LEVEL = 180 / <integer 30..255>
|       | ,HIGH-LEVEL = 240 / <integer 30..255>
```

Operands

REQUEST-CONTROL = *NONE

Priority control is not activated. This is the default when the DBH starts up.

REQUEST-CONTROL = *STD
Activates priority control for the independent DBH. Default values apply to the various priority-control operands. These defaults are described under *PARAMETERS(...). During the DBH session, you can turn priority control on or off using the appropriate administration statement (see "MODIFY-REQUEST-CONTROL").

REQUEST-CONTROL = *PARAMETERS(...)

QUEUE-PRIORITY = *PARAMETERS(...)
Specifies the weights of the different priority classes. You can also change the weights subsequently using an administration statement (see "MODIFY-REQUEST-CONTROL").

QUEUE-1 = 3 / <integer 1..99>
Specifies the weight of the first priority class, Queue-1. The default is 3.

QUEUE-2 = 2 / <integer 1..99>
Specifies the weight of the second priority class, Queue-2. The default is 2.

QUEUE-3 = 1 / <integer 1..99>
Specifies the weight of the third priority class, Queue-3. The default is 1.

QUEUE-RANGE = *PARAMETERS(...)

Specifies the limit values between the priority classes Queue-1 and Queue-2, and Queue-2 and Queue-3. You can modify the limit values during the DBH session using the appropriate administration statement (see "MODIFY-REQUEST-CONTROL").

LOW-LEVEL = 180 / <integer 30..255>
Specifies the limit value between the first and second priority classes, Queue-1 and Queue-2. The LOW-LEVEL value must be lower than the HIGH-LEVEL value. The default for LOW-LEVEL is 180.

HIGH-LEVEL = 240 / <integer 30..255>
Specifies the limit value between the second and third priority classes, Queue-2 and Queue-3. The HIGH-LEVEL value must be higher than the LOW-LEVEL value. The default for HIGH-LEVEL is 240.

Considerations when assigning weights

In order to achieve optimal weight assignment for the priority classes, you should ask the following questions:

- How many users of each priority class are working concurrently?
- Which priority class is to receive preferential treatment, and which requests hinder it?
- Is the load distribution dependent on the time of day or date? Must the weighting be changed at specific times?

You should take the answers to these questions into consideration when assigning the weights.

If the load distribution shifts dramatically and unforseeably, you can change the weights dynamically using appropriate administration statements (see "MODIFY-REQUEST-CONTROL").

3.2.2.21 RESTART-CONTROL

This statement controls how long it takes until the DBH becomes available in the event of a restart.

Definition

A lower-level DBH option of SYSTEM-STRATEGIES

Function

You can use the DBH option RESTART-CONTROL to optimize the time it takes for the DBH to become available again after a system failure. You will find general information on restarting in the section “Restarting a DBH session”.

You can influence how long a restart takes in two different ways:

- The BUFFER-LIMIT and TALOG-LIMIT operands affect the frequency of write accesses to the database. The lower the values for BUFFER-LIMIT and TALOG-LIMIT, the more afterimages are written to the database during operation. This reduces the time required for physical repair in the event of a restart. However, if very low values are selected, the increased number of read and write accesses during normal operation lead to loss of performance. It therefore makes sense to monitor the number of accesses in the “I/O” DBH form of the SESAM /SQL utility SESMON (see "DBH form I/O"). You can adjust the values of the operands during the DBH session by means of the MODIFY-RESTART-CONTROL administration statement (see "MODIFY-RESTART-CONTROL").
- The LOGICAL-ROLLBACK operand allows you to delay the logical reset until the start of normal operation. The transactions that were open when the system failed are thus not reset until after the restart. This reduces the time required until the system is available again.

```
RESTART-CONTROL

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | BUFFER-LIMIT= *PARAMETERS(...)

    |    *PARAMETERS(...)

    |       | SYSTEM-DATA-BUFFER = 50 / <integer 1..90>

    |       | ,USER-DATA-BUFFER = 50 / <integer 1..90>

    | ,TALOG-LIMIT = 512 / <integer 128..524288>

    | ,LOGICAL-ROLLBACK = *IMMEDIATELY / *DELAYED
```

Operands

RESTART-CONTROL = *STD

The default values are used. The default values are described under *PARAMETERS.

RESTART-CONTROL = *PARAMETERS(...)

You can change the values set for BUFFER-LIMIT and TALOG-LIMIT to suit requirements during the DBH session by means of the MODIFY-RESTART-CONTROL administration statement (see "MODIFY-RESTART-CONTROL"). You must specify the BUFFER-LIMIT and TALOG-LIMIT operands **before** a system crash. Specifying them at the time of the external restart has no effect. The LOGICAL-ROLLBACK operand, on the other hand, is evaluated at the restart. It can therefore also be set immediately before an external restart.

BUFFER-LIMIT = *PARAMETERS(...)
These operands specify the limit as a percentage for the occupation of the two buffers by writable blocks. As of this limit, the buffers are written to the database.

SYSTEM-DATA-BUFFER = 50 / <integer 1..90>
Specifies a percentage limit for the writing of afterimages to the database for system access data. If the share of the total buffer space occupied by the buffers to be written exceeds the percentage specified for SYSTEM-DATA-BUFFER, afterimages are written to the database.

USER-DATA-BUFFER = 50 / <integer 1..90>
Specifies a percentage limit for the writing of afterimages to the database for user data. If the share of the total buffer space occupied by the buffers to be written exceeds the percentage specified for USER-DATA-BUFFER, afterimages are written to the database.

TALOG-LIMIT = 512 / <integer 128..524288>
Specifies the limit for how much of the TA-LOG file can be occupied by afterimages. In the event of physical repair, this part of the TA-LOG file must be read. If its size exceeds the specified limit, the afterimages are written to the database. TALOG-LIMIT is specified in units of 4 Kbytes.

LOGICAL-ROLLBACK =
Influences the time at which transactions are logically reset in the event of a restart.

LOGICAL-ROLLBACK = *IMMEDIATELY
The transactions are reset before the resumption of normal operation.

LOGICAL-ROLLBACK = *DELAYED

Delays transaction resetting until the resumption of normal operation. This reduces the time required until the system becomes available again. However, if too many updates have to be performed at the beginning of normal operation, this can result in loss of performance and the growth in size of the TA-LOG file. In certain cases, delaying the logical reset is not effective (see the section “Controlling the duration of the restart”).

The LOGICAL-ROLLBACK operand cannot be changed during DBH operation.

3.2.2.22 RETRIEVAL-CONTROL

This statement influences the processing strategy of the DBH.

Definition

A lower-level DBH option of SYSTEM-STRATEGIES

Function

The DBH option RETRIEVAL-CONTROL allows you to influence the processing strategy of the DBH.

In a sequential search, the RETRIEVAL-CONTROL option causes the DBH under certain conditions to divide the processing of retrieval statements into sections, and interrupt or even completely abort them. Interrupted retrieval statements are placed in a queue until processing continues. The tasks that are released during the interrupts or after being aborted become available for other requests.

If the search involves secondary indexes, RETRIEVAL-CONTROL causes the DBH to continue the search sequentially when it is appropriate in view of the set of records expected to be found. The DBH can then interrupt or abort the sequential processing, as described above.

```
RETRIEVAL-CONTROL

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | INTERRUPT-EXECUTION = *PARAMETERS(...)

    |    *PARAMETERS(...)

    |       | USER-DATA-ACCESS = 1000 / <integer 1..2147483647>

    |       | ,INDEX-EVALUATION = 2147483647 / <integer 16..2147483647>

    | ,ABORT-EXECUTION = 2000 / <integer 1.. 2147483647>
```

Operands

RETRIEVAL-CONTROL = *STD

The search strategy of the DBH changes in the case of retrieval statements that take a long time to process. Under what conditions and how the search strategy is influenced is defined by default values. These are described under *PARAMETERS.

RETRIEVAL-CONTROL = *PARAMETERS(...)

INTERRUPT-EXECUTION = *PARAMETERS(...)
Under certain conditions, the DBH interrupts the processing of retrieval statements that take a long time to process.

USER-DATA-ACCESS = 1000 / <integer 1..2147483647>
Sequential processing is divided into sections if the number of logical accesses of blocks of user data exceeds a defined limit value. Processing is interrupted on completion of each section. The default for this limit value is 1000.
If necessary, you can change the value specified here during the DBH session by means of the MODIFY-RETRIEVAL-CONTROL administration statement.

INDEX-EVALUATION = 99999999 / <integer 16..2147483647>

Processing with secondary indexes is continued sequentially if the set of records expected to be found exceeds a defined limit value. The default limit value is 99,999,999.

Based on the statistical data kept for each column or attribute, the DBH determines the set of records expected to be found for the subquestion processed first. If the set of records found exceeds the limit value, the DBH continues sequential processing. As of this point, the processing strategy of the DBH follows the rules for sequential processing.

If the set of records found does not exceed the limit value, the DBH processes further subquestions using secondary indexes.

If necessary, you can change the value specified here during the DBH session by means of the MODIFY-RETRIEVAL-CONTROL administration statement (see "MODIFY-RETRIEVAL-CONTROL").

ABORT-EXECUTION = 2000 / <integer 1.. 2147483647>

Sequential processing is aborted completely when the number of failed logical accesses of user data exceeds a defined limit value. The default limit value is 2000.

If necessary, you can change the value specified here during the DBH session by means of the MODIFY-RETRIEVAL-CONTROL administration statement. Using the pragma LIMIT ABORT_EXECUTION <integer 1.. 2147483647> it can also be changed in relation to a request, see the “ SQL Reference Manual Part 1: SQL Statements”, section “Pragmas”.

3.2.2.23 SECURITY

This statement prevents unauthorized access and controls SAT logging.

Definition

A lower-level DBH option of ADMINISTRATION

Function

CALL DML tables can be protected by password. Security relevant events can be logged with SAT.

You use the DBH option SECURITY to specify the following:

- The number of password violations a CALL DML user working in interactive mode may commit
- Logging security-relevant events with SAT

If a user exceeds the specified number of permitted violations, he or she is prevented from sending any further requests to the DBH during this DBH session. The system administrator can use the RELEASE-USER-RESOURCES administration statement to reset the user's resources and lift the lock (see "RELEASE-USER-RESOURCES").

If a CALL DML user has already opened a logical file, it is not necessary for the system administrator to lift the lock. The CALL DML user can carry out a user-close and subsequently work with the DBH again. If the user-close is carried out during a transaction, the user cannot send any more requests to the DBH until after the end-of-transaction statement.

You can adjust the values of the option during the DBH session by means of the MODIFY-SECURITY administration statement (see "MODIFY-SECURITY").

```
SECURITY

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | PRIVILEGE-VIOLATIONS = 10 / <integer 1..99>

    | ,SAT-SUPPORT = *OFF / *ON
```

Operands

SECURITY = *STD

The default values are used. The default values are described under *PARAMETERS.

SECURITY = *PARAMETERS(...)

PRIVILEGE-VIOLATIONS = 10 / <integer 1..99>
The number of attempts permitted to an SQL user to access a table or the number of permitted password violations in the case of CALL DML OPEN statements. The default value for PRIVILEGE-VIOLATIONS is 10 (for users working in the interactive mode).
With application programs in batch mode, only one password violation or access attempt is permitted.

SAT-SUPPORT = *OFF / *ON
Controls the SAT logging. In the case of SAT-SUPPORT=*ON security relevant events are transmitted to SAT for logging. See also the “ Core manual”, section “Logging of security relevant events with SAT”.

3.2.2.24 SERVICE-TASKS

This statement specifies the number of tasks for service requests.

Definition

A lower-level option of CPU-RESOURCES

Function

CPU-intensive activities, such as the sorting of intermediate results and some utility functions, can last a very long time. To avoid being blocked during this time, the SESAM/SQL DBH can relocate these CPU-intensive activities as separate tasks, which are then available for service requests.

At the end of the session, the DBH terminates all service tasks. Service requests that are still open or running are aborted.

You use the DBH option SERVICE-TASKS to specify how many service tasks are to be available at the beginning of the DBH session and the maximum number of service tasks that can be started in the course of the session.
You can also specify whether the DBH is to use the multitask sorting facility of the software product SORT.
You can also enter specifications for transferring information by email.

> To ensure that the service tasks can also be started, the following points must be observed:
>
> - There must be no read password to the module library from which SESAM/SQL is started.
> - When SESAM/SQL is started, no temporary file must be used as a module library.
> - The module library must be available for the entire duration of the DBH session.
> - All files assigned with the link name BLSLIB*nn*, where *nn* = 00 to 99, when SESAM/SQL is started must be accessible for the entire duration of the DBH session.

You can adjust the values of the option during the DBH session by means of the MODIFY-SERVICE-TASKS administration statement (see "MODIFY-SERVICE-TASKS").

```
SERVICE-TASKS

= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | INITIAL = 1 / <integer 0..32>
    | ,MAXIMUM = *STD / <integer 0..64>
    | ,JOBCLASS = *STD / *DBH-JOBCLASS / <name 1..8>
    | ,WORK-FILES = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |   *PUBLIC-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
    |     | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
    |     | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
    |   *PRIVATE-DISK(...)
    |     | CAT-ID = *STD / <cat-id>
```

```
|       | ,PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
|       | ,SECONDARY-ALLOCATION = *STD / <integer 1..32767>
|       | ,DEVICE-TYPE = <structured-name 1..8>
|       | ,VOLUME = list-poss(6):<alphanum-name 1..6>
| ,RECORDS-PER-CYCLE = *STD / <integer 1..2147483647>
| ,MAIL = *NONE / *PARAMETERS(...)
|   *PARAMETERS(...)
|       | OUTPUT = *ALL / *ERROR
```

Operands

SERVICE-TASKS = *STD

The number of service tasks is determined by default values; these are described under *PARAMETERS.

SERVICE-TASKS = *PARAMETERS(...)

INITIAL = 1 / <integer 0..32>
Number of service tasks started at the beginning of the DBH session.
The default value for INITIAL is 1, which means that when the DBH is loaded, at least one service task is started and immediately available for service requests. When using the linked-in DBH, no more than one service task can be started.

MAXIMUM = *STD / <integer 0..64>
Maximum number up to which further service tasks may be started during the DBH session.
The default value for MAXIMUM is the greater of the following two values (with a maximum of 64):

a. the maximum number of concurrent active threads, specified by means of the DBH option THREADS (see "THREADS")
b. the value assigned to the INITIAL operand

- In the case of the independent DBH, note the following:
  if you enter 0 for MAXIMUM, this value is automatically set to 1. If you specify a value for MAXIMUM that is greater than 0 but less than that for INITIAL, you receive an error message.
- In the case of a linked-in DBH, note the following:
  no more than one service task is started, even if you specify a value greater than 1 for MAXIMUM. If you specify MAXIMUM = 0, no service task is started. In this case, the DBH can carry out the sorting of intermediate result sets as a subroutine in its own task. Other service requests, such as the utility statements COPY, LOAD and RECOVER, are rejected.

JOBCLASS = *STD / <name 1..8>
Name of the BS2000 job class.
The batch job is to run in the default job class or in a job class that you specify after making your selection. The standard job class is the default job class for batch jobs of the DBH identifier. It is taken from the JOIN entry of the user ID.

JOBCLASS = *DBH-JOBCLASS
The name of the BS2000 job class is the job class of the DBH when the service task is started as long as the DBH is started as a batch job. In interactive mode the specification of *DBH-JOBCLASS is ignored and the procedure is the same as for JOBCLASS=*STD.

WORK-FILES =
Temporary SORT work files which are generated within a service task are created before each sort run and deleted after it.

WORK-FILES = *PUBLIC-DISK(...)
Temporary SORT work files are stored on a public disk.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the temporary SORT work files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space. The default value is 120.

SECONDARY-ALLOCATION = *STD / <integer 1..32767>
Number of PAM pages for subsequent extensions to files. The default value is 120.

WORK-FILES = *PRIVATE-DISK(...)

Temporary SORT work files are stored on a private disk.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the temporary SORT work files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space. The default value is 120.

SECONDARY-ALLOCATION = *STD / <integer 1..32767>
Number of PAM pages for subsequent extensions to files. The default value is 120.

DEVICE-TYPE = <structured-name 1..8>
BS2000 device type to which the required disks are assigned.
If you make an incorrect entry, the system prompts you in interactive mode to make the entry again (see the device type table in the BS2000 manual “ Commands”).

VOLUME = list-poss(6): <alphanum-name 1..6>
VSNs of the disks on which the file is to be created. Up to 6 VSNs are permitted, each of which can be specified only once. VOLUME is not evaluated unless the DEVICE-TYPE operand is specified.

RECORDS-PER-CYCLE = *STD / <integer 1..2147483647>
Number of records to be sorted per cycle by a sort subtask during multitask sorting. SESAM/SQL passes this value to the BS2000 utility SORT in the CYCLE parameter (see also the SORT manual “ SDF Format”). The “ Performance” manual describes the criteria on the basis of which the RECORDS-PER-CYCLE value should be assigned. The total number of records to be sorted divided by CYCLE is the optimal number of work files to be created. However, no more than 9 temporary work files can be created. The number of work files

determines the number of sort subtasks started by the SORT utility (number of sort subtasks = number of work files minus 1). In single-task sorting, work files are not created unless one of the following operands is specified: CAT-ID, PRIMARY-ALLOCATION or DEVICE-TYPE.

*STD means that the DBH does not use the multitask sorting facility.

MAIL =

Controls the output of information by email.

MAIL = *NONE
The information is not sent by email.

MAIL = *PARAMETERS(...)
Controls the volume of information which is sent by email.

OUTPUT = *ALL
When the service task terminates its SYSLST file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email"). The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST (SVT, TSN=<tsn of the service task>” is entered under “Subject:”. The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The SYSLST file is attached to the email.

OUTPUT = *ERROR
Only if the server task terminates with an error its SYSLST file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email"). The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST (SVT, TSN=<tsn of the service task>” is entered under “Subject:”. The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The SYSLST file is attached to the email.

Information on service tasks

Information about the utilization of the service tasks in the current session is provided by the SESAM/SQL utility routine SESMON in the SERVICE TASKS form (see "DBH form SERVICE TASKS").

3.2.2.25 SESSION-LOGGING-ID

This statement identifies session-specific log files.

Definition

A lower-level DBH option of FILE-RESOURCES

Function

The session-specific log file CO-LOG contains in its name a session identification whose default is the BS2000 task sequence number (TSN).
The CO-LOG file is also identified by a file counter whose default value at the beginning of the DBH session is 1.
The default file name of the request log file (CO-LOG), for example, is SESAM.CO-LOG.*ssss.iiii*, where *ssss* stands for the session identification and *iiii* for the initial value of the file counter.

The DBH option SESSION-LOGGING-ID allows you to specify different values for the session identification and the initial file-counter value.

You can adjust the values of the operands during the DBH session by means of the MODIFY-SESSION-LOGGING-ID administration statement (see "MODIFY-SESSION-LOGGING-ID").

```
SESSION-LOGGING-ID
```

```
= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | SESSION-ID = *STD / <alphanum-name 4..4>

    | ,START-NUMBER = 0001 / <integer 1..9999>
```

Operands

SESSION-LOGGING-ID = *STD

The session identification and file counter have default values. These are described under *PARAMETERS(...).

SESSION-LOGGING-ID = *PARAMETERS(...)

SESSION-ID =
Specifies the session identification for the CO-LOG file.

SESSION-ID = *STD
The default value for the session identification corresponds to the BS2000 task sequence number (TSN).

SESSION-ID = <alphanum-name 4..4>
Session identification that you yourself can specify.

START-NUMBER =
Specifies the initial value of the file counter for the CO-LOG file.

START-NUMBER = 0001
Sets the file counter to the default value of 1 at the beginning of the DBH session.

START-NUMBER = <integer 1..9999>
Initial value of the file counter that you yourself can specify.
The four-digit value is used in filenames (e.g. 0003).

3.2.2.26 SET-DBH-OPTIONS

This statement causes the DBH options to be read in.

Definition

First DBH start statement

Function

The DBH start statement SET-DBH-OPTIONS causes the DBH options to be read in and initiates the parametrization of the DBH.

SET-DBH-OPTIONS is the first start statement you have to enter. You can follow it with the END statement END concludes input of the DBH start statements.

Do not enter the END statement after SET-DBH-OPTIONS if you want to enter another start statement. You can enter the SET-DBH-OPTIONS statement only once at each DBH startup.

SET-DBH-OPTIONS
DBH-IDENTIFICATION=*STD/*PARAMETERS(...) ,ADMINISTRATION=*STD/*PARAMETERS(...) ,CPU-RESOURCES=*STD/*PARAMETERS(...) ,FILE-RESOURCES=*STD/*PARAMETERS(...) ,LINKED-IN-ATTRIBUTES=*STD/*PARAMETERS(...) ,RECOVER-OPTIONS=*STD/*PARAMETERS(...) ,STORAGE-SIZE=*STD/*PARAMETERS(...) ,SYSTEM-LIMITS=*STD/*PARAMETERS(...) ,SYSTEM-STRATEGIES=*STD/*PARAMETERS(...)

Operands

The operands of the DBH start statement SET-DBH-OPTIONS are higher-level DBH options. If you do not want to use the defaults, enter the relevant higher-level DBH options here with the values you want to use for the lower-level DBH options. The DBH options are described in detail in this reference section. You will find a brief description of the DBH options as of "DBH options".

3.2.2.27 SPACES

This statement specifies the number of simultaneously accessible spaces.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

A space is a BS2000 file. Each database has one catalog space, which contains the metadata with the database schemas, and up to 999 user spaces, in which tables and indexes are stored.

You use the DBH option SPACES to specify the total number of spaces to be concurrently accessible during the DBH session. It does not matter whether or not the spaces really are accessed. If database A consists of 10 spaces, for example, you have to specify a value >= 10 for the SPACES option even if only 5 spaces are accessed concurrently in this case.

Note that the number of simultaneously accessible spaces affects the number of possible CREATE SPACE statements.

You can adjust the value of the option during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

SPACES
= *STD / <integer 1..254000>

Operands

SPACES = *STD / <integer 1..254000>

Specifies the maximum number of simultaneously accessible spaces during the DBH session. The default for SPACES depends on the maximum number of entries the SQL database catalog can contain (see the DBH option SQL-DATABASE-CATALOG on "SQL-DATABASE-CATALOG"). It is 10 * (DBH option SQL-DATABASE-CATALOG). The maximum value for SPACES is 1000 times the value of SQL-DATABASE-CATALOG, since a database can contain a maximum of 1000 spaces.If a value less than 2 * (DBH option SQL-DATABASE-CATALOG) is specified, it is corrected automatically to 2 * (DBH option SQL-DATABASE-CATALOG).

Information on the number of space accesses

You obtain information on the total number of space accesses during the DBH session from the “I/O” form of the SESAM/SQL monitor SESMON (see "DBH form I/O"). SESCOSP, which permits the evaluation of request logging (see "Evaluating request logging with SESCOSP"ff), also provides information on the number of times individual spaces are accessed.

3.2.2.28 SQL-DATABASE-CATALOG

This statement limits the number of entries in the SQL database catalog.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

You use the DBH option SQL-DATABASE-CATALOG to specify the maximum number of entries the SQL database catalog can contain.

You edit the SQL database catalog by means of the DBH start statement ADD-SQL-DATABASE-CATALOG-LIST (see "ADD-SQL-DATABASE-CATALOG-LIST"). You cannot enter more databases in the catalog than specified in the DBH option SQL-DATABASE-CATALOG, but it is permitted to enter fewer databases. If more databases are required during the DBH session, you can enter them using the appropriate administration statement until the maximum permissible number of entries is reached.

SQL-DATABASE-CATALOG
= 1 / <integer 1..254>

Operands

SQL-DATABASE-CATALOG = 1 / <integer 1..254>

Maximum number of database entries in the SQL database catalog. The default is 1, but the SESAM/SQL DBH can handle up to 254 databases simultaneously.

> Note that a free entry in the SQL database catalog is required for each CREATE CATALOG and each CREATE REPLICATION statement.

3.2.2.29 SQL-SUPPORT

This statement supports the SQL interface and specifies limit values.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

You use the DBH option SQL-SUPPORT to specify whether the DBH is to support the SQL interface. If necessary, you can also specify limit values for important SQL-specific resources.

If the DBH supports the SQL interface, users can use both CALL DML statements and SQL and utility statements. However, this presupposes that transaction management is activated (see the DBH option TRANSACTION-SECURITY on "TRANSACTION-SECURITY").

```
SQL-SUPPORT

= *YES(...) / *NO

  *YES(...)

    | INACTIVITY-TIME = 0 / <integer 0..999>

    | ,CURSORS = 70 / <integer 0..262143>

    | ,INTERNAL-SORT-LIMIT = 200 / <integer 1..2147483647>

    | ,PLANS = 70 / <integer 1..999999>
```

Operands

SQL-SUPPORT = *YES(...)

The DBH supports the SQL interface.

INACTIVITY-TIME = 0 / <integer 0..999>
Maximum permitted inactivity time in minutes at the DBH's inactivity check. The inactivity time is the time during which an SQL conversation does not have a transaction open. If the inactivity time exceeds the specified limit value, the resources of the SQL conversation are reset when there is a bottleneck.
The default value for INACTIVITY-TIME is 0; i.e. an inactivity check is not carried out, and inactive conversations are not eliminated.

> ¡ You can adjust the value of the INACTIVITY-TIME operand by means of the SET-USER-INACTIVE-TIME administration statement (see "SET-USER-INACTIVE-TIME") to suit requirements.

CURSORS = 70 / <integer 0..262143>

Number of concurrent cursors that can be declared in the DBH session.

> ¡ You can adjust the value of the CURSORS operand by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

INTERNAL-SORT-LIMIT = 200 / <integer 1..2147483647>
Maximum number of records that a cursor table can contain if the records are to be sorted in accordance with

the cursor declaration.
If the number of records found during the sort exceeds this limit value, processing of the statement is aborted.

> You can adjust the value of the INTERNAL-SORT-LIMIT operand by means of the MODIFY-SQL-SORT-LIMIT administration statement (see "MODIFY-SQL-SORT-LIMIT") to suit requirements.

PLANS = 70 / <integer 1..999999>
Minimum number of concurrently available SQL access plans.

> You can adjust the value of the PLANS operand by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

SQL-SUPPORT = *NO
The DBH does not support the SQL interface. As a result, the DBH does not require so many Mbytes of virtual address space.
You cannot specify SQL-SUPPORT=*NO unless only CALL DML applications are to run in the DBH session. Utility statements are not permitted in this session either.

Size of the plan buffer

The plan buffer is an area of main memory reserved by the DBH exclusively for SQL access plans.

The size of the plan buffer depends essentially on the size of the area for retrieval statements. This is set by means of the DBH option COLUMNS.
In the case of the default values PLAN=70, COLUMNS=256 and USERS=24, the size of the plan buffer is approximately 1.1 Mbytes.

The “ Performance” manual describes when it makes sense to modify the size of the plan buffer the next time the DBH is started up.

3.2.2.30 STORAGE-SIZE

This statement sets buffer and container sizes.

Definition

Higher-level DBH option

Function

STORAGE-SIZE is the higher-level option of DBH options with which you can set the size of buffers and containers.

```
STORAGE-SIZE

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | CURSOR-BUFFER = *STD

    | ,TRANSFER-CONTAINER = *STD

    | ,WORK-CONTAINER = *STD

    | ,SYSTEM-DATA-BUFFER = 208

    | ,USER-DATA-BUFFER = 80
```

Operands

STORAGE-SIZE = *STD

Default values apply to all STORAGE-SIZE operands, which are described under the entries for the relevant lower-level DBH options.

STORAGE-SIZE = *PARAMETERS(...)
The operands of STORAGE-SIZE are DBH options. Their operands are described under the entries for the relevant options.

3.2.2.31 SUBORDERS

This statement makes SQL scans and logical files available.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

You use the DBH option SUBORDERS to specify the number of suborders for which resources are to be provided during the DBH session. This restricts main-memory requirements.

The meaning of suborders differs depending on whether it applies to an SQL or a CALL DML application:

- In the case of an SQL application, suborders are SQL scans, i.e. subareas of an SQL access plan, the evaluation rule for an SQL statement.
- In the case of a CALL DML application, suborders are logical files for CALL DML requests.

```
SUBORDERS

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | INITIAL = 24 / <integer 6..262143>

    | ,MAXIMUM = *STD / <integer 6..262143>
```

Operands

SUBORDERS = *STD

The default numbers of SQL scans or logical files (suborders) permitted apply. These are described under *PARAMETERS.

SUBORDERS = *PARAMETERS(...)

INITIAL = 24 / <integer 6..262143>

Initial value for the number of concurrent suborders for which resources are to be available.

The default value for INITIAL is 24.

An internal comparison ensures that the permissible number of suborders is at least as high as the maximum number of concurrent users (see the DBH option USERS on "USERS"). If the maximum number of concurrent users is greater than the value specified here, the initial value is increased automatically.

MAXIMUM = *STD / <integer 6..262143>
Maximum permissible number of concurrent suborders.
If the initial value specified for the INITIAL operand is not high enough, SESAM/SQL allows resources to be made available dynamically for additional suborders. The default value for MAXIMUM is the initial value for the permissible number of concurrent suborders specified by means of the INITIAL operand.

You can change the maximum value for the permissible number of concurrent suborders during the DBH session by means of the MODIFY-SUBORDER-LIMIT administration statement(see "MODIFY-SUBORDER-LIMIT").

Information on suborders

You will find the number of used suborders and the maximum number of available suborders in the “SYSTEM INFORMATION” form of the SESAM/SQL monitor SESMON (see "DBH form SYSTEM INFORMATION").

The “ Performance” manual describes the cases in which it makes sense to modify the number of available resources for suborders at the next DBH startup.

3.2.2.32 SYSTEM-DATA-BUFFER

This statement sets the size of the buffer for system-access data.

Definition

A lower-level DBH option of STORAGE-SIZE

Function

The SESAM/SQL DBH has two separate buffers: one for blocks with system-access data and one for blocks with user data. The blocks are each 4 Kbytes in size. All users use both buffers together for all connected databases.

You use the DBH option SYSTEM-DATA-BUFFER to set the initial size of the buffer for system-access data, i.e. data required by the system for the physical organization of memory.

The final size of the buffer for system-access data depends on the DBH option THREADS (see "THREADS") and the system threads dependent on this. The minimum value for the buffer size is 96 KB per thread. If the specified value is smaller than the minimum value, then the value is automatically increased to the minimum value as long as the specified value is within the permitted range of values (see below). Otherwise the DBH is terminated.

You can adjust the value of the option during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

SYSTEM-DATA-BUFFER
= 208 / <integer 80..64000000>

Operands

SYSTEM-DATA-BUFFER = 208 / <integer 80..64000000>

Initial size of the buffer for system-access data in units of 1 Kbyte. The default is 208 Kbytes.

The maximum value for SYSTEM-DATA-BUFFER is 64 000 000 Kbytes (64 Gbytes) on all current BS2000 servers.

> The sum of the sizes for SYSTEM-DATA-BUFFER and USER-DATA-BUFFER may not exceed 64 000 000 Kbytes (64 Gbytes).
> If the system configuration so permits, the SYSTEM-DATA-BUFFER is created in a data space. Otherwise it is created in the normal class 6 memory.

Information on input/output behavior

The “I/O” form of the SESAM/SQL monitor SESMON contains information on input/output behavior which is helpful for setting the optimum size of the buffer for system-access data (see "DBH form I/O").
The “ Performance” manual describes the cases in which it makes sense to modify the size of the buffer for system-access data at the next DBH startup.

3.2.2.33 SYSTEM-LIMITS

This statement specifies limit values.

Definition

Higher-level DBH option

Function

SYSTEM-LIMITS is the higher-level option of all DBH options with which you can specify limit values for the DBH session.

```
SYSTEM-LIMITS

 = *STD / *PARAMETERS

   *PARAMETERS(...)

      | COLUMNS = 256

      | ,SQL-SUPPORT = *YES(...)

      | ,SUBORDERS = *STD

      | ,SYSTEM-THREADS = *STD

      | ,THREADS = 1

      | ,USERS = *STD

      | ,SQL-DATABASE-CATALOG = 1

      | ,OLD-TABLE-CATALOG = 0

      | ,SPACES = *STD
```

Operands

SYSTEM-LIMITS = *STD

Default values apply to all the operands of SYSTEM-LIMITS. They are described under the relevant lower-level DBH options.

SYSTEM-LIMITS = *PARAMETERS(...)
The various operands of SYSTEM-LIMITS are DBH options. Their operands are described under the relevant DBH options.

3.2.2.34 SYSTEM-STRATEGIES

This statement defines a processing strategy.

Definition

Higher-level DBH option

Function

SYSTEM-STRATEGIES is the higher-level option of DBH options with which you can define a processing strategy.

```
SYSTEM-STRATEGIES
```

```
= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | REQUEST-CONTROL = *NONE

    | ,RESTART-CONTROL = *STD

    | ,RETRIEVAL-CONTROL = *STD

    | ,TRANSACTION-SECURITY = *YES(...)
```

Operands

SYSTEM-STRATEGIES = *STD

Default values apply to all the operands of SYSTEM-STRATEGIES. They are described under the relevant lower-level DBH options.

SYSTEM-STRATEGIES = *PARAMETERS(...)
The operands of SYSTEM-STRATEGIES are DBH options. Their operands are described under the entries for the relevant options.

3.2.2.35 SYSTEM-THREADS

This statement specifies the number of concurrent system threads.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

System threads are created in order to deal with internal system requests to the spaces. The lower-level DBH option SYSTEM-THREADS specifies the number of these system threads.

In the case of the linked-in DBH, the DBH option SYSTEM-THREADS is ignored; the default value of 1 applies to WRITE-THREADS.

You can adjust the value of the option with the administration statement RELOAD-DBH-SESSION (see "RELOAD-DBH-SESSION") when the DBH modules are reloaded.

```
SYSTEM-THREADS
= *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | WRITE-THREADS = *STD / <integer 1..512>
```

Operands

SYSTEM-THREADS = *STD

Specifies the maximum number of concurrently active system threads. The default values are described under *PARAMETERS.

SYSTEM-THREADS = *PARAMETERS(...)

WRITE-THREADS =
Specifies the maximum number of concurrent system threads that deal with open write requests to the spaces. These system threads are known as write threads.

WRITE-THREADS = *STD / <integer 1..512>
You can specify the number of write threads independently of the number of threads specified by means of the DBH option THREADS.
However, the default value is obtained from the number of threads divided by 32. If this calculation does not produce an integer, the figure is rounded up to the next integer.

The size of the value specified for the WRITE-THREADS operand has the following effects:

- If a very low number is selected, the TA-LOG files will be larger than expected. The CPU requirements will be considerably increased. As a result the writing of the after-images, as specified in the option RESTART-CONTROL, may no longer be guaranteed.
- A very high value, on the other hand, affects the address space requirements rather than the CPU requirements.

Information on system threads

You will find information on the number of concurrent system threads in the “SYSTEM THREADS” form of the SESAM/SQL monitor SESMON (see "DBH form SYSTEM THREADS").

3.2.2.36 THREADS

This statement specifies the number of concurrent threads.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

In multithread mode, the DBH processes several requests simultaneously. Each request is stored in a thread.

You use the DBH option THREADS to specify how many threads are to be available in the DBH session for requests from applications. This restricts indirectly the number of concurrent requests.

You will avoid system bottlenecks by specifying a number of threads that is less than or equal to the number of tasks of all the application programs. You should also count applications such as the SESUTI utility monitor or the SESADM administration program for this purpose. If a system bottleneck occurs because the value specified for the THREADS option is too low, it is removed by rolling back the transaction. This ensures that not all threads are blocked by blocked requests.

For an independent DBH an internal comparison is carried out so that the value for THREADS is greater than or equal to twice the value of the DBH-TASKS option. For a linked-in DBH the DBH option THREADS is ignored, i.e. the value 1 is the default value for threads.

You can adjust the value of the option with the administration statement RELOAD-DBH-SESSION (see "RELOAD-DBH-SESSION") when the DBH modules are reloaded.

THREADS
= 1 / <integer 1..1024>

THREADS = 1 / <integer 1..1024>

Specifies the maximum number of concurrently active threads.

Information on thread utilization

You can obtain information on the current utilization of the threads from the “SYSTEM-INFORMATION” form of the SESAM/SQL monitor SESMON (see "DBH form SYSTEM INFORMATION"). The “ Performance” manual describes the cases in which it makes sense to modify the number of threads at the next DBH startup.

3.2.2.37 TRANSACTION-SECURITY

Activates transaction management

Definition

A lower-level DBH option of SYSTEM-STRATEGIES

Function

Transaction management coordinates concurrent database accesses. In the event of an error, it guarantees data consistency by means of rollback mechanisms.

Every DBH session runs with transaction management.

The DBH option TRANSACTION-SECURITY allows you to modify the rollback criteria for transactions locking other transactions to suit the requirements of the DBH session.

You can change the LOCK-TIME, INACTIVITY-TIME and LOCK-ESCALATION operands during the DBH session by using the MODIFY-TRANSACTION-SECURITY administration statement (see "MODIFY-TRANSACTION-SECURITY").

You can adjust the values of the MAX-ISOLATION-LEVEL operand during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

```
TRANSACTION-SECURITY

= *YES(...)  1
    *YES(...)
        |   LOCK-TIME = 4 / <integer 1..999>
        |   ,INACTIVITY-TIME = *STD / <integer 1..999>
        |   ,MAX-ISOLATION-LEVEL = *STD / *REPEATABLE-READ
        |   ,LOCK-ESCALATION = *STD / *PARAMETERS(...)
        |       *PARAMETERS(...)
        |           |  NUMBER-RECORDS = 4000 / <integer 1..2147483647>
        |           |  ,PERCENTAGE-RECORDS = 50 / <integer 0..100>
        |           |  ,NUMBER-INDEX-VALUES = 1000 / <integer 1..2147483647>
```

[1]For reasons of compatibility, the value TRANSACTION-SECURITY = *NO of SESAM/SQL < V7.0 can still be specified. However, it is ignored. A warning is issued. TRANSACTION-SECURITY = *YES(...) is always used.

Operands

TRANSACTION-SECURITY = *YES(...)

The DBH always runs with transaction management.

LOCK-TIME = 4 / <integer 1..999>
Specifies the number of minutes after which an inactive transaction locking other transactions is to be rolled back. The default is 4 minutes.

INACTIVITY-TIME = *STD / <integer 1..999>
Specifies the number of minutes after which an open but inactive transaction is to be rolled back. The value for INACTIVITY-TIME must be greater than or equal to that for LOCK-TIME. The default value is 10 times the value for LOCK-TIME.

MAX-ISOLATION-LEVEL = *STD / *REPEATABLE-READ
The maximum isolation level for access to user tables is defined. If *STD is specified, access occurs under the isolation level SERIALIZABLE.

If you specified the value *REPEATABLE-READ, all attempts of applications to access data under the isolation level SERIALIZABLE are rejected with SQLSTATE in the DBH session.

If *REPEATABLE-READ is specified, no transaction locks are used by the DBH on index values, provided the index is not a unique index nor an index of a reference condition.

If an SQL statement is to work with a DBH for which the value *REPEATABLE-READ is specified for MAX-ISOLATION-LEVEL, you must either execute a SET TRANSACTION in the application before each transaction, or set the option ISOL-LEVEL=REPEATABLE-READ in the user configuration file.

LOCK-ESCALATION =
Specifies what happens when there are transaction locks. If a specified number of locked values in an index or number of locked records in a table is exceeded, SESAM/SQL attempts to lock the entire index or table. This is referred to as lock escalation.

A change to the settings for LOCK-ESCALATION can have a considerable effect on the behavior of the DBH as regards storage space requirements and the probability of transaction conflicts:

- If low values are selected, an attempt is made at a very early stage to lock the entire table or index. This saves storage space, since the locks do not have to be managed individually. However, it increases the probability of transaction conflicts occurring.
- Higher values reduce the risk of transaction conflicts. In this case, however, the storage space requirements increase because the locks have to be managed individually.

The current number of records as of which the entire table is to be locked is obtained from PERCENTAGE-RECORDS and compared with the value of NUMBER-RECORDS. The lower value is used as the limit.

If lock escalation is not successful immediately, only the directly affected record or index value is locked. After this unsuccessful escalation attempt, the next attempt is not started for another 100 locks.

LOCK-ESCALATION = *STD
The default values for the size of the escalation operands are indicated under *PARAMETERS(...).

LOCK-ESCALATION = *PARAMETERS(...)

NUMBER-RECORDS = 4000 / <integer 1..2147483647>
Specifies the number of records in a table that a user can lock. If more records than this are locked, the whole table is locked.

PERCENTAGE-RECORDS = 50 / <integer 0..100>
Specifies the percentage of all the records in the table that a user can lock. If a higher percentage is locked, an attempt is made to lock all the records of the table. If 0 is selected as the percentage, an attempt is made to lock the entire table immediately.

NUMBER-INDEX-VALUES = 1000 / <integer 1..2147483647>

Specifies the number of values that a user can lock in an index. If a larger number than this are locked, an attempt is made to lock the entire index rather than just individual values.

3.2.2.38 TRANSFER-CONTAINER

This statement sets the size of the transfer container.

Definition

A lower-level DBH option of STORAGE-SIZE

Function

When SQL access plans are processed, SQL scans are produced. These are subareas of an evaluation rule for an SQL statement. During the processing of CALL DML applications, inquiry and response areas are requested for each logical file by means of the OPEN statement.

The SESAM/SQL DBH reserves the above areas in the transfer container, where they are available for further processing.

You use the DBH option TRANSFER-CONTAINER to set the size of the transfer container. The size of the transfer container is specified in units of 1 Kbyte.

The storage space requirements for the transfer container depend on the maximum number of concurrent scans or logical files in existence and their question-and-answer requirements.

You can adjust the values of the option during the DBH session by means of the MODIFY-STORAGE-SIZE administration statement (see "MODIFY-STORAGE-SIZE").

```
TRANSFER-CONTAINER

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | INITIAL = 64 / <integer 64..1000000>

    | ,MAXIMUM = 16000 / <integer 64..1000000>
```

Operands

TRANSFER-CONTAINER = *STD

The default values for the size of the transfer container are described under *PARAMETERS(...).

TRANSFER-CONTAINER = *PARAMETERS(...)

Changes the size of the transfer container.

INITIAL = 64 / <integer 64..1000000>
Sets the size of the transfer container at the beginning of the DBH session. The default is 64 Kbytes.

> The initial value for TRANSFER-CONTAINER is compared to the initial value of SUBORDERS. This results in the following minimum value for the initial value of TRANSFER-CONTAINER:
> (initial value of SUBORDERS + x) * 4 [Kbytes]
>
> The following applies to x:
>
> - in the independent DBH:
>
> x = 1 + 2 * min(10, [SQL-DATABASE-CATALOG/4]) + 4 * ([THREADS+10]/10)

- in a linked-in DBH: x = 3

MAXIMUM = 16000 / <integer 64..1000000>
Defines the maximum size which the transfer container may reach. The default is 16000 Kbytes.

If the maximum value is smaller than the specified value for INITIAL, then the value of MAXIMUM is adjusted according to the value of INITIAL.
The maximum value for TRANSFER-CONTAINER is also compared to the maximum value of SUBORDERS in the same manner as for the initial value. This results in the following maximum value for TRANSFER-CONTAINER:

(maximum value of SUBORDERS + x) * 4 [Kbyte]

The following applies to x:

- in an independent DBH:

x = 1 + 2 * min(10, [SQL-DATABASE-CATALOG/4])

- in a linked-in DBH: x = 3

Information on the transfer container

You can obtain information on the transfer container and its extensions from the operational statistics of the SESAM /SQL monitor SESMON.
You will find information in the “SYSTEM-INFORMATION” form (see "DBH form SYSTEM INFORMATION") on the size of the container and its extensions.

The “ Performance” manual describes the cases in which it makes sense to modify the container size at the next DBH startup.

3.2.2.39 USER-DATA-BUFFER

This statement sets the size of the user-data buffer.

Definition

A lower-level DBH option of STORAGE-SIZE

Function

The SESAM/SQL DBH administers two separate buffers: one for blocks with systemaccess data (see the DBH option SYSTEM-DATA-BUFFER on "SYSTEM-DATA-BUFFER") and one for blocks with user data. The blocks are each 4 Kbytes in size. All users use both buffers together for all connected databases.

You use the DBH option USER-DATA-BUFFER to set the initial size of the buffer for user data.

The final size of the buffer for user data depends on the DBH option THREADS and the system threads depending on THREADS. The minimum value for the buffer size is 96 KB per thread. If the specified value is smaller than the minimum value, then the value is automatically increased to the minimum value as long as the specified value is within the permitted range of values (see below). Otherwise the DBH is terminated.

You can adjust the value of the option during the DBH session by means of the RECONFIGURE-DBH-SESSION administration statement (see "RECONFIGURE-DBH-SESSION").

USER-DATA-BUFFER
= 80 / <integer 80..64000000>

Operands

USER-DATA-BUFFER = 80 / <integer 80..64000000>

Initial size of the buffer for user data in units of 1 Kbyte. The default is 80 Kbytes.

The maximum value for USER-DATA-BUFFER is 64 000 000 Kbytes (64 Gbytes) on all current BS2000 servers.

> The sum of the sizes for SYSTEM-DATA-BUFFER and USER-DATA-BUFFER may not exceed 64 000 000 Kbytes (64 Gbytes).
> If the system configuration so permits, the SYSTEM-DATA-BUFFER is created in a data space. Otherwise it is created in the normal class 6 memory.

Information on input/output behavior

The “I/O” form of the SESAM/SQL monitor SESMON contains information on input/output behavior. This information helps you set the optimal size of the buffer for user data (see "DBH form I/O").

The “ Performance” manual describes the cases in which it makes sense to modify the size of the buffer for user data at the next DBH startup.

3.2.2.40 USERS

This statement specifies the number of users.

Definition

A lower-level DBH option of SYSTEM-LIMITS

Function

You use the DBH option USERS to specify how many parallel users you want to permit in the DBH session.
A user of the DBH in timesharing mode is an interactive or batch program, identified in each case by a TSN (task sequence number). In transaction mode (openUTM/DCAM), a user is defined by an active terminal or by a user ID and terminal together (in the case of openUTM, by means of the LTERM or TPOOL control statement). In openUTM with conversation stacking and asynchronous conversations, each conversation is a user of the SESAM/SQL DBH.

You can adjust the value of the option with the administration statement RELOAD-DBH-SESSION (see "RELOAD-DBH-SESSION") when the DBH modules are reloaded.

USERS
= *STD / <integer 1..32767>

Operands

USERS = *STD / <integer 1..32767>

Maximum number of concurrent users. The default value for the total number of concurrent users is 24. An internal comparison ensures that the number of available SQL scans or logical files specified using the INITIAL operand of the DBH option SUBORDERS is taken into account.

The upper limit for the number of concurrent users depends on how many users your license for the SESAM/SQL database system permits.

3.2.2.41 WORK-CONTAINER

This statement sets the size of the work container.

Definition

A lower-level DBH option of STORAGE-SIZE

Function

The SESAM/SQL DBH checks that each statement is lexically, syntactically and semantically correct. Based on a correct statement or subarea of a statement, the DBH generates an optimized format known as the internal statement format.

Based on a CALL DML statement, the DBH generates the internal statement format directly. Based on an SQL statement, the DBH initially generates an SQL access plan, i.e. an evaluation rule for the SQL statement. An SQL access plan consists of at least one but usually several subareas known as SQL scans. The optimized format of a scan forms the internal statement format.

The internal statement formats are stored for further processing in the work container so that follow-up statements can refer back to them.

The DBH option WORK-CONTAINER allows you to adjust the size of the work container to the requirements of the application. The size of the work container is specified in units of 1 Kbyte.

You can adjust the values of the option during the DBH session by means of the MODIFY-STORAGE-SIZE administration statement (see "MODIFY-STORAGE-SIZE").

```
WORK-CONTAINER

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | INITIAL = *STD / <integer 24..1000000>

    | ,MAXIMUM = *STD / <integer 24..1000000>
```

Operands

WORK-CONTAINER = *STD

The default value for the size of the work container is described under *PARAMETERS(...).

WORK-CONTAINER = *PARAMETERS(...)
Changes the size of the work container.

INITIAL = *STD / <integer 24..1000000>
Initial size of the work container in Kbytes. DBH carries out an internal comparison with the initial value of SUBORDERS when an initial value is specified:

Minimum value (or default value if no value was specified explicitly) for INITIAL for WORK-CONTAINER:

(Initial value of SUBORDERS + x) * 4 Kbyte

The additional amount x is calculated in the same manner as the additional amount for the initial value of TRANSFER-CONTAINER (see "TRANSFER-CONTAINER").

MAXIMUM = *STD / <integer 24..1000000>
Maximum size of the work container in Kbytes. DBH carries out an internal comparison with the maximum value of SUBORDERS when a maximum value is specified:

Minimum value (or default value if no value was specified explicitly) for MAXIMUM for WORK-CONTAINER:

(Maximum value of SUBORDERS+ x) * 4 Kbyte

The additional amount x is calculated in the same manner as the additional amount for the maximum value of TRANSFER-CONTAINER (see "TRANSFER-CONTAINER").

Space required to store the internal statement format in the WORK-CONTAINER

A resource bottleneck can be prevented for WORK-CONTAINER if its maximum size is the product of the maximum number of SUBORDERS and the space required to store an average internal statement format. The space required to store an average internal statement format can be derived from the following table. It depends on the DBH option COLUMNS (see "COLUMNS"f).

Average number of COLUMNS in the internal statement format	Space required to store internal statement formats in Kbytes
up to 1	12
up to 28	16
up to 63	20
up to 93	24
up to 122	28
up to 151	32
up to 210	40
up to 268	48
up to 327	56
up to 385	64
up to 438	72
up to 502	80
up to 561	88
up to 619	96
up to 736	112
up to 853	128
up to 970	144
up to 1024	152

Table 10: Space required to store internal statement formats

Information on the work container

You can obtain information on the work container and its extensions from the operational statistics of the SESAM /SQL monitor SESMON.
The “SYSTEM-INFORMATION” form indicates the size of the container, the percentage of its capacity being used, the number of area requests made, and the number of times the container has been accessed (see "DBH form SYSTEM INFORMATION").

The “ Performance” manual describes the cases in which it makes sense to modify the container size at the next DBH startup.

4 Distributed processing with SESAM/SQL-DCN

SESAM/SQL-DCN is a chargeable add-on product for the SESAM/SQL-Server database system.

In distributed processing with SESAM/SQL-DCN, an application program can work with more than one SESAM /SQL DBH during a single session. The interaction between the application program and the various DBHs can take place within the same configuration, across configurations on the same machine, or across different machines.

The central component of the product SESAM/SQL-DCN is the distribution component SESDCN. This component must be started in all configurations involved in distributed processing.

The principles of distributed processing are described in the “ Core manual”.

This chapter describes how to start, parametrize and terminate SESDCN. It covers the following:

- Starting the SESDCN distribution component
- SESDCN control statements
- Terminating SESDCN
- SESDCN restart
- Maximum values for working with SESAM/SQL-DCN

The dynamic administration of SESDCN is described in chapter “DBH and SESDCN administration”.

If the add-on product SESAM/SQL-DCN is not available, all SESDCN-specific commands and statements are rejected.

4.1 Starting the SESDCN distribution component

Like the independent DBH, SESDCN runs in BS2000 as a separate task and should preferably be started as a batch job.

The execution phase of SESDCN exists as a link and load module (LLM) called SESDCN and is started with the START-SESAM-DCN command (see section “Starting SESAM/SQL programs via start commands”).

The SESDCN distribution component is parametrized at startup by means of SESDCN control statements. These contain the DCN options and the distribution rule for SESDCN. The SESDCN control statements are not described in this section. You will find a detailed description of them in a separate section of this chapter (see section “SESDCN control statements”).

SESAM/SQL-DCN share modules

You can administer parts of SESAM/SQL-DCN as share modules and load them on a shareable basis or as subsystems. You will find information on this in section “SESAM/SQL share modules”.

4.1.1 Requirements for starting

General requirements

The prerequisites for starting SESDCN are similar to those for starting the SESAM/SQL DBH (see "Requirements for starting"). The section on the sequence of commands for starting SESDCN indicates the files you have to activate and the assignments that are necessary before SESDCN is started (see "Starting SESDCN in batch mode and passing control statements").

In addition, the application programs involved in distribution must not yet be loaded, since only application programs loaded after SESDCN is started can access data in the network.

The prerequisites for restarting SESDCN are described in section “SESDCN restart requirements”.

4.1.2 Starting SESDCN in batch mode and passing control statements

In batch mode, you start SESDCN as a batch job with the command

```
/ENTER-PROCEDURE FROM-FILE=filename
```

where *filename* is the name of an SDF-P procedure created by the user in which he or she enters the sequence of commands for starting SESDCN.

Sequence of commands for starting SESMON

```
[/MODIFY-TEST-OPTIONS DUMP=YES]       1.
[/ASSIGN-SYSDTA TO-FILE = *SYSCMD]    2.
/START-SESAM-DCN                      3.
[//SESDCN control statements]         4.
[//END]                               5.
```

1. Causes each requested memory dump to be output and a message issued to SYSOUT. You should enter this statement in order to receive the diagnostic documentation required in the event of an error.
2. Assigns the SYSDTA system file to the SYSCMD procedure file. This assignment is not necessary if you enter the SESDCN control statements in an S procedure.
3. Starts the SESDCN distribution component with the start command START-SESAM-DCN (see also section “Starting SESAM/SQL programs via start commands”).
4. You must enter the SESDCN control statements here (see section “SESDCN control statements” on "Entering SESDCN control statements"). You enter them if you do not want to use the defaults or do not assign the control statements via an input file.
5. The END statement concludes input of the SESDCN control statements.

Entering the SESDCN control statements

There are two ways to enter the SESDCN control statements:

1. You can enter the control statements in the procedure in the sequence of commands for starting SESDCN. In this case, you must assign the SYSDTA system file to SYSCMD before the START-SESAM-DCN command. The start parameters must come immediately after the start command START-PROGRAM.
2. You can pass the control statements via an input file (see section “Passing control statements by means of an input file”).

In both cases, you must conclude input of the control statements with the END statement. In the event of an invalid control statement, SESDCN aborts. A distribution rule is not generated if this happens.

4.1.3 Starting SESDCN in interactive mode and passing control statements

When you start the SESDCN distribution component in the interactive mode, the terminal remains busy throughout the SESDCN session. You can only start other programs from this terminal by means of batch jobs. To avoid interrupting the SESDCN session, you should therefore start SESDCN in batch mode.

You can start SESDCN for the interactive mode either directly on screen or by means of a procedure; accordingly, you enter the SESDCN control statements either directly on screen or in the procedure. The control statements must come immediately after the start command START-SESAM-DCN.

Alternatively, you can enter the SESDCN control statements via an input file (see "Passing control statements by means of an input file").

Parameter input is logged to SYSLST. You can specify the extent of logging by means of the MODIFY-SDF-OPTIONS statement's LOGGING parameter (see the BS2000 manual “ Commands”). It is advisable to specify LOGGING=ACCEPTED-FORM here so that only the entered start parameters are logged, not the default values.

The sequence of commands for starting SESDCN is described on "Starting SESDCN in batch mode and passing control statements". You will find information on entering the SESDCN control statements on "Passing control statements by means of an input file".

Starting SESDCN and entering control statements interactively

When you start SESDCN directly on screen in the interactive mode, the SYSDTA system file must be assigned to the terminal.

Once the START-SESAM-DCN command has been entered, SESDCN requests the control statements interactively.

If one of the control statements is invalid, you receive an error message. You can then correct it. The corrected version of the control statement is not logged to SYSLST.

Starting SESDCN and entering start statements in a procedure

When you start and parametrize SESDCN in a procedure, you must assign the SYSDTA system file to the SYSCMD procedure file:

Example

In the interactive mode, you enter the SESDCN control statements in an SESDCN start procedure.

```
/SET-PROCEDURE-OPTIONS
...
/START-SESAM-DCN
//SESDCN control statements
...
//END
...
/EXIT-PROCEDURE
```

4.1.4 Passing control statements by means of an input file

You can pass the SESDCN control statements to SESDCN via an input file in either the batch or interactive mode.

Before starting SESDCN, you must assign the input file. There are several ways to do this. You can assign the input file:

- as a SESDCN configuration file using the link name SESCONF
- as a SYSDTA system file
- as a global configuration file using the CONNECT-SESAM-CONFIGURATION command.

Assigning a SESDCN configuration file

You assign the SESDCN configuration file with the following command:

```
ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=filename
```

Assigning a SYSDTA system file as the input file

```
ASSIGN-SYSDTA TO-FILE=filename
```

In section “Examples of SESDCN parametrization”, you will find some examples of SESDCN input files.

Assigning a global configuration file

This global configuration file can contain configuration parameters for several components of the SESAM/SQL system (see "Entering the DBH start parameters for the independent DBH").

Notes on entering control statements

When entering SESDCN control statements, you must comply with a number of rules (which apply regardless of whether you are entering them in an input file, a procedure or interactively):

- Each line must begin with the escape symbol (//).
- Each control statement must begin on a new line.
- If a control statement takes up more than one line, you must conclude each line with the continuation character (-); the last line of a control statement must not be concluded with the continuation character (-).
- The parameters must be entered in a fixed order (see "SESDCN control statements").
- The last line must contain an END statement; this END statement concludes input of the SESDCN control statements.

If one of the SESDCN control statements is invalid, SESDCN aborts and you receive an error message. A distribution rule is not generated.

4.1.5 Starting several SESDCNs

In most cases, the SESDCN distribution component need only be loaded once per configuration. However, a large number of remote accesses of an SESDCN can have a negative effect on runtime behavior.
You can increase total throughput of remote accesses by starting several SESDCNs. Remote requests are then processed concurrently by more than one SESDCN.

A DBH may only ever be assigned to one SESDCN.
(A DBH can occur in several entries of the ADD-DISTRIBUTION-RULE-LIST statement if this DBH has several catalogs. The same LINK-NAME must be used in all these DBH entries and then describes the route via a DCN in the ADD-NETWORK-LINK-LIST statement. If this condition is not met, the statement is rejected with message SEN1003.)

Several DBHs can be assigned to a single SESDCN.

The SESDCN started first in a configuration assumes the function of the master DCN. When necessary, the master DCN carries out the SESDCN restart.

When starting the master DCN, you must define the distribution rule by means of the appropriate control statements. You can start the other SESDCNs in a configuration without defining the distribution rule, i.e. without supplying information on databases.

4.2 Terminating SESDCN

You terminate SESDCN with the STOP-DCN administration statement (see "STOP-DCN").

An SESDCN that has been terminated can no longer act as a remote DCN. Remote accessing of databases to which the SESDCN is assigned by the distribution rule as the remote DCN is no longer possible.

The loaded distribution rule is retained until all SESDCNs, DBHs and application programs belonging to the same configuration are terminated.

4.3 Restarting an SESDCN session

After a system crash, SESDCN must be restarted to preserve transaction consistency (see the “ Core manual”). SESDCN is responsible for coordinating, executing and monitoring the restart. If SESDCN has been loaded several times in the configuration, the SESDCN loaded first assumes this task. The SESDCN that carries out the restart is called the master DCN.

A SESDCN restart is the logical continuation of the previous SESDCN session. All the DCN options and distribution-rule entries of the previous session therefore apply.

4.3.1 SESDCN restart requirements

If an SESDCN restart is to be carried out, the following requirements must be met:

To ensure that the master DCN can access the backup file of the aborted session during a restart, you must make the backup file available before starting the master DCN. There are two ways to do this:

- You can catalog the backup file with its default name under the identifier where the master DCN is started. The default name of the backup file is SES.DLG*c*, where *c* stands for the name of the configuration to which the master DCN belongs.
- You can assign the backup file to the master DCN under the link name SESDLG.

To ensure that the restart can be carried out for all configurations on a backup computer, the configuration names must be unique throughout the network.
You assign the configuration name by means of the DCN option DCN-IDENTIFICATION (see section “SET-DCN-OPTIONS”).

4.3.2 Restarting SESDCN on a backup computer

SESDCN can be restarted on the computer on which SESDCN was originally started (the cold-start computer) or on a different computer, the backup computer.

To restart an SESDCN on a backup computer, the whole SESDCN configuration must be on the restart computer. You must therefore move all the components of this configuration to the backup computer. These can be:

- application programs
- SESDCNs
- DBHs
- the SES.DLG*c* backup file
- DBH-specific files

The DBHs involved must be able to access all the databases assigned to them. All the required tasks must be restartable on the new computer.

Updating the distribution rules

If the SESDCN restart is executed on a backup computer, the location of the restart configuration in the network changes for the other configurations. The master DCN of the restart configuration therefore automatically replaces the name of the original computer with the name of the backup computer in the distribution rules of the remote computers known and accessible to it. This ensures that communication is still possible with the configurations on these machines.

If not all the remote computers in the network are accessible to the master DCN at the time of the SESDCN restart, the master DCN cannot change all the distribution rules automatically. In this case, you have to update the distribution-rule entries yourself using the MODIFY-DISTRIBUTION-RULE-ENTRY administration statement (see "MODIFY-DISTRIBUTION-RULE-ENTRY").

If several configurations execute their SESDCN restart on a backup computer after a crash, the associated master DCNs again cannot update all the distribution rules. They can update the distribution rules of configurations that have remained in the same position. They are accessible, but not those of configurations that have moved to another computer. In this case too, you have to update some of the distribution-rule entries manually using the MODIFY-DISTRIBUTION-RULE-ENTRY administration statement.

The master DCN updates the distribution rules only in the memory pool and the log file. Any input files that exist are not affected.

4.3.3 Synchronization of DBH, SESDCN and UTM restarts

To guarantee an error-free restart of a distributed database system in a UTM environment, all the systems involved are started with restart data (warm start).
A cold start of one of the systems after a system failure is not permitted and usually not possible. It can be triggered by deleting the backup files, for example.

The following table indicates what happens if one of the systems involved is cold started:

Start of	Coldstart	Warm start	Consequence
DBH	x		The UTM application is aborted.
SESDCN	x		The UTM application is aborted.
openUTM		x	The UTM application is aborted.
DBH		x	Restart information for openUTM is lost as a result of the cold start; data consistency is not guaranteed.
SESDCN		x	Restart information for openUTM is lost as a result of the cold start; data consistency is not guaranteed.
openUTM	x		Restart information for openUTM is lost as a result of the cold start; data consistency is not guaranteed.
DBH		x	Transactions in the state PTC are not known to the DBH and can be rolled back by means of administration statements. The session can be continued, but data consistency is not guaranteed.
SESDCN	x		Transactions in the state PTC are not known to the DBH and can be rolled back by means of administration statements. The session can be continued, but data consistency is not guaranteed.
DBH	x		The databases are not consistent and must be repaired.

Table 11: Consequences of a DBH, SESDCN or openUTM restart

Start sequence

The systems involved in a restart forward information on their current processing status to their partners. On the basis of this information the transactions affected by the system failure are rolled back or terminated. To guarantee complete consistency, you must start the systems in the following order:

1. DBHs
2. SESDCNs
3. UTM applications
4. other application programs

If you start a UTM application before the DBH or before SESDCN, it is aborted.

4.4 SESDCN control statements

SESDCN control statements are for parametrizing the SESDCN distribution component. They contain the options and the definition of the distribution rule. The distribution rule defines the associated access path for every database involved in the distributed system.

SESDCN uses the BS2000 dialog interface SDF (**S**ystem **D**ialog **F**acility). SDF supports statement input in a form-driven dialog, analyzes the syntax of the statements entered and passes them to SESDCN for processing. You will find a detailed description of the SDF dialog interface in the “ SDF Dialog Interface”.

4.4.1 Entering SESDCN control statements

You enter the SESDCN control statements when you start SESDCN (see section “Starting the SESDCN distribution component”).

The following table provides an overview of the SESDCN control statements in the order in which they are entered:

SESDCN control statement	Function
SET-DCN-OPTIONS	Causes the options to be read in.
ADD-DISTRIBUTION-RULE-LIST	Defines the distribution rule.
ADD-NETWORK-LINK-LIST	Defines the distribution rule.

Table 12: SESDCN control statements

The first SESDCN control statement is SET-DCN-OPTIONS (see "SET-DCN-OPTIONS"). You enter this statement once and once only.

It is followed by the ADD-DISTRIBUTION-RULE-LIST control statement (see "ADD-DISTRIBUTION-RULE-LIST"). You use this statement to enter the databases to be included in the distributed system. You must enter this statement at least once for the master DCN. The master DCN is the first SESDCN started in a configuration. If you start additional SESDCNs in the configuration of the master DCN, you need not enter the ADD-DISTRIBUTION-RULE-LIST statement for these SESDCNs.
You can enter the required databases using a single statement, or you can enter them in groups using several ADD-DISTRIBUTION-RULE-LIST statements.

Finally, you can enter the ADD-NETWORK-LINK-LIST statement (see "ADD-NETWORK-LINK-LIST"). You use this to assign the logical link names of the databases to their respective physical access paths, thus defining the network for the distribution component.

You can do this using a single statement, or you can enter the databases in groups using several ADD-NETWORK-LINK-LIST statements.

You need not enter the ADD-NETWORK-LINK-LIST statement if the ADD-DISTRIBUTION-RULE-LIST statement contains only databases stored on the home system that belong to the same configuration as the SESDCN to be started.

> If databases or CALL-DML tables that are not included in the distribution rule are addressed during ongoing operation, these are searched for in the DBH whose NAM identifier is specified in the current configuration file. In this case SESAM/SQL implicitly creates an entry in the distribution rule for each DBH addressed in this way.
> Together with the entries created implicitly by SESAM/SQL the distribution rule may contain a maximum of 340 entries. You can, however, explicitly enter at least 300 databases and their associated networks.

END statement

The END statement concludes input of SESDCN control statements.

Example 1

The END statement concludes input of the SESDCN control statements after the information on the structure of the distribution rule.

```
//SET-DCN-OPTIONS ...
//ADD-DISTRIBUTION-RULE-LIST ...
//ADD-NETWORK-LINK-LIST ...
//END
```

Example 2

The END statement concludes parameter input immediately after the SET-DCN-OPTIONS statement.

```
//SET-DCN-OPTIONS ...

//END
```

4.4.2 Syntax and functions of the SESDCN control statements

This section describes the SESDCN control statements in the order in which they are entered.

The syntax of the SESDCN control statements corresponds to that of SDF (see the “ Commands” manual).

The SESDCN control statements can be abbreviated in accordance with the SDF conventions.

Most of the parameters of the SESDCN control statements have defaults. When entering the SESDCN control statements, you need only include those parameters to which you want to assign values other than the defaults.

4.4.2.1 SET-DCN-OPTIONS

Reading in DCN options

Function

The SESDCN control statement SET-DCN-OPTIONS causes the DCN options to be read in.

DCN options parametrize the SESDCN distribution component and thus define the essential properties of SESDCN operation. SESAM/SQL-DCN recognizes the following options:

DCN-OPTION	Short description of function
ADMINISTRATOR	Assigns administrator authorization
COLDSTART	Requests a cold start.
DCN-IDENTIFICATION	Assigns a DCN name and configuration name.
REMOTE-ACCESS	Permits access from a remote computer.
SESDLG-PASSWORD	Assigns a password.
SYSTEM-LIMITS	Specifies limit values for the permitted number of users and transaction applications, and defines reset criteria for transactions.

Table 13: Functions of the DCN options

You can modify some DCN options dynamically using administration statements (see section “SESDCN administration statements”).

You must enter the SET-DCN-OPTIONS statement once and once only.

```
SET-DCN-OPTIONS
DCN-IDENTIFICATION = *STD / *PARAMETERS(...)
    *PARAMETERS(...)
        |   CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
        |   ,DCN-NAME = *BLANK / <alphanum-name 1..1>
,SYSTEM-LIMITS = *STD / *PARAMETERS(...)
    *PARAMETERS(...)
        |   USERS = 128 / <integer 1..16000>
        |   ,APPLICATIONS = 64 / <integer 1..128>
        |   ,LOCK-TIME = 8 / <integer 1..999>
        |   ,INACTIVITY-TIME = 12 / <integer 1..999>
,COLDSTART = *NO / *YES
,REMOTE-ACCESS = *ALLOWED / *NOT-ALLOWED
,SESDLG-PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
```

,ADMINISTRATOR = *NONE / *ANY(...) / *TIME-SHARING-USER(...) / *APPLICATION-USER(...)

*ANY(...)

| PASSWORD = <c-string 3..3> / <x-string 5..6>

*TIME-SHARING-USER(...)

| PASSWORD = <c-string 3..3> / <x-string 5..6>

| ,HOST-NAME = <name 1..8>

| ,USER-ID = <name 1..8>

*APPLICATION-USER(...)

| PASSWORD = <c-string 3..3> / <x-string 5..6>

| ,HOST-NAME = <text 1..8>

| ,APPLICATION-NAME = <text 1..8>

| ,CUSTOMER-NAME = <text 1..8>

Operands

DCN-IDENTIFICATION =

Identifies the SESDCN distribution component to be parametrized.

DCN-IDENTIFICATION = *STD
The SESDCN distribution component is identified by default values. These are described under PARAMETERS(...).

DCN-IDENTIFICATION = *PARAMETERS(...)

CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>

Name of the configuration to which the SESDCN component is assigned.The configuration name must be unique throughout the network so that, if necessary, it is possible to execute a restart on a machine other than the cold-start computer (see "Restarting SESDCN on a backup computer").

DCN-NAME = *BLANK / <alphanum-name 1..1>
Name of the distribution component to be parametrized.
The default is a blank.

SYSTEM-LIMITS =
Specifies limit values for the SESDCN session.

SYSTEM-LIMITS = *STD
The default limit values of the SESDCN session apply. These are described under PARAMETERS(...).

SYSTEM-LIMITS = *PARAMETERS(...)

USERS = 128 / <integer 1..16000>
Maximum number of permitted users in the session.
In timesharing mode, the users are the interactive or batch programs that run simultaneously. In transaction mode, they are the simultaneously open transaction conversations defined by the active terminals.

The default value for USERS is 128. The upper limit for the number of concurrent users depends on the total number of users your SESAM/SQL-DCN license permits. The license specifies the total number of DCN option USERS of all master DCNs in a hardware installation.

You cannot change the number of permitted users specified here in the current session; you first have to restart SESDCN. Note that you cannot restart SESDCN until all the application programs involved and all the SESAM/SQL-DBHs and SESDCNs have been terminated.

SESDCN sets up the number of areas specified by USERS for user tables in the common memory pool. The common memory pool is created by the first SESDCN in a configuration to be loaded, the master DCN. The entry for USERS is ignored by subsequently loaded SESDCNs of the same configuration.
You should therefore assign the same value for USERS to all master DCNs in the session.

APPLICATIONS = 64 / <integer 1..128>
Maximum number of transaction applications that can participate in the SESDCN session. The default is 64.

LOCK-TIME = 8 / <integer 1..999>

Specifies the time in minutes after which an inactive transaction locking another transaction is rolled back. Default: 8 minutes.

INACTIVITY-TIME = 12 / <integer 1..999>
Specifies the time in minutes after which an inactive open transaction is rolled back. Transactions locked by other transactions are not affected by this. The value for INACTIVITY-TIME must be greater than or equal to that for LOCK-TIME. Default: 12 minutes.

COLDSTART =
Requests a cold start.

COLDSTART = *NO
This is the default.
When a restart is executed on a backup computer (see "Restarting SESDCN on a backup computer"), the host names in the distribution rules of the remote DCNs are updated only in the memory pool and in DCN LOG. Any input files of the remote DCNs are therefore not affected by this updating. The next cold start would cancel the updating.
The default is therefore that a warm start is executed when the SESDCN acting as the remote DCN is started again after a distribution rule has been updated (because a SESDCN has been restarted on a backup computer). When a warm start is executed, SESDCN evaluates the updated distribution rule backed up in DCN LOG.

COLDSTART = *YES
Causes a cold start to be executed even if the distribution rule of this SESDCN has been updated (because of a SESDCN restart on a backup computer). When a cold start is executed, SESDCN evaluates the SESDCN control statements in the input or procedure files. Any entries in DCN LOG that contradict these are ignored.
You cannot force a cold start if there are still transactions in the PTC state. If this is the case, SESDCN carries out a warm start.
Note that the host names in the ADD-NETWORK-LINK-LIST statement must be up to date.

REMOTE-ACCESS =
Specifies whether or not access is permitted from a remote computer.

REMOTE-ACCESS = *ALLOWED
This is the default. Access from a remote computer is permitted.

REMOTE-ACCESS = *NOT-ALLOWED
Remote access is not permitted. SESDCN does not accept requests from remote computers.

SESDLG-PASSWORD =

Gives the distribution components the password with which you can access the SESDCN backup file. The password protection applies to write and read accesses. At the same time, this also protects the file from being deleted.If there is not a SESDCN backup file at the time when the SESDCN session is started, the file is generated by SESDCN and is password-protected if a password is specified in the SESDLG-PASSWORD operand.

SESDLG-PASSWORD = *NONE
No password is specified.

SESDLG-PASSWORD = <c-string 1..4> / <x-string 1..8> /
<integer -2147483648..2147483647>
Gives the specified password to the distribution components.

> Password protection is only set by SESDCN when generating the SESDLG file.
> If there is already a SESDCN backup file without password protection at the time when SESDCN is started under the name assigned with the link name SESDLG or under the default name, then this file will not be protected by a password from SESDCN even when the SESDLG-PASSWORD operand contains a value.
>
> If an existing SESDCN backup file is assigned a password later or a password is to be changed, you must do this using the SDF command MODIFY-FILE-ATTRIBUTES.

ADMINISTRATOR =
Specifies the user or user group authorized to issue administration statements using the CALL interface and SESADM.

ADMINISTRATOR = *NONE
Administration using the CALL interface and SESADM is not possible. The MODIFY-ADMINISTRATION administration statement is also rejected.

ADMINISTRATOR = *ANY(...)
Users who are not system administrators can also issue administration statements using the CALL interface and SESADM.

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

ADMINISTRATOR = *TIME-SHARING-USER(...)

Only the system administrator can issue administration statements using the CALL interface and SESADM. The administrator is a timesharing user and is identified by the system user identification.

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

HOST-NAME = <name 1..8>
Name of the computer from which administration is to be carried out

USER-ID = <name 1..8>
User ID of the system administrator

ADMINISTRATOR = *APPLICATION-USER(...)
Only the system administrator can issue administration statements using the CALL interface and SESADM. The system administrator is an application user and is identified by the system user identification.

PASSWORD = <c-string 3..3> / <x-string 5..6>
Password providing protection against unauthorized administration using the CALL interface and SESADM

HOST-NAME = <text 1..8>
Name of the computer from which administration is to be carried out

APPLICATION-NAME = <text 1..8>
Name of the application from which administration is to be carried out

CUSTOMER-NAME = <text 1..8>
Name of the user. If the system administrator is working under openUTM, the KDCSIGN name must be specified here. If the system administrator is working under DCAM, the name by which the system administrator identifies himself or herself at the program interface must be specified.

4.4.2.2 ADD-DISTRIBUTION-RULE-LIST

This statement adds databases to the distribution rule.

Function

The SESDCN control statement ADD-DISTRIBUTION-RULE-LIST describes which databases are added to the distribution rule and via which logical connections the different databases can be accessed.

You must use this statement to enter all databases that are to be involved in distributed processing and accessible from the configuration to which the SESDCN to be loaded belongs. This also applies to base tables processed by CALL DML application programs.

You can use the statement once or several times. Together with the entries created implicitly by SESAM/SQL (see the note on "Entering SESDCN control statements") the distribution rule may contain a maximum of 340 entries. You can, however, explicitly enter at least 300 databases and their associated networks.

You can use the ADD-DISTRIBUTION-RULE-ENTRY administration statement to enter additional databases in the distribution rule during the SESDCN session (see "ADD-DISTRIBUTION-RULE-ENTRY"). The administration statements MODIFY-DISTRIBUTION-RULE-ENTRY and REMOVE-DISTRIBUTION-RULE-ENTRY allow you to change entries in the distribution rule and remove them from it, respectively (see "MODIFY-DISTRIBUTION-RULE-ENTRY" and "REMOVE-DISTRIBUTION-RULE-ENTRY").

```
ADD-DISTRIBUTION-RULE-LIST

 CATALOG-NAME-1 = *NONE / <filename 1..18 without-cat-gen-vers>(...)

      |     LINK-NAME = *HOME / <name 1..8>

      |     ,DBH-NAME = *BLANK / <alphanum-name 1..1>

 .

 .

 ,CATALOG-NAME-200 = *NONE / <filename 1..18 without-cat-gen-vers>(...)

      |     LINK-NAME = *HOME / <name 1..8>

      |     ,DBH-NAME = *BLANK / <alphanum-name 1..1>
```

Operands

CATALOG-NAME-1 =

First database entered in the distribution rule.

CATALOG-NAME-1 = *NONE
This is the default.
No more databases are entered.

CATALOG-NAME-1 = <filename 1..18 without-cat-gen-vers>(...)
Logical name of the database to be entered.
The logical database name is the name by which a database is addressed in the application program.

You also have to specify the logical database name for each base table to be processed by CALL DML application programs.
The logical database name must be unique throughout the network.

LINK-NAME = *HOME / <name 1..8>
Logical link name that provides the logical link to the associated entry in the ADD-NETWORK-LINK-LIST statement (see "ADD-NETWORK-LINK-LIST").
If the database to be entered is assigned to the same configuration to which SESDCN is to belong, select the default: *HOME. In this case the link to a remote DCN is unnecessary. A corresponding entry in the ADD-NETWORK-LINK-LIST statement is not required.

DBH-NAME = *BLANK / <alphanum-name 1..1>
Name of the DBH responsible for working with the database.
The default is a blank.
.

CATALOG-NAME *-n* =
n = 2-200
nth database entered in the distribution rule.
Up to 200 database entries are permitted.
The parameters of this operand are identical to those of the CATALOG-NAME-1 operand described above.

4.4.2.3 ADD-NETWORK-LINK-LIST

This statement describes the physical access path for distributed databases.

Function

The SESDCN control statement ADD-NETWORK-LINK-LIST describes the network for the databases entered in the distribution rule. It assigns the access paths in the network to the logical link names of the databases.

You use this statement to define the access path for all databases entered in the distribution rule that are not in the home configuration. You specify the computer on which the relevant databases are stored, which remote DCN is responsible for forwarding requests in the case of remote access, and which configuration the remote DCN belongs to.

A corresponding entry in the ADD-NETWORK-LINK-LIST statement is not necessary for databases in the home configuration.

You can enter the ADD-NETWORK-LINK-LIST statement once or several times. Together with the entries created implicitly by SESAM/SQL (see the note on "Entering SESDCN control statements") the distribution rule may contain a maximum of 340 entries. You can, however, explicitly enter at least 300 databases and their associated networks.

```
ADD-NETWORK-LINK-LIST

LINK-NAME-1 = *NONE / <name 1..8>(...)
   <name 1..8>(...)
      | PROCESSOR-NAME = <name 1..8>
      | ,CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
      | ,DCN-NAME = *BLANK / <alphanum-name 1..1>
.
.
, LINK-NAME-200 = *NONE / <name 1..8>(...)
   <name 1..8>(...)
      | PROCESSOR-NAME = <name 1..8>
      | ,CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
      | ,DCN-NAME = *BLANK / <alphanum-name 1..1>
```

Operands

LINK-NAME-1 =

First entry of the network access path of a database in the distribution rule. A network access path must be entered for all databases that are not in the home configuration. The home configuration is the configuration in which this SESDCN is started.The network access path consists of the name of the computer on which the database is stored and the configuration name and DCN name of the associated remote DCN. The remote DCN belongs to the same configuration to which the database is assigned. It forwards remote accesses of the database, i.e. accesses from application programs belonging to a different configuration: the configuration of the SESDCN to be loaded, for example.

LINK-NAME-1 = *NONE
This is the default. No other network access path is entered.

LINK-NAME-1 = <name 1..8>(...)
Logical link name that provides the logical link to the associated entry or entries in the ADD-DISTRIBUTION-RULE-LIST statement (see "ADD-DISTRIBUTION-RULE-LIST").
The logical link name specifies for which database in the distribution rule the network access path is to be entered. This name provides the link to the associated remote DCN.

PROCESSOR-NAME = <name 1..8>
Symbolic device name of the computer on which the database is stored and on which the associated remote DCN must be loaded.

CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
Configuration name of the configuration to which the remote DCN is assigned. The default is a blank.

DCN-NAME = *BLANK / <alphanum-name 1..1>
Name of the remote DCN.
The default is a blank.

.

LINK-NAME-*n* =
n = 2-200
nth entry of the network access path of a database in the distribution rule. Up to 200 entries are permitted.
The parameters of this operand are identical to those of the LINK-NAME-1 operand described above.

4.4.3 Examples of SESDCN parametrization

This section describes two examples of the parametrization of SESDCN by means of control statements. In the first, the communication between application programs and DBHs is to be implemented for local access, whereas in the second, it is to be implemented for remote access as well. The control statements for parametrizing SESDCN are stored in an input file that must be assigned before SESDCN is started.

Local access

In distributed processing with SESAM/SQL-DCN, application programs can communicate with more than one DBH. If the application program and database belong to the same configuration, we talk about local access (see the “Core manual”).

The figure 2 shows a simple example of local access. The two loaded DBHs on processor P412 with their assigned databases belong to configuration S. To enable application programs belonging to this configuration to access the databases via these two DBHs, the SESDCN distribution component with the DCN name R and the configuration name S is loaded.

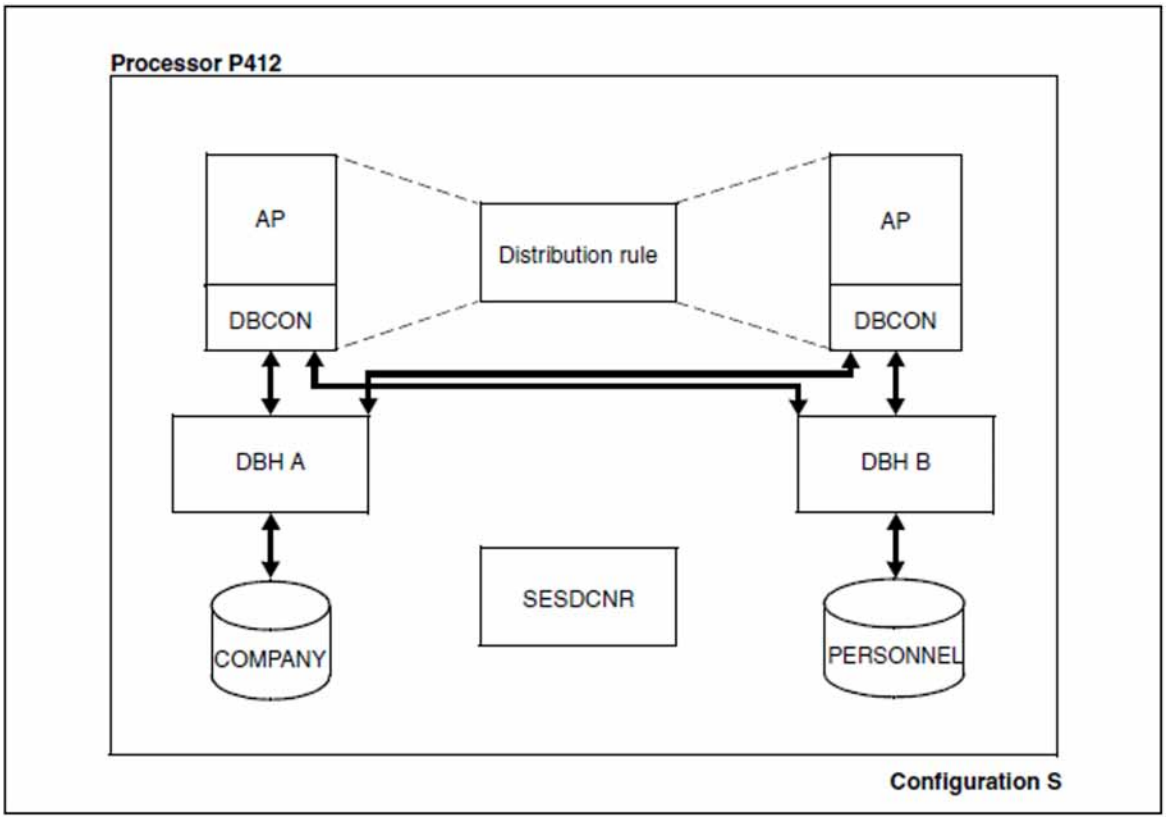


Figure 2: Example of a distributed application with local access

The input file for the control statements of the SESDCNR distribution component contains the following:

```
//SET-DCN-OPTIONS -
// DCN-IDENTIFICATION=*PARAMETERS -
// (CONFIGURATION-NAME=S,DCN-NAME=R)
//ADD-DISTRIBUTION-RULE-LIST -
// CATALOG-NAME-1=COMPANY(LINK-NAME=*HOME,DBH-NAME=A), -
```

```
// CATALOG-NAME-2=STAFF(LINK-NAME=*HOME,DBH-NAME=B)
//END
```

An ADD-NETWORK-LINK-LIST statement is not necessary, since all the databases entered in the distribution rule are assigned to the same configuration.

Remote access

In distributed processing, if application programs access databases in other configurations, we talk about remote access. Remote access can take place on a single computer or across computer boundaries.

The figure 3 shows a simple example of a distributed application with remote access. An SESDCN and a DBH are loaded in configuration A on processor P614. Processor P616 contains two configurations (B and C), each with an SESDCN and a DBH. Application programs are to be able to access all databases in the network.

The names of the configurations are unique throughout the network so that, in the event of a computer failure, a restart is possible on the other computer at any time (see section “Restarting an SESDCN session”).

For the sake of clarity, only the access paths going from configuration A to configurations B and C are shown in the following figure.

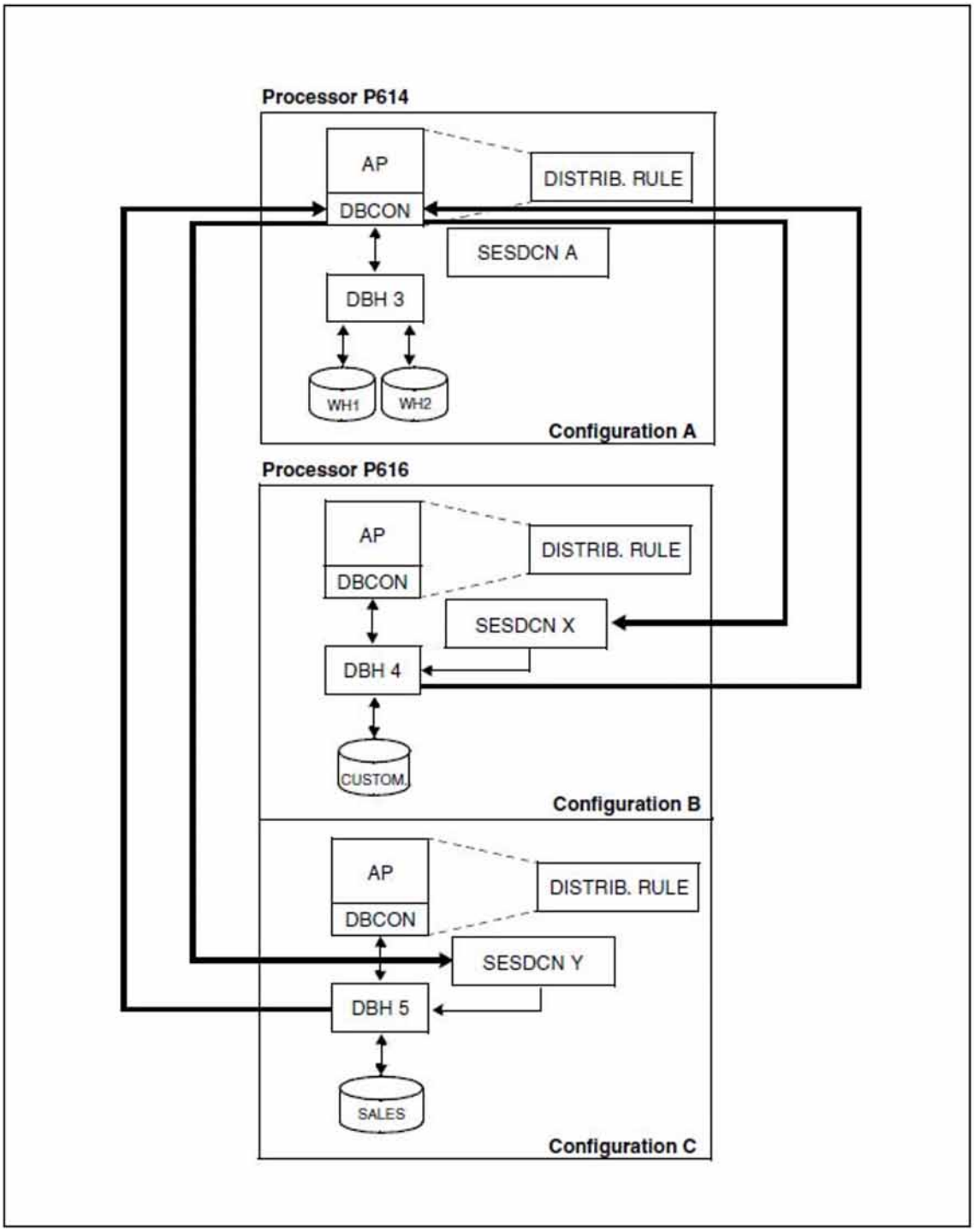


Figure 3: Example of a distributed application with remote access

The input file for the control statements of the SESDCNA distribution component contains the following:

```
//SET-DCN-OPTIONS -
// DCN-IDENTIFICATION=*PARAMETERS -
// (CONFIGURATION-NAME=A,DCN-NAME=A)-
//ADD-DISTRIBUTION-RULE-LIST -
// CATALOG-NAME-1=WAREHOUSE1(LINK-NAME=*HOME,DBH-NAME=3), -
// CATALOG-NAME-2=WAREHOUSE2(LINK-NAME=*HOME,DBH-NAME=3), -
```

```
// CATALOG-NAME-3=CUSTOMERS(LINK-NAME=PARTNER1,DBH-NAME=4), -
// CATALOG-NAME-4=SALES(LINK-NAME=PARTNER2,DBH-NAME=5)
//ADD-NETWORK-LINK-LIST -
// LINK-NAME1=PARTNER1(PROCESSOR-NAME=P616, -
//                     CONFIGURATION-NAME=B, -
//                     DCN-NAME=X), -
// LINK-NAME2=PARTNER2(PROCESSOR-NAME=P616, -
//                     CONFIGURATION-NAME=C, -
//                     DCN-NAME=Y)
//END
```

The input file for the control statements of the SESDCNX distribution component contains the following:

```
//SET-DCN-OPTIONS -
// DCN-IDENTIFICATION=*PARAMETERS -
// (CONFIGURATION-NAME=B,DCN-NAME=X)
//ADD-DISTRIBUTION-RULE-LIST -
// CATALOG-NAME-1=WAREHOUSE1(LINK-NAME=PARTNER2,DBH-NAME=3), -
// CATALOG-NAME-2=WAREHOUSE2(LINK-NAME=PARTNER2,DBH-NAME=3), -
// CATALOG-NAME-3=CUSTOMERS(LINK-NAME=*HOME,DBH-NAME=4), -
// CATALOG-NAME-4=SALES(LINK-NAME=PARTNER1,DBH-NAME=5)
//ADD-NETWORK-LINK-LIST
// LINK-NAME1=PARTNER1(PROCESSOR-NAME=P616, -
//                     CONFIGURATION-NAME=C, -
//                     DCN-NAME=Y), -
// LINK-NAME2=PARTNER2(PROCESSOR-NAME=P614, -
//                     CONFIGURATION-NAME=A, -
//                     DCN-NAME=A)
//END
```

the following:

```
//SET-DCN-OPTIONS -
// DCN-IDENTIFICATION=*PARAMETERS -
// (CONFIGURATION-NAME=C,DCN-NAME=Y)
//ADD-DISTRIBUTION-RULE-LIST -
// CATALOG-NAME-1=WAREHOUSE1(LINK-NAME=PARTNER2,DBH-NAME=3), -
// CATALOG-NAME-2=WAREHOUSE2(LINK-NAME=PARTNER2,DBH-NAME=3), -
// CATALOG-NAME-3=CUSTOMERS(LINK-NAME=PARTNER1,DBH-NAME=4), -
// CATALOG-NAME-4=SALES(LINK-NAME=*HOME,DBH-NAME=5)
//ADD-NETWORK-LINK-LIST
// LINK-NAME1=PARTNER1(PROCESSOR-NAME=P616, -
//                     CONFIGURATION-NAME=B, -
//                     DCN-NAME=X), -
// LINK-NAME2=PARTNER2(PROCESSOR-NAME=P614, -
//                     CONFIGURATION-NAME=A, -
//                     DCN-NAME=A)
//END
```

5 DBH and SESDCN administration

This chapter describes both DBH and SESDCN administration. It covers:

- Administration interfaces
- the syntax and functions of the administration statements

5.1 Administration interfaces

There are three interfaces for dynamic DBH and SESDCN administration:

- administration using INFORM-PROGRAM (also with SEND-MSG, for reasons of compatibility)
- administration using a CALL DML program
- administration using the SESADM administration program.

In addition, the SESAM/SQL monitor SESMON provides information on current operation that is useful for administration. You can use the DBH and SESDCN statistics produced by SESMON to work out the optimal settings for the DBH options and SESDCN control statements (see chapter “Outputting operational data with SESMON”).

> The following always applies: Output sent to SYSOUT and SYSLST is not upwardly compatible. Output layout may vary depending on the version used.

5.1.1 Administration using INFORM-PROGRAM

When you use the BS2000 command INFORM-PROGRAM for administration, messages are sent to the DBH or SESDCN in the form of administration commands (STXIT routine). Replies are written to SYSLST and SYSOUT. For reasons of compatibility the earlier command SEND-MSG can also be used.

There are two ways to enter the administration commands:

- at the terminal at which the DBH or SESDCN was started in interactive mode
- at the console of the BS2000 system administrator.

The BS2000 STXIT routine restricts inputs to a length of 60 characters. You therefore have to abbreviate the keywords of some of the administration commands. No blanks may be entered between the syntax elements. All entries are automatically converted from lower to upper case.

An administration command sent to the SESAM-DBH can be distributed among several INFORM-PROGRAM commands. To do this, the entry of the INFORM-PROGRAM command to be continued must end with a comma. The next INFORM-PROGRAM command will then be interpreted as the continuation of the administration command. The administration command is only executed once the entire command entered is complete. This does not apply to SESDCN.

A job sent via the INFORM-PROGRAM interface is generally only approved once the SESAM/SQL-DBH system start has been completed (shown by outputting the message “SES0060 SYSTEM READY”). The same is true for SESAM-DCN (shown by outputting the message “SEN3020 SESDCN READY”).

The acknowledge is sent by the administration via INFORM-PROGRAM only after the job has been completely processes, i.e. the behavior is the same as for the administration when sent via the CALL-DML interface or via SESADM.

If an internal restart is triggered while an administration command is being processed, the command is aborted. It can happen in this case that the acknowledge is not sent although the command has already been executed. However, it may be that the command was not executed.

Entering commands at the terminal

If the DBH or SESDCN was started in interactive mode from the system administrator's terminal, you can enter the administration commands from this terminal.

To enter an administration command, proceed as follows:

- Press the key [K2] or [EM] [DUE] thus interrupting the task of the DBH (independent DBH or linked-in application) or SESDCN.
- Issue a INFORM-PROGRAM command to pass the administration command to the DBH or SESDCN task:

  `/INFORM-PROGRAM MSG='`*`admin-cmd`*`'`

  Any quotes in the text of the administration command must be entered twice.

Because the DBH or SESDCN is interrupted for the command input and cannot continue working during this time, it is not advisable to enter an administration command at the terminal.

at the console of the BS2000 system administrator.

If the DBH or SESDCN was started in batch mode, you must enter the administration commands at the console of the BS2000 system administrator. You enter the INFORM-PROGRAM command as follows:

`/INFORM-PROGRAM JOB-ID=*TSN(TSN=`*`tsn`*`),MSG='`*`admin-cmd`*`'`

where:

tsn is the task sequence number of the task in which the independent DBH, linked-in application or SESDCN was started.

Any quotes in the text of the administration command must be entered twice.

In addition, a user with the privilege OPERATING or TSOS can also administer DBHs or SESDCN via /INFORM-PROGRAM where the DBHs or SESDCN are started in the interactive mode from any terminal.

Outputs occurring during administration via INFORM-PROGRAM are made on the same terminal as the DBH messages according to the option MSG-OUTPUT.

Several concurrent STXIT routines

One INFORM-PROGRAM command can activate several STXIT routines concurrently. In the case of the linked-in DBH, for example, STXIT routines of both the DBH and the application program can be started.

When there are several concurrently active STXIT routines, you can direct the INFORM-PROGRAM command at a specific component, such as the DBH:

```
/INFORM-PROGRAM JOB-ID=*TSN(TSN=tsn),MSG='component,admin-cmd'
```

where:

component is the component to which the INFORM-PROGRAM command is to be directed. You can enter any of the following for *component*:

SES:	DBH-STXIT
S63:	SEDI63-STXIT
DCN:	SESDCN-STXIT

component is an optional entry for the administration of SES DCN and the independent DBH, but it is mandatory for the linked-in DBH.

Outputting successor responses

Large database configurations supply extensive outputs, e.g. in the inquiry commands. An output section (a so-called response) has a maximum size of 32000 bytes. Larger output volumes are output in multiple responses.

You can call successor responses in the DBH using the administration command NEXT.

Successor responses can occur in the following administration commands because one output section may not suffice:

USER,CATALOG	(DBH and DCN)
USER,INACT	(DBH)
USER,PR	(DBH and DCN)
USER,SPACE	(DBH)
USER,TA	(DBH and DCN)
SPACE	(DBH)

OPT,CDBC-INFO	(DBH)
OPT,DBC-INFO	(DBH)

In the case of the administration command HOLD-TA information about transactions is output in successor responses if transactions still exist.

In the case of the administration command STOP information about transactions is output in successor responses if PTC transactions, utility statements or lock sequences still exist. In this case the system is not terminated.

5.1.1.1 NEXT (administration command)

This command calls successor responses in the case of DBH administration commands.

In SESDCN and at administration statement level all responses are output in full; there are no successor responses there. See also section “Controlling output of the administration statements”.

Function

A (partial) response has a maximum size of 32000 bytes. Larger output volumes are output in multiple responses.

You use the NEXT command to call successor responses in the DBH if the previous (incomplete) output was terminated with the message SES7351.

If an output terminates with the message SES7352, no successor responses exist.

Any command other than NEXT terminates all pending successor outputs. These can then no longer be called later.

Administration command in ISP format and at the CALL DML interface

```
NEXT
```

The administration command NEXT has no operands.

Example

```
/INFORM-PROGRAM JOB-ID=*TSN(TSN=tsn),MSG='SES,NEXT'
```

5.1.2 Administration using a CALL DML program

In a CALL DML program, you can issue administration commands to the DBH or to SESDCN by means of a CALL DML statement. The following prerequisites must be fulfilled:

- When the DBH or SESDCN is started, administration via the CALL interface must be permitted (using the DBH or DCN option ADMINISTRATOR).
- The ADMINISTRATOR open statement must be issued in the CALL DML program to initiate the administration of the DBH (see the “ CALL-DM Applications” manual). Until the close statement, only administration statements in this program run can be executed.

Administration via the CALL interface can be carried out from any BS2000 user ID.

No blanks may be entered between the syntax elements.

The transfer areas of the CALL DML statement for administration contain the following:

statement area:	where the application program enters the statement in the form of an administration command in ISP format.
acknowledgment area:	where the DBH or SESDCN reports the acknowledgment of the administration command.
response area:	where the responses to administration calls are output.

The inquiry area is not evaluated by the DBH or SESDCN.

You will find a thorough description of the different transfer areas in the “ CALL-DM Applications” manual.

Outputting successor responses

Large database configurations supply extensive outputs, e.g. in the inquiry commands. An output section (a so-called response) has a maximum size of 32000 bytes. Larger output volumes are output in multiple responses.

Successor responses can occur in the following administration commands because one output section may not suffice:

USER,CATALOG	(DBH and DCN)
USER,INACT	(DBH)
USER,PR	(DBH and DCN)
USER,SPACE	(DBH)
USER,TA	(DBH and DCN)
SPACE	(DBH)
OPT,CDBC-INFO	(DBH)
OPT,DBC-INFO	(DBH)

In the case of the administration command HOLD-TA information about transactions is output in successor responses if transactions still exist.

In the case of the administration command STOP information about transactions is output in successor responses if PTC transactions, utility statements or lock sequences still exist. In this case the system is not terminated.

Outputting DBH successor responses

You can call DBH successor responses using the following extension of the CALL-DML interface (op code 010):

```
'<password>010NEXT9 '
```

The statement supplies a successor response only if the preceding administration response was acknowledged with the status “0A/00” (overflow in the response area) and terminated with the message SES7351.

The end of the response output is indicated with the status”00” and terminated with message SES7215.

If no successor response exists the statement is acknowledged with the status “02/01”.

Any administration statement other than NEXT terminates all pending successor outputs. These can then no longer be called later.

Outputting DCN successor responses

You can call DCN successor responses using the following extension of the CALL-DML interface (op code 020).

```
'<password>020<Antwortlänge> [P<host-name>] [C<cnf>] [D<nam>] ANEXT9 '
```

The statement supplies a successor response only if the preceding administration response was acknowledged with the status “0A/AB” (overflow in the response area) and terminated with the message SEN2052.

The end of the response output is indicated with the status”00” and terminated with message SEN2014.

If no successor response exists the statement is acknowledged with the status “02/SE” and message SEN2053.

The response length specified must be the same as that of the preceding administration statement. Otherwise the statement is acknowledged with the status “02/SF” and terminated with message SEN2005.

Any command other than NEXT terminates all pending successor outputs. These can then no longer be called later.

If no further storage area is free for the connection to an administration program, the status “0B/A1” is reported.

5.1.3 Administration using SESADM

SESADM is an easy-to-use, SDF-based program for administering the independent DBH and SESDCN.

SESADM reads in administration input via SYSDTA.

By default output is sent to SYSOUT when run in interactive mode and to SYSLST when run in batch mode. Outputs (e.g. of SHOW statements) can also be written to a temporary file and output using SHOW-FILE. The information outputs of most SHOW statements can be output in S variables and processed further in S procedures.

In interactive mode status messages and error messages are sent to SYSOUT and SYSLST, and in batch mode to the console and SYSLST. In addition, the message number of the last response to an administration statement is recorded in an S variable and in a temporary job variable. This permits automatic administration with SESADM.

The SESADM administration program builds on the CALL DML interface. It forms a CALL DML statement from the entries made by the user, and issues it to the connection module via CALL SESAM. The connection module converts the statements and forwards them to the DBH or the responsible SESDCN distribution component.

SESADM automatically checks the output of the statements. If a status other than “00” is returned, SESADM sets task switch 11 and outputs a corresponding message (see also section “Setting task switches”). In both interactive and batch mode, SESADM then continues running and awaits further input.

Access from the World Wide Web

You can also access the administration program SESADM, the performance monitor SESMON and the utility monitor SESUTI all from a unified access on the World Wide Web (WWW or Web for short) with the aid of the software product WebTransactions (WebTA).

To access the SESAM programs via the Web, you only need a standard browser in addition to the software product WebTransactions.

Web access is described in the document “ WebTA access for SESAM/SQL” shipped together with SESAM/SQL-Server. This document is also available from our manual server under the software product SESAM/SQL.

Embedding of SESADM

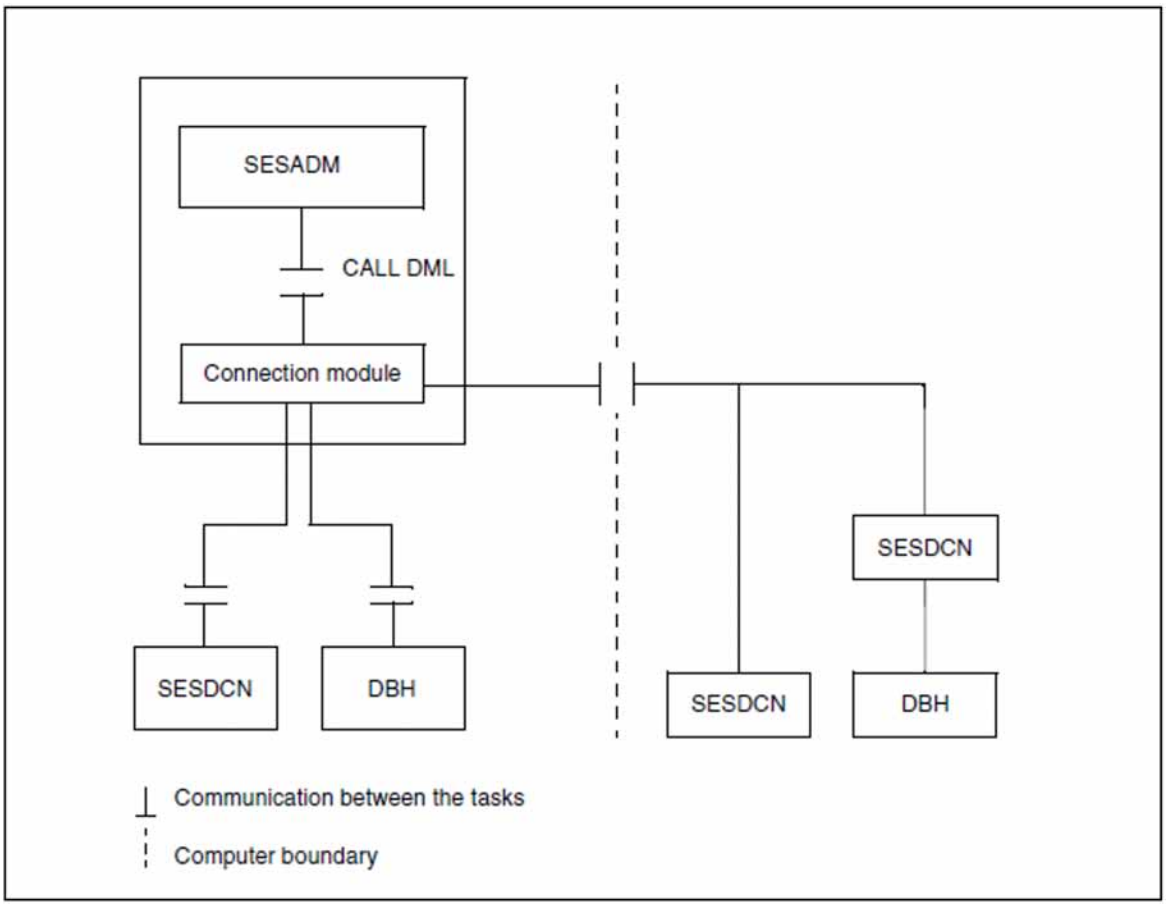


Figure 4: The SESADM administration interface

It is not possible to use SESADM to administer the linked-in DBH.

The SESADM program runs in the SESAM/SQL configuration specified in the configuration file assigned to this program. If SESADM is started without a configuration file, it runs in the default configuration "BLANK".

If the configuration in which SESADM is running is a non-distributed configuration (SESDCN not loaded), only the default DBH of this specific configuration can be administered with this SESADM. This is the DBH whose DBH name and configuration identifier are specified in the configuration file assigned to the SESADM program; if no configuration file is assigned, it is the DBH with the NAM identifier "BLANK" and the configuration identifier ' 'BLANK".

Specification of any other values in the CONFIGURATION-NAME and DBH-NAME parameters in the SESADM control statement START-DBH-ADMINISTRATION, or any attempt to administer SESDCN will be rejected with error messages.

Please note the following when administering a SESAM/SQL DBH or a SESAM/SQL DCN.

Administration of a SESAM/SQL DBH

If the DBH to be administered is running in a non-distributed configuration:

- Start SESADM in the same configuration and assign the DBH to be administered, i.e. the SESADM program must be assigned a configuration file in which the DBH name and configuration identifier of the DBH to be administered are specified.
- If the configuration file is missing or has not been assigned, then you can input additional identification for the DBH to be administered via the parameters DBH-NAME and CONFIGURATION-NAME in the SESADM control statement START-DBH-ADMINISTRATION. Otherwise the DBH is assigned the NAM identifier "BLANK" and the configuration identifier "BLANK".

If the DBH to be administered is running in a distributed configuration:

- As above

or

- Start SESADM in any distributed configuration (this can be the same as that in which the DBH is running, or a different one) whose distribution rule contains at least one database which is assigned to the DBH to be administered; then identify the DBH to be administered using the DBH-NAME and CONFIGURATION-NAME parameters in the SESADM control statement START-DBH-ADMINISTRATION.

Administration of a SESAM/SQL DCN

Start SESADM in any distributed configuration (this can be the same as that in which SESDCN is running, or a different one) and identify the SESDCN to be administered using the DCN-NAME and CONFIGURATION-NAME parameters in the SESADM control statement START-DCN-ADMINISTRATION.

5.1.3.1 Starting SESADM

SESADM runs in BS2000 as a separate task. Accordingly, you can start SESADM in interactive or batch mode. You can also call SESADM from the utility monitor (see the “ Utility Monitor” manual).

You can also access the administration program SESADM from the World Wide Web, see "Administration using SESADM".

Sequence of commands for starting SESMON

`[/ASSIGN-SYSDTA TO-FILE=`*`sesadm-inp`*	1.
`[/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=`*`configuration-file`*`]`	2.
`/START-SESAM-ADMINISTRATION`	3.

1. Assigns the SESADM input file. You only need to enter this command if you are entering the SESADM statements from an input file.
2. Assigns a configuration file which can contain as a parameter the name of the configuration under which SESADM is to run, for example. This command is optional and it is one of the ways to assign the configuration file (see the “ Core manual”).
3. Starts the administration program with the start command START-SESAM-ADMINISTRATION (see section “Starting SESAM/SQL programs via start commands”).

5.1.3.2 Entering the SESADM statements

The SESADM administration program reads in the administration inputs with the help of SDF (see the “ SDF Dialog Interface”).

SESADM recognizes the following statements:

- SESADM control statements (see "SESADM control statements")
- administration statements (see "Alphabetical reference section").

The syntax of the SESADM control and administration statements complies with SDF conventions.

There are different ways of entering SESADM statements, depending on the mode used:

In interactive mode:

- you can enter them interactively, directly on screen, with SDF support
- you can enter them in the SESADM start procedure
- you can enter them in the SESADM input file, which must be assigned before SESADM is started.

In batch mode:

- you can enter them in the procedure for starting SESADM
- you can enter them in the SESADM input file, which must be assigned before SESADM is started.

Entering SESADM statements interactively

For the interactive entry of SESADM statements directly on screen, SDF offers two modes:

- expert mode (the default)
- Nonexpert mode

You can use the SDF statement MODIFY-SDF-OPTIONS whenever you like to set the mode.

In **expert mode**, a system prompt (//) appears after SESADM starts. If you enter a question mark (?) here instead of an SESADM statement, SDF changes to nonexpert mode and displays the corresponding selection screens. When you enter an SESADM statement, SDF switches back to expert mode and the system prompt appears again.

In **nonexpert mode**, SDF displays a selection screen of the permitted statements when SESADM is started. When you select a statement, SDF supports operand input by means of appropriate SDF selection screens (see section “SDF selection screens for SESADM administration”).

Entering SESADM state ments via an input file or procedure

When entering SESADM statements via an input file or in a start procedure, the following rules apply:

- The first statement must be the SESADM control statement START-DBH-ADMINISTRATION or START-DCN-ADMINISTRATION. This SESADM control statement identifies the DBH or SESDCN to be administered. All the administration statements that come immediately after it relate to the DBH or SESDCN thus identified.
- The first statement is followed by administration statements to the identified DBH or SESDCN.

- The administration statements can be followed by another identifying SESADM control statement, in turn followed by associated administration statements. This step can be repeated several times.
- The SDF standard statement STEP marks a section of administration statements (see "SDF standard statement STEP").
- The SESADM control statement END terminates administration with SESADM (see "END").

Example

The example below shows an input file for administration with SESADM. The input file contains both DBH and SESDCN administration statements.

```
//START-DCN-ADMINISTRATION -
//      PASSWORD='ADM',DCN-NAME=A,CONFIGURATION-NAME=2,HOST-NAME=P614
//SHOW-USERS
//START-DBH-ADMINISTRATION -
//      PASSWORD='ADM',DBH-NAME=7
//SHOW-DBH-OPTIONS
//STEP
//MODIFY-ADMINISTRATION NEW-PASSWORD='XTI'
//MODIFY-SUBORDER-LIMIT LIMIT=1000
//SHOW-CATALOG-USERS
//STEP
//ROLLBACK-TRANSACTION USER=*TSN(TSN=3343)
//STEP
//START-DCN-ADMINISTRATION -
//      PASSWORD='ADM',DCN-NAME=B
//STEP
//SHOW-DISTRIBUTION-RULE-ENTRIES
//REMOVE-DISTRIBUTION-RULE-ENTRY CATALOG-NAME=TEST1
//STEP
//START-DCN-ADMINISTRATION -
//      PASSWORD='ADM',DCN-NAME=C
//SHOW-DISTRIBUTION-RULE-ENTRIES
//MODIFY-DISTRIBUTION-RULE-ENTRY -
//       HOST-NAME=P614,NEW-NAME=P616
//ADD-DISTRIBUTION-RULE-ENTRY
//       CATALOG-NAME=TEST2,DCN-NAME=C,CONFIGURATION-NAME=X,HOST-NAME=P616
//SHOW-TRANSACTIONS
//STEP
//ROLLBACK-TRANSACTION USER=*BY-SELECT(HOST-NAME=P614, -
//       *TSN=(TSN=2012),CUSTOMER-NAME=TEST)
//END
```

5.1.3.3 SDF selection screens for SESADM administration

This section describes the selection screens provided by SDF when you enter the SESADM statements interactively (see "Entering the SESADM statements"). In nonexpert mode, SDF always displays the selection masks; in expert mode, only if you enter a question mark (?).

SDF provides the following selection screens:

- START-MENÜ
- DBH-MENÜ
- DCN-MENÜ

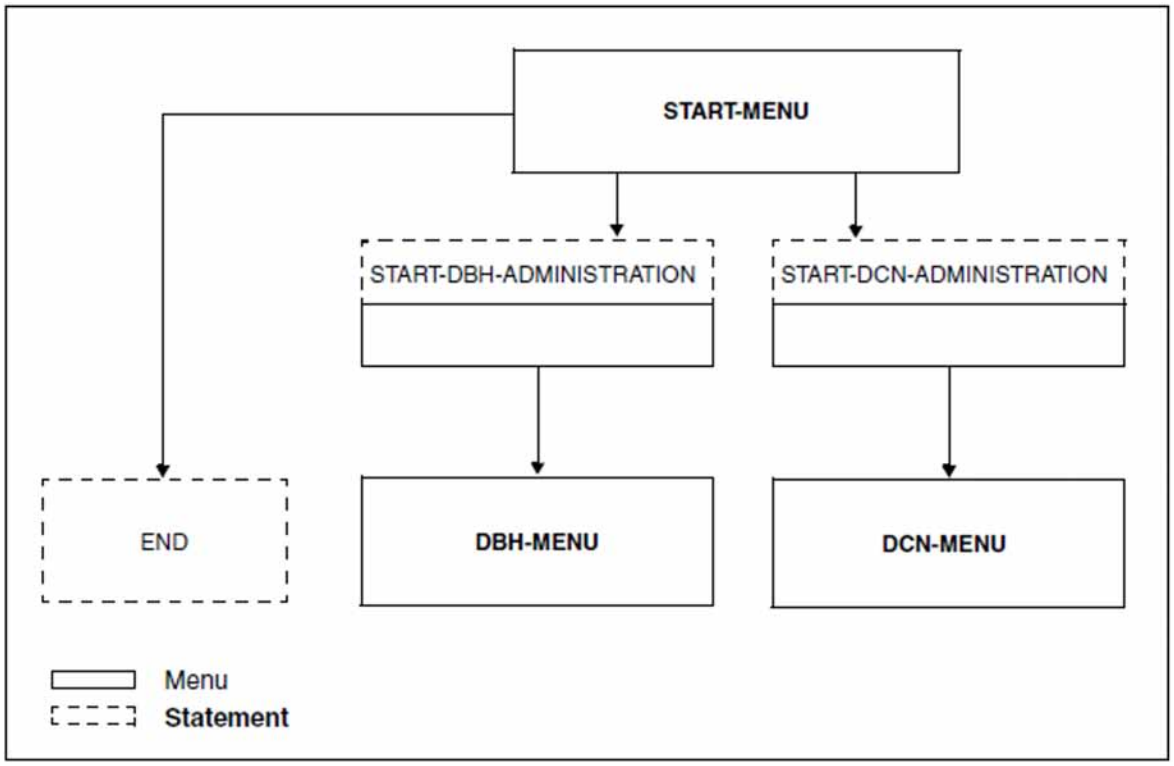


Figure 5: SDF selection screens for SESADM administration

SDF displays the **START-MENU** selection screen after SESADM is started. In it you select one of the following three SESADM control statements:

- START-DBH-ADMINISTRATION (see "START-DBH-ADMINISTRATION")
- START-DCN-ADMINISTRATION (see "START-DCN-ADMINISTRATION")
- END (see "END").

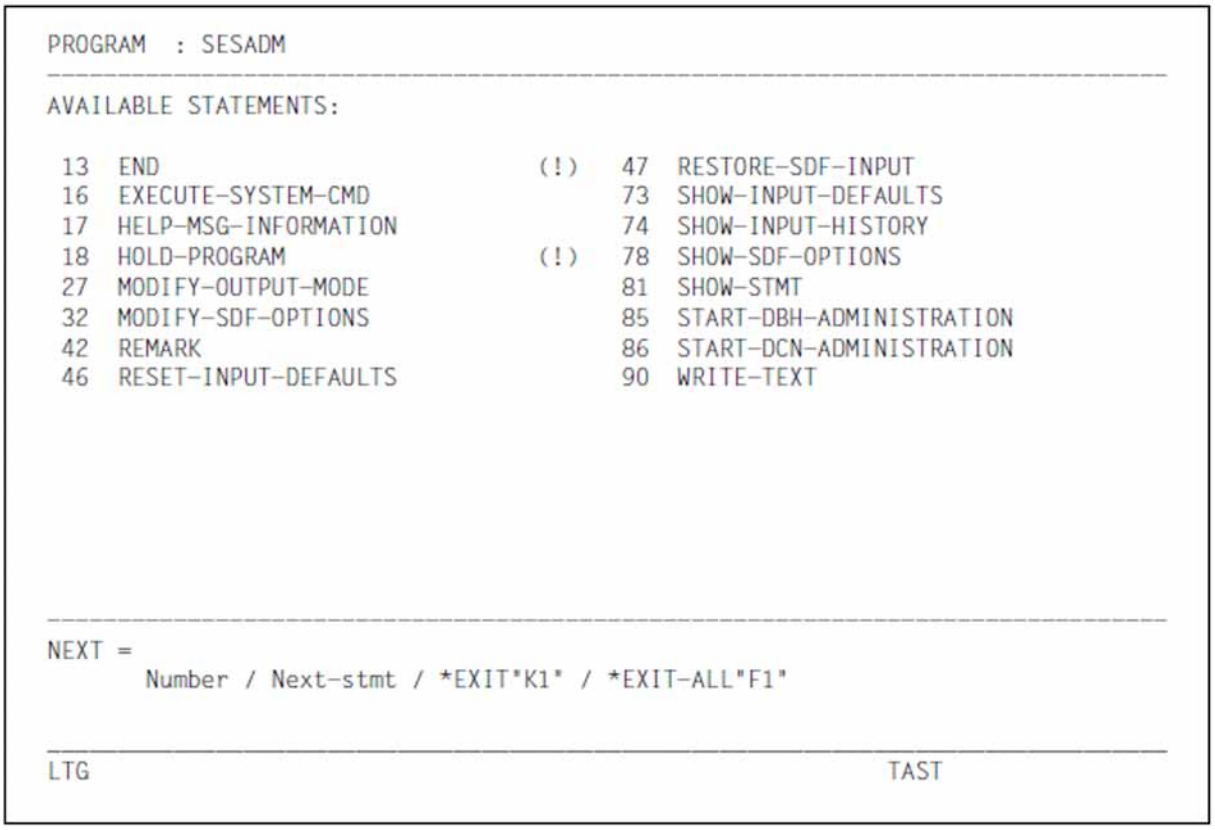
DBH-MENU is the selection screen for administration statements to the DBH. SDF switches to DBH-MENU when you select the START-DBH-ADMINISTRATION control statement from the START-MENU selection screen and enter the associated parameters.

DCN-MENU is the selection screen for administration statements to SESDCN. SDF switches to DCN-MENU when you select the START-DCN-ADMINISTRATION control statement from the START-MENU selection screen and enter the associated parameters.

You will find alphabetically arranged descriptions of all the administration statements starting on "Alphabetical reference section".

All three menus contain the SDF standard statements MODIFY-SDF-OPTIONS, REMARK, RESTORE-SDF-INPUT, SHOW-INPUT-HISTORY, SHOW-SDF-OPTIONS, and WRITE-TEXT. You will find descriptions of the SDF standard statements in the “ SDF Dialog Interface”.

START MENU

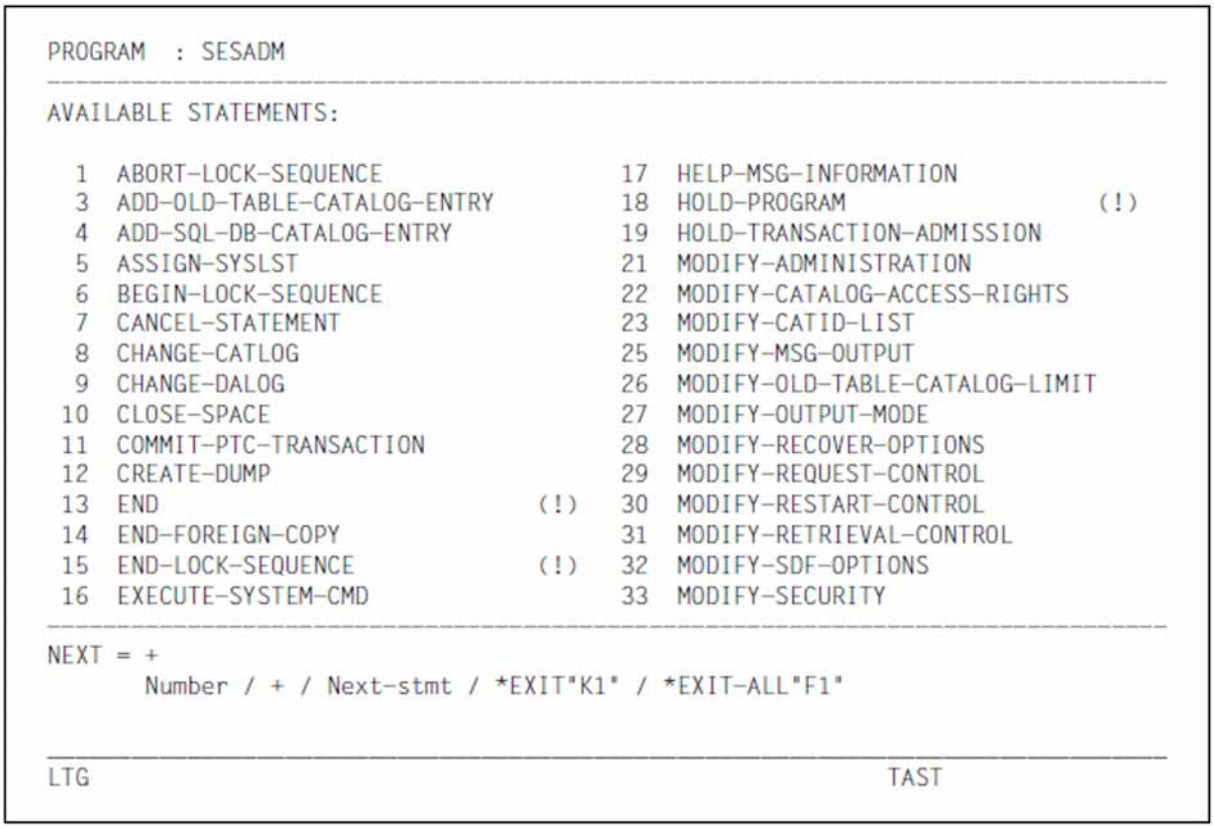


Screen description

NEXT= Enter here the number of the statement you want to use.

DBH-MENU

The DBH-MENU selection screen is divided into three pages. The first page is shown below. The subsequent pages are structured in the same way.



Screen description

NEXT= Enter here the number of the statement you want to use. There are three pages of statements. To turn to the next page, enter a plus sign (+). To turn back to the previous page (in a subsequent page), enter a minus sign (-).

DCN-MENU

The DBH-MENU selection screen is divided into three pages. The first page is shown below. The list of statements is merely supplemented by “ `90 WRITE-TEXT`” on the next page.

```
PROGRAM  : SESADM
------------------------------------------------------------------------------
AVAILABLE STATEMENTS:

  2  ADD-DISTRIBUTION-RULE-ENTRY          47  RESTORE-SDF-INPUT
 12  CREATE-DUMP                          48  RESUME-TRANSACTION-ADMISSION  (!)
 13  END                            (!)   49  RESUME-USER-ADMISSION          (!)
 16  EXECUTE-SYSTEM-CMD                   53  ROLLBACK-TRANSACTION
 17  HELP-MSG-INFORMATION                 63  SET-USER-CALL-TRACE
 18  HOLD-PROGRAM                   (!)   65  SET-USER-MSG-TRACE
 19  HOLD-TRANSACTION-ADMISSION           71  SHOW-DISTRIBUTION-RULE-ENTRIES(!)
 20  HOLD-USER-ADMISSION            (!)   73  SHOW-INPUT-DEFAULTS
 21  MODIFY-ADMINISTRATION                74  SHOW-INPUT-HISTORY
 24  MODIFY-DISTRIBUTION-RULE-ENTRY       75  SHOW-LAST-RESULT              (!)
 27  MODIFY-OUTPUT-MODE                   78  SHOW-SDF-OPTIONS
 32  MODIFY-SDF-OPTIONS                   81  SHOW-STMT
 42  REMARK                               82  SHOW-TRANSACTIONS
 43  REMOVE-DISTRIBUTION-RULE-ENTRY       84  SHOW-USERS                    (!)
 46  RESET-INPUT-DEFAULTS                 89  STOP-DCN                      (!)
------------------------------------------------------------------------------
NEXT =
       Number / Next-stmt / *EXIT"K1" / *EXIT-ALL"F1"


------------------------------------------------------------------------------
LTG                                                          TAST
```

Screen description

NEXT=	Enter here the number of the statement you want to use. There are two pages of statements. To turn back to the previous page, enter a minus sign (-).

5.1.3.4 SESADM control statements

On the following pages, you will find detailed descriptions of the SESADM control statements:

- START-DBH-ADMINISTRATION
- START-DCN-ADMINISTRATION
- END

5.1.3.5 START-DBH-ADMINISTRATION

This statement starts DBH administration.

Function

START-DBH-ADMINISTRATION is an SESADM control statement. If you are working in nonexpert mode, when you issue this statement, the DBH-MENU selection screen appears.

When you enter the START-DBH-ADMINISTRATION statement, you also specify which DBH you want to administer.

START-DBH-ADMINISTRATION
PASSWORD = <c-string 3..3> / <x-string 5..6> ,DBH-NAME = *STD / <alphanum-name 1..1> / *BLANK ,CONFIGURATION-NAME = *HOME / <alphanum-name 1..1> / *BLANK ,HOST-NAME = *HOME / <name 1..8>

Operands

PASSWORD = <c-string 3..3> / <x-string 5..6>

Password that prevents unauthorized administration of the DBH using SESADM. It is assigned by means of the DBH option ADMINISTRATOR when the DBH is started.

DBH-NAME = *STD / <alphanum-name 1..1> / *BLANK
Name by which the DBH to be administered is identified. The default is the DBH name assigned via the configuration file. In a non-distributed configuration, only the default value can be specified.

CONFIGURATION-NAME = *HOME / <alphanum-name 1..1> / *BLANK

Configuration name of the DBH to be administered. The default value is the name of the home configuration in which SESADM is running. In a non-distributed configuration, only the default value can be specified.

HOST-NAME = *HOME / <name 1..8>
Symbolic device name of the computer on which the DBH is loaded. The default value is the name of the home system. In a non-distributed configuration, only the default value can be specified.

5.1.3.6 START-DCN-ADMINISTRATION

This statement starts SESDCN administration.

Function

START-DCN-ADMINISTRATION is an SESADM control statement. If you are working in nonexpert mode, when you issue this statement, the DCN-MENU selection screen appears.

When you enter the START-DCN-ADMINISTRATION statement, you also specify which SESDCN you want to administer.

START-DCN-ADMINISTRATION
PASSWORD = <c-string 3..3> / <x-string 5..6> ,DCN-NAME = *BLANK / <alphanum-name 1..1> ,CONFIGURATION-NAME = *HOME / <alphanum-name 1..1> / *BLANK ,HOST-NAME = *HOME / <name 1..8>

Operands

PASSWORD = <c-string 3..3> / <x-string 5..6>

Password that prevents unauthorized administration of the DBH using SESADM. It is assigned by means of the DCN option ADMINISTRATOR when SESDCN is started.

DCN-NAME = *BLANK / <alphanum-name 1..1>
Name by which the SESDCN distribution component to be administered is identified. The default is a blank.

CONFIGURATION-NAME = *HOME / <alphanum-name 1..1> / *BLANK

Configuration name of the SESDCN distribution component to be administered. The default is the name of the home configuration in which SESADM is running.

HOST-NAME = *HOME / <name 1..8>
Symbolic device name of the computer on which the SESDCN distribution component is loaded. The default is the name of the home system from which administration is carried out.

5.1.3.7 END

This statement terminates SESADM.

Function

END is a SESADM control statement. When you issue it, SESADM terminates.

SESADM automatically checks the output of the statements. If a status other than “00” is returned, SESADM sets task switch 11 and outputs a corresponding message.

```
END
```

Operands

This statement has no operands.

5.1.3.8 Controlling output of the administration statements

Large database configurations supply extensive outputs, e.g. in the SHOW statement of SESADM. The statements in the DBH and DCN menus below enable you to control the output of administration statements to SYSOUT /SYSLST or to a temporary file:

- MODIFY-OUTPUT-MODE
- SHOW-LAST-RESULT

5.1.3.9 MODIFY-OUTPUT-MODE

This statement defines the output destination of administration statements.

Function

The MODIFY-OUTPUT-MODE statement defines the output destination for all the following administration statements

MODIFY-OUTPUT-MODE
OUTPUT = *STD / *SHOW-FILE / *MAIL

Operands

OUTPUT= *STD

In interactive mode all outputs are directed to SYSOUT, in batch mode to SYSLST.

OUTPUT= *SHOW-FILE
Output information of an administration statement (e.g. of a SHOW statement) is written to the temporary file “#SESADM.*tsn*.RESULT” and output using the BS2000 command SHOW-FILE. This provides you with user-friendly navigation functions, e.g. scrolling.

In the case of OUTPUT=*SHOW-FILE messages are also always directed to SYSOUT when interactive mode is used, and always to SYSLST in batch mode.

This parameter is effective only when SESADM is called in interactive mode.

OUTPUT= *MAIL

Output information of administration statements (e.g. of a SHOW- or ROLLBACK-TRANSACTION statement) is written to the temporary file “#SESADM. *tsn*.RESULT-F”. It is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email") when the next MODIFY-OUTPUT-MODE statement is issued or when the DBH program terminates. The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SESADM” is entered under “Subject:”. The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The temporary file is attached to the email.

Messages or information of the DBH are not output to SYSLST or SYSOUT when OUTPUT=*MAIL.

After MODIFY-OUTPUT-MODE OUTPUT=*MAIL has been entered for the first time, the MAIL parameters for this SESADM can no longer be changed. SESADM must be restarted to do this.

5.1.3.10 SHOW-LAST-RESULT

This statement displays the last output of an administration statement

Function

The SHOW-LAST-RESULT statement outputs the last output of an administration statement from the temporary file “#SESADM.*tsn*.RESULT” using the BS2000 command SHOW-FILE.

```
SHOW-LAST-RESULT
```

Operands

This statement has no operands.

The statement makes sense only if output to a temporary file was selected beforehand by means of MODIFY-OUTPUT-MODE OUTPUT=*SHOW-FILE.

5.1.3.11 Structured output in S variables

The software product SDF-P is a procedural language which enhances the command language of BS2000 into a programming language. Procedures which are created in accordance with the rules of SDF-P are call structured procedures (S procedures). SDF-P offers a variables concept in which both composite variables (S variables) and variable streams (S variable streams) can be processed. Detailed information on SDF-P and on working with S variables is provided in the manuals “ Programming in the Command Language” and “ Commands, Volume 1”.

SESADM supplies outputs in S variables for most SHOW administration statements if the `STRUCTURE-OUTPUT` operand (see "Syntax for output in S variables") has a value other than `*NONE`. The S variables can be evaluated and processed using SDF-P. This facilitates automatic analysis using SESADM.

Administration statements with output in S variables

SESADM offers output in S variables for the following administration statements:

Administration statement	Application area
SHOW-CATALOG-USERS	DBH administration
SHOW-CATID-LIST	DBH administration
SHOW-DISTRIBUTION-RULE-ENTRIES	DCN administration
SHOW-INACTIVE-SQL-USERS	DBH administration
SHOW-MAIL-PARAMETERS	DBH administration
SHOW-OLD-TABLE-CATALOG-ENTRIES	DBH administration
SHOW-PARTITIONS	DBH administration
SHOW-SPACE-USERS	DBH administration
SHOW-SQL-DB-CATALOG-ENTRIES	DBH administration
SHOW-TRANSACTIONS	DBH and DCN administration
SHOW-USERS	DBH and DCN administration
SHOW-USER-SPACES	DBH administration

Table 14: Administration statements with output in S variables

Output in S variables is a component part of each command description. The structure and content of the S variables are listed in tabular form after the command descriptions, specifying the name, meaning and content.

The structure of the S variables is predefined for each statement and is guaranteed for subsequent versions, i.e. it will not be changed in a way that makes it incompatible.

Name components of the S variables

The names of the list elements (e.g. `CATID`) are predefined for each information command. They follow the S variable name declared by the user. The names are based as far as possible on the corresponding output names. If the information at this level is structured further in a hierarchy, further names will follow, separated by a period.

Content of the S variables

The content of the S variables matches the corresponding output values. The data type of the SESADM S variables is always <string>, i.e. a string is output.

Working with S variables in SESADM

Output in S variables takes place in addition to the normal output if the STRUCTURE-OUTPUT operand is not specified as *NONE. However, no separator or header lines are output in S variables.

The STRUCTURE-OUTPUT operand (see "Syntax for output in S variables") controls the output in S variables:

- STRUCTURE-OUTPUT=*NONE supplies no output in S variables
- STRUCTURE-OUTPUT=<name of the S variable>(WRITE-MODE=*REPLACE/*EXTEND) supplies output in the specified S variable. Beforehand the S variable must be declared as a dynamic, structured list variable in the S procedure. The S variable can be overwritten (*REPLACE) or extended (*EXTEND). See the example on "Structured output in S variables".
- STUCTURE-OUTPUT=*SYSINF supplies output to the variable stream SYSINF to which an S variable is assigned by means of ASSIGN-STREAM. The S variable is updated. The content can be deleted using the SDF-P command FREE-VARIABLE. See the example on "Structured output in S variables".

In the case of structured output in S variables, each of the current items of information is stored as the content of an S variable. Independently of the layout of the screen or list output the user accesses the required item of information in the information command via the names and structure identifiers of the S variables.

It is recommendable to use different S variables for different administration statements as the structure elements generated are dependent on the statement.

Example

In the example below the entries of the SQL directories of two DBHs of a configuration are written and output in an S variable:

```
/DECLARE-VARIABLE -
/        VARIABLE-NAME=ADMVAR(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC)), -
/        MULTIPLE-ELEMENTS=*LIST                                          1.
/START-SESAM-ADMINISTRATION                                               2.
//START-DBH-ADMINISTRATION PASSWORD=C'ADM',DBH-NAME=W                     3.
//SHOW-SQL-DB-CATALOG-ENTRIES SELECT=*ALL, -
//                    STRUCTURE-OUTPUT=ADMVAR(WRITE-MODE=*REPLACE)        4.
<catalog-name>       | Copy-|Class| UserID   | Status  | Info
  <physical-name>    | Nr   |     |          |         |
-------------------------------------------------------------
DCCAT                |      |ADMIN| SQLDB112 | ACTIVE  |
  WIE                |      |     |          |         |
VIRST                |      |ADMIN| SQLDB112 | ACTIVE  |
  WIE.FIRST          |      |     |          |         |
% <date> <time> SES7215 END OF OUTPUT
//START-DBH-ADMINISTRATION PASSWORD=C'ADM',DBH-NAME=V                     5.
//SHOW-SQL-DB-CATALOG-ENTRIES SELECT=*ALL, -
//                    STRUCTURE-OUTPUT=ADMVAR(WRITE-MODE=*EXTEND)         6.
<catalog-name>       | Copy-|Class| UserID   | Status  | Info
  <physical-name>    | Nr   |     |          |         |
-------------------------------------------------------------
DCCAT                |      |ADMIN| SQLDB112 | ACTIVE  |
  WIE                |      |     |          |         |
```

```
SECON                 |      |ADMIN| SQLDB112 | ACTIVE  |
  WIE.SEC             |      |     |          |         |
% <date> <time> SES7215 END OF OUTPUT
//END                                                                    7.
Continued on the next page
```

1. The composite structured S variable ADMVAR of the type list is created.
2. SESADM is started.
3. The administration of DBH W is started.
4. Statement for outputting the DBHs' SQL directories. The S variable ADMVAR is overwritten (*REPLACE, default value). The output on SYSLST follows.
5. The administration of DBH V is started.
6. Statement for outputting the DBHs' SQL directories. The S variable ADMVAR is update (*EXTEND). The output on SYSLST follows.
7. The administration with SESADM is terminated.

```
/SHOW-VARIABLE VAR-NAME=ADMVAR, -
/               INFORMATION=*PARAMETER(LIST-INDEX-NUMBER=*YES) 8.
(OUT) ADMVAR#1.CATALOG = DCCAT
(OUT) ADMVAR#1.ACCESS-MODE = ADMIN
(OUT) ADMVAR#1.USERID = SQLDB112
(OUT) ADMVAR#1.STATE = ACTIVE
(OUT) ADMVAR#1.INFORMATION =
(OUT) ADMVAR#1.PHYSICAL-NAME = WIE
(OUT) ADMVAR#1.COPY-NUMBER =
(OUT) ADMVAR#2.CATALOG = VIRST
(OUT) ADMVAR#2.ACCESS-MODE = ADMIN
(OUT) ADMVAR#2.USERID = SQLDB112
(OUT) ADMVAR#2.STATE = ACTIVE
(OUT) ADMVAR#2.INFORMATION =
(OUT) ADMVAR#2.PHYSICAL-NAME = WIE.FIRST
(OUT) ADMVAR#2.COPY-NUMBER =
(OUT) ADMVAR#3.CATALOG = DCCAT
(OUT) ADMVAR#3.ACCESS-MODE = ADMIN
(OUT) ADMVAR#3.USERID = SQLDB112
(OUT) ADMVAR#3.STATE = ACTIVE
(OUT) ADMVAR#3.INFORMATION =
(OUT) ADMVAR#3.PHYSICAL-NAME = WIE
(OUT) ADMVAR#3.COPY-NUMBER =
(OUT) ADMVAR#4.CATALOG = SECON
(OUT) ADMVAR#4.ACCESS-MODE = ADMIN
(OUT) ADMVAR#4.USERID = SQLDB112
(OUT) ADMVAR#4.STATE = ACTIVE
(OUT) ADMVAR#4.INFORMATION =
(OUT) ADMVAR#4.PHYSICAL-NAME = WIE.SEC
(OUT) ADMVAR#4.COPY-NUMBER =
```

8. The content of the S variable ADMVAR is output.In the case of the list elements, the element number is appended to the name instead of (*LIST).

When the variable stream SYSINF is used, the corresponding statements are:

```
/DECLARE-VARIABLE -
/        VARIABLE-NAME=ADMVAR(TYPE=*STRUCTURE(DEFINITION=*DYNAMIC)), -
/        MULTIPLE-ELEMENTS=*LIST
/ASSIGN-STREAM STREAM-NAME=SYSINF,TO=*VARIABLE(VARIABLE-NAME=ADMVAR)   1.
/START-SESAM-ADMINISTRATION
//START-DBH-ADMINISTRATION PASSWORD=C'ADM',DBH-NAME=W
//SHOW-SQL-DB-CATALOG-ENTRIES SELECT=*ALL,STRUCTURE-OUTPUT=*SYSINF     2.
//START-DBH-ADMINISTRATION PASSWORD=C'ADM',DBH-NAME=V
//SHOW-SQL-DB-CATALOG-ENTRIES SELECT=*ALL,STRUCTURE-OUTPUT=*SYSINF
//END
/SHOW-VARIABLE VAR-NAME=ADMVAR, -
//              INFORMATION=*PARAMETER(LIST-INDEX-NUMBER=*YES)
```

1. The ASSIGN-STREAM command assigns an S variable stream for structured outputs to an (output) server which controls further processing of the variable stream. The server's output is the S variable.
2. The output is directed to the S variable stream SYSINF and consequently to the S variable.

Special cases

If no output information is present, the content of the S variable concerned is deleted if `WRITE-MODE=*REPLACE` for `WRITE-MODE=*EXTEND`, it remains unchanged.

If output information is present but cannot be written in the S variable, a message to this effect (`SEA7032`, `SEA7033`) is issued. These message numbers are also noted in the S variable SESADM-RESULT, see section “Result of the administration statements”.

5.1.3.12 Result of the administration statements

In interactive mode status messages and error messages are sent to SYSOUT and SYSLST, and in batch mode to the console and SYSLST.

In addition, the message number of the last response of the DBH or SESDCN to an administration statement is recorded in an S variable and in a job variable. This permits automatic administration with SESADM.

The S variable is called **SESADM-RESULT**. It has to be declared within the SDF procedure beforehand.

The (temporary) job variable is called **#SESAM.SESADM.JV**.

The following contents are possible for successful statements:

- `SES7215` (DBH), `SEN2014` (SESDCN) for “successful information output”
- `SES7290` (DBH), `SEN2051` (SESDCN) for “successful administration statement without information output”

The message number of the related error message is output for unsuccessful statements.

If no response exists, blanks are written to the S variable and job variable. The S variable or job variable is not supplied with values if SESADM cannot generate or supply them.

> ¡ You should delete the content of the S variable or job variable before entering an administration statement as in some cases (e.g. syntax error is detected by SDF) SESAM/SQL is not assigned control and the content of the S variable/job variable remains unchanged.

5.1.3.13 SDF standard statement STEP

The SDF standard statement STEP marks a section of administration statements within a SESADM input file in procedure or batch mode.

In the event of a faulty statement, the spin-off mechanism is triggered. This means that all subsequent statements up to the next STEP statement are ignored. If no STEP statement is found before the END statement, SESADM is terminated with a warning.

If syntax errors occur in all SESADM statement or semantic errors occur in the SESADM statement PREPARE-FOREIGN-COPY, statement input can be continued with the next statement section.

Syntax errors

The example below shows the use of the STEP statement in the case of syntax errors.

```
/START-SESAM-ADMINISTRATION
//START-DBH-ADMINISTRATION PASSWORD='ADM'
//MODIFY-TRANSACTION-SECURITY SELECT=*LOCK-DATE(183)         1.
//STEP
//MODIFY-TRANSACTION-SECURITY SELECT=*INACTIVITY-TIME(1000)  1.
//STEP
//MODIFY-TRANSACTION-SECURITY SELECT=*LOCK-ESCALATION( -     2.
// NUMBER-RECORDS=12345678 -
// ,PERCENTAGE-RECORDS=75 -
// ,NUMBER-INDEX-VALUES=7654321)
//END
```

1. Each of the first two statements contains a syntax error. A specific message is output, and it is displayed that all statement will be skipped until STEP is recognized (message CMD0230). As this fact is already reported by SDF, no further message to this effect is issued by SESADM. Only message SEA7003 is issued which displays the syntax error.
2. This statement with correct syntax is always executed.

Semantic errors in PREPARE-FOREIGN-COPY

In the case of the following semantic errors in the PREPARE-FOREIGN-COPY statement, the statement sequence continues with the next STEP statement:

- The file specified with SPACE-NAMES does not exist (message SEA7026)
- The file specified with SPACE-NAMES is empty (message SEA7027)
- The file specified with SPACE-NAMES contains more than 999 lines (message SEA7028)
- The file specified with SPACE-NAMES contains formally incorrect space names (message SEA7029)
- A DMS error occurred in the file specified with SPACE-NAMES (message SEA7030)

If no subsequent STEP exists, SESADM terminates with a warning message (SEZ4603).

In addition to the specific message, SESADM shows in message SEA7034 that all statements up to the next STEP or END will be skipped.

5.2 Administration statements and commands

Administration statements and commands are available to you for dynamic administration:

- DBH-specific administration statements and commands allow you to monitor and influence the DBH session. You can also modify a large number of DBH options during ongoing operation, see for example "MODIFY-MSG-OUTPUT". You can save the current DBH options to a file and use them again in the next DBH session, see "SAVE-DBH-OPTIONS".
- In distributed processing with SESAM/SQL-DCN, SESDCN-specific administration statements and commands allow you to administer the SESDCN distribution component.

Most administration statements and commands take effect immediately they are issued. With some of them there may be a wait until the required resources are available.

Depending on the interface you use for administration, you issue administration statements or administration commands.

- Administration statements are for administration with SESADM. Their syntax complies with SDF syntax rules (e. g. SHOW-INACTIVE-SQL-USERS). They can be abbreviated in accordance with SDF conventions.
- When you carry out administration using INFORM-PROGRAM or a CALL DML program, you use administration commands, whose syntax is not subject to SDF rules (e.g. USER,INACT).

Administration statements and administration commands offer the same functionality.

The following sections briefly describe the syntax and functions of the administration statements. You will find a list of administration statements and the equivalent commands as of "Equivalent administration statements and commands".

Peculiarity of the linked-in DBH

You have to use administration commands to administer the linked-in DBH with INFORM-PROGRAM or an integrated CALL DML program. It is not possible to use SESADM to administer the linked-in DBH.

An administration command to the linked-in DBH is not processed until the next statement is issued in the application program and control is thus passed to the DBH. In contrast, administration statements and commands to the independent DBH are processed immediately.

5.2.1 Summary of contents

You will find below an overview of all the administration commands, divided up into groups according to their purpose.

5.2.1.1 DBH administration statements

The DBH administration statements are divided up into three tables as follows:

- The table 15 describes all administration statements that provide user-specific information.
- The table 16 contains all administration statements that display or change the DBH start statements and options issued.
- The table 17 contains all other statements that can be used to control the DBH session.

Administration statement	Brief description
SHOW-USER-SPACES	Displays the spaces used by the user and thus locked
SHOW-CALL-DML-SUBORDERS	Displays the number of active CALL DML requests of selected users
SHOW-CATALOG-USERS	Displays the number of active users within a TA for selected databases
SHOW-INACTIVE-SQL-USERS	Displays all inactive SQL users
SHOW-SPACE-USERS	Displays all active users of a selected space within a TA together with relevant additional information
SHOW-TRANSACTIONS	Displays all open transactions of selected users with relevant additional information
SHOW-USERS	Displays all active users together with relevant additional information
SHOW-CATID-LIST	Displays the current CATID list

Table 15: DBH administration statements for user-specific information

Administration statement	Brief description
ADD-OLD-TABLE-CATALOG-ENTRY	Adds an entry to the CALL DML table catalog
ADD-SQL-DB-CATALOG-ENTRY	Adds an entry to the SQL database catalog
MODIFY-ADMINISTRATION	Changes the administration authorization
MODIFY-CATALOG-ACCESS-RIGHTS	Changes the access rights for the specified database
MODIFY-MAIL-PARAMETERS	Changes the parameters for the email output of the DBH
MODIFY-MSG-OUTPUT	Changes the DBH output
MODIFY-OLD-TABLE-CATALOG-LIMIT	Changes the maximum number of entries in the CALL-DML table catalog
MODIFY-RECOVER-OPTIONS	Changes the options for subsequent RECOVER or REFRESH runs

MODIFY-REQUEST-CONTROL	Changes the parameters for priority control
MODIFY-RESTART-CONTROL	Influences the duration of any restart
MODIFY-RETRIEVAL-CONTROL	Changes the criterion for interrupting retrieval statements
MODIFY-SECURITY	Changes the maximum permissible number of password violations
MODIFY-SERVICE-TASKS	Changes the number and attributes of service tasks
MODIFY-SESSION-LOGGING-ID	Changes the identification of session-related files
MODIFY-SQL-SORT-LIMIT	Changes the limit value for the number of sort records found in a cursor table
MODIFY-STORAGE-SIZE	Changes the maximum size of transfer and work containers
MODIFY-SUBORDER-LIMIT	Changes the limit value for the maximum number of SQL scans and/or logical files of CALL DML requests
MODIFY-TRANSACTION-SECURITY	Changes the transaction security parameters
RECONFIGURE-DBH-SESSION	Changes DBH options dynamically
RELOAD-DBH-SESSION	Reloads DBH modules and changes DBH options
REMOVE-OLD-TABLE-CATALOG-ENTRY	Removes an entry from the CALL DML table catalog
REMOVE-SQL-DB-CATALOG-ENTRY	Removes an entry from the SQL database catalog
REUSE-OLD-TABLE-CATALOG-ENTRY	Create valid reference to an existing table entry in the CALL DML table catalog
REUSE-PARTITIONS	Restores the availability of partitions
SAVE-DBH-OPTIONS	Saves the current DBH options
SET-ACCOUNTING-PARAMETER	Controls logging of request accounting for the RAV procedure
SET-REQUEST-CONTROL	Turns priority control on and off
SET-USER-INACTIVE-TIME	Sets a time period after which a user's open but inactive transactions are rolled back
SHOW-DBH-MEDIA-CATALOG	Displays the currently valid DBH option MEDIA-CATALOG
SHOW-DBH-OPTIONS	Displays the currently valid DBH options except for MEDIA-CATALOG
SHOW-MAIL-PARAMETERS	Outputs parameters for the email output of the DBH

SHOW-OLD-TABLE-CATALOG-ENTRIES	Displays the entries in the CALL DML table catalog
SHOW-PARTITIONS	Displays the availability of partitions
SHOW-SQL-DB-CATALOG-ENTRIES	Displays the entries in the SQL database catalog

Table 16: Administration statements for DBH start statements and options

Administration statement	**Brief description**
ABORT-LOCK-SEQUENCE	Terminate lock sequence of another user
ASSIGN-SYSLST	Switches a SYSLST file
BEGIN-LOCK-SEQUENCE	Starts a lock sequence
CANCEL-STATEMENT	Cancels a DML statement
CHANGE-CATLOG	Switches a CAT-LOG file and DA-LOG files
CHANGE-DALOG	Switches DA-LOG files
CLOSE-SPACE	Close user space physically
COMMIT-PTC-TRANSACTION	Commits a PREPARE-TO-COMMIT transaction
CREATE-DUMP	Creates a main-memory dump
CREATE-SERVICE-TASK-DUMP	Creates a main-memory dump of a service task
END-FOREIGN-COPY	Cancel “copy pending” state on spaces after foreign copy. Changes the database status after PREPARE-FOREIGN-COPY with physical closure of the database files.
END-LOCK-SEQUENCE	Terminate lock sequence
HOLD-TRANSACTION-ADMISSION	Suspends admission of further transactions
MODIFY-CATID-LIST	Updates the CATID list
PREPARE-FOREIGN-COPY	Closes a database in order to be able to create a foreign copy
RELEASE-USER-RESOURCES	Releases all a user's resources
RESUME-TRANSACTION-ADMISSION	Permits resumption of transaction admission
ROLLBACK-PTC-TRANSACTION	Rolls back a user's PREPARE-TO-COMMIT transaction
ROLLBACK-TRANSACTION	Rolls back a user's transaction
SET-DBH-MSG-TRACE	Controls the logging of DBH messages
SET-DIAGNOSIS-DUMP-PARAMETER	Controls the creation of a dump
SET-SAT-SUPPORT	

	Controls the logging of security relevant events with SAT
SET-SESSION-DIAGNOSIS	Display for controlling the deadlock analysis
SET-SQL-DB-CATALOG-STATUS	Changes the status of a database
SET-TUNING-TRACE	Controls request logging
STOP-DBH	Terminates a DBH session

Table 17: Other DBH administration statements that control database operation

5.2.1.2 SESDCN administration statements

The following table provides an overview of all the administration statements that are available for the administration of SESDCN:

Administration statement	Brief description
ADD-DISTRIBUTION-RULE-ENTRY	Adds a database to the distribution rule
CREATE-DUMP	Creates a main-memory dump
HOLD-TRANSACTION-ADMISSION	Suspends admission of further transactions
HOLD-USER-ADMISSION	Suspends admission of further users
MODIFY-ADMINISTRATION	Changes the administration authorization
MODIFY-DISTRIBUTION-RULE-ENTRY	Changes the host name in the distribution rule
REMOVE-DISTRIBUTION-RULE-ENTRY	Removes a database entry or database entries from the distribution rule
RESUME-TRANSACTION-ADMISSION	Permits resumption of transaction admission
RESUME-USER-ADMISSION	Resumes admission of new users
ROLLBACK-TRANSACTION	Rolls back a user's transaction
SET-USER-CALL-TRACE	Controls the logging of the statements of selected users
SET-USER-MSG-TRACE	Controls the logging of SESAM/SQL messages of selected users
SHOW-DISTRIBUTION-RULE-ENTRIES	Displays active databases entered in the distribution rule
SHOW-TRANSACTIONS	Displays all open transactions of selected users with relevant additional information
SHOW-USERS	Displays all active users together with relevant additional information
STOP-DCN	Terminates SESDCN

Table 18: SESDCN administration statements

5.2.1.3 Administration statements for changing DBH options

There are three categories of administration statements for changing DBH options in the ongoing DBH session:

1. Some administration statements change DBH options which have no influence on the storage structure and mode of operation of the DBH.
2. The administration statement RECONFIGURE-DBH-SESSION reconfigures the ongoing DBH session and changes the following DBH options:
   - COLUMNS
   - CURSOR-BUFFER
   - SPACES
   - SQL-SUPPORT (CURSORS and PLANS operands)
   - SYSTEM-DATA-BUFFER
   - TRANSACTION-SECURITY (MAX-ISOLATION-LEVEL operand)
   - USER-DATA-BUFFER
3. The administration statement RELOAD-DBH-SESSION reloads the DBH modules, reconfigures the DBH session and, if required, changes the following DBH options:
   - DBH-TASKS
   - SYSTEM-THREADS
   - THREADS
   - USERS

The change to the DBH options is optional.

The table below shows which options can be changed with which administration statement.

Higher-level option	**Lower-level option Operand**	**Administration statement to change it**
ADMINISTRATION	ADMINISTRATOR MSG-OUTPUT SECURITY	MODIFY-ADMINISTRATION MODIFY-MSG-OUTPUT MODIFY-SECURITY
CPU-RESOURCES	DBH-TASKS SERVICE-TASKS	RELOAD-DBH-SESSION MODIFY-SERVICE-TASKS
FILE-RESOURCES	SESSION-LOGGING-ID	MODIFY-SESSION-LOGGING-ID
RECOVER-OPTIONS		MODIFY-RECOVER-OPTIONS
STORAGE-SIZE	CURSOR-BUFFER TRANSFER-CONTAINER WORK-CONTAINER SYSTEM-DATA-BUFFER USER-DATA-BUFFER	RECONFIGURE-DBH-SESSION MODIFY-STORAGE-SIZE MODIFY-STORAGE-SIZE RECONFIGURE-DBH-SESSION RECONFIGURE-DBH-SESSION
SYSTEM-LIMITS	COLUMNS OLD-TABLE-CATALOG	RECONFIGURE-DBH-SESSION MODIFY-OLD-TABLE-CATALOG-LIMIT

	SPACES	RECONFIGURE-DBH-SESSION
	SQL-SUPPORT	SET-USER-INACTIVE-TIME
	INACTIVITY-TIME	
	CURSORS	RECONFIGURE-DBH-SESSION
	INTERNAL-SORT-LIMIT	MODIFY-SQL-SORT-LIMIT
	PLANS	RECONFIGURE-DBH-SESSION
	SUBORDERS	MODIFY-SUBORDER-LIMIT
	MAXIMUM	
	SYSTEM-THREADS	RELOAD-DBH-SESSION
	THREADS	RELOAD-DBH-SESSION
	USERS	RELOAD-DBH-SESSION
SYSTEM-STRATEGIES	REQUEST-CONTROL	MODIFY-REQUEST-CONTROL
	QUEUE-RANGE	MODIFY-REQUEST-CONTROL
	QUEUE-PRIORITY	MODIFY-REQUEST-CONTROL
	RESTART-CONTROL	
	TALOG-LIMIT	MODIFY RESTART-CONTROL
	SYSTEM-DATA-BUFFER-LIMIT	MODIFY RESTART-CONTROL
	USER-DATA-BUFFER-LIMIT	MODIFY RESTART-CONTROL
	RETRIEVAL-CONTROL	
	INDEX-EVALUATION	MODIFY-RETRIEVAL-CONTROL
	USER-DATA-ACCESS	MODIFY-RETRIEVAL-CONTROL
	ABORT-EXECUTION	MODIFY-RETRIEVAL-CONTROL
	TRANSACTION-SECURITY	
	LOCK-TIME	MODIFY-TRANSACTION-SECURITY
	INACT-TIME	MODIFY-TRANSACTION-SECURITY
	MAX-ISOLATION-LEVEL	RECONFIGURE-DBH-SESSION
	LOCK-ESCALATION	MODIFY-TRANSACTION-SECURITY

5.2.2 Equivalent administration statements and commands

This section contains two tables listing all the administration statements (for administration using SESADM) and administration commands (for administration using INFORM-PROGRAM or a CALL DML program), and indicating which statements and commands are equivalent.

The table 20 contains the administration statements in alphabetical order. The table 21 has the administration commands in alphabetical order.

Administration statement in SDF format	**Administration command in ISP format and at the CALL DML interface**
ABORT-LOCK-SEQUENCE	LSQ,A[BORT]
ADD-DISTRIBUTION-RULE-ENTRY	UPD [1]
ADD-OLD-TABLE-CATALOG-ENTRY	OPT,CDBC-INS
ADD-SQL-DB-CATALOG-ENTRY	OPT,DBC-INS
ASSIGN-SYSLST	SYSLST
BEGIN-LOCK-SEQUENCE	LSQ,B[EGIN]
CANCEL-STATEMENT	CANCEL
CHANGE-CATLOG	CAW
CHANGE-DALOG	DAW
CLOSE-SPACE	CLOSE,S[PACE]
COMMIT-PTC-TRANSACTION	CPTC
CREATE-DUMP	DUMP[,STOP]
CREATE-SERVICE-TASK-DUMP	DUMP,SVT
END-FOREIGN-COPY	FCOPY,E[ND]
END-LOCK-SEQUENCE	LSQ,E[ND]
HOLD-TRANSACTION-ADMISSION	HOLD-TA (with SESDCN: HOLD,TA) WAIT,HOLD [2]
HOLD-USER-ADMISSION	HOLD,USER
MODIFY-ADMINISTRATION	OPT,ADM
MODIFY-CATALOG-ACCESS-RIGHTS	OPT,ACCESS
MODIFY-CATID-LIST	UPD,C[ID-LIST]
MODIFY-DISTRIBUTION-RULE-ENTRY	UPDPRO
MODIFY-MSG-OUTPUT	OPT,MSG-OUTPUT

MODIFY-MAIL-PARAMETERS	[3]
MODIFY-OLD-TABLE-CATALOG-LIMIT	OPT,CDBC-LIMIT
MODIFY-RECOVER-OPTIONS	OPT,RECOVER
MODIFY-REQUEST-CONTROL	OPT,REQ-CON
MODIFY-RESTART-CONTROL	OPT,RESTART
MODIFY-RETRIEVAL-CONTROL	OPT,RET-CON
MODIFY-SECURITY	OPT,SECURITY
MODIFY-SERVICE-TASKS	OPT,SVT
MODIFY-SESSION-LOGGING-ID	OPT,LOG-ID
MODIFY-SQL-SORTLIMIT	OPT,SORTLIMIT
MODIFY-STORAGE-SIZE	OPT,STO-SIZE
MODIFY-SUBORDER-LIMIT	OPT,SUBORDERS
MODIFY-TRANSACTION-SECURITY	OPT,TA-SEC
PREPARE-FOREIGN-COPY	FCOPY,P[REPARE]
RECONFIGURE-DBH-SESSION	[3]
RELEASE-USER-RESOURCES	FREE FREE,USER,INACT
RELOAD-DBH-SESSION	[3]
REMOVE-DISTRIBUTION-RULE-ENTRY	DEL,DB DEL,PRO-NAME
REMOVE-OLD-TABLE-CATALOG-ENTRY	OPT,CDBC-DEL
REMOVE-SQL-DB-CATALOG-ENTRY	OPT,DBC-DEL
RESUME-TRANSACTION-ADMISSION	GO-TA (at SESDCN: GO,TA)
RESUME-USER-ADMISSION	GO,USER
REUSE-OLD-TABLE-CATALOG-ENTRY	OPT,CDBC-REUSE
REUSE-PARTITIONS	PARTITION,REUSE
ROLLBACK-PTC-TRANSACTION	RPTC
ROLLBACK-TRANSACTION	RESET
SAVE-DBH-OPTIONS	OPT,SAVE
SET-ACCOUNTING-PARAMETER	ACC,TP
SET-DBH-MSG-TRACE	TRACE

SET-DIAGNOSIS-DUMP-PARAMETER	OPT,DIAG,diag-spec
SET-REQUEST-CONTROL	OPT,REQ-CON or [,ON/,OFF]
SET-SAT-SUPPORT	OPT,SAT
SET-SQL-DB-CATALOG-STATUS	OPT,DBC-UPD
SET-TUNING-TRACE	COS
SET-USER-CALL-TRACE	TRACE=PAR,TYPE=CALL... TRACE={ON|OFF...}
SET-USER-INACTIVE-TIME	OPT,INACT
SET-USER-MSG-TRACE	TRACE=PAR,TYPE=MSG... TRACE={ON|OFF...}
SHOW-CALL-DML-SUBORDERS	USER
SHOW-CATALOG-USERS	USER,C[ATALOG]
SHOW-CATID-LIST	INF,C[ID-LIST]
SHOW-DBH-MEDIA-CATALOG	OPT,MEDIA
SHOW-DBH-OPTIONS	OPT
SHOW-DISTRIBUTION-RULE-ENTRIES	USER,CATALOG
SHOW-INACTIVE-SQL-USERS	USER,INACT
SHOW-OLD-TABLE-CATALOG-ENTRIES	OPT,CDBC-INFO
SHOW-MAIL-PARAMETERS	[3]
SHOW-PARTITIONS	PARTITION,INF
SHOW-SPACE-USERS	USER,SPACE
SHOW-SQL-DB-CATALOG-ENTRIES	OPT,DBC-INFO
SHOW-TRANSACTIONS	USER,TA
SHOW-USERS	USER,PR
SHOW-USER-SPACES	SPACE
STOP-DBH	STOP[,KEEP] STOP,DELETE
STOP-DCN	STOP

Table 20: List of equivalent administration statements and commands

[1]only in ISP format

[2]only on the CALL-DML interface

[3]not available in ISP format and not available on the CALL-DML interface

Administration command in ISP format and at the CALL DML interface	**Administration statement in SDF format**
ACC,TP	SET-ACCOUNTING-PARAMETER
CAW	CHANGE-CATLOG
CANCEL	CANCEL-STATEMENT
CLOSE,S[PACE]	CLOSE-SPACE
COS	SET-TUNING-TRACE
CPTC	COMMIT-PTC-TRANSACTION
DAW	CHANGE-DALOG
DEL,DB DEL,PRO-NAME	REMOVE-DISTRIBUTION-RULE-ENTRY
DUMP[,STOP]	CREATE-DUMP
DUMP,SVT	CREATE-SERVCIE-TASK-DUMP
FCOPY,E[ND]	END-FOREIGN-COPY
FCOPY,P[REPARE]	PREPARE-FOREIGN-COPY
FREE FREE,USER,INACT	RELEASE-USER-RESOURCES
GO-TA (at SESDCN: GO,TA)	RESUME-TRANSACTION-ADMISSION
GO,USER	RESUME-USER-ADMISSION
HOLD-TA (with SESDCN: HOLD,TA)	HOLD-TRANSACTION-ADMISSION
HOLD,USER	HOLD-USER-ADMISSION
INF,C[ID-LIST]	SHOW-CATID-LIST
LSQ,A[BORT]	ABORT-LOCK-SEQUENCE
LSQ,B[EGIN]	BEGIN-LOCK-SEQUENCE
LSQ,E[ND]	END-LOCK-SEQUENCE
OPT	SHOW-DBH-OPTIONS
OPT,ACCESS	MODIFY-CATALOG-ACCESS-RIGHTS
OPT,ADM	MODIFY-ADMINISTRATION
OPT,CDBC-DEL	REMOVE-OLD-TABLE-CATALOG-ENTRY
OPT,CDBC-INFO	SHOW-OLD-TABLE-CATALOG-ENTRIES

OPT,CDBC-INS	ADD-OLD-TABLE-CATALOG-ENTRY
OPT,CDBC-LIMIT	MODIFY-OLD-TABLE-CATALOG-LIMIT
OPT,CDBC-REUSE	REUSE-OLD-TABLE-CATALOG-ENTRY
OPT,DBC-DEL	REMOVE-SQL-DB-CATALOG-ENTRY
OPT,DBC-INFO	SHOW-SQL-DB-CATALOG-ENTRIES
OPT,DBC-INS	ADD-SQL-DB-CATALOG-ENTRY
OPT,DBC-UPD	SET-SQL-DB-CATALOG-STATUS
OPT,DIAG,diag-spec	SET-DIAGNOSIS-DUMP-PARAMETER
OPT,INACT	SET-USER-INACTIVE-TIME
OPT,LOG-ID	MODIFY-SESSION-LOGGING-ID
OPT,MEDIA	SHOW-DBH-MEDIA-CATALOG
OPT,MSG-OUTPUT	MODIFY-MSG-OUTPUT
OPT,RECOVER	MODIFY-RECOVER-OPTIONS
OPT,REQ-CON	MODIFY-REQUEST-CONTROL
OPT,REQ-CON	SET-REQUEST-CONTROL
OPT,RESTART	MODIFY-RESTART-CONTROL
OPT,RET-CON	MODIFY-RETRIEVAL-CONTROL
OPT,SAT	SET-SAT-SUPPORT
OPT,SAVE	SAVE-DBH-OPTIONS
OPT,SECURITY	MODIFY-SECURITY
OPT,SORTLIMIT	MODIFY-SQL-SORTLIMIT
OPT,STO-SIZE	MODIFY-STORAGE-SIZE
OPT,SUBORDERS	MODIFY-SUBORDER-LIMIT
OPT,SVT	MODIFY-SERVICE-TASKS
OPT,TA-SEC	MODIFY-TRANSACTION-SECURITY
PARTITION,REUSE	REUSE-PARTITIONS
PARTITION,INF	SHOW-PARTITIONS
RESET	ROLLBACK-TRANSACTION
RPTC	ROLLBACK-PTC-TRANSACTION
SPACE	SHOW-USER-SPACES

STOP,DELETE STOP[,KEEP]	STOP-DBH
STOP	STOP-DCN
SYSLST	ASSIGN-SYSLST
TRACE=PAR,TYPE=CALL... TRACE={ON|OFF...}	SET-USER-CALL-TRACE
TRACE=PAR,TYPE=MSG... TRACE={ON|OFF...}	SET-USER-MSG-TRACE
TRACE={ON|OFF...}	SET-DBH-MSG-TRACE
UPD [1]	ADD-DISTRIBUTION-RULE-ENTRY
UPD,C[ID-LIST]	MODIFY-CATID-LIST
UPDPRO	MODIFY-DISTRIBUTION-RULE-ENTRY
USER	SHOW-CALL-DML-SUBORDERS
USER,C[ATALOG]	SHOW-CATALOG-USERS
USER,CATALOG	SHOW-DISTRIBUTION-RULE-ENTRIES
USER,INACT	SHOW-INACTIVE-SQL-USERS
USER,PR	SHOW-USERS
USER,SPACE	SHOW-SPACE-USERS
USER,TA	SHOW-TRANSACTIONS
WAIT,HOLD [2]	HOLD-TRANSACTION-ADMISSION
[3]	MODIFY-MAIL-PARAMETERS SHOW-MAIL-PARAMETERS RECONFIGURE-DBH-SESSION RELOAD-DBH-SESSION

Table 21: List of equivalent administration commands and statements

[1]only in ISP format
[2]only on the CALL-DML interface
[3]not available in ISP format and not available on the CALL-DML interface

5.2.3 Descriptions in alphabetical order

This section provides a detailed description of the syntax and functions of the administration statements and commands.

5.2.3.1 Description format

The descriptions of the administration statements all have the same structure. There is an entry for each administration statement;;The entry's heading is the name of the administration statement.

Each entry consists of several parts.

An entry may not include all the parts if some have no meaning for that statement. The most important sections are described overleaf.

Name of the administration statement

Brief description

Scope of validity

This section indicates the program to which the administration statement belongs, i.e. whether it is:

- a DBH administration statement (DBH administration)
- an SESDCN administration statement (SESDCN administration)
- a DBH and an SESDCN administration statement (DBH and SESDCN administration)

See also

This section contains references to DBH start statements, SESDCN control statements, DBH or DCN options or administration statements that are relevant to the statement described because they provide additional information.

Function

This section provides a detailed description of the administration statement.

It is followed by a syntax diagram.

Name of the administration statement
Operands

Operands

This section contains explanations of the various operands. The operands are dealt with in the order in which they occur in the syntax diagram.

Output in S variables

Describes the information output of SHOW statements in S variables.

Example

Provides an example of the information output of SHOW statements to SYSOUT.

Administration command in ISP format and at the CALL DML interface

This section describes the syntax of the administration command that has the same function as the administration statement.

The syntax of the administration command is only explained if it is not self-explanatory or if its function deviates from that of the administration statement.

5.2.3.2 Syntax for the identification of users

In some administration statements and commands you have to select one or more users to which the statement or command is to apply.

The identification of users is represented in abbreviated form in the syntax diagrams:

- <user-identification> in the syntax of administration statements
- *user-identification* in the syntax of administration commands.

In fact, the identification of users consists of up to four different partial qualifications.

A <user-identification> is partially qualified if not all data was specified for a user specification. A partially qualified <user-identification> can do more than identify a user. Depending on the administration command, a partially qualified <user-identification> is only permitted if it identifies exactly one user.

A <user-identification> is fully qualified when the user specifies all the relevant partial qualifications for unique identification.

Syntax of <user-identification> in administration statements

The complete syntax diagram of <user-identification> is as follows:

```
<user-identification> = *TSN(...) / *BY-SELECT(...)

  *TSN(...)
    | TSN = <alphanum-name 4..4>
  *BY-SELECT(...)
    | HOST-NAME = *NONE / <text 1..8>
    | ,APPLICATION-NAME = *NONE / *TSN(...) / <text 1..8> / <x-text 1..8>
    |       *TSN(...)
    |             |     TSN = <alphanum-name 4..4>
    | ,CUSTOMER-NAME = *NONE / <text 1..8>
    | ,CONVERSATION-ID = *NONE / <x-text 1..8>
```

Operands

<user-identification> =

Identifies an SQL or CALL DML user or a group of such users.

<user-identification> = *TSN(...)
User's short ID.
The short ID is adequate for users from a timesharing program on the same computer.

TSN = <alphanum-name 4..4>

Task sequence number of the timesharing user's task

<user-identification> = *BY-SELECT(...)
Identifies the user (timesharing user or application user) by means of the user name, which consists of different partial qualifications.

HOST-NAME = *NONE / <text 1..8>
Symbolic host name. The default is *NONE (i.e. no entry is made for this operand).

APPLICATION-NAME = *NONE / *TSN(...) / <text 1..8>
Name of the application. The default is *NONE (i.e. no entry is made for this operand).The name of the application depends on the user:

- JDBC: „DBAxxxxx“ (xxxxx = DBA server's 5-digit port number)
- ADO.NET: „ADOxxxxx“ (xxxxx = DBA server's 5-digit port number)
- PHP: „PHPxxxxx“ (xxxxx = DBA server's 5-digit port number)
- otherwise: name of the application

APPLICATION-NAME = *TSN(...)
The name of the application corresponds to the TSN of the timesharing user's task.

TSN = <alphanum-name 4..4>
Task sequence number of the timesharing user's task

CUSTOMER-NAME = *NONE / <text 1..8> / <x-text 1..8>
Name of the user.
The default is *NONE (i.e. no entry is made for this operand).
The name of the user depends on the operating mode:

- TIAM: program name
- openUTM: KDCSIGN name
- DCAM: any name.
- JDBC, ADO.NET, PHP: IP address of the JDBC, ADO.NET or PHP client in hexadecimal form (e.g. 7FC864C8 for IP address 127.200.100.200)

CONVERSATION-ID = *NONE / <x-text 1..8>
UTM conversation ID.
The default is *NONE (i.e. no entry is made for this operand).
The operation ID depends on the user:

- JDBC, ADO.NET, PHP: connection ID of the JDBC,ADO.NET or PHP client to the DBA server
- openUTM: conversation number of a UTM conversation (must be entered whenever the user is an asynchronous UTM conversation)

Unique user identification

In some user-specific administration statements, it is enough to specify a partially qualified <user-identification>. Other statements require that the user be identified uniquely. In such cases, you must specify all the relevant partial qualifications of <user-identification>. These cases are known as fully qualified.

Which partial qualifications of <user-identification> are relevant depends on the answers to the following questions:

- Is the user local or remote?
- Is the user a timesharing, application user, JDBC, ADO.NET or PHP?

The following overview indicates what you have to enter to identify a user uniquely:

User	Entry

Local user in timesharing mode	`TSN = <alphanum-name 4..4>`
Remote user in timesharing mode	`HOST-NAME = <text 1..8>` `,APPLICATION-NAME = *TSN(...)`
Local user in transaction mode (non-openUTM)	`APPLICATION-NAME = <text 1..8>` `,CUSTOMER-NAME = <text 1..8>`

Local user in transaction mode (openUTM)	`APPLICATION-NAME = <text 1..8>`
	`,CUSTOMER-NAME = <text 1..8>`
	`[,CONVERSATION-ID = <x-text 1..8>]`

Remote user in transaction mode (non-openUTM)	`HOST-NAME = <text 1..8>`
	`,APPLICATION-NAME = <text 1..8>`
	`,CUSTOMER-NAME = <text 1..8>`

Remote user in transaction mode (openUTM)	`HOST-NAME = <name 1..8>`
	`,APPLICATION-NAME = <text 1..8>`
	`,CUSTOMER-NAME = <text 1..8>`
	`[,CONVERSATION-ID = <x-text 1..8>]`

JDBC
ADO.NET
PHP
	`HOST-NAME = <name 1..8>`
	`,APPLICATION-NAME = <text 1..8>`
	`,CUSTOMER-NAME = <x-text 1..8>`
	`,CONVERSATION-ID = <x-text 1..8>`

Table 22: Unique user identification

Syntax of user-identification in administration commands

The syntax of *user-identification* in the administration commands is as follows:

```
user-identification ::= {user-spec|tsn-spec}
user-spec ::= USER=[host],[{appl|tsn-spec}],[user][,vg-id]
tsn-spec ::= TSN=tsn
```

In cases in which unique identification of a user is required, you must specify all the relevant partial qualifications. The following options are available to do this:

User	Entry
Local user in timesharing mode	`TSN=`*`tsn`*
Remote user in timesharing mode	`USER=`*`host`*`,TSN=`*`tsn`*`,`

Local user in transaction mode (nonopenUTM)	USER=,*appl*,[*user*]
Local user in transaction mode (openUTM)	USER=,*appl*,[*user*][,*vg-id*]
Remote user in transaction mode (non-openUTM)	USER=*host*,*appl*,[*user*]
Remote user in transaction mode (openUTM)	USER=*host*,*appl*,[*user*][*vg-id*]

Table 23: Partial qualifications with unique user identification

5.2.3.3 Identification of system-internal users

The identifications shown in the table below allow requests of system-internal users to be identified, together with their type. These identifications can be output with the corresponding SHOW functions if there are any internal requests.

Identification	Type of the system-internal user

`HOST-NAME = SESAM`
`,APPLICATION-NAME = SEND`
`,CUSTOMER-NAME = MESSAGE`
	System-internal administrator
	(this is an internal user who issues administration commands via INFORM-PROGRAM)

`HOST-NAME = SESAM`
`,APPLICATION-NAME = RESTART`
`,CUSTOMER-NAME = <x-text 1..8>`
	Internal user who, in the event of a restart, delays the rollback of transactions until the resumption of normal operation

`HOST-NAME = SESAM`
`,APPLICATION-NAME = AUTTRAN`
`,CUSTOMER-NAME = <x-text 1..8>`
	Internal user (autonomous transaction)

`HOST-NAME = SYSTEM`
`,APPLICATION-NAME = SYSTEM`
`,CUSTOMER-NAME = SYSTEM`
	All other internal users

Table 24: Unique user identification

5.2.3.4 Syntax for output in S variables

Information output in S variables is controlled using the STRUCTURE-OUTPUT operand in the SHOW statements of SESADM:

```
SHOW- . . .

 STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

     <structured-name 1..20>(...)

         |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT =

In S procedures, specifies whether the information output of the SHOW statements is to be to S variables.

STRUCTURE-OUTPUT = *NONE
No output in S variables.

STRUCTURE-OUTPUT = *SYSINF
Output on the S variable stream SYSINF to which an S variable is assigned.

STRUCTURE-OUTPUT = <structured-name 1..20>(...)
Output in the S variable with the specified name.
The S variable must have been defined beforehand in the S procedure.
The structure of the output depends on the particular statement.

WRITE-MODE = *REPLACE / *EXTEND
Defines whether the content of the S variable should be overwritten (*REPLACE) or updated (*EXTEND).

5.2.3.5 Alphabetical reference section

The administration statements are described below in alphabetical order.

The syntax of the equivalent administration command is given after the description of each administration statement.

> Note when entering administration commands that the operands must not be separated by blanks.

5.2.3.6 ABORT-LOCK-SEQUENCE

This statement terminates the lock sequence of another user.

Scope of validity

DBH administration

See also

- DBH administration statement BEGIN-LOCK-SEQUENCE on "BEGIN-LOCK-SEQUENCE"
- DBH administration statement END-LOCK-SEQUENCE on "END-LOCK-SEQUENCE"

Function

The ABORT-LOCK-SEQUENCE statement terminates a lock sequence that is uniquely identified by specifying the identification of the user.

All locks requested on databases and spaces during the lock sequence are released when the lock sequence is terminated.

> i You cannot specify your own identification or that of the system-internal administrator (see "Identification of system-internal users").

The system administrator can protect himself or herself against the termination of his or her lock sequence by changing the password or administration authorization. The administrator can make these changes by using the DBH option ADMINISTRATOR or the DBH administration statement MODIFY-ADMINISTRATION (see "ADMINISTRATOR" or "MODIFY-ADMINISTRATION").

You must observe the following points when using the ABORT-LOCK-SEQUENCE statement:

- The administration statement is rejected if the SESAM DBH was loaded without transaction management.
- The user addressed by means of the ABORT-LOCK-SEQUENCE administration statement must be in a lock sequence. You can use the SHOW-USERS statement (see "SHOW-USERS") to find out whether there is a lock sequence active for a specific user.
- On input of the next statement, the addressed user receives a message indicating that the lock sequence has been terminated.

ABORT-LOCK-SEQUENCE
USER = <user-identification>

Operands

USER = <user-identification>

Identifies the user whose lock sequence is to be terminated. The identification must be fully qualified (see the section “Syntax for the identification of users”).

Administration command in ISP format and at the CALL DML interface

The LSQ,ABORT administration command corresponds to the ABORT-LOCK-SEQUENCE administration statement here.

```
LSQ,A[BORT],user-identification
```

user-identification must be unique. You therefore have to specify all the relevant partial qualifications.

5.2.3.7 ADD-DISTRIBUTION-RULE-ENTRY

This statement adds a database to the distribution rule.

Scope of validity

SESDCN administration

See also

- SESDCN control statement ADD-DISTRIBUTION-RULE-LIST on "ADD-DISTRIBUTION-RULE-LIST"
- SESDCN control statement ADD-NETWORK-LINK-LIST on "ADD-NETWORK-LINK-LIST"

Function

The ADD-DISTRIBUTION-RULE-ENTRY statement adds a database entry to the distribution rule.

The logical name of the database you want to add to the distribution rule must be unique in the distributed application and thus throughout the network. If the distribution rule already contains an entry with the same logical database name, the administration statement is rejected.

Together with the entries created implicitly by SESAM/SQL (see the note on "Entering SESDCN control statements") the distribution rule may contain a maximum of 340 entries. You can, however, explicitly enter at least 300 databases and their associated networks. If the distribution rule is full, SESAM/SQL-DCN rejects the administration statement.

The administration statement is also rejected if a DBH would be assigned to several SESDCNs as a result of accepting it.

ADD-DISTRIBUTION-RULE-ENTRY
CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers> ,DBH-NAME = *BLANK / <alphanum-name 1..1> ,DCN-NAME = *BLANK / <alphanum-name 1..1> ,CONFIGURATION-NAME = *STD / <alphanum-name 1..1> / *BLANK ,HOST-NAME = *STD / <name 1..8>

Operands

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical database name of the SESAM/SQL database or CALL DML table to be entered. The logical database name is the name by which a database or CALL DML table is addressed in the application program. The logical database name must be unique throughout the network.

DBH-NAME = *BLANK / <alphanum-name 1..1>

Name of the SESAM/SQL DBH to which the database is assigned

DCN-NAME = *BLANK / <alphanum-name 1..1>
Name of the remote DCN that forwards remote accesses to the database

CONFIGURATION-NAME = *STD / <alphanum-name 1..1> / *BLANK
Name of the configuration to which the remote DCN is assigned.

The default is the name of the configuration in which administration is being carried out, i.e. the configuration assigned to the SESDCN that is carrying out administration. This is the configuration specified by the last START-DCN-ADMINISTRATION statement.

HOST-NAME = *STD / <name 1..8>
Symbolic device name of the computer on which the database to be entered resides.The default is the name of the computer on which the SESDCN carrying out administration is running. This is the computer specified by the last START-DCN-ADMINISTRATION statement.

Administration command in ISP format

The UPD administration command corresponds to the ADD-DISTRIBUTION-RULE-ENTRY administration statement here.

[*tsn*,]UPD,CATALOG=*db-name*,PRO=*processor*[,CNF=*k*][,DCN=*x*][,DBH=*z*]

CNF, DCN and DBH can contain blanks as parameter values. You must enter an underscore (_) as a metacharacter for a blank.

5.2.3.8 ADD-OLD-TABLE-CATALOG-ENTRY

This statement adds an entry to the CALL DML table catalog.

Scope of validity

DBH administration

See also

- DBH start statement ADD-OLD-TABLE-CATALOG-LIST on "ADD-OLD-TABLE-CATALOG-LIST"
- DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST"
- DBH option OLD-TABLE-CATALOG on "OLD-TABLE-CATALOG"

Function

The ADD-OLD-TABLE-CATALOG-ENTRY statement adds an entry to the CALL DML table catalog.

You can only use this administration statement to add a new entry if there is a free entry in the CALL DML table catalog. If the CALL DML table catalog is full, the administration statement is rejected and an error message appears.

The logical database to which you want to assign the CALL DML table must be entered in the SQL database catalog. You can use the SHOW-SQL-DB-CATALOG-ENTRIES administration statement to check this. If the logical database name does not exist, the CALL DML table you want to enter cannot be accessed. The CALL DML table remains inaccessible even if you add the logical database name subsequently, using the ADD-SQL-DB-CATALOG-ENTRY administration statement or the CREATE-CATALOG statement, for example.

The CALL DML table name must be unique (even if a replication is involved); in other words, this name must not already exist in the CALL DML table catalog.

SESAM/SQL does not check whether the CALL DML table you want to enter really does exist. Neither does it check whether the table is accessible to CALL DML.

If the CALL DML table is not available after the administration statement is accepted, all subsequent CALL DML Open statements are rejected and the status indicated. If you then use the SQL statement CREATE TABLE to create the appropriate CALL DML table subsequently, it will only be accessible to CALL DML when you create a valid reference to an existing entry using the REUSE-OLD-TABLE-CATALOG-ENTRY administration statement.

```
ADD-OLD-TABLE-CATALOG-ENTRY
```

```
CALL-DML-TABLE-NAME = <text 1..17>
,CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
,SCHEMA-NAME = <c-string 1..31 with-lower-case>
,TABLE-NAME = <c-string 1..31 with-lower-case>
```

Operands

CALL-DML-TABLE-NAME = <text 1..17>

Name of the CALL DML table, as used in the CALL-DML Open statement. You must enter this name in full (17 characters) . It must be unique, even if a replication is involved.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
Logical name of the database to which the CALL DML table is to be assigned. This name must be entered in the SQL database catalog.

SCHEMA-NAME = <c-string 1..31 with-lower-case>
Name of the schema to which the CALL DML table is to be assigned.

TABLE-NAME = <c-string 1..31 with-lower-case>
Base table name corresponding to CALL-DML-TABLE-NAME.

Administration command in ISP format and at the CALL DML interface

The OPT,CDBC-INS administration command corresponds to the ADD-OLD-TABLE-CATALOG-ENTRY administration statement here.

```
OPT,CDBC-INS,C[ALL-DML-]T[ABLE]=call-dml-table-name
             ,C[ATALOG]=catalog-name
             ,S[CHEMA]=schema-name
             ,T[ABLE]=table-name
```

call-dml-table-name, catalog-name, schema-name and*table-name* must not contain any commas, since commas act as delimiters.

If you conclude the OPT,CDBC-INS administration command with a comma, the DBH expects more entries for it.

To ensure these are assigned correctly, you must always include the parameter CALL-DML-TABLE=*call-dml-table-name* with them!

If you want the DBH to execute the administration command, you must not conclude it with a comma. If you conclude an administration command with a comma and follow it with another administration command, the DBH does not execute the first of the two commands and an error message appears.

When an input takes place via INFORM-PROGRAM, lower case is automatically converted to upper case. Consequently if *schema-name* or *table-name* in the catalog contain lower case letters, no usable CDBC entry is created when input is made via INFORM-PROGRAM.

5.2.3.9 ADD-SQL-DB-CATALOG-ENTRY

This statement adds an entry to the SQL database catalog.

Scope of validity

DBH administration

See also

- DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST"
- DBH option SQL-DATABASE-CATALOG on "SQL-DATABASE-CATALOG"

Function

The ADD-SQL-DB-CATALOG-ENTRY statement adds an entry to the SQL database catalog.

You can only use this administration statement to add a new entry if there is a free entry in the SQL database catalog. If the SQL database catalog is full, the administration statement is rejected and an error message appears.

The database name you enter in the SQL database catalog must be unique; in other words, neither the logical nor the physical database name must already be entered in the SQL database catalog. This also applies to replications.

```
ADD-SQL-DB-CATALOG-ENTRY

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
,PHYS-CATALOG-NAME = *NONE / <filename 1..18 without-cat-user-gen-vers>
,USER-ID = *DBH-USER-ID / <name 1..8>
,PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
,ACCESS = *ALL / *PARAMETERS (...) / *COPY
      *PARAMETERS (...)
            |     READ = *YES / *NO
            |     ,WRITE = *YES / *NO
            |     ,CAT-ADMINISTRATION = *YES / *NO
            |     ,REPLICATION = *YES / *NO
,COPY-NUMBER = *NONE / <integer 1..999999>
```

Operands

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical name of the database to be entered. The logical database name is the name by which a database is addressed in the application program. The logical database name must be unique; in other words, this name must not already be entered in the SQL database catalog. This also applies to replications.

PHYS-CATALOG-NAME = *NONE / <filename 1..18 without-cat-user-gen-vers>
Physical name of the database to be entered.
The physical database name must be unique; in other words, this name must not already be entered in the SQL database catalog. This also applies to replications.

If you select *NONE, the default, for PHYS-CATALOG-NAME, SESAM/SQL forms the physical database name from the logical database name you assigned by means of CATALOG-NAME.

USER-ID = *DBH-USER-ID / <name 1..8>
DB user ID of the database.
The default for USER-ID is the DBH user ID.

PASSWORD = *NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>
BS2000 password that protects the files in the database from unauthorized access. If you select the default (*NONE), the DBH can only access the database when the BS2000 files in the database are not password protected.

ACCESS =
Defines which access functions are permitted with regard to the database.

ACCESS = *ALL
Allows unrestricted access to the database. Permits read and write access to user data and metadata.

ACCESS = *PARAMETERS (...)
Restricts access to the database.
The parameters are arranged in a hierarchical structure. If CAT-ADMINISTRATION is permitted, so, implicitly, are WRITE and READ In the same way, WRITE implicitly permits READ.

At least one parameter must be specified with *YES.

READ = *YES

Permits read access to user data and metadata. The data can be queried but not changed.

> When READ=*YES (and also WRITE=*NO and CAT-ADMINISTRATION=*NO) is specified for a database, accesses to this database are accelerated by means of simplified transaction locks.

READ = *NO
Prevents read access to user data and metadata.
The database entered cannot be accessed until the lock is lifted by means of the MODIFY-CATALOG-ACCESS-RIGHTS administration statement.
It makes no sense to select READ = NO if you use the default (YES) for either CAT-ADMINISTRATION or WRITE, since CAT-ADMINISTRATION implicitly permits WRITE, and WRITE implicitly permits READ.

WRITE = *YES
Permits read and write access to user data. Metadata can be read but not changed. WRITE implicitly permits READ.

WRITE = *NO
Prevents write access to user data and metadata.
It makes no sense to select WRITE = NO if you assign the default (YES) to CAT-ADMINISTRATION, since CAT-ADMINISTRATION implicitly permits WRITE.

CAT-ADMINISTRATION = *YES
Permits read and write access to both user data and metadata. CAT-ADMINISTRATION implicitly permits WRITE and thus also READ.

CAT-ADMINISTRATION = *NO
Prevents write access to the metadata in the catalog space of the database. It therefore prevents the use of the following statements:

- SQL statements for schema definition and management and memory management
- Utility statements

REPLICATION = *YES
The database to be entered is a replication. Only read access and the REFRESH REPLICATION utility statement are permitted for a replication.

REPLICATION = *NO
The database to be entered is not a replication.

ACCESS = *COPY

After entry in the SQL-DB database catalog, only read access is permitted for this database (as with ACCESS = READ) in addition to the COPY utility statement. The other access functions are not permitted. As soon as COPY has been successfully executed, the parameter CAT-ADMINISTRATION=YES is set (i.e. read and write access is permitted for both user data and metadata).

If you want to enter the duplicate of a database in the SQL database catalog, you should select ACCESS=COPY. This prevents changes being made to the duplicate of the database before a SESAM backup copy has been created (see the “ Core manual”).

COPY-NUMBER = *NONE
The database to be entered is not a SESAM backup copy.

COPY-NUMBER = <integer 1..999999>
The SESAM backup copy of the catalog space with the specified number is added to the SQL database catalog. If the number of a SESAM backup copy was specified for the COPY-NUMBER operand, specification of ACCESS is not permitted. In this case, the access rights READ=YES, WRITE=NO and CAT-ADMIN=NO are set for ACCESS by default.

Administration command in ISP format and at the CALL DML interface

The OPT,DBC-INS administration command corresponds to the ADD-SQL-DB-CATALOG-ENTRY administration statement here.

```
OPT,DBC-INS,C[ATALOG]=catalog-name
          [,P[HYSICAL-NAME]=physical-name]
          [,U[SERID]=userid]
          [,P[ASS]W[ORD]=password]
          [,A[CCESS]=(access)]
          [,COPY-NR=copynr]
access ::= R[EAD],W[RITE],A[DMIN],C[OPY],REPL[ICATION]
```

catalog-name and *physical-name* must not contain any commas, since commas act as delimiters.

If *access* is not specified, the access functions READ, WRITE and ADMIN are permitted. If *access* was specified, specification of *copyno* is not permitted.

copyno contains the number of the SESAM backup copy of the catalog space. If *copyno* was specified:

- the specified database must be a complete SESAM backup copy of a database, i.e. one created by means of COPY CATALOG
- specification of *access* is not permitted. ACCESS is set to READ by default.

READ permits only read access, WRITE implicitly permits READ, and ADMIN implicitly permits WRITE and thus also READ.

The value REPLICATION means that this database is a replication. With this entry, only a replication can be accessed. Only read access and the REFRESH REPLICATION utility statement are permitted for a replication.

If you conclude the OPT,DBC-INS administration statement with a comma, the DBH expects more entries for it. To ensure that these are assigned correctly, you must enter the parameter CATALOG=*catalog-name* for them.

If you want the DBH to execute the administration command, you must not conclude it with a comma. If you conclude an administration command with a comma and follow it with another administration command, the DBH does not execute the first of the two commands, and an error message appears.

5.2.3.10 ASSIGN-SYSLST

Switches a SYSLST file

Scope of validity

DBH administration

Function

You can use the ASSIGN-SYSLST statement to switch the SYSLST file of the first DBH-TASK. The command only affects the SYSLST file of the first DBH-TASK that is logging the currently running session.

ASSIGN-SYSLST
TO-FILE = <filename 1..54> / *PRIMARY

Operands

TO-FILE =

Assigns the SYSLST file to an output destination.

TO-FILE = <filename 1..54>
Name of the file to which you assign the SYSLST file. The name must comply with BS2000 conventions.

TO-FILE = *PRIMARY
Resets the SYSLST file to the primary assignment.

Administration command in ISP format and at the CALL DML interface

The SYSLST administration command corresponds to the ASSIGN-SYSLST administration statement here.

```
SYSLST={filename|(PRIMARY)}
```

Note when entering the administration command that the file name must not be longer than 54 characters.

5.2.3.11 BEGIN-LOCK-SEQUENCE

This statement starts a lock sequence.

Scope of validity

DBH administration

See also

- DBH administration statement END-LOCK-SEQUENCE on "END-LOCK-SEQUENCE"
- DBH administration statement ABORT-LOCK-SEQUENCE on "ABORT-LOCK-SEQUENCE"

Function

The BEGIN-LOCK-SEQUENCE statement puts the administrator in a state in which all locks subsequently placed on database catalogs and spaces are maintained. This state continues until the lock sequence is terminated by one of the administration statements END-LOCK-SEQUENCE or ABORT-LOCK-SEQUENCE.

You must observe the following points:

- The BEGIN-LOCK-SEQUENCE statement itself does not impose any locks.
- Only administration statements are permitted in the lock sequence.
- The ROLLBACK-TRANSACTION administration statement is not permitted in a lock sequence and leads to an error message. The RELEASE-USER-RESOURCES statement is ignored in a lock sequence. Another lock sequence can thus only be terminated with ABORT-LOCK-SEQUENCE.
- The user must not have opened any other lock sequences in the DBH in which the statement was issued. For administration via the CALL DML interface, this applies to the DBH at which the command is issued.
- The administration statement is rejected if the SESAM DBH was loaded without transaction management.

BEGIN-LOCK-SEQUENCE
MODE = *WAIT / *FORCED

Operands

MODE =

Specifies how the locks are applied in a lock sequence if they cannot be implemented immediately. This is the case when the object to be locked is still locked by another user.

MODE = *WAIT
There is a wait until the user responsible for locking the object unlocks it.

MODE = *FORCED
If possible, the user responsible for locking the object is rolled back.
This is not possible if the transaction to be rolled back is in the PTC status, is running in another lock sequence, or the lock sequence is locked by a utility statement. In this case, the current administration statement is rejected with an appropriate message.

Administration command in ISP format and at the CALL DML interface

The LSQ,BEGIN administration command corresponds to the BEGIN-LOCK-SEQUENCE administration statement here.

```
LSQ,B[EGIN],mode

mode ::= {W[AIT]|F[ORCED]}
```

The *mode* parameter must be specified.

5.2.3.12 CANCEL-STATEMENT

Cancels a DML statement

Scope of validity

DBH administration

Function

The CANCEL-STATEMENT statement cancels the processing of a statement that manipulates a user's data.

The following must be observed:

- The user must be active in the DBH.
- If a rollback of the transaction has already been registered for the user, the DBH accepts the administration statement but does not execute it.
- A CANCEL-STATEMENT statement cannot cancel a previous CANCEL-STATEMENT statement.

CANCEL-STATEMENT
USER = <user-identification>

Operands

USER = <user-identification>

Identifies the user to which the CANCEL-STATEMENT statement applies. The identification must be unique. You must specify all the relevant partial qualifications (see section “Syntax for the identification of users”).

Administration command in ISP format and at the CALL DML interface

The CANCEL administration command corresponds to the CANCEL-STATEMENT administration statement here.

```
CANCEL,user-identification
```

5.2.3.13 CHANGE-CATLOG

This statement changes the CAT-LOG file and the DA-LOG files and creates a copy of the CAT-REC file.

Scope of validity

DBH administration

Function

You can use the CHANGE-CATLOG statement to close the CAT- LOG file and the DA-LOG files for the specified database or databases and open the next one, and to create a copy of the CAT-REC file.

The copy of the CAT-REC file is created in the media of the CAT-LOG file. It cannot be created on tape.

If the database – and hence also the CAT-REC file – is located in a DB user ID, a file is created with the name *catalog-name*.CAT-REC.COPY. For more information about “Database files and job variables on foreign user IDs“ see the section of the same name in the “ Core manual”. If the file cannot be created on the DB user ID it is created on the DBH user ID.

The CHANGE-CATLOG statement is rejected with an error message in the following situations:

- the specified database does not belong to the configuration
- the specified database is being run without LD backup
- READ access only is allowed for the specified database
- the specified database is in the “copy pending” state
- the specified database cannot be locked during processing (due to parallel utility, DDL or administration statements)
- the specified database is a SESAM backup copy or a replication
- SELECT = *ALL is specified and one of the databases cannot be locked during processing (due to parallel utility, DDL or administration statements). The name of the database where the error occurred is announced; the action was executed for all databases with LD backup that are before this database in the ordered list of DBC entries (corresponding to the output for SHOW-SQL-DB-CATALOG-ENTRIES)
- the specified database is closed or defective
- there is a resource bottleneck.

If you do not specify any databases, the CAT-LOG file and the DA-LOG files are switched in the order given in the database directory and the action is performed for all databases with an LD backup. For databases for which logging is not activated, the statement is ignored.

```
CHANGE-CATLOG

SELECT = *ALL / *LOGICAL(...) / *PHYSICAL(...)

  *LOGICAL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *PHYSICAL(...)

    | PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
```

Operands

SELECT =

Specifies the databases for which the CAT-LOG file and DA-LOG files are to be changed.

SELECT = *ALL
The CAT-LOG file and DA-LOG files of all the databases for which logging is activated are changed.

SELECT = *LOGICAL(...)
Selection of the database(s) whose CAT-LOG and DA-LOG files are to be changed

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical name of the database

SELECT = *PHYSICAL(...)
Selection of the database(s) whose CAT-LOG and DA-LOG files are to be changed

PHYS-CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Physical name of the database

Administration command at the CALL DML interface

The CAW administration command corresponds to the CHANGE-CATLOG administration statement here. The CAW administration statement can be issued at the CALL DML interface or through the INFORM-PROGRAM interface.

```
CAW[,name]

name ::= {C[ATALOG]=catalog-name|P[HYSICAL-NAME]=physical-name}
```

You can specify the logical name of the database in *catalog-name* using up to 18 characters.

You can specify the physical name of the database in *physical-name* using up to 18 characters.

5.2.3.14 CHANGE-DALOG

Switches DA-LOG files

Scope of validity

DBH administration

Function

You can use the CHANGE-DALOG statement when logging to close the DA-LOG files for the specified database or databases and open the next DA-LOG file.

If you specify databases that do not belong to the configuration or for which logging is not activated, the statement is rejected.

If you do not specify any databases, the DA-LOG files are changed in the order of the catalog list. If logging is not activated, the statement is ignored.

The CHANGE-DALOG statement is not permitted while a database is being processed by a utility statement, and it is rejected with an error message.

The CHANGE-DALOG statement is rejected with an error message in the following situations:

- the specified database does not belong to the configuration
- logging is not turned on for the specified database
- READ access only is allowed for the specified database
- the specified database is a SESAM backup copy or a replication
- the specified database is closed or defective
- the specified database is in the “copy pending” state
- the specified database cannot be locked during processing (due to parallel utility, DDL or administration statements)
- if SELECT = *ALL is specified and one of the databases cannot be locked during processing (due to parallel utility, DDL or administration statements). The name of the database where the error occurred is announced; the action was executed for all databases with LD backup that are before this database in the ordered list of DBC entries (corresponding to the output for SHOW-SQL-DB-CATALOG-ENTRIES)
- if there is a resource bottleneck.

```
CHANGE-DALOG
```

```
SELECT = *ALL / *LOGICAL(...) / *PHYSICAL(...)

  *LOGICAL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *PHYSICAL(...)

    | PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
```

Operands

SELECT =

Specifies the databases for which the DA-LOG files are to be changed.

SELECT = *ALL
The DA-LOG files for all databases for which logging is activated are changed.

SELECT = *LOGICAL(...)
Selection of the database(s) whose DA-LOG file is to be changed

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical name of the database

SELECT = *PHYSICAL(...)
Selection of the database(s) whose DA-LOG file is to be changed

PHYS-CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Physical name of the database

Administration command at the CALL DML interface

The DAW administration command corresponds to the CHANGE-DALOG administration statement here. The DAW administration command cannot be issued at the CALL DML interface and through the INFORM-PROGRAM interface.

DAW[,*name*]

name ::= {C[ATALOG]=*catalog-name*|P[HYSICAL-NAME]=*physical-name*}

You can specify the logical name of the database in *catalog-name* using up to 18 characters. You can specify the physical name of the database in *physical-name* using up to 18 characters.

5.2.3.15 CLOSE-SPACE

Close user space physically

Scope of validity

DBH administration

See also

- DBH administration statement SHOW-SPACE-USERS on "SHOW-SPACE-USERS"
- DBH administration statement BEGIN-LOCK-SEQUENCE on "BEGIN-LOCK-SEQUENCE"
- DBH administration statement END-LOCK-SEQUENCE on "END-LOCK-SEQUENCE"

Function

The CLOSE-SPACE statement physically closes the specified user space.

Partitions of a partitioned table located on this space are marked as “logically not available”.

> In the ongoing DBH session you can use the administration statement SHOW-PARTITIONS (see "SHOW-PARTITIONS") to be shown whether partitions are available. The administration statement REUSE-PARTITIONS (see "REUSE-PARTITIONS") enables you to restore the logical availability. Further information on partitions is provided in the “ Core manual”.

The prerequisite for this is that the user space must be free of transactions. This is ensured by the DBH for the duration of the close operation. You can find out which users are currently using a space by using the SHOW-SPACE-USERS statement.

In the following situations it is necessary to close a user space physically:

- when a foreign copy is created using the TimeFinder function (see the “ Core manual”)
- when private disks are used for spaces and the storage area of the spaces is to be changed (with the BS2000 command MODIFY-FILE-ATTRIBUTES, see the BS2000 manual “ Commands”)

> Note that closing the user spaces does not lock them in the sense that it prevents further accesses. The space is opened again each time it is accessed. You can prevent this happening by issuing the CLOSE-SPACE statement in a lock sequence. Lock sequences are opened with BEGIN-LOCK-SEQUENCE. END-LOCK-SEQUENCE terminates the sequence and unlocks the space again.

If the space is already closed when the statement is issued, the statement is executed without a message being issued. Logical files or cursor positions that reference the space are lost when it is physically closed.

CLOSE-SPACE
SPACE-NAME = <filename 1..18 without-cat-user-gen-vers> ,CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Operands

SPACE-NAME = <filename 1..18 without-cat-user-gen-vers>

Name of the space to be closed. The name of the space is specified without the physical database name.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical name of the database to which the space to be closed belongs

Administration command in ISP format and at the CALL DML interface

The CLOSE,SPACE administration command corresponds to the CLOSE-SPACE administration statement here.

```
CLOSE,S[PACE]=space-name,C[ATALOG]=catalog-name
```

5.2.3.16 COMMIT-PTC-TRANSACTION

This statement commits a PTC transaction.

Scope of validity

DBH administration

See also

The ROLLBACK-PTC-TRANSACTION administration statement on "ROLLBACK-PTC-TRANSACTION"

Function

The COMMIT-PTC-TRANSACTION statement commits a user's PTC transactions. A PTC transaction is a distributed transaction that has the status “prepared to commit” (see the “ Core manual”).

openUTM-D and SESAM/SQL-DCN permit distributed transactions involving several DBHs. Distributed or PTC transactions are divided into three phases:

1. the processing of the transaction
2. the prepare-to-commit phase
3. the commitment (completion) of the transaction

In an external restart of the SESAM/SQL DBH, the transaction and restart mechanism of openUTM-D or SESAM /SQL-DCN controls the final commitment and, if appropriate, rolling back of PTC transactions.

If a restart through openUTM-D or SESAM/SQL-DCN is not possible because, for example, the computers involved are no longer accessible, the system administrator can use the COMMIT-PTC-TRANSACTION administration statement to complete and thus commit a user's PTC transactions in the node for which he or she is responsible.

This statement is only relevant to PTC transactions. If no PTC transactions are active in the DBH, the statement is rejected. It is also rejected if the DBH was loaded without transaction management.

If a user does not have any transaction in the PTC status, then a message is output.

> i You should only use the COMMIT-PTC-TRANSACTION administration statement after careful consideration; it can destroy the cross-node logical consistency of the transactions involved.

COMMIT-PTC-TRANSACTION
USER = <user-identification>

Operands

USER = <user-identification>

Identifies the user whose PTC transactions are to be completed. The identification must be unique. You must specify all the relevant partial qualifications (see section “Syntax for the identification of users”).

Administration command in ISP format and at the CALL DML interface

The CPTC administration command corresponds to the COMMIT-PTC-TRANSACTION administration statement here.

CPTC,*user-identification*

5.2.3.17 CREATE-DUMP

This statement creates a main memory dump and optionally terminates the DBH or SESDCN session.

Scope of validity

DBH and SESDCN administration

Function

You use the CREATE-DUMP statement to create a main-memory dump.

The DBH or SESDCN session normally continues after this statement is issued. However, you can cause the session to be terminated once the dump has been created.

CREATE-DUMP
CANCEL-PROGRAM = *NO / *YES

Operands

CANCEL-PROGRAM = *NO

This is the default. The DBH or SESDCN session continues after the dump is created. In DBH administration, the DBH issues the consistency check SESADXS: XX/05/99 after this statement is issued.

CANCEL-PROGRAM = *YES
After the dump is created, the DBH or SESDCN session is terminated.
In DBH administration, the DBH issues the consistency check SESADXS: XX/40/99 after this statement is issued.
If databases are being edited (by means of SQL statements or CALL DML statements) when the session is terminated, an external restart is necessary. If transaction management was not set for the session, the databases involved have to be repaired by means of media recovery (see the “ Core manual”).

Administration command in ISP format and at the CALL DML interface

The DUMP[,STOP] administration command corresponds to the CREATE-DUMP administration statement here. The STOP parameter is not permitted for SESDCN. The SESDCN session therefore cannot be terminated by means of the DUMP command.

```
DUMP[,STOP]
```

5.2.3.18 CREATE-SERVICE-TASK-DUMP

This statement creates a main memory dump of a service task.

Scope of validity

DBH administration

Function

You use the CREATE-SERVICE-TASK-DUMP statement to create a main-memory dump of a service task.

The service task continues after this statement is issued.

CREATE-SERVICE-TASK-DUMP
TSN =<tsn>

Operands

TSN=<tsn>

<tsn> is the tsn **(t**ask **s**equence **n**umber) of the service task to create a main memory dump. The tsn of the service task may be looked up in the output mask SERVICE ORDERS of the performance monitor SESMON. Alternatively the mask SERVICE TASKS with the consumption values of the active service tasks can be used..

Administration command in ISP format and at the CALL DML interface

The DUMP,SVT,TSN=<tsn> administration command corresponds to the CREATE-SERVICE-TASK-DUMP administration statement here.

```
DUMP,SVT,TSN=<tsn>
```

5.2.3.19 END-FOREIGN-COPY

This statement cancels any “copy pending” state in the spaces after a foreign copy. Changes the database status after PREPARE-FOREIGN-COPY with physical closure of the database files.

Scope of validity

DBH administration

See also

- DBH administration statement PREPARE-FOREIGN-COPY on "PREPARE-FOREIGN-COPY"
- DBH administration statement BEGIN-LOCK-SEQUENCE on "BEGIN-LOCK-SEQUENCE"
- DBH administration statement END-LOCK-SEQUENCE on "END-LOCK-SEQUENCE"

Function

Use the END-FOREIGN-COPY statement to cancel any “copy pending” state which has been created by placing a utility statement preceding the foreign copy.

> **CAUTION!** If the “copy pending” state is canceled by means of END-FOREIGN-COPY without a consistent foreign copy having been created, the space can no longer be repaired correctly by means of RECOVER.

If the database is in the CLOSED status as a result of PREPARE-FOREIGN-COPY with physical closure of the database, the database is checked. If the database if unchanged, it is assigned the ACTIVE status. If the database has been changed, it is assigned the FREE status and its resources are released.

If the statement is issued in a lock sequence, the locks on the database entries and spaces are not canceled until the lock sequence is terminated. Lock sequences can be opened with BEGIN-LOCK-SEQUENCE and terminated with END-LOCK-SEQUENCE.

The statement END-FOREIGN-COPY can also be specified without the prior statement PREPARE-FOREIGN-COPY. If END-FOREIGN-COPY is not preceded by the statement PREPARE-FOREIGN-COPY, the “copy pending” state will be cancelled from all catalog spaces.

If a preceding PREPARE-FOREIGN-COPY statement is used in the same administration session, the statement END-FOREIGN-COPY will only be active in the spaces entered in the PREPARE-FOREIGN-COPY statement.

If several PREPARE-FOREIGN-COPY statements have been specified, END-FOREIGN-COPY only acts on the last PREPARE-FOREIGN-COPY statement specified.

```
END-FOREIGN-COPY
```

```
SELECT = *LOGICAL(...) / *PHYSICAL(...)

  *LOGICAL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *PHYSICAL(...)

    | PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
```

Operands

SELECT =

Specifies whether the entry is identified by means of the logical or physical name.

SELECT = *LOGICAL(...)
Selects the database in whose space the “copy pending” state is to be canceled.

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical name of the database

SELECT = *PHYSICAL(...)
Selects the database in whose space the “copy pending” state is to be canceled.

PHYS-CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Physical name of the database

Administration command in ISP format and at the CALL DML interface

The FCOPY,END administration command corresponds to the END-FOREIGN-COPY administration statement here.

FCOPY,E[ND],*name*

name ::= {C[ATALOG]=*catalog-name*|P[HYSICAL-NAME]=*physical-name*}

You can specify the logical name of the database in *catalog-name* using up to 18 characters.

You can specify the physical name of the database in *physical-name* using up to 18 characters.

5.2.3.20 END-LOCK-SEQUENCE

This statement terminates a lock sequence.

Scope of validity

DBH administration

See also

- DBH administration statement ABORT-LOCK-SEQUENCE on "ABORT-LOCK-SEQUENCE"
- DBH administration statement BEGIN-LOCK-SEQUENCE on "BEGIN-LOCK-SEQUENCE"

Function

The END-LOCK-SEQUENCE statement terminates a lock sequence of the current user.

All of the requested locks on databases and spaces during the lock sequence are canceled by this statement.

Note the following points:

- The administration statement is rejected if the SESAM DBH was loaded without transaction management.
- The user must be in a lock sequence. If this is not the case, the statement is rejected and an error message is issued to this effect.

```
END-LOCK-SEQUENCE
```

Operands

This statement has no operands.

Administration command in ISP format and at the CALL DML interface

The LSQ,END administration command corresponds to the END-LOCK-SEQUENCE administration statement here.

```
LSQ,E[ND]
```

5.2.3.21 HOLD-TRANSACTION-ADMISSION

This statement suspends admission of any further transactions.

Scope of validity

DBH and SESDCN administration

See also

The RESUME-TRANSACTION-ADMISSION administration statement on "RESUME-TRANSACTION-ADMISSION"

Function

You can use the HOLD-TRANSACTION-ADMISSION administration statement to suspend processing of transactions and thus prevent new transactions being admitted.

Application programs affected by this suspension of transaction processing learn of it when a transaction is started. An SQL program by means of the appropriate SQLSTATE, and a CALL DML program by means of the status 9K.

If requests with open transactions still exist when the administration statement is issued, SESAM/SQL displays them in tabular form.

In DBH administration, but not SESDCN administration, you can cause transaction processing to be suspended only once the DBH is transaction free.
This is the case when all transactions have been committed or rolled back. SESAM/SQL outputs a message indicating this to the administration program, which then continues its work.

You can permit the resumption of transaction processing by means of the RESUME-TRANSACTION-ADMISSION administration statement.

HOLD-TRANSACTION-ADMISSION
WAIT-EVENT = *NONE / *TRANSACTION-FREE-STATE

Operands

WAIT-EVENT =

Specifies whether or not the transaction-free state must be reached before transaction processing can be suspended.

WAIT-EVENT = *NONE
This is the default. Transaction processing is suspended regardless of whether the DBH is transaction free.

WAIT-EVENT = *TRANSACTION-FREE-STATE
Transaction processing is not suspended until the DBH is transaction free. This operand is only permitted in DBH administration. It is not permitted in SESDCN administration.

Administration command in ISP format and at the CALL DML interface

Two administration commands fulfill the function of the HOLD-TRANSACTION-ADMISSION administration statement:

- HOLD-TA (for DBH administration) or HOLD,TA (for SESDCN administration)
- WAIT,HOLD.

The HOLD-TA administration command is permitted for DBH administration and HOLD,TA for SESDCN administration. HOLD,TA has the same function as the HOLD-TRANSACTION-ADMISSION administration statement with the operand WAIT-EVENT = *NONE.

```
HOLD-TA
```

```
HOLD,TA
```

The WAIT,HOLD administration command is permitted only in DBH administration and can only be issued via the CALL DML interface.
It has the same function as the HOLD-TRANSACTION-ADMISSION administration statement with the operand WAIT-EVENT = *TRANSACTION-FREE-STATE.

```
WAIT,HOLD
```

5.2.3.22 HOLD-USER-ADMISSION

This statement suspends admission of any further users.

Scope of validity

SESDCN administration

See also

The RESUME-USER-ADMISSION administration statement on "RESUME-USER-ADMISSION"

Function

You use the HOLD-USER-ADMISSION statement to suspend the admission of any further users.

No further users from the administered configuration are admitted. However, users active up to this point can continue their work.

```
HOLD-USER-ADMISSION
```

Operands

This statement has no operands.

Administration command in ISP format and at the CALL DML interface

The HOLD,USER administration command corresponds to the HOLD-USER-ADMISSION administration statement here.

```
HOLD,USER
```

5.2.3.23 MODIFY-ADMINISTRATION

Changes the administration authorization

Scope of validity

DBH and SESDCN administration

See also

- The DBH option ADMINISTRATOR on "ADMINISTRATOR"
- SESDCN control statement SET-DCN-OPTIONS on "SET-DCN-OPTIONS"

Function

The MODIFY-ADMINISTRATION administration statement allows you to change the administration authorization to suit requirements. Administration authorization is set by means of the DBH option ADMINISTRATOR or the SESDCN control statement SET-DCN-OPTIONS.

You can change both the password, which prevents unauthorized administration of the DBH or SESDCN, and the identification (system user identification) of the administrator. Note that the system user identification entered here is also required for certain utlity statements, e.g. CREATE CATALOG, in authorization checking.

The MODIFY-ADMINISTRATION administration statement is not permitted unless you have made the appropriate settings for the DBH option ADMINISTRATOR when starting the SESAM/SQL DBH. In the case of SESDCN, this is done using SET-DCN-OPTIONS.

> ¡ You can even use this statement to withdraw your own administration authorization.

```
MODIFY-ADMINISTRATION

NEW-PASSWORD = *UNCHANGED / <c-string 3..3> / <x-string 5..6>
,ADMINISTRATOR = *UNCHANGED / *TIME-SHARING-USER(...) / *APPLICATION-USER(...) / *ANY
    *TIME-SHARING-USER(...)
        |    HOST-NAME = <name 1..8>
        |    ,USER-ID = <name 1..8>
    *APPLICATION-USER(...)
        |    HOST-NAME = <text 1..8>
        |    ,APPLICATION-NAME = <text 1..8>
        |    ,CUSTOMER-NAME = <text 1..8>
```

Operands

NEW-PASSWORD = *UNCHANGED / <c-string 3..3> / <x-string 5..6>

New password that is to replace the existing one. The default is *UNCHANGED. The existing password is not changed.

ADMINISTRATOR =
Identifies the user or user group authorized to issue administration statements using the CALL interface and SESADM.

ADMINISTRATOR = *UNCHANGED
This is the default. The administration authorization remains the same. It remains as set by means of the DBH option ADMINISTRATOR.

ADMINISTRATOR = *TIME-SHARING-USER(...)
Changes the administration authorization.
A timesharing user, identified by his or her system user identification, is the administrator.

HOST-NAME = <name 1..8>
Name of the computer from which administration is to be carried out

USER-ID = <name 1..8>
User ID of the administrator

ADMINISTRATOR = *APPLICATION-USER(...)
Changes the administration authorization.

An application user identified by the system user ID is the administrator.

HOST-NAME = <text 1..8>
Name of the computer from which administration is to be carried out

APPLICATION-NAME = <text 1..8>
Name of the application from which administration is to be carried out

CUSTOMER-NAME = <text 1..8>

Name of the user.If the administrator is working under UTM, you must enter the KDCSIGN name here. If the administrator is working under DCAM, you must enter the name by which the system administrator is identified at the programming interface.

ADMINISTRATOR = *ANY
Permits all users to issue administration statements using the CALL interface and SESADM, provided they know the password.

Administration command in ISP format and at the CALL DML interface

The OPT,ADM administration command corresponds to the MODIFY-ADMINISTRATION administration statement here.

```
OPT,ADM=xxx[,admin-name]

admin-name ::= {T[IME]-S[HARING]-U[SER]=host,userid|
               A[PPLICATION]-U[SER]=host,appl,kdcsign|
               ANY}
```

If you want to change an existing password, enter the new password for *xxx*.

If you want to change only the identification of the administrator, *admin-name*, enter the existing password for *xxx*.

admin-name must be unique! You therefore have to specify all the relevant partial qualifications.

5.2.3.24 MODIFY-CATALOG-ACCESS-RIGHTS

Changes the access rights for the specified database

Scope of validity

DBH administration

See also

Administration statement ADD-SQL-DB-CATALOG-ENTRY on "ADD-SQL-DB-CATALOG-ENTRY"

Function

The MODIFY-CATALOG-ACCESS-RIGHTS administration statement allows you to change the access rights for a named database.

This may be necessary when the WRITE access function for a database has been set internally to READ as a result of error recovery. If a consistency check has taken place in the CAT-LOG or DA-LOG file, SESAM/SQL turns off logging for these files in order to obtain partial availability of the DBH. In error recovery, SESAM/SQL sets the access function for the affected database to READ. Once the cause of the error has been eliminated, you can use the MODIFY-CATALOG-ACCESS-RIGHTS statement to permit the WRITE access function for the affected database again.

A transactionless state is not required for the MODIFY-CATALOG-ACCESS-RIGHTS statement.

The change of access rights is transaction-synchronized; in other words, the access rights are changed under the protection of an exclusive transaction lock on the database entry, and the change is synchronized.

Changes to the access functions do not take effect until a statement is issued again. Statements active in the DBH are not notified of the change.

```
MODIFY-CATALOG-ACCESS-RIGHTS
```

```
CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
,ACCESS = *ALL / *PARAMETERS(...)
    *PARAMETERS(...)
        |   READ = *YES / *NO
        |   ,WRITE = *YES / *NO
        |   ,CAT-ADMINISTRATION = *YES / *NO
```

Operands

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical database name of the database for which the access rights are to be modified. The database name must not refer to the SESAM backup copy or a replication of a database.

ACCESS =
Specifies which access functions are to be permitted or suppressed for the database.

ACCESS = *ALL
Allows unrestricted access to the database. Permits read and write access to user data and metadata.

ACCESS = *PARAMETERS(...)
Restricts access to the database.
The parameters are arranged in a hierarchical structure. If CAT-ADMINISTRATION is permitted, so, implicitly, are WRITE and READ In the same way, WRITE implicitly permits READ. At least one parameter must be specified with *YES.

READ = *YES
Permits read access to user data and metadata. The data can be queried but not changed.

> When READ=*YES (and also WRITE=*NO and CAT-ADMINISTRATION=*NO) is specified for a database, accesses to this database are accelerated by means of simplified transaction locks.

READ = *NO
Prevents read access to user data and metadata.

It makes no sense to select READ = NO if you assign the default (YES) to either CAT-ADMINISTRATION or WRITE, since CAT-ADMINISTRATION implicitly permits WRITE and WRITE implicitly permits READ.

WRITE = *YES
Permits read and write access to user data. Metadata can be read but not changed.

WRITE = *NO

Prevents write access to user data and metadata. It makes no sense to select WRITE = NO if you assign the default (YES) to CAT-ADMINISTRATION, since CAT-ADMINISTRATION implicitly permits WRITE.

CAT-ADMINISTRATION = *YES
Permits read and write access to both user data and metadata.

CAT-ADMINISTRATION = *NO
Prevents write access to the metadata in the catalog space of the database. It therefore prevents the use of the following statements:

- SQL statements for schema definition and management and memory management
- Utility statements

Administration command in ISP format and at the CALL DML interface

The OPT,ACCESS administration command corresponds to the MODIFY-CATALOG-ACCESS-RIGHTS administration statement here.

`OPT,ACCESS=(`*`access`*`),C[ATALOG]=`*`catalog-name`*

`access` `::= R[EAD],W[RITE],A[DMIN]`

The administration command is not effective unless you explicitly specify at least one access function for *access*.

catalog-name must not contain any commas as characters because commas function as separators. *catalog-name* must not refer to a SESAM backup copy or a replication of a database.

READ permits only read access, WRITE implicitly permits READ, and ADMIN implicitly permits WRITE and thus also READ.

A transactionless state is not required for the OPT,ACCESS=(ADMIN) command. The change of access rights is synchronized instead.

You cannot subsequently modify the ACCESS=COPY access function. When appropriate, it is changed internally by the COPY utility statement.

5.2.3.25 MODIFY-CATID-LIST

This statement updates the CATID list.

Scope of validity

DBH administration

Function

It is possible to specify a CATID list. This list limits the internal search for files to the CATIDs specified in this list. It can have a positive effect on the performance of the system, since it is no longer necessary to include all CATIDs in the search. However, performance is, of course, also dependent on the length of the specified list. When specifying a CATID list, note that a maximum of 50 CATIDs are evaluated.

The DBH takes the CATID list from a file. You can use the MODIFY-CATID-LIST statement to cause the DBH to take a new CATID list or to stop evaluating an existing CATID list. The modification takes effect the next time a file is opened in the DBH.

If the DBH has already been assigned a file with a CATID list in the start procedure by means of the ADD-FILE-LINK command, this list takes effect at the startup of the DBH.

The CATID of the default pubset must be contained in the CATID list. If this CATID is not in the list, the DBH adds it to the list. In addition, a warning is issued.

The file that contains the CATID list must be a SAM file and can, for example, be created with the file editor EDT (see the manual “ EDT (BS2000) Statements”). The defined record length must be variable. There is a CATID without colons or blanks in each row. The CATID can be between 1 and 4 characters in length. After the descriptions of the operands below you will find an example of a CATID list created with EDT.

MODIFY-CATID-LIST
FILE-NAME = *STD / *NONE / <filename 1..54>

Operands

FILE-NAME =

Name of the file that contains the CATID list

FILE-NAME = *STD
By default, the CATID file assigned to the DBH at the time of the administration statement with the link name SESAMCID is used.

FILE-NAME = *NONE

No CATID list is transferred. Any existing CATID list in the DBH is deleted. The DBH searches for all CATIDs. This is what happens in the versions up to and including version 3.0.

FILE-NAME = <filename 1..54>
Name of the file that contains the new CATID list. If a CATID list is already active, the file specified here must be in the same pubset as the file of the active list.

The administration statement is rejected if the CATIDs of the currently open database files are not contained in the new CATID list specified.

Example

CATID list created using the EDT file editor:

```
2OGB
2OGE
B321
3OAM
3OLR
C245
L324
LO21
2KL3
PL34
PO54
P567
P123
P425
0L43
```

Administration command in ISP format and at the CALL DML interface

The UPD,CID-LIST administration command corresponds to the MODIFY-CATID-LIST administration statement here.

```
UPD,C[ID-LIST],file

file ::= {NONE|STD|FILE=filename}
```

filename must comply with BS2000 conventions. In INFORM-PROGRAM administration you must adhere to the permissible length for *filename* (see section “Administration using INFORM-PROGRAM”).

5.2.3.26 MODIFY-DISTRIBUTION-RULE-ENTRY

This statement changes the host name in the distribution rule.

Scope of validity

SESDCN administration

Function

You use the MODIFY-DISTRIBUTION-RULE-ENTRY statement to change the host name in the distribution rule.

This can be necessary, for example, in the following case:
If the SESDCN restart is carried out on a computer other than the cold-start computer, the master DCN updates the distribution rules for all remote computers that have worked with it between the cold start and the abortion of the session and are accessible to it at the time of the restart. If not all these partners are accessible to the master DCN at the time of the restart, the system administrator must update some distribution rules manually and use the MODIFY-DISTRIBUTION-RULE-ENTRY statement to replace the host name.

MODIFY-DISTRIBUTION-RULE-ENTRY
HOST-NAME = <name 1..8> ,NEW-NAME = <name 1..8> ,CONFIGURATION-NAME = *ANY / <alphanum-name 1..1> / *BLANK

Operands

HOST-NAME = <name 1..8>

Symbolic device name of the computer replaced by the new computer.

NEW-NAME = <name 1..8>
Symbolic device name of the new computer replacing the previous one.

CONFIGURATION-NAME = *ANY / <alphanum-name 1..1> / *BLANK
Configuration name that defines which distribution-rule entries are affected by the change of host name.

CONFIGURATION-NAME = *ANY
Changes the host name in all distribution-rule entries that contain the old host name.

CONFIGURATION-NAME = <alphanum-name 1..1> / *BLANK

Changes the host name only in distribution-rule entries that contain the old host name and the specified configuration name.

Administration command in ISP format and at the CALL DML interface

The UPDPRO administration command corresponds to the MODIFY-DISTRIBUTION-RULE-ENTRY administration statement here.

```
[tsn,]UPDPRO,PRO-OLD=r-name1,PRO-NEW=r-name2[,CNF=k]
```

CNF can contain a blank as a parameter value. You must enter an underscore (_) as a metacharacter for a blank.

5.2.3.27 MODIFY-MAIL-PARAMETERS

This statement changes the parameters for the email output of the SESAM/SQL DBH.

Scope of validity

DBH administration

See also

The DBH option MSG-OUTPUT on "MSG-OUTPUT"

Function

You use the MODIFY-MAIL-PARAMETERS statement to change the parameters for the email output of the SESAM /SQL DBH.

MODIFY-MAIL-PARAMETERS
FILE-NAME = *STD / <filename 1..54 without-cat-user-gen-vers>

Operands

FILE-NAME =

Name of the MAIL parameter file for the email output of the DBH.

FILE-NAME = *STD
The MAIL parameter file with the link name SESMAIL is evaluated.

FILE-NAME = <filename 1..54 without-cat-user-gen-vers>
The MAIL parameter file with the specified name is evaluated.

5.2.3.28 MODIFY-MSG-OUTPUT

Changes the DBH output

Scope of validity

DBH administration

See also

The DBH option MSG-OUTPUT on "MSG-OUTPUT"

Function

You use the MODIFY-MSG-OUTPUT statement to change the outputs of the SESAM/SQL DBH. You can change the output destination and mask out specific outputs.

You can completely suppress OPEN and CLOSE messages for logical files.

When using the linked-in DBH, it can be a good idea to request that DBH outputs only be written to SYSOUT so that outputs of the application program to SYSLST are not interspersed with DBH outputs.

Consistency checks and the end-of-program message are always output to the console and cannot be suppressed.

```
MODIFY-MSG-OUTPUT

MSG-OUTPUT = *STD / *MSG(...) / *OPEN-CLOSE-MSG(...) / *MAIL(...)
    *MSG(...)
        | OUTPUT = *ALL / *SYSOUT / *SYSLST
    *OPEN-CLOSE-MSG(...)
        | OUTPUT = *NONE / *ALL / *SYSOUT / *SYSLST
    *MAIL(...)
        | OUTPUT = *NONE / *PARAMETERS(...)
        |       *PARAMETERS(...)
        |             |    COLLECT = *NO / *YES
        |             |    ,IMMEDIATE = *NO / *YES
```

Operands

MSG-OUTPUT = *STD / *MSG(...) / *OPEN-CLOSE-MSG(...) / *MAIL(...)

Changes the DBH output.

MSG-OUTPUT = *STD
The DBH output, except for OPEN and CLOSE messages, is sent to SYSOUT or the console and SYSLST. OPEN and CLOSE messages are completely suppressed. This output corresponds to the default value of the DBH option MSG-OUTPUT. This corresponds to specifying *MSG(OUTPUT=*ALL) and *OPEN-CLOSE-MSG(OUTPUT=*NONE).

MSG-OUTPUT = *MSG(OUTPUT = *ALL / *SYSOUT / *SYSLST)
Defines the output medium for the DBH - except for OPEN and CLOSE messages.

OUTPUT = *ALL
The DBH output is sent to SYSOUT or the console and SYSLST.

OUTPUT = *SYSOUT

The DBH output is sent to SYSOUT or the console.

OUTPUT = *SYSLST
The DBH output is only sent to SYSOUT.

MSG-OUTPUT = *OPEN-CLOSE-MSG(OUTPUT = *NONE / *ALL / *SYSOUT / *SYSLST)

The output of OPEN and CLOSE messages from SES1001, SES1011, SES1201, SES2015 and SES2023 can be suppressed within the framework defined by the MSG= parameter. OPEN and CLOSE messages for databases are not affected by this.

OUTPUT = *NONE
Suppresses OPEN and CLOSE messages completely.

OUTPUT = *ALL
The DBH outputs the OPEN and CLOSE messages as specified by the *MSG(...) parameter.

OUTPUT = *SYSOUT
The DBH outputs OPEN and CLOSE messages only to SYSOUT, provided the parameter *MSG(...) permits it.

OUTPUT = *SYSLST
The DBH outputs OPEN and CLOSE messages only to SYSLST, provided the parameter *MSG(...) permits it.

MSG-OUTPUT = *MAIL(OUTPUT = *NONE / *PARAMETERS)

Controls the output of information by email.

OUTPUT = *NONE
The information is not sent by email.

OUTPUT = *PARAMETERS(...)
Controls the volume of information which is sent by email.

COLLECT = *NO
Outputs to SYSLST are not sent by email.

COLLECT = *YES
All outputs to SYSLST are also stored in a temporary file. The temporary file is updated until an administration command MODIFY-MSG-OUTPUT is issued or to the end of the DBH session. Then the temporary file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email"). The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST(DBH)” is entered under “Subject:”.
The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The temporary file is attached to the email.

IMMEDIATE = *NO
Messages are not sent by email.

IMMEDIATE = *YES
All messages with message numbers which are entered in the MAIL parameter file (see "Sending

important information of the DBH session by email") are immediately sent by email to the address which is entered in the MAIL parameter file. The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

The message number is entered under “Subject:”. The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The text of the email contains the message.

> When running in the batch mode, the DBH output is sent to the console when *SYSOUT is specified.

Administration command in ISP format and at the CALL DML interface

The OPT,MSG-OUTPUT administration command corresponds to the MODIFY-MSG-OUTPUT administration statement here.

```
OPT,MSG-OUTPUT,output
output ::= {STD | M[SG]={ALL|SYSOUT|SYSLST} |
            O[PEN-CLOSE-MSG]={NONE|ALL|SYSOUT|SYSLST} |
            MAIL={NONE|COLLECT|IMMEDIATE|ALL}}
```

MAIL=ALL corresponds to COLLECT=*YES and IMMEDIATE=*YES in the operand description above.

5.2.3.29 MODIFY-OLD-TABLE-CATALOG-LIMIT

Changes the maximum number of entries in the CALL-DML table catalog

Scope of validity

DBH administration

See also

DBH option OLD-TABLE-CATALOG on "OLD-TABLE-CATALOG"

Function

You use the MODIFY-OLD-TABLE-CATALOG-LIMIT statement to change the maximum number of entries in the CALL-DML table catalog.

MODIFY-OLD-TABLE-CATALOG-LIMIT
LIMIT = <integer 0..254>

Operands

LIMIT = <integer 0..254>

Maximum number of table entries in the CALL DML table catalog. When 0 is specified CALL-DML accesses are not processed during the DBH session if no entry is assigned. If the value entered is less than the number of entries assigned, the number of assigned entries is taken as the new maximum number.

Administration command in ISP format and at the CALL DML interface

The OPT,CDBC-LIMIT administration command corresponds to the MODIFY-OLD-TABLE-CATALOG-LIMIT administration statement here.

```
OPT,CDBC-LIMIT=n
```

A number between 0 and 254 may be entered for *n*.

5.2.3.30 MODIFY-RECOVER-OPTIONS

Changes the options for subsequent RECOVER or REFRESH runs

Scope of validity

DBH administration

See also

DBH option RECOVER-OPTIONS on "RECOVER-OPTIONS"

Function

You use the MODIFY-RECOVER-OPTIONS statement to change the following for subsequent RECOVER or REFRESH runs:

- The size of the buffer for system-access data
- The size of the buffer for user data
- The storage information for the transaction log files (TA-LOG files)
- The storage information for the restart log file (WA-LOG file)

If you change the storage information for the transaction log files or the restart log file, SESAM/SQL causes all service tasks to delete their TA-LOG or WA-LOG files. The new storage information for the TA-LOG or WA-LOG files then becomes effective upon the next request to the service task.

> ¡ User-defined settings for the TA-LOG or WA-LOG files (CREATE-FILE) are lost when the files are deleted.

```
MODIFY-RECOVER-OPTIONS
RECOVER-OPTIONS = *STD / *PARAMETERS(...)
  *PARAMETERS(...)
    | SYSTEM-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..1000000>
    | ,USER-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..1000000>
    | ,MEDIA-CATALOG = *UNCHANGED / *STD / *PARAMETERS(...)
    |   *PARAMETERS(...)
    |     | TALOG-SUPPORT = *UNCHANGED / *STD / *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
    |     |   *PUBLIC-DISK(...)
    |     |     | CAT-ID = *UNCHANGED / *STD / <cat-id>
    |     |     | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
    |     |     | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
    |     |   *PRIVATE-DISK(...)
    |     |     | CAT-ID = *UNCHANGED / *STD / <cat-id>
    |     |     | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
    |     |     | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
    |     |     | ,DEVICE-TYPE = <structured-name 1..8>
    |     |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
```

```
|     | ,WALOG-SUPPORT = *UNCHANGED / *STD / *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
|     |   *PUBLIC-DISK(...)
|     |     | CAT-ID = *UNCHANGED / *STD / <cat-id
|     |     | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
|     |   *PRIVATE-DISK(...)
|     |     | CAT-ID = *UNCHANGED / *STD / <cat-id>
|     |     | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
|     |     | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
|     |     | ,DEVICE-TYPE = <structured-name 1..8>
|     |     | ,VOLUME = list-poss(6): <alphanum-name 1..6>
```

Operands

RECOVER-OPTIONS = *STD

The default values are used for all options (see the description of the operands below).

RECOVER-OPTIONS = *PARAMETERS(...)

Changes the settings for the options concerned.

SYSTEM-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..1000000>

Size of the buffer for system-access data in a RECOVER or REFRESH run in units of 1 Kbyte.

USER-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..1000000>

Size of the buffer for user data in a RECOVER or REFRESH run in units of 1 Kbyte.

MEDIA-CATALOG = *STD
The TA-LOG and WA-LOG files are created with default values on a public disk.

MEDIA-CATALOG = *PARAMETERS(...)
Changes the settings for the TA-LOG and WA-LOG files.

TALOG-SUPPORT =
Defines the volume for the TA-LOG files.
There are a total of two TA-LOG files (TA-LOG1, TA-LOG2) that are written to one after the other in rotation.

TALOG-SUPPORT = *PUBLIC-DISK(...)
Causes the TA-LOG files to be created on a public disk.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space.
The specified number of PAM pages is reserved immediately. It should correspond to the expected file size.
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY-ALLOCATION = *STD / <integer 1..32767>
Number of PAM pages for subsequent extensions to files
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

TALOG-SUPPORT = *PRIVATE-DISK(...)

The TA-LOG files are to be created on a private disk. To do this, you must make an entry for DEVICE-TYPE.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *STD / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space.
The specified number of PAM pages is reserved immediately. It should correspond to the expected file size.
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

SECONDARY ALLOCATION = *STD / <integer 1..32767>
Number of PAM pages for subsequent extensions to files
The DBH calculates the default value internally. If you enter a value lower than the calculated default, the DBH uses the default.

DEVICE-TYPE = <structured-name 1..8>
BS2000 device type to which the required disks are assigned.
If you make an incorrect entry, the DBH aborts the start procedure.

VOLUME = list-poss(6): <alphanum-name 1..6>
VSNs of the disks on which the files are to be created.
Up to 6 VSNs are permitted, each of which can be specified only once.
VOLUME is not evaluated unless the DEVICE-TYPE parameter is specified.

WALOG-SUPPORT =
Defines the volume for the WA-LOG file.
The operands are identical to those in the operand descriptions for TALOG-SUPPORT.

Administration command in ISP format and at the CALL DML interface

The OPT,RECOVER administration command corresponds to the MODIFY-RECOVER-OPTIONS administration statement here.

```
OPT,RECOVER[,S[YSTEM-DATA]-B[UFFER]=m][,U[SER-DATA]-B[UFFER]=n]
           [,{T[ALOG]-STD | talog-spec}][,{W[ALOG]-STD | walog-spec}]
talog-spec ::= T[ALOG]-V[OL]=(vsn)[,T[ALOG]-D[EV]=device]
               [,T[ALOG]-S[PACE]-P[RIMARY]=space]
               [,T[ALOG]-S[PACE]-S[ECONDARY]=space]
               [,{T[ALOG]-C[ATID]-STD | T[ALOG]-C[ATID]=cat-id}]
walog-spec ::= W[ALOG]-V[OL]=(vsn)[,W[ALOG]-D[EV]=device]
               [,W[ALOG]-S[PACE]-P[RIMARY]=space]
               [,W[ALOG]-S[PACE]-S[ECONDARY]=space]
               [,{W[ALOG]-C[ATID]-STD | W[ALOG]-C[ATID]=cat-id}]
```

If a parameter is not specified, its setting remains unchanged.

The value *0* or *<integer 80..1000000>* may be entered for `m` and `n`.
When `0` is entered the default value of the DBH option is used, i.e. SYSTEM-DATA-BUFFER=*STD or USER-DATA-BUFFER=*STD.

The values defined beforehand for the relevant operands may be entered for *vsn*, *device*, *space* and *cat-id*.

When `TALOG-STD` or `WALOG-STD` is entered, the default values defined beforehand for the relevant operands are used for the file.

Multiple volumes *vsn* are specified separated by commas.
If a file is to reside on a public volume, the word *PUBLIC* (with or without parentheses) is specified in place of (*vsn*).

space is used to specify the values for PRIMARY- and SECONDARY-ALLOCATION. If `0` is specified, the default value is set for these parameters, as it is for`TALOG-STD` and `WALOG-STD`.

The default catalog ID of the DBH is selected with `TALOG-CATID-STD` or `WALOG-CATID-STD`.

In INFORM-PROGRAM administration you must adhere to the permissible length for *talogspec* and *walog-spec* (see section “Administration using INFORM-PROGRAM”).

5.2.3.31 MODIFY-REQUEST-CONTROL

Changes the parameters for priority control

Scope of validity

DBH administration

See also

- DBH option REQUEST-CONTROL on "REQUEST-CONTROL"
- The SET-REQUEST-CONTROL administration statement on "SET-REQUEST-CONTROL"

Function

The MODIFY-REQUEST-CONTROL administration statement allows you to change as required the parameters set by means of the DBH option REQUEST-CONTROL for the priority control of the DBH.

This administration statement is only permitted when the DBH was loaded with priority control.

```
MODIFY-REQUEST-CONTROL

REQUEST-CONTROL = *STD / *QUEUE-RANGE(...) / *QUEUE-PRIORITY(...)
    *QUEUE-RANGE(...)
        |    LOW-LEVEL = 180 / <integer 30..255>
        |    ,HIGH-LEVEL = 240 / <integer 30..255>
    *QUEUE-PRIORITY(...)
        |    QUEUE-1 = 3 / <integer 1..99>
        |    ,QUEUE-2 = 2 / <integer 1..99>
        |    ,QUEUE-3 = 1 / <integer 1..99>
```

Operands

REQUEST-CONTROL = *STD

Defaults apply to all priority-control parameters.

REQUEST-CONTROL = *QUEUE-RANGE(...)
Specifies the limit values between the priority classes Queue-1 and Queue-2, and Queue-2 and Queue-3.

LOW-LEVEL = 180 / <integer 30..255>
Specifies the limit value between the first and second priority classes, Queue-1 and Queue-2. The default for LOW-LEVEL is 180.
This value must be less than the value for HIGH-LEVEL.

HIGH-LEVEL = 240 / <integer 30..255>
Specifies the limit value between the second and third priority classes, Queue-2 and Queue-3. The default for HIGH-LEVEL is 240.
This value must be greater than the value for LOW-LEVEL.

REQUEST-CONTROL = *QUEUE-PRIORITY(...)
Specifies the weights of the different priority classes.

QUEUE-1 = 3 / <integer 1..99>
Specifies the weight of the first priority class, Queue-1.
The default is 3.

QUEUE-2 = 2 / <integer 1..99>
Specifies the weight of the second priority class, Queue-2.
The default is 2.

QUEUE-3 = 1 / <integer 1..99>
Specifies the weight of the third priority class, Queue-3.
The default is 1.

Administration command in ISP format and at the CALL DML interface

Two OPT,REQ-CON administration commands, each with different operands, fulfill the function of the MODIFY-REQUEST-CONTROL administration statement.

`OPT,REQ-CON,H[IGH-LEVEL]=`*n*`[,L[OW-LEVEL]=`*m*`]`

`OPT,REQ-CON,Q[UEUE]-1=`*x*`[,Q[UEUE]-2=`*y*`][,Q[UEUE]-3=`*z*`]`

5.2.3.32 MODIFY-RESTART-CONTROL

This statement influences the duration of a restart.

Scope of validity

DBH administration

See also

DBH option RESTART-CONTROL on "RESTART-CONTROL"

Function

The MODIFY-RESTART-CONTROL administration statement allows you to modify the BUFFER-LIMIT and TALOG-LIMIT parameters specified using the DBH option RESTART-CONTROL.

You can use these parameters to reduce the duration of any restart by influencing the writing of afterimages to the database. It is possible to select different values for the buffers of the system access data and user data. Parameters not specified in the MODIFY-RESTART-CONTROL statement remain unchanged.

Low parameter values cause the afterimages to be written to the database frequently. This reduces the duration of physical repair in the event of a restart. However, if performance losses are incurred as a result of parameter values being selected that are too low and of an increased number of physical read and write accesses, you can use MODIFY-RESTART-CONTROL to adjust the parameter values during operation.

```
MODIFY-RESTART-CONTROL

RESTART-CONTROL = *BUFFER-LIMIT(...) / *TALOG-LIMIT(...)

  *BUFFER-LIMIT(...)

    | SYSTEM-DATA-BUFFER = 50 / <integer 1..90>

    | ,USER-DATA-BUFFER = 50 / <integer 1..90>

  *TALOG-LIMIT(...)

    | LIMIT = 128 / <integer 128..524288>
```

Operands

RESTART-CONTROL =

Specifies which value is to be changed.

RESTART-CONTROL = *BUFFER-LIMIT(...)
Controls the physical writing of the afterimages to the database.

SYSTEM-DATA-BUFFER = 50 / <integer 1..90>
Specifies a percentage limit for the writing of afterimages to the database for system access data. If the share of the total buffer space occupied by the buffers to be written exceeds the percentage specified for SYSTEM-DATA-BUFFER, afterimages are written to the database.

USER-DATA-BUFFER = 50 / <integer 1..90>
Specifies a percentage limit for the writing of afterimages to the database for user data. If the share of the total buffer space occupied by the buffers to be written exceeds the percentage specified for USER-DATA-BUFFER, afterimages are written to the database.

RESTART-CONTROL = *TALOG-LIMIT(...)

Specifies the limit for how much of the TA-LOG file can be occupied by afterimages. As of this limit, afterimage blocks are written to the database.

LIMIT = 512 / <integer 128..524288>

Specifies the new limit value in units of 4 Kbytes.

Administration command in ISP format and at the CALL DML interface

Three administration commands with the format OPT,RESTART, each with specific operands, correspond to the MODIFY-RESTART-CONTROL administration statement here.

```
OPT,RESTART,B[UFFER-LIMIT],U[SER-DATA-BUFFER]=n[,S[YSTEM-DATA-BUFFER]=m]
OPT,RESTART,B[UFFER-LIMIT],S[YSTEM-DATA-BUFFER]=n[,U[SER-DATA-BUFFER]=m]
OPT,RESTART,T[ALOG-LIMIT]=l
```

Percentage values expressed as integers between 1 and 90 can be specified for n and m.

An integer between 128 and 524288 can be specified for l.

5.2.3.33 MODIFY-RETRIEVAL-CONTROL

This statement changes the limit value for the interruption of retrieval statements.

Scope of validity

DBH administration

See also

DBH option RETRIEVAL-CONTROL on "RETRIEVAL-CONTROL"

Function

The MODIFY-RETRIEVAL-CONTROL administration statement allows you to change the parameters INDEX-EVALUATION, USER-DATA-ACCESS and ABORT-EXECUTION of the DBH option RETRIEVAL-CONTROL to suit requirements.

```
MODIFY-RETRIEVAL-CONTROL

SELECT = *INDEX-EVALUATION(...) / *USER-DATA-ACCESS(...) / *ABORT-EXECUTION(...)
    *INDEX-EVALUATION(...)
        |    LIMIT = <integer 16..2147483647>
    *USER-DATA-ACCESS(...)
        |    LIMIT = <integer 1..2147483647>
    *ABORT-EXECUTION(...)
        |    LIMIT = <integer 1..2147483647>
```

Operands

SELECT = *INDEX-EVALUATION(...)

Specifies a limit value as of which the processing of retrieval statements is interrupted via secondary indexes and continued sequentially. The limit value refers to an expected set of records found, which the DBH calculates for the first subquestion to be processed.

LIMIT = <integer 16..2147483647>
Specifies the new limit value.

SELECT = *USER-DATA-ACCESS(...)

Specifies the number of logical USER-DATA inputs/outputs for CALL DML and SQL before there is an interruption.

LIMIT = <integer 1..2147483647>
Specifies the new limit value.

SELECT = *ABORT-EXECUTION(...)
Specifies the number of failed logical USER-DATA inputs/outputs for CALL-DML and SQL as of which the search is aborted.

A request related value defined using the pragma LIMIT ABORT_EXECUTION <integer 1. 2147483647> cannot be changed with this statement, see the “ SQL Reference Manual Part 1: SQL Statements”, section “Pragmas”.

LIMIT = <integer 1..2147483647>
Specifies the new limit value.

Administration command in ISP format and at the CALL DML interface

The OPT,RET-CON administration command corresponds to the MODIFY-RETRIEVAL-CONTROL administration statement here.

```
OPT,RET-CON,INDEX-EVAL=n
OPT,RET-CON,USER-DATA-ACC=y
OPT,RET-CON,ABORT=z
```

A number between 16 and 2^{31}-1 can be specified for n.

A number between 1 and 2^{31}-1 can be specified for y.

A number between 1 and 2^{31}-1 can be specified for z.

5.2.3.34 MODIFY-SECURITY

Changes the maximum permissible number of password violations

Scope of validity

DBH administration

See also

The DBH option SECURITY on "SECURITY"

Function

For users operating in interactive mode you use the MODIFY-SECURITY statement to change the maximum permissible number of password violations by a CALL-DML user.

A reduction of the maximum number means that the user is locked earlier in the event of password violations. For users with more password violations than the new maximum number this means that they are immediately locked for this DBH session if they have not opened any logical file.

Increasing the maximum number has no effect on locked users whose number of password violations already exceeds the new maximum number. On the other hand, locked users who are guilty of fewer password violations than the new maximum number are now permitted to work with the DBH again. Such a user can also perform a user close. It is no longer necessary to have the system administrator cancel this lock.

This administration statement has no effect on batch programs.

```
MODIFY-SECURITY

SECURITY = *PRIVILEGE-VIOLATIONS(...)

  *PRIVILEGE-VIOLATIONS(...)

    | NUMBER = 10 / <integer 1..99>
```

Operands

SECURITY = *PRIVILEGE-VIOLATIONS(NUMBER = 10 / <integer 1..99>)

Changes the maximum permissible number of password violations.

NUMBER = 10 / <integer 1..99
Specifies the maximum permissible number of password violations.

Administration command in ISP format and at the CALL DML interface

The OPT,SECURITY administration command corresponds to the MODIFY-SECURITY administration statement here.

```
OPT,SECURITY[,P[RIVILEGE]-V[IOLATIONS]=n]
```

A number between 1 and 99 may be entered for *n*.

5.2.3.35 MODIFY-SERVICE-TASKS

This statement changes the number and attributes of the service tasks.

Scope of validity

DBH administration

See also

DBH option SERVICE-TASKS on "SERVICE-TASKS"

Function

You use the MODIFY-SERVICE-TASKS statement to change the number and attributes of the service tasks which may be started during a session.

Information about the utilization of the service tasks in the current session is provided by the SESAM/SQL utility routine SESMON in the SERVICE TASKS form (see "DBH form SERVICE TASKS").

```
MODIFY-SERVICE-TASKS

INITIAL = *UNCHANGED / *STD / <integer 0..32>
,MAXIMUM = *UNCHANGED / *STD / <integer 1..64>
,JOBCLASS = *UNCHANGED / *STD / *DBH-JOBCLASS / <name 1..8>
,WORK-FILES = *UNCHANGED / *STD / *PUBLIC-DISK(...) / *PRIVATE-DISK(...)
  *PUBLIC-DISK(...)
    | CAT-ID = *UNCHANGED / *STD / <cat-id>
    | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
    | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
  *PRIVATE-DISK(...)
    | CAT-ID = *UNCHANGED / *STD / <cat-id>
    | ,PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>
    | ,SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>
    | ,DEVICE-TYPE = <structured-name 1..8>
    | ,VOLUME = list-poss(6):<alphanum-name 1..6>
,RECORDS-PER-CYCLE = *UNCHANGED / *STD / <integer 1..2147483647>
,MAIL = *UNCHANGED / *NONE / *ALL / *ERROR
```

Operands

INITIAL = *UNCHANGED / *STD / <integer 0..32>

Minimum number of service tasks which should exist. If fewer service tasks exist, additional ones are started until this value is reached. If you specify *STD the default value of the DBH option (one service task) is assumed. In the case of a linked-in DBH this parameter is ignored.

MAXIMUM = *UNCHANGED / *STD / <integer 1..64>
Maximum number up to which further service tasks may be started during the DBH session. If more service tasks already exist, the next service task is started only when this value is fallen below. If you specify *STD the default value of the DBH option (see "SERVICE-TASKS") is assumed. In the case of a linked-in DBH this parameter is ignored.

> Note that if the values of INITIAL and MAXIMUM are increased it may not be possible to start enough service tasks with the new attributes if the BS2000 job class does not permit this.
> Note that if MAXIMUM is reduced without changing the attributes for service tasks it may initially not be possible to start any service tasks because more than the maximum number of service tasks are running. These are not terminated prematurely by SESAM/SQL.
>
> Note that if MAXIMUM is reduced and the attributes for service tasks are changed the running service tasks can only be terminated after the request has terminated. This can mean that initially no service tasks with the new attributes can be started. See also the note on "MODIFY-SERVICE-TASKS".

JOBCLASS = *UNCHANGED / *STD / *DBH-JOBCLASS / <name 1..8>

Name of the BS2000 job class.

When *STD is specified the batch job is to run in the standard job class. The standard job class is the default job class for batch jobs of the DBH identifier. It is taken from the JOIN entry of the user ID. When *DBH-JOBCLASS is specified the name of the BS2000 job class is the job class of the DBH when the service task is started, provided the DBH is started as a batch job. In interactive mode the specification of *DBH-JOBCLASS is ignored and the procedure is the same as for JOBCLASS=*STD.

> Note that if the BS2000 job class is changed the system settings for the service tasks (e.g. CPU-LIMIT, CLASS-LIMIT) can change. This may mean that not enough service tasks with the new attributes are started.

WORK-FILES = *UNCHANGED / *STD / *PUBLIC-DISK(...) / *PRIVATE-DISK(...)

Temporary SORT work files which are generated within a service task are created before each sort run and deleted after it. If you specify *STD the default value of the DBH option (see "SERVICE-TASKS") is assumed.

WORK-FILES = *PUBLIC-DISK(...)
Temporary SORT work files are stored on a public disk.

CAT-ID = *UNCHANGED / *STD / <cat-id>
Catalog ID of the pubset on which the temporary SORT work files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>

Number of PAM pages for the initial allocation of memory space. The default value is 120.

SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>

Number of PAM pages for subsequent extensions to files The default value is 120.

WORK-FILES = *PRIVATE-DISK(...)
Temporary SORT work files are stored on a private disk.

CAT-ID = *UNCHANGED / *STD / <cat-id>
Catalog ID of the pubset on which the temporary SORT work files are to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..16777215>

Number of PAM pages for the initial allocation of memory space. The default value is 120.

SECONDARY-ALLOCATION = *UNCHANGED / *STD / <integer 1..32767>

Number of PAM pages for subsequent extensions to files The default value is 120.

DEVICE-TYPE = <structured-name 1..8>
BS2000 device type to which the required disks are assigned. If you make an incorrect entry, the system prompts you in interactive mode to make the entry again (see the device type table in the BS2000 manual “ Commands”).

VOLUME = list-poss(6): <alphanum-name 1..6>
VSNs of the disks on which the file is to be created. Up to 6 VSNs are permitted, each of which can be specified only once. VOLUME is not evaluated unless the DEVICE parameter is specified.

> Note that if PRIMARY-ALLOCATION or SECONDARY-ALLOCATION is increased this may result in an increased storage requirement.

RECORDS-PER-CYCLE = *UNCHANGED / *STD / <integer 1..2147483647>

Number of records to be sorted per cycle by a sort subtask during multitask sorting. SESAM/SQL passes this value to the BS2000 utility SORT in the CYCLE parameter (see also the SORT manual “ SDF Format”). The “ Performance” manual describes the criteria on the basis of which the RECORDS-PER-CYCLE value should be assigned. The total number of records to be sorted divided by CYCLE is the optimal number of work files to be created. However, no more than 9 temporary work files can be created. The number of work files determines the number of sort subtasks started by the SORT utility (number of sort subtasks = number of work files minus 1). In single-task sorting, work files are not created unless one of the following parameters is specified: CAT-ID, PRIMARY-ALLOCATION or DEVICE-TYPE.

*STD means that the DBH does not use the multitask sorting facility.

MAIL = *UNCHANGED / *NONE / *ALL / *ERROR
Controls the output of information by email.

MAIL = *NONE
The information is not sent by email.

MAIL = *ALL
When the service task terminates its SYSLST file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email"). The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST (SVT, TSN=<tsn of the service task>” is entered under “Subject:”.The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The SYSLST file is attached to the email.

MAIL = *ERROR
Only if the server task terminates with an error its SYSLST file is sent by email to the address which is entered in the MAIL parameter file (see "Sending important information of the DBH session by email") eingetragen ist. The MAIL parameter file must have been assigned with the link name SESMAIL when the DBH started.

“SYSLST (SVT, TSN=<tsn of the service task>” is entered under “Subject:”. The sender is the sender address entered in the MAIL parameter file under “SENDER=”. The SYSLST file is attached to the email.

> **i** You use the JOB-CLASS, WORK-FILES and RECORDS-PER-CYCLE parameters to change the attributes of a service task. To enable these to become effective as soon as possible, SESAM/SQL immediately terminates any inactive service tasks. Active service tasks are flagged for termination after the current job has been processed. The attributes then become effective with the next job.

Administration command in ISP format and at the CALL DML interface

The OPT,SVT administration command corresponds to the MODIFY-SERVICE-TASKS administration statement here.

```
OPT,SVT[,{INI[TIAL]-STD | INI[TIAL]=m}][,{MAX[IMUM]-STD | MAX[IMUM]=n}]
       [,J[OB]-C[LASS]={STD|DBH|(name)}][,R[ECORDS]-P[ER]-C[YCLE]=r]
       [,{W[ORK-FILE]-STD|file-spec}][,MAIL={NONE | ALL | ERROR}]
file-spec ::= W[ORK-FILE]-V[OL]=(vsn)[,D[EV]=device]
              [,S[PACE]-P[RIMARY]=space][,S[PACE]-S[ECONDARY]=space]
              [,{C[ATID]-STD | C[ATID]=cat-id}]
```

At least one parameter must be specified. If a parameter is not specified, its setting remains unchanged.

The values defined beforehand for the relevant operands may be entered for *m*, *n*, *r*, *name*, *vsn*, *device*, *space* and *cat-id*.

When `WORK-FILE-STD` is entered, the default values defined beforehand for the relevant operands are used for the file.

The MAIL parameters are described under the MAIL operand.

Multiple volumes *vsn* for the work file are specified separated by commas.

If the work file is to reside on a public volume, the word *PUBLIC* (with or without parentheses) is specified in place of (*vsn*).

space is used to specify the values for PRIMARY- and SECONDARY-ALLOCATION. If `0` is specified, the default value of the DBH is set for these parameters, as it is for `WORKFILE-STD`.

The default catalog ID of the DBH is selected with `CATID-STD`.

In INFORM-PROGRAM administration you must adhere to the permissible length for *filespec* (see section “Administration using INFORM-PROGRAM”).

5.2.3.36 MODIFY-SESSION-LOGGING-ID

Changes the identification of session-related files

Scope of validity

DBH administration

See also

DBH option SESSION-LOGGING-ID on "SESSION-LOGGING-ID"

Function

You use the MODIFY-SESSION-LOGGING-ID statement to change the session identification and the file counter of the CO-LOG file.

The next time request logging takes place (see “SET-TUNING-TRACE”) the changed identifications for the CO-LOG file are taken into account. This administration statement cannot be used when request logging is enabled.

MODIFY-SESSION-LOGGING-ID
SESSION-ID = *UNCHANGED / *STD / <alphanum-name 4..4> ,START-NUMBER = *UNCHANGED / <integer 1..9999>

Operands

SESSION-ID =

Specifies the session identification for the CO-LOG file.

SESSION-ID = *STD
The default value for the session identification corresponds to the BS2000 task sequence number (TSN).

SESSION-ID = <alphanum-name 4..4>
Session identification that you yourself can specify.

START-NUMBER =

Specifies the initial value of the file counter for the CO-LOG file.

START-NUMBER = <integer 1..9999>
Initial value of the file counter that you yourself can specify.
The four-digit value is used in filenames (e.g. 0003).
If an overflow occurs (> 9999), 0001 is used again.

Administration command in ISP format and at the CALL DML interface

The OPT,LOG-ID administration command corresponds to the MODIFY-SESSION-LOGGING-ID administration statement here.

```
OPT,LOG-ID[,S[ESSION-ID]=m][,S[TART]-N[UMBER]=n]
```

If a parameter is not specified, its setting remains unchanged.

Precisely four alphanumeric characters or *TSN* (corresponds to the SESSION-ID=*STD operand) must be entered for `m`.

A number between 0 and 9999 may be entered for *n*.
When 0 is specified, the standard initial value 0001 is used for the file counter.

5.2.3.37 MODIFY-SQL-SORT-LIMIT

This statement changes the maximum number of records in the sorting of a cursor table.

Scope of validity

DBH administration

See also

DBH option SQL-SUPPORT on "SQL-SUPPORT"

Function

The MODIFY-SQL-SORT-LIMIT administration statement allows you to change the INTERNAL-SORT-LIMIT parameter of the DBH option SQL-SUPPORT to suit requirements.

This statement is only permitted when the DBH supports the SQL interface.

MODIFY-SQL-SORT-LIMIT
INTERNAL-SORT-LIMIT = <integer 1..2147483647>

Operands

INTERNAL-SORT-LIMIT = <integer 1..2147483647>

Specifies the maximum number of records a cursor table can contain if the records are to be sorted in accordance with the cursor declaration. If the number of records found during the sort exceeds this limit value, processing of the statement is aborted.

Administration command in ISP format and at the CALL DML interface

The OPT,SORTLIMIT administration command corresponds to the MODIFY-SQL-SORT-LIMIT administration statement here.

```
OPT,SORTLIMIT=n
```

5.2.3.38 MODIFY-STORAGE-SIZE

Changes the maximum size of transfer and work containers

Scope of validity

DBH administration

See also

DBH option STORAGE-SIZE on "STORAGE-SIZE"

Function

You use the MODIFY-STORAGE-SIZE statement to change the maximum size of transfer and work containers.

If the maximum value for the SUBORDERS option is increased (see “MODIFY-SUBORDER-LIMIT”), the maximum values for transfer and work containers do not change automatically. They can be adjusted using this statement.

A change to the maximum value for transfer and work containers depends on the current, possibly increased maximum value for the SUBORDERS options (see formulas on "TRANSFER-CONTAINER" and "WORK-CONTAINER"). A new maximum value for transfer and work containers must at least match the value obtained using one of the related formulas. If a lower value is specified, the minimum value obtained using the relevant formula applies.

> ¡ Consequently, despite a lower value being specified, the maximum value for transfer and work containers can increase if the maximum value for SUBORDERS was increased beforehand.

Information about transfer and work containers is provided in the operational statistics of the SESAM/SQL monitor SESMON. Details on the containers can be found in the “SYSTEM-INFORMATION” form (see "DBH form SYSTEM INFORMATION").

```
MODIFY-STORAGE-SIZE

STORAGE = *TRANSFER-CONTAINER(...) / *WORK-CONTAINER(...) ,
    *TRANSFER-CONTAINER(...)
        |    MAXIMUM = <integer 64..1000000>
    *WORK-CONTAINER(...)
        |    MAXIMUM = <integer 24..1000000>
```

Operands

STORAGE = *TRANSFER-CONTAINER(MAXIMUM = <integer 64..1000000>)

Defines the maximum size which the transfer container may reach.

> ¡ If the new maximum value is lower than the value of the relevant formula on "TRANSFER-CONTAINER", the value of the formula is used as the new maximum value.

STORAGE = *WORK-CONTAINER(MAXIMUM = <integer 24..1000000>)

Defines the maximum size which the work container may reach.

> If the new maximum value is lower than the value of the relevant formula on "WORK-CONTAINER", the value of the formula is used as the new maximum value.

Administration command in ISP format and at the CALL DML interface

The OPT,STO-SIZE administration command corresponds to the MODIFY-STORAGE-SIZE administration statement here.

```
OPT,STO-SIZE,{T[RANSFER]-C[ONT]=m | W[ORK]-C[ONT]=n}
```

A number between 64 or 24 and 1000000 may be entered for *m* or *n* respectively.

5.2.3.39 MODIFY-SUBORDER-LIMIT

This statement limits the maximum number of concurrent suborders.

Scope of validity

DBH administration

See also

The DBH option SUBORDERS on "SUBORDERS"

Function

The MODIFY-SUBORDER-LIMIT administration statement allows you to change the MAXIMUM parameter of the DBH option SUBORDERS to suit requirements.

You use this statement to change the maximum permitted number of suborders (SQL scans or logical files) in the DBH session.

MODIFY-SUBORDER-LIMIT
LIMIT = <integer 6..262143>

Operands

LIMIT = <integer 6..262143>

Specifies the maximum permissible number of concurrent orders.

When you enter a value for LIMIT which is lower than the total of all suborders which currently exist, SESAM/SQL determines an internal value for LIMIT. This value corresponds to the total of all current suborders of all users.

The number of parallel suborders cannot fall any lower than the number reached during the currently running DBH session.

Administration command in ISP format and at the CALL DML interface

The OPT,SUBORDERS administration command corresponds to the MODIFY-SUBORDER-LIMIT administration statement here.

```
OPT,SUBORDERS=n
```

5.2.3.40 MODIFY-TRANSACTION-SECURITY

Changes the transaction security parameters

Scope of validity

DBH administration

See also

DBH option TRANSACTION-SECURITY on "TRANSACTION-SECURITY"

Function

The MODIFY-TRANSACTION-SECURITY statement allows you to change the LOCK-TIME parameter, the INACTIVITY-TIME parameter or the LOCK-ESCALATION parameter of the DBH option TRANSACTION-SECURITY during operation.

Only the parameter that is specified in the MODIFY-TRANSACTION-SECURITY statement is changed. All the other values remain unchanged. The values changed for the LOCK-ESCALATION parameter do not take effect until a transaction accesses a table or index after the modification. The changes have no effect on transactions that already have access to a table or index when the option is changed.

```
MODIFY-TRANSACTION-SECURITY

SELECT = *LOCK-TIME(...) / *INACTIVITY-TIME(...) / *LOCK-ESCALATION(...)
   *LOCK-TIME(...)
      |  TIME = 4 / <integer 1..999>
   *INACTIVITY-TIME(...)
      |  TIME = 40 / <integer 1..999>
   *LOCK-ESCALATION(...)
      |  NUMBER-RECORDS = 4000 / <integer 1..2147483647>
      |  ,PERCENTAGE-RECORDS = 50 / <integer 0..100>
      |  ,NUMBER-INDEX-VALUES = 1000 / <integer 1..2147483647>
```

Operands

SELECT = *LOCK-TIME(...)

Changes the time after which an inactive transaction that is locking other transactions is to be rolled back.

TIME = 4 / <integer 1..999>
New value in minutes. The value specified for LOCK-TIME must not be greater than the current value of INACTIVITY-TIME. If it is, the statement is rejected.

SELECT = *INACTIVITY-TIME(...)
Specifies the number of minutes after which an open but inactive transaction is to be rolled back.

TIME = 40 / <integer 1..999>
New value in minutes. The value specified for INACTIVITY-TIME must not be lower than the current value of LOCK-TIME. If it is, the statement is rejected.

SELECT = *LOCK-ESCALATION(...)

Changes the limit values as of which locks escalate. Escalation is when SESAM/SQL attempts to lock the entire index or table.

NUMBER-RECORDS = 4000 / <integer 1..2147483647>

Specifies the number of records in a table that a user can lock. If more records than this are locked, the whole table is locked.

PERCENTAGE-RECORDS = 50 / <integer 0..100>

Specifies the percentage of all the records in the table that a user can lock. If a greater percentage is locked, SESAM/SQL attempts to lock the whole table. If 0 is specified as the percentage, the entire table is locked immediately.

NUMBER-INDEX-VALUES = 1000 / <integer 1..2147483647>

Specifies the number of values that a user can lock in an index. If a larger number than this are locked, an attempt is made to lock the entire index rather than just individual values.

Administration command in ISP format and at the CALL DML interface

Five administration commands with the format OPT,TA-SEC, each with specific operands, correspond to the MODIFY-TRANSACTION-SECURITY administration statement here.

```
OPT,TA-SEC,L[OCK-TIME]=n
OPT,TA-SEC,I[NACT-TIME]=m
OPT,TA-SEC,E[SCALATE-LOCK],N[UMBER]-R[ECORDS]=l
        {[,P[ERCENTAGE]-R[ECORDS]=k][,N[UMBER]-I[NDEX-VALUES]=j]|
         [,N[UMBER]-I[NDEX-VALUES]=j][,P[ERCENTAGE]-R[ECORDS]=k]}
OPT,TA-SEC,E[SCALATE-LOCK],P[ERCENTAGE]-R[ECORDS]=k
        {[,N[UMBER]-R[ECORDS]=l][,N[UMBER]-I[NDEX-VALUES]=j]|
         [,N[UMBER]-I[NDEX-VALUES]=j][,N[UMBER]-R[ECORDS]=l]}
OPT,TA-SEC,E[SCALATE-LOCK],N[UMBER]-I[NDEX-VALUES]=j
        {[,P[ERCENTAGE]-R[ECORDS]=k][,N[UMBER]-R[ECORDS]=l]|
         [,N[UMBER]-R[ECORDS]=l][,P[ERCENTAGE]-R[ECORDS]=k]}
```

A number from 1 to 999 can be specified for n and m.

A number from 1 to 2^{31}-1 can be specified for l and j.

A percentage value from 0 to 100 can be specified for k.

5.2.3.41 PREPARE-FOREIGN-COPY

This statement closes spaces of a database logically and physically in order to prepare for the creation of a foreign copy.

Scope of validity

DBH administration

See also

- DBH administration statement END-FOREIGN-COPY on "END-FOREIGN-COPY"
- DBH administration statement BEGIN-LOCK-SEQUENCE on "BEGIN-LOCK-SEQUENCE"
- DBH administration statement END-LOCK-SEQUENCE on "END-LOCK-SEQUENCE"

Function

At the time when a foreign copy is created, the spaces concerned must be properly closed. You can use the PREPARE-FOREIGN-COPY statement to close the spaces logically and physically.

In logical closure, the update is interrupted on the space by a transaction lock. The contents of the buffers of the selected database are written to the corresponding files. These files remain physically open.

As long as the database is not updated, a foreign copy can be created. Updates must be prevented by organizational means or by issuing the PREPARE-FOREIGN-COPY statement in a lock sequence, which is better. Read access to the database is also possible during the lock sequence.
Lock sequences are opened with BEGIN-LOCK-SEQUENCE and terminated with END-LOCK-SEQUENCE.

In the case of physical closure the contents of the buffers of the selected database are retained. This means that foreign copies are also possible which require the database files to be physically closed (e.g. SNAP dump). The database is placed in the CLOSED status. In this status accesses to the database are rejected with SQLSTATE. This remains invisible for application programs if the statement sequence PREPARE-FOREIGN-COPY (with CLOSE) and END-FOREIGN-COPY is used within a lock sequence.

The logging for a space, a space set or the entire catalog can be turned on with the statement PREPARE-FOREIGN-COPY. The logging information can be used for repair with foreign copies generated after the statement PREPARE-FOREIGN-COPY.

The statement PREPARE-FOREIGN-COPY can be used independently of the statement END-FOREIGN-COPY.

If the statement END-FOREIGN-COPY follows in the same administration session, this only acts on the spaces entered in the preceding statement PREPARE-FOREIGN-COPY.

If the logging is only to be turned on for a few spaces and other spaces are to be backed up as a space set, a PREPARE-FOREIGN-COPY statement must first be given for turning on the logging. This is then followed by another PREPARE-FOREIGN-COPY statement with all the spaces to be backed up. The final END-FOREIGN-COPY statement then acts on all the spaces backed up.

```
PREPARE-FOREIGN-COPY

SELECT = *LOGICAL(...) / *PHYSICAL(...)

  *LOGICAL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *PHYSICAL(...)
```

```
      |  PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
,SPACE-NAMES = *ALL(...) / *CATALOG / *FROM-FILE(...)
               list-poss(200): <filename 1..18 without-cat-user-gen-vers>
  *ALL(...)
      |  CLOSE = *NO / *YES
  *FROM-FILE(...)
      |  FILE-NAME = <filename 1..54>
,LOGGING = *UNCHANGED / *ON
```

Operands

SELECT =

Specifies the name by means of which the database is identified.

SELECT = *LOGICAL(...)
Selects the database to be logically closed.

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical database name

SELECT = *PHYSICAL(...)
Selects the database to be logically closed.

PHYS-CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Physical database name

SPACE-NAMES =

Specifies the spaces to be closed.

The backups of the spaces closed with this statement can be entered in a space list in the utility statement RECOVER (using foreign copy); see the utility statement RECOVER in the “ SQL Reference Manual Part 2: Utilities ” manual.

SPACE-NAMES = *ALL(...)
All user spaces and the catalog space are logically closed.

CLOSE = *NO
The database files remain physically open.

CLOSE = *YES
The database files are also physically closed.
The database is placed in the CLOSED status. An END-FOREIGN-COPY statement must be entered in the same DBH session.

SPACE-NAMES = *CATALOG
Only the catalog space is logically closed. You can then only generate a consistent foreign copy of the catalog space.

SPACE-NAMES = *FROM-FILE(...)
Selects the spaces which are to be logically closed by listing them in a file.

FILE-NAME = <filename 1..54>
File name of a SAM file which contains a space name in each line. Up to 999 spaces can be specified in this way.

Example

File with a list of space names which was created with the EDT editor:

```
TABLESPACE
INDEXSPACE
BLOBSPACE
```

SPACE-NAMES =
list-poss(200): <filename 1..18 without-cat-user-gen-vers>
The user spaces entered are logically closed. You can then only generate a consistent foreign copy of these spaces.

LOGGING =

Specifies whether the logging is turned on for the spaces specified in the operand SPACE-NAMES.

LOGGING = *UNCHANGED
The settings for the logging remain unchanged.

LOGGING = *ON
The logging is turned on for the spaces specified in the operand SPACE-NAMES. Logging for individual spaces can only be turned on if the catalog is already logging. The backups on the spaces which are closed with this statement form the basis of the logging.

Administration command in ISP format and at the CALL DML interface

The FCOPY,PREPARE administration command corresponds to the PREPARE-FOREIGN-COPY administration statement here.

```
FCOPY,P[REPARE],supplement
       [,{space|CLOSE}][,L[OGGING-ON]]
supplement ::= {C[ATALOG]=catalog-name|
                P[HYSICAL-NAME]=physical-name}
space ::= S[PACE]={(space-name,...)|C[ATALOG]}
```

Up to 999 user spaces can be specified for *space-name*. If only one *space-name* is specified, the parentheses can be omitted. The values defined beforehand in the relevant operands may be entered for *catalog-name* and *physical-name*.

In INFORM-PROGRAM administration you must adhere to the permissible length for the command (see section “Administration using INFORM-PROGRAM”).

5.2.3.42 RECONFIGURE-DBH-SESSION

This statement changes DBH options in the current DBH session.

Scope of validity

DBH administration

See also

- DBH administration statement RELOAD-DBH-SESSION on "RELOAD-DBH-SESSION"
- The DBH option COLUMNS on "COLUMNS"
- The DBH option CURSOR-BUFFER on "CURSOR-BUFFER"
- The DBH option SPACES on "SPACES"
- The DBH option SQL-SUPPORT on "SQL-SUPPORT"
- The DBH option SYSTEM-DATA-BUFFER on "SYSTEM-DATA-BUFFER"
- The DBH option TRANSACTION-SECURITY on "TRANSACTION-SECURITY"
- The DBH option USER-DATA-BUFFER on "USER-DATA-BUFFER"

Function

See section “Dynamic reconfiguration of the DBH session”.

You can use the RECONFIGURE-DBH-SESSION statement to change some DBH options in the current DBH session without reloading the DBH modules. The statement cannot be specified in a linked-in DBH.

```
RECONFIGURE-DBH-SESSION

CURSOR-BUFFER = *UNCHANGED / *STD / *PARAMETERS(...)
   *PARAMETERS(...)
      | BUFFER-SIZE = *UNCHANGED / *STD / <integer 72..1500000>
      | ,FRAME-SIZE = *UNCHANGED / *STD / <integer 4..32>
,SYSTEM-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..64000000>
,USER-DATA-BUFFER = *UNCHANGED / *STD / <integer 80..64000000>
,SQL-SUPPORT = *UNCHANGED / *STD / *PARAMETERS(...)
   *PARAMETERS(...)
      | CURSORS = *UNCHANGED / *STD / <integer 0..262143>
      | ,PLANS = *UNCHANGED / *STD / <integer 1..999999>
,SPACES = *UNCHANGED / *STD / <integer 1..101600>
,COLUMNS = *UNCHANGED / *STD / <integer 256..1024>
,TRANSACTION-SECURITY = *UNCHANGED / *STD / *PARAMETERS(...)
   *PARAMETERS(...)
      | MAX-ISOLATION-LEVEL = *UNCHANGED / *STD / *REPEATABLE-READ / *SERIALIZABLE
```

Operands

See the operand descriptions for the relevant DBH options.

When *UNCHANGED is specified, the value of the DBH option remains unchanged. When *STD is specified, the default value of the DBH option is used.

Further information

The final size of the buffer for system access data and of the buffer for user data is hardware-dependent and also dependent on the DBH option THREADS and the system threads which depend on this. The minimum value for the buffer size is 96 KB per thread. If the value specified is less than the minimum value, it is automatically increased to the minimum value provided the value specified is within the permitted range.

The total of the sizes for SYSTEM-DATA-BUFFER and USER-DATA-BUFFER may not exceed 64 000 000 KB (64 GB).

The minimum size of CURSOR-BUFFER per thread is dependent on the value of FRAME-SIZE and must in all cases be eight times the value of FRAME-SIZE. If more is specified by the user, the user specifications apply. However, these are always rounded up to a multiple of the FRAME-SIZE value.

The value for SPACES depends on the maximum number of entries which the SQL database directory may contain. When you enter a value for SPACES which is lower than the total of the spaces which currently exist, SESAM/SQL selects an appropriate value for SPACES. This is basically the same as the total number of spaces of all SQL database directories which are currently active.

When the value of the DBH option COLUMNS is increased, the maximum size of the work container should be adjusted accordingly in order to prevent resource bottlenecks in the WORK-CONTAINER.

In the case of RECONFIGURE-DBH-SESSION the DBH’s memory is restructured or recreated. This also causes the counters which the SESMON performance monitor evaluates for statistical purposes to be reset to “0”.

5.2.3.43 RELEASE-USER-RESOURCES

This statement releases all a user's resources.

Scope of validity

DBH administration

Function

The RELEASE-USER-RESOURCES statement releases all a user's resources.

You can use this statement to intervene in the case of the undefined abortion of an application program, for example, by releasing the resources of the associated user and thereby disconnecting this user.

You can determine all of the users known to the DBH who have no open transactions and who have not issued any statements for a certain length of time, and then release all the resources of these users.

In a DBH session with transaction management, the DBH rolls back all a user's currently active transactions before releasing the user's resources. If the resources of a UTM user are to be released, the DBH backs up all the information on the rolling back of transactions that have started in order to ensure a synchronized UTM restart.

All users are rolled back that have a (partially qualified) user identifier. If a user is currently inside of a transaction, then this transaction is rolled back if it does not issue any commands within the time specified in INACTIVE-TIME.

If a utility is currently active for the user, then the call for this user is ignored.

The system administrator cannot release his or her own resources. If SESAM/SQL recognizes the user identification as being that of the system administrator, then the call for this user is ignored.

```
RELEASE-USER-RESOURCES

USER = *ALL (...) / <user-identification>
   *ALL (...)
     |  INACTIVE-TIME = <integer 1 .. 99>
```

Operands

USER =

Identifies the user or a group of users when partially qualified whose resources are to be released.

USER = *ALL (...)
If *ALL is specified, all the resources are released for all of the users known to the DBH who have no open transactions and who have not issued any statements since the time specified with INACTIVE-TIME.

INACTIVE-TIME = <integer 1 .. 99>
Time in minutes.

USER = <user-identification>
Identifies one or more users whose resources are to be released. One partial qualification is enough to cause the resources of all affected users to be released (see the section “Syntax for the identification of users”).

Administration command in ISP format and at the CALL DML interface

Here, two administration commands fulfill the function of the RELEASE-USER-RESOURCES administration statement, namely the administration commands FREE and FREE,USER,INACT.

```
FREE,user-identification
```

```
FREE,USER,INACT=n
```

A 2-digit number between 1 and 99 can be specified for *n*.

> Users that have open transactions or active lock sequences are not affected by the FREE,USER,INACT call.
> The call handles CALL DML users as well as SQL users.
>
> *user-identification* can be specified as a partial qualification (see "Syntax for the identification of users").
>
> If the call was recognized as having correct syntax, then it is acknowledged as having been accepted. This means that a negative acknowledge is not sent when no specific user was referred to or no user was eliminated by a call with the correct syntax.

5.2.3.44 RELOAD-DBH-SESSION

This statement reloads DBH modules and loads a DBH correction version during ongoing operation.

Scope of validity

DBH administration

See also

- DBH administration statement RECONFIGURE-DBH-SESSION on "END-FOREIGN-COPY"
- The DBH option DBH-TASKS on "DBH-TASKS"
- The DBH option SYSTEM-THREADS on "SYSTEM-THREADS"
- The DBH option THREADS on "THREADS"
- The DBH option USERS on "USERS"

Function

See section “Dynamic reconfiguration of the DBH session”.

The RELOAD-DBH-SESSION statement reloads the DBH modules.
It cannot be specified in the linked-in DBH.

This also enables you to use a new correction version for the DBH without interrupting ongoing operation. Before the statement is issued, an appropriate correction package must have been installed for the DBH using IMON. This new correction version of the DBH modules is now loaded during ongoing operation using RELOAD-DBH-SESSION. As required, the applications can be reloaded successively and they then use the new DBH correction version.

In addition, new values can be set for the DBH options DBH-TASKS, SYSTEM-THREADS, THREADS and USERS. When you enter a value for USERS which is lower than the number of user which currently exist, SESAM/SQL selects the current number of external users for as the value for USERS.

When the statement is entered with the default values (*UNCHANGED), only the DBH modules are reloaded. The values of the DBH options remain as they are.

If the DBH options cannot be changed, the DBH modules are not reloaded, but the DBH session is aborted. The DBH can be started again with the old options by means of an external restart.

```
RELOAD-DBH-SESSION

DBH-TASKS = *UNCHANGED / *STD / <integer 1..16>

,SYSTEM-THREADS = *UNCHANGED / *STD / *PARAMETERS(...)

   *PARAMETERS(...)

      |  WRITE-THREADS = *UNCHANGED / *STD / <integer 1..512>

,THREADS = *UNCHANGED / *STD / <integer 1..1024>

,USERS = *UNCHANGED / *STD / <integer 1..32767>
```

Operands

See the operand descriptions for the relevant DBH options.

When *UNCHANGED is specified, the value of the DBH option remains unchanged. When *STD is specified, the default value of the corresponding DBH option is used.

Further information

All DBH tasks with the exception of the first DBH task are stopped. The remaining DBH task starts the other DBH tasks using an ENTER command. The other DBH tasks are assigned the same job class as the first DBH task.

SESAM/SQL performs an internal comparison, so the value for THREADS is greater than or equal to double the value of the DBH-TASKS option. This can also result in implicit adjustments to the buffer settings.

In the case of RECONFIGURE-DBH-SESSION the DBH’s memory is restructured or recreated. This also causes the counters which the SESMON performance monitor evaluates for statistical purposes to be reset to “0”.

5.2.3.45 REMOVE-DISTRIBUTION-RULE-ENTRY

This statement removes a database entry from the distribution rule.

Scope of validity

SESDCN administration

Function

You use the REMOVE-DISTRIBUTION-RULE-ENTRY statement to remove either a single database entry or all database entries of a specific computer from the distribution rule.

Users who are already active when you do this can continue to work with the databases removed from the distribution rule. New users are rejected.

```
REMOVE-DISTRIBUTION-RULE-ENTRY

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers> / *PROCESSOR(...)

    *PROCESSOR(...)

        |    NAME = <name 1..8>
```

Operands

CATALOG-NAME =

Specifies which entries are to be removed from the distribution rule.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
Logical name of the database whose entry is removed from the distribution rule.

CATALOG-NAME = *PROCESSOR(...)
The entries of all databases on the specified computer are removed from the distribution rule.

NAME = <name 1..8>
Symbolic device name of the computer whose database entries are removed from the distribution rule.

Administration command in ISP format and at the CALL DML interface

Two administration commands fulfill the function of the REMOVE-DISTRIBUTION-RULE-ENTRY administration statement: DEL,CATALOG and DEL,PRO-NAME.

`DEL,CATALOG=`*`db-name`*

`DEL,PRO=`*`processor`*

5.2.3.46 REMOVE-OLD-TABLE-CATALOG-ENTRY

This statement removes an entry from the CALL DML table catalog.

Scope of validity

DBH administration

See also

- DBH start statement ADD-OLD-TABLE-CATALOG-LIST on "ADD-OLD-TABLE-CATALOG-LIST"
- DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST"
- The DBH option OLD-TABLE-CATALOG on "OLD-TABLE-CATALOG"

Function

The REMOVE-OLD-TABLE-CATALOG-ENTRY statement removes an entry from the CALL DML table catalog.

You can use this administration statement to remove either a single CALL DML table from the catalog or all entries assigned to a specific database.

A CALL DML table to be removed from the CALL DML table catalog must not be active. If it is active (i.e. currently being accessed), the DBH rejects the REMOVE-OLD-TABLE-CATALOG-ENTRY administration statement for this table.

```
REMOVE-OLD-TABLE-CATALOG-ENTRY

SELECT = *SQL(...) / *CALL-DML(...)

  *SQL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *CALL-DML(...)

    | CALL-DML-TABLE-NAME = <text 1..17>
```

SELECT =

Specifies which entries are to be removed from the CALL DML table catalog.

SELECT = *SQL(...)
Removes from the CALL DML table catalog all entries in which a specific CATALOG-NAME is specified.

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical database name as a criterion for removing the associated entries. Only those CALL DML tables that are not currently active are affected.

SELECT = *CALL-DML(...)
Removes a specific entry from the CALL DML table catalog.

CALL-DML-TABLE-NAME = <text 1..17>
Name of the CALL DML table as a criterion for removing the associated entry. If the table is active, it is not affected.

Administration command in ISP format and at the CALL DML interface

The OPT,CDBC-DEL administration command corresponds to the REMOVE-OLD-TABLE-CATALOG-ENTRY administration statement here.

```
OPT,CDBC-DEL,{C[ALL-DML-]T[ABLE]=call-dml-table-name|
C[ATALOG]=catalog-name}
```

5.2.3.47 REMOVE-SQL-DB-CATALOG-ENTRY

This statement removes an entry from the SQL database catalog.

Scope of validity

DBH administration

See also

- DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST"
- DBH option SQL-DATABASE-CATALOG on "SQL-DATABASE-CATALOG"

Function

The REMOVE-SQL-DB-CATALOG-ENTRY statement removes the entry for a database from the SQL database catalog.

The database must not be active. The REMOVE-SQL-DB-CATALOG-ENTRY administration statement is only executed when the database has a not ACTIVE status.

```
REMOVE-SQL-DB-CATALOG-ENTRY

SELECT = *LOGICAL(...) / *PHYSICAL(...)

  *LOGICAL(...)

    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

  *PHYSICAL(...)

    | PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
```

Operands

SELECT =

Specifies the database name that identifies the entry to be removed.

SELECT = *LOGICAL(...)
Causes the entry with the specified logical database name to be removed.

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical name of the database whose entry is to be removed from the SQL database catalog

SELECT = *PHYSICAL(...)
Causes the entry with the specified physical database name to be removed.

PHYS-CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Physical name of the database whose entry is to be removed from the SQL database catalog

Administration command in ISP format and at the CALL DML interface

The OPT,DBC-DEL administration command corresponds to the REMOVE-SQL-DB-CATALOG-ENTRY administration statement here.

OPT,DBC-DEL,*supplement*

supplement ::= {C[ATALOG]=*catalog-name*|P[HYSICAL-NAME]=*physical-name*}

5.2.3.48 RESUME-TRANSACTION-ADMISSION

Permits resumption of transaction admission

Scope of validity

DBH and SESDCN administration

See also

The HOLD-TRANSACTION-ADMISSION administration statement on "HOLD-TRANSACTION-ADMISSION"

Function

The RESUME-TRANSACTION-ADMISSION administration statement allows you to permit the resumption of transaction processing and admit transactions again after transaction processing has previously been suspended by means of the HOLD-TRANSACTION-ADMISSION administration statement.

The DBH does not accept this administration statement unless the HOLD-TRANSACTION-ADMISSION statement has been used previously. SESDCN accepts the statement even without the HOLD-TRANSACTION-ADMISSION statement.

```
RESUME-TRANSACTION-ADMISSION
```

Operands

This statement has no operands.

Administration command in ISP format and at the CALL DML interface

The GO-TA administration command corresponds to the RESUME-TRANSACTION-ADMISSION administration statement in DBH administration, and the GO,TA administration command corresponds to it in SESDCN administration here.

```
GO[-TA]
```

```
GO,TA
```

5.2.3.49 RESUME-USER-ADMISSION

This statement permits resumption of user admission.

Scope of validity

SESDCN administration

See also

The HOLD-USER-ADMISSION administration statement on "HOLD-USER-ADMISSION"

Function

You use the RESUME-USER-ADMISSION statement to permit the resumption of user admission.

It cancels the effect of the HOLD-USER-ADMISSION statement. Users can again be admitted during the session.

```
RESUME-USER-ADMISSION
```

Operands

This statement has no operands.

Administration command in ISP format and at the CALL DML interface

The GO,USER administration command corresponds to the RESUME-USER-ADMISSION administration statement here.

```
GO,USER
```

5.2.3.50 REUSE-OLD-TABLE-CATALOG-ENTRY

This statement creates a valid reference to a table entry in the CALL DML database catalog.

Scope of validity

DBH administration

See also

- Administration statement ADD-OLD-TABLE-CATALOG-ENTRY on "ADD-OLD-TABLE-CATALOG-ENTRY"
- Administration statement SHOW-OLD-TABLE-CATALOG-ENTRIES on "SHOW-OLD-TABLE-CATALOG-ENTRIES"

Function

With the REUSE-OLD-TABLE-CATALOG-ENTRY statement, an attempt is made to create a valid reference to a table for an entry in the CALL DML database catalog.

Before this statement is called, the entry addressed must exist in the CALL DML database catalog with the CALL DML table name; otherwise, the statement will be rejected.

The table type is not checked. The user must ensure that only references to tables of the type CALL DML only or CALL DML/SQL are made.

If the table does not exist in the database, an assignment is not possible. The entry continues to have the status “invalid”. If the database is not open at the time of the call, the entry also continues with the status “invalid”.

If you do not specify any CALL DML table name in the call, the action is performed for all entries in the CALL DML database catalog which have the status “invalid”. If no entry has the status “invalid”, the statement is ignored without an error message.

If the call is successful, the entry subsequently takes on the status “valid”. The specified table can now be accessed with CALL DML.

If the call is not successful, the table is not available and thus has the status “invalid”. Subsequent CALL DML statements are rejected with a status message.

If the table is created subsequently with the SQL statement CREATE TABLE, it is initially not yet available for CALL DML. In this case, you must issue the ADD-OLD-TABLE-CATALOG-ENTRY or REUSE-OLD-TABLE-CATALOG-ENTRY administration statement with a corresponding table entry after CREATE TABLE.

REUSE-OLD-TABLE-CATALOG-ENTRY
CALL-DML-TABLE-NAME = *ALL / <text 1..17>

Operands

CALL-DML-TABLE-NAME = *ALL / <text 1..17>

Name of the CALL DML table (maximum 17 positions). If the default value *ALL is specified, an attempt is made to create a valid reference for all table entries in the CALL DML database catalog.

Administration command in ISP format and at the CALL DML interface

The OPT,CDBC-REUSE administration command corresponds to the REUSE-OLD-TABLE-CATALOG-ENTRY administration statement here.

```
OPT,CDBC-REUSE[,C[ALL-DML-]T[ABLE]=call-dml-table-name]
```

call-dml-table-name must not contain any commas as characters because commas function as separators.

5.2.3.51 REUSE-PARTITIONS

This statement restores the logical availability of partitions in a partitioned table.

Scope of validity

DBH administration

See also

The administration statement SHOW-PARTITIONS on "SHOW-PARTITIONS"

Function

A number of accesses to partitioned tables (e.g. DML statements) require the logical availability of the partitions concerned. This means that SESAM/SQL checks the internally registered availability status, but not whether the partition is actually physically available.

You use the REUSE-PARTITIONS statement to check the physical availability of partitions in a partitioned table which are logically not available. If a partition proves to be physically available, it is marked internally as logically available and thus released for the abovementioned accesses.

> In the ongoing DBH session you can use the administration statement SHOW-PARTITIONS (see "SHOW-PARTITIONS") to be shown whether partitions are available. Further information on partitions is provided in the “ Core manual”.

REUSE-PARTITIONS performs the check on the specified partitioned table in a transaction-free status. The call can therefore take some time if transactions are running.

REUSE-PARTITIONS is used for a partitioned table only if the specified table is already physically open. A base table is opened the first time this table is accessed in a DBH session.

REUSE-PARTITIONS
CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers> ,SCHEMA-NAME = <c-string 1..31 with-lower-case> ,TABLE-NAME = <c-string 1..31 with-lower-case>

Operands

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical name of the database.

SCHEMA-NAME = <c-string 1..31 with-lower-case>
Name of the schema in the base table.

TABLE-NAME = <c-string 1..31 with-lower-case>
Name of the base table.

Administration command in ISP format and at the CALL DML interface

The PARTITION,REUSE administration command corresponds to the REUSE-PARTITIONS administration statement here.

```
PARTITION,REUSE,C[ATALOG]=name,S[CHEMA]=name,T[ABLE]=name
```

The values defined beforehand for the relevant operands may be entered for *name*.

5.2.3.52 ROLLBACK-PTC-TRANSACTION

This statement rolls back a PTC transaction.

Scope of validity

DBH administration

See also

The COMMIT-PTC-TRANSACTION administration statement on "COMMIT-PTC-TRANSACTION"

Function

The ROLLBACK-PTC-TRANSACTION statement rolls back a user's PTC transaction. A PTC transaction is a distributed transaction that has the status “prepared to commit” (see the “ Core manual”).

The functional description of the COMMIT-PTC-TRANSACTION administration statement tells you when it may be necessary to manually roll back or commit PTC transactions.

This statement is only relevant to PTC transactions. If no PTC transactions are active in the DBH, the statement is rejected. It is also rejected if the DBH was loaded without transaction management.

If a user does not have any transaction in the PTC status, then a message is output.

> ¡ You should exercise careful consideration before using the ROLLBACK-PTC-TRANSACTION administration statement; it can destroy the logical consistency of the transaction.

ROLLBACK-PTC-TRANSACTION
USER = <user-identification>

Operands

USER = <user-identification>

Identifies the user whose PTC transaction is to be rolled back. The identification must be unique. You must specify all the relevant partial qualifications (see section “Syntax for the identification of users”).

Administration command in ISP format and at the CALL DML interface

The RPTC administration command corresponds to the ROLLBACK-PTC-TRANSACTION administration statement here.

```
RPTC,user-identification
```

5.2.3.53 ROLLBACK-TRANSACTION

Rolls back a user's transaction

Scope of validity

DBH and SESDCN administration

Function

The ROLLBACK-TRANSACTION statement rolls back the transaction of a specific user.

A user's transaction can only be rolled back if it is open.

Resources requested outside the scope of transaction are not released.

You cannot use this statement to roll back PTC transactions. You require the ROLLBACK-PTC-TRANSACTION administration statement to do that.

It is not possible with the ROLLBACK-TRANSACTION statement to roll back transactions that have the status UTI (see "SHOW-TRANSACTIONS"), since these transactions are really only statements. These statements are only displayed with SHOW-TRANSACTIONS because they can hold locks. To cancel such statements, use CANCEL-STATEMENT (see "CANCEL-STATEMENT").

The ROLLBACK-TRANSACTION administration statement is rejected if the DBH was loaded without transaction management.

The statement is also rejected if the user addressed has opened a lock sequence.

ROLLBACK-TRANSACTION
USER = <user-identification>

Operands

USER = <user-identification>

Identifies the user whose transaction is to be rolled back. The identification must be unique. You must specify all the relevant partial qualifications (see section “Syntax for the identification of users”). In SESDCN administration, you must enter names for APPLICATION-NAME and CUSTOMER-NAME in the *BY-SELECT(...) operand for <user-identification>. An entry for HOST-NAME is optional. If you select the default (*NONE) for HOST-NAME, SESAM /SQL uses the name of the computer on which the SESDCN to be administered is running.

Administration command in ISP format and at the CALL DML interface

The RESET administration command corresponds to the ROLLBACK-TRANSACTION administration statement here.

```
RESET,user-identification
```

5.2.3.54 SAVE-DBH-OPTIONS

This statement saves the DBH options with their current values.

Scope of validity

DBH administration

Function

You use the SAVE-DBH-OPTIONS administration statement to save the DBH options with their values that are currently set in the DBH session to a SAM file. You can use this file as a DBH configuration file when DBH is started.
The content of the output file corresponds to the outputs in the case of SHOW-DBH-OPTIONS, SHOW-SQL-DB-CATALOG-ENTRIES and SHOW-OLD-TABLE-CATALOG-ENTRIES.
In the output file these outputs are converted to the corresponding DBH start statements SET-DBH-OPTIONS, ADD-SQL-DATABASE-CATALOG-LIST and ADD-OLD-TABLE-CATALOG-LIST. The passwords required are supplemented.

SAVE-DBH-OPTIONS
TO-FILE = *SESCONF / <filename 1..54>

Operands

TO-FILE =

Determines the file to which the DBH options and their current values are to be saved.

TO-FILE = *SESCONF
The DBH options and their current values are saved to the file which is currently assigned to the DBH as a DBH configuration file via the link name SESCONF. The DBH options entered there which were valid when the DBH started are overwritten.

TO-FILE = <filename 1..54>
Explicit specification of a file to which the DBH options and their current values are saved. If no user ID is specified, the file is created in the DBH ID. If a user ID is specified, the file must be shareable or co-ownership must have been declared there for the DBH ID. An existing file of the same name is overwritten.

Example

File output following entry of the administration statement SAVE-DBH-OPTIONS TO-FILE=<file>:

```
//SET-DBH-OPTIONS-
//    DBH-IDENTIFICATION=*PARAMETERS(-
//        CONFIGURATION-NAME=Z-
//       ,DBH-NAME=X-
//    )-
//   ,ADMINISTRATION=*PARAMETERS(-
//        ACCOUNTING=*OFF-
//       ,ADMINISTRATOR=*ANY(-
//            PASSWORD=X'E7E7E7'-
//        )-
//       ,MSG-OUTPUT=*PARAMETERS(-
//            MSG=*ALL-
```

```
//           ,OPEN-CLOSE-MSG=*NONE-
//           ,MAIL=*NONE-
//         )-
//       ,SECURITY=*PARAMETERS(-
//            PRIVILEGE-VIOLATIONS=10-
//           ,SAT-SUPPORT=*OFF-
//         )-
//    )-
//   ,CPU-RESOURCES=*PARAMETERS(-
//        DBH-TASKS=1 -
//       ,SERVICE-TASKS=*PARAMETERS(-
//            INITIAL=1 -
//           ,MAXIMUM=1 -
//           ,JOBCLASS=*STD    -
//           ,WORK-FILES=*PUBLIC-DISK(-
//                CAT-ID=*STD-
//               ,PRIMARY-ALLOCATION=*STD-
//               ,SECONDARY-ALLOCATION=*STD-
//            )-
//           ,RECORDS-PER-CYCLE=*STD-
//         )-
//    )-
.
.
```

Administration command in ISP format and at the CALL DML interface

The OPT,SAVE administration command corresponds to the SAVE-DBH-OPTIONS administration statement here.

```
OPT,SAVE,F[ILE]-N[AME]={*S[ESCONF]|<filename 1..54>}
```

5.2.3.55 SET-ACCOUNTING-PARAMETER

This statement controls the logging of request accounting.

Scope of validity

DBH administration

See also

The DBH option ACCOUNTING on "ACCOUNTING"

Function

You use the SET-ACCOUNTING-PARAMETER administration statement to control the logging of request accounting for the BS2000 RAV accounting procedure (see the BS2000 manual “ Introduction to System Administration”).

```
SET-ACCOUNTING-PARAMETER
ACCOUNTING = *ON(...) / *OFF
  *ON(...)
    | CPU-TIME = *NO / *YES
```

Operands

ACCOUNTING = *ON(...)

Turns on logging of request accounting. If logging is already on, the administration statement is rejected.

CPU-TIME = *NO / *YES
Turns CPU-time counting on/off.

ACCOUNTING = *OFF
Turns off logging of request accounting.
If logging is already off, the administration statement is rejected.

Administration command in ISP format and at the CALL DML interface

The ACC,TP administration command corresponds to the SET-ACCOUNTING-PARAMETER administration statement here.

```
ACC,TP={ON[,CPU]|OFF}
```

5.2.3.56 SET-DBH-MSG-TRACE

Controls the logging of DBH messages

Scope of validity

DBH administration

Function

You use the SET-DBH-MSG-TRACE administration statement for diagnostic purposes. It controls the logging of messages exchanged between the task of the DBH and the task of an application program (message trace).

```
SET-DBH-MSG-TRACE

TRACE = *ON(...) / *OFF(...)
  *ON(...)
    | OUTPUT = SYSLST / list-poss(2): SYSLST / SYSOUT
  *OFF(...)
    | OUTPUT = (SYSLST,SYSOUT) / list-poss(2): SYSLST / SYSOUT
```

Operands

TRACE = *ON(...) / *OFF(...)

Turns the message trace of the DBH on/off at the start of the next send request.

TRACE = *ON(...)
The message trace is turned on.

OUTPUT =
Specifies the output medium for the message trace of the DBH.

OUTPUT = SYSLST
This is the default. The message trace is logged to SYSLST.

OUTPUT = list-poss (2): SYSLST / SYSOUT
The message trace is logged only to SYSLST, only to SYSOUT, or to SYSLST and SYSOUT.

TRACE = *OFF(...)

The message trace is turned off.

OUTPUT =
Specifies the output medium for the message trace of the DBH that will be turned off.

OUTPUT = (SYSLST,SYSOUT)
This is the default. The message trace is not logged to SYSLST or SYSOUT.

OUTPUT = list-poss (2): SYSLST / SYSOUT
The message trace is logged only to SYSLST, only to SYSOUT, or to neither of them.

Administration command in ISP format and at the CALL DML interface

The TRACE administration command corresponds to the SET-DBH-MSG-TRACE administration statement here.

```
TRACE={ON|OFF}[,OUTPUT={SYSOUT|SYSLST|(SYSOUT,SYSLST)]}
```

5.2.3.57 SET-DIAGNOSIS-DUMP-PARAMETER

Controls the creation of a dump

Scope of validity

DBH administration

Function

You use the SET-DIAGNOSIS-DUMP-PARAMETER administration statement to specify for which user, and after which DBH replies, a dump file is to be created.

As soon as the execution of a statement has been completed, the DBH returns the result of processing (see the “Messages” manual). What the DBH returns depends on whether the statement is an SQL or CALL DML statement:

- after SQL statements, the DBH returns an SQLSTATE
- after CALL DML statements, the DBH returns a status.

You can use the SET-DIAGNOSIS-DUMP-PARAMETER administration statement to cause returned SQLSTATEs or statuses to be checked against an SQLSTATE or status defined in the administration statement. If the two values are the same, the DBH outputs message SES6016 and creates a dump file (see also chapter “Error handling”ff).

```
SET-DIAGNOSIS-DUMP-PARAMETER

DUMP = *ON(...) / *OFF(...) / *INIT
  *ON(...)
    | SELECT = *SQL(...) / *CALL-DML(...)
    |    *SQL(...)
    |       | SQLSTATE = <alphanum-name 5..5>
    |    *CALL-DML(...)
    |       | STATE = <alphanum-name 2..2>
    | ,USER = *ALL / <user-identification>
  *OFF(...)
    | SELECT = *SQL / *CALL-DML
```

Operands

DUMP =

Turns checking of the SQLSTATE or CALL DML status on/off.

DUMP = *ON(...)
Turns on the checking of the SQLSTATE or CALL DML status.

SELECT =
Specifies if the check of the SQLSTATE or of the CALL DML status is to be turned on.

SELECT = *SQL(...)
Turns on checking of the SQLSTATE.

SQLSTATE = <alphanum-name 5..5>
Specifies the SQLSTATE upon whose occurrence a user dump file is created.

SELECT = *CALL-DML(...)
Turns on checking of the CALL DML status.

STATE = <alphanum-name 2..2>
Specifies the CALL DML status upon whose occurrence a user dump file is created.

USER =
Specifies for which users or which group of users (for partial qualification) the checking of SQLSTATE or of the CALL DML status is to be turned on.

USER = *ALL
Turns on checking of the SQLSTATE or CALL DML status for all users.

USER = <user-identification>
Identifies one or more users for whom the checking of the SQLSTATE or of the CALL DML status is to be turned on. The syntax of the parameter is described in section “Syntax for the identification of users”. The identification entered here is valid until the administration statement is issued again.

DUMP = *OFF(...)

Turns off checking of the SQLSTATE or CALL DML status.

SELECT =
Specifies if the check of the SQLSTATE or of the CALL DML status is to be turned off.

SELECT = *SQL
Turns off checking of the SQLSTATE.

SELECT = *CALL-DML
Turns off checking of the CALL DML status.

DUMP = *INIT
Reinitializes dump control. A dump is always output when the next consistency check takes place, see section “Main-memory dump”.

Administration command in ISP format and at the CALL DML interface

The OPT,DIAG administration command corresponds to the SET-DIAGNOSIS-DUMP-PARAMETER administration command here.

```
OPT,DIAG,{diag-spec[,user-identification]|NOSTATUS|NOSQLSTATE|INITDUMP}
```

```
diag-spec ::= {STATUS=y|SQLSTATE=z}
```

5.2.3.58 SET-REQUEST-CONTROL

Turns priority control on and off

Scope of validity

DBH administration

See also

- DBH option REQUEST-CONTROL on "REQUEST-CONTROL"
- The MODIFY-REQUEST-CONTROL administration statement on "MODIFY-REQUEST-CONTROL"

Function

You use the SET-REQUEST-CONTROL administration statement to turn the priority control of the independent DBH on or off.

This administration statement is only permitted when the DBH was loaded with priority control.

If you turn priority control on with SET-REQUEST-CONTROL, the values set by means of the DBH option apply to weights and levels.

SET-REQUEST-CONTROL
REQUEST-CONTROL = *ON / *OFF

Operands

REQUEST-CONTROL = *ON

Turns on the priority control of the independent DBH. If priority control is already on, the statement is rejected.

REQUEST-CONTROL = *OFF
Turns off the priority control of the independent DBH.
If priority control is already on, the statement is rejected.

Administration command in ISP format and at the CALL DML interface

The OPT,REQ-CON administration command with specific operands corresponds to the SET-REQUEST-CONTROL administration statement.

```
OPT,REQ-CON[,ON|,OFF]
```

If you enter the OPT,REQ-CON administration command without operands, priority control is initialized with the relevant default values for weights and levels.

5.2.3.59 SET-SAT-SUPPORT

This statement controls the logging of security relevant events with SAT in the current DBH session.

Scope of validity

DBH administration

Function

Use the statement SET-SAT-SUPPORT to turn the logging of security relevant events with SAT in the current DBH session on and off.

SET-SAT-SUPPORT
SAT = *ON / *OFF

Operands

SAT = *ON / *OFF

Turns the logging of security relevant events on or off. See also “ Core manual”, section “Logging of security relevant events with SAT”.

Administration command in ISP format and at the CALL DML interface

The OPT,SAT administration command with specific operands corresponds to the SET-SAT-SUPPORT administration statement.

```
OPT,SAT=[ON|OFF]
```

5.2.3.60 SET-SESSION-DIAGNOSIS

This statement switches the deadlock analysis on and off.

Scope of validity

DBH administration

Function

The SET-SESSION-DIAGNOSIS statement controls the additional information output (deadlock analysis) when a deadlock occurs. Detailed information on the subject of deadlocks is provided in the “ Core manual”.

```
SET-SESSION-DIAGNOSIS

DIAGNOSIS = *ON(...) / *OFF(...)
  *ON(...)
    | SELECT = *DEADLOCK-ANALYSIS
  *OFF(...)
    | SELECT = *DEADLOCK-ANALYSIS
```

Operands

DIAGNOSIS =

When a deadlock occurs, defines whether further information on the deadlock should be output to SYSLST in addition to message SES3201.

DIAGNOSIS = *ON(SELECT=*DEADLOCK-ANALYSIS)
A deadlock analysis is output.

DIAGNOSIS = *OFF(SELECT=*DEADLOCK-ANALYSIS)
No further information is output.

Example

SYSLST output after SET-SESSION-DIAGNOSIS DIAGNOSIS=*ON(SELECT= *DEADLOCK-ANALYSIS) and after a deadlock has occurred:

```
******************** DEADLOCK-BEGIN ********************
USER:    D016ZE09TSN=8NLGDSQL
OBJECT:  000200020008000000000000700050401 ROW
USER:    D016ZE09TSN=8NLFDSQL
OBJECT:  00020002000B000000000000600050401 ROW
******************** DEADLOCK-END **********************
<date> <time> SES3201 DEADLOCK RESOLUTION
SESAM USER ID: D016ZE09TSN=8NLGDSQL
%//
```

Explanation of the outputs

The lines between `DEADLOCK-BEGIN` and `DEADLOCK-END` describe the cycle of the waiting users, starting from the initiating transaction.
Here one USER and one OBJECT line are written alternately. The order of the lines defines the wait relationships. A user waits for the next user; the initiating object stands between them. The last user output waits for the first user. In the example the user task with TSN 8NLG waits for the user tasks with TSN 8NLF and vice versa.

Message `SES3201` is then issued. In the example the user task was reset with TSN 8NLG.

USER output line
Contains the user's ID.
In the case of UTM users the session counter appears.

OBJECT output line The output for the object contains:

- internal information for object identification (depending on the object), e.g.:

  Distance 0: identification number of the associated database.
  This number corresponds to the order in which the databases are output in the case of SHOW-SQL-DB-CATALOG-ENTRIES, see "SHOW-SQL-DB-CATALOG-ENTRIES".

  Distance 2: identification number of the associated space (not for DBC). This corresponds to the content of the SPACE_ID column in the SYS_SPACES table in SYS_INFO_SCHEMA (not for object type DBC), see " SQL Reference Manual Part 1: SQL Statements".

  Distance 4: identification number of the table (in the case of TABLE and ROW). This corresponds to the content of the TABLE_ID column in the SYS_TABLES table in SYS_INFO_SCHEMA.

  These identification numbers are also contained in the SESCOSP reports (see section "Catalog-space table assignment table"). This table can also be used to ascertain the associated names if an access took place during logging.

  Please contact Customer Support for a detailed analysis.

- the object type

TABLE	Locks a table
ROW	Locks a record in a table
INDEX	Locks an entire index
SI-VAL	Locks a value in the index
DBC	Removes an entry from the SQL database (catalog lock)
SPACE	Locks the user space
PLAN	Planned lock
META	Locks a metadata area

Administration command in ISP format and at the CALL DML interface

Not available.

5.2.3.61 SET-SQL-DB-CATALOG-STATUS

Changes the status of a database

Scope of validity

DBH administration

See also

Administration statement SHOW-SQL-DB-CATALOG-ENTRIES on "SHOW-SQL-DB-CATALOG-ENTRIES"

Function

The SET-SQL-DB-CATALOG-STATUS statement changes the status of a database.

The possible statuses of a database and what they mean are explained in the functional description of the SHOW-SQL-DB-CATALOG-ENTRIES administration statement.

Depending on the current status of the database, i.e. its status before you issue the SET-SQL-DB-CATALOG-STATUS administration statement, the statement has different results. A distinction can be drawn between the following cases:

- When the status of the database is ACTIVE before you issue the administration statement:
  - If you specify STATUS = ACTIVE in the administration statement, there is no response. The status of the database remains ACTIVE.
  - If you specify STATUS = FREE in the administration statement, then the call is synchronized if internal or external transactions involving this database are still open. After synchronization is completed, the database is physically closed and the status of the database is set to FREE in the SQL database catalog.
- When the status of the database is FREE or LOCKED before you issue the administration statement:
  - If you specify STATUS = ACTIVE in the administration statement, the database is physically opened.
  - If you specify STATUS = FREE in the administration statement, nothing happens. The status of the database remains FREE.
- When the status of the database is CLOSED before you issue the administration statement:
  - If you specify STATUS = ACTIVE in the administration statement, there is no response. The status of the database remains CLOSED
  - If you specify STATUS = FREE in the administration statement, the database is physically closed and released.
- When the status of the database is RECOVER, REFRESH or REORG, you cannot change the status using this statement.

```
SET-SQL-DB-CATALOG-STATUS
```

```
 STATUS = *ACTIVE / *FREE
 ,SELECT = *LOGICAL(...) / *PHYSICAL(...)
    *LOGICAL(...)
       |  CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
    *PHYSICAL(...)
       |  PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
```

Operands

STATUS = *ACTIVE / *FREE

Status to which the database is to be set

SELECT =Specifies the name by means of which the entry is identified.

SELECT = *LOGICAL(...)

The entry is identified by means of the logical database name.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical name of the database whose status is to be changed.

SELECT = *PHYSICAL(...)

The entry is identified by means of the physical database name.

PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Physical name of the database whose status is to be changed

Administration command in ISP format and at the CALL DML interface

The OPT,DBC-UPD administration command corresponds to the SET-SQL-DB-CATALOG-STATUS administration statement here.

```
OPT,DBC-UPD,supplement,S[TATUS]=state
        state                 ::= {A[CTIVE] | F[REE]}
        supplement          ::= {C[ATALOG]=catalog-name|P[HYSICAL-NAME]=physical-name}
```

5.2.3.62 SET-TUNING-TRACE

Controls request logging

Scope of validity

DBH administration

Function

You use the SET-TUNING-TRACE statement to control the request logging of the DBH. You can turn request logging on or off and specify the volume for the request log file (CO-LOG) to which the DBH writes its data. You can also specify that the CPU time is to be measured.

When request logging is turned on, the DBH opens the CO-LOG file in order to log the data in it. If you turn request logging off, the CO-LOG file is closed, even when there are requests still open.

As soon as the CO-LOG file is closed, the SESCOSP utility edits and outputs the data collected (see chapter “Evaluating request logging with SESCOSP”).

Each time request logging is turned on, the file count (*iiii*) in the default filename of the CO-LOG file (SESAM.CO-LOG.*ssss.iiii*) is incremented by 0001. The placeholder *ssss* stands for the session ID, which corresponds by default to the BS2000 task sequence number, TSN (see the “ Core manual”).

In contrast to the recording of operational data with the SESMON performance monitor (see chapter “Outputting operational data with SESMON”), this type of request logging uses the resources of the DBH.

The USER-SELECT parameter allows you to restrict request logging to individual users. This makes it possible to carry out logging for extended periods without subjecting the DBH to excessive loads. When evaluating data with SESCOSP, note that the users defined in SESCOSP represent a genuine subset of the users defined in USER-SELECT. In SESCOSP the users to be evaluated can be further restricted by means of the SELECT parameter USERS (see "The SELECT parameter USERS").

SET-TUNING-TRACE

```
TRACE = *ON(...) / *OFF
  *ON(...)
    | CPU-TIME = *NO / *YES
    | ,PROTOCOL = *STD / *LONG
    | ,USER-SELECT = *ALL / <user-identification>
    | ,SUPPORT = *PUBLIC-DISK(...) / *PRIVATE-DISK(...)                    1
    |   *PUBLIC-DISK(...)
    |   |                    CAT-ID = *STD / <cat-id>
    |   |                    ,PRIMARY-ALLOCATION = 192 / <integer 1..16777215>
    |   |                    ,SECONDARY-ALLOCATION = 24 / <integer 0..32767>
    |   *PRIVATE-DISK(...)
    |   |                    CAT-ID = *STD / <cat-id>
    |   |                    ,PRIMARY-ALLOCATION = 192 / <integer 1..16777215>
```

```
|    |                    ,SECONDARY-ALLOCATION = 24 / <integer 0..32767>
|    |                    ,DEVICE-TYPE = <structured-name 1..8>
|    |                    ,VOLUME = list-poss(6): <alphanum-name 1..6>
```

[1]The *TAPE operand of older SESAM/SQL versions is no longer available.

Operands

TRACE = *ON(...)

Turns on request logging.

CPU-TIME = *NO / *YES
Specifies whether or not the CPU time is to be recorded. The default is NO, the CPU time is not recorded.

PROTOCOL = *STD / *LONG
Specifies the type of logging.

The following applies to the PROTOCOL=*STD:

- language-specific parts of the SQL statement are only logged to a limited degree; SQL descriptors are not logged.
- specific parts of CALL-DML are only logged to a limited degree (statement, query area and response area).

Unlogged parts cannot be evaluated with SESCOSP (see "Evaluating request logging with SESCOSP").

The setting PROTOCOL=*LONG enables full logging. In this case, the protocol generation loads the DBH still further and the CO-LOG file is made much larger. The setting of the PRIMARY and SECONDARY ALLOCATION must be adapted to the CO-LOG file where necessary.

USER-SELECT = *ALL / <user-identification>

Restricts the statement to a group of users. This partial qualification can have more than one match. The syntax of the parameter is described in section “Syntax for the identification of users”. The default is *ALL (no restriction applies).

> The wildcard * (asterisk) enables you to select a subset in the suboperands `HOST-NAME`, `APPLICATION-NAME` and `CUSTOMER-NAME` of the <user-identification>. The hosts, users and applications whose names begin with the specified string (up to the first *) are selected.
>
> Examples:
>
> `HOST-NAME=D016*` selects all hosts whose names begin with `D016`.
>
> `APPLICATION-NAME=UTM*` selects all applications whose names begin with `UTM`.
>
> `CUSTOMER-NAME=UPIC*` selects all users whose names begin with `UPIC`.
>
> `...-NAME=*` selects all hosts, applications or users.

SUPPORT =
Specifies the volume for the CO LOG file.

SUPPORT = *PUBLIC-DISK(...)
Causes the CO LOG file to be created on a public disk.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the file is to be stored. The default is the DBH catalog ID.

PRIMARY-ALLOCATION = 192 / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space. The specified number of PAM pages is reserved immediately.

SECONDARY-ALLOCATION = 24 / <integer 0..32767>
Number of PAM pages for subsequent extensions to files

SUPPORT = *PRIVATE-DISK(...)
Causes the CO LOG file to be created on a private disk. To do this, you must make an entry for DEVICE-TYPE.

CAT-ID = *STD / <cat-id>
Catalog ID of the pubset on which the file is to be stored.

PRIMARY-ALLOCATION = 192 / <integer 1..16777215>
Number of PAM pages for the initial allocation of memory space.
The specified number of PAM pages is reserved immediately.

SECONDARY-ALLOCATION = 24 / <integer 0..32767>
Number of PAM pages for subsequent extensions to files

DEVICE-TYPE = <structured-name 1..8>

BS2000 device type to which the required disks are assigned.If you make an incorrect entry, the system prompts you in interactive mode to make the entry again (see the device type table in the BS2000 manual “ Commands”).

VOLUME = list-poss(6): <alphanum-name 1..6>
VSNs of the disks on which the file is to be created. Up to six VSNs are permitted. You can enter each VSN only once.

TRACE = *OFF
Turns request logging off.

Administration command in ISP format and at the CALL DML interface

The COS administration command corresponds to the SET-TUNING-TRACE administration statement here.

```
COS,TP={ON[,CPU][,L[ONG]][,user-identification][,file-spec] | OFF}

    file-spec ::= V[OL]=(vsn)[,D[EV]=device]
                  [,S[PACE]=(space)][,C[ATID]=cat-id]
```

The partial qualification for *user-identification* can have more than one match.

> The wildcard * (asterisk) enables you to select a subset in the suboperands *host* , *appl* and *user* of the *user-identification* , see the description above under USER-SELECT.

In INFORM-PROGRAM administration you must adhere to the permissible length for *filespec* (see section “Administration using INFORM-PROGRAM”).

No tape device type may be specified for *file-spec*.

5.2.3.63 SET-USER-CALL-TRACE

This statement controls the logging of statements.

Scope of validity

SESDCN administration

Function

You use the SET-USER-CALL-TRACE administration statement for diagnostic purposes. It controls the logging of the call trace, i.e. of the CALL DML or SQL statements of one or more application programs and the associated SESAM/SQL responses (see also section “Diagnostics traces”).

You can turn the call trace on or off for all active users, selected users or a single user.

Statements issued in all application programs, such as SESDCN, that belong to the same configuration and in which the specified user(s) is/are active are logged.

```
SET-USER-CALL-TRACE
 TRACE = *ON(...) / *OFF(...)
   *ON(...)
     | OUTPUT = SYSLST / list-poss(2): SYSLST / SYSOUT
   *OFF(...)
     | OUTPUT = (SYSLST,SYSOUT) / list-poss(2): SYSLST / SYSOUT
 ,USER = <user-identification> / *ALL
```

Operands

TRACE = *ON(...) / *OFF(...)

Turns the call trace on or off.

TRACE = *ON(...)
Turns the call trace on.

OUTPUT =
Specifies the output medium for the call trace of the application program.

OUTPUT = SYSLST
This is the default. The call trace is logged to SYSLST.

OUTPUT = list-poss (2): SYSLST / SYSOUT
The call trace is logged only to SYSLST, only to SYSOUT or to both SYSLST and SYSOUT.

TRACE = *OFF(...)
Turns the call trace off.

OUTPUT =
Specifies the output medium for the call trace of the application program to be turned off.

OUTPUT = (SYSLST,SYSOUT)
This is the default. Logging of the call trace is turned off for SYSLST as well as for SYSOUT.

OUTPUT = list-poss (2): SYSLST / SYSOUT
Logging of the call trace is turned off for SYSLST, for SYSOUT or for both.

USER = <user-identification>
Identifies exactly one user whose call trace is to be turned on or off. The user must be identified uniquely (see the section “Syntax for the identification of users”).

USER = *ALL
Turns the call trace on or off for all active users.

Administration command in ISP format and at the CALL DML interface

The TRACE administration command in the following two formats corresponds to the SET-USER-CALL-TRACE administration statement:

Format 1:
The TRACE command sets the parameters that are to apply when logging is subsequently turned on.

```
TRACE=PAR,TYPE=CALL[,OUTPUT={SYSOUT|SYSLST|(SYSOUT,SYSLST)}]
```

Format 2:
The TRACE command turns logging on or off user specifically.

```
TRACE={ON|OFF}[,USER=[host],[application],[user][,vgid]]
```

If you do not specify the USER parameter, logging of the call trace is turned on or off for all users in the configuration.

The same TRACE command can activate the message trace (see "SET-USER-MSG-TRACE") or the call and message trace if you enter the parameter TYPE=MSG or TYPE=(CALL,MSG) instead of TYPE=CALL for the TRACE command in format 1.

Logging is always carried out based on the current values of the TYPE and OUTPUT parameters, i.e. on the values last set by means of the TRACE command in format 1.

If you do not issue a TRACE command in format 1 during the session, the defaults CALL and SYSLST apply to the TYPE and OUTPUT parameters.

5.2.3.64 SET-USER-INACTIVE-TIME

This statement specifies the maximum permissible inactivity time.

Scope of validity

DBH administration

See also

DBH option SQL-SUPPORT on "SQL-SUPPORT"

Function

The SET-USER-INACTIVE-TIME administration statement allows you to modify as required the maximum permissible inactivity time set by means of the DBH option SQL-SUPPORT.

The SET-USER-INACTIVE-TIME administration statement is only permitted in an SQL session. It is also effective when the DBH has been started without inactivity checking.

SET-USER-INACTIVE-TIME
TIME-LIMIT = <integer 1..999> / *NONE

Operands

TIME-LIMIT = <integer 1..999> / *NONE

Specifies the maximum permissible inactivity time in minutes. *NONE turns off inactivity checking, in which case the permissible inactivity time is unlimited.

Administration command in ISP format and at the CALL DML interface

The OPT,INACT administration command corresponds to the SET-USER-INACTIVE-TIME administration statement here.

```
OPT,INACT=[n|OFF]
```

5.2.3.65 SET-USER-MSG-TRACE

This statement controls the logging of SESAM/SQL messages.

Scope of validity

SESDCN administration

Function

You use the SET-USER-MSG-TRACE administration statement for diagnostic purposes. It controls the logging of SESAM/SQL messages exchanged between the task of an application program and the task of a SESAM/SQL DBH (message trace). In addition to application-program statements and the associated SESAM/SQL replies, the message trace also receives internal SESAM/SQL administration information (see also section “Diagnostics traces”).

You can turn the message trace on or off for all active users or for selected users.

Statements issued in all application programs that belong to the same configuration as SESDCN and in which the specified user(s) is/are active are logged.

```
SET-USER-MSG-TRACE

TRACE = *ON(...) / *OFF(...)
   *ON(...)
      | OUTPUT = SYSLST / list-poss(2): SYSLST / SYSOUT
   *OFF(...)
      | OUTPUT = (SYSLST,SYSOUT) / list-poss(2): SYSLST / SYSOUT
,USER = <user-identification> / *ALL
```

Operands

TRACE = *ON(...) / *OFF(...)

Turns the message trace on or off for the medium specified in OUTPUT.

TRACE = *ON(...)
The message trace is turned on.

OUTPUT =
Specifies the output medium for the message trace of the application program.

OUTPUT = SYSLST
This is the default. The message trace is logged to SYSLST.

OUTPUT = list-poss (2): SYSLST / SYSOUT
The message trace is logged only to SYSLST, only to SYSOUT, or to SYSLST and SYSOUT.

TRACE = *OFF(...)
The message trace is turned off.

OUTPUT =
Specifies the output medium for the message trace of the application program to be turned off.

OUTPUT = (SYSLST,SYSOUT)
This is the default. Logging of the message trace is turned off for SYSLST as well as for SYSOUT.

OUTPUT = list-poss (2): SYSLST / SYSOUT
Logging of the message trace is turned off for SYSLST, for SYSOUT or for both.

USER = <user-identification>
Identifies exactly one user whose message trace is to be turned on or off. The user must be identified uniquely (see the section “Syntax for the identification of users”).

USER = *ALL
Turns the message trace on or off for all active users.

Administration command in ISP format and at the CALL DML interface

The TRACE administration command in the following two formats corresponds to the SET-USER-MSG-TRACE administration statement:

Format 1:
The TRACE command sets the parameters that are to apply when logging is subsequently turned on.

```
TRACE=PAR,TYPE=MSG[,OUTPUT={SYSOUT|SYSLST|(SYSOUT,SYSLST)}]
```

Format 2:
The TRACE command turns logging on or off user specifically.

```
TRACE={ON|OFF}[,USER=[host],[application],[user][,vgid]]
```

If you do not specify the USER parameter, logging of the message trace is turned on or off for all users in the configuration.

The same TRACE command can activate the call trace (see "SET-USER-CALL-TRACE") or the call trace and message trace if you enter the parameter TYPE=CALL or TYPE=(CALL,MSG) instead of TYPE=MSG for the TRACE command in format 1.

Logging is always carried out based on the current values of the TYPE and OUTPUT parameters, i.e. on the values last set by means of the TRACE command in format 1.

If you do not issue a TRACE command in format 1 during the session, the defaults CALL and SYSLST apply to the TYPE and OUTPUT parameters.

5.2.3.66 SHOW-CALL-DML-SUBORDERS

This statement displays the number of active CALL DML requests.

Scope of validity

DBH administration

Function

The SHOW-CALL-DML-SUBORDERS administration statement displays the number of currently open logical files, i. e. the number of active CALL DML requests when the statement is issued.

This statement can apply to all users or a selected one.

If no user or CALL DML requests are active, this is indicated to you.

SHOW-CALL-DML-SUBORDERS
USER = *ALL / <user-identification>

Operands

USER =

Identifies the user or users for whom the number of active CALL DML requests is displayed.

USER = *ALL
This is the default. It displays the total number of CALL DML requests.

USER = <user-identification>
Displays information only on the user with the identification specified here. The syntax of the parameter is described in section “Syntax for the identification of users”.

The following is displayed:

- the user identification
- the user's conversation number (if the user is an asynchronous UTM application)
- the number of active CALL DML requests

Example

Output to the screen after entering the SHOW-CALL-DML-SUBORDERS administration statement:

```
%SHOW-CALL-DML-SUBORDERS USER=*ALL
%  <date> <time> SES7292 NUMBER OF LOGICAL FILES IN THE SYSTEM 9
%//
```

Administration command in ISP format and at the CALL DML interface

The USER administration command corresponds to the SHOW-CALL-DML-SUBORDERS administration statement here.

USER[,*user-identification*]

If you enter the administration command without *user-identification*, the total number of active CALL DML requests of all users is displayed.

5.2.3.67 SHOW-CATALOG-USERS

Displays the number of active users of selected databases

Scope of validity

DBH administration

Function

The SHOW-CATALOG-USERS administration statement displays the number of users accessing selected databases when you issue the statement. This means that only those users are output that have access to the database within the transaction. CALL DML users that only have one CALL DML open on a table in the database but do not yet have access within a transaction are not taken into account.

If no databases are entered in the SQL database catalog, you receive a message to this effect.

The following is displayed:

- the logical name(s) of the selected database(s)
- the status of each database (see the SHOW-SQL-DB-CATALOG-ENTRIES administration statement on "SHOW-SQL-DB-CATALOG-ENTRIES")
- the number of active SQL and CALL DML users.

```
SHOW-CATALOG-USERS

 CATALOG-NAME = *ALL / <filename 1..18 without-cat-user-gen-vers>

 ,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

     <structured-name 1..20>(...)

         |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

CATALOG-NAME = *ALL / <filename 1..18 without-cat-user-gen-vers>

Logical name of the selected databaseThe default is *ALL, which means that all databases are selected.

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Logical database name	var(*LIST).CATALOG	S	<filename 1..18>
Database status	var(*LIST).STATE	S	ACTIVE / FREE / LOCKED / REORG / RECOVER / REFRESH

Number of active users	var(*LIST).USER	S	<integer>

Example

Output to the screen after entering the SHOW-CATALOG-USERS administration statement:

```
%SHOW-CATALOG-USERS
CATALOG  DCCAT                STATE ACTIVE         0 USERS
CATALOG  DB6LOKG              STATE ACTIVE         7 USERS
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The USER,CATALOG administration command corresponds to the SHOW-CATALOG-USERS administration statement here.

USER,C[ATALOG][=*catalog-name*]

5.2.3.68 SHOW-CATID-LIST

This statement displays the current CATID list.

Scope of validity

DBH administration

Function

The administration statement SHOW-CATID-LIST outputs the current CATID list.

The following is displayed:

- file name of the CATID list used last
- the current CATIDs.

```
SHOW-CATID-LIST

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)
    <structured-name 1..20>(...)
        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
catalog identifier	var(*LIST).CATID	S	<cat-id 1..4>

Example

Output to the screen after entering the SHOW-CATID-LIST administration statement:

```
%SHOW-CATID-LIST
FILE :5OC1:$SQLDB112.WIE.CATIDS C
------------------------------------------------------------------
5OC1 ! 5OC2 ! 5OC3 ! 5OC4 ! 5OC5 ! 50C6 ! 50C7 ! 50C8 ! 50C9 ! 50CA
   S !    T !  AB3 !   C4 !    X ! 1OSQ ! 3OSQ ! 4OSQ !      !
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The INF,C administration command corresponds to the SHOW-CATID-LIST administration statement here.

```
INF,C[id-list]
```

5.2.3.69 SHOW-DBH-MEDIA-CATALOG

Displays the DBH option MEDIA-CATALOG

Scope of validity

DBH administration

See also

DBH option MEDIA-CATALOG on "MEDIA-CATALOG"

Function

The SHOW-DBH-MEDIA-CATALOG statement displays the currently valid entries of the DBH option MEDIA-CATALOG.

```
SHOW-DBH-MEDIA-CATALOG
```

Operands

This statement has no operands.

Example

Output to the screen after entering the SHOW-DBH-MEDIA-CATALOG administration statement:

```
%SHOW-DBH-MEDIA-CATALOG
Media catalog output
------------------------------------------------------------------------------
TALOG-SUPPORT:           SUPPORT=*PUBLIC-DISK
                         CAT-ID=*STD
                         PRIMARY-ALLOCATION=*STD    ,SECONDARY-ALLOCATION=*STD
WALOG-SUPPORT:           SUPPORT=*PUBLIC-DISK
                         CAT-ID=*STD
                         PRIMARY-ALLOCATION=*STD    ,SECONDARY-ALLOCATION=*STD
CURSOR-MEDIA: SUPPORT-1: SUPPORT=*PUBLIC-DISK
                         CAT-ID=*STD
                         PRIMARY-ALLOCATION=*STD    ,SECONDARY-ALLOCATION=*STD
CURSOR-MEDIA: SUPPORT-2: SUPPORT=*PUBLIC-DISK
                         CAT-ID=*STD
                         PRIMARY-ALLOCATION=*STD    ,SECONDARY-ALLOCATION=*STD
.
.
CURSOR-MEDIA: SUPPORT-5: SUPPORT=*PUBLIC-DISK
                         CAT-ID=*STD
                         PRIMARY-ALLOCATION=*STD    ,SECONDARY-ALLOCATION=*STD
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The OPT,MEDIA administration command corresponds to the SHOW-DBH-MEDIA-CATALOG administration statement here.

```
OPT,MEDIA
```

5.2.3.70 SHOW-DBH-OPTIONS

Displays the DBH options.

Scope of validity

DBH administration

Function

The SHOW-DBH-OPTIONS statement displays the currently valid entries of all DBH options except MEDIA-CATALOG.

You can display entries of the DBH option MEDIA-CATALOG separately by means of the SHOW-DBH-MEDIA-CATALOG administration statement.

```
SHOW-DBH-OPTIONS
```

Operands

This statement has no operands.

Example

Output to the screen (several screens) after entering the SHOW-DBH-OPTIONS administration statement:

```
%SHOW-DBH-OPTIONS
SESAM-DBH options:
------------------------------------------------------------------------------
ACCOUNTING             *OFF
ADMINISTRATOR          *ANY
COLUMNS                256
CURSOR-BUFFER          BUFFER-SIZE=512    , FRAME-SIZE=4
DBH-IDENTIFICATION     CONFIGURATION-NAME=Z, DBH-NAME=X
DBH-TASKS              1
LINKED-IN-ATTRIBUTES   CODED-CHARACTER-SET=*NONE
MSG-OUTPUT             MSG           =*SYSOUT, *SYSLST
                       OPEN-CLOSE-MSG=*NONE
                       MAIL: COLLECT=*YES,  IMMEDIATE=*YES
OLD-TABLE-CATALOG      5
RECOVER-OPTIONS        SYSTEM-DATA-BUFFER=1500   , USER-DATA-BUFFER=1000
                       TALOG-SUPPORT=*PUBLIC-DISK
                       CAT-ID=*STD
                       PRIMARY-ALLOCATION=*STD    , SECONDARY-ALLOCATION=*STD
                       WALOG-SUPPORT=*PUBLIC-DISK
. . .
USERS                  24
USER-DATA-BUFFER       1536
WORK-CONTAINER         INITIAL=120    , MAXIMUM=120
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The OPT administration command corresponds to the SHOW-DBH-OPTIONS administration statement here.

OPT

5.2.3.71 SHOW-DISTRIBUTION-RULE-ENTRIES

This statement displays the active databases in the distribution rule.

Scope of validity

SESDCN administration

Function

The SHOW-DISTRIBUTION-RULE-ENTRIES statement displays all the databases entered in the distribution rule.

The following appears for each database:

- the logical database name (CATALOG)
- the number of users (USER)
- the host name (PRO-NAME)
- DBH name
- DCN name
- the configuration name (CNF-NAME).

```
SHOW-DISTRIBUTION-RULE-ENTRIES

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)
    <structured-name 1..20>(...)
        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output information	Name of the S variable	T	Contents
Logical database name	var(*LIST).CATALOG	S	<filename 1..18>
Number of users	var(*LIST).USER	S	<integer>
Host name	var(*LIST).HOST	S	<name 1..8>
DBH name	var(*LIST).DBH-NAME	S	<alphanum-name 1..1> / "BLANK"
DCN name	var(*LIST).DCN-NAME	S	<alphanum-name 1..1> / "BLANK"
Configuration name	var(*LIST).CNF-NAME	S	<alphanum-name 1..1> / "BLANK"

Example

Output to the screen after entering the SHOW-DISTRIBUTION-RULE-ENTRIES administration statement:

```
%SHOW-DISTRIBUTION-RULE-ENTRIES
    CATALOG          !  USER ! PRO-NAME !DBH-NAME!DCN-NAME!CNF-NAME!
                     !       !          !        !        !        !
-------------------------------------------------------------------------------
  DB6LOKG            !     7 ! D016ZE09 !   6    !   A    !   G    !
  DB7LOKG            !     0 ! D016ZE09 !   7    !   A    !   G    !
  DB0LOKG            !     0 ! D016ZE09 !        !   A    !   G    !
  DCCAT              !     0 ! D016ZE09 !   7    !   A    !   G    !
  DCCAT8             !     0 ! D016ZE09 !   8    !   A    !   G    !
  DB6REMG            !     0 ! D016ZE07 !   6    !   A    !   G    !
  DB7REMG            !     0 ! D016ZE07 !   7    !   A    !   G    !
  DB8REMG            !     0 ! D016ZE07 !   8    !   A    !   G    !
  DB6LOKH            !     0 ! D016ZE09 !   6    !   A    !   H    !
  DB7LOKH            !     0 ! D016ZE09 !   7    !   A    !   H    !
  DB8LOKH            !     0 ! D016ZE09 !   8    !   A    !   H    !
  DB6LOKI            !     0 ! D016ZE09 !   6    !   A    !   I    !
  DB7LOKI            !     0 ! D016ZE09 !   7    !   A    !   I    !
  DB8LOKI            !     0 ! D016ZE09 !   8    !   A    !   I    !
% <date> <time> SEN2014 END OF THE 'SESDCN' ADMINISTRATION STATEMENT
%//
```

Administration command in ISP format and at the CALL DML interface

The USER,CATALOG administration command corresponds to the SHOW-DISTRIBUTION-RULE-ENTRIES administration statement here.

```
USER,CATALOG
```

5.2.3.72 SHOW-INACTIVE-SQL-USERS

This statement displays inactive SQL users.

Scope of validity

DBH administration

Function

The SHOW-INACTIVE-SQL-USERS administration statement displays all currently inactive SQL users. If there is no user in the system, a message will appear stating this fact.

The following appears:

- the user identification
- the user's conversation number (if the user is an asynchronous UTM application)
- the inactivity time in seconds (INACT-TIME).
  The inactivity time is the time in which an SQL conversation has not opened a transaction.

```
SHOW-INACTIVE-SQL-USERS

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

    <structured-name 1..20>(...)

        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output information	Name of the S variable	T	Contents
Host name	var(*LIST).HOST	S	<name 1..8>
Application	var(*LIST).APPLICATION	S	<name 1..8>
User	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number	var(*LIST).CONVERSATION	S	<x-text 1..8> / (void)
Inactivity time (in seconds)	var(*LIST).INACT-TIME	S	<integer>

Example

Output to the screen after entering the SHOW-INACTIVE-SQL-USERS administration statement:

```
%SHOW-INACTIVE-SQL-USERS
USER-ID: D016ZE09TSN=2BBZUTMTOOL           INACT-TIME:           44 (SEC.)
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The USER,INACT administration command corresponds to the SHOW-INACTIVE-SQL-USERS administration statement here.

```
USER,INACT
```

5.2.3.73 SHOW-MAIL-PARAMETERS

Outputs parameters for the email output of the DBH

Scope of validity

DBH administration

Function

The administration statement SHOW-MAIL-PARAMETERS outputs the currently valid parameters for the email output of the DBH and the name of the MAIL parameter file used (see "Sending important information of the DBH session by email").

```
SHOW-MAIL-PARAMETERS

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)
    <structured-name 1..20>(...)
        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Message number	var(*LIST).MESSAGE-NR	S	<text 1..7>

Example

Output to the screen after entering the SHOW-MAIL-PARAMETERS administration statement:

```
%SHOW-MAIL-PARAMETERS
FILE:    :2OSG:$TEST.SESCONF.MAIL
ADDRESS: MY.NAME@HOME.DE
SENDER:  MY.NAME@HOME.DE
------------------------------------------------------------------------------
 SES60** !         !         !         !         !         !         !
%  <date> <time> SES7215 END OF OUTPUT
%//
```

-

5.2.3.74 SHOW-OLD-TABLE-CATALOG-ENTRIES

Displays the entries in the CALL DML table catalog

Scope of validity

DBH administration

See also

- DBH start statement ADD-OLD-TABLE-CATALOG-LIST on "ADD-OLD-TABLE-CATALOG-LIST"
- Administration statement REUSE-OLD-TABLE-CATALOG-ENTRY on "REUSE-OLD-TABLE-CATALOG-ENTRY"

Function

The SHOW-OLD-TABLE-CATALOG-ENTRIES statement displays the entries in the CALL DML table catalog. The following appears for each entry:

- the name of a CALL DML table (call dml table)
- the logical name of the database to which the CALL DML table is assigned (catalog name)
- the name of the schema to which the CALL DML table is assigned (schema name)
- the name of the base table to which the CALL DML table is assigned (table name)
- the status of the entry:

valid:

When analyzing the entry, an assignment to a CALL DML table was detected;

this assignment still exists.

invalid:
When analyzing the entry, either no assignment to an existing table could be found or this assignment has been lost again. This situation occurs in the following cases:

- The database has either been closed again in the meantime (by means of the SET-CATALOG-STATUS administration statement, for example) or was not open when the entry was analyzed.
- The table to which the assignment is to be made is not known (or no longer known) in the database (e.g. because DROP TABLE was issued for the
table, for example).

SHOW-OLD-TABLE-CATALOG-ENTRIES

```
SELECT = *ALL / *SQL(...) / *CALL-DML(...)
  *SQL(...)
    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
  *CALL-DML(...)
    | CALL-DML-TABLE-NAME = <text 1..17>
,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)
  <structured-name 1..20>(...)
```

| WRITE-MODE = *REPLACE / *EXTEND

Operands

SELECT =

Specifies which entries in the CALL DML table catalog are to be displayed.

SELECT = *ALL
Specifies that all entries in the CALL DML table catalog are to be displayed.

SELECT = *SQL(...)
Specifies that those entries containing a specific CATALOG-NAME are to be displayed.

CATALOG-NAME =
<filename 1..18 without-cat-user-gen-vers>
Logical database name that functions as a criterion for displaying the associated entries

SELECT = *CALL-DML(...)
Specifies that only the entry containing the specified CALL-DML-TABLE-NAME is to be displayed.

CALL-DML-TABLE-NAME = <text 1..17>
Name of the CALL DML table whose entry is displayed

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Name of the CALL DML table	var(*LIST).CALL-DML-TABLE	S	<text 1..17>
Logical database name	var(*LIST).CATALOG	S	<filename 1..18>
Schema name	var(*LIST).SCHEMA	S	<text 1..31>
Name of the base table	var(*LIST).TABLE	S	<text 1..31>
Status of the entry	var(*LIST).STATE	S	VALID / INVALID

Example

Output to the screen after entering the SHOW-OLD-TABLE-CATALOG-ENTRIES administration statement:

```
%SHOW-OLD-TABLE-CATALOG-ENTRIES
<call-dml-table> |<catalog-name>     |<schema-name> / <table-name>    |<status>
-------------------------------------------------------------------------------
DB6LOKG          |DB6LOKG            |SCHEMA112                       | valid
                 |                   |TAB112                          |
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The OPT,CDBC-INFO administration command corresponds to the SHOW-OLD-TABLE-CATALOG-ENTRIES administration statement here.

```
OPT,CDBC-INFO[,supplement]

supplement :: = {C[ATALOG]=catalog-name|
                 C[ALL-DML-]T[ABLE]=call-dml-table-name}
```

If you do not specify the *supplement* parameter, the whole CALL DML table catalog is displayed.

5.2.3.75 SHOW-PARTITIONS

This statement displays information on the logical availability of partitions in a partitioned table.

Scope of validity

DBH administration

See also

The administration statement REUSE-PARTITIONS on "REUSE-PARTITIONS"

Function

You use the SHOW-PARTITIONS statement to display the availability status of all partitions in an open partitioned table.

For each configured partition, one information line is displayed (in ascending order of the partition numbers) which contains the following:

- Number of the partition
- Space name
- Status:
  - AVAILABLE (partition is logically available)
  - NOT AVAILABLE (partition is not logically available)
    This status is also displayed if access to the partition is only possible conditionally, e.g. if the space is in the “copy pending” status.
- Additional information:
  Here, for example, the SQL state or the DMS error code is displayed if the partition is unavailable for reasons which are known.

> i You can restore the logical availability of partitions in the ongoing DBH session with the administration statement REUSE-PARTITIONS (see "REUSE-PARTITIONS"). Further information on partitions is provided in the “ Core manual”.

```
SHOW-PARTITIONS

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

,SCHEMA-NAME = <c-string 1..31 with-lower-case>

,TABLE-NAME = <c-string 1..31 with-lower-case>

,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

    <structured-name 1..20>(...)

        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Logical name of the database.

SCHEMA-NAME = <c-string 1..31 with-lower-case>
Name of the schema in the base table.

TABLE-NAME = <c-string 1..31 with-lower-case>
Name of the base table.

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Number of the partition	var(*LIST).PARTITION	S	<integer 1..16>
Space name	var(*LIST).SPACE	S	<filename 1..18>
Status of the partition	var(*LIST).STATE	S	AVAILABLE / NOT AVAILABLE
Additional information	var(*LIST).INFORMATION	S	SQL-STATE: *xxxxx* / DMS-ERROR: *xxxx* / CC-NUMBER: *xx/xx/xx* / '' (void)

Example

Output to the screen after entering the SHOW-PARTITIONS administration statement:

```
%SHOW-PARTITIONS CAT=CAT-1,SCHEMA='SCH-1',TABLE='TAB-1'
------------------------------------------------------------------------
 NUMBER ! SPACE-NAME         ! STATUS        ! INFO
------------------------------------------------------------------------
      2 ! TABLESPACE         ! AVAILABLE     !
      3 ! TABLESP002         ! AVAILABLE     !
      9 ! TABLESP003         ! NOT AVAILABLE ! SQL-State: 81SA6
%  <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The PARTITION,INF administration command corresponds to the SHOW-PARTITIONS administration statement here.

```
PARTITION,INF,C[ATALOG]=name,S[CHEMA]=name,T[ABLE]=name
```

The values defined beforehand for the relevant operands may be entered for *name*.

5.2.3.76 SHOW-SPACE-USERS

This statement displays the users currently working in a space.

Scope of validity

DBH administration

Function

The SHOW-SPACE-USERS administration statement displays all users currently working in the selected space of a specific database. A user works in a space when using tables or indexes created in this space. In addition, system-generated identifications of the internal administrator (see "Syntax for the identification of users") are output if transactions are executed for the internal administrator in the space.

Only those users are output that have access within a transaction to the space. CALL DML users that only have one CALL DML open on a table of the space but do not have access within the transaction are not taken into account.

The following information is displayed:

- the user identification
- the user's conversation number (if the user is an asynchronous UTM application)
- the program runtime in seconds (DURATION-TIME)
- the access mode (ACCESS) of the space (READ, WRITE or ADMIN)
- the state of the space (“copy”, “check pending” or “defect”). You will find information on the different space states in the “ Core manual”.

```
SHOW-SPACE-USERS

SPACE-NAME = <filename 1..18 without-cat-user-gen-vers>

,CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

    <structured-name 1..20>(...)

        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

SPACE-NAME = <filename 1..18 without-cat-user-gen-vers>

Name of the space whose users are displayed. The name of the space is the name of the space file in BS2000.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
Logical name of the selected database

> If the space cannot be identified based on the two names, a message to this effect is output.

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Host name	var(*LIST).HOST	S	<name 1..8>
Application	var(*LIST).APPLICATION	S	<name 1..8>
User	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number	var(*LIST).CONVERSATION	S	<x-text 1..8> / '' (void)
Program runtime (in seconds)	var(*LIST).DURATION-TIME	S	<integer>
Access mode	var(*LIST).ACCESS-MODE	S	WRITE / READ / ADMIN

Example

Output to the screen after entering the SHOW-SPACE-USERS SPACE-NAME=TABLESPACE,CATALOG-NAME=CAT-1 administration statement:

```
SPACE  TABLESPACE
------------------------------------------------------------------------
  USER-IDENTIFICATION               !DURATION-TIME! ACCESS
------------------------------------------------------------------------
 D016ZE07TSN=12LDSESUTI             !         20  ! READ
%  <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The USER,SPACE administration command corresponds to the SHOW-SPACE-USERS administration statement here.

USER,SPACE=*space-name*,C[ATALOG]=*catalog-name*

5.2.3.77 SHOW-SQL-DB-CATALOG-ENTRIES

This statement displays entries from the SQL database catalog.

Scope of validity

DBH administration

See also

DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST"

Function

The SHOW-SQL-DB-CATALOG-ENTRIES statement allows you to display entries in the SQL database catalog. The following appears for each entry:

- the logical database name
- the physical database name
- in addition, the version number of the SESAM backup copy of the catalog space, if the database is a SESAM backup copy of a database
- the access mode (Class) permitted for this database (READ, WRITE, ADMIN and REPLICATION)
- the DB user ID of the database (USERID)
- the status of the database (Status) with additional information (Info)

  The following statuses are possible:
  - STATUS = ACTIVE, which means the database is open during the current DBH session
  - STATUS = CLOSED, which means the database is closed after PREPARE-FOREIGN-COPY with physical closure of the database files
  - STATUS = FREE, which means the database is physically closed and unlocked
  - STATUS = LOCKED, which means the database is not available during the current session because of a lock. You receive additional information with this status as to why the database is not available. If there is a DMS error, you receive the DMS error key.
    If the problem (e.g. DMS error) has been eliminated, you can use the SET-SQL-DB-CATALOG-STATUS administration statement to set the status of the database to ACTIVE.
  - STATUS = RECOVER, i.e. the database is currently in a state of recovery. The status of the database changes automatically to ACTIVE at the end of the recovery if the recovery was successful, or it changes to LOCKED if it could not be completed successfully.
  - STATUS = REORG, i.e. the database is currently in a state of reorganization. The status of the database changes automatically to ACTIVE at the end of the reorganization if the reorganization was successful, or it changes to LOCKED if it could not be completed successfully.
  - STATUS = REFRESH, i.e. the database is currently in a state of refresh. The status of the database changes automatically to ACTIVE at the end of the refresh if the refresh was successful, or it changes to LOCKED if it could not be completed successfully.
- A timestamp in the format `YYYY-MM-dd hh:mm:ss` which specifies when the status of the database was determined.

SHOW-SQL-DB-CATALOG-ENTRIES
SELECT = *ALL / *LOGICAL(...) / *PHYSICAL(...)

```
  *LOGICAL(...)
    | CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
  *PHYSICAL(...)
    | PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)
  <structured-name 1..20>(...)
    | WRITE-MODE = *REPLACE / *EXTEND
```

Operands

SELECT =

Specifies which entries from the SQL database catalog are to be displayed.

SELECT = *ALL
Specifies that all entries from the SQL database catalog are to be displayed.

SELECT = *LOGICAL(...)
Specifies that those entries containing the specified CATALOG-NAME are to be displayed.

CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>
Logical name of the database whose entry is displayed. If the database is a SESAM backup copy, the number of the SESAM backup copy of the catalog space is also output on the output medium.

SELECT = *PHYSICAL(...)

Specifies that those entries containing the specified PHYS-CATALOG-NAME are to be displayed.

PHYS-CATALOG-NAME = <filename 1..18 without-cat-user-gen-vers>

Physical name of the database whose entry is displayed. If the database is a SESAM backup copy, the number of the SESAM backup copy of the catalog space is also output on the output medium.

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Logical database name	var(*LIST).CATALOG	S	<filename 1..18>
Physical database name	var(*LIST).PHYSICAL-NAME	S	<filename 1..18>
Version number of the SESAM backup copy	var(*LIST).COPY-NUMBER	S	<integer 1..999999>
Access mode	var(*LIST).ACCESS-MODE	S	WRITE / READ / ADMIN / REPL
DB user ID of the database	var(*LIST).USERID	S	<name 1..8>

Database status	var(*LIST).STATE	S	ACTIVE / FREE / LOCKED / REORG / RECOVER / REFRESH / CLOSED
Additional information	var(*LIST).INFORMATION	S	SQL-STATE: *xxxxx* / DMS-ERROR: *xxxx* / CC-NUMBER: *xx/xx/xx* / '' (void)
Timestamp	var(*LIST).TIMESTAMP	S	`yyyy-mm-dd hh:mm:ss`

Example

Output to the screen after entering the SHOW-SQL-DB-CATALOG-ENTRIES administration statement:

```
%SHOW-SQL-DB-CATALOG-ENTRIES
<catalog-name>       | Copy-|Class| UserID   | Status  | Info
  <physical-name>    |  Nr  |     |          |         |
-------------------------------------------------------------------------------
DCCAT                |      |ADMIN| SQLDB112 | ACTIVE  |
  DB112.G6           |      |     |          |         | <date> <time>
DB6LOKG              |      |ADMIN| SQLDB112 | ACTIVE  |
  CATG6              |      |     |          |         | <date> <time>
% <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The OPT,DBC-INFO administration command corresponds to the SHOW-SQL-DB-CATALOG-ENTRIES administration statement here.

`OPT,DBC-INFO[,`*`supplement`*`]`

`supplement` `::= {C[ATALOG]=`*`catalog-name`*`|P[HYSICAL-NAME]=`*`physical-name`*`}`

If you do not specify the *supplement* parameter, the whole SQL database catalog is displayed.

5.2.3.78 SHOW-TRANSACTIONS

This statement displays open transactions.

Scope of validity

DBH and SESDCN administration

Function

The SHOW-TRANSACTIONS administration statement displays a table of all open transactions. In SESDCN administration, you can use the statement to display the open transactions of selected users.

This statement is rejected if there are currently no open transactions active in the DBH or known to SESDCN. In this case you receive a message to this effect. The statement is also rejected if the DBH was loaded without transaction management.

What is displayed in DBH administration differs from what is displayed in SESDCN administration.

In DBH administration, the following information appears for each open transaction:

- The user identification
  If the user is an asynchronous UTM application, the conversation number is also displayed.
- The number of executed I/Os (NUMBER OF I/OS)
- The runtime in seconds since the beginning of the transaction (DURATION TIME)
- The status of the transaction (STATUS)
  The following transaction statuses are possible:
  - 'BLANK' (the request is being processed in a thread)
  - LOCK (the transaction is locked by another transaction)
  - FREE (the request is unlocked)
  - PTC (the transaction is at the prepare-to-commit stage)
  - ADM (the user is the system administrator)
  - RESET (the transaction is being rolled back)
  - CAN (the SQL data-manipulation statement is canceled).
  - UTI (utility statement)
  - INTERN (internal system transaction)
- The request identification of the locking transaction if the status of the transaction is LOCK (user-identification BY LOCK).
  If the user is an asynchronous UTM application, the conversation number is also displayed.
- In addition, the system-generated identifications of the internal administrator (see "Identification of system-internal users") are output if the internal administrator has locked transactions that are to be output.
- Utility statements are output here, as well, although they do not represent transactions that are visible from the outside. The utility statements are displayed nevertheless because they can hold locks.

In SESDCN administration, the following information is displayed for each open transaction:

- The user identification
  If the user is an asynchronous UTM application, the conversation number is also displayed.
- the runtime (in seconds:minutes) since the beginning of the transaction (RUN-TIME).

- the activity level of the transaction as a measure of the work completed by the transaction (ACTIVITY LEVEL). The activity level is the sum of all physical I/Os that a transaction has initiated. However, a correct value is only obtained when logging of request accounting is activated for all the DBHs involved by means of the DBH option ACCOUNTING (see "ACCOUNTING") or the administration statement SET-ACCOUNTING-PARAMETER (see "SET-ACCOUNTING-PARAMETER").
- the name of the computer on which the transaction is or was active (LOCATION). If the transaction has not yet been active on any computer, “NO” appears here.
- The status of the transaction (STATUS)
  The following transaction statuses are possible:
  - R (READ)
  - W (WRITE)
  - R/W (READ and WRITE)
  - BTA (beginning of the transaction)
  - PTC (the transaction is in the state “prepared to commit”)
  - LOCK (the transaction is locked by another transaction)
  - UNLC (the transaction is not locked).
- the lock time (in hours:minutes) if the state of the transaction is LOCK (LOCK-TIME).

```
SHOW-TRANSACTIONS

USER = *ALL / <user-identification>

,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

    <structured-name 1..20>(...)

        |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

USER =

Identifies the user or users whose transactions are to be displayed.

USER = *ALL
This is the default. All open transactions are displayed. In DBH administration, only the default is permitted.

USER = <user-identification>
Only the open transactions of the user with the specified identification are displayed. The syntax of the parameter is described in section “Syntax for the identification of users”.
This parameter is permitted only in SESDCN administration.

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables (DBH administration)

Output information	Name of the S variable	T	Contents

Host name (current user)	var(*LIST).HOST	S	<name 1..8>
Application (current user)	var(*LIST).APPLICATION	S	<name 1..8>
User (current user)	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number (current user)	var(*LIST).CONVERSATION	S	<x-text 1..8> / '' (void)
Number of inputs/outputs	var(*LIST).IO	S	<integer>
Runtime (in seconds)	var(*LIST).DURATION-TIME	S	<integer>
User's state	var(*LIST).STATE	S	ADM / LOCK / RESET / INTERN / CAN / PTC / FREE / UTI / '' (void)
Host name (locking transaction)	var(*LIST).LOCK-HOST	S	<x-text 1..8> / '' (void)
Application (locking transaction)	var(*LIST). LOCK-APPLICATION	S	<x-text 1..8> / '' (void)
User (locking transaction)	var(*LIST).LOCK-CUSTOMER	S	<x-text 1..8> / '' (void)
Conversation number (locking transaction)	var(*LIST). LOCK-CONVERSATION	S	<x-text 1..8> / '' (void)

Output in S variables (DBH administration)

Output information	Name of the S variable	T	Contents
Host name (current user)	var(*LIST).HOST	S	<name 1..8>
Application (current user)	var(*LIST).APPLICATION	S	<name 1..8>
User (current user)	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number (current user)	var(*LIST).CONVERSATION	S	<x-text 1..8> / 0
User ID (BS2000), job name (DCAM), program unit (UTM)	var(*LIST).USER-NAME	S	<name 1..8>
Runtime (hh:mm)	var(*LIST).RUN-TIME	S	<time 5 .. 5>
Activity level	var(*LIST).ACTIVITY-LEVEL	S	<integer>
Location	var(*LIST).LOCATION(*LIST). HOST	S	NO / <name 1..8>

State of the transaction	var(*LIST).LOCATION(*LIST). TA-STATE	S	BTA / PTC / R / W / R/W
Lock state	var(*LIST).LOCATION(*LIST). LOCK-STATE	S	LOCK / UNLC
Lock time (hh:mm)	var(*LIST).LOCATION(*LIST). LOCK-TIME	S	<time 5..5> / '' (void)

Example 1

Output to the screen after entering the SHOW-TRANSACTIONS administration statement in DBH administration:

```
%SHOW-TRANSACTIONS
 USER-IDENTIFICATION ! NUMBER  ! DURATION! STATUS ! USER-IDENTIFICATION
 ACTIVE              ! OF I/OS ! TIME    !        ! BY LOCK
-------------------------------------------------------------------------
 D016ZE07TSN=8GPZ    !        0!        0! ADM    !
 ADMINIST            !         !         !        !
 D016ZE09TSN=2A2V    !        0!     2053!        !
 USER0005            !         !         !        !
 D016ZE09TSN=2A2V    !        0!     2035!        !
 USER0006            !         !         !        !
 D016ZE09TSN=2A2V    !       12!     2123!        !
 USER0001            !         !         !        !
 D016ZE09TSN=2A2V    !        0!     2106!        !
 USER0002            !         !         !        !
 D016ZE09TSN=2A2V    !        0!     2085!        !
 USER0003            !         !         !        !
 D016ZE09TSN=2A2V    !        0!     2070!        !
 USER0004            !         !         !        !
% <date> <time> SES7215 END OF OUTPUT
%//
```

Example 2

Output to the screen after entering the SHOW-TRANSACTIONS administration statement in SESDCN administration:

```
%SHOW-TRANSACTIONS
          USER                          ! RUN-!ACTIVITY!LOCATION!  STATE ! LOCK!
                                         TIME!  LEVEL !        !        ! TIME
-------------------------------------------------------------------------------
D016ZE09TSN=7QLXSEDI63  00000000SQLDB112!00:04!  00000 !D016ZE08!BTA UNLC!
                                        !        !D016ZE09!BTA UNLC!
D016ZE09TSN=7QLZDSQL    00000000SQLDB112!00:02!  00000 !D016ZE08! R  UNLC!
                                        !     !        !D016ZE09! R  UNLC!
D016ZE09TSN=7R12USER000100000000SQLDB112!00:38!  00000 !D016ZE08! R  UNLC!
                                        !     !        !D016ZE09!BTA UNLC!
D016ZE09TSN=7R12USER000200000000SQLDB112!00:36!  00000 !D016ZE08!BTA UNLC!
                                        !     !        !D016ZE09! R  UNLC!
%  <date> <time> SEN2014 END OF THE 'SESDCN' ADMINISTRATION STATEMENT
%//
```

When a user is active at multiple locations, one output line is generated per location (Location). The first output line then contains all the information, and the following lines only the “Location”, “State” and “Lock Time” data.

Administration command in ISP format and at the CALL DML interface

The USER,TA administration command corresponds to the SHOW-TRANSACTIONS administration statement here.

```
USER,TA
```

5.2.3.79 SHOW-USERS

This statement displays active users.

Scope of validity

DBH and SESDCN administration

Function

The SHOW-USERS administration statement displays all currently active users. Active users are those currently working with the DBH or with SESDCN.

If no users are currently active, you receive a message to this effect.

What is displayed in DBH administration differs from what is displayed in SESDCN administration.

In DBH administration, the following information is displayed for each active user:

- user identification
  If the user is an asynchronous UTM application, the conversation number is also displayed. System-generated identifications may also be displayed here, after a restart, for example (see "Identification of system-internal users"). If there is a lock sequence active for the system-internal user, this user is displayed here.
- number of password violations, if any
- user's activity status
  The following activity statuses are possible:
  - 'BLANK' (the request is being processed in a thread)
  - LOCK (the request is locked by another request)
  - FREE (the request is unlocked)
  - PTC (the transaction is at the prepare-to-commit stage)
  - ADM (the user is the system administrator)
  - ADM (the user is the system administrator)
  - CAN (the SQL data-manipulation statement is canceled by means of the CANCEL-STATEMENT administration statement)
  - UTI (utility monitor request)
  - INTERN (internal system transaction)
- string LOCK-SEQ if the user has opened a lock sequence
- number of logical files in CALL DML, if any are used
- number of password violations, if any

In SESDCN administration, the following information is displayed for each active user:

- the user identification
  If the user is an asynchronous UTM application, the conversation number is also displayed.
- the application's mode
  The following modes are possible:
  - UTM (UTM transaction application)
  - DCAM (DCAM transaction application)
  - TIAM (interactive mode)

- BAT (transaction application in batch mode).

- the time (in hours:minutes) since the beginning of the conversation (ELAPSTIME)
- the name of the computer on which the addressed databases are stored (LOCATION).
  If no database has yet been addressed, the name of the home system appears here.
- the name of the database(s) being accessed by the SESDCN user (CATALOG IN USE).
  If no database has yet been addressed, “NO” appears here.

```
SHOW-USERS

 STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

     <structured-name 1..20>(...)

         |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables (DBH administration)

Output information	Name of the S variable	T	Contents
Host name	var(*LIST).HOST	S	<name 1..8>
Application	var(*LIST).APPLICATION	S	<name 1..8>
User	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number	var(*LIST).CONVERSATION	S	<x-text 1..8> / '' (void)
Numbering	var(*LIST).NUMBER	S	<integer>
User’s state	var(*LIST).STATE	S	ADM / LOCK / RESET / INTERN / CAN / PTC / FREE / UTI / '' (void)
Display of the lock sequence	var(*LIST).LOCK-SEQUENCE	S	LOCK-SEQ / '' (void)
Number of logical files	var(*LIST).LOGICAL-FILES	S	<integer>
Number of password violations	var(*LIST). PASSWORD-ERRORS	S	<integer>

Output in S variables (DCN administration)

Output information	Name of the S variable	T	Contents
Host name	var(*LIST).HOST	S	<name 1..8>
Application	var(*LIST).APPLICATION	S	<name 1..8>

User	var(*LIST).CUSTOMER	S	<name 1..8>
Conversation number	var(*LIST).CONVERSATION	S	<x-text 1..8> / 0
Number of password violations	var(*LIST). PASSWORD-ERRORS	S	<integer>
User ID (BS2000), job name (DCAM), program unit (UTM)	var(*LIST).USER-NAME	S	<name 1..8>
Mode	var(*LIST).MODE	S	TIAM / DCAM / UTM / BAT
Runtime (hh:mm)	var(*LIST).ELAPSED-TIME	S	<time 5 .. 5>
Location	var(*LIST).LOCATION(*LIST). HOST	S	NO / <name 1..8>
Database name	var(*LIST).LOCATION(*LIST). CATALOG	S	NO / NOT IN DIST. RULE / <filename 1..18>

Example 1

Output to the screen after entering the SHOW-USERS administration statement in DBH administration:

```
%SHOW-USERS
    1.: D016ZE07TSN=8GPZADMINIST             ADM
        NO. OF LOG. FILES IN CALL-DML:          1
    2.: D016ZE09TSN=2A2VUSER0005
        NO. OF LOG. FILES IN CALL-DML:          1
    3.: D016ZE09TSN=2A2VUSER0006
        NO. OF LOG. FILES IN CALL-DML:          1
    4.: D016ZE09TSN=2A2VUSER0001
        NO. OF LOG. FILES IN CALL-DML:          1
    5.: D016ZE09TSN=2A2VUSER0002
        NO. OF LOG. FILES IN CALL-DML:          1
    6.: D016ZE09TSN=2A3VADMINIST             ADM
        NO. OF LOG. FILES IN CALL-DML:          1
    7.: D016ZE09TSN=2A2VUSER0003
        NO. OF LOG. FILES IN CALL-DML:          1
    8.: D016ZE09TSN=2A2VUSER0004
        NO. OF LOG. FILES IN CALL-DML:          1
% <date> <time> SES7215 END OF OUTPUT
%//
```

Example 2

Output to the screen after entering the SHOW-USERS administration statement in SESDCN administration:

```
%SHOW-USERS
            USER                        !MODE!ELAPS!LOCATION!     CATALOG
                                        !    ! TIME!        !     IN USE
```

```
--------------------------------------------------------------------------------
D016ZE09TSN=7QLXSEDI63  00000000SQLDB112!TIAM!00:04!D016ZE08!JDB3
                                        !    !     !D016ZE08!ACCOUNT3
                                        !    !     !D016ZE09!JDB
D016ZE09TSN=7QLZDSQL    00000000SQLDB112!TIAM!00:02!D016ZE08!ZIRD
                                        !    !     !D016ZE09!VIRST
D016ZE09TSN=7R12USER000100000000EXAMPLE !DCAM!00:38!D016ZE08!JDB3
                                        !    !     !D016ZE09!JDB
D016ZE09TSN=7R12USER000200000000EXAMPLE !DCAM!00:36!D016ZE08!ACCOUNT3
                                        !    !     !D016ZE09!ACCOUNT
%  <date> <time> SEN2014 END OF THE 'SESDCN' ADMINISTRATION STATEMENT
%//
```

When a user is active at multiple locations (Location, Catalog in Use), one output line is generated per location. The first output line then contains all the information, and the following lines only the “Location”, and “Catalog in Use” data.

Administration command in ISP format and at the CALL DML interface

The USER,PR administration command corresponds to the SHOW-USERS administration statement here.

```
USER,PR
```

5.2.3.80 SHOW-USER-SPACES

Displays the spaces used by the user and thus locked

Scope of validity

DBH administration

Function

The SHOW-USER-SPACES statement specifies all the spaces that are used by a specified user and are thus locked.

For every space identified in this way, the logical database name and the name of the space are displayed. In addition, it is indicated whether the space is accessed exclusively.

```
SHOW-USER-SPACES

 USER = <user-identification>

 ,STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>(...)

     <structured-name 1..20>(...)

         |    WRITE-MODE = *REPLACE / *EXTEND
```

Operands

USER = <user-identification>

Identifies the user. The identification must be fully qualified (see the section “Syntax for the identification of users”).

STRUCTURE-OUTPUT = *NONE / *SYSINF / <structured-name 1..20>

(WRITE-MODE = *REPLACE / *EXTEND)

In S procedures, specifies whether the information is to be output in S variables (see section “Syntax for output in S variables”).

Output in S variables

Output information	Name of the S variable	T	Contents
Numbering	var(*LIST).NUMBER	S	<integer>
Logical database name	var(*LIST).CATALOG	S	<filename 1..18>
Space name	var(*LIST).SPACE	S	<filename 1..18>
Access mode	var(*LIST).LOCK-MODE	S	EXCLUSIV / '' (void)

Example

Screen output after entry of the SHOW-USER-SPACES administration statement in DBH administration:

```
USER:           TSN=12LD
-------------------------------------------------------------------------
 NR.    ! CATALOG-NAME       ! SPACE-NAME         ! LOCK-MODE
```

```
-------------------------------------------------------------------------
     1 ! AUFTRAGKUNDEN      ! CATALOG            !
     2 ! AUFTRAGKUNDEN      ! TABLESPACE         !
%  <date> <time> SES7215 END OF OUTPUT
%//
```

Administration command in ISP format and at the CALL DML interface

The SPACE administration command corresponds to the SHOW-USER-SPACES administration statement here.

SPACE,*user-identification*

5.2.3.81 STOP-DBH

Terminates a DBH session

Scope of validity

DBH administration

Function

The STOP-DBH statement terminates the DBH session.

Before the DBH session is terminated, the DBH rolls back all open transactions except PTC transactions, empties all buffers and closes the databases involved. If there are PTC transactions, utility statements or lock sequences in the system, SESAM/SQL rejects the statement. A table containing an overview of these PTC transactions or statements is output. The TA-LOG file remains active.

In the case of an application with SESAM/SQL and openUTM, the STOP-DBH statement allows you to choose whether the restart information for the UTM warm start is to be backed up or deleted.

The restart information is backed up by default. This is important when a DBH restart, but not a UTM warm start, has been carried out after a system failure. Only if the restart information of the DBH (TA-LOG and WA-LOG file) has been backed up is a UTM warm start possible in the following DBH session.
You can terminate the DBH up to 64 times with restart-information backup without endangering the UTM warm start.

STOP-DBH
UTM-SESSION-INFO = *KEEP / *DELETE

Operands

UTM-SESSION-INFO =

Specifies whether the restart information for the UTM warm start is to be backed up or deleted.

UTM-SESSION-INFO = *KEEP
This is the default. The restart information from the TA-LOG and WA-LOG files of the DBH is backed up in case a UTM warm start becomes necessary.

UTM-SESSION-INFO = *DELETE
The restart information for the UTM warm start is deleted.

Administration command in ISP format and at the CALL DML interface

Two administration commands fulfill the function of the STOP-DBH administration statement: STOP[,KEEP] and STOP,DELETE.

```
STOP[,KEEP]
```

```
STOP,DELETE
```

If you enter only STOP, SESAM/SQL uses the STOP,KEEP command internally.

5.2.3.82 STOP-DCN

Terminates SESDCN

Scope of validity

SESDCN administration

Function

The STOP-DCN statement terminates the SESDCN without error.

If an SESDCN has been terminated, remote access is then no longer possible to databases to which it is assigned as the remote DCN in the distribution rule.

The loaded distribution rule is retained, however, until all application programs and DBHs in the distributed system and all other SESDCNs in this configuration have been terminated.

```
STOP-DCN
```

Operands

This statement has no operands.

Administration command in ISP format and at the CALL DML interface

The STOP administration command corresponds to the STOP-DCN administration statement here.

```
STOP
```

6 Evaluating request logging with SESCOSP

SESAM/SQL logs request-specific data; From the user's viewpoint, a SESCOSP request corresponds to a statement.

The system administrator uses the SET-TUNING-TRACE administration statement or the COS administration command (see "SET-TUNING-TRACE") to turn request logging on and off. The system administrator controls the logged data amount by means of the operand `PROTOCOL=*STD/*LONG`. When request logging is on, the DBH logs data in a request log file (the CO-LOG file). The request log file contains a large amount of data, which the SESCOSP utility then evaluates.

The logged data indicates which statements have been executed for which program, and which spaces, tables and databases have been addressed. The system administrator can use the logged data to follow what happened during the relevant period and carry out a precise analysis. In this way, SESCOSP reports can be used as a basis for finding out performance-critical statements. The “ Performance” manual provides assistance in analyzing performance problems and indicates any measures required to improve throughput.

The user can set parameters to cause SESCOSP to evaluate only a subset of the data logged in the request log file. You can limit the evaluation to selected databases, parts of databases, users, statements and time periods.

SESCOSP reports essentially provide the following information:

- for each statement: I/Os and time behavior
- for each statement step: I/Os, time behavior and accesses to resources
- for each transaction: I/Os and time behavior
- for each statement group: termination statistics in the form of total and average values.

SESCOSP outputs the data in selectable report formats.

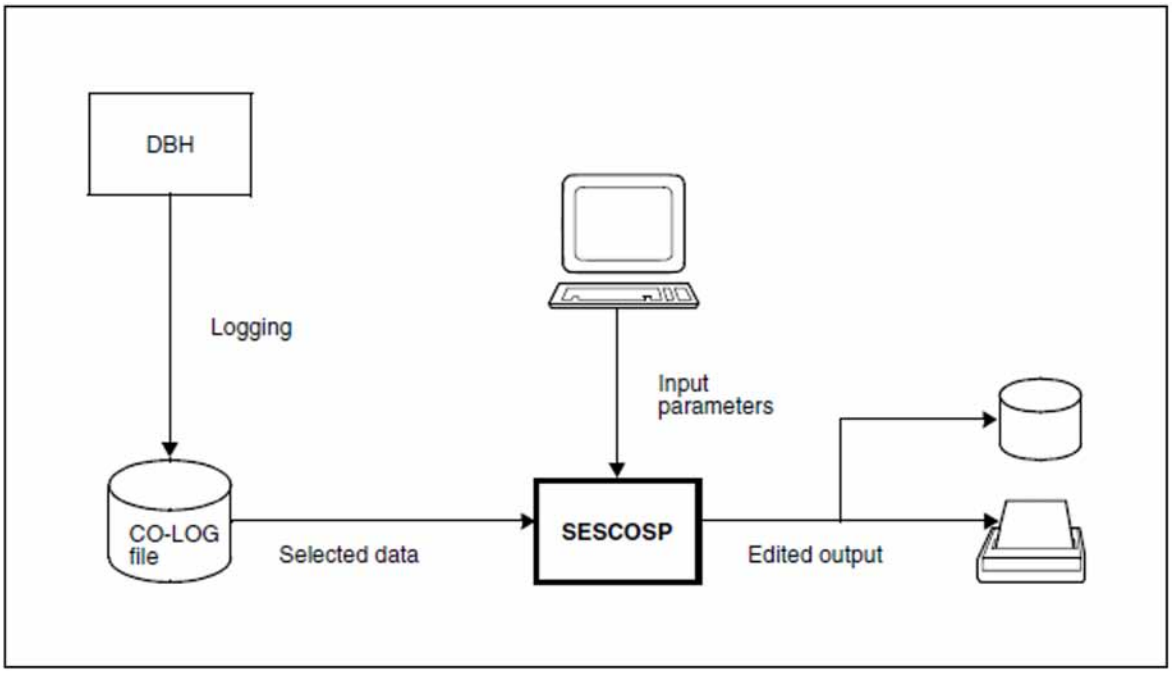


Figure 6: Request logging and evaluation with SESCOSP

The following always applies: Output sent to SYSOUT and SYSLST is not upwardly compatible. Output layout may vary depending on the version used.

6.1 Calling SESCOSP

Sequence of commands for starting SESMON

```
/START-SESAM-TUNING-TRACE-EVAL
```

Starts SESCOSP with the start command START-SESAM-TUNING-TRACE-EVAL (see section “Starting SESAM /SQL programs via start commands”).

Example

SESCOSP is started by means of the procedure BEISP.START.COSP.

```
/CALL-PROCEDURE NAME=BEISP.START.COSP,PROCEDURE-PARAMETERS=NO,
/                      LOGGING=PARAMETERS(COMMANDS=YES,DATA=YES)
```

The input parameters are read in via an SDF screen. Since only the name of the CO-LOG file to be evaluated is specified, SESCOSP generates a report with the default settings.

/BEGIN-PROCEDURE LOGGING=ALL

/ASSIGN-SYSLST TO-FILE=LST.SESDBB.P.SESCOSP.K6

/ASSIGN-SYSDTA TO-FILE=*PRIMARY

/MOD-MSG-ATTRIBUTES TASK-LANGUAGE=D

/MODIFY-SDF-OPTIONS GUIDANCE=MAXIMUM

/START-SESAM-TUNING-TRACE-EVAL

```
PROGRAMM   : SESCOSP                    STATEMENT: CREATE-REPORT
------------------------------------------------------------------------------
INPUT-FILE           = SESAM.CO-LOG.70EX.0002
                       Filename_1..54_without-generation-version
                       Input file, Standard: SESAM.CO-LOG.<TSN>.<NR>
OUTPUT               = *SYSLST
                       *SYSLST or filename_1..54_without-generation-version
                       Name of the output file
SELECT               = *ALL
                       *ALL or *PARAMETERS()
                       Selection criteria for input
REPORT-FORMAT        = *STD
                       *STD or *PARAMETERS()
                       Generic term for all selection criteria for the
                       output
------------------------------------------------------------------------------
NEXT = *CONTINUE
       *EXECUTE"F3" / + / Follow-up statement / *CONTINUE / *EXIT"K1" /
       *EXIT-ALL"F1" / *TEST"F2"
MESSAGE:  CMD0175 OTHER OPERATIONS DESIRED? PRESS *EXIT KEY
```

//CREATE-REPORT INPUT-FILE=SESAM.CO-LOG.70EX.0002

/ASSIGN-SYSLST TO-FILE=*PRIMARY

```
/MODIFY-SDF-OPTIONS GUIDANCE=EXPERT
/END-PROCEDURE
```

6.2 SESCOSP input parameters

SESCOSP expects input parameters from SYSDTA.

6.2.1 The higher-level parameter CREATE-REPORT

Function

You use CREATE-REPORT to specify which request log file SESCOSP is to evaluate, where SESCOSP is to write the report to, which data SESCOSP is to evaluate, and how the SESCOSP statistics should look.

If you enter the name of the request log file (CO-LOG file) to be evaluated in the appropriate SDF screen and send it off with the default settings, SESCOSP evaluates all the data and writes I/O statistics for each statement (see "The REPORT-FORMAT parameter PROTOCOL") to SYSLST.

All the operands of CREATE-REPORT are described below as individual parameters.

CREATE-REPORT
INPUT-FILE = ,OUTPUT = ,SELECT = ,REPORT-FORMAT =

Operands

INPUT-FILE =

See the “The individual parameter INPUT-FILE”

OUTPUT =
See the “The individual parameter OUTPUT”

SELECT =
See the “The individual parameter SELECT”

REPORT-FORMAT =
See the “The individual parameter REPORT-FORMAT”

6.2.2 The individual parameter INPUT-FILE

Function

You use this parameter to specify the file name of the request log file that SESCOSP is to evaluate.

INPUT-FILE
= <filename 1..54_without_generation_version>

Operands

INPUT-FILE = <filename 1..54_without_generation_version>

Name of the request log file (CO-LOG file) that SESCOSP is to evaluate. If necessary, see the SET-TUNING-TRACE administration statement on "SET-TUNING-TRACE". Request logging must be turned off for the file to be evaluated.

6.2.3 The individual parameter OUTPUT

Function

You use this parameter to specify whether SESCOSP is to write the edited data to SYSLST or a file.

OUTPUT
= *SYSLST / <filename 1..54_without_generation_version>

Operands

OUTPUT = *SYSLST

Writes the report to SYSLST.

OUTPUT = <filename 1..54_without_generation_version>
Writes the report to the specified file. SYSLST is assigned to the specified file internally to ensure that output is always directed to SYSLST. At the end of the program run SYSLST is reset to the primary assignment even when it already had an assignment beforehand.

6.2.4 The individual parameter SELECT

Function

You use this parameter to specify which data from the request log file is to be evaluated by SESCOSP.

The type of evaluation depends not only on the parameters specified for SELECT but also on whether statements or transactions are evaluated. You specify this by means of the REPORT-FORMAT parameter PROTOCOL (see "The REPORT-FORMAT parameter PROTOCOL"). For each SELECT parameter, the differences in the logging of statements and transactions are described below.

```
SELECT

= *ALL / *PARAMETERS(...)

  *PARAMETERS(...)

    |  CATALOG-NAMES =

    |  ,USERS =

    |  ,STATEMENTS =

    |  ,TIME =

    |  ,CPU-TIME=

    |  ,ELAPSED-TIME =

    |  ,NUMBER-OF-LOGICAL-IO=
```

Operands

SELECT = *ALL

SESCOSP evaluates all databases, all spaces, all tables and all statements and the report is not limited to specific users or specific time periods.

SELECT = *PARAMETERS(...)

The options that allow you to limit the data to be collected are described below individually as SELECT parameters.

6.2.4.1 The SELECT parameter CATALOG-NAMES

Function

You use this parameter to specify the databases to be evaluated. For each database, the report can be limited to individual schemata, tables and spaces.

A statement is logged when the specified database is accessed, or when selected schemata, tables or spaces are accessed.

A statement may also evaluated when a table is not accessed. An example of this is the end statement of a transaction.

A transaction is logged when at least one statement of the transaction accesses the specified database or selected schemata, tables or spaces.

Note that the selection of individual schemata, tables or spaces affects:

- what is output in the STEP-IO-STATISTICS and STEP-COMPLEXITY lists
- the statistical resource data in the output lines and the termination statistics, in particular the selected logged resource consumption and
- the checking of the conditions specified for the SELECT parameters CPU-TIME and NUMBER-OF-LOGICAL-IO.

```
CATALOG-NAMES

= *ALL / list-poss(10): <filename 1..18_without-all>(...)

  <filename 1..18_without-all>(...)

    |  SCHEMA-NAMES = *ALL / list-poss(10): <c-string 1..31>(...)

    |     <c-string 1..31> (...)

    |        |  TABLE-NAMES = *ALL / list-poss(10): <text 1..31> / <c-string 1..31>

    |  SPACE-NAMES = *ALL / list-poss(10):<filename 1..18 without-all>
```

Operands

CATALOG-NAMES = *ALL

All databases are to be evaluated. Specifying *ALL does not have the same effect as specifying all databases individually because internal tables are also evaluated.

CATALOG-NAMES = list-poss(10): <filename 1..18_without-all>(...)
You specify up to ten databases that are to be evaluated. The databases are identified by their physical database names (see the DBH start statement ADD-SQL-DATABASE-CATALOG-LIST on "ADD-SQL-DATABASE-CATALOG-LIST").

<filename 1..18_without-all>(...)

SCHEMA-NAMES = *ALL
All this database's schemas are to be evaluated.

SCHEMA-NAMES = list-poss(10): <c-string 1..31>(...)
You specify up to ten of this database's schemas that are to be evaluated.

<c-string 1..31> (...)

TABLE-NAMES = *ALL
All tables of this schema are to be evaluated.

TABLE-NAMES = list-poss(10): <text 1..31> / <c-string 1..31>
You specify up to ten of this schema's tables that are to be evaluated.

SPACE-NAMES = *ALL
All this database's spaces are to be evaluated.

SPACE-NAMES = list-poss(10): <filename 1..18 without-all>
You specify up to ten of this database's spaces that are to be evaluated.

Example

The following input file for SESCOSP creates a report relating only to the data of the HEI database and the DC_SCHEMA schema.

```
//  CREATE-REPORT -
//      INPUT-FILE      = CO-LOG.70EX.0002, -
//      OUTPUT          = *SYSLST, -
//      SELECT          = *PARAMETERS ( -
//          CATALOG-NAMES   = HEI ( -
//              SCHEMA-NAMES    = 'DC_SCHEMA' ( -
//                  TABLE-NAMES     = *ALL ), -
//              SPACE-NAMES     = *ALL ), -
//          USERS           = *ALL, -
//          STATEMENTS      = *ALL, -
//          TIME            = *ALL, -
//          CPU-TIME        = *ALL, -
//          ELAPSED-TIME    = *ALL, -
//          NUMBER-OF-LOGICAL-IO = *ALL ),-
//      REPORT-FORMAT   = *STD
```

6.2.4.2 The SELECT parameter USERS

Function

You use this parameter to specify the timesharing and application users to be evaluated.

> Please note that you can restrict logging to individual users by means of the USER-SELECT parameter of the SET-TUNING-TRACE administration statement (see "SET-TUNING-TRACE"). Only from these users can you make a selection for evaluation in SESCOSP by means of the USERS parameter.

USERS
= *ALL / *BY-SELECT(...) *BY-SELECT(...) | HOST-NAME = *NONE / <text 1..8> | ,APPLICATION-NAME = *NONE / *TSN (TSN=<alphanum-name 4..4>) / <text 1..8> | ,CUSTOMER-NAME = *NONE / <text 1..8> | ,CONVERSATION-ID = *NONE / <x-string 1..8> / <alphanum-name 1..8> | ,TAC = *NONE / <alphanum-name 1..8>

Operands

USERS = *ALL

All timesharing and application users are evaluated.

USERS = *BY-SELECT(...)
Only specific timesharing and application users are evaluated. You must specify a value other than *NONE for at least one operand.

HOST-NAME =
The data to be evaluated is selected by specifying a host system.

HOST-NAME = *NONE
The information to be output is not limited to that on a specific host system.

HOST-NAME = <text 1..8>
Only the data of the specified host system is evaluated. The host system is identified by its symbolic host name. In non-distributed processing, HOMEPROC always identifies the local system.

> The wildcard * (asterisk) enables you to select a subset
>
> The hosts whose names begin with the specified string (up to the first *) are selected.
>
> For example `HOST-NAME=D016*` selects all hosts whose name begins with `D016`. `HOST-NAME=*` selects all hosts.

APPLICATION-NAME =

The data to be evaluated is selected by specifying an application.

APPLICATION-NAME = *NONE
The information to be output is not limited to that of a specific application.

APPLICATION-NAME = *TSN (TSN= <alphanum-name 4..4>)
In TIAM mode you select the application by specifying the task sequence number (TSN).

APPLICATION-NAME = <text 1..8>
In UTM mode you select the application by specifying the UTM application name, in DCAM mode by specifying the DCAM application name.

> The wildcard * (asterisk) enables you to select a subset
>
> The applications whose names begin with the specified string (up to the first *) are selected.
>
> For example `APPLICATION-NAME=TSN*` selects all TIAM applications. `APPLICATION-NAME=*` selects all applications.

CUSTOMER-NAME =
The data to be evaluated is selected by specifying a user.

CUSTOMER-NAME = *NONE
The information to be output is not limited to that of a specific user.

CUSTOMER-NAME = <text 1..8>
In UTM mode the user is specified by means of the KDCSIGN name, in TIAM and DCAM modes by means of the program name.

> The wildcard * (asterisk) enables you to select a subset
>
> The users whose names begin with the specified string (up to the first *) are selected.
>
> For example `CUSTOMER-NAME=UPIC*` selects all users whose name begins with `UPIC`. `CUSTOMER-NAME=*` selects all users.

CONVERSATION-ID =
The data to be evaluated is selected by specifying a UTM conversation.

CONVERSATION-ID = *NONE
The data to be evaluated is not limited to that of a specific UTM conversation.

CONVERSATION-ID = <x-string 1..8> / <alphanum-name1..8>
The UTM conversation is identified by the UTM conversation number.

TAC =

The data to be evaluated is selected by means of the UTM transaction code, the DCAM user name or the BS2000 user ID.

TAC = *NONE
No selection by means of the UTM transaction code, the DCAM user name or the BS2000 user ID

TAC = <alphanum-name 1..8>
In UTM mode, the transaction code is specified here, in DCAM mode the DCAM user name, and in TIAM mode the BS2000 user ID.

6.2.4.3 The SELECT parameter STATEMENTS

Function

You use this parameter to specify the statements to be evaluated.

If TRANSACTION is selected for the REPORT-FORMAT format, it is not possible to specify TERMINAL-NUMBER and STATEMENT-NUMBER. If you try to do this, an error message appears.

```
STATEMENTS

= *ALL / *PARAMETERS(...)

  *PARAMETERS

    | TERMINAL-NUMBER = <x-string 1..4> / <integer..32767>

    | ,STATEMENT-NUMBER = <x-string 1..4> / <integer..32767>
```

Operands

STATEMENTS = *ALL

All statements are evaluated.

STATEMENTS = *PARAMETERS(...)

TERMINAL-NUMBER = <x-string 1..4> / <integer..32767>
Only statements entered at the specified terminal are evaluated. You specify the terminal by means of the SESAM/SQL internal terminal number (ITN). The ITN can be obtained from SESCOSP reports.

STATEMENT-NUMBER = <x-string 1..4> / <integer..32767>
Only the specified statement is evaluated. You specify the statement by means of the user-specific sequence number. The user-specific sequence number can be obtained from SESCOSP reports.

6.2.4.4 The SELECT parameter TIME-PERIOD

Function

You use this parameter to specify that SESCOSP is to evaluate only statements or transactions logged during a specific time period.

```
TIME-PERIOD

= *ALL / *INTERVAL(...)

  *INTERVAL(...)

    | FROM = *PARAMETERS(...)

    |      | DATE = <date>

    |      | ,TIME = <time>

    | ,TO = *PARAMETERS(...)

    |      | DATE = <date>

    |      | ,TIME = <time>
```

Operands

TIME-PERIOD = *ALL

The data to be evaluated is not limited to the statements or transactions of a specific time period.

TIME-PERIOD = *INTERVAL(...)
Statements or transactions logged in the period from the start time to the end time are evaluated. The start time must be before or equal to the end time.

A statement is logged when it is processed in full in the specified time period. A transaction is selected when all the statements of the transaction lie within the limits of the time period.

FROM = *PARAMETERS(...)

DATE = <date>
Date of the start time, to be entered in the format *yyyy-mm-dd*

TIME = <time>
Start time, to be entered in the format *hh*:*mm*:*ss*

TO = *PARAMETERS(...)

DATE = <date>
Date of the end time, to be entered in the format *yyyy-mm-dd*

TIME = <time>
End time, to be entered in the format *hh*:*mm*:*ss*

6.2.4.5 The SELECT parameter CPU-TIME

Function

You use this parameter to specify that SESCOSP is to evaluate only statements or transactions whose CPU time is within a specific period. You cannot select a CPU time period unless the CPU time was recorded during logging (see the SET-TUNING-TRACE administration statement on "SET-TUNING-TRACE").

The CPU time of a transaction is obtained by adding up all the CPU times of the statements involved in the transaction.

```
CPU-TIME

= *ALL / *INTERVAL(...)

  *INTERVAL(...)

    | FROM = <integer 0..2147483647>

    | ,TO = <integer 1..2147483647>
```

Operands

CPU-TIME = *ALL

The data to be evaluated is not limited to that logged during a specific CPU-time period.

CPU-TIME = *INTERVAL(...)
Statements and transactions are logged whose consumed CPU time lies between the CPU time start value (FROM) and the CPU time end value (TO). The time specified for FROM must be before or the same as that specified for TO.

FROM = <integer 0..2147483647>
CPU start time, to be entered in milliseconds or microseconds, depending on the setting of the TIME-PRECISION parameter.
Default: 0

TO = <integer 1..2147483647>
CPU end time, to be entered in milliseconds or microseconds, depending on the setting of the TIME-PRECISION parameter.
Default: 2147483647

6.2.4.6 The SELECT parameter ELAPSED-TIME

Function

You use this parameter to specify that SESCOSP is to evaluate only statements or transactions whose elapsed time is within a specific time period. Elapsed time is elapsed real time.

When transactions are evaluated, in addition to the elapsed times of the statements in the DBH, the elapsed times for the user are also taken into consideration.

```
ELAPSED-TIME

= *ALL /*INTERVAL(...)

  *INTERVAL(...)

    | FROM = <integer 0..2147483647>

    | ,TO = <integer 1..2147483647>
```

Operands

ELAPSED-TIME = *ALL

The data to be evaluated is not limited by the elapsed time.

ELAPSED-TIME = *INTERVAL(...)
The elapsed time of the statements or transactions to be evaluated is between the times specified for FROM and TO. The time specified for FROM must be before or the same as that specified for TO.

FROM = <integer 0..2147483647>
Start time, to be entered in milliseconds or microseconds, depending on the setting of the TIME-PRECISION parameter.
Default: 0

TO = <integer 1..2147483647>
End time, to be entered in milliseconds or microseconds, depending on the setting of the TIME-PRECISION parameter.
Default: 2147483647

6.2.4.7 The SELECT parameter NUMBER-OF-LOGICAL-IO

Function

You use this parameter to specify that SESCOSP is to evaluate only statements whose number of logical I/Os lies within a specified range.

The number of logical I/Os of a transaction is the sum of all the logical I/Os of the individual statements involved in the transaction.

```
NUMBER-OF-LOGICAL-IO

= *ALL /*INTERVAL(...)

  *INTERVAL(...)

    | FROM = <integer 0..2147483647>

    | ,TO = <integer 0..2147483647>
```

Operands

NUMBER-OF-LOGICAL-IO = *ALL

The evaluation is limited on the basis of the number of logical I/Os.

NUMBER-OF-LOGICAL-IO = *INTERVAL(..)
Only statements or transactions whose number of logical I/Os lie within the specified range are evaluated.

FROM = <integer 0..2147483647>
Minimum number of logical I/Os
Default: 0

TO = <integer 0..2147483647>
Maximum number of logical I/Os
Default: 2147483647

6.2.5 The individual parameter REPORT-FORMAT

Function

You use this parameter to specify the form in which the SESCOSP statistics are to appear.

REPORT-FORMAT
REPORT-FORMAT = *STD / *PARAMETERS(...) *PARAMETERS(...) | PROTOCOL = | ,TIME-PRECISION =

Operands

REPORT-FORMAT = *STD

The default settings apply to the type of statistics and the time precision and these defaults are described below for each REPORT-FORMAT parameter.

REPORT-FORMAT = *PARAMETERS(...)
The options available for formatting the data for output are described individually below and they are the REPORT-FORMAT parameters.

6.2.5.1 The REPORT-FORMAT parameter PROTOCOL

Function

You use this parameter to specify the type of statistics. The statistics are to be output as statement or transaction statistics. The format of the different reports is described in section “SESCOSP report output”.

```
PROTOCOL

= *STATEMENT(...) / *TRANSACTION

  *STATEMENT (...)

    | INFORMATION = *STRING-FORMAT / *IO-STATISTICS /

    |     *STEP-IO-STATISTICS / *STEP-COMPLEXITY
```

Operands

PROTOCOL = *STATEMENT(...)

Information is output relating to statements or statement steps.

INFORMATION = *STRING-FORMAT
The statements are output in printable form.
See "Variable parts of the report" for the structure of the report.

INFORMATION = *IO-STATISTICS
A set of I/O statistics is output for each statement.
See "Variable parts of the report" for the structure of the report.

INFORMATION = *STEP-IO-STATISTICS
A set of I/O statistics is output for each statement step.
See "Variable parts of the report" for the structure of the report.

INFORMATION = *STEP-COMPLEXITY
A set of resource statistics (complexity) is output for each statement step.

See "Variable parts of the report" for the structure of the report.

PROTOCOL = *TRANSACTION
A set of statistics is output relating to transactions.
See "Variable parts of the report" for the structure of the report.

The following input file for SESCOSP generates a set of resource statistics for each statement step.

```
//  CREATE-REPORT -
//      INPUT-FILE      = CO-LOG.70EX.0002, -
//      OUTPUT          = *SYSLST, -
//      SELECT          = *ALL, -
//      REPORT-FORMAT   = *PARAMETERS ( -
//          PROTOCOL        = *STATEMENT ( -
//              INFORMATION     = *STEP-COMPLEXITY))
```

6.2.5.2 The REPORT-FORMAT parameter TIME-PRECISION

Function

You use this parameter to specify whether time is to be specified in milliseconds or microseconds.

```
TIME-PRECISION

= *STD / *PARAMETERS(...)

  *PARAMETERS(...)

    | CPU-TIME = *MILLI-SECONDS / MICRO-SECONDS

    | ,ELAPSED-TIME = *MILLI-SECONDS / MICRO-SECONDS
```

Operands

TIME-PRECISION = *STD

The CPU time and elapsed time are output in milliseconds.

TIME-PRECISION = *PARAMETERS(...)

CPU-TIME = *MILLI-SECONDS / MICRO-SECONDS
The CPU time is output in milliseconds or microseconds.

ELAPSED-TIME = *MILLI-SECONDS / MICRO-SECONDS
The elapsed real time is output in milliseconds or microseconds.

6.3 SESCOSP report output

Each report output by SESCOSP has the following basic structure:

1. SESCOSP input parameter
   - System type
   - Operating-system version of the DBH user ID
   - DBH options
2. Variable part of the report in accordance with the REPORT-FORMAT parameter PROTOCOL
   - Output of statements
   - Output of input/output statistics per statement
   - Output of input/output statistics per statement step
   - Output of resource statistics per statement step (complexity)
   - Output of transaction statistics
3. Termination statistics
   - Time period involved
4. Catalog-space table assignment table

6.3.1 Output of general information and the DBH options

The output of the SESCOSP input parameters, the system type, the operating system version of the DBH user ID and the DBH options is structured in the same way in all reports.

Example

```
SESCOSP/SQL  VERS. <version>   TSN:9H16  ID.:
SQLDB112                                          <date> <time>       PAGE    1
//
     CREATE-REPORT INPUT-FILE = SESAM.CO-LOG, OUTPUT = SESAM.COSP.OUT.TA.ALL, SELECT
     = *PARAMETERS( CATALOG-NAMES=*ALL ,USERS=*ALL ,STATEMENTS=*ALL ,
     TIME-PERIOD=*ALL ,CPU-TIME=*ALL ,ELAPSED-TIME=*ALL ,
     NUMBER-OF-LOGICAL-IO=*ALL ), REPORT-FORMAT = *PARAMETERS ( PROTOCOL =
     *TRANSACTION, TIME-PRECISION=*PARAMETERS(CPU-TIME=*MICRO-SECONDS ,
     ELAPSED-TIME=*MILLI-SECONDS ))

Computer type:     7.500- S210-K

BS2000_Version: <version>

ACCOUNTING            *OFF
ADMINISTRATOR         *ANY
COLUMNS               1024
CURSOR-BUFFER         BUFFER-SIZE=1408   , FRAME-SIZE=4
DBH-IDENTIFICATION    CONFIGURATION-NAME=Z, DBH-NAME=X
.
.
```

6.3.2 Variable parts of the report

The variable parts of the report are described below in accordance with the REPORT-FORMAT parameter PROTOCOL.

> If an integer value is too large for an output field, the value is output as a floatingpoint number. In this case, a certain amount of precision is lost.

Output of statements

PROTOCOL=*STATEMENT(INFORMATION=*STRING-FORMAT)

SESCOSP provides the following information for each statement.

Heading	Contents
LFD	Internal record number by means of which the lines belonging to different SESCOSP reports can be assigned
DATE	Date (*yyyy-mm-dd*) on which the request was accepted
TIME	Time (*hh*:*mm*:*ss*) at which the request was accepted
HOST	In distributed processing, the host name; in nondistributed processing, this is always HOMEPROC
APPL/TSN	UTM or DCAM application name; in TIAM mode, the task sequence number TSN = *tsn*
USER/PRG	KDCSIGN name in UTM mode or DCAM or TIAM program name; possibly LOGON name
CONV-ID	Conversation ID In UTM mode: the UTM conversation number (specified in hexadecimal format)
TAC/U-ID	UTM transaction code or BS2000 user ID or DCAM user name
ITN	SESAM/SQL internal terminal number (specified in hexadecimal format)
ST-ID	Number of the statement. Forms with the ITN the unique identification of a statement (specified in hexadecimal format).
STM	Short ID of the SQL statement, of the SQL statement group, or CALL DML operation code DCL: DECLARE CURSOR SEL: SELECT OPC: OPEN CURSOR FET: FETCH STC: STORE CURSOR RST: RESTORE CURSOR CLC: CLOSE CURSOR SEL: SELECT

STM	INS: INSERT UPD: UPDATE MRG: MERGE DEL: DELETE CPR: CALL CTA: COMMIT TRANSACTION RTA: ROLLBACK TRANSACTION PTC: prepare to commit DDL: DDL statement UTI: utility statement PRP: dynamic statement SQL: other SQL statement CALL DML operation code 01: administration 1: cursor file handling 2: open 3: attribute information 4: record output 5: query 6: search 7/6: sequential search call 7/9: sequential update call 8: close 9: update (add, modify, delete) 90B: begin transaction 90R: roll back transaction 90C: end of transaction 90P: prepare to commit ?: unknown op code
CPUUnits [1]	CPU time used by the statement in milliseconds or microseconds
BruttoEI	Elapsed time of the statement in the DBH (without the connection module) (Elapsed Time)
PhysIO [1]	Number of physical read and write accesses
In CALL DML	
Statement	Statement in printable form (for CO-LOG files with standard logging max. 500 bytes)
Acknowledgement	Acknowledgment in dump format
Query area	Values in dump format (for CO-LOG files with standard logging max. 1000 bytes)
Answer area	Values in dump format (for CO-LOG files with standard logging max. 1000 bytes)
In SQL	
ICSQL-Params	ICSQL parameters in printable format (for CO-LOG files with standard logging max. 500 bytes)
SQL-String	SQL string in printable format (for CO-LOG files with standard logging max. 29696 bytes)
In-Descriptors	

	Input descriptors (see the section “Representation of data ...” below)(only for CO-LOG files with LONG logging). The information on length, type, column name and value is output for each entry.
Diagnostic	Diagnostic area in printable format (for CO-LOG files with standard logging max. 500 byte)
Output	Values in printable format (see the section “Representation of data ...” below) only for CO-LOG files with LONG logging). The information on length, type, column name and value is output for each entry. When the block mode (FETCH) is used, the information for all the records supplied is output in succession, the values (results) playing an essential role here. In the case of multiple fields the coefficient precedes the values.

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here

The extent of logging of CO-LOG files is set in the administration statement SET-TUNING-TRACE by means of the operand PROTOCOL=*STD/*LONG, see "SET-TUNING-TRACE".

Representation of data of the data types (NATIONAL) CHARACTER (VARYING)

The input descriptors and the values (In-Descriptors and Output columns) are output for the data types CHARACTER (VARYING) in printable form, e.g.:

```
In-Descriptors: ITEM 1: TYPE = CHAR
                                (20)
                BBB
```

Printable data of the data types NATIONAL CHARACTER (VARYING) is converted with the CCSN EDF03IRV to CHARACTER format and output. If the data contains non-printable characters, these are also output in the continuation line in hexadecimal format in the Unicode encoding form UTF-16. Data that cannot be converted appears in the CHARACTER format as blanks. If the data contains non-printable characters, these are also output in the continuation line in hexadecimal format in the Unicode encoding form UTF-16. Example (00D6=Ö is not printable):

```
In-Descriptors:   ITEM    1:  TYPE = NCHAR(10)
                      C   4   E
                  00D6004300340045
```

Example 1: (normal case, in-descriptors and output)

1. PREPARE statement with SQL string

```
SESCOSP/SQL  VERS. <version>   TSN:4IYB  ID.:SQLDB112 *** STRING-FORMAT ***                    <date> <time>          PAGE     1
 LFD      DATE     TIME     HOST APPL/TSN USER/PRG  CONV-ID TAC/U-ID  ITN STID STM CPUUnits   BruttoEl PhysIO
*************************************************************************************************************************
0017 <date> <time>     D016ZE09 TSN=94E7 UDML202  00000000 XDL1LOE  0002 0017 PRP        0         15      0
     ICSQL-Params:     ICSQL-Version = X'02', ICSQL-Type = 'PRE', User-Authorization-ID = 'D0USER', Consistency-Level = X'03', TA-A
                       ccess-Mode = X'02', Current-Catalog-Name = 'STANCAT4', Is-Static-Statement = FALSE, Statement-Class = X'0205
                       ', Statement-ID = 'SQLENTRQ.ST1'
     SQL-String:       --%PRAGMA DIALECT FULL --%PRAGMA CATALOG "STANCAT4"--%PRAGMA SCHEMA "SCHEMA_0001"--%PRAGMA MODULE "SQLENT
                       RQ"--%PRAGMA PREFETCH 20                          SELECT T1.COLUMN_0001, T1.COLUMN_0003, T2.COLUMN_0017
                        FROM   TABLE_0001  T1,  TABLE_0001  T2  /* self join */ WHERE T1.COLUMN_0001 > ?  AND T2.COLUMN_0002 > ? AN
                       D  CAST( SUBSTRING(T1.COLUMN_0011 FROM  1 FOR 3000) AS VARCHAR(3000)) =  CAST( SUBSTRING(T2.COLUMN_0011 FROM
                         1 FOR 3000) AS VARCHAR(3000))   ORDER BY CAST( SUBSTRING(T1.COLUMN_0017 FROM  1 FOR 3000) AS VARCHAR(3000)
                       ) ASC
     Diagnostic:       SQL-State = '00000'
     Output:           ITEM    1:  TYPE = CHAR(50)                    NAME = COLUMN_0001
                       <UNKNOWN>
                       ITEM    2:  TYPE = CHAR(50)                    NAME = COLUMN_0003
                       <UNKNOWN>
                       ITEM    3:  TYPE = VARCHAR(16000)              NAME = COLUMN_0017
                       <UNKNOWN>
```

2. SQL statement OPEN cursor

```
SESCOSP/SQL  VERS. <version>   TSN:4IYB  ID.:SQLDB112 *** STRING-FORMAT***                    <date> <time>            PAGE     1
 LFD       DATE     TIME     HOST APPL/TSN USER/PRG  CONV-ID TAC/U-ID  ITN STID STM CPUUnits   BruttoEl PhysIO
*************************************************************************************************************************
0018 <date> <time>     D016ZE09 TSN=94E7 UDML202  00000000 XDL1LOE  0002 0018 OPC        0       2537     31
     ICSQL-Params:     ICSQL-Version = X'02', ICSQL-Type = 'OPN', User-Authorization-ID = 'D0USER', Consistency-Level = X'03', TA-A
                       ccess-Mode = X'02', Current-Catalog-Name = 'STANCAT4', Is-Static-Statement = FALSE, Statement-Class = X'010B
                       ', Statement-ID = 'SQLENTRQ.ST1', Cursor-ID = 'SQLENTRQ.CU03', Is-Scroll-Cursor = FALSE
     In-Descriptors:   ITEM    1:  TYPE = CHAR(20)
                       BBB
                       ITEM    2:  TYPE = DECIMAL(10,2)
                       +00000001.11
     Diagnostic:       SQL-State = '00000'
*************************************************************************************************************************
```

3. SQL FETCH statement with result values

```
SESCOSP/SQL  VERS. <version>   TSN:4IYB  ID.:SQLDB112 *** STRING-FORMAT ***                    <date> <time>            PAGE     1
 LFD       DATE     TIME     HOST APPL/TSN USER/PRG  CONV-ID TAC/U-ID  ITN STID STM CPUUnits   BruttoEl PhysIO
*************************************************************************************************************************
0019 <date> <time>     D016ZE09 TSN=94E7 UDML202  00000000 XDL1LOE  0002 0019 FET        0          3      1
     ICSQL-Params:     ICSQL-Version = X'02', ICSQL-Type = 'FCH', User-Authorization-ID = 'D0USER', Consistency-Level = X'03', TA-A
                       ccess-Mode = X'02', Current-Catalog-Name = 'STANCAT4', Is-Static-Statement = FALSE, Statement-Class = X'010A
                       ', Statement-ID = 'SQLENTRQ.ST1', Cursor-ID = 'SQLENTRQ.CU03', Fetch-Orientation = X'0000000000000000'
     Diagnostic:       SQL-State = '00000'
     Output:           ITEM    1:  TYPE = CHAR(50)
                       VOLLMF
                       ITEM    2:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    3:  TYPE = VARCHAR(16000)
                       THIS WILL BE A SEMI-LONG VCHAR
*************************************************************************************************************************
```

Example 2: BLOCK MODE (Output)

SQL FETCH statement with result values (PREPARE and FETCH as in the normal case)

```
*************************************************************************************************************************
0012 <date> <time>     D016ZE09 TSN=94E7 UDML202  00000000 XDL1LOE  0002 0012 FET        0          3      0
     ICSQL-Params:     ICSQL-Version = X'02', ICSQL-Type = 'FCH', User-Authorization-ID = 'D0USER', Consistency-Level = X'03', TA-A
                       ccess-Mode = X'02', Current-Catalog-Name = 'STANCAT4', Is-Static-Statement = FALSE, Statement-Class = X'010A
                       ', Statement-ID = 'SQLENTRQ.ST1', Cursor-ID = 'SQLENTRQ.CU02', Fetch-Orientation = X'0000000000000000'
     Diagnostic:       SQL-State = '00000'
     Output:           ROW     1, SQL-STATE = 00000 ------------------------------------------------------------------------------

                       ITEM    1:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    2:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    3:  TYPE = VARCHAR(16000)
                       THIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEM
                       I-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS W
                       ILL BE A SEMI-LONG VCHARTHIS WILL BE A S
                       ROW     2, SQL-STATE = 00000 ------------------------------------------------------------------------------
                       ITEM    1:  TYPE = CHAR(50)
                       BBBAkfdljhgslkdjfhglkdfhg
                       ITEM    2:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    3:  TYPE = VARCHAR(16000)
                       THIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEM
                       I-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS W
                       ILL BE A SEMI-LONG VCHARTHIS WILL BE A S
                       ROW     3, SQL-STATE = 00000 ------------------------------------------------------------------------------
                       ITEM    1:  TYPE = CHAR(50)
                       CCCAkfdljhgslkdjfhglkdfhg
                       ITEM    2:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    3:  TYPE = VARCHAR(16000)
                       THIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEM
                       I-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS W
                       ILL BE A SEMI-LONG VCHARTHIS WILL BE A S
                       ROW     4, SQL-STATE = 00000 ------------------------------------------------------------------------------
                       ITEM    1:  TYPE = CHAR(50)
                       DDDAkfdljhgslkdjfhglkdfhg
                       ITEM    2:  TYPE = CHAR(50)
                       AAAAkfdljhgslkdjfhglkdfhg
                       ITEM    3:  TYPE = VARCHAR(16000)
                       THIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEM
                       I-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS WILL BE A SEMI-LONG VCHARTHIS W
                       ILL BE A SEMI-LONG VCHARTHIS WILL BE A S
*************************************************************************************************************************
```

Example 3: Output of values of multiple fields (Output)

SQL FETCH statement with result values (PREPARE and FETCH as in the normal case)

```
*************************************************************************************************************************
0061 <date> <time>     D016ZE09 TSN=94E7 UDML202  00000000 XDL1LOE  0002 0061 FET        0          9      0
     ICSQL-Params:     ICSQL-Version = X'02', ICSQL-Type = 'FCH', User-Authorization-ID = 'D0USER', Consistency-Level = X'03', TA-A
                       ccess-Mode = X'02', Current-Catalog-Name = 'STANCAT4', Is-Static-Statement = FALSE, Statement-Class = X'010A
                       ', Statement-ID = 'SQLENTRQ.ST1', Cursor-ID = 'SQLENTRQ.CU07', Fetch-Orientation = X'0000000000000000'
     Diagnostic:       SQL-State = '00000'
     Output:           ITEM    1:  TYPE = CHAR(256)
                         1 AAAAAAAAAABBBBBBBBBB1CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         2 AAAAAAAAAABBBBBBBBB2CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         3 AAAAAAAAAABBBBBBBBB3CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         4 AAAAAAAAAABBBBBBBBB4CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         5 AAAAAAAAAABBBBBBBBB5CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         6 AAAAAAAAAABBBBBBBBB6CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         7 AAAAAAAAAABBBBBBBBB7CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         8 AAAAAAAAAABBBBBBBBB8CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                         9 AAAAAAAAAABBBBBBBBB9CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                        10 AAAAAAAAAABBBBBBBBB0CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                        11 AAAAAAAAAABBBBBBBBB1CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                        12 AAAAAAAAAABBBBBBBBB2CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                       197 AAAAAAAAAABBBBBBBBB7CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                       198 AAAAAAAAAABBBBBBBBB8CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
```

```
                199 AAAAAAAAAABBBBBBBBBB9CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                200 AAAAAAAAAABBBBBBBBBB0CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                ITEM    2:  TYPE = CHAR(256)
                  1 AAAAAAAAAABBBBBBBBBB1CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                  2 AAAAAAAAAABBBBBBBBBB2CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
                  3 AAAAAAAAAABBBBBBBBBB3CCCCCCCCCDDDDDDDDDDEEEEEEEEEEAAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEE
**************************************************************************************************************************
```

Output of I/O statistics for each statement

PROTOCOL=*STATEMENT(INFORMATION=*IO-STATISTICS)

SESCOSP provides the following information for each statement:

Heading	Contents
LFD	See the “The STRING-FORMAT report”
DATE	See the “The STRING-FORMAT report”
TIME	See the “The STRING-FORMAT report”
APPL /TSN	See the “The STRING-FORMAT report”
TAC/U-ID	See the “The STRING-FORMAT report”
ITN	See the “The STRING-FORMAT report”
ITN	See the “The STRING-FORMAT report”
ST-ID	See the “The STRING-FORMAT report”
DBC#	Internal database number by means of which the database name can be obtained via the catalog-space table assignment table (see "Catalog-space table assignment table") (specified in hexadecimal format)
C-ID	In SQL: number of the statement that declared the cursor, if the evaluated statement refers to a cursor In CALL DML: logical file identifier
STM	See the “The STRING-FORMAT report”
CPUUnits [1]	See the “The STRING-FORMAT report”
BruttoEI	See the “The STRING-FORMAT report”
NettoEI [1]	Activity time of the statement; i.e. the sum of the active time of the user and the active time of all subfunctions (=statement steps)
PhDBIO [1]	Number of physical read and write accesses of the database
LgDBIO [1]	Number of logical read and write accesses of the database
PhCuIO	Number of physical read and write accesses of cursor files
LgCuIO	Number of logical read and write accesses of cursor files

Table 26: The IO-STATISTICS report

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here

Example

```
SESCOSP/SQL  VERS. <version>    TSN:9H16  ID.:SQLDB112                                          <date> <time>      PAGE    1
//
     CREATE-REPORT INPUT-FILE = SESAM.CO-LOG, OUTPUT = SESAM.COSP.OUT.IOSTAT, SELECT
     = *ALL, REPORT-FORMAT = *PARAMETERS ( PROTOCOL = *STATEMENT ( INFORMATION =
     *IO-STATISTICS ), TIME-PRECISION=*PARAMETERS(CPU-TIME=*MICRO-SECONDS ,
     ELAPSED-TIME=*MICRO-SECONDS ))
.
.
SESCOSP/SQL  VERS. <version>  TSN:9H16  ID.:SQLDB112  *** IO-STATISTICS ***                    <date> <time>       PAGE    3
 LFD       DATE     TIME APPL/TSN TAC/U-ID  ITN STID DBC# C-ID STM CPUUnits   BruttoEl  NettoEl PhDBIO   LgDBIO PhCuIO   LgCuIO
0079      <date>  <time> TSN=1A2Z DBSRE    0004 0283 0000  A2  90C      429       2808      426       0        0      0        0
007B      <date>  <time> TSN=1A24 DBSRE    0011 0C14 0001  A1  6       1874      11899    11899       0        4      0        0
007C      <date>  <time> TSN=1A2Y DBSRE    0001 13A5 0001  A1  6       1700       2581     2581       0        6      0        0
007E      <date>  <time> TSN=1A4G XDL1DBT3 0015 0026 0003 0026 SEL     4616      10771    10771       0        2      0        0
0080      <date>  <time> TSN=1A4M XDL1DBT3 000B 0043 0003 0041 CLC     2802      13086    13086       0        0      0        0
0081      <date>  <time> TSN=1A4P XDL1DBT3 000C 0025 0003 0025 SEL     4191       4531     4531       0        2      0        0
0082      <date>  <time> TSN=1A4H XDL1DBT3 0005 0034 0002 0034 SEL     3454       3662     3662       0        2      0        0
007A      <date>  <time> TSN=1A25 DBSRE    0009 0BFB 0001  A1  90B     2129      85790    12139       0        7      0        0
007F      <date>  <time> TSN=1A22 DBSRE    001C 0851 0001  A1  6       1698      67756    67756       0        5      0        0
0083      <date>  <time> TSN=1A23 DBSRE    0002 0A96 0001  A2  90B     1187       1193     1193       0        4      0        1
0084      <date>  <time> TSN=1A24 DBSRE    0016 0866 0001  A2  90B     2180      11168    11168       0        4      0        1
0085      <date>  <time> TSN=1A22 DBSRE    0017 06D7 0000  A4  90C      355        357      357       0        0      0        0
0086      <date>  <time> TSN=1A23 DBSRE    000D 0CB5 0001  A1  6       1610       1611     1611       0        6      0        0
0087      <date>  <time> TSN=1A25 DBSRE    000E 0884 0001  A2  90B     2100       2188     2188       0        4      0        1
008A      <date>  <time> TSN=1A4G XDL1DBT3 0015 0027 0003 0027 SEL     5105      10402    10402       0        2      0        0
008C      <date>  <time> TSN=1A4M XDL1DBT3 000B 0044 0000      CTA      421        425      425       0        0      0        0
008D      <date>  <time> TSN=1A24 DBSRE    0011 0C15 0001  A1  6       1910       4149     4149       0        7      0        0
0089      <date>  <time> TSN=1A4N XDL1DBT3 001B 002D 0002 002D OPC    10817      63424    31374       1        5      0        6
008E      <date>  <time> TSN=1A25 DBSRE    0009 0BFC 0001  A1  6       1741       7107     7107       0        4      0        0
008B      <date>  <time> TSN=1A4H XDL1DBT3 0005 0035 0001 0035 SEL     5775      42389    25097       0        2      0        0
.
.
```

Output of I/O statistics for each statement step

PROTOCOL=*STATEMENT(INFORMATION=*STEP-IO-STATISTICS)

SESCOSP provides this information for each statement step (see the SUB subfunction) and each SQL table.

Heading	Contents
LFD	See the “The STRING-FORMAT report”
ITN	See the “The STRING-FORMAT report”
ST-ID	See the “The STRING-FORMAT report”
DBC#	See the “The IO-STATISTICS report”
S#	Identification number of the space in the database. See the assignment table on "Catalog-space table assignment table" (specified in hexadecimal format)
TB#	Identification number of the table in the space. See the assignment table on "Catalog-space table assignment table" (specified in hexadecimal format)
C-ID	See the “The IO-STATISTICS report”
SUB	Subfunction (statement step) PRIO: statement start (request acceptance) OPT: optimizer INT: interpreter BTSC: base-table-scan TTSC: temp-table-scan UPSC: update-scan SVT: deactivation of service task
STM	See the “The STRING-FORMAT report”
	CPU time used for the statement step, in milliseconds or microseconds

CPUUnits [1]	
BruttoEl	Elapsed time of the statement step in the DBH
NettoEl [1]	Activity time of the statement step
PhACIO	Number of physical read and write accesses of the access data
LgACIO	Number of logical read and write accesses of the access data
PhDAIO	Number of physical read and write accesses of the primary data
LgDAIO	Number of logical read and write accesses of the primary data
PhCuIO [2]	Number of physical read and write accesses of cursor files
LgCuIO [2]	Number of logical read and write accesses of cursor files

Table 27: The STEP-IO-STATISTICS report

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here
[2]When SUB=SVT, these values come from the service task

Example

```
SESCOSP/SQL  VERS. <version>     TSN:9H16  ID.:SQLDB112                                          <date> <time>       PAGE     1
//
     CREATE-REPORT INPUT-FILE = SESAM.CO-LOG, OUTPUT = SESAM.COSP.OUT.STEPIO, SELECT
     = *ALL, REPORT-FORMAT = *PARAMETERS ( PROTOCOL = *STATEMENT ( INFORMATION =
     *STEP-IO-STATISTICS ), TIME-PRECISION=*PARAMETERS(CPU-TIME=*MICRO-SECONDS ,
     ELAPSED-TIME=*MILLI-SECONDS ))
.
.
SESCOSP/SQL  VERS. <version>   TSN:9H16  ID.:SQLDB112  *** STEP-IO-STATISTICS ***              <date> <time>        PAGE   25
 LFD  ITN STID DBC# S#  TB# C-ID SUB  STM CPUUnits BruttoEl  NettoEl PhACIO   LgACIO PhDAIO   LgDAIO PhCuIO   LgCuIO
0203 0005 0045 0001         0045 INT  SEL     1132        2        2      0        0      0        0      0        0
               0001 02 0002      BTSC         1607        2        2      0        1      0        1      0        0
0205 001B 0047 0001         0047 PRIO OPC      791       10       10      0        0      0        0      0        0
               0001 02 0002      BTSC         1761        3        3      0        0      0        0      0        0
0205 001B 0047 0001         0047 OPT  OPC      180        0        0      0        0      0        0      0        0
0205 001B 0047 0001         0047 INT  OPC      470        1        1      0        0      0        0      0        0
               0001 02 0002      BTSC         5943       23       23      0        2      0        2      0        6
                                 SVT           120             1234                                      890     1200
0206 0026 00B8 0000          AA  PRIO 90C      352        1        1      0        0      0        0      0        0
0206 0026 00B8 0000          AA  INT  90C      368        1        1      0        0      0        0      0        0
0207 001A 0014 0000              PRIO CTA      688      100        2      0        0      0        0      0        0
0170 0020 01AA 0001          AD  PRIO 6        333        2        2      0        0      0        0      0        0
0170 0020 01AA 0001          AD  INT  6        583        6        6      0        0      0        0      0        0
               0001 02 0002      BTSC         2907     3722       16      0        7      0        2      0        0
0209 000A 0050 0000              PRIO CTA      340        0        0      0        0      0        0      0        0
0201 0011 0C20 0001          A1  PRIO 6        257        0        0      0        0      0        0      0        0
0201 0011 0C20 0001          A1  INT  6        312        0        0      0        0      0        0      0        0
               0001 02 0002      BTSC         1412      182      166      0        4      0        0      0        0
01D0 0024 0001 0001         0001 PRIO SQL     1476        2        2      0        0      0        0      0        0
               0001 01 0005      BTSC         4435       33       33      0        3      0        3      0        0
01D0 0024 0001 0001         0001 OPT  SQL   118578      939      786      0        0      0        0      0        0
               0000 00 0000      BTSC          547        1        1      0        0      0        0      0        0
               0001 01 0007      BTSC         1629       33       33      0        1      0        1      0        0
               0001 01 000B      BTSC         2284       16        3      0        2      0        1      0        8
               0001 01 000F      BTSC         1420        2        2      0        4      0        1      0        0
               0001 01 0012      BTSC        50593      459      185      0        2      0       12      3      347
01D0 0024 0001 0001         0001 INT  SQL     1080        1        1      0        0      0        0      0        0
               0001 02 0002      BTSC         1262       10       10      0        1      0        1      0        0
020A 001B 0048 0001         0047 PRIO FET      497        3        3      0        0      0        0      0        0
020A 001B 0048 0001         0047 OPT  FET     2305       15       15      0        0      0        0      0        0
020A 001B 0048 0001         0047 INT  FET      606        1        1      0        0      0        0      0        0
               0001 02 0002      BTSC          277        0        0      0        0      0        0      0        2
.
.
```

Output of resource statistics for each statement step (complexity)

PROTOCOL=*STATEMENT(INFORMATION=*STEP-COMPLEXITY)

SESCOSP provides this information for each statement step (see the SUB subfunction) and each SQL table.

Heading	Contents
LFD	See the “The STRING-FORMAT report”

ITN	See the “The STRING-FORMAT report”
ST-ID	See the “The STRING-FORMAT report”
DBC#	See the “The IO-STATISTICS report”
S#	See the “The STEP-IO-STATISTICS report”
TB#	See the “The STEP-IO-STATISTICS report”
C-ID	See the “The IO-STATISTICS report”
SUB	See the “The STEP-IO-STATISTICS report”
STM	See the “The STRING-FORMAT report”
CPUUnits [1]	See the “The STEP-IO-STATISTICS report”
BruttoEI	See the “The STEP-IO-STATISTICS report”
NettoEl [1]	See the “The STEP-IO-STATISTICS report”
#SI	Number of indexes to be evaluated
#Att	Number of columns (attributes) to be projected
#Pra	Number of predicates to be checked
#Dea	Number of deactivations
#Cal	Number of kernel calls
#int.SortRec	Number of records in internal SORT requests
#ext.SortRec	Number of records in external SORT requests

Table 28: The STEP-COMPLEXITY report

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here

Example

```
SESCOSP/SQL  VERS. <version>   TSN:9H16  ID.:SQLDB112                                          <date> <time>       PAGE    1
//
     CREATE-REPORT INPUT-FILE = SESAM.CO-LOG, OUTPUT = SESAM.COSP.OUT.STCOMP, SELECT
     = *ALL, REPORT-FORMAT = *PARAMETERS ( PROTOCOL = *STATEMENT ( INFORMATION =
     *STEP-COMPLEXITY ) )
.
.
SESCOSP/SQL  VERS. <version>   TSN:9H16  ID.:SQLDB112  *** STEP-COMPLEXITY ***                 <date> <time>       PAGE   25
 LFD  ITN STID DBC# S#  TB# C-ID SUB  STM CPUUnits BruttoEl  NettoEl   #SI  #Att  #Pra  #Dea  #Cal  #int.SortRec  #ext.SortRec
0203 0005 0045 0001          0045 INT  SEL        1        2        2                        0
               0001 02 0002       BTSC           2        2        2     0    28     0     0     3
0205 001B 0047 0001          0047 PRIO OPC        1       10       10                        0
               0001 02 0002       BTSC           2        3        3     0    30     1     0     1
0205 001B 0047 0001          0047 OPT  OPC        0        0        0                        0
0205 001B 0047 0001          0047 INT  OPC        0        1        1                        0
               0001 02 0002       BTSC           6       23       23     0    30     1     0     1
0206 0026 00B8 0000           AA  PRIO 90C        0        1        1                        0
0206 0026 00B8 0000           AA  INT  90C        0        1        1                        0
0207 001A 0014 0000               PRIO CTA        1      100        2                        0
0170 0020 01AA 0001           AD  PRIO 6          0        2        2                        0
0170 0020 01AA 0001           AD  INT  6          1        6        6                        0
               0001 02 0002       BTSC           3     3722       16     1     2     1     1     1
                                  SVT          230             1200                                    13123456        1255100
0209 000A 0050 0000               PRIO CTA        0        0        0                        0
0201 0011 0C20 0001           A1  PRIO 6          0        0        0                        0
0201 0011 0C20 0001           A1  INT  6          0        0        0                        0
               0001 02 0002       BTSC           1      182      166     1     2     1     0     1
01D0 0024 0001 0001          0001 PRIO SQL        1        2        2                        0
               0001 01 0005       BTSC           4       33       33     0     0     0     0     3
01D0 0024 0001 0001          0001 OPT  SQL      119      939      786                        0
               0000 00 0000       BTSC           1        1        1     0     0     0     0     1
               0001 01 0007       BTSC           2       33       33     0     8     0     0     3
               0001 01 000B       BTSC           2       16        3     0     3     3     0     4
               0001 01 000F       BTSC           1        2        2     0     1     6     0     3
               0001 01 0012       BTSC          51      459      185     0    11     3     1   170
01D0 0024 0001 0001          0001 INT  SQL        1        1        1                        0
```

```
              0001 02 0002      BTSC           1       10       10     0    29     1     0     3
020A 001B 0048 0001          0047 PRIO FET      0        3        3                        0
020A 001B 0048 0001          0047 OPT  FET      2       15       15                        0
020A 001B 0048 0001          0047 INT  FET      1        1        1                        0
              0001 02 0002      BTSC           0        0        0     0    30     1     0     1
020B 001F 0267 0000           A0  PRIO 90C      0        0        0                        0
020B 001F 0267 0000           A0  INT  90C      0        0        0                        0
020E 0005 0046 0002          0046 PRIO SEL      0        0        0                        0
020E 0005 0046 0002          0046 OPT  SEL      2       10       10                        0
020E 0005 0046 0002          0046 INT  SEL      1        1        1                        0
              0002 02 0002      BTSC           1        2        2     0    28     0     0     3
020C 0015 003C 0003          003C PRIO SEL      1        1        1                        0
020C 0015 003C 0003          003C OPT  SEL      2       29       16                        0
020C 0015 003C 0003          003C INT  SEL      1        1        1                        0
              0003 04 0005      BTSC           1        3        1     0    28     0     0     3
020F 000D 0CC3 0001           A1  PRIO 90B      0        0        0                        0
020F 000D 0CC3 0001           A1  INT  90B      0        0        0                        0
              0001 02 0002      BTSC           1        1        1     1     2     1     0     1
.
```

.

.

Output of transaction statistics

Output of transaction statistics

PROTOCOL=*TRANSACTION

SESCOSP provides the following information for each transaction.

Heading	Contents
LFD	The sequential number of the transaction's last statement is logged
DATE	Start date (*yyyy-mm-dd*) of the transaction
TIME	Start time (*hh*:*mm*:*ss*) of the transaction
APPL /TSN	See the “The STRING-FORMAT report”
USER /PRG	See the “The STRING-FORMAT report”
TAC/U-ID	See the “The STRING-FORMAT report”
ITN	See the “The STRING-FORMAT report”
ST-ID	The number of the transaction's last statement is logged
CPUUnits 1	Sum of the CPU time used by all of the transaction's statements in milliseconds or microseconds
BruttoEl	Sum of the elapsed times of all of the transaction's statements in the DBH (without connection module)
TransEl	Total time of the transaction, including communication times and the time taken by the user to pause for thought
PhACIO	Sum of the physical read and write accesses to the access data of all of the transaction's statements
LgACIO	Sum of the logical read and write accesses to the access data of all of the transaction's statements
PhDAIO	Sum of the physical read and write accesses to the primary data of all of the transaction's statements
LgDAIO	Sum of the logical read and write accesses to the primary data of all of the transaction's statements

PhCuIO	Sum of the physical read and write accesses to the cursor files of all of the transaction's statements
LgCuIO	Sum of the logical read and write accesses to the cursor files of all of the transaction's statements

Table 29: The TRANSACTION report

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here

Example

```
SESCOSP/SQL  VERS. <version>   TSN:9H16 ID.:SQLDB112                                          <date> <time>       PAGE     1
//
     CREATE-REPORT INPUT-FILE = SESAM.CO-LOG, OUTPUT = SESAM.COSP.OUT.TA.ALL, SELECT
     = *PARAMETERS( CATALOG-NAMES=*ALL ,USERS=*ALL ,STATEMENTS=*ALL ,
     TIME-PERIOD=*ALL ,CPU-TIME=*ALL ,ELAPSED-TIME=*ALL ,
     NUMBER-OF-LOGICAL-IO=*ALL ), REPORT-FORMAT = *PARAMETERS ( PROTOCOL =
     *TRANSACTION, TIME-PRECISION=*PARAMETERS(CPU-TIME=*MICRO-SECONDS ,
     ELAPSED-TIME=*MILLI-SECONDS ))
.
.
SESCOSP/SQL  VERS. <version>    TSN:9H16 ID.:SQLDB112 *** TRANSACTION ***                  <date> <time>       PAGE     2
 LFD       DATE     TIME APPL/TSN USER/PRG TAC/U-ID  ITN STID CPUUnits BruttoEl  TransEl PhACIO LgACIO PhDAIO LgDAIO PhCuIO LgCuIO
00C9 <date> <time>       TSN=1A24 USER0001 DBSRE    0011 0C1A     4685       14      336      0     16      0      2      0      0
00CB <date> <time>       TSN=1A24 USER0002 DBSRE    0016 086B     2006       10      124      0      4      0      0      0      1
00D2 <date> <time>       TSN=1A23 USER0001 DBSRE    000D 0CB6     5642      102     1684      0     14      0      2      0      0
00D6 <date> <time>       TSN=1A25 USER0002 DBSRE    000E 0887     2327        3       43      0      4      0      0      0      1
00D7 <date> <time>       TSN=1A2Z USER0002 DBSRE    0004 0287     2576       25      123      0      4      0      0      0      1
00E0 <date> <time>       TSN=1A2Z USER0018 DBSRE    0012 026D     2621       73      368      0      4      0      0      0      1
00E1 <date> <time>       TSN=1A23 USER0002 DBSRE    0002 0A99     2366       26      197      0      4      0      0      0      1
.
.
```

6.3.3 Termination statistics

Heading	Contents
Function	Function for which the values are added up. A function in this case is a CALL DML operation code or an SQL statement (group). See “The STRING-FORMAT report” (field 10, STM).
# Auftr	Number of requests for this function
in %	Requests as a percentage of the total number of evaluated requests
Phys. I/O [1]	Number of physical disk accesses
Log. I/O [1]	Number of logical read and write accesses
L:P-AC	Number of logical read and write accesses for the physical read and write accesses of the access data
L:P-DA	Number of logical read and write accesses for the physical read and write accesses of the primary data
L:P-Cu	Number of logical read and write accesses for the physical read and write accesses of the temporary data
Ig:Auf	Number of read and write accesses per request
CPUUnits [1]	Total CPU time
Mitl-CPU [1]	Average CPU requirements per request
WaitTime [1]	Average wait time in the DBH of a job (the wait time in the DBH is equal to the difference between the average net elapsed time and the average CPU time).
Mitl-Ela	Average gross elapsed time

Table 30: Termination statistics

[1]In SESAM/SQL V9.0 and higher, the values from the service tasks are also contained here

Example

```
SESCOSP/SQL  VERS. <version>   TSN:9H16 ID.:SQLDB112 *** SUMMARY ***                          <date> <time>        PAGE     1
Funktion                     #Auftr in% Phys.I/O   Log. I/O L:P-AC L:P-DA L:P-Cu Lg:Auf   CPUUnits Mitl-CPU WaitTime Mitl-Ela
OPEN CURSOR                      40   2        1         293      0     56      0      7     259599     6489       17       47
FETCH                            40   2        0          58      0      0      0      1     138819     3470       10       25
STORE CURSOR                      9   0        0           0      0      0      0      0      19891     2210        0       10
CLOSE CURSOR                     31   1        0           0      0      0      0      0      74353     2398        4       22
SELECT                          164  10       13         368    202     13      0      2     890109     5427       52       84
UPDATE                            8   0        4         188      0     12      0     23      67916     8489       44      124
COMMIT WORK                     169  10        0           0      0      0      0      0      69899      413        8       11
OTHER SQL-STATEMENT              11   0        8         864      0     26    128     78     421031    38275      253      438
SUCHFRAGE                       249  15        0        1375      0      0      0      5     465313     1868       93      103
BEGIN TRANSACTION               435  27        0       12323      0      0      0     28    1262338     2901       41       59
COMMIT TRANSACTION              435  27        0           0      0      0      0      0     226812      521        6        9
Time period captured :  <date> <time>   until  <date> <time>
```

6.3.4 Catalog-space table assignment table

Heading	Meaning
DBC#	Identification number of the database
S#	Identification number of the space
TB#	Identification number of the table
Catalog-Name	Logical database name
Space-Name	Space name
COPY-#	Copy number of the space
Schema-Name	Schema name
Table-Name	Table name

Table 31: Catalog-space table assignment table

Example

```
SESCOSP/SQL  VERS. <version>  TSN:9H16  ID.:SQLDB112  *** Catalog-Space-Table-Assignment       <date> <time>       PAGE    1
DBC# S#  TB# Catalog-Name       Space-Name         COPY-# Schema-Name                    Table-Name
0001 01 0005 KON1CATALOG        CATALOG                   DEFINITION_SCHEMA             SYSTEM_ENTRIES
0001 01 0007 KON1CATALOG        CATALOG                   DEFINITION_SCHEMA             TABLES
0001 01 000B KON1CATALOG        CATALOG                   DEFINITION_SCHEMA             UNIQUE_CONSTRAINTS
0001 01 000F KON1CATALOG        CATALOG                   DEFINITION_SCHEMA             PRIVILEGES
0001 01 0012 KON1CATALOG        CATALOG                   DEFINITION_SCHEMA             INDEXES
0001 02 0002 KON1CATALOG        SPACE0001                 KON1GESAMT                    KON1SDB0
0002 01 0005 KON3CATALOG        CATALOG                   DEFINITION_SCHEMA             SYSTEM_ENTRIES
0002 01 0007 KON3CATALOG        CATALOG                   DEFINITION_SCHEMA             TABLES
0002 01 000B KON3CATALOG        CATALOG                   DEFINITION_SCHEMA             UNIQUE_CONSTRAINTS
0002 01 000F KON3CATALOG        CATALOG                   DEFINITION_SCHEMA             PRIVILEGES
0002 01 0012 KON3CATALOG        CATALOG                   DEFINITION_SCHEMA             INDEXES
0002 02 0002 KON3CATALOG        SPACE0001                 KON3GESAMT                    KON3SDB0
0003 01 0005 KON4CATALOG        CATALOG                   DEFINITION_SCHEMA             SYSTEM_ENTRIES
0003 01 0007 KON4CATALOG        CATALOG                   DEFINITION_SCHEMA             TABLES
0003 01 000B KON4CATALOG        CATALOG                   DEFINITION_SCHEMA             UNIQUE_CONSTRAINTS
0003 01 000F KON4CATALOG        CATALOG                   DEFINITION_SCHEMA             PRIVILEGES
0003 01 0012 KON4CATALOG        CATALOG                   DEFINITION_SCHEMA             INDEXES
0003 04 0002 KON4CATALOG        SPACE0003                 KON4NETZ                      KON4SDB0
```

7 Outputting operational data with SESMON

The SESMON performance monitor collects system data on current database operation and edits it on the basis of various statistical criteria.

The system administrator can see from the data obtained what effect the DBH and DCN options have on the time behavior of the database system, the extent to which its resources are utilized, and where bottlenecks could develop. From this, he/she can work out how to set the database system to meet the requirements of the specific application.

The “ Performance” manual provides assistance in analyzing performance problems and indicates any measures that may have to be taken.

Access from the World Wide Web

You can also access the administration program SESADM, the performance monitor SESMON and the utility monitor SESUTI all from a unified access on the World Wide Web (WWW or Web for short) with the aid of the software product WebTransactions (WebTA).

To access the SESAM programs via the Web, you only need a standard browser in addition to the software product WebTransactions.

Web access is described in the document “ WebTA access for SESAM/SQL” shipped together with SESAM/SQL-Server. This document is also available from our manual server under the software product SESAM/SQL.

Output via SNMP to a management platform

You can use a SESAM subagent to transfer the data from the performance monitor SESMON via SNMP to a management platform, see "Output of statistical data via SNMP".

Output via openSM2

You can also transfer the data from the performance monitor SESMON to the software monitor openSM2 in the BS2000 system, see "Outputting statistical data via openSM2".

Counter and counter overflow

In the case of large databases and long-running database operation, the probability that counters of the performance monitor will become unwieldy or even overflow increases. For information on this see the control options in SESMON ("Overview of the statistical output of SESMON") and the information on counter overflows ("DBH form I/O").

How SESMON works

The components shown in the figure below store data in the various pools. SESMON reads data from the pools, edits it, and outputs it to forms, printed lists or files. SESMON draws a distinction between SESDCN statistical output and DBH statistical output.

After SESMON is started interactively, you are requested to specify in a selection form which statistical information you require. SESMON then displays the selected forms cyclically: the CONFIGURATION forms first, if requested, and then the selected SESDCN forms, followed by the selected DBH forms for each DBH.

SESMON runs as a separate task and does not affect runtime behavior.

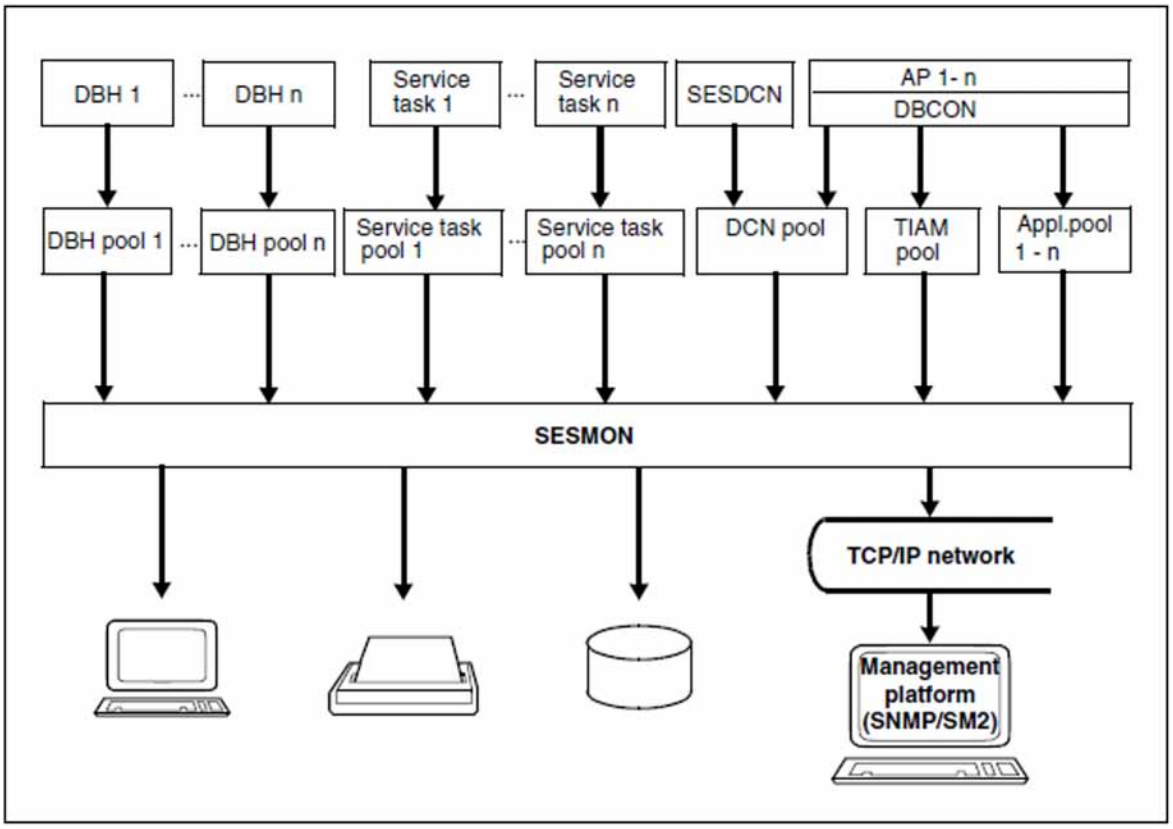


Figure 7: Collecting and evaluating data with SESMON

After an internal restart of SESAM/SQL-DBH or SESDCN, interval counters are not supplied for the first output.

> The following always applies: Output sent to SYSOUT and SYSLST is not upwardly compatible. Output layout may vary depending on the version used.

7.1 Overview of the statistical output of SESMON

The different output options are described below:

- Output on screen
- Output to a file
- Output to SYSLST
- Output via SNMP to a management platform
- Output via openSM2 in the BS2000 system

Output on screen

To allow you to select the information you require easily, it is subdivided into a number of topics and distributed across a number of different forms.

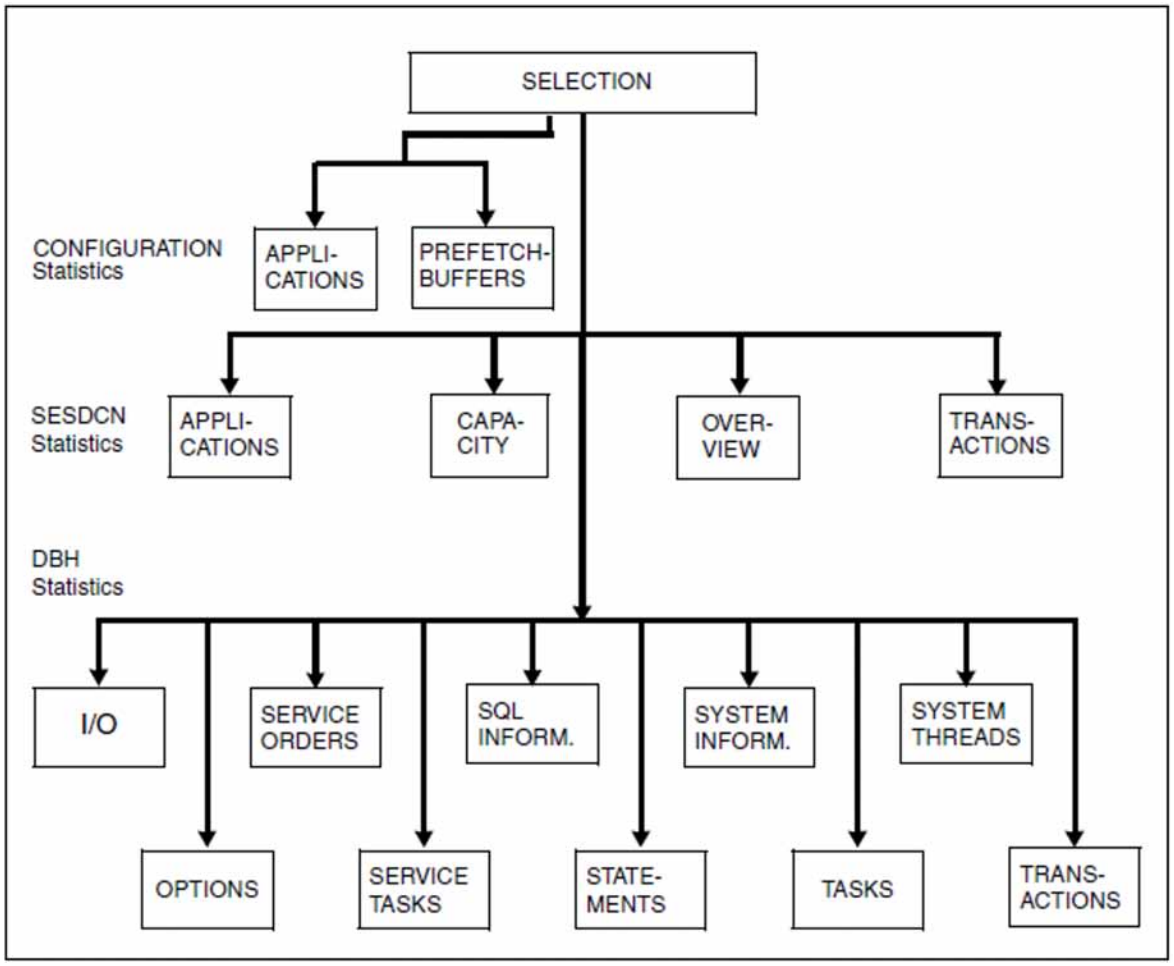


Figure 8: SESMON statistics output

The CONFIGURATION form “APPLICATIONS”

The form provides information on transaction applications which do not operate in distributed mode.

The CONFIGURATION form “PREFETCH-BUFFERS”

The form shows the buffer space made available when requested for block mode and the percentage of the buffer being used.

The SESDCN form “APPLICATIONS”

This form provides information on transaction applications which operate in distributed mode.

The SESDCN form “CAPACITY”

This form contains information on message volume, pool assignment and resource bottlenecks.

The SESDCN form “OVERVIEW”

The form contains general information on SESDCN operation, such as the set DCN options and values that apply to the whole SESDCN session.

The SESDCN form “TRANSACTIONS”

This form provides information on the state and behavior of transactions.

The DBH form “I/O”

This form contains information on logical and physical read and write accesses.

The DBH form “OPTIONS”

This form displays the current settings of the DBH options.

The DBH form “SERVICE ORDERS”

The form shows the DDL or utility statements currently being processed in the service tasks as well as the user ID of the user for each request. If there is a COPY, EXPORT, RECOVER, REFRESH, REORG SPACE or UNLOAD service request, its progress is indicated in detail.

The DBH form “SERVICE TASKS”

This form supplies values on the service tasks and the requests to be processed therein.

The DBH form “SQL INFORMATION”

This form displays information on SQL statements, SQL access plans, plan buffers and calls of specific SQL components.

The DBH form “STATEMENTS”

The form provides an overview of all active statements. It shows both the statements running within transactions as well as those running outside transactions.

The DBH form “SYSTEM INFORMATION”

This form displays the current settings and values for DBH operation, in particular the size and occupancy of individual containers. You can also use this form to obtain information on the progress of a DBH session restart.

The DBH form “SYSTEM THREADS”

This form provides information on active system threads. The system threads shown are write threads that deal with open write requests to spaces.

The DBH form “TASKS”

This form provides extended statistical data for a multitasking system that can be used to determine the utilization of the individual DBH tasks.

The DBH form “TRANSACTIONS”

This form provides information on the number of transactions in the various transaction states and the number of DML statements.

You will find the exact layout of the forms as of "Layout of the forms".

Resetting counters in the statistical outputs

In the SELECTION form of SESMON (see "The layout of the SELECTION form") you can also specify the value “r” or “R” (reset) for the screen output (Output Medium Screen).
When the statistical values are output, this value causes the counters which run during the entire DBH session, e.g. the number of transactions in the DBH form TRANSACTIONS, to be reset. Counters whose objects have a shorter lifetime, e.g. counters for objects such as user session, statement or file, cannot be reset.

Internally SESMON stores the current absolute counter statuses and displays these relative to the counts stored, starting at zero, in the follow-up forms. This makes it easier to monitor the counts at particular intervals.

The counter statuses stored become invalid in the following cases:

- Restart of the DBH or DCN session
- Internal restart of the DBH
- Change of the configuration or of the DBH in the SELECTION form of SESMON
- Restart of SESMON

In these cases the absolute counter statuses are displayed (again).

You can also specify the value “c” or “C” (continue) for Output Medium Screen. This causes the counter statuses to be displayed in the follow-up forms relative to the counter status stored when the refresh time or the form selection is changed. If this was not preceded by a “reset” or the “reset” has become invalid, a message is issued to indicate that the function required cannot be executed.

Output to SYSLST

The SYSLST forms contain the same information as the screen forms, but in some cases it is more detailed. In batch mode, forms are output to SYSLST in accordance with the specifications in the SET-MONITOR-OPTIONS statement. If SESMON is started interactively, only those forms selected from the “SELECTION” form are output to SYSLST (see "The layout of the SELECTION form").

You will find the exact layout of the output as of "Layout of output to SYSLST".

Output to a file

SESMON outputs all the statistical data to a file so that it can be processed by an evaluation program.

You will find the exact layout of the output as of "Layout of output to a file".

Output via SNMP to a management platform

SESMON transfers the statistical data via a SESAM subagent and SNMP to a management plattform, see "Output of statistical data via SNMP".

Output via openSM2

SESMON also transfers the statistical data to the software monitor openSM2 in the BS2000 system, see "Outputting statistical data via openSM2".

7.2 Running SESMON

- Starting SESMON
  - Start parameters when running SESMON in interactive mode
  - Start parameters when running SESMON in batch mode
- SESMON administration
- Automatic termination of SESMON operation

7.2.1 Starting SESMON

Sequence of commands for starting SESMON

```
[/ADD-FILE-LINK LINK-NAME=SEMSTAT,FILE-NAME=file]                    1.
/START-SESAM-PERF-MONITOR                                            2.
```

1. *file* stands for the name of the file to which SESMON writes the statistical data. Without this statement, SESMON writes the data to a SAM file called SESMON.STATISTIK.*yyyy-mm-dd*.hhmmss by default.
2. Starts SESMON with the start command START-SESAM-PERF-MONITOR (see section “Starting SESAM/SQL programs via start commands”).

> With SESAM/SQL-Server, the concurrent use of multiple correction delivery statuses is possible. When the performance monitor is started up, the correction delivery status of the DBH to be monitored is then specified with the VERSION parameter. If this is not done, the performance monitor may start up with the most recent correction delivery status while the DBH is still working with an older status. In this case, you receive a message indicating that the DBH to be monitored does not exist (see also section “Specifying the correction status with the start command”).

7.2.1.1 Start parameters when running SESMON in interactive mode

The start parameters are entered in the SELECTION form, which SESMON displays after it is started in interactive mode. To see the layout of the “SELECTION” form, refer to section “The layout of the SELECTION form”.

7.2.1.2 Start parameters when running SESMON in batch mode

In batch mode, SESMON expects the start parameters from SYSDTA via the SET-MONITOR-OPTIONS statement (see page below).
Form output is not possible in batch mode. The default value for SYSLST output can also be changed in batch mode.

Function

The statement contains the start parameters for SESMON in batch mode.

```
SET-MONITOR-OPTIONS

RUNTIME = *INFINITE / *DBHDCN / <integer 60..99999>
,CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
,DBH-NAMES = *NONE / list-poss(10) : *BLANK / <alphanum-name 1..1>
,OUTPUT = *SNMP / *SM2 / list-poss(2) : *FILE(...) / *SYSLST(...)
  *FILE(...)
    | REFRESH-TIME = *NONE / <integer 1..999>
  *SYSLST(...)
    | REFRESH-TIME = *NONE / <integer 1..999>
    | ,SESAM-CONFIGURATION = *NO / *YES(...)
    |   *YES(...)
    |     | APPLICATIONS = *NO / *YES
    |     | ,PREFETCH-BUFFERS = *NO / *YES
    | ,SESDCN = *NO / *YES(...)
    |   *YES(...)
    |     | APPLICATIONS = *NO / *YES
    |     | ,CAPACITY = *NO / *YES
    |     | ,OVERVIEW = *NO / *YES
    |     | ,TRANSACTIONS = *NO / *YES(...)
    |     |       *YES(...)
    |     |         | NUMBER = 200 / <integer 10..999>
    | ,SESAM-DBH = *NO / *YES(...)
    |   *YES(...)
    |     |   I-O = *NO / *YES
    |     | ,SERVICE-ORDERS = *NO / *YES
    |     | ,SERVICE-TASKS = *NO / *YES
    |     | ,SQL-INFORMATION = *NO / *YES
    |     | ,STATEMENTS = *NO / *YES
    |     | ,SYSTEM-INFORMATION = *NO / *YES
    |     | ,SYSTEM-THREADS = *NO / *YES
    |     | ,TASKS = *NO / *YES
```

```
|       | ,TRANSACTIONS = *NO / *YES(...)
|       |       *YES(...)
|       |           | NUMBER = 200 / <integer 10..999>
```

Operands

RUNTIME = *INFINITE / *DBHDCN / <integer 60..99999>

You specify the length of the time period during which SESMON is to collect data. *INFINITE specifies that SESMON runs until it is terminated by the STOP administration command (see "SESMON administration"), the cancellation of the SESMON task or the termination of all tasks monitored by SESMON. The termination of all the tasks monitored by SESMON does not cause SESMON to be terminated when the CONFIGURATION forms have not been selected. You specify the length of the SESMON runtime in seconds.

RUNTIME = *DBHDCN
SESMON is automatically terminated when all the relevant DBHs and DCNs are terminated.

CONFIGURATION-NAME = *BLANK / <alphanum-name 1..1>
You enter the name of the configuration in which SESMON is to carry out statistical analyses.

DBH-NAMES = *NONE / list-poss(10) : *BLANK / <alphanum-name 1..1>

You enter the names of the DBHs for which SESMON is to carry out statistical analyses. By default (*NONE), statistical analyses is not carried out on any DBHs.

Where OUTPUT=*SNMP/*SM2, SESMON only collects data for the first DBH name. Where OUTPUT=*SNMP and DBH-NAMES=*NONE, SESMON collects all configuration data not assigned to a DBH. Where OUTPUT=*SM2, a DBH name must be specified.

OUTPUT = *SNMP / *SM2 / list-poss(2) : *FILE(...) / *SYSLST(...)
You specify where the statistical data is to be output and at what intervals new values are to be output (the refresh time). This is also the time period during which the values are collected. In each case, the values apply to the period immediately prior to output whose length is specified here.

OUTPUT = *SNMP
SESMON transfers the statistical data via a SESAM subagent and SNMP to a management plattform, see "Output of statistical data via SNMP".

OUTPUT = *SM2
SESMON transfers the statistical data to the software monitor openSM2 in the BS2000 system, see "Outputting statistical data via openSM2".

OUTPUT = *FILE(...)
The output destination for the statistical data is a file. The name of the file is either specified by means of the ADD-FILE-LINK command and assigned under the link name SEMSTAT when SESMON is started, or the default name (SESMON.STATISTIK.*yyyy-mm-dd.hhmmss*) is used.

REFRESH-TIME = *NONE / <integer 1..999>

Refresh time in seconds; *NONE means that only one analysis is carried out and SESMON then terminates if it does not have to support any other output destinations.

OUTPUT = *SYSLST(...)
The output destination for the statistical data is SYSLST.

REFRESH-TIME = *NONE / <integer 1..999>
Refresh time in seconds; *NONE means that only one analysis is carried out and SESMON then terminates if it does not have to support any other output destinations.

SESAM-CONFIGURATION = *NO / *YES(...)
Selection of the masks that refer to the configuration

***YES(...)**

APPLICATIONS = *NO / *YES
You specify whether the “APPLICATIONS” form is to be output for nondistributed application users.

PREFETCH-BUFFERS = *NO / *YES
You specify whether the “PREFETCH-BUFFERS” form is to be output.

SESDCN = *NO / *YES(...)
Selection of the masks that refer to SESDCN

***YES(...)**

APPLICATION = *NO / *YES
You specify whether the “APPLICATIONS” form is to be output for distributed application users.

CAPACITY = *NO / *YES
You specify whether the “CAPACITY” form is to be output.

OVERVIEW = *NO / *YES
You specify whether the “OVERVIEW” form is to be output.

TRANSACTIONS = *NO / *YES(...)
You specify whether the “TRANSACTIONS” form is to be output.

***YES(...)**

NUMBER = 200 / <integer 10..999>
You specify the maximum number of transactions output to SYSLST.

SESAM-DBH = *NO / *YES(...)

Selection of masks that refer to the DBH.

***YES(...)**

I-O = *NO / *YES
You specify whether the “I/O” form is to be output.

SERVICE-ORDERS = *NO / *YES
You specify whether the “SERVICE-ORDERS” form is to be output.

SERVICE-TASKS = *NO / *YES
You specify whether the “SERVICE-TASKS” form is to be output.

SQL-INFORMATION = *NO / *YES
You specify whether the “SQL-INFORMATION” form is to be output.

STATEMENTS = *NO / *YES
You specify whether the “STATEMENTS” form is to be output.

SYSTEM-INFORMATION = *NO / *YES
You specify whether the “SYSTEM-INFORMATION” form is to be output.

SYSTEM-THREADS = *NO / *YES
You specify whether the “SYSTEM-THREADS” form is to be output.

TASKS = *NO / *YES
You specify whether the “TASKS” form is to be output.

TRANSACTIONS = *NO / *YES(...)
You specify whether the “TRANSACTIONS” form is to be output.

***YES(...)**

NUMBER = 200 / <integer 10..999>
You specify the maximum number of transactions output to SYSLST.

7.2.2 SESMON administration

SESMON is administered by the command INFORM-PROGRAM. For reasons of compatibility the earlier command SEND-MSG can also be used.

Calling the SELECTION form in interactive mode

After interrupting SESMON with the key, you can enter K2 `/INFORM-PROGRAM MSG='SEL'` in order to call the SELECTION form, modify the values and continue SESMON analysis with modified parameter values. Alternatively, you can output the SELECTION form by means of hitting the K1 key.

Terminating SESMON in interactive mode

After interrupting SESMON with the K2 key, you can terminate SESMON by entering `/INFORM-PROGRAM MSG='STOP'.` Alternatively, you can terminate SESMON by hitting the K3 key.

Terminating SESMON in batch mode

When SESMON is running in batch mode, you terminate it with`/INFORM-PROGRAM JOB-ID=*TSN(TSN=tsn), MSG='STOP'` from the console of the BS2000 system administrator. You specify the task sequence number (`tsn`).

SESMON can be terminated by means of /INFORM-PROGRAM also by a user with the privilege OPERATING or TSOS.

7.2.3 Automatic termination of SESMON operation

Interactive mode

When all the tasks SESMON has to monitor are terminated, the SELECTION form is displayed.

If a CONFIGURATION form (APPLICATIONS or PREFETCH-BUFFERS) was marked, the SELECTION form does not appear and the CONFIGURATION form(s) continue to be output to the selected output destination**.** SESMON is not terminated even if the DBH is terminated.

Batch mode

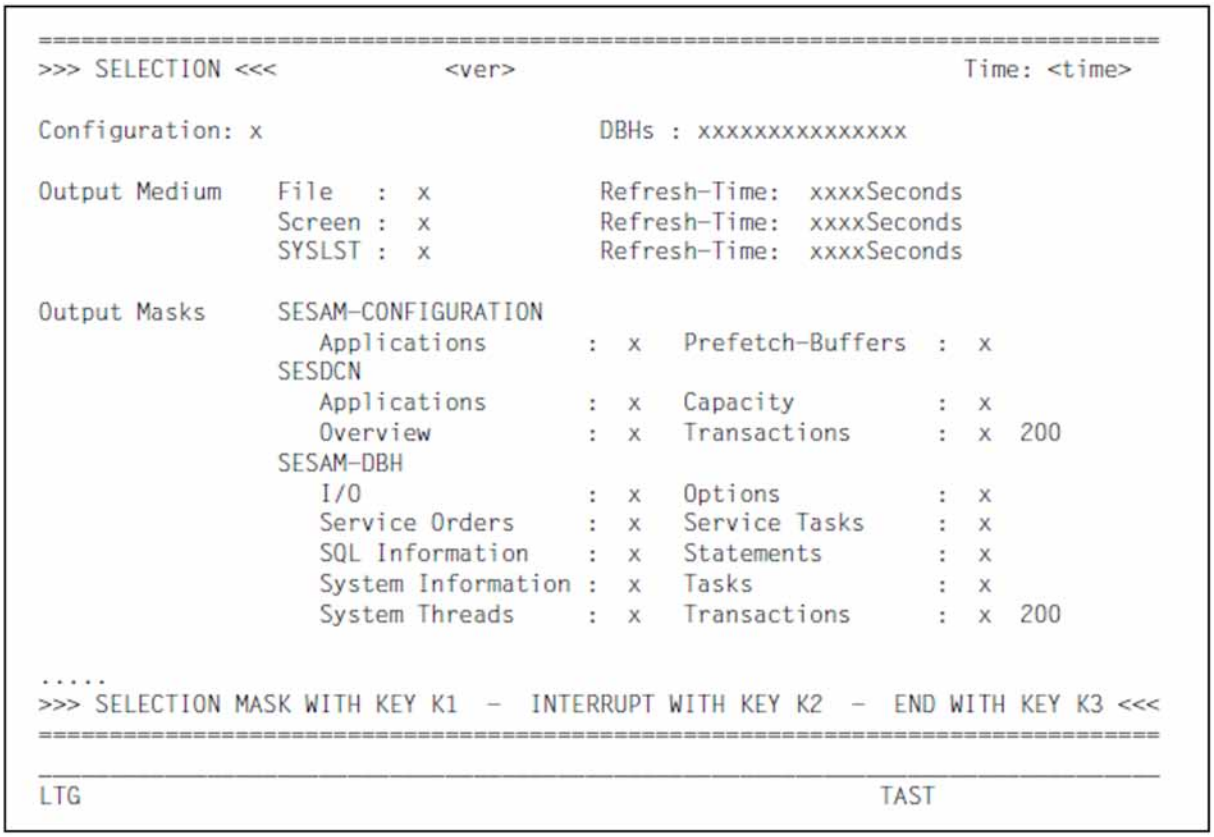
In batch mode, SESMON terminates when all the tasks it has to monitor are terminated.

If a CONFIGURATION form was requested to be output (APPLICATIONS or PREFETCH-BUFFERS), it continues to be output to file or SYSLST after all the tasks are terminated. SESMON is not terminated even if the DBH is terminated.

Task switches at termination of SESMON

Normal termination	task switches 11 and 12 are not set.
Termination with a warning	task switch 11 is set.
Termination with an error	task switch 12 is set.

7.3 The layout of the SELECTION form



The user can enter selection criteria in the fields marked with „x“.

Field descriptions

<ver> SESMON version.

Time Time at which the form is output.

Configuration

Enter the configuration SESMON is to monitor. All letters, all digits and the blank are permitted for the configuration name.

DBHs

user can enter up to ten SESAM/SQL DBHs for which the DBH forms are to be output. All letters, all digits and the blank are permitted.

Output Medium

You specify here the destination(s) to which the information is to be output by entering x or X against each output destination. You also set the refresh time for each medium selected.

This is the period of time in seconds to elapse before the next output on this medium. It is also the time period during which the values are collected. In each case, the values apply to a period immediately prior to output whose length is specified here.

The values 0 to 999 are permitted as refresh times for output.

If you enter 0, the output is executed only once.

In the case of screen output with a refresh time of 0, you can select only one form. The displayed form remains on screen.

Resetting counters

You can also enter “r” or “R” (reset) against Output Medium Screen. This causes the counters which run during the entire DBH session to be reset, see "Overview of the statistical output of SESMON".

You can also enter “c” or “C” (continue) against Output Medium Screen. This causes the counter statuses to be displayed in the follow-up forms relative to the counter status stored when the refresh time or the form selection is changed.

Output Masks

You must enter x or X against each form you require.

With SESDCN and SESAM-DBH, TRANSACTIONS (the number of logged transactions) has a default value of 200. Values from 10 to 999 can be entered.

This selection applies only to output to the screen and to SYSLST. Output to a file always includes all data.

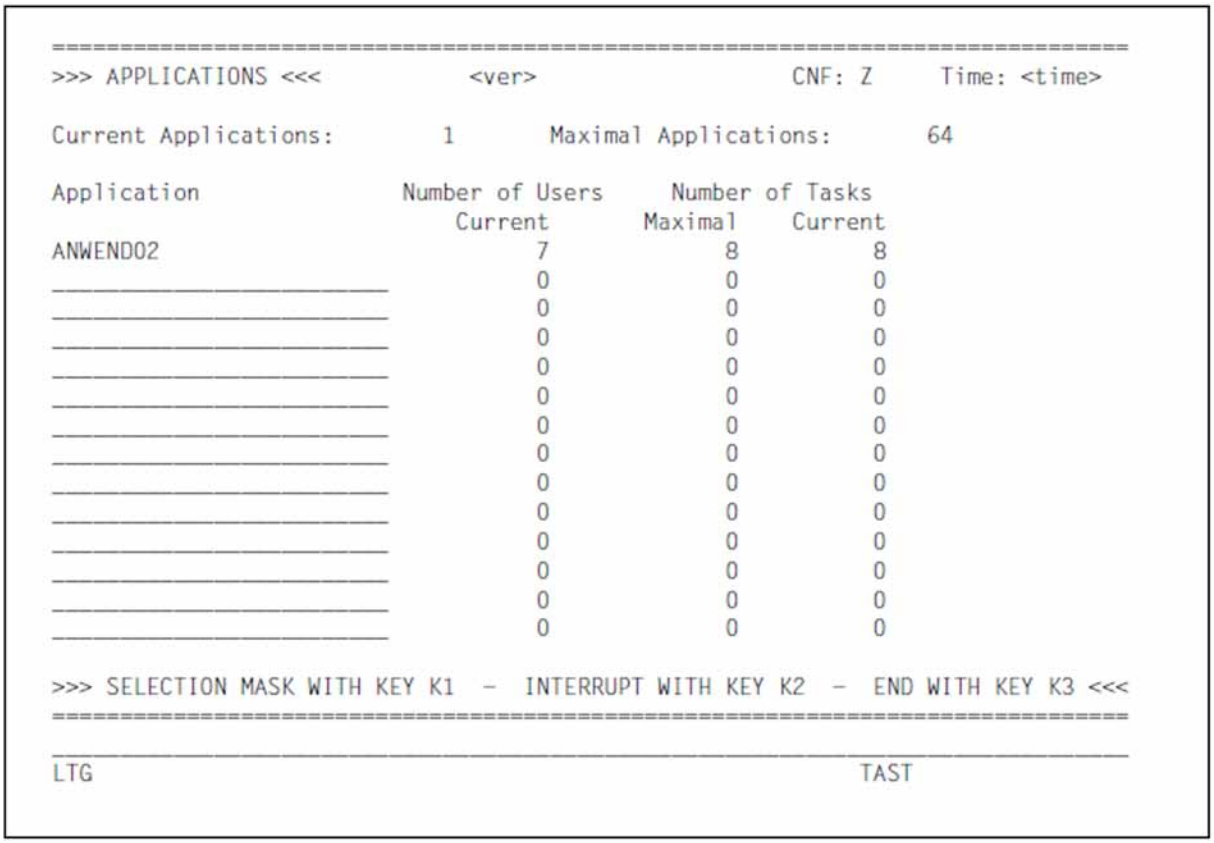
..... Messages are output in these lines.

7.4 Layout of the forms

- CONFIGURATION form APPLICATIONS
- CONFIGURATION form PREFETCH-BUFFERS
- SESDCN form APPLICATIONS
- SESDCN form CAPACITY
- SESDCN form OVERVIEW
- SESDCN form TRANSACTIONS
- DBH form I/O
- DBH form OPTIONS
- DBH form SERVICE ORDERS
- DBH form SERVICE TASKS
- DBH form SQL INFORMATION
- DBH form STATEMENTS
- DBH form SYSTEM INFORMATION
- DBH form TASKS
- DBH form SYSTEM THREADS
- DBH form TRANSACTIONS

7.4.1 CONFIGURATION form APPLICATIONS

The form provides information on transaction applications which do not operate in distributed mode.



```
================================================================================
>>> APPLICATIONS <<<          <ver>                    CNF: Z     Time: <time>

Current Applications:       1       Maximal Applications:       64

Application              Number of Users     Number of Tasks
                            Current       Maximal    Current
ANWEND02                          7             8          8
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0
________________________          0             0          0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
________________________________________________________________________________
LTG                                                                      TAST
```

Field descriptions

<ver>	Version of the DBH.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Current Applications

Number of transaction applications currently active.

Maximal Applications

Maximum possible number of transaction applications.

Application

Name of the transaction application to which the numbers output in this line apply (eight characters). Only those transaction applications that are currently active are listed.

Number of Users Current

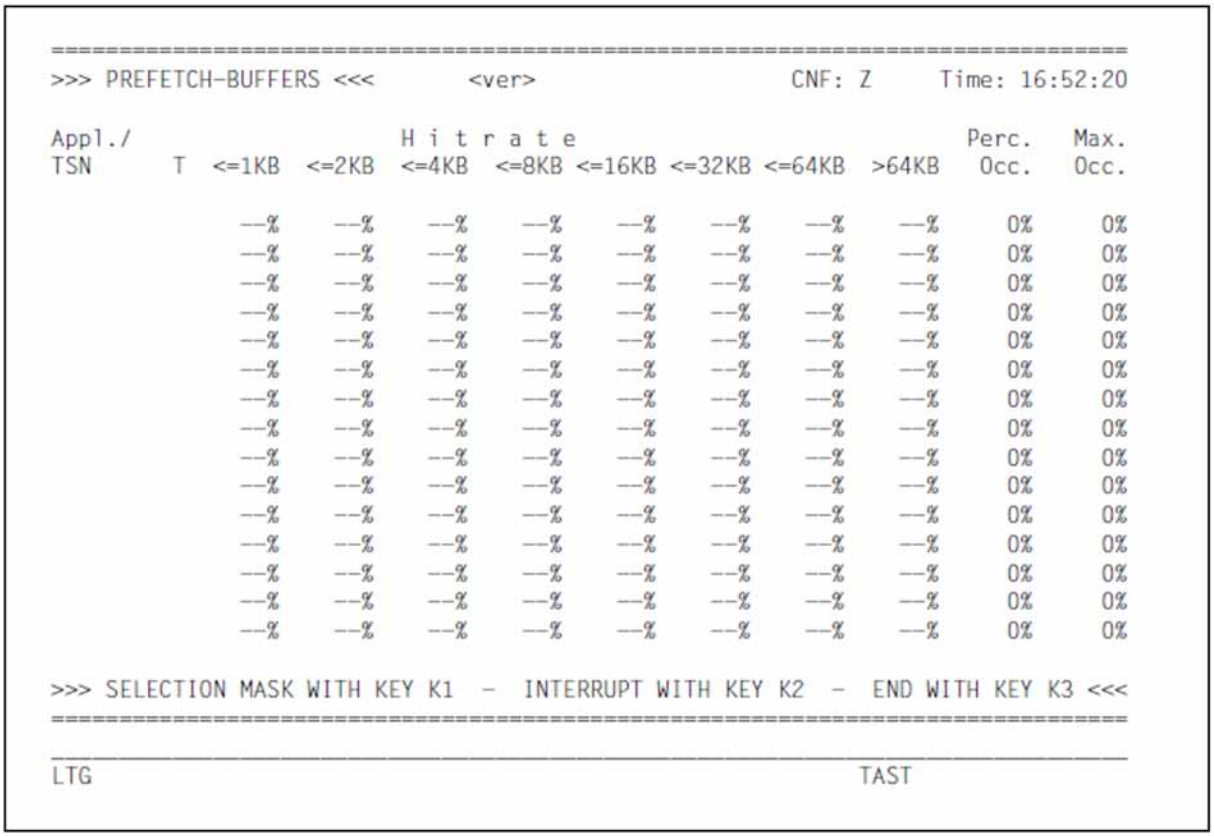
Current number of users of the specified transaction application. The output is sorted in descending order according to the number of active users. A maximum of 14 transaction applications are output.

Number of Tasks Maximal/Current

Maximum or current number of tasks of this transaction application.

7.4.2 CONFIGURATION form PREFETCH-BUFFERS

This form contains information on the buffer space made available when requested:



Field descriptions

<ver>	Version of the DBH.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Appl./TSN

Name of the openUTM transaction application (up to 8 characters) or the TSN in the case of a TIAM application.

T (Type)

Type of the application.

U: UTM application

T: TIAM application

Hitrate

Percentage of the requested memory space made available. The information is subdivided on the basis of how much memory space is requested. The entry “--%” means that no requests have yet been made in this memory class.

Percentage Occupation

Percentage of the buffer occupied.

Maximal Occupation

Maximum percentage of the buffer that can be occupied.

The entries (of which there can be up to 15) are sorted on the basis of the quality of the buffer parametrization. Up to 100 entries are output to SYSLST.

The following aspects are included in the quality assessment:

- Hit rate for memory requests:

  Hit rate in percent =
  (total memory space received /
  total memory space requested) * 100
- Maximum percentage of the buffer occupied

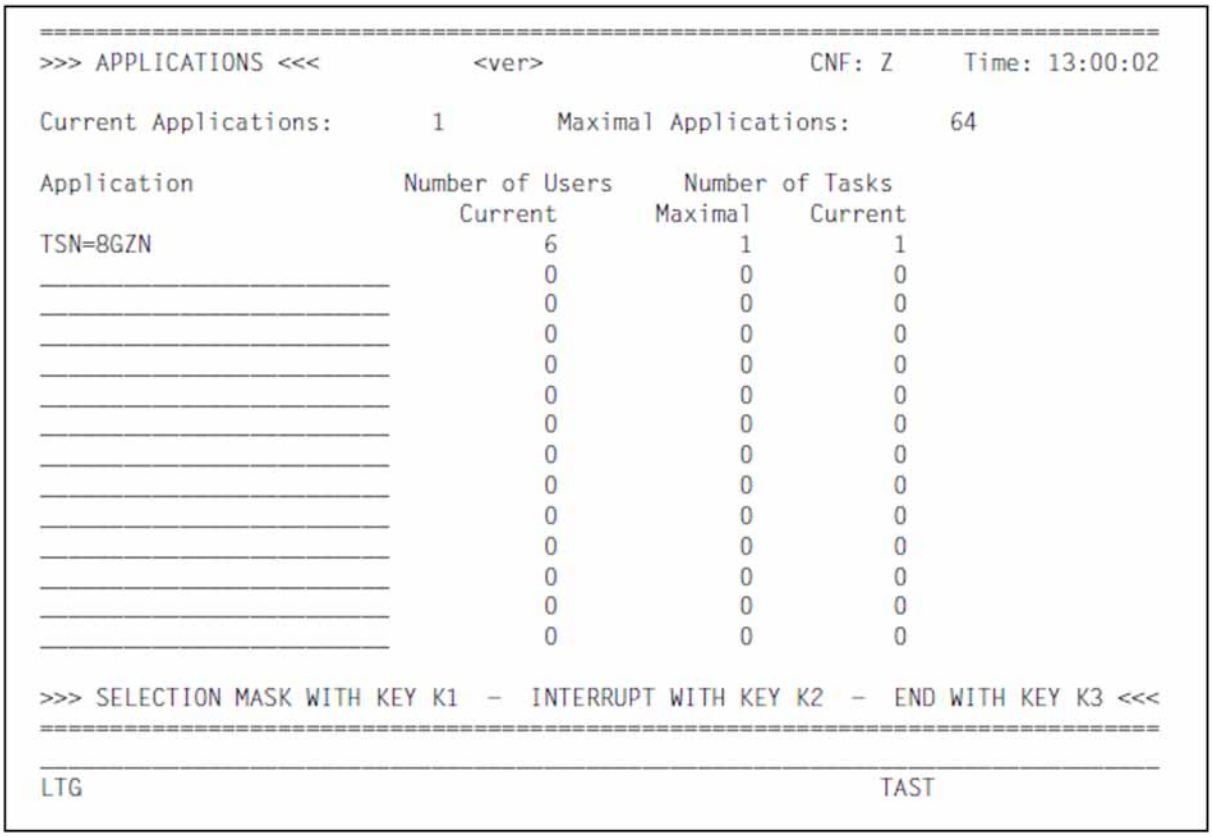
The quality of the buffer parametrization is calculated by means of the following formula:

Average hit rate when memory space is requested *
maximum percentage of the buffer that can be occupied

The form merely indicates whether the requested memory area is made available. It does not say anything about the use of the memory area.

7.4.3 SESDCN form APPLICATIONS

This form provides information on distributed transaction applications.



```
================================================================================
>>> APPLICATIONS <<<          <ver>                  CNF: Z     Time: 13:00:02

Current Applications:       1        Maximal Applications:       64

Application              Number of Users     Number of Tasks
                            Current       Maximal    Current
TSN=8GZN                          6             1          1
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0
________                          0             0          0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
LTG                                                              TAST
```

Field descriptions

<ver>	SESDCN version.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Current Applications

Number of transaction applications currently active.

Maximal Applications

Maximum possible number of transaction applications.

Application

Name of the transaction application to which the numbers output in this line apply (eight characters). Only those transaction applications that are currently active are listed.

Number of Users Current

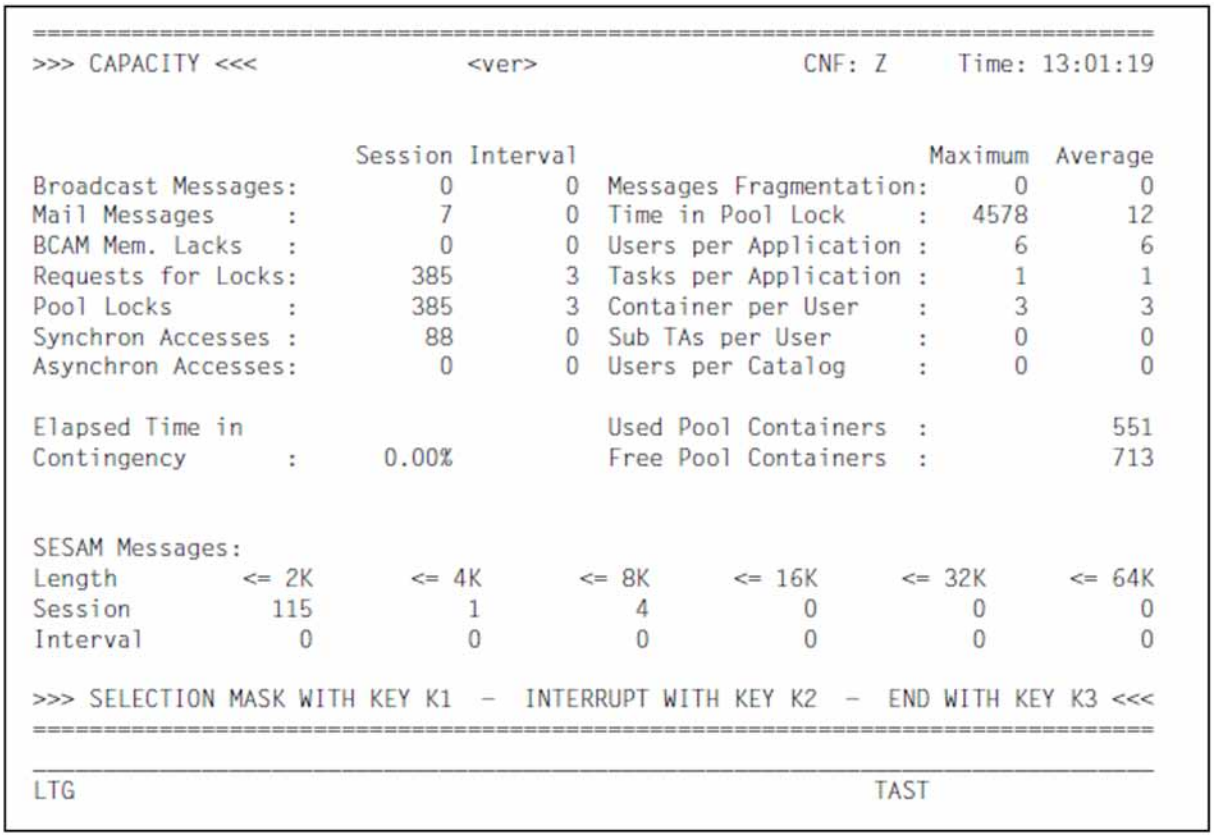
Current number of active users (in the sense of the DCN option USERS) of the specified transaction application. The output is sorted in descending order according to the number of active users. A maximum of 14 transaction applications are output.

Number of Tasks Maximal/Current

Maximum or current number of tasks of this transaction application.

7.4.4 SESDCN form CAPACITY

This form contains information on message volume, pool utilization and resource bottlenecks.



```
================================================================================
>>> CAPACITY <<<                <ver>                    CNF: Z     Time: 13:01:19


                       Session Interval                          Maximum  Average
Broadcast Messages:          0        0  Messages Fragmentation:      0        0
Mail Messages     :          7        0  Time in Pool Lock     :   4578       12
BCAM Mem. Lacks   :          0        0  Users per Application :      6        6
Requests for Locks:        385        3  Tasks per Application :      1        1
Pool Locks        :        385        3  Container per User    :      3        3
Synchron Accesses :         88        0  Sub TAs per User      :      0        0
Asynchron Accesses:          0        0  Users per Catalog     :      0        0

Elapsed Time in                          Used Pool Containers  :             551
Contingency       :      0.00%           Free Pool Containers  :             713


SESAM Messages:
Length         <= 2K       <= 4K       <= 8K      <= 16K      <= 32K      <= 64K
Session          115           1           4           0           0           0
Interval           0           0           0           0           0           0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
________________________________________________________________________________
LTG                                                          TAST
```

Field descriptions

<ver> SESDCN version.

CNF Name of the configuration monitored.

Time

Time at which the form is output.

Broadcast Messages

Number of messages sent additionally because of preparation to commit a transaction, the end of a transaction, the end of a UTM conversation or the rolling back of a transaction (in this SESDCN session and per time period).

Mail Messages

Number of messages that could not be sent until later because processing was interrupted (in this SESDCN session and per time period).

BCAM Memory Lacks

Number of remote messages that could not be transferred because not enough memory was available for them (in this SESDCN session and per time period).

Requests for Locks

Number of requested pool locks (in this SESDCN session and per time period).

Pool Locks

Number of granted pool locks (in this SESDCN session and per time period).

Synchron Accesses

Number of synchronous database accesses (in this SESDCN session and per time period).

Asynchron Accesses

Number of asynchronous accesses at the CALL DML interface by means of CALL SESPUT and CALL SESGET(W) (in this SESDCN session and per time period).

Message Fragmentation

Number of message fragments into which the DCAM messages sent so far had to be subdivided (maximum and average over the whole SESDCN session). Messages have to be divided up if they are longer than the maximum send length for the addressed computer.

Time in Pool Lock

Number of milliseconds for which a transaction was locked in the case of a pool access (maximum and average over the whole SESDCN session).

Users per Application

Maximum number of current users of a transaction application and average number over the whole SESDCN session.

Tasks per Application

Maximum number of current tasks belonging to a transaction application and average number over the whole SESDCN session.

Container per User

Maximum number of pool containers (pool container: 256-byte unit) used by a user in this session and average number of pool containers used by a user (snapshot).

Sub TAs per User

Maximum number of subtransactions per user in this session and average number of subtransactions per user (snapshot).

Users per Catalog

Maximum current number of users accessing a database and average number over the session.

Elapsed Time in Contingency

Specifies how much time SESDCN has so far spent on asynchronous activities. The value is a percentage of total SESDCN runtime.

Used Pool Containers

Number of containers in the SESDCN pool that are currently used.

Free Pool Containers

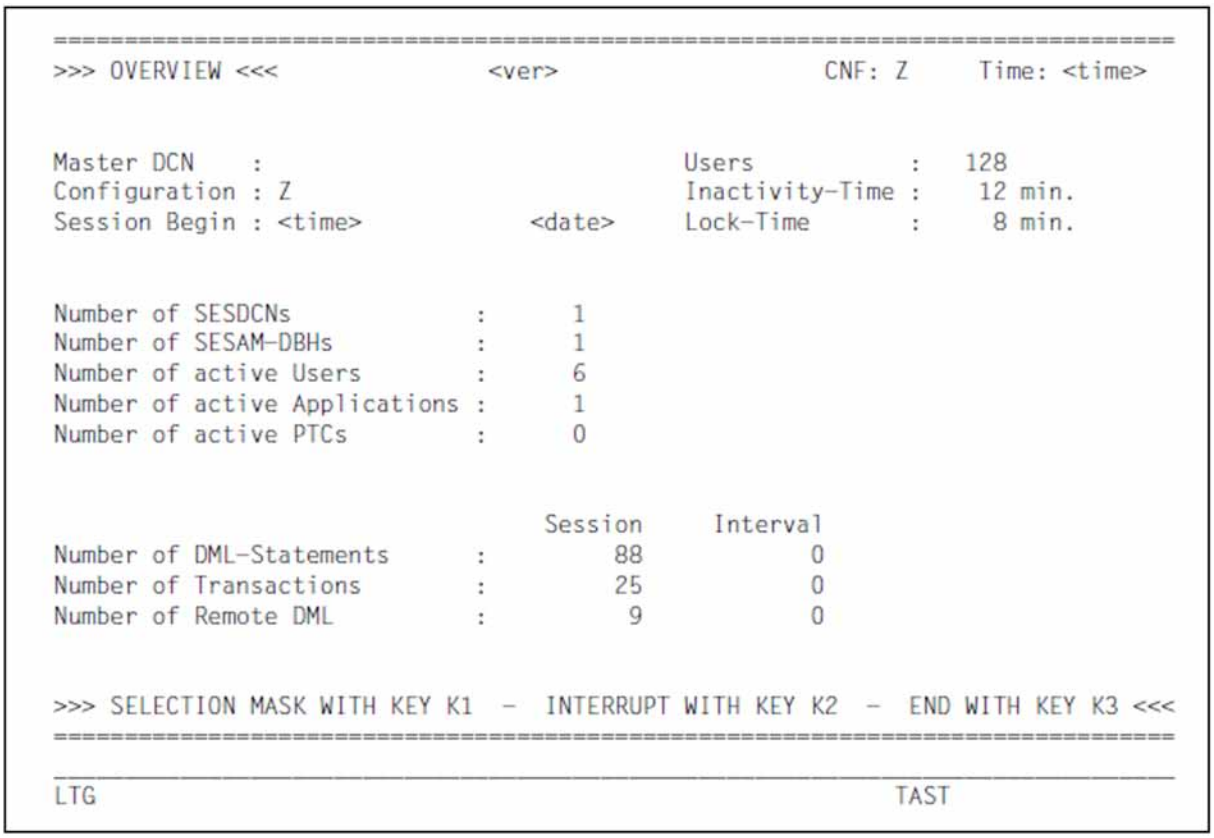
Number of containers that are still free.

SESAM Messages

Number of SESAM/SQL messages (remote and local) in this SESDCN session and per time period. The SESAM/SQL messages are divided into six different classes. The counter of the class to which the message belongs on the basis of its length is incremented.

7.4.5 SESDCN form OVERVIEW

The form contains general information on SESDCN operation, such as the set DCN options and values that apply to the whole SESDCN session.



Field descriptions

<ver>	SESDCN version.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Master DCN

Name of the master DCN that opened the SESDCN pool.

Configuration

Name of the configuration monitored.

Session Begin

Date and time the SESDCN session opened on the first SESDCN (master DCN) being started.

Users

Value for the USERS parameter of the DCN option SYSTEM-LIMITS of the master DCN. USERS specifies the maximum number of users.

Inactivity-Time

(field name in the case of output to SYSLST: NATL)

Value for the INACTIVITY-TIME parameter of the DCN option SYSTEM-LIMITS of the master DCN. The inactivity time is a reset criterion for transactions.

Lock-Time

(field name in the case of output to SYSLST: LCTL)
Value for the LOCK-TIME parameter of the DCN option SYSTEM-LIMITS of the master DCN. This value is a reset criterion for transactions.

Number of SESDCNs

Current number of SESDCNs loaded on the local computer and belonging to the selected configuration.

Number of SESAM-DBHs

Current number of DBHs belonging to the selected configuration on the local computer and participating in distributed processing.

Number of active Users

Number of currently active users.

Number of active Applications

Number of transaction applications currently active.

Number of active PTCs

Number of subtransactions currently in the PTC state. The number of DBHs on which the transaction is active is counted for each transaction.

Number of DML Statements

Specifies how many DML statements (CALL DML and SQL DML) have been entered in this SESDCN session /time period.

Number of Transactions

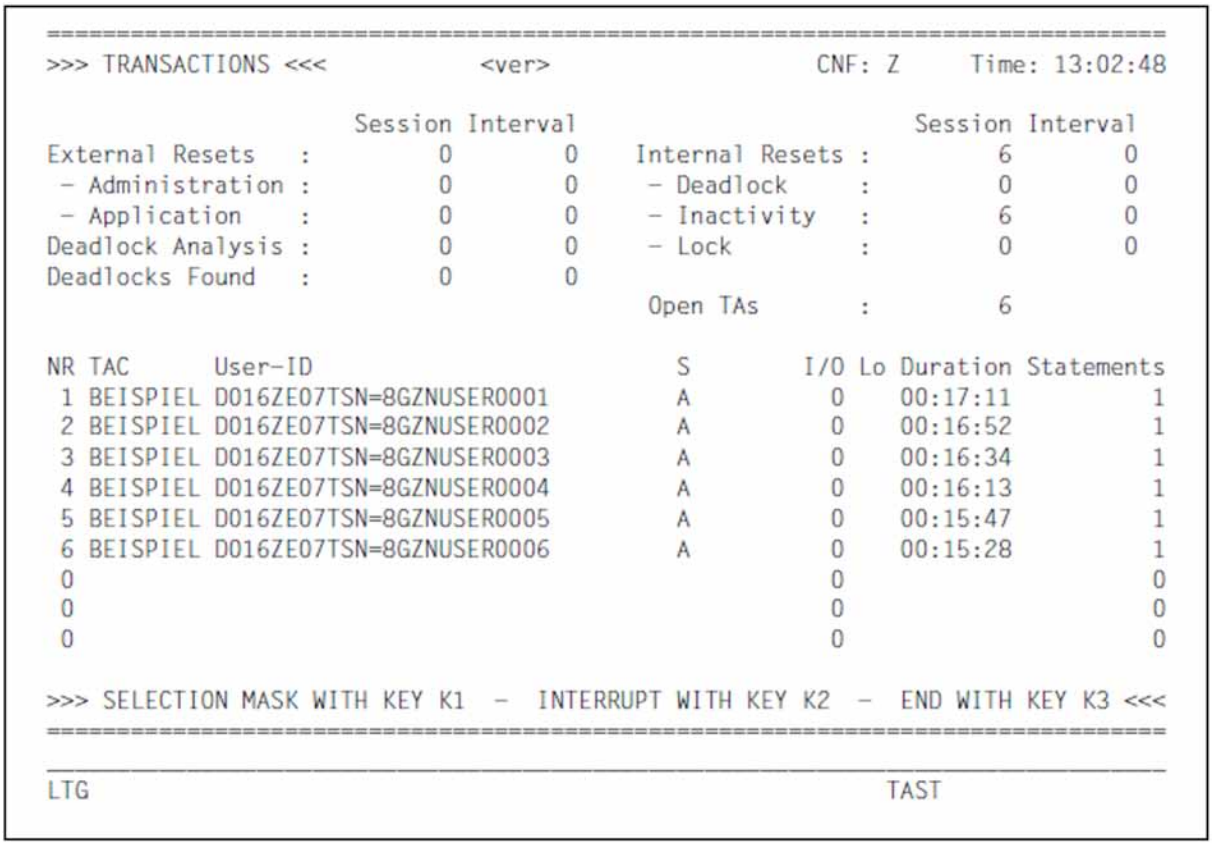
Specifies how many transactions have been started in this SESDCN session/time period.

Number of Remote DML

Specifies the number of accesses of databases in other configurations in this SESDCN session/time period.

7.4.6 SESDCN form TRANSACTIONS

This form provides information on the state and behavior of transactions.



```
================================================================================
>>> TRANSACTIONS <<<          <ver>                  CNF: Z     Time: 13:02:48

                      Session Interval                       Session Interval
External Resets   :         0        0    Internal Resets :         6        0
 - Administration :         0        0     - Deadlock     :         0        0
 - Application    :         0        0     - Inactivity   :         6        0
Deadlock Analysis :         0        0     - Lock         :         0        0
Deadlocks Found   :         0        0
                                          Open TAs       :         6

NR TAC      User-ID                          S        I/O Lo Duration Statements
 1 BEISPIEL D016ZE07TSN=8GZNUSER0001         A          0    00:17:11          1
 2 BEISPIEL D016ZE07TSN=8GZNUSER0002         A          0    00:16:52          1
 3 BEISPIEL D016ZE07TSN=8GZNUSER0003         A          0    00:16:34          1
 4 BEISPIEL D016ZE07TSN=8GZNUSER0004         A          0    00:16:13          1
 5 BEISPIEL D016ZE07TSN=8GZNUSER0005         A          0    00:15:47          1
 6 BEISPIEL D016ZE07TSN=8GZNUSER0006         A          0    00:15:28          1
 0                                                      0                      0
 0                                                      0                      0
 0                                                      0                      0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
LTG                                                              TAST
```

Field descriptions

<ver>	SESDCN version.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

External Resets

Number of transactions rolled back by means of administration commands or statements or a statement in the application program (in this SESDCN session and per time period)

Administration

Number of transactions rolled back by means of administration commands or statements (in this SESDCN session and per time period)

Application

Number of transactions rolled back by means of a statement in the application program (in this SESDCN session and per time period)

Deadlock Analysis

Number of deadlock analyses carried out (in this SESDCN session and per time period)

Deadlocks Found

Number of deadlock analyses in which deadlocks were detected (in this SESDCN session and per time period)

Internal Resets

Number of transactions rolled back for internal reasons (in this SESDCN session and per time period). Internal reasons for this are deadlock or the inactivity time (“Inactivity”) or lock time (“Lock”) being exceeded.

Deadlock

Number of transactions rolled back because of deadlock (in this SESDCN session and per time period)

Inactivity

(field name in the case of output to SYSLST: NATL)
Number of transactions rolled back because the inactivity time has been exceeded (in this SESDCN session and per time period). You set the inactivity time by means of the INACTIVITY-TIME parameter of the DCN option SYSTEM-LIMITS.

Lock

(field name in the case of output to SYSLST: LCTL)
Number of transactions rolled back because the lock time has been exceeded (in this SESDCN session and per time period). You set the lock time by means of the LOCK-TIME parameter of the DCN option SYSTEM-LIMITS.

Open TAs

Number of currently open transactions

The following information relates to individual transactions. The transactions are sorted in descending order according to their duration. A maximum of nine transactions are output.

NR

Unique (two-digit) transaction number assigned by SESMON.

TAC

UTM transaction code used to call the application program that opened this transaction (eight characters).
In the case of TIAM applications, the user ID of the application program appears here, in the case of DCAM applications the request name.

User-ID

Identification of the user who opened the transaction (32 characters). It is output in 32 positions in the following form: HOST-NAME, APPLICATION-NAME, CUSTOMER-NAME, CONVERSATION-ID (see "Syntax for the identification of users")

S (State)

State of the transaction (one character):
A: active
P: Prepare-To-Commit
L: locked

I/O

Weighted sum of the physical inputs and outputs of this transaction. The weighted sum is obtained by counting write accesses twice and read accesses once.

Lo (Lock)

Number of the transaction locking this transaction (2 digits). If the output transaction is locked by a transaction that can no longer be displayed on the screen, this is indicated by two asterisks (**).

Duration

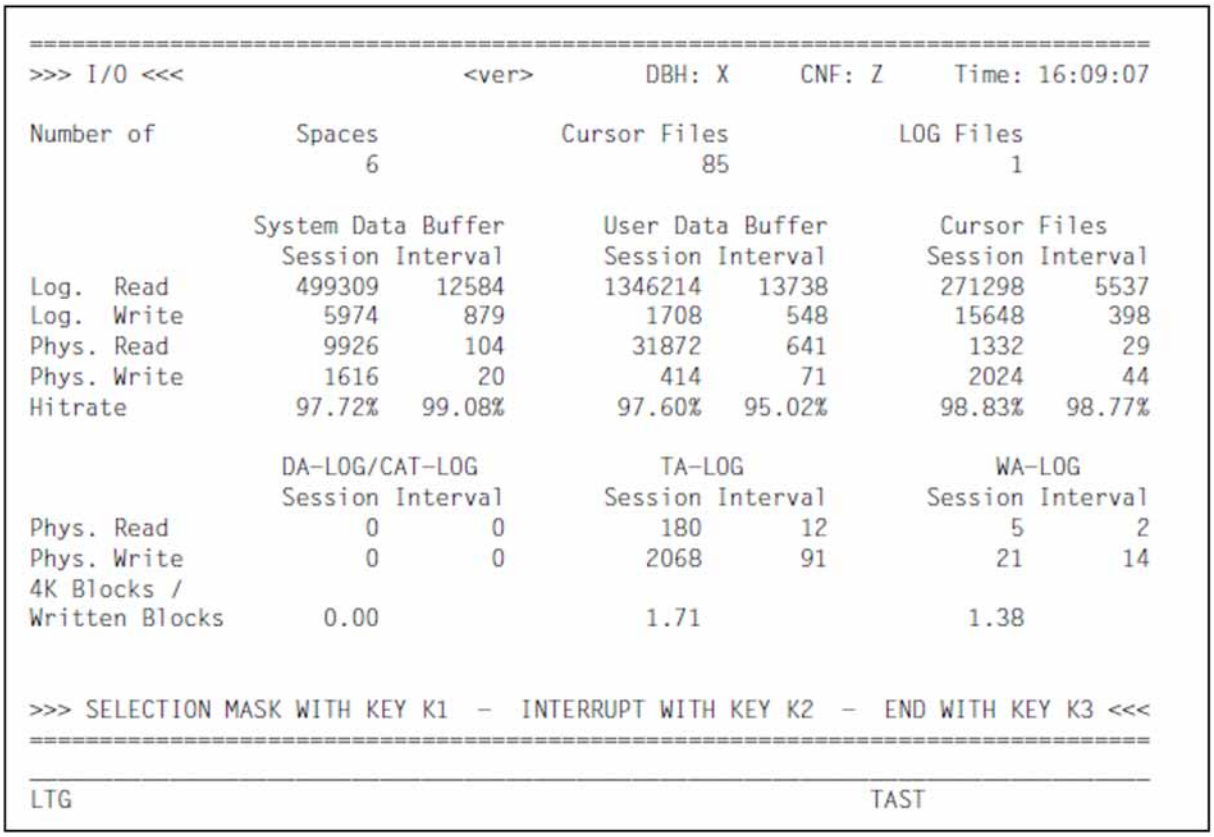
Elapsed time of the transaction in the form *hh*:*mm*:*ss*.

Statements

Number of statements of this transaction that have been executed. Only those statements that have not been processed completely by the SQL runtime system are counted. In cases where CALL DML statements are chained, each individual statement is counted.

7.4.7 DBH form I/O

This form contains information on logical and physical read and write accesses.



```
================================================================================
>>> I/O <<<                    <ver>        DBH: X     CNF: Z     Time: 16:09:07

Number of          Spaces             Cursor Files            LOG Files
                        6                       85                    1

                System Data Buffer       User Data Buffer        Cursor Files
                  Session Interval       Session Interval       Session Interval
Log.  Read         499309    12584       1346214    13738        271298     5537
Log.  Write          5974      879          1708      548         15648      398
Phys. Read           9926      104         31872      641          1332       29
Phys. Write          1616       20           414       71          2024       44
Hitrate            97.72%   99.08%        97.60%   95.02%        98.83%   98.77%

                  DA-LOG/CAT-LOG             TA-LOG                 WA-LOG
                  Session Interval       Session Interval       Session Interval
Phys. Read              0        0           180       12             5        2
Phys. Write             0        0          2068       91            21       14
4K Blocks /
Written Blocks       0.00                   1.71                   1.38


>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
_______________________________________________________________________________
LTG                                                          TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Number of Spaces

Number of catalog and user spaces.

Number of Cursor Files

Number of logical cursors output.

Number of LOG Files

Current number of DA-LOG and CAT-LOG files.

The following outputs apply only to the DBH tasks, not to the service tasks.

Log. Read

Total number of logical read accesses of the spaces or the cursor files (for the session and per time period).

Log. Write

Total number of logical write accesses of the spaces or the cursor files (for the session and per time period).

Phys. Read

Number of physical read accesses of the spaces or the cursor files (for the session and per time period).

Phys. Write

Number of physical write accesses of the spaces and cursor files (for the session and per time period).

Hitrate

Number of logical file accesses of the spaces and cursor files as a percentage of total accesses (for the session and per time period).

Phys. Read

Number of physical read accesses of the LOG files, i.e. DA-LOG with CAT-LOG, TA-LOGs and WA-LOG (for the session and per time period).

Phys. Write

Number of physical write accesses of the LOG files, i.e. DA-LOG with CAT-LOG, TA-LOGs and WA-LOG (for the session and per time period)

4K Blocks / Written Blocks

Specifies the average number of 4K blocks written during a write operation. The value is determined as of the beginning of the session for the LOG files.

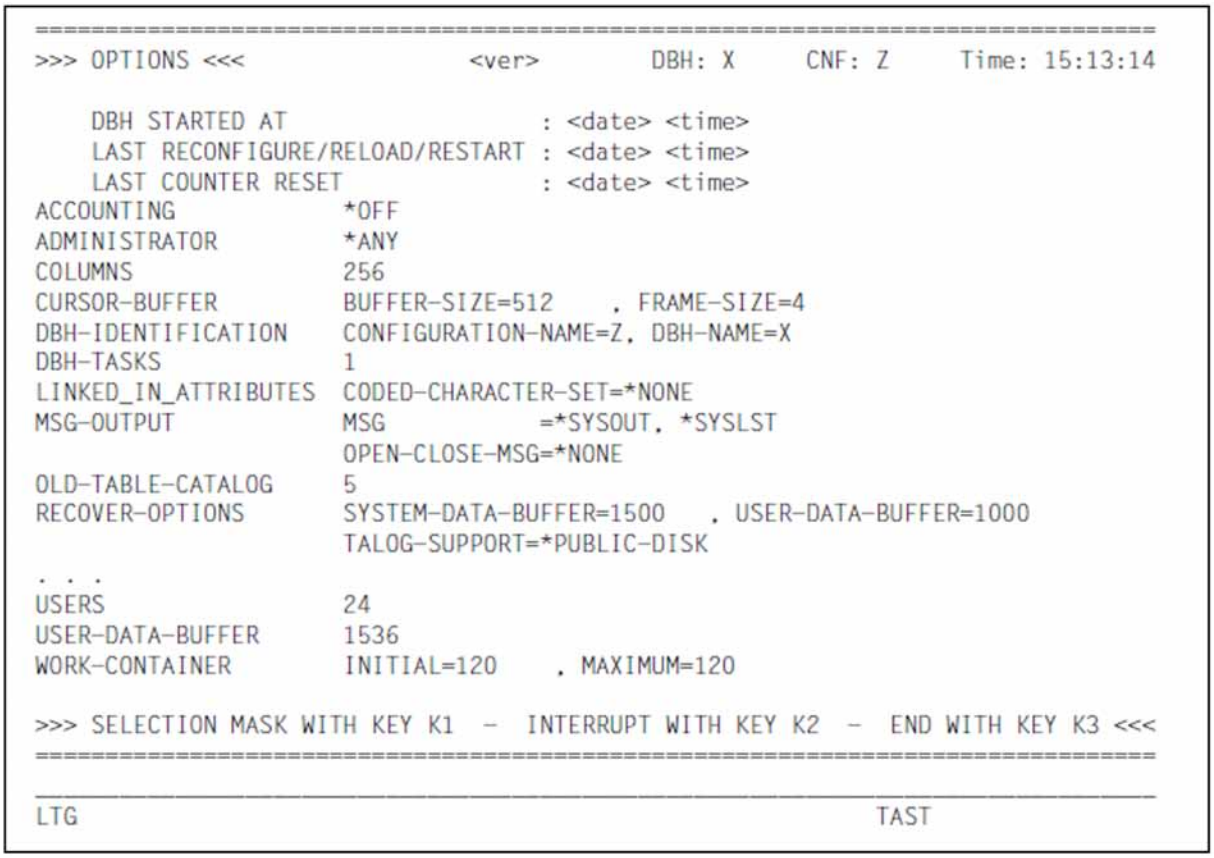
> These outputs are affected by counter overflows in a particular way. DBH counters which are incremented over a long period (e.g. Log. Read) can overflow and start again at 0. If SESMON finds this, no further hit rates are issued. * is output in their place.
>
> You can reset counters to zero in the SELECTION form, see "Overview of the statistical output of SESMON" and "The layout of the SELECTION form".

In output to SYSLST you will find additional information in the DBH form “I/O” (see "Layout of output to SYSLST"). The read and write behavior is indicated there for each file.

7.4.8 DBH form OPTIONS

This form displays the currently set DBH options and provides information on the DBH start time.

The output can extend to three screen pages. The page is changed when the refresh time specified in the SELECTION form for the output medium has elapsed.



```
================================================================================
>>> OPTIONS <<<                 <ver>        DBH: X     CNF: Z     Time: 15:13:14

    DBH STARTED AT                    : <date> <time>
    LAST RECONFIGURE/RELOAD/RESTART   : <date> <time>
    LAST COUNTER RESET                : <date> <time>
ACCOUNTING              *OFF
ADMINISTRATOR           *ANY
COLUMNS                 256
CURSOR-BUFFER           BUFFER-SIZE=512     , FRAME-SIZE=4
DBH-IDENTIFICATION      CONFIGURATION-NAME=Z, DBH-NAME=X
DBH-TASKS               1
LINKED_IN_ATTRIBUTES    CODED-CHARACTER-SET=*NONE
MSG-OUTPUT              MSG             =*SYSOUT, *SYSLST
                        OPEN-CLOSE-MSG=*NONE
OLD-TABLE-CATALOG       5
RECOVER-OPTIONS         SYSTEM-DATA-BUFFER=1500    , USER-DATA-BUFFER=1000
                        TALOG-SUPPORT=*PUBLIC-DISK
. . .
USERS                   24
USER-DATA-BUFFER        1536
WORK-CONTAINER          INITIAL=120     , MAXIMUM=120

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
--------------------------------------------------------------------------------
LTG                                                                      TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

DBH STARTED AT

Start time of the DBH (cold start and external restart)

LAST RECONFIGURE/RELOAD/RESTART

Time of the last DBH statement RECONFIGURE-DBH-SESSION or RELOAD-DBH-SESSION or of the last internal DBH restart.

LAST COUNTER RESET

Time the counters for form output were reset.

7.4.9 DBH form SERVICE ORDERS

The form shows the DDL or utility statements which are currently being processed in the service tasks, as well as the identification of the user for each request. If a ALTER PARTITIONING FOR TABLE, COPY, EXPORT, RECOVER, REFRESH, REORG SPACE or UNLOAD service request exists, the progress of this request is shown.

The Service Orders form describes each request in two lines:

- the first line describes the request
- the second line displays information on the progress of the request, if any is available

The requests are sorted in two levels based on:

- The status of the request (S field) in the following order
  1. Status “W” (waiting)
  2. Status “A” (active)
  3. Status “F” (finished)
- Within a status, the entries are sorted in descending order based on their duration.

A maximum of 5 requests are output to the screen, and a maximum of 100 to SYSLST.

```
================================================================================
>>> SERVICE ORDERS <<<         <ver>        DBH: 8     CNF: K     Time: <time>

TSN  User-ID                          Order                          S  Duration
             Information                                  Current      Total Done

7WP2 D016ZE09TSN=7V2ESESUTI           EXPORT TABLE                   A  00:00:04
     Table: PTJDB                                         153999     206991  74%

7WMA D016ZE09TSN=7V3ASESUTI           REORG SPACE:FIRST PHASE        A  00:00:03
     Space: $SQLDB111.CME.PART2                              680      13992   4%

7WP7 D016ZE09TSN=7V2ESESUTI           LOAD OFFLINE: PROCESS INPUT    A  00:00:03
     Table: TABX                                           43358          0   0%

7WPA D016ZE09TSN=0NKLSESUTI           COPY                           A  00:00:02
     Database: DCCAT09            Spaces:  5   DISC       10048      24000  41%

     D016ZE09TSN=7V2DDSQL             CREATE INDEX                   F  00:00:00
                                                                0          0   0%

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
```

Field descriptions

First display line (for all requests)

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.

Time Time at which the form is output.

TSN TSN of the service task currently processing the request. You can use this specification to find the associated entry on the DBH form SERVICE TASKS.

If the request is not yet processed or is already processed, this field is empty.

User-ID

Identification of the user (32 positions). Comprises HOST-NAME, APPLICATION-NAME, CUSTOMER-NAME and CONVERSATION-ID (see section “Syntax for the identification of users”).

Order

Type of request (30 positions)

If sort requests are processed in the service task when processing DML statements, “SORT” is output.

Otherwise the statement that the user issued or in the context of which the service task runs is shown.

Possible statements here are:

ALTER DATA

ALTER PARTITIONING

ALTER SPACE

ALTER TABLE

CHECK CONSTRAINTS

CHECK FORMAL

COPY

CREATE CATALOG

CREATE INDEX

CREATE REPLICATION

CREATE SPACE

CREATE TABLE

DROP INDEX

DROP SPACE

DROP TABLE

EXPORT TABLE

IMPORT TABLE: `<import-step>`

where `<import-step>` can have the following values:
PROCESS INPUT (process input file)
GENERATE INDEX (create index)

LOAD OFFLINE: `<load-step>`

where <load-step> can have the following values:

PROCESS INPUT (process input file)
GENERATE INDEX (create index)

MIGRATE

RECOVER: <recover-step>

where <recover-step> can have the following values:
"BLANK" (introduces request)
RESTORE CATALOG_SPACE (reading in the catalog space)
RESTORE USER-SPACES (reading in the user spaces)
RESTORE FROM REPL (catalog space and user spaces are read in here in a service task)
REPAIR CATALOG_SPACE (applying the changes logged in the CAT-LOG file)
REPAIR USER-SPACES (applying the changes logged in the DA-LOG files)
GENERATE INDEX (setting up the indexes)
DATA ADAPTION (data matching of catalog space and user spaces)

RECOVER INDEX

REFRESH REPLICATION

REFRESH SPACE

REORG SPACE: <reorg-step>

where <reorg-step> can have the following values:
"BLANK" (introduces request)
FIRST PHASE (reorganization)
RENAME (renaming)
COPY (copying)

REORG STATISTICS

UNLOAD

S Status of the request (one position)

W (Waiting): The request is not yet processed by a service task.

A (Active): The request is currently being processed by a service task.

F (Finished): The request has been processed by the service task and can be retrieved from the DBH.

Duration

Specifies the duration thus far of the request with respect to the current status, i.e. Duration specifies how long a request has already had the status “W”, “A” or “F”.

Second display line

This display line is only displayed for particular requests.

Second display line

This display line is only displayed for particular requests.

INFORMATION

Depending on the type of request (content of the ORDER field in the first display line), the field contains the object which is currently being processed:

ORDER	INFORMATION
ALTER PARTITIONING: MOVE ROWS	Table name
ALTER PARTITIONING: GEN INDEX	Table name
ALTER PARTITIONING: CHG ROWIDS	Table name
EXPORT TABLE	Table name
IMPORT TABLE: PROCESS INPUT	Table name
IMPORT TABLE: GENERATE INDEX	Table name
LOAD OFFLINE: PROCESS INPUT FILE	Table name
LOAD OFFLINE: GENERATE INDEX	Table name
UNLOAD	Table name
RECOVER: RESTORE CATALOG_SPACE	Database name and type of backup (DISC, HSMS, ARCH)
COPY	Database name, number of spaces and type of backup
RECOVER: RESTORE USER SPACES	Database name, number of spaces and type of backup
RECOVER: RESTORE FROM REPL	Database name, number of spaces and type of backup
RECOVER: REPAIR CATALOG_SPACE	Name of the current logging file
RECOVER: REPAIR USER SPACES	Name of the current logging file
REFRESH REPLICATION: REPAIR USER SPACES	Name of the current logging file
REFRESH SPACE: REPAIR USER SPACES	Name of the current logging file
REORG	Name of the space
REORG SPACE: FIRST PHASE	Name of the space
REORG SPACE: RENAME	Name of the space
REORG SPACE: COPY	Name of the space

CURRENT

Number of blocks or records (up to 10 characters) processed so far.

In the case of ALTER PARTITIONING: MOVE ROWS / CHG ROWID, EXPORT TABLE, UNLOAD TABLE, IMPORT TABLE: PROCESS INPUT and LOAD OFFLINE: PROCESS INPUT FILE the number of records is contained here, otherwise the number of blocks.

In the case of IMPORT TABLE: GENERATE INDEX and LOAD OFFLINE:
GENERATE INDEX the value “0” is displayed here because in this phase the access data is maintained and no meaningful value can be displayed.

TOTAL

Total number of blocks or records (up to 10 characters) to be processed.

In the case of EXPORT TABLE and UNLOAD TABLE the number of records is contained here, otherwise the number of blocks.

In the case of IMPORT TABLE: PROCESS INPUT, IMPORT TABLE: GENERATE INDEX, LOAD OFFLINE: PROCESS INPUT and LOAD OFFLINE: GENERATE INDEX the value “0” is displayed here because the number of records in the input file is not known.

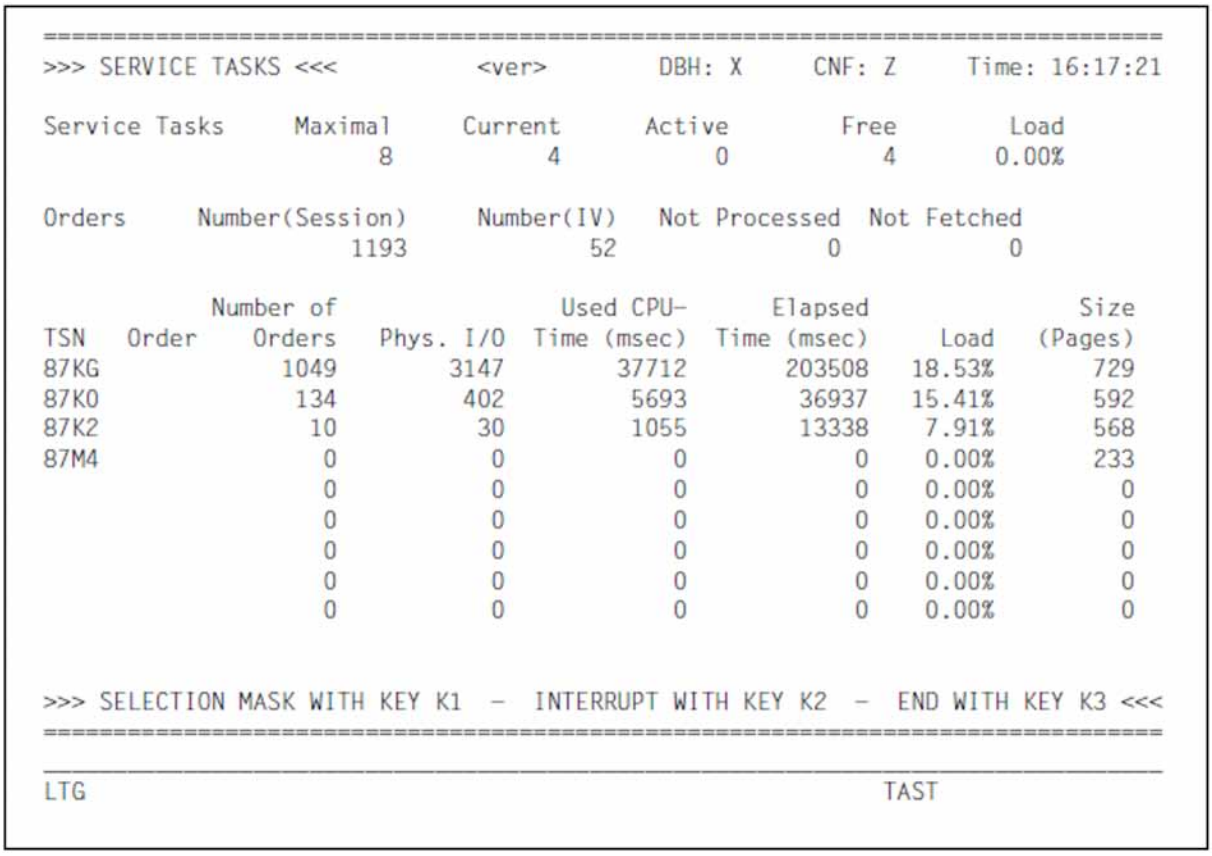
DONE

Processing progress: Relationship of “CURRENT” to “TOTAL” in percent.

In the case of ALTER PARTITIONING, IMPORT TABLE: PROCESS INPUT, IMPORT TABLE: GENERATE INDEX, LOAD OFFLINE: PROCESS INPUT and LOAD OFFLINE: GENERATE INDEX the value “0” is displayed here because the value for TOTAL is not known.

7.4.10 DBH form SERVICE TASKS

This form supplies values relating to the service tasks and the requests to be processed.



```
================================================================================
>>> SERVICE TASKS <<<          <ver>        DBH: X     CNF: Z     Time: 16:17:21

Service Tasks     Maximal     Current      Active        Free        Load
                        8           4           0           4       0.00%

Orders     Number(Session)     Number(IV)   Not Processed  Not Fetched
                      1193             52               0            0

            Number of                 Used CPU-      Elapsed                Size
TSN   Order     Orders   Phys. I/O  Time (msec)  Time (msec)     Load   (Pages)
87KG             1049        3147        37712       203508   18.53%       729
87K0              134         402         5693        36937   15.41%       592
87K2               10          30         1055        13338    7.91%       568
87M4                0           0            0            0    0.00%       233
                    0           0            0            0    0.00%         0
                    0           0            0            0    0.00%         0
                    0           0            0            0    0.00%         0
                    0           0            0            0    0.00%         0
                    0           0            0            0    0.00%         0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================
LTG                                                                      TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Maximal

Maximum possible number of service tasks set by means of the DBH option SERVICE-TASKS.

Current

Number of started service tasks.

Active

Number of currently active service tasks.

Free

Number of inactive service tasks that are currently free.

Load

Active tasks as a percentage of started service tasks.

Number (Session)

Number of requests for service tasks in this session.

Number (IV)

Number of requests for service tasks per time period.

Not Processed

Number of requests not yet processed by the service tasks.

Not Fetched

Number of requests processed by the service tasks but not yet fetched by the DBH.

The following outputs relate to individual service tasks sorted in descending order on the basis of the amount of CPU time they have used (“Used CPU-Time”). A maximum of nine service tasks are output.

TSN	TSN of the service task.
Order	Type of request currently being processed by the service task: SORT: sort request REP: recovery request UTI: utility statement

Number of Orders

Number of requests for service tasks so far processed (ten digits).

Phys. I/O

Number of physical inputs and outputs of the service task (ten digits).

Used CPU-Time (msec)

Total CPU time of the service task in milliseconds (ten digits). If the CPU time cannot be determined because, for example, the DBH and thus also the service tasks are running under a different ID than the monitor or the task is not known to the system any more, then “********” is output.

Elapsed Time (msec)

Total duration of the service task in milliseconds (ten digits).

Load

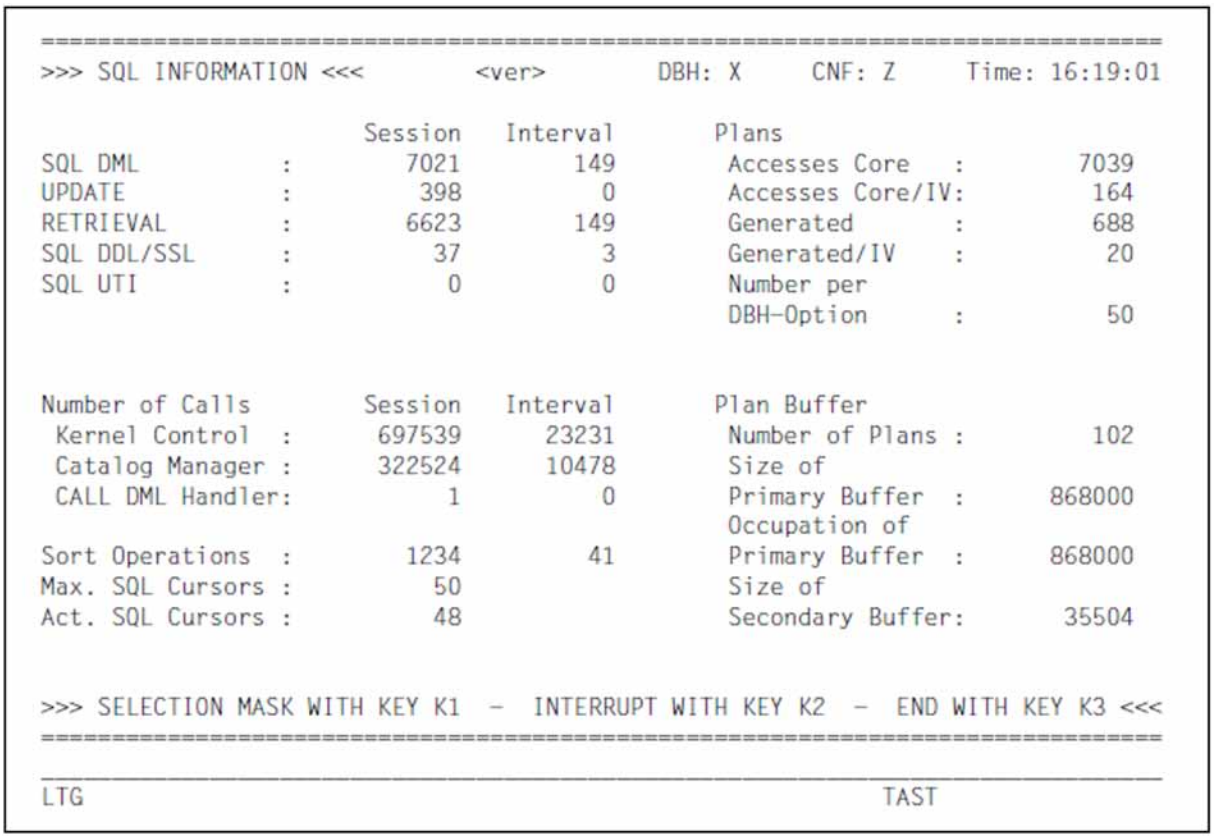
“Used CPU-Time” as a percentage of “Elapsed Time”

Size (Pages)

Memory space (in pages of 4 Kbytes) used by the service task.

7.4.11 DBH form SQL INFORMATION

This form displays information on SQL statements, SQL access plans, plan buffers and calls of specific SQL components.



Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

SQL DML

Number of SQL DML statements (in this session and per time period).

UPDATE

Number of SQL update statements (in this session and per time period).

RETRIEVAL

Number of SQL retrieval statements (in this session and per time period).

SQL DDL/SSL

Number of SQL DDL and SQL SSL statements (in this session and per time period).

SQL UTI

Number of utility statements (in this session and per time period).

Accesses Core

Number of accesses of SQL access plans during the whole session.

Accesses Core/IV

Number of accesses of SQL access plans per time period.

Generated

Number of SQL access plans generated in this session. SQL access plans are also regenerated when they have been displaced from the plan buffer.

Generated/IV

Number of SQL access plans generated per time period. SQL access plans are also regenerated when they have been displaced from the plan buffer.

Number per DBH-Option

Number of SQL access plans set by means of the DBH option SQL-SUPPORT.

Kernel Control

Number of kernel-control calls (in this session and per time period).

Catalog Manager

Number of catalog-manager calls (in this session and per time period).

CALL DML Handler

Number of CALL DML handler calls (in this session and per time period).

Sort Operations

Number of SQL sort operations (in this session and per time period).

Max. SQL Cursors

Maximum number of open SQL cursors in the session.

Act. SQL Cursors

Number of currently open SQL cursors (snapshot).

Number of Plans

Number of SQL access plans in the plan buffer (snapshot).

Size of Primary Buffer

Size of the primary plan buffer in bytes.

Occupation of Primary Buffer

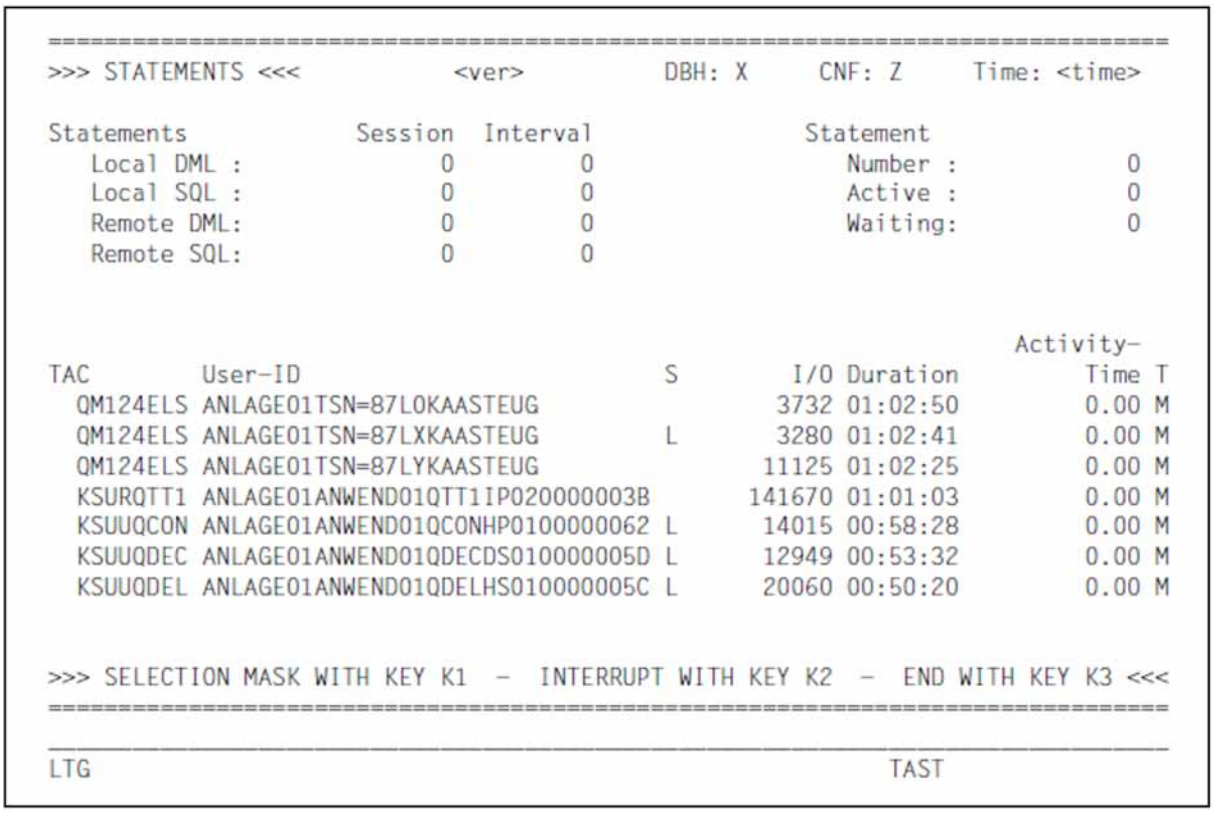
Space in bytes used in the primary plan buffer (snapshot).

Size of Secondary Buffer

Size of the secondary plan buffer in bytes.

7.4.12 DBH form STATEMENTS

The form provides an overview of all active statements. It displays statements both within transactions and those outside transactions.



Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Local DML

Number of local CALL DML statements (for the session and per time period).

Local SQL

Number of all local SQL statements, including utility statements (for the session and per time period).

Remote DML

Number of CALL DML statements entered from a different configuration (for the session and per time period).

Remote SQL

Number of all SQL statements, including utility statements, entered from another configuration (for the session and per time period).

Number

Current number of statements (actual value).
This value is the sum of the two subsequent fields Active and Waiting.

Active

Number of active statements (actual value), i.e. all statements marked “A” in column S (see below).

Waiting

Number of waiting statements (actual value), i.e. all statements with the status “L” and “W” in column S (see below).

Up to nine statements are output, sorted in descending order based on their duration. The following specifications refer to individual statements.

TAC

UTM transaction code used to call the application program that opened this transaction (eight characters).
In the case of TIAM applications, the user ID of the application program appears here, in the case of DCAM applications the request name.

User-ID

Identification of the user who opened the transaction (32 characters). It is output in 32 positions in the following form: HOST-NAME, APPLICATION-NAME, CUSTOMER-NAME, CONVERSATION-ID (see section “Syntax for the identification of users”).

S Status of the statements (one position)

- A: The statement is being processed by a thread.
- L: The statement ran into a transaction lock.
  (The lock relationships of the transactions can be taken from the DBH form TRANSACTIONS.)
- W: The statement is in the priority queue or is waiting for a service task.

I/O Number of weighted inputs and outputs of this statement (i.e. write accesses are counted twice, read accesses once).

Duration

Duration of the statement (elapsed time) in the format *hh*:*mm*:*ss*.

Activity-Time

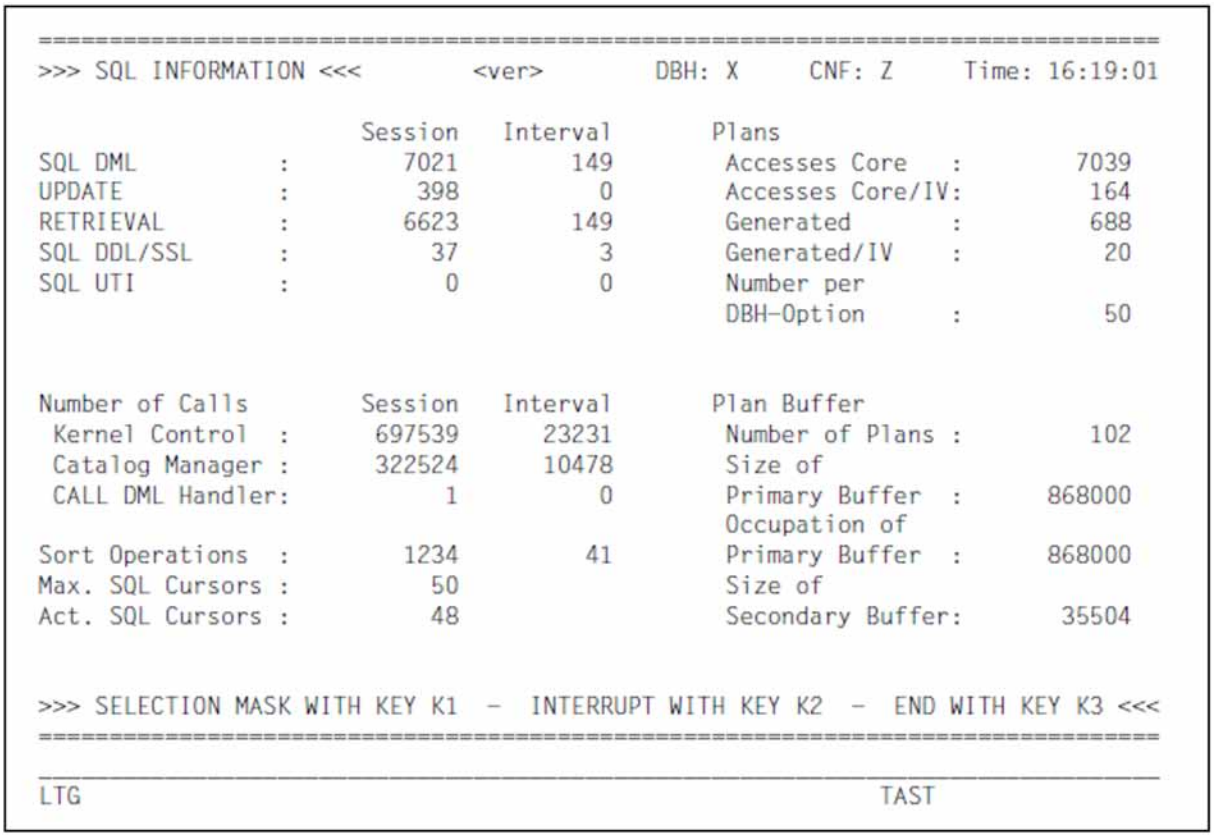
Elapsed activity time of the statement in the DBH in seconds.

T Application type

M:	DML statement
D:	DDL statement
U:	Utility statement
'BLANK' (space):	The statement type is not yet known.

7.4.13 DBH form SYSTEM INFORMATION

This form displays the current settings and values for DBH operation, in particular the size and occupancy of individual containers. You can also use this form to obtain information on the progress of a DBH session restart.



Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Active Suborders for CALL DML

Number of active suborders for CALL DML (logical files).

Active Suborders for SQL DML

Number of active suborders for SQL (SQL-DML).

Available Suborders Maximal

Maximum value for the number of concurrent suborders (DBH option SUBORDERS, MAXIMUM parameter).

Available Suborders Current

Initial value for the number of concurrent suborders (DBH option SUBORDERS, INITIAL parameter).

Used Suborders Maximal

Maximum number of used suborders in the session.

Used Suborders Current

Current number of used suborders. This value is the sum of the values for “Active Suborders for CALL DML” and “Active Suborders for SQL DML”.

Parallel Open Spaces

Current number of concurrently open spaces.

Open Cursors

Current number of open internal cursor files .

Accounting

ON/OFF specifies whether Accounting is currently switched on or off.

SESCOS

ON/OFF specifies whether SESCOS is currently switched on or off.

Request Control

ON/OFF specifies whether priority control is currently switched on or off.

Container Current Size

Current size of the work container resp. of the transfer container in the DBH session in Kbytes.

Container Current Percentage Occupation

Current percentage of the container occupied in relation to the current size.

Container Maximal Size

Maximum occupation for the container according to the DBH options.

Container Maximal Percentage Occupation

Maximum percentage of the container occupied in the DBH session in relation to the maximum value specified in the DBH options.

Threads Active

Number of currently active threads (threads that are in the “running” state). The number of active threads is smaller than or equal to the number of tasks.

Threads Ready

Current number of ready threads (threads that are in the “ready to run” state).

Threads Free

Number of currently free threads (threads that currently do not have a job).

Threads Waiting

Current number of threads that are not active due to a semaphore, block or termination of a system job.

Threads TA-Lock

Current number of threads that are not active due to a TA lock.

Threads Service Task

Current number of threads that are not active because they are waiting for a service task.

Threads I/O

Current number of threads that are not active due to I/O.

Valid Logging Info - Number of Blocks

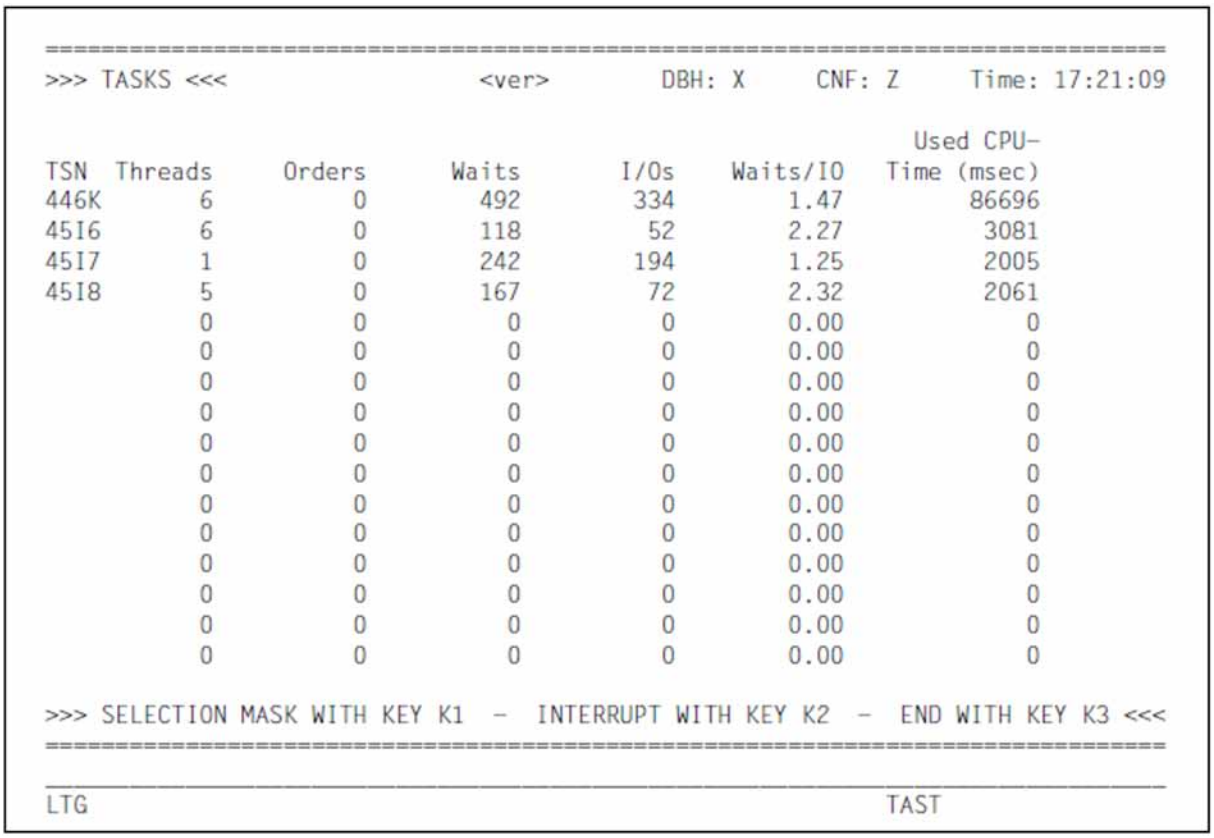
Number of logging blocks that have accrued for a restart (number of blocks with PBI info+number of blocks with LBI info+number of blocks with AI info). This value is displayed only in an ongoing DBH session. During a DBH session restart, 0 is displayed.

Restart Blocks Current: ... Total: ... Done: ...%

Number of processed logging blocks or total number of logging blocks to be processed (number of blocks with LBI info+number of blocks with AI info) and the processing progress (relation of “Current” to “Total” in percent). These values enable you to observe the progress of the restart. These values are displayed only while the DBH session is restarting. In an ongoing DBH session, zeroes are displayed.

7.4.14 DBH form TASKS

This mask provides extended statistical data for a multitasking system that can be used to determine the utilization of individual DBH tasks.



```
================================================================================
>>> TASKS <<<                  <ver>        DBH: X     CNF: Z     Time: 17:21:09

                                                                  Used CPU-
TSN  Threads     Orders      Waits       I/Os    Waits/IO   Time (msec)
446K       6          0        492        334        1.47        86696
45I6       6          0        118         52        2.27         3081
45I7       1          0        242        194        1.25         2005
45I8       5          0        167         72        2.32         2061
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0
           0          0          0          0        0.00            0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================

LTG                                                                  TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

In the table of individual tasks one line is output per task, resulting in a maximum of 16 lines. The tasks are always output in the same order, sorted according to the internal task ID, together with the following values:

TSN

Task TSN.

Threads

Number of threads currently assigned to this task.

Orders

Number of orders still in the communication queue for the task.

Waits

Number of wait states due to I/O in the interval.

I/Os

Number of I/Os triggered in the interval.

Waits/IO

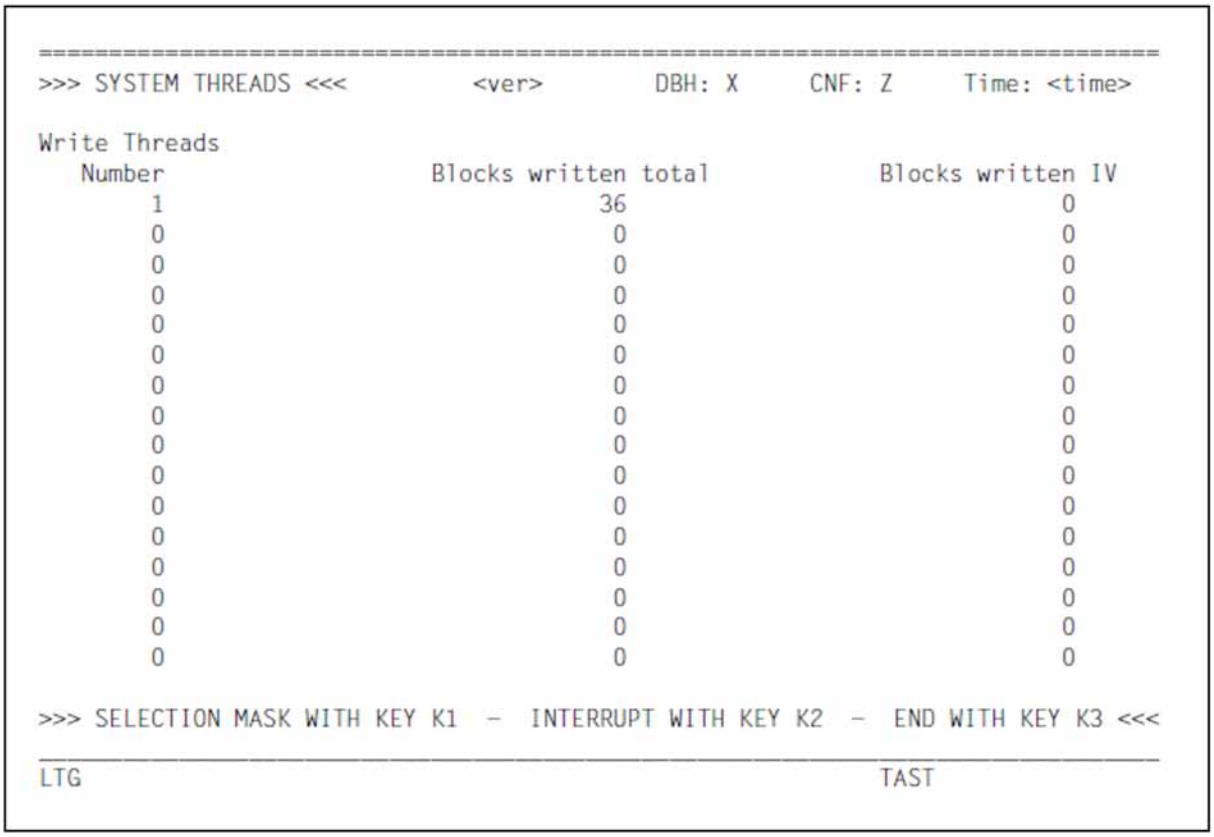
Ratio of the two previous values. Number of wait states due to I/O / Number of I/Os triggered.

Used CPU-Time (10 digits)

Total CPU time used by the corresponding task in milliseconds.
If the CPU time cannot be determined because the DBH is running under a different ID than the monitor or the task is not known to the system, for example, then “**********” is output.

7.4.15 DBH form SYSTEM THREADS

The form provides information on the currently active write threads.



```
================================================================================
>>> SYSTEM THREADS <<<          <ver>         DBH: X      CNF: Z      Time: <time>

Write Threads
   Number                    Blocks written total            Blocks written IV
        1                                      36                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0
        0                                       0                            0

>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
________________________________________________________________________________
LTG                                                                        TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	NAM identifier of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

The currently active write threads are displayed in this form. The maximum number of lines displayed corresponds to the number of write threads specified by means of the DBH option SYSTEM-THREADS (see "SYSTEM-THREADS"). One line is output in the table for each write thread, up to a maximum of 16 lines. If more than 15 write threads are created, the values are added together as of the 16th write thread.

Number

Sequence of the write thread.

Written Blocks (Session)

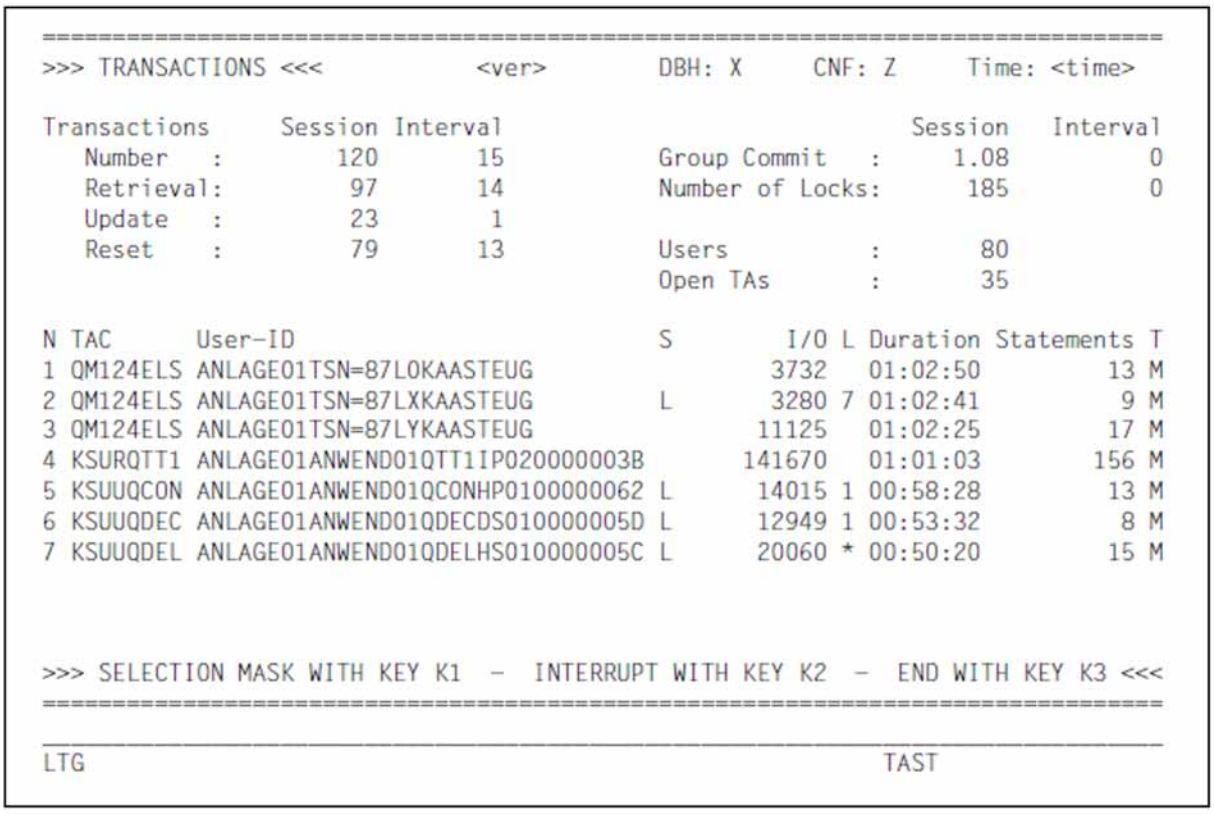
Number of blocks written physically to the spaces by this thread in the DBH session.

Written Blocks (IV)

Number of blocks written physically to the spaces in the last measurement period by this thread.

7.4.16 DBH form TRANSACTIONS

This form provides information on the number of transactions. The duration, type and number of I/Os is output for each transaction.



```
================================================================================
>>> TRANSACTIONS <<<          <ver>       DBH: X     CNF: Z     Time: <time>

Transactions     Session Interval                          Session   Interval
   Number   :        120       15          Group Commit   :    1.08          0
   Retrieval:         97       14          Number of Locks:    185          0
   Update   :         23        1
   Reset    :         79       13          Users          :     80
                                           Open TAs       :     35

N TAC      User-ID                          S        I/O L Duration Statements T
1 QM124ELS ANLAGE01TSN=87L0KAASTEUG                 3732   01:02:50         13 M
2 QM124ELS ANLAGE01TSN=87LXKAASTEUG          L      3280 7 01:02:41          9 M
3 QM124ELS ANLAGE01TSN=87LYKAASTEUG                11125   01:02:25         17 M
4 KSURQTT1 ANLAGE01ANWEND01QTT1IP020000003B        141670   01:01:03        156 M
5 KSUUQCON ANLAGE01ANWEND01QCONHP0100000062 L       14015 1 00:58:28         13 M
6 KSUUQDEC ANLAGE01ANWEND01QDECDS010000005D L       12949 1 00:53:32          8 M
7 KSUUQDEL ANLAGE01ANWEND01QDELHS010000005C L       20060 * 00:50:20         15 M



>>> SELECTION MASK WITH KEY K1  -  INTERRUPT WITH KEY K2  -  END WITH KEY K3 <<<
================================================================================

LTG                                                                      TAST
```

Field descriptions

<ver>	Version of the DBH.
DBH	Name of the DBH to which the data in the form relates.
CNF	Name of the configuration monitored.
Time	Time at which the form is output.

Number

Number of completed transactions (for the session and per time period), including internal transactions.

Retrieval

Number of completed retrieval transactions (for the session and per time period), including internal transactions.

Update

Number of completed update transactions (for the session and per time period).

Reset

Number of rolled-back transactions (for the session and per time period).

Group Commit / Session

Average number of transactions involved in a group commit (for the session).

Group Commit / Interval

Average number of transactions involved in a group commit within the last time interval.

Number of Locks / Session

Number of transaction locks for the session.

Number of Locks / Interval

Number of transaction locks per time interval.

Users

Number of currently active users.

Open TAs

Number of currently open transactions

The following information relates to individual transactions. A maximum of nine transactions are output. They are sorted in ascending order according to their duration. The maximum number of transactions to be output to SYSLST is determined by the user (range of values: 10-999, default value: 200).

N	Unique transaction number assigned by SESMON (one digit)
TAC	UTM transaction code used to call the application program that opened this transaction (eight characters). In the case of TIAM applications, the user ID of the application program appears here, in the case of DCAM applications the request name.

User-ID

Identification of the user who opened the transaction (32 characters). It is output in 32 positions in the following form: HOST-NAME, APPLICATION-NAME, CUSTOMER-NAME, CONVERSATION-ID (see section “Syntax for the identification of users”).

S (State)

State of the transaction (one character):

A:	The transaction is being processed by a thread.
L:	The transaction is locked.
P:	The transaction is in the PTC state.
R:	The transaction has been rolled back.
W:	The transaction has been canceled due to a service task.
F:	The transaction has been unlocked with INFORM-PROGRAM JOB-ID=*TSN(TSN=*tsn*), MSG='*component*,RELEASE-USER-RESOURCES' or the corresponding CALL-DML administration statement.
'BLANK' (space):	The state of the transaction does not correspond to any of the above categories.

I/O

Weighted sum of the logical inputs and outputs of this transaction. The weighted sum is obtained by counting write accesses twice and read accesses once.

L (Lock)

Number of the transaction that locks this transaction (one digit). If the output transaction is locked by a transaction that can no longer be displayed on the screen, this is indicated by “*” (asterisk).

Duration

Elapsed time of the transaction in the form *hh*:*mm*:*ss*

Statements

Number of statements executed so far in this transaction. Only those statements received by the DBH for processing are counted. Statements that have already been processed by the connection module or the SQL runtime system are not counted. Chained CALL DML statements are counted as one statement.

T (Type)

Transaction type
M: DML transaction
D: DDL transaction
U: internal utility transaction

7.5 Layout of output to a file

The SESMON output file is a SAM file with a block size of 32768 bytes (BLKSIZE=(STD,16)). The records have a variable format (RECFORM=V).

The file can contain the following record types:

- configuration records (record identifier: “KON”)
- option records (record identifier: “OPT”)
- CNF records (record identifier: “CNF”)
- DCN records (record identifier: “DCN”)
- DBH records (record identifier “DBH”).

The SESMON output file has the following structure:

```
KON-Satz
[OPT-Sätze] (soviele OPT-Sätze, wie DBHs in der Auswertung)
[CNF-Satz]
[DCN-Satz]
[DBH-Sätze] (max. 10)
[KON-Satz]    1.
[OPT-Sätze]   2.
[CNF-Satz]    3.
[DCN-Satz]    4.
[DBH-Sätze]   5.

[KON-Satz]    1.
[OPT-Sätze]   2.
[CNF-Satz]    3.
[DCN-Satz]    4.
[DBH-Sätze]   5.
...
```

1. The KON record is only output again if the contents of fields in the KON record have changed.
2. Only those OPT records in which the contents of fields have changed are output again.
3. Output after the currently set output time interval has elapsed.
4. Output after the currently set output time interval has elapsed.
5. Output after the currently set output time interval has elapsed.

7.5.1 Layout of the output records

In the description of the output records below, the “Field contents” column contains a brief description of each field output either in a form or to SYSLST. This description is largely identical to the name of the field in the relevant form or in the SYSLST log.

You will find detailed descriptions of the fields output in forms or to SYSLST in section “Layout of the forms” or, in the case of fields that are only output to SYSLST, in section “Layout of output to SYSLST”.

All distances refer to the beginning of the record length field. If the records are read in by means of C (fread), you receive them without the four-byte long record length field. In these cases, the distances to the beginning of the record are four bytes less than the values specified here.

7.5.1.1 Configuration record

Distance	Length	Field contents
Header:		
0	2	Length of the record
2	2	Empty
4	3	“KON” record ID (printable)
7	1	Filler
8	8	Time in STCK format
16	10	Date (*yyyy-mm-dd*), local time (printable)
26	8	Timestamp (*hh*:*mm*:*ss*), local time (printable)
34	10	Date (format *yyyy-mm-dd*), universal time (printable)
44	8	Time stamp (format *hh*:*mm*:*ss*), universal time (printable)
52	12	Filler for extensions
General values not output in a form:		
64	4	Valid output time interval
68	8	Hardware type (SINF macro) (printable)
76	12	Operating system name and version (SINF macro) (printable)
88	4	Main memory in bytes (SINF macro). In the event of an overflow (X'FFFFFFFF'), the field at distance 100 contains the correct value.
92	4	Number of CPUs (SINF macro)
96	4	Number of active CPUs (SINF macro)
100	4	Main memory in Megabyte (SINF macro)
104	12	Filler for extensions
116	4	HSI type (SINF macro) (printable)
120	8	DCAM processor name (printable)
128	8	DCN version (printable)
136	2	DCN corrections version status (printable)
138	22	Filler for extensions
Values for the SESDCN form OVERVIEW:		

160	1	Name of the master DCN (printable)
161	1	Configuration (printable)
162	10	Session Begin date (*yyyy-mm-dd*), local time (printable)
172	8	Session Begin timestamp (*hh*:*mm*:*ss*), local time (printable)
180	4	Users option
184	4	Inactivity-Time option
188	4	Lock-Time option

Table 32: Layout of the configuration record

7.5.1.2 Option record

Distance	Length	Field contents
Header:		
0	2	Length of the record
2	2	Empty
4	3	“OPT” record ID (printable)
7	1	NAM ID of the DBH to which the record refers (printable)
8	8	Time in STCK format
16	10	Date (*yyyy-mm-dd*), local time (printable)
26	8	Timestamp (*hh*:*mm*:*ss*), local time (printable)
34	10	Date (format *yyyy-mm-dd*), universal time (printable)
44	8	Time stamp (format *hh*:*mm*:*ss*), universal time (printable)
52	12	Filler for extensions
Current values:		
64	8	DBH version (printable)
72	2	Corrections version status of the DBH (printable)
74	6	Filler for extensions
80	n * 79 bytes	DBH options (currently up to n=57 lines of 79 characters each, printable)

Table 33: Layout of the option record

When output is via the SNMP subagent SESAM-MON, the (variable) number of lines required are displayed for the DBH options.

> The maximum length of this option record has increased since SESAM/SQL V3.2. The actual length is variable and is stored as before in the first two bytes of the option record.

7.5.1.3 CNF record

Distance	Length	Field contents or reference to the field in the form
Header:		
0	2	Length of the record
2	2	Empty
4	3	“CNF” record ID (printable)
7	1	Filler
8	8	Time in STCK format
16	10	Date (*yyyy-mm-dd*), local time (printable)
26	8	Timestamp (*hh*:*mm*:*ss*), local time (printable)
34	10	Date (format *yyyy-mm-dd*), universal time (printable)
44	8	Time stamp (format *hh*:*mm*:*ss*), universal time (printable)
52	12	Filler for extensions
Current values:		
Values for the CNF form APPLICATIONS:		
64	4	Current Applications
68	4	Maximal Applications

Table 34: Layout of the CNF record

7.5.1.4 SESDCN record

Distance	Length	Field contents or reference to the field in the form
Header:		
0	2	Length of the record
2	2	Empty
4	3	“DCN” record ID (printable)
7	1	Filler
8	8	Time in STCK format
16	10	Date (*yyyy-mm-dd*), local time (printable)
26	8	Timestamp (*hh*:*mm*:*ss*), local time (printable)
34	10	Date (format *yyyy-mm-dd*), universal time (printable)
44	8	Time stamp (format *hh*:*mm*:*ss*), universal time (printable)
52	12	Filler for extensions
Current values:		
Values for the SESDCN form APPLICATIONS:		
64	4	Current Applications
68	4	Maximal Applications
72	8	Filler for extensions
Values for the SESDCN form CAPACITY:		
80	4	Messages Fragmentation (Maximum)
84	4	Time in Pool Lock (Maximum)
88	4	Users per Application (Maximum)
92	4	Tasks per Application (Maximum)
96	4	Container per User (Maximum)
100	4	Sub TAs per User (Maximum)
104	4	Users per Catalog (Maximum)
108	4	Messages Fragmentation (Average)
112	4	Time in Pool Lock (Average)
116	4	Users per Application (Average)
120	4	Tasks per Application (Average)

124	4	Container per User (Average)
128	4	Sub TAs per User (Average)
132	4	Users per Catalog (Average)
136	4	Elapsed time in contingency as a percentage of the total elapsed time (float)
140	4	Used Pool Containers
144	4	Free Pool Containers
148	12	Filler for extensions
Values for the SESDCN form OVERVIEW:		
160	4	Number of SESDCNs
164	4	Number of SESAM-DBHs
168	4	Number of active Users
172	4	Number of active Applications
176	4	Number of active PTCs
180	28	Filler for extensions
Values for the SESDCN form TRANSACTIONS:		
208	4	Open TAs
212	60	Filler for extensions
Cumulative values:		
Values for the SESDCN form CAPACITY:		
272	4	Broadcast Messages
276	4	Mail Messages
280	4	BCAM Memory Lacks
284	4	Requests for Locks
288	4	Pool Locks
292	4	Synchron Accesses
296	4	Asynchron Accesses
300	4	SESAM Messages up to 2 Kbyte
304	4	SESAM Messages up to 4 Kbyte
308	4	SESAM Messages up to 8 Kbyte
312	4	SESAM Messages up to 16 Kbyte

316	4	SESAM Messages up to 32 Kbyte
320	4	SESAM Messages up to 64 Kbyte
324	12	Filler for extensions
Values for the SESDCN form OVERVIEW:		
336	4	Number of DML-Statements
340	4	Number of Transactions
344	4	Number of Remote DML
348	20	Filler for extensions
Values for the SESDCN form TRANSACTIONS:		
368	4	External Resets
372	4	External Resets per Administration
376	4	External Resets per Application
380	4	Deadlock Analysis
384	4	Deadlocks Found
388	4	Internal Resets
392	4	Internal Resets per Deadlock
396	4	Internal Resets per Inactivity
400	4	Internal Resets per Lock

Table 35: Layout of the SESDCN record

7.5.1.5 DBH record

Distance	Length	Field contents
Header:		
0	2	Length of the record
2	2	Empty
4	3	“DBH” record ID (printable)
7	1	NAM ID of the DBH to which the record refers (printable)
8	8	Time in STCK format
16	10	Date (*yyyy-mm-dd*), local time (printable)
26	8	Timestamp (*hh*:*mm*:*ss*), local time (printable)
34	10	Date (format *yyyy-mm-dd*), universal time (printable)
44	8	Time stamp (format *hh*:*mm*:*ss*), universal time (printable)
52	12	Filler for extensions
Current values:		
Values for the DBH form I/O:		
64	4	Number of Spaces
68	4	Number of Cursor Files
72	4	Number of LOG Files
76	8	Filler for extensions
84	4	4Kbyte Blocks / Written Blocks on LOG Files (float)
88	8	Filler for extensions
96	64	Filler for extensions
Values for the DBH form SERVICE TASKS:		
160	4	Service Tasks Maximal
164	4	Service Tasks Current
168	4	Service Tasks Active
172	4	Service Tasks Free
176	4	Service Tasks Load (float)
180	4	Orders Not Processed
184	4	Orders Not Fetched

188	4	Filler for extensions
Values for the DBH form SQL INFORMATION:		
192	4	Max. SQL Cursors
196	4	Act. SQL Cursors
200	4	Plans: Number per DBH-Option
204	4	Plan Buffer: Number of Plans
208	4	Plan Buffer: Size of Primary Buffer
212	4	Plan Buffer: Occupation of Primary Buffer
216	4	Plan Buffer: Size of Secondary Buffer
220	20	Filler for extensions
Values for the DBH form SYSTEM INFORMATION:		
240	4	Active Suborders for CALL DML
244	4	Active Suborders for SQL DML
248	4	Available Suborders Maximal
252	4	Available Suborders Current
256	4	Used Suborders Maximal
260	4	Used Suborders Current
264	8	Filler for extensions
272	4	Parallel Open Spaces
276	4	Open Cursors
280	4	Work Container Maximum
284	4	Work-Container Current
288	4	Transfer-Container Maximum
292	4	Transfer-Container Current
296	4	Threads Active
300	4	Threads Ready
304	4	Threads Free
308	4	Threads Waiting
312	4	Threads TA-Lock
316	4	Threads Service Task

320	4	Threads I/O
324	4	Valid Logging-Info - Number of Blocks
328	8	Filler for extensions
Values for the DBH form TRANSACTIONS:		
336	4	Active users
340	4	Open TAs
344	4	Group Commit (float)
348	4	Group Commit/IV (float)
Values for the DBH form STATEMENTS:		
352	4	Statements Number
356	4	Statements Active
360	4	Statements Waiting
364	20	Filler for extensions
Cumulative values:		
Values for the DBH form I/O:		
384	4	Log. Read on Spaces
388	4	Log. Write on Spaces
392	4	Phys. Read on Spaces
396	4	Phys. Write on Spaces
400	4	Log. Read on Spaces (System Data)
404	4	Log. Write on Spaces (System Data)
408	4	Phys. Read on Spaces (System Data)
412	4	Phys. Write on Spaces (System Data)
416	4	Log. Read on Spaces (User Data)
420	4	Log. Write on Spaces (User Data)
424	4	Phys. Read on Spaces (User Data)
428	4	Phys. Write on Spaces (User Data)
432	4	Log. Read on Cursor Files
436	4	Log. Write on Cursor Files
440	4	Phys. Read on Cursor Files

444	4	Phys. Write on Cursor Files
448	4	Phys. Read on LOG Files
452	4	Phys. Write on LOG Files
456	4	Phys. Read on DA-LOG and CAT-LOG
460	4	Phys. Write on DA-LOG and CAT-LOG
464	4	Phys. Read on TA-LOG
468	4	Phys. Write on TA-LOG
472	24	Filler for extensions
Values for the DBH form SERVICE TASKS:		
496	4	Number of Orders
500	12	Filler for extensions
Values for the DBH form SQL INFORMATION:		
512	4	SQL DML
516	4	UPDATE
520	4	RETRIEVAL
524	4	SQL DDL/SSL
528	4	SQL UTI
532	4	Plans: Accesses Core
536	4	Plans: Generated
540	4	Number of Calls Kernel Control
544	4	Number of Calls Catalog Manager
548	4	Number of Calls CALL DML Handler
552	4	Number of Sort Operations
556	4	Number of Calls Authentication[1]
560	4	Number of Calls Authorization [1]
564	4	Number of Authentication Opt.[1]
568	4	Number of Authorization Opt.[1]
572	36	Filler for extensions
Values for the DBH form TRANSACTIONS:		
608	4	Number of Transactions

612	4	Number of Retrieval Transactions
616	4	Number of Update Transactions
620	4	Number of Reset Transactions
624	4	Number of Locks
628	12	Filler for extensions
Values for the DBH form STATEMENTS:		
640	4	Statements Local DML
644	4	Statements Local SQL
648	4	Statements Remote DML
652	4	Statements Remote SQL
656	16	Filler for extensions
Values not output in a form:		
672	10	DBH session begin date (format *yyyy-mm-dd*, local time)[2]
682	8	DBH session begin time (format *hh*:*mm*:*ss*, local time) [2]

Table 36: Layout of the DBH record

[1]Field is only output to SYSLST and to a file

[2]Field is only output to a file

7.5.2 Example of editing the SESMON output file with C

If you want to read the records using C (fread), you receive the records without the fourbyte long record length field. In this case, you have to reduce all distances by four.

```
.
.
char        buf_in[32768];          /* Buffer for reading          */
                                    /* ( 32768 (corresponding to   */
                                    /* STD,16) is taken as the     */
                                    /* buffer size so that the     */
                                    /* buffer size does not have   */
                                    /* to be changed if the length */
                                    /* of a record increases )     */
int         nread;                  /* Number of characters read   */
FILE       *dz;                     /* File pointer                */
.
.
/* Open file */
dz = fopen("link=sesmon","rb,type=record");
while ( !feof(dz) && !ferror(dz) )
{
   /* Read a record */
   nread = fread( (void *)buf_in, 1, sizeof(buf_in), dz);
   if ( nread > 0 )
   {
       /* Edit a record that has been read */
       .
       .
   }
}
if ( ferror(dz) )
{
   /* Output error message */
   .
   .
}
fclose(dz);
.
.
```

7.6 Layout of output to SYSLST

The contents of most of the fields are explained in the descriptions of the forms (see "Layout of the forms"). This section describes only those additional items that are output to SYSLST but not elsewhere.

Page header

```
SESMON <ver>     INTERVAL: 20 SECONDS     DATE: <date>     TIME: <time>     PAGE: 1


================================================================================================================================
```

Hardware and software environment

The information on the hardware and software environment is output only once, at the beginning of and after a live migration.

```
>>> HARDWARE AND SOFTWARE <<<

Hardware Type:  S190-40    BS2000 <ver>         Hostname: D016ZE09          Main Memory: 1232.0 MByte

Number of CPUs:  2         Number of active CPUs:  2                      HSI-Type: XS31          Configuration: S

SESDCN                     Master DCN:          Users:     0               NATL:   0 min           LCTL:   0 min
================================================================================================================================
```

DBH options

The currently set options and information on the DBH start time are output. The information is output again automatically after the options have been changed.

```
>>> OPTIONS <<<
 1.DBH    SESAM-DBH <ver>
          DBH STARTED AT                  : <date> <time>
          LAST RECONFIGURE/RELOAD/RESTART : <date> <time>
          ACCOUNTING             *OFF
          ADMINISTRATOR          *ANY
          COLUMNS                256
          CURSOR-BUFFER          BUFFER-SIZE=512    , FRAME-SIZE=4
          DBH-IDENTIFICATION     CONFIGURATION-NAME=Z, DBH-NAME=X
          DBH-TASKS              1
          LINKED-IN-ATTRIBUTES   CODED-CHARACTER-SET=*NONE
          MSG-OUTPUT             MSG           =*SYSOUT, *SYSLST
                                 OPEN-CLOSE-MSG=*NONE
                                 MAIL=*NONE
          OLD-TABLE-CATALOG      5
          RECOVER-OPTIONS        SYSTEM-DATA-BUFFER=1500   , USER-DATA-BUFFER=1000
                                 TALOG-SUPPORT=*PUBLIC-DISK
                                 CAT-ID=*STD
                                 PRIMARY-ALLOCATION=*STD    , SECONDARY-ALLOCATION=*STD
                                 WALOG-SUPPORT=*PUBLIC-DISK
                                 CAT-ID=*STD
                                 PRIMARY-ALLOCATION=*STD    , SECONDARY-ALLOCATION=*STD
          REQUEST-CONTROL        *NONE
          RESTART-CONTROL        BUFFER-LIMIT:SYSTEM-DATA-BUFFER=50
                                 BUFFER-LIMIT:USER-DATA-BUFFER=50
                                 TALOG-LIMIT=512
                                 LOGICAL-ROLLBACK=*IMMEDIATELY
          RETRIEVAL-CONTROL      USER-DATA-ACCESS=1000      , INDEX-EVALUATION=2147483647
                                 ABORT-EXECUTION=2000
          SECURITY               PRIVILEGE-VIOLATIONS=10
                                 SAT-SUPPORT=*OFF
          SERVICE-TASKS          INITIAL=1 , MAXIMUM=1   , JOBCLASS=*STD
                                 WORK-FILES=*PUBLIC-DISK
                                 CAT-ID=*STD
                                 PRIMARY-ALLOCATION=*STD    , SECONDARY-ALLOCATION=*STD
                                 RECORDS-PER-CYCLE=*STD
          SESSION-LOGGING-ID     SESSION-ID=*STD, START-NUMBER=0001
          SPACES                 100
          SQL-DATABASE-CATALOG   10
          SQL-SUPPORT            *YES
                                 INACTIVITY-TIME=1  , CURSORS=70
                                 INTERNAL-SORT-LIMIT=5000
                                 PLANS=70
          SUBORDERS              INITIAL=24    , MAXIMUM=24
          SYSTEM-DATA-BUFFER     1536
          SYSTEM-THREADS         WRITE-THREADS=1
          THREADS                2
          TRANSACTION-SECURITY   *YES
                                 LOCK-TIME=4  , INACTIVITY-TIME=40
                                 MAX-ISOLATION-LEVEL=*STD
                                 LOCK-ESCALATION:NUMBER-RECORDS=4000
                                 LOCK-ESCALATION:PERCENTAGE-RECORDS=50
                                 LOCK-ESCALATION:NUMBER-INDEX-VALUES=1000
          TRANSFER-CONTAINER     INITIAL=120    , MAXIMUM=16000
          USERS                  24
          USER-DATA-BUFFER       1536
          WORK-CONTAINER         INITIAL=120    , MAXIMUM=120
================================================================================================================================
```

The CONFIGURATION form “APPLICATIONS”

```
>>> APPLICATIONS <<<

 Application                 Number of Users       Number of Tasks       Current Applications:        1
                               Current            Maximal  /  Current    Maximal Applications:       64
 APPLIC02                         7                  8     /    8
==========================================================================================================
```

The CONFIGURATION form “PREFETCH-BUFFERS”

```
>>> PREFETCH-BUFFERS <<<
  Appl./                                 H  I  T  R  A  T  E                                  Perc.     Max.
   TSN         T    <=1KB    <=2KB    <=4KB    <=8KB   <=16KB   <=32KB   <=64KB    >64KB     Occ.     Occ.
  TSN=1Z03     T    --%      --%      --%      --%      --%      --%      --%      --%       6%      60%
________     _      --%      --%      --%      --%      --%      --%      --%      --%       0%       0%
________     _      --%      --%      --%      --%      --%      --%      --%      --%       0%       0%
________     _      --%      --%      --%      --%      --%      --%      --%      --%       0%       0%
________     _      --%      --%      --%      --%      --%      --%      --%      --%       0%       0%
. . .
==========================================================================================================
```

The SESDCN form “APPLICATIONS”

A maximum of 50 transaction applications are output. The output is sorted in descending order according to the number of active users (Number of User Current).

```
>>> APPLICATIONS <<<
 Application                 Number of Users       Number of Tasks        Current Applications:        1
                               Current            Maximal  / Current      Maximal Applications:       64
 ANWEND01                         6                  1  /
==========================================================================================================
```

The SESDCN form “CAPACITY”

```
>>> CAPACITY <<<
                Session  Interval                                Maximum Average
 Broadcast Messages :          0         -     Message Fragmentation :       0       0     Elapsed Time in
 BCAM Memory Lacks  :          0         -     Time in Pool Lock     :    4578      12     Contingency          :
0.00%
 Mail Messages      :          7         -     Users per Application :       6       6     Used Pool Containers :
554
 Requests for Locks :        381         -     Tasks per Application :       1       1     Free Pool Containers :
710
 Pool Locks         :        381         -     Container per User    :       3       3
 Synchron Accesses  :         88         -     Sub TAs per User      :       1       0
 Asynchron Accesses :          0         -     Users per Catalog     :       1       1

SESAM MESSAGES:
 Length          <= 2K         <= 4K         <= 8K       <= 16K        <= 32K        <= 64K
 Session           115             1             4            0             0             0
 Interval            -             -             -            -             -             -

 ==========================================================================================================
```

The SESDCN form “OVERVIEW”

```
>>> OVERVIEW <<<
 Master DCN    :                Number of SESDCNs            :      1                                  Session Interval
 Configuration : Z              Number of SESAM-DBHs         :      1      Number of DML-Statements:         88       -
 Session Begin : <date>         Number of active Users       :      7      Number of Transactions  :         25       -

                 <time>         Number of active Applications:      1      Number of Remote DML    :          9       -
 Users         : 128            Number of active PTCs        :      0
 NATL          : 12  min
 LCTL          : 8   min
 ==========================================================================================================
```

The SESDCN form “TRANSACTIONS”

A maximum of 200 transactions are output. They are sorted according to their duration. If a transaction is locked by a transaction that can no longer be output, this is indicated in the “Lock” column by two asterisks (**).

```
>>> TRANSACTIONS <<<
                        Session  Interval                          Session Interval                        Session  Interval
 Deadlock Analysis :         0         -    External Resets    :        0         -   Internal Resets :        6         -
 Deadlock Found    :         0         -    - Administration   :        0         -   - Deadlock      :        0         -
                                            - Application      :        0         -   - NATL          :        6         -

 Open Transactions :         7                                                        - LCTL          :        0         -

 TA-NR   TAC        User-ID                          S         I/O         Lock     Duration    Statements
```

```
    1   EXAMPLE  D016ZE07TSN=8GZNUSER0001           A          0                 00:13:50            1
    2   EXAMPLE   D016ZE07TSN=8GZNUSER0002          A          0                 00:13:31            1
    3   EXAMPLE   D016ZE07TSN=8GZNUSER0003          A          0                 00:13:13            1
    4   EXAMPLE   D016ZE07TSN=8GZNUSER0004          A          0                 00:12:52            1
    5   EXAMPLE   D016ZE07TSN=8GZNUSER0005          A          0                 00:12:26            1
    6   EXAMPLE   D016ZE07TSN=8GZNUSER0006          A          0                 00:12:07            1
    7   SQLDB112   D016ZE07TSN=8G40SEDI63           A          0                 00:03:14            2
==========================================================================================================
```

The DBH form “I/O”

```
>>> I/O <<<                                DBH: X

  Number of           Spaces       Cursor Files      LOG Files
                           2                  0              3

                  System Data Buffer         User data Buffer          Cursor Files
                    Session Interval         Session Interval         Session Interval
  Log. Read             260       39             114       21              34       15
  Log. Write             16        2               4        1              13        5
  Phys. Read             82        0              18        0               5        2
  Phys. Write            11        0               3        0               5        2
  Hitrate            66.30%   100.00%         82.20%   100.00%         78.72%    80.00%

                     DA-LOG/CAT-LOG              TA-LOG                   WA-LOG
                    Session Interval         Session Interval         Session Interval
  Phys. Read              0        0               1        0               0        0
  Phys. Write             6        1               8        1               0        0
  4K Blocks /
  Written Blocks       1.00                    2.12                     0.00

  Spaces and Cursors
                                        Data      Log. Read          Log. Write         Phys. Read          Phys. Write    Hitrate
  Name                                        Session Interval    Session Interval    Session Interval    Session Interval
  REQUESTCUST.CATALOG                  SYS       240       35          9        0         74        0          7        0  67.47%
                                       USR       100       20          2        0         17        0          2        0  81.37%
  <CURSOR FILES>                                  34       15         13        5          5        2          5        2  78.72%
  REQUESTCUST.CUST                     SYS        20        4          7        2          8        0          4        0  55.56%
                                       USR        14        1          2        1          1        0          1        0  87.50%
  LOG Files
                                                                  Phys. Read           Phys. Write         4K Blocks /
  File Name                                                     Session Interval      Session Interval   Written Blocks
  SESAMZX.TA-LOG                                                    1        0            8        1             2.12
  $ID1.REQUESTCUST.0003.D.0001                                      0        0            3        1             1.00
  $ID1.REQUESTCUST.0003.C.0001                                      0        0            3        0             1.00
==========================================================================================================
```

Field descriptions

Spaces and Cursors

This output contains a maximum of 100 entries. The entries are sorted in decreasing order according to their number of I/Os.

Name

Space name or <CURSOR-FILES>. In the case of space names, a distinction is drawn between USR and SYS.

Log. Read

Number of logical read accesses of the catalog space and user spaces and of the cursor files since the database was opened and per time period.

Log. Write

Number of logical write accesses of the catalog space and user spaces and of the cursor files since the database was opened and per time period.

Phys. Read

Number of physical read accesses of the catalog space and user spaces and of the cursor files since the database was opened and per time period.

Phys. Write

Number of physical write accesses of the catalog space and user spaces and of the cursor files since the database was opened and per time period.

Hitrate

Logical accesses of the catalog space and user spaces and of the cursor files as a percentage of the total number of accesses.

LOG Files

This output contains a maximum of 100 entries. The entries are sorted in decreasing order according to their number of I/Os.

File Name

File name of the LOG file.

Phys. Read

Number of physical read accesses of the specified LOG file (for the session and per time period).

Phys. Write

Number of physical write accesses of the specified LOG file (for the session and per time period). The I /Os for RECOVER are not included here.

4K Blocks/Written Blocks

Average number of 4-Kbyte blocks written in the case of a write request.

The DBH form “SERVICE ORDERS”

A maximum of 100 requests are output.

Detailed outputs for the requests in the *User-ID / Order* information block may be provided in the *ALTER PARTITIONING: MOVE ROWS / CHG ROWID*, *RESTORE PROGRESS*, *RECOVER/REFRESH PROGRESS*, *REORG SPACE PROGRESS* and *UNLOAD/EXPORT/LOAD/IMPORT TABLE PROGRESS* information blocks.

```
>>> SERVICE ORDERS <<<                      DBH: 8
User-ID                          Order                          TSN   S  Duration
D016ZE09TSN=7V3ASESUTI           UNLOAD                         7WP7  A  00:00:03
D016ZE09TSN=7V2ESESUTI           REORG SPACE:FIRST PHASE        7WP8  A  00:00:02
D016ZE09TSN=7V4ESESUTI           ALTER PARTITIONING             7WP9  A  00:00:01
D016ZE09TSN=7V2DDSQL             CREATE INDEX                   7WP2  A  00:00:00
D016ZE09TSN=7V2DDSQL             CREATE INDEX                        F  00:00:00
==============================================================================================
UNLOAD/EXPORT/LOAD/IMPORT TABLE/ALTER PARTITIONING PROGRESS:
                                                Number of Records
TSN   Table                                Current       Total    Done
7WP7  PTJDB                                  81856      206991     39%
==============================================================================================
REORG SPACE PROGRESS:
                                                Number of Blocks
TSN   Space Name                           Current       Total    Done
7WP8  $SQLDB111.CME.SPACE1                     752        6768     11%
==============================================================================================
UNLOAD/EXPORT/LOAD/IMPORT TABLE/ALTER PARTITIONING PROGRESS:
                                                Number of Records
TSN   Table                                Current       Total    Done
7WP9  PTX23                                    730           0      0%
==============================================================================================
```

The DBH form “SERVICE TASKS”

```
>>> SERVICE TASKS <<<                      DBH: X
Service Tasks   Maximal    Current    Active      Free       Load
                      1          1         0         1      0.00%
Orders    Number(Session) Number(IV)  Not Processed   Not Fetched
                       10          6              0             0
               Number of                 Used-CPU-      Elapsed                  Size      Memory      Pages/
TSN     Order     Orders    Phys. I/O   Time (msec)  Time (msec)       Load    (Pages)   Extension   Extension
66WI                  10           30           141         5736      2.46%        515           1      256.00
==============================================================================================================
```

Field descriptions

A maximum of 50 service tasks are output. The output is sorted in descending order according to the amount of CPU time used (Used-CPU-Time).

Memory Extension

Number of dynamic memory extensions in the service task.

Pages/Extension

Average number of requested pages per memory extension in the service task.

The DBH form “SQL INFORMATION”

```
>>> SQL INFORMATION <<<                        DBH: X
                              Session  Interval    Plans                                                  Session  Interval
  SQL DML             :          22         2     Accesses        :         23    Sort Operations  :          1         0
  UPDATE              :           1         0     Accesses/IV     :          3
  RETRIEVAL           :          21         2     Generated       :          3    Max. SQL Cursors :          1
  SQL DDL/SSL         :           0         0     Generated/IV    :          1    Act. SQL Cursors :          1
  SQL UTI             :           0         0     Number per
                                                  DBH-Option      :         70
 Number of Calls              Session  Interval
  Kernel Control      :         131        30    Plan Buffer
  Catalog Manager     :          70        16     Number of Plans :          3
  CALL DML Handler    :           5         0     Size of
  Authentication      :          26         3     Primary Buffer  :    1148000
  Authorization       :           9         2     Occupation of
                                                  Primary Buffer  :      12000
  Authentication Opt. :          24         3     Size of
  Authorization Opt.  :           5         1     Secondary Buffer:      32792
==================================================================================================================
```

Field descriptions

Authentication

Number of authentication calls (for the session and per time period). Authentication calls check whether the user is allowed to work with the specified authorization key.

Authorization

Number of authorization calls for DML (for the session and per time period).
Authorization calls check whether the user is allowed to work with the desired database objects (e.g. a table) in the desired way (e.g. update them).

Authentication Opt.

Number of successful authentication optimizations (for the session and per time period).

Authorization Opt.

Number of successful authorization optimizations for DML (for the session and per time period).

The DBH form “STATEMENTS”

```
>>> STATEMENTS <<<                          DBH: X

Statements        Session   Interval             Statements
   Local DML :          5          0             Number :            0
   Local SQL :         53          5             Active :            0
   Remote DML:          0          0             Waiting:            0
   Remote SQL:          0          0
                                                                          Activity-
TAC        User-ID                          S        I/O Duration        Time  Type
  QM124ELS SYSTEM01TSN=87L0KAASTEUG                  3732 01:02:50        0.00  M
  QM124ELS SYSTEM01TSN=87LXKAASTEUG          L       3280 01:02:41        0.00  M
  QM124ELS SYSTEM01TSN=87LYKAASTEUG                 11125 01:02:25        0.00  M
  KSURQTT1 SYSTEM01ANWEND01QTT1IP020000003B        141670 01:01:03        0.00  M
  KSUUQCON SYSTEME01ANWEND01QCONHP0100000062 L      14015 00:58:28        0.00  M
  KSUUQDEC SYSTEM01ANWEND01QDECDS010000005D  L      12949 00:53:32        0.00  M
  KSUUQDEL SYSTEM01ANWEND01QDELHS010000005C  L      20060 00:50:20        0.00  M
==================================================================================================================
```

The DBH form “SYSTEM INFORMATION”

In the case of list output either the logging information or the information about the DBH session restart (separated according to AI and LBI blocks) is displayed.

List output in an ongoing DBH session:

```
>>> SYSTEM INFORMATION <<<                 DBH: W


Active Suborders              Available Suborders       Used Suborders
   for CALL DML:         0        Maximal:     32767        Maximal:        32
   for SQL DML :         1        Current:       201        Current:         1


Parallel Open                 Accounting     :  OFF
         Spaces:        31     SESCOS         :  OFF
   Open Cursors:         1     Request Control:   ON


                    Current      Percentage         Maximal  Percentage
Container              Size      Occupation            Size  Occupation
    Work               5000          0.52%          222222       0.09%
    Transfer           5000          0.11%         1000000       0.01%


Threads                                          Service
     Active     Ready      Free   Waiting   TA-Lock      Task       I/O
          1         0        18         0         0         1         0


Valid Logging Information - Number of Blocks:        453
================================================================================================================================
```

List output during DBH restart:

```
>>> SYSTEM INFORMATION <<<                 DBH: W


Active Suborders              Available Suborders       Used Suborders
   for CALL DML:         0        Maximal:         0        Maximal:         0
   for SQL DML :         0        Current:         0        Current:         0


Parallel Open                 Accounting     :
         Spaces:         0     SESCOS         :
   Open Cursors:         0     Request Control:


                    Current      Percentage         Maximal  Percentage
Container              Size      Occupation            Size  Occupation
   Work                   0          0.00%               0       0.00%
   Transfer               0          0.00%               0       0.00%


Threads                                          Service
     Active     Ready      Free   Waiting   TA-Lock      Task       I/O
          0         0         0         0         0         0         0


Physical Repair : Current Blocks         30   Total Blocks         90      Done  33.33%
Logical Rollback: Current Blocks          0   Total Blocks          0      Done   0.00%
================================================================================================================================
```

The DBH form “TASKS”

```
>>> TASKS <<<                      DBH: X
                                                                  Used CPU-
TSN     Threads    Orders       Waits/IV      IOs/IV     Waits/IO    Time (msec)
8ENU          0         0              -           -            -          4647
================================================================================================================================
```

The DBH form “SYSTEM THREADS”

```
>>> SYSTEM THREADS <<<                            DBH: 0
WRITE THREADS
Number     Blocks Written Total       Blocks Written Intervall
     1             5374                       8
================================================================================================================================
```

The DBH form “TRANSACTIONS”

```
>>> TRANSACTIONS <<<                      DBH: X
Transactions     Session   Interval      Users             :          2
   Number   :          9          1      Open TAs          :          0
   Retrieval:          5          0      Group Commit      :       0.00
   Update   :          4          1      Group Commit/IV   :       0.00
   Reset    :          2          0      Number of Locks   :          0
                                         Number of Locks/IV:          0
TA-NR  TAC       User-ID                        S         I/O   Lock  Duration  Statements  Type
    1  SQLABT98  SYSTEM01QMUJBS010000004C  A       55926         00:00:04         181   DML
 .
 .
 .
Locked Transactions:
TA-NR  Lock-Type  Object-Type  Space-Name                              Table-/
                                                                      Index-ID
 .
 .
 .
================================================================================================================================
```

Field descriptions

A maximum of 200 transactions are output. They are sorted according to their duration. If a transaction is locked by a transaction that can no longer be output, this is indicated in the “Lock” column by two asterisks (**).

Type

Transaction type
DML: DML transaction
DDL: DDL transaction
'BLANK' (blank): the transaction type is still undefined

Locked Transactions

Only transactions marked as locked in upper part of the table. More information is given here on the locked object.

TA-NR

Unique number of the locked transaction assigned by SESMON. This number can be used to assign the entry to the associated entry under TA-NR in the upper part of the table.

Lock-Type

The lock type:

- shared lock
- exclusive lock
- preannounced exclusive lock (required)

Object-Type

The object type of the locked object:

- catalog (whole database)
- Space
- table
- record
- Index
- SI entry (secondary index entry)
- plan (SQL access plan)
- metadata (metadata of a catalog)

Space-Name

Name of the space. It indicates the space in which the locked object is located.

- For the object types “space”, “table”, “record”, “index” and “SI entry” in the form *log.catalogname.spacename.*
- For the object types “catalog” and “metadata” in the form *log.catalogname.*
- For the object type “plan” spaces are placed here.

Table-/Index-ID (five decimal digits)

Internal table number when a table or record is locked, internal index number when an index or SI entry is locked.If the locked object is a record, the table in which the record is locked is specified here. If the locked object is an SI entry, the index containing the locked SI entry is specified. Blank spaces are placed here for all object types other than table, record, index and SI entry.

The universal user can create an assignment table that assigns the correct table name and index name to the corresponding internal table number and internal index number. There are two ways to do this:

a. by issuing the following SQL statements for each database (see the “ SQL Reference Manual Part 1: SQL Statements”):

```
SELECT TABLE_SCHEMA,TABLE_NAME,TABLE_ID FROM
       SYS_INFO_SCHEMA.SYS_TABLES
SELECT INDEX_SCHEMA,INDEX_NAME,INDEX_ID FROM
       SYS_INFO_SCHEMA.SYS_INDEXES
```

b. by using the utility monitor to call the forms “SYS_INFO_SCHEMA,TABLES” and “SYS_INFO_SCHEMA, INDEXES” for each database (see the “ Utility Monitor” manual).

7.7 Output of statistical data via SNMP

There are two subagents for the SNMP management of SESAM/SQL in BS2000:

- Subagent for the management of SESAM databases in the software product SSC-BS2.This supplies information on SESAM databases and SESAM-DBHs.
  This subagent is described in detail in the manual “ SNMP Management für BS2000”.
- Subagent for outputting data from the performance monitor SESMON. This subagent is referred to as **SESAM-MON-Subagent**.

This section provides a short introduction to the structure of an SNMP installation for BS2000 and then discusses the use of the SESAM-MON-Subagent.

Structure of an SNMP installation in BS2000

SNMP stands for **S**imple **N**etwork **M**anagement **P**rotocol and was developed as a protocol for network management services in the TCP/IP internet. SNMP’s range of application has since been extended to include system management, application management and even management of middleware products such as databases and transaction monitors. Similarly to TCP/IP, the name SNMP does not just stand for the protocol but for the entire management system which is based on SNMP. SNMP employs a client/server architecture, where the management platform is the client and the management agents are the servers.

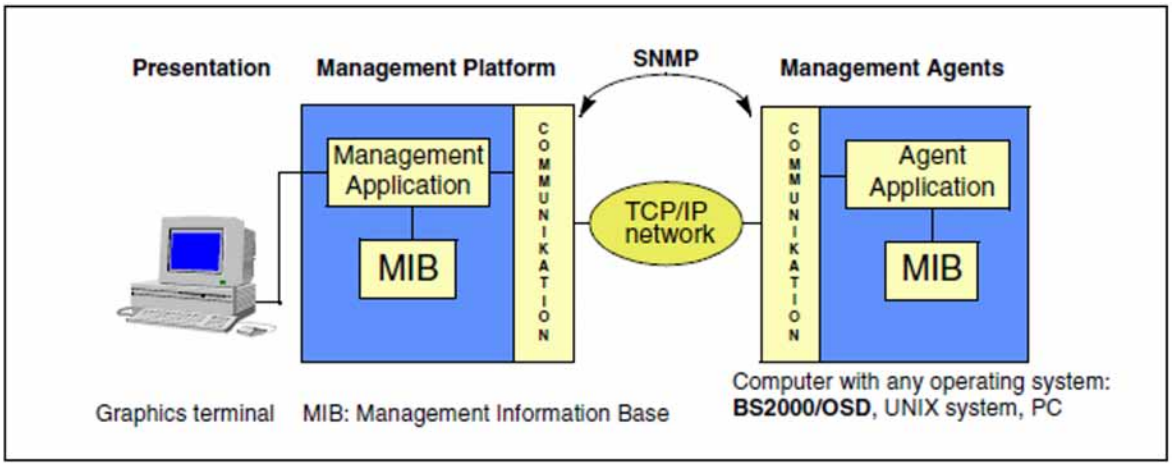


Figure 9: Communication between management platform and agents via SNMP

The SNMP manager (Management Application) is the software which generates the requests to the individual agents and sends them via SNMP to the corresponding agents. The SNMP manager displays the information received from the agent.

An SNMP agent (Agent Application) is the software which receives, executes and responds to the requests sent by the SNMP manager. For the SNMP management in BS2000 there is an SNMP agent available in the software product SBA-BS2 which is structured according to a master-subagent principle.
The master agent centrally performs basic tasks such as processing of the SNMP protocol, safety functions, work distribution, etc. Each subagent is only responsible for a specific subarea of the monitored components. The subagent communicates with the master agent. The subagents are self-contained and can be started and terminated at any time. Examples: HSMS subagent, FT subagent, both SESAM/SQL subagents.

In the **M**anagement **I**nformation **B**ase (MIB), the management-specific objects of the relevant component are defined and the object attributes are described. The MIB therefore forms the basis of the communication between the management platform and an agent.

You will find further information in the manual “ SNMP Management für BS2000”.

Prerequisites for using SESAM-MON-Subagents

The local computer must be embedded in an SNMP environment. This means that the SNMP agent for BS2000 (i.e. the BS2000 master agent) must be active on the computer.

Communication between management platform and SESAM-MON-Subagent

The task of the management platform is to request the required data via the BS2000 master agent (GET-Request) and to present it. The management platform can define refresh times or request data directly, i.e. refresh the data in the memory pool regardless of the refresh times (SET-Request, MIB fields `sesamCnfRecCacheTime` or `sesamCnfRecUpdate`).

The MIB is shipped as part of SESAM/SQL-Server (File “fscSesam.my” in the library SYSLIB.SESAM-SQL.*version*.SNMP-SA).

The layout of the MIB is based on the layout of the file output from SESMON (see "Layout of output to a file"). You will find information on the individual fields online at the management station.

The statistical data generated by the performance monitor SESMON are supplied via the SESAM-MON-Subagent to the management platform.

Starting the SESAM-MON-Subagent in Emanate environment (up to and including OSD V10)

The SESAM-MON subagent is a subagent which is started in the POSIX shell or in BS2000.

The SESAM-MON subagent logs on with the BS2000 master agent, transfers its MIB to the BS2000 master agent, sets up resources and waits for requests from the management platform (via the BS2000 master agent).

1. Starting in BS2000:

/START-SNMP-SESAM-MON
VERSION= ***STD** / <product-version> , **MONJV**= ***NONE** / <filename 1 .. 54 without-gen-vers> , **CPU-LIMIT**= ***STD** / <integer 1 .. 32767> / ***NO** , **JOB-CLASS**= ***STD** / <name 1 .. 8> , **TIMER-INTERVAL** = 5 / <integer 1 .. 32767>

2. Starting in the POSIX shell with:

   `sesam-monagt [-t <int>]`

   The operand `<int>` corresponds to the operand TIMER-INTERVAL.

Operands

VERSION = *STD / <product-version>

Defines the version of the agent to be started. This statement is not currently evaluated.

MONJV = *NONE / <filename 1..54 without-gen-vers>
Name of the job variable which is to monitor the agent.
The default setting is *NONE - no monitoring by a job variable.

CPU-LIMIT = *STD / <integer 1 .. 32767> / *NO
Entry of the maximum CPU runtime in seconds.
If *STD is entered, the default value of the jobclass is used.

JOB-CLASS = *STD / <name 1..8>
Job class with which the agent is started.
If *STD is entered, the default job class of the user id is used.

TIMER-INTERVAL = 5 / <integer 1 .. 32767>
Timer interval in which the agent checks whether there are requests from the command program. The timer interval is set as default to five seconds.

Starting the SESAM-MON-Subagent in Net-SNMP environment (from OSD V11)

If a Net-SNMP instance is detected during the installation of SESAM the SESAM-MON-subagent is installed to start automatically during the startup of the posix subsystem. So a manual start of the subagent usually is not needed.

if desired or needed it can be done within the posix shell as shown above. A manual start via BS2000 command is not intended anymore.

Starting the performance monitors

You start the SESMON performance monitor or monitors which are to operate together with the SESAM-MON subagent as a batch job. In this case you indicate the operand OUTPUT=*SNMP under the statement SET-MONITOR-OPTIONS (see "Start parameters when running SESMON in batch mode").

A performance monitor sets up resources on starting and waits for requests from the SESAM-MON subagents. Performance monitors and SESAM-MON subagents can be started independently.

How SESAM-MON subagents work

For a description of the working principles of the management platform and SNMP agents work, see the manual “SNMP Management für BS2000”.

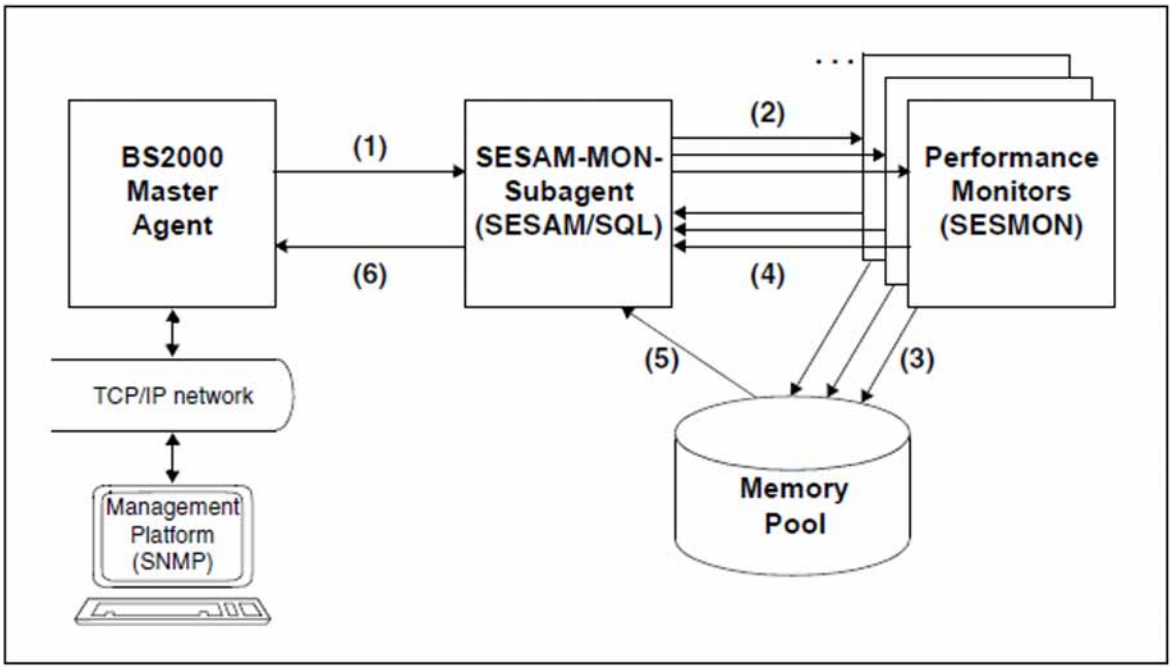


Figure 10: How SESAM-MON subagents work

1. The SESAM-MON subagent receives via the BS2000 master agent a request to prepare data from a management platform. The request may relate to all SESAM applications on a computer.
2. The SESAM-MON subagent communicates with one or more SESAM performance monitors via memory pools. Thus data from all SESAM applications on a computer can be requested by the management platform. The SESAM-MON subagent uses the different performance monitors, depending on the configuration and the DBHs.

   To start the retrieval of information from the memory pools, the SESAM-MON-subagent transmits a signal via P1-Eventing to the appropriate performance monitor.
3. The performance monitors write the result data independently into the common memory pool.
4. After the result data have been written, the performance monitors notify the SESAM-MON subagent via SOLSIG of the event identification. The SESAM-MON subagent waits from (2) in P1 eventing.
5. The SESAM-MON subagent transfers the result data from the common memory pool to the MIB and on to the management platform.
6. Request end for the SESAM-MON subagent.

Termination of the SESAM-MON subagent in Emanate environment (up to and including OSD V10)

The SESAM-MON subagent is terminated (regardless of the environment in which it was started) in BS2000 with the command STOP-SNMP-SESAM-MON (no operands) or in the POSIX shell with:

```
sesam-moncmd T.
```

The SESAM-MON subagent logs off with the BS2000 master agent, releases resources and is terminated.

Termination of the SESAM-MON subagent in Net-SNMP environment (from OSD V11)

If a Net-SNMP instance is detected during the installation of SESAM the SESAM-MON-subagent is installed to terminate automatically during the tremination of the posix subsystem. So a manual termination of the subagent usually is not needed.

if desired or needed it can be done within the posix shell as shown above. A manual termination via BS2000 command is not intended anymore.

Termination of the performance monitors

You terminate the SESMON performance monitor or monitors which operate together with the SESAM-MON subagent as described on "SESMON administration".
A performance monitor is automatically terminated when all the tasks which SESMON is to monitor are completed.

7.8 Outputting statistical data via openSM2

With SESAM/SQL and BS2000 you can transfer data from the performance monitor SESMON to the software monitor openSM2 and there display it in the report SESAM_SQL or edit it graphically using the INSPECTOR component.

Information on openSM2 is provided in the “ openSM2 (BS2000)” manual.

Prerequisite in SESAM/SQL

To transfer statistical data from SESAM/SQL to openSM2, start SESMON in batch mode (see "Start parameters when running SESMON in batch mode"):

```
/START-SESAM-PERF-MONITOR
```

```
//SET-MONITOR-OPTIONS ...,OUTPUT=*SM2
```

Only one DBH per SESMON instance can be specified for `OUTPUT=*SM2`. A new SESMON instance must be started to output data of each additional DBH.

The interval at which SESMON transfers the data to openSM2 is automatically set to approx. 30% of the SM2 monitoring cycle. It cannot be set manually.

Prerequisite in BS2000

To transfer the statistical data in openSM2, start the measurement program for SESAM/SQL in the administration part of openSM2:

```
/START-SM2
*CALL-ADMINISTRATION-PART
//START-MEASUREMENT-PROGRAM TYPE=*SESAM-SQL
...
```

The monitoring program can be deactivated again using the `STOP-MEASUREMENT-PROGRAM` statement.

Report SESAM_SQL of openSM2

You can display the performance data of the SESAM monitor in openSM2 via the report SESAM_SQL. See also the “ openSM2 (BS2000)” manual.

You can start monitoring using the command string below, for example:

```
/START-SM2
*REPORT SESAM_SQL
*OUTPUT TERMINAL=YES,INTERVAL=20
*START
...
```

In this case the statistical data is displayed on the terminal every 20 seconds, e.g.:

```
AILEEN2  SM2 SESAM SQL           CYCLE:   20 S             08-09-03  15:42:05
    DB CONF | UPDTA | RETTA | RESTA | PLGEN | RDSYS | WRSYS | RDUSR | WRUSR
            | (1/S) | (1/S) | (1/S) | (1/S) | (1/S) | (1/S) | (1/S) | (1/S)
    --------+-------+-------+-------+-------+-------+-------+-------+------
    SESAM10 |   5.0 |   5.3 |   5.7 |   8.0 |   8.3 |   8.7 |  10.3 |  10.0
    SESAM11 |  15.0 |  15.3 |  15.7 |  18.0 |  18.3 |  18.7 |  20.3 |  20.0
    SESAM12 |  28.3 |  28.7 |  29.0 |  31.3 |  31.7 |  32.0 |  33.7 |  33.3
```

where

DB CONF	Name of the DB configuration.
UPDTA (1/S)	Number of update transactions per second
RETTA (1/S)	Number of retrieval transactions per second
RESTA (1/S)	Number of reset transactions per second
PLGEN (1/S)	Number of SQL plan generations per second
RDSYS (1/S)	Number of logical read accesses to the system data buffer
WRSYS (1/S)	Number of logical write accesses to the system data buffer
RDUSR (1/S)	Number of logical read accesses to the user data buffer
WRUSR (1/S)	Number of logical write accesses to the user data buffer

The monitored data is supplied asynchronously to openSM2 by SESAM/SQL and applies for one or more cycles defined by SESAM/SQL which do not need to match the SM2 cycle exactly. Both differences in the duration of the cycle and temporal displacements between the SESAM/SQL and SM2 cycles can occur here.
The duration of one or more SESAM/SQL cycles is used to standardize the monitored data to one second. The data is therefore exact, but it only matches the SM2 cycle to a limited extent.

Displaying the statistical data using INSPECTOR

The statistical data of SESAM/SQL can be evaluated and edited graphically on a workstation using the openSM2 component INSPECTOR, together with other user values of the BS2000 system. Which counters are to be displayed can be selected in the report group SESAM-SQL.

8 DA-LOG formatting by SEDI70

The SEDI70 program formats CAT-LOG and DA-LOG files (logging files for the catalog space and user spaces) for printing according to the input parameters of the media recovery files.

The following information is important to interpret the output:

- The CAT-LOG file contains all changes to the metadata that are applied to the catalog during a DBH session. These changes are logged as DMLs (INSERT, UPDATE, DELETE) because the metadata is stored in the catalog as a table.
  Changes to the metadata can be made
  - by a user (e.g. by CREATE TABLE, ALTER TABLE ...)
  - by the system (e.g. by entering the DA-LOG files in the SYS_INFO_SCHEMA.SYS_DA_LOGS or by changing the “time of change” time stamp in the SYS_INFO_SCHEMA.SYS_SPACES during the first update of a space within a DBH session)
- All changes to the user tables that are made during a DBH session to user spaces when the logical data backup is switched on are logged in the DA-LOG file.
  There is a system table in the user space with the table ID 1. It is used to ensure consistency between the catalog space and user space.
- CAT-LOG and DA-LOG files are divided into “units”. The units contained in a CAT-LOG or DA-LOG file can be found in the CAT-REC file or in the SYS_INFO_SCHEMA.SYS_DA_LOGS. New units are created in the following cases, for example:
  - When a file is opened (beginning of session, CAT-LOG/DA-LOG switch, COPY CATALOG)
  - In the case of logical starting points (COPY SPACE)
  - In the case of selection (LOAD)
- Changes are logged in the order in which the were made. A change is only written permanently by a COMMIT WORK. The association between the change and the corresponding COMMIT WORK is done using the 28-byte user ID. If a restart label (unit record) exists between the change and a follow-up COMMIT WORK, then the change is rolled back.
- Changes that have the same user ID and the same statement ID were produced by an external statement. For example, an update statement with 100 matches produces 100 update records in the log files. If an error occurs in the nth match (e.g. a data error), then a cancel statement takes effect. This means that all update increments with the same user ID and the same statement ID are invalid. The transaction boundary is not affected by this, i. e. it is still open and a COMMIT WORK later on commits all remaining changes of the user.

> ¡ The following always applies: Output sent to SYSOUT and SYSLST is not upwardly compatible. Output layout may vary depending on the version used.

8.1 SEDI70 call

the SEDI70 utility can be started with the following command

```
/START-SESAM-LOG-FILE-EVAL
```

See also the section “Starting SESAM/SQL programs via start commands”.

8.2 SEDI70 input parameters

SEDI70 expects the input parameters from SYSDTA.

8.2.1 The higher-level parameter CREATE-REPORT

Function

You use CREATE-REPORT to specify which file SEDI70 is to evaluate, where SEDI70 is to write the report to, which data SEDI70 is to evaluate and how the SEDI70 statistics should look.

If you enter the name of the log file (DA-LOG file) to be evaluated in the appropriate SDF screen and send it off with the default settings, SEDI70 produces a report on all the data and outputs it to SYSLST.

The input for SEDI70 is entered in the SDF format in the same manner as the input for SESAM-DBH options and the SESCOSP control.

All the operands of CREATE-REPORT are described below as individual parameters.

CREATE-REPORT
INPUT-FILE = ,OUTPUT = ,SELECT =

Operands

INPUT-FILE =

See the “The individual parameter INPUT-FILE”

OUTPUT =

See the “The individual parameter OUTPUT”

SELECT =

See the “The individual parameter SELECT”

The individual parameter INPUT-FILE

Function

You use this parameter to specify the file name of the log file that SEDI70 is to evaluate.

INPUT-FILE
= <filename 1..54_without_generation_version> / *PARAMETERS(...) *PARAMETERS(...) | USER-IDENTIFICATION = *OWN / <name 1..8> | ,CATALOG-NAME = <filename 1..18 without-all> | ,COPY-NUMBER = 1 / <integer 1..999999> | ,LOG-FILE-TYPE = *DA-LOG / *CAT-LOG | ,SUB-NUMBER-FROM = 1 / <integer 1..9999> | ,SUB-NUMBER-TO = 9999 / <integer 1..9999>

Operands

INPUT-FILE =

Name of the log file that SEDI70 is to evaluate.

INPUT-FILE = <filename 1..54_without_generation_version>

Explicit specification of the log file.

INPUT-FILE = *PARAMETERS(...)

Specify one or more log files that belong together by entering the database name, backup copy number, type and a range of file subnumbers.

This will designate the following log files:

```
[$user-id.]catalog.cccccc.x.ffff ... [$user-id.]catalog.cccccc.x.tttt
```

The following terms have the following meanings:

`user-id`	BS2000 user ID under which the log files are stored (see the operand USER-IDENTIFICATION)
`catalog`	Name of the catalog (see the operand CATALOG-NAME)
`cccccc`	Backup copy number (see the operand COPY-NUMBER)
`x`	Type of job log file (see the operand LOG-FILE-TYPE)
`ffff`	“From” file subnumber (see the operand SUB-NUMBER-FROM)
`tttt`	“To” file subnumber (see the operand SUB-NUMBER-TO)

USER-IDENTIFICATION = *OWN / <name 1..8>

BS2000 user ID under which the log files are stored.

USER-IDENTIFICATION = *OWN

The log files are stored under the current BS2000 user ID.

USER-IDENTIFICATION = <name 1..8>

The log files are stored under the specified BS2000 user ID.

CATALOG-NAME = <filename 1..18 without-all>

is the name of the database.

COPY-NUMBER = 1 / <integer 1..999999>

Number of the backup copy.

LOG-FILE-TYPE = *DA-LOG / *CAT-LOG

Type of log file.The name component used is D when *DA-LOG is specified and C when *CAT-LOG C is specified.

SUB-NUMBER-FROM = 1 / <integer 1..9999>

“From” file subnumber.

SUB-NUMBER-TO = 9999 / <integer 1..9999>

“To” file subnumber.The SUB-NUMBER-FROM must be smaller than or equal to the SUB-NUMBER-TO.

The individual parameter OUTPUT

Function

You use this parameter to specify whether SEDI70 is to write the formatted data to SYSLST or a file.

OUTPUT
= *SYSLST / <filename 1..54>

Operands

OUTPUT = *SYSLST

Writes the report to SYSLST.

OUTPUT = <filename 1..54>

Writes the report to the specified file.

> The specified file is always assigned to SYSLST internally and the output is thus always sent to the SYSLST system file. At the end of the program run SYSLST is reset to the primary assignment even when it already had an assignment beforehand.

The individual parameter SELECT

Function

You use this parameter to specify which information is to be output by SEDI70.

SELECT
= *ALL / *PARAMETERS(...) *PARAMETERS(...) | USERS = | ,SPACE-ID =

Operands

SELECT = *ALL

All information is output.

SELECT = *PARAMETERS(...)

The options that allow you to limit the data to be collected are described below individually as SELECT parameters. The information may be limited to one or more timesharing or application users as well as to one space or table.

The SELECT parameter USERS

Function

You use this parameter to specify the timesharing and application users to be output.

```
USERS

= *ALL / *BY-SELECT(...)

  *BY-SELECT(...)

    | HOST-NAME = *NONE / <text 1..8>

    | ,APPLICATION-NAME = *NONE / *TSN (TSN=<alphanum-name 4..4>) / <text 1..8>

    | ,CUSTOMER-NAME = *NONE / <text 1..8>
```

Operands

USERS = *ALL

All timesharing and application users are evaluated.

USERS = *BY-SELECT(...)

Only specific timesharing and application users are evaluated. You must specify a value other than *NONE for at least one operand if *BY-SELECT is specified.

HOST-NAME =

The information to be output is selected by specifying a host system.

HOST-NAME = *NONE

The information to be output is not limited to that on a specific host system.

HOST-NAME = <text 1..8>

Only the information from the user of the specified host system is output. The host system is identified by its symbolic host name. In non-distributed processing, HOMEPROC always identifies the local system.

APPLICATION-NAME =

The information to be output is selected by specifying an application.

APPLICATION-NAME = *NONE

The information to be output is not limited to that of a specific application.

APPLICATION-NAME = *TSN (TSN=<alphanum-name 4..4>)

In TIAM mode you select the application by specifying the task sequence number (TSN). This means that only the information that comes from the TIAM user with the specified TSN is output.

APPLICATION-NAME = <text 1..8>

Only the information that comes from the user of the specified application is output. In the openUTM mode you select the application by specifying the openUTM application name and in the DCAM mode by specifying the DCAM application name.

CUSTOMER-NAME =

The information to be output is selected by specifying a user, i.e. by specifying the user name.

CUSTOMER-NAME = *NONE

The information to be output is not limited to that of a specific user.

CUSTOMER-NAME = <text 1..8>

Only the information from the user of the specified user name is output.The name of the user is independent of the user's mode: In openUTM mode the user is specified with the KDCSIGN name, in TIAM mode with the program name and in DCAM mode with any name.

The SELECT parameter SPACE-ID

Function

You use this parameter to specify the space to be output.

SPACE-ID
= *ALL / <integer 1..1000>(...) <integer 1..32767>(...) | TABLE-ID = *ALL / <integer..32767>

Operands

SPACE-ID = *ALL

The information to be output is not limited to a certain space.

SPACE-ID = <integer 1..1000>(...)

Output is limited to the space with the specified space number. The association between the space number and the space name can be determined from the catalog tables SYS_INFO_SCHEMA.SYS_SPACES and SYS_INFO_SCHEMA.SYS_TABLES from the SPACE_ID and SPACE_NAME fields, respectively.Only the information pertaining to the specified space is output.

TABLE-ID =

The information to be output is limited to that of the specified table.

TABLE-ID = *ALL

The information to be output is not limited to that of a table.

TABLE-ID = <integer 1..32767>

The information to be output is limited to that of the table with the specified table number.The association between the table number and the table name can be determined from the catalog table SYS_INFO_SCHEMA.SYS_TABLES from the TABLE_ID and TABLE_NAME fields.

Only the information pertaining to the specified table is output.

Example

In the following SEDI70 is started with the following parameters:

- The corresponding log files to be evaluated,
- The output format (SYSLST)
- The information to be output

```
/SET-PROCEDURE-OPTIONS -
/               LOGGING-ALLOWED = *YES, -
/               DATA-ESCAPE-CHAR = *STD
/DECLARE-PARAMETER -
/               NAME = ( SYSLST              ( 'LST.SEDI70' ), -
/                        USER-IDENTIFICATION ( '*OWN' ), -
/                        CATALOG-NAME        ( *PROMPT ), -
/                        COPY-NUMBER         ( '000001' ), -
/                        LOG-FILE-TYPE       ( '*DA-LOG' ), -
/                        SUB-NUMBER-FROM     ( '0001' ), -
/                        SUB-NUMBER-TO       ( '0001' ), -
/                        OUTPUT              ( '*SYSLST' ), -
/                        USERS               ( '*ALL' ), -
/                        SPACE-ID            ( '*ALL' ) )
/MODIFY-TERMINAL-OPTIONS -
/               OVERFLOW-CONTROL = *NO-CONTROL
/WRITE-TEXT
'*********************************************************'
/WRITE-TEXT     'Start  "SEDI70"'
/WRITE-TEXT
'*********************************************************'
/REMARK 'Meaning of the procedure parameters'
/REMARK 'SYSLST: SYSLST assignment for the procedure'
/REMARK '&USER-IDENTIFICATION: User ID under which the log files are
stored'
/REMARK '&CATALOG-NAME:Name of the catalog that the log files belong to'
/REMARK '&COPY-NUMBER:Number of the copy that the log files belong to'
/REMARK '&LOG-FILE-TYPE: Type of log files (*CAT-LOG or *DA-LOG )'
/REMARK '&SUB-NUMBER-FROM: Subnumber of the 1st log file to evaluate'
/REMARK '&SUB-NUMBER-TO:Subnumber of the last log file to evaluate'
/REMARK '&OUTPUT: Name of the file that SEDI70 writes its output'
/REMARK '&USERS:User for which the report is to be created'
/REMARK '&SPACE-ID:Space ID for which the report is to be created'
/ASSIGN-SYSDTA  TO-FILE = *SYSCMD
/ASSIGN-SYSLST  TO-FILE = &SYSLST

/START-SESAM-LOG-FILE-EVAL
//CREATE-REPORT INPUT-FILE = *PARAMETERS ( -
//                  USER-IDENTIFICATION = &USER-IDENTIFICATION, -
//                  CATALOG-NAME = &CATALOG-NAME, -
//                  COPY-NUMBER = &COPY-NUMBER, -
//                  LOG-FILE-TYPE = &LOG-FILE-TYPE, -
//                  SUB-NUMBER-FROM = &SUB-NUMBER-FROM, -
//                  SUB-NUMBER-TO = &SUB-NUMBER-TO ), -
//               OUTPUT = &OUTPUT, -
//               SELECT = *PARAMETERS ( -
//                   USERS = &USERS, -
//                   SPACE-ID = &SPACE-ID )
/
/MODIFY-TERMINAL-OPTIONS -
/               OVERFLOW-CONTROL = *USER-ACKNOWLEDGE
```

8.3 SEDI70 report output

Every list that SEDI70 outputs has the following basic structure:

1. General information (window line, title line)
2. File identification records
3. Variable part of the list

8.3.1 General information

The output of the SEDI70 input parameters and the SESAM version of the DBH identifier is structured the same for all reports.

Window line

A window line is output at the beginning of a page of an output file.

Example

```
SEDI70/SQL VERS. <version> TSN:3Q2S ID.:SQLDB111 SEDI70 - LOGGING FILE -
EVALUATION <date> <time> PAGE 1
```

(It appears in a single line without a carriage return.)

Title line

The title line follows the window line.

Example

USER IDENTIFICATION TA ID STMT ACTION SPC TAB REC NR

Explanation

Heading	Information specified
USER-IDENTIFICATION	User ID
TA ID	Transaction identifier
STMT	Statement identifier
ACTION	Type of change: INSERT/UPDATE/DELETE
SPC	SPACE identifier
TAB	Table identifier
REC NR	Internally assigned record number

8.3.2 File identification records

A description of the file identification records follows. File identification records appear when a new log unit has been started

- at the beginning of a new DBH session
- at the beginning of a new file (previous one was full, RECOVER statement, DAW, etc.)
- when a backup copy of a space is made.

Examples

```
SEDI70/SQL   VERS. <version>  TSN:3Q2S  ID.:SQLDB111 SEDI70 - LOGGING FILE - EVALUATION      <date> <time>       PAGE     1
USER IDENTIFICATION          TA ID      STMT     ACTION    SPC   TAB     REC NR
*****************************************************************************************************************************
**********          FILE NAME:  CME.000008.D.0003                                          BLOCK:  00000001          **********
**********          DBH SESSION WITH TRANSACTION-SECURITY = *YES                           <date> <time>  LOC        **********
*****************************************************************************************************************************
```

The following file identification record appears after a DBH restart when a new file is started:

```
>SEDI70/SQL   VERS. <version>  TSN:3Q2S  ID.:SQLDB111 SEDI70 - LOGGING FILE - EVALUATION      <date> <time>      PAGE     3
USER IDENTIFICATION          TA ID      STMT     ACTION    SPC   TAB     REC NR
*****************************************************************************************************************************
**********          FILE NAME:  CME.000008.D.0003                                          BLOCK:  00000001          **********
**********          <<< D B H - R E S T A R T >>>                                          <date> <time>  LOC        **********
*****************************************************************************************************************************
```

All transactions that are not terminated by a COMMIT are rolled back.

8.3.3 Variable part of the list

Primary data

Using the record specifications already described, the primary data changes made are output attribute by attribute.

Example

In the following example some common data types are used.
(The title line was repeated for the sake of clarity).

```
USER IDENTIFICATION          TA ID      STMT     ACTION   SPC   TAB    REC NR
D016ZE09TSN=3Q2QDSQL       0000000B   00000004   INSERT     2    13   00000001
     BAN: 0001   SAN: AAA   POS#:       FCT: N   VAL:  1
                                                       F1404040404040404040
          0002        AAB                    0         A
                                                       C1404040404040404040
          0003        AAC                    0         B
                                                       00424040404040404040404040404040404040404040
          0004        AAD                    0         1 6 . 3
                                                       01630F
          0005        AAE                    0         1 6 . 3
                                                       F0F1F6F3F0
          0006        AAF                    0         2 0 0 6
                                                       000007D6
          0007        AAG                    0         2 0 0 6
                                                       07D6
          0008        AAH                    0         9 . 8 E 7
                                                       475D75C8
          0009        AAJ                    0         9 . 8 E 7
                                                       475D75C800000000
          000A        AAK                    0         9 . 8 E 7
                                                       475D75C800000000
          9001                               0         D 0 U S E R
                                                       C4F0E4E2C5D940404040404040404040404040
D016ZE09TSN=3Q2QDSQL       0000000C   00000007   UPDATE     2    13   00000001
     BAN: 0001   SAN: AAA   POS#:       FCT: -   VAL:  1
                                                       F1404040404040404040
          000C        AAM                    0         1 0 : 4 4 : 3 4 . 3 7 0
                                                       000A002C00220172
D016ZE09TSN=3Q2QDSQL       0000000D   00000009   DELETE     2    13   00000001
     BAN: 0001   SAN: AAA   POS#:       FCT: -   VAL:  1
                                                       F1404040404040404040
```

Explanation

Heading	Contents
BAN	Binary attribute name (the association between the binary attribute name and the column name can be obtained using the catalog table SYS_INFO_SCHEMA.SYS_COLUMNS via the SESAM_BAN and COLUMN_NAME fields).
SAN	Symbolic attribute name
POS#	Position number (only contains data when multiple fields used)
FCT	The action carried out with the following attributes (the hexadecimal string is always interpreted as a character string and is converted): 0: Change to the new attribute value (no information relating to old value) L: Value was deleted N (primary key only): New primary key, i.e. a new record - (primary key only): No change (used to identify the record)
VAL	New attribute value (readable/hexadecimal representation)

Representing primary data

The individual values of a logged modification to primary data are output in both printable and hexadecimal format, e.g. (data type CHARACTER (VARYING)):

```
SAN: AAAPOS#:  FCT: - VAL: H U G O

               C8E4C7D64040404040404040404040404040
```

Non-printable data appears as blanks and in its hexadecimal representation.

Printable data of the data types NATIONAL CHARACTER (VARYING) is converted with the CCSN EFD041 to CHARACTER format and output. Data that cannot be converted appears as blanks. Hexadecimal output in the continuation line corresponds to the Unicode encoding form UTF-16, e.g.:

```
SAN: AAAPOS#:  FCT: - VAL: HUGO
               004800550047004F00200020002000200020020020
```

Secondary data

Secondary data changes are not output!

Transaction borders

All changes made to the affected user are committed or rolled back at a transaction border. Internally the result of the transaction is logged in two steps: If identical transaction IDs (TA_ID) appear, then the transaction is considered to be committed, but if they do not appear, the transaction is considered to be rolled back. This information is presented clearly in the SEDI70 log so that a conclusion can be reached even when files are exchanged between the two steps.

Example

```
D016ZE09TSN=3Q2QDSQL       ========>  PREPARE TO COMMIT (TA ID: 00000000D)
D016ZE09TSN=3Q2QDSQL       ========>
COMMIT WORK       (TA ID: 000000005, TIME: <date> <time> LOC)
D016ZE09TSN=3Q2QDSQL       ========>  ROLLBACK WORK     (TA ID: 00000000D)
```

Canceling a statement

When a statement is canceled, all changes with the corresponding statement identification are rolled back (due to SQL state, INTR CANCEL).

Examples

Of a cancellation:

```
D016ZE09TSN=3Q2QDSQL 0000000E =====> CANCEL STATEMENT
```

Of the normal termination of a statement (only for the option NOTAS):

```
D016ZE09TSN=3Q2QDSQL 0000010B =====> END OF STATEMENT
```

DDL statements

Logging is done using an internal representation. The data is attempted to be formatted similar to SQL statements. The ALTER SPACE statement always appears when the metadata of a space is changed. It is used to modify the space timestamp. The space timestamp is used to check the consistency against the metadata in the catalog space.

Examples of DDL statements that affect the user space

a) CREATE TABLE

```
D016ZE09TSN=3Q2QDSQL       00000002   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       00000002   CREATE TABLE                               (TABLE ID =    13)
                                      USING SPACE SPACE1                         (SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       00000002   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>
D016ZE09TSN=3Q2QDSQL       00000002   ALTER TABLE             (TABLE ID =    13, SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   PKEY                          ( BAN = 0001, SAN = AAA )
                                      CHARACTER(10)                              PRIMARY KEY
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   CHAR01                        ( BAN = 0002, SAN = AAB )
                                      CHARACTER(10),
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   NCHAR13                        ( BAN = 0003, SAN = AAC )
                                      NATIONAL CHAR(10)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   DECI02                        ( BAN = 0004, SAN = AAD )
                                      DECIMAL(5,2)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   NUMI03                        ( BAN = 0005, SAN = AAE )
                                      NUMERIC(5,2)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   INT04                         ( BAN = 0006, SAN = AAF )
                                      INTEGER(31)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   SMALL05                       ( BAN = 0007, SAN = AAG )
                                      SMALLINT(15)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   REAL07                        ( BAN = 0008, SAN = AAH )
                                      REAL(21)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   DOUBLE08                      ( BAN = 0009, SAN = AAJ )
                                      DOUBLE PRECISION(53)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   FLOAT09                       ( BAN = 000A, SAN = AAK )
                                      FLOAT(50)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   DATE10                        ( BAN = 000B, SAN = AAL )
                                      DATE
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   TIME11                        ( BAN = 000C, SAN = AAM )
                                      TIME
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   TIME12                        ( BAN = 000D, SAN = AAN )
                                      TIMESTAMP
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   VARCHAR06                     ( BAN = 9001 )
                                      CHARACTER VARYING(300)
D016ZE09TSN=3Q2QDSQL       00000002   ADD COLUMN   NVARCHAR14                    ( BAN = 9002 )
                                      NATIONAL CHAR VARYING(300)

D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK (TA ID: 000000005)
```

b) ALTER TABLE ADD COLUMN

```
D016ZE09TSN=3Q2QDSQL       0000000D   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       0000000D   ALTER TABLE             (TABLE ID =    13, SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       0000000D   ADD COLUMN   NCHAR10                     ( BAN = 000E, SAN = AAP )
                                      NATIONAL CHAR(10)
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 00000000F)
```

c) ALTER TABLE ALTER COLUMN

```
D016ZE09TSN=3Q2QDSQL       0000000F   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       0000000F   ALTER TABLE             (TABLE ID =    13, SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       0000000F   ALTER COLUMN                               ( BAN = 0006, SAN = AAF )
                                      NATIONAL CHAR(20)
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 000000010)
```

d) ALTER TABLE DROP COLUMN

```
D016ZE09TSN=3Q2QDSQL       00000011   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       00000011   ALTER TABLE             (TABLE ID =    13, SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       00000011   DROP COLUMN                                ( BAN = 0006, SAN = AAF )
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 000000014)
```

e) DROP TABLE

```
D016ZE09TSN=3Q2QDSQL       00000012   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0001, SAN = AAA )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0002, SAN = AAB )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0003, SAN = AAC )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0004, SAN = AAD )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0005, SAN = AAE )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0006, SAN = AAF )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0007, SAN = AAG )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0008, SAN = AAH )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 0009, SAN = AAJ )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 000A, SAN = AAK )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 000B, SAN = AAL )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 000C, SAN = AAM )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 000D, SAN = AAN )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 9001 )
D016ZE09TSN=3Q2QDSQL       00000012   DROP COLUMN                                ( BAN = 9002 )
D016ZE09TSN=3Q2QDSQL       00000012   DROP TABLE                                 ( TABLE-ID = 13 )
                                      USING SPACE SPACE1                         ( SPACE-ID = 2 )
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 000000011)
```

f) Possible data types for ADD COLUMN

```
D016ZE09TSN=3Q2QDSQL       00000013   ADD COLUMN   CHAR011                        ( BAN = 0006, SAN = AAF )
                                                   CHARACTER(10),
D016ZE09TSN=3Q2QDSQL       00000014   ADD COLUMN   NCHAR113                       ( BAN = 000F, SAN = AAQ )
                                                   NATIONAL CHAR(10)
D016ZE09TSN=3Q2QDSQL       00000015   ADD COLUMN   DECI102                        ( BAN = 0010, SAN = AAR )
                                                   DECIMAL(5,2)
D016ZE09TSN=3Q2QDSQL       00000016   ADD COLUMN   NUMI103                        ( BAN = 0011, SAN = AAS )
                                                   NUMERIC(5,2)
D016ZE09TSN=3Q2QDSQL       00000017   ADD COLUMN   INT104                         ( BAN = 0012, SAN = AAT )
                                                   INTEGER(31)
D016ZE09TSN=3Q2QDSQL       00000018   ADD COLUMN   SMALL105                       ( BAN = 0013, SAN = AAU )
                                                   SMALLINT(15)
D016ZE09TSN=3Q2QDSQL       00000019   ADD COLUMN   REAL107                        ( BAN = 0014, SAN = AAV )
                                                   REAL(21)
D016ZE09TSN=3Q2QDSQL       0000001B   ADD COLUMN   FLOAT109                       ( BAN = 0015, SAN = AAW )
                                                   FLOAT(50)
D016ZE09TSN=3Q2QDSQL       0000001C   ADD COLUMN   DATE110                        ( BAN = 0016, SAN = AAX )
                                                   DATE
D016ZE09TSN=3Q2QDSQL       0000001D   ADD COLUMN   TIME111                        ( BAN = 0017, SAN = AAY )
                                                   TIME
D016ZE09TSN=3Q2QDSQL       0000001E   ADD COLUMN   TIME112                        ( BAN = 0018, SAN = AAZ )
                                                   TIMESTAMP
D016ZE09TSN=3Q2QDSQL       0000001F   ADD COLUMN   VARCHAR106                     ( BAN = 9003 )
                                                   CHARACTER VARYING(300)
D016ZE09TSN=3Q2QDSQL       00000020   ADD COLUMN   NVARCHAR114                    ( BAN = 9004 )
                                                   NATIONAL CHAR VARYING(300)
```

SSL statements

Logging is done using an internal representation. The data is attempted to be formatted similar to SQL statements.

Example

a) CREATE SPACE

```
D016ZE09TSN=3Q2QDSQL      00000027    CREATE SPACE SPACE25           (SPACE ID = 6)

D016ZE09TSN=3Q2QDSQL      ========> COMMIT WORK     (TA ID: 000000016)
```

b) DROP SPACE

```
D016ZE09TSN=3Q2QDSQL       00000029   DROP SPACE SPACE25         (SPACE ID = 6)

D016ZE09TSN=3Q2QDSQL       ========> COMMIT WORK (TA ID: 00000001A)
```

c) CREATE INDEX

```
D016ZE09TSN=3Q2QDSQL       00000023   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       00000023   CREATE INDEX                           (INDEX ID =    35)
                                      ON TABLE                               (TABLE ID =    13, SPACE ID =     2)
                                      USING SPACE SPACE1                     (SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 000000013)
```

d) DROP INDEX

```
D016ZE09TSN=3Q2QDSQL       00000025   ALTER SPACE             (SPACE ID =     2, TIMESTAMP= <date> <time>)
D016ZE09TSN=3Q2QDSQL       00000025   DROP INDEX                             (INDEX ID =    34)
                                      ON TABLE                               (TABLE ID =    13, SPACE ID =     2)
                                      USING SPACE SPACE1                     (SPACE ID =     2)
D016ZE09TSN=3Q2QDSQL       ========>  COMMIT WORK      (TA ID: 000000014)
```

9 Error handling

This chapter is concerned with the recovery of errors.

It describes how SESAM/SQL responds to errors and how the system administrator obtains the necessary diagnostics documentation when required.

9.1 Consistency checks

The various components of SESAM/SQL carry out consistency checks within the field of their relevant activities. Any inconsistencies the components find are output to the console or the data display terminal (central error messages) and to SYSLST.

Consistency checks are output in the form *module-name: nn/gg/uu*. where

modulename	is the name of the module or component that detects the error or event
nn	is the number of the consistency check
ww	is the error weight
uu	is the subnumber

module-name, *nn* and *gg* allow the error to be localized in subsequent diagnostics.

The error weight *ww* indicates what effect the error or event has on the current session and, in particular, what state the affected databases are in. It also indicates whether any actions need to be taken and, if so, what needs to be done.

Some consistency checks are output after events that are not necessarily the result of errors and do not require any action to be taken. The administration statement CREATE-DUMP, for instance, is always followed by consistency check SESADXS: XX/05/99 or SESADXS XX/40/99.

Other consistency checks are caused by inconsistencies that the DBH can correct immediately when it is running with transaction management so that no action needs to be taken.

Another group of consistency checks are caused by problems which the system administrator can eliminate with a minimum of effort. These include, for instance, consistency checks that occur following a **DMS error**. The system administrator must first check whether the cause of the reported **DMS error** can be eliminated.

Many consistency checks, however, are caused by errors and require precise diagnostics to be carried out with the relevant diagnostics documentation.

If a consistency check occurs repeatedly, the diagnostic documents are only generated the first time the check occurs. One requirement for the suppression of a second dump is that the same consistency check occurs two times in a row. See also the section “Dump repetition for consistency checks”

The importance of the error weight depends on the SESAM component which reports the consistency check. The message number indicates which component issued the consistency check:

Component	Message number
DBH	`SES6000`
Service task	`SED9001`
DBCON	`SEK0001`
SESDCN	`SEN0001,SEN0003`

Table 37: Components and the corresponding message numbers

9.1.1 Error weight of consistency checks with message numbers SES6000 and SED9001

05	Dump created
10	Transaction rolled back
11	Transaction rolled back because of DMS error
12	Transaction rolled back with application program dump
30	Restart because of system inconsistency
32	I/O error space
33	Maximum space size reached
34	Secondary index defective
36	Table defective
37	Space defective
38	Database catalog space defective
39	Catalog record defective
40	Backup error
50	Timeout during task synchronization
80	System defective
82	System defective with system dump
83	System defective and dump in local task
88	System and current database defective

A brief explanation follows of how the DBH responds to the different error weights when the DBH session is running with transaction management.

Transaction management can be deactivated for a session in which only CALL DML applications are running. Important DBH backup functions are not active in this case and the DBH cannot react appropriately to consistency checks with certain error weights. The user may then have to take more extensive action. This is described separately under the heading “Response”.

The following table contains only consistency checks that require special responses.

Consistency checks that are not listed here are to be handled according to their error weights.

Consistency check	**Detecting functional unit**	**Diagnostic documentation**
SEA....: AD/*gg*/*uu*	SESADM	SESADM dump and maybe a DBH trace
SED....: XX/*gg*/*uu*	DDL functions and Utilityfunctions in the DBH	SESAM dump. SYSLST of the service task
SEDFRVM: XX/*gg*/*uu* SEDFSVM: XX/*gg*/*uu* SESFRVM: XX/*gg*/*uu* SESFSVM: XX/*gg*/*uu*	Free space administration of the space or of the DATABASE KEYS	SESAM dump Storing of the affected Spaces DA-LOGs since the last data backup
SEDTA..: XX/*gg*/*uu*	DDL transaction management in the service task	SESAM dump SYSLST of the service task DDLTA file
SEFBM..: XX/*gg*/*uu* SESLBV: XX/*gg*/*uu*	Transaction logging	SESAM dump, TA-LOG1,

SESLR: XX/*ggl uu* SESSDH: XX/*ggl uu* SESSPM: XX/*ggl uu* SESTACT: XX/*ggl uu*		TA-LOG2, WA-LOG
SEL....: XX/80/*uu* SEL....: XX/88/*uu*	DDL interpreter	SESAM dump and catalog space
SEM....: XM/*ggl uu*	SESAM monitor	SESAM monitor dump
SESADXS: XX/05/99 SESADXS: XX/40/99	Administration handler after administration statement CREATE-DUMP or CREATE-DUMP,STOP	SESAM dump
SESAGPM: AA/*ggl uu* SESAGPC: AA/*ggl uu*	Logical data backup	SESAM dump, DA-LOG, CAT-LOG
SESCO..:XX/05/*uu*	Job logging	Correct DMS error in the CO-LOG file, restart job logging
SESLOMA: XX/*ggl uu*	Recovery	SESAM dump, DA-LOGs, CAT-LOGs, CATALOG space, Cat-Rec file
SEU....: XX/80/*uu* SEU....: XX/88/*uu*	Utility interpreter	SESAM dump and catalog space
SEUTA.. : XX/*ggl uu*	DDL transaction management	DDLTA file

Error weight 05 - Dump created

Meaning

An event has occurred during the run, but processing of a request, the transaction and the DBH session can be continued. Later, however, there may be a need for clarification. A dump is generated for this purpose.

Examples of these events are:

- The DBH has received a defective message.
  In this case the DBH sends a request to the sender task to write a dump.
- An error has occured when writing to the CO-LOG file.
  In this case the DBH turns off the file concerned.

The DBH session continues both during and after this error.

Response

In the case of the CO-LOG file, eliminate the cause of the DMS error. In the case of defective message reception, save the DBH dump and the application program dump and send this with the error message report to customer support. If the errors occurred after the execution of EXPORT or IMPORT statements, you should also save the export file as an additional diagnostic document.

Error weight 10 - Transaction rolled back

Meaning

During processing of a request, an error occurs that does not affect either concurrent requests or the overall system. After rolling back the current transaction, the DBH continues with the session.

Response

Save the dump and send it to customer services along with the error message.

Error weight 11 - Transaction rolled back due to DMS error

Meaning

A DMS error occurred during the processing of a job that does not affect parallel jobs or the overall system. After rolling back the current transaction, the DBH continues with the session.

Response

Recover the DMS error.

Error weight 12 - Transaction rolled back with application program dump

Meaning

An inconsistency between the application program and the DBH has been found that does not affect parallel jobs or the overall system. After rolling back the current transaction, the DBH continues with the session.

Response

Save the DBH dump and the application program dump and send them to customer services along with the error message.

Error weight 30 - Restart due to system inconsistency

Meaning

An inconsistency has occurred in the system which cannot be allocated to any specific job but which can be recovered through a restart. All requester sessions are closed. If transaction management is not active, the session is terminated.

Response

Save the dump and send it to customer services along with the error message.

Error weight 32 - I/O error space

Meaning

After an I/O error in a space the DBH executes an internal restart and rolls back all open transactions so that the space is in a consistent state again. All requester sessions are closed. The DBH session is continued, the space remains in the session. If transaction management is not active in the session, the affected space no longer used.

Response

Recover the hardware problem and inform customer services of the error report. If transaction management was not active in the session, repair the affected space using Media Recovery facilities.

Error weight 33 - Maximum space size reached

Meaning

The maximum possible space size of 64 Gbytes has been reached. The space (created in SESAM/SQL V7.0 or lower) can no longer be expanded. The DBH executes an internal restart and rolls back all the open transactions, which ensures that the space is once more in a consistent state. All requester sessions are closed. The DBH session is continued, the space remains in the session.

Response

Reorganize the space using the utility statement REORG SPACE. Relieve the load on the space by deleting records which are no longer required, by relocating tables and indexes, or by partitioning tables.

Error weight 34 - Secondary index defective

Meaning

The DBH rolls back all open transactions in the course of an internal restart. The index is marked as defective. All requester sessions are closed. The DBH session is continued without the affected index.

Response

Regenerate the defective index using the RECOVER INDEX utility statement and inform customer services of the error message.

Error weight 36 - Table defective

Meaning

A table is inconsistent. Other tables and indexes in the space are generally not involved. The DBH rolls back open transactions by means of an internal restart. The affected table is marked as defective. All requester sessions are closed and the DBH session is continued. The defective table can no longer be accessed.

Response

Report the error message to customer services. If a diagnosis requires information on the space containing the defective table, make a copy of the space. Then use the RECOVER utility statement to repair the space containing the defective table to make the table accessible again.

Error weight 37 - Space defective

Meaning

A space is inconsistent. The DBH rolls back open transactions by means of an internal restart. The defective space is closed and flagged as “defective”. The DBH only opens it again for processing after recovery. All requester sessions are closed. The DBH session continues.

Response

Report the error message to customer services. If a diagnosis requires information on the defective space, make a copy of the space. Then repair the space using the RECOVER utility statement.

Error weight 38 - Database catalog space defective

Meaning

Inconsistency in a catalog space. The DBH rolls back open transactions by means of an internal restart. The database is closed and the catalog space is flagged as “defective”. The DBH only opens the database again for processing after recovery. All requester sessions are closed. The DBH session continues.

Response

Report the error message to customer services. If information on the defective catalog space is required for diagnosis, first make a backup copy of the catalog space, then use the RECOVER CATALOG_SPACE or RECOVER CATALOG ... SCOPE PENDING utility statement to repair it. The advantage of using RECOVER CATALOG ... SCOPE PENDING is that it repairs defective data and index spaces in addition to the catalog space.

Error weight 39 - Database CAT-REC file defective

Meaning

Inconsistency in the CAT-REC file of the affected database. The DBH sets the database to ACCESS=READ, rolls back all open transactions in an internal restart, closes all requester sessions and then continues with the DBH session.

Response

Check whether the CAT-REC file comes from a previous version or whether some other handling error occurred. Make the correct CAT-REC file available, if possible. If the reason for the error was not a handling error, report the error message to customer services. If a diagnosis requires information on the defective CAT-REC file, make a copy of the CAT-REC file and the catalog space. Create a new CAT-REC file with the help of customer services where necessary.

Error weight 40 - Management error

Meaning

Error in transaction management or in error handling. All files opened by the DBH are closed and the DBH is terminated. No internal restart.

Response

If transaction management is active in the session: perform an external restart and report the error message to customer services. In a CALL DML session without transaction management, use the resources of media recovery to repair CALL DML tables being directly updated at the time of termination of the session. Report the error message to customer services.

Error weight 80 - System defective

Meaning

Consistent continuation of the DBH session is not possible due to an error in internal system procedures or in the management tables. The DBH generates a dump and then terminates the session.

Response

Send the dump and the error message to customer services.

Error weight 82 - System defective and system dump

Meaning

As a result of an error in coordination with the operating system, it is not possible for the DBH session to continue in a consistent state. The DBH generates a system dump, if possible, and then terminates the session.

Response

Send the dump and the error message to customer services.

Error weight 83 - System defective and local dump

Meaning

Consistent continuation of the DBH session is not possible due to an error in the internal system procedures which affects the local task data modules. The DBH generates a task dump and terminates the session.

Response

Report the error message to customer services.

Error weight 88 - System and current database defective

Meaning

A catalog space inconsistency indicates that there is a serious error in the administration tables of the DBH. The database is closed and the catalog space is flagged as “defective”. The DBH only opens the database again for processing after recovery. The DBH then proceeds as described for error weight 80 (system defective).

Response

Send the dump and the error message to customer services. If a diagnosis requires information on the defective catalog space, make a copy of the catalog space. Then repair the catalog space using the RECOVER utility statement.

9.1.2 Error weight of consistency checks with message numbers SEN0001, SEN0003 and SEK0001

83	System defective
82	System defective (with system dump)
71	System bottleneck
72	DBH defective
70	Job inconsistent
05	Dump only

The table below shows which diagnostics documents must be obtained for which consistency checks and which actions must be taken.

Consistency check	**Meaning**	**Diagnostic documentation** **Action**
SEC<***>: *nn*/72/*uu*	The communication module DBCON of the application program has detected inconsistencies in the interoperation with the SESAM/SQL-DBH and initiates a remote dump of the DBH	Dump of application program and remote dump of the DBH. Send both dumps to system customer support.
SEC<***>: *nn*/83/*uu* SEC<***>: *nn*/70/*uu* SEC<***>: *nn*/05/*uu*	System defective due to error in the communication system	Dump. Send dump to system customer support
SEC<***>: *nn*/71/*uu*	Error in BS2000 services, e.g. no more free main memory; more precise specification in message SEN3***output at same time	Dump. If possible, eliminate the BS2000 problem. Otherwise send dump to system customer support.

If one of these consistency checks occurs in a DBH or DCN task, a dump is created and the task terminated. A consistency check in an application program task also leads to a dump being created. The relevant application program receives the CDML status 89 or the SQLSTATE 40SA8. Each subsequent request of this application program is also acknowledged with the status 89 or SQLSTATE 40SA8.

When a consistency check occurs, a correction run is executed for the affected common memory pool. If another consistency check occurs during this run, the common memory pool is defective. Any SESAM request that requires this common memory pool for its execution is rejected with the status CDML 9E or with SQLSTATE 81SC5.

9.2 Messages

In addition to consistency checks, SESAM/SQL offers the following messages when errors occur:

- DBH, SESDCN and utility messages output as system messages via BS2000
- CALL DML status messages
- SQL status codes (SQLSTATEs and SQLCODEs)
- precompiler messages

CALL DML status messages and SQLSTATES are described in the “ Messages” manual. You will find precompiler messages in the “ ESQL-COBOL for SESAM/SQL-Server” manual.

9.2.1 Messages in special situations

Status dump

If a dump was requested for a CALL DML or SQL status using one of the administration statements below, message SES6016 is output to inform you of the event.

```
SET-DIAGNOSIS-DUMP-PARAMETER
    DUMP=*ON(SELECT=*CALL-DML(STATE=<alphanum-name 2..2>))
SET-DIAGNOSIS-DUMP-PARAMETER
    DUMP=*ON(SELECT=*SQL(SQLSTATE=<alphanum-name 5..5>))
```

The same applies to a request using the equivalent administration command

`OPT,DIAG,STATUS=<xx>` or `OPT,DIAG,SQLSTATE=<yyyy>`.

The dump is written as a result.

File bottleneck

If a bottleneck occurs in one of the logging files or a DBH cursor file or a space, a filespecific message is output before a restart is initiated. The following rules apply:

File bottleneck	Message number	Inserts supplied
CAT-REC	SES6017	Catalog name
DALOG/CATLOG	SES6018	Catalog name
TALOG1/2	SES6019	-
WALOG	SES6020	-
Cursor file	SES6021	-
Space	SES6026	-

The DBH session is continued after a follow-up transaction restart.

Explanations of the file bottleneck handling

CAT-REC bottleneck or inconsistency

Meaning

There is either an inconsistency between the catalog space and the CAT-REC file or a bottleneck has occurred while writing the CAT-REC file. The exact cause is output first, followed by message SES6017, which announces the restart. During the restart all open transactions are rolled back, all request sessions are closed and only the ACCESS = *PARAMETERS (READ=*YES) access function is permitted for the affected database. After that the DBH continues with the session.

Response

When there is enough space again or the correct CAT-REC file is available again, write access to the database can be authorized again using the MODIFY-CATALOG-ACCESS-RIGHTS administration statement.

DA-LOG/CAT-LOG bottleneck

Meaning

A bottleneck has occurred while writing the DA-LOG file or CAT-LOG file. An internal restart is announced with message SES6018. During the restart all open transactions are rolled back, all job sessions are closed and only the ACCESS = *PARAMETERS (READ=*YES) access function is permitted for the affected database. After that the DBH continues with the session.

Response

When there is enough space again, write access to the database can be authorized again using the MODIFY-CATALOG-ACCESS-RIGHTS administration statement.

TA-LOG bottleneck

Meaning

A bottleneck has occurred while writing to one of the TA-LOG files. An internal restart is announced with message SES6019. During the restart all open transactions are rolled back, all job sessions are closed. This lowers the utilization of the TA-LOG files and the DBH continues with the session.

Response

None

WA-LOG bottleneck

Meaning

A bottleneck has occurred while writing to the WA-LOG file. An internal restart is announced with message SES6020 . During the restart all open transactions are rolled back, all job sessions are closed. This lowers the utilization of the WA-LOG file and the DBH continues with the session.

Response

None

Cursor file bottleneck

Meaning

A bottleneck has occurred while writing to a CURSOR file. An internal restart is announced with message SES6021. During the restart all open transactions are rolled back, all job sessions are closed. This lowers the utilization of the cursor files and the DBH continues with the session.

Response

None

Space file bottleneck

Meaning

The maximum space size has been reached. The internal restart is announced with the message SES6026. During the restart all open transactions are rolled back, all job sessions are closed.

The space is placed in the “reorg pending” state. In the “reorg pending” state only read accesses, deletion of records with DELETE, deletion of tables with DROP TABLE ... DEFERRED, deletion of indexes with DROP INDEX ... DEFERRED and reorganization with REORG SPACE are permitted.

Response

The space’s “reorg pending” state can be reset again using the utility statement REORG SPACE.

9.3 Setting task switches

Task switches are used in BS2000 to control tasks. 32 task-specific switches, which can be set to ON or OFF, are available for each task. At the beginning of the task they are reset (to OFF), but can be changed in the course of the task (SHOW-JOB-SWITCH, see the BS2000 manual “ Commands”). SESAM/SQL uses task switches in error recovery.

Switch 11

The DBH sets switch 11 when errors occur that do not cause the DBH session to abort, e.g. after errors during entry of DBH options or after a consistency check without session abortion. After the DBH session is terminated, a message appears, telling you that the DBH has been terminated with an indication of the reason why.

SESADM sets switch 11 if a status other than “00” is returned.

Switch 12

The DBH sets switch 12 in the case of errors which cause the DBH session to abort. You receive a message to the effect that the DBH has been terminated with an error.

SESADM sets switch 12 in the following cases:

- Formal errors, when user entries lead to abortion (e.g. in the event of errors in the configuration file)
- Resource bottlenecks, when status 2B, 2I, 2M or 85 occurs
- System errors, when a consistency check and/or status 89, 9K or 9E occurs.

SESADM sets task switch but not by evaluating the feedback of a syntactically correct request.

Switch 13

The user can set switch 13 to cause a dump to be created automatically at the end of the program.

9.4 End of session after ABEND events

The DBH includes STXIT routines for specific error events of the STXIT event class, such as program errors or ABEND events (ABEND=Abnormal End).

Before an error event takes effect, the DBH executes an appropriate STXIT routine and tries to terminate the session normally. It then issues the STOP,KEEP admission command (see "STOP-DBH"f) internally. In the case of program errors, a dump is output first.

When an error event occurs in a session with transaction management, the DBH tries to terminate in such a way that the consistency of the data is preserved. If the DBH indicates errors by means of a job variable, an external restart may be necessary.

In a session without transaction management in which only CALL DML applications have been running, CALL DML being directly updated at the time of termination must be repaired by means of media recovery (see the “ Core manual”).

Irrespective of the backup function, the DBH sets switch 12 after the DBH session is terminated, and you receive a message.

9.5 Diagnostics documentation

The causes of errors are not always immediately evident. To diagnose complex errors, you need diagnostics documentation that allows you to establish the status of the system and follow the exact course of the session.

Main-memory dumps and log files are the essential diagnostics documents. In order to receive the necessary documentation in the event of an error, the following BS2000 commands must be included in the sequence of commands for starting the DBH, SESDCN or a SESAM/SQL application:

- `/MODIFY-TEST-OPTIONS DUMP=YES`
  The parameter DUMP=YES in the MODIFY-TEST-OPTIONS command causes each dump to be output and a message is issued to SYSOUT.
- `/MODIFY-JOB-OPTIONS LOGGING=PARAMETERS(LISTING=YES)`
  The LOGGING parameter causes the request to be logged to SYSLST.

In special situations SESAM/SQL not only attempts to create a user dump but also attempts to create a system dump as a troubleshooting aid in cases where interaction with BS2000 is involved. In order to enable this additional dump, the user ID must be permitted to set the test privilege READ-PRIVILEGE to a value of ar least 3.

If this is not possible, as an alternative, SESAM/SQL generates a user dump so that at least this diagnostic document exists. However, error diagnosis will continue to be difficult if no system dump is available.

You can use administration statements or commands to request certain diagnostics utilities (see the overview on "Overview of helpful administration statements"). In the following description, administration statements for administration with SESADM are preferred. However, you can of course use the corresponding administration command in ISP format instead of the administration statement described. These administration commands are specified at the end of the descriptions of their equivalent administration statements in the alphabetical reference section of the manual (see overview on "Equivalent administration statements and commands").

The SESAM/SQL monitor SESMON and the SESCOSP request-logging facility also provide information on database operation that can be useful in diagnosing errors (see chapter “Outputting operational data with SESMON” and chapter “Evaluating request logging with SESCOSP”).

BS2000 diagnostic utilities, such as AID, are described in the relevant BS2000 manuals.

9.5.1 Main-memory dump

One of the most important diagnostics documents is the main-memory dump.

Normally, SESAM/SQL generates a dump automatically. The system administrator can also specifically request a dump.

Automatically generated dump

If a consistency check occurs as a result of a particular error situation, the DBH, SESDCN or the affected DBCON module automatically generates a dump file of the associated task. If necessary, SESAM/SQL also generates a dump file for service tasks (see section “Diagnostics documents for service tasks”).

The name of the dump file contains the following variables:

userid	User ID under which the dump file is stored
jobname	Job name of the DBH, SESDCN or the application program
tsn	Task sequence number of the DBH, SESDCN or the application program
i	Consecutive number identifying the nth dump in the task

Dump file of the DBH or SESDCN task

The dump file can also be created under an alternative identifier if the join entry of the identifier forces it.

The dump files of the DBH, service or SESDCN tasks have the following name:

*$userid.*DUMP[*.jobname*].*tsn.i*

Dumps in multitasking DBH

In general, the multitasking DBH dumps are generated in the start task. Dumps are only generated in the other DBH tasks in some exceptional cases (e.g. in the case of error weight “83”).

Dump file of the application program task

The dump file can be created under an alternative ID.

The dump file is created under the start ID of the user task, and the file name is:

$userid.DUMP[*.jobname*].*tsn.i*

Remote dump

Remote dumps are dumps that occur in a communication partner. They are usually triggered in conjunction with a separate dump in the other communication partner.

Remote dumps occur in the following cases:

- As a result of a consistency check in the SESAM/SQL DBH, one of the consistency checks MM/03/11, MQ/03/22 or MY/03/28 is issued in the application program (the DBCON or DCN communication component), and a remote dump is created.
- As a result of a consistency check in an application program (the DBCON or DCN communication component), the DBH message **SEK9107** is issued and a remote dump is created.

Both the actual dump and the remote dump are required for diagnostic purposes.

Dump repetition for consistency checks

When a consistency check takes place, SESAM/SQL normally generates only one dump. If the same consistency check takes place again, no further dump is generated (dump suppression).

If required, the dump control can be reinitialized using the administration statement SET-DIAGNOSIS-DUMP-PARAMETER DUMP=*INIT (see "SET-DIAGNOSIS-DUMP-PARAMETER"), i.e. when a consistency check takes place a dump is generated again. Subsequently dump suppression once more comes into force.

Conditional dump

You can cause the DBH or the DBCON module of the application program to automatically generate a dump file (conditional dump) when certain SQL DML or CALL DML messages occur.

The name of a dump file like this is the same as that of an automatically generated dump file (see "Main-memory dump"):

$userid.DUMP[*.jobname*].*tsn.i*

$SYSUSER.DUMP[*.jobname*].*tsn.i.userid.*

Causing a conditional dump of the DBH task to be generated

The SET-DIAGNOSIS-DUMP-PARAMETER administration statement allows you to request the DBH, given certain prerequisites, to generate a conditional dump (see "SET-DIAGNOSIS-DUMP-PARAMETER"). You also use this administration statement to specify the message (SQLSTATE or CALL DML status) upon whose occurrence the DBH is to generate the dump and, if necessary, to name a specific user to which the dump is to relate.

After the administration statement is issued, the DBH outputs message SES6016 at each occurrence of the SQLSTATE or CALL DML status and creates a dump file at the first occurrence. A dump file is otherwise only created if another consistency check occurs in the meantime.

Causing a conditional dump of the application-program task to be generated

If the DBCON module of an application program is to generate a conditional dump automatically, in the case of TIAM and DCAM applications, the configuration file of the application program must contain the following connection-module parameter:

DIAG-DUMP={(SQLSTATE=*ccsss*)|(STATUS=*ssuu*)}.

In the case of a UTM application, the connection-module parameter can also be specified as a start parameter of the UTM application

.SESAM DIAG-DUMP={(SQLSTATE=*ccsss*) | (STATUS=*ssuu*)}

cc	class
sss	subclass of the selected SQLSTATE
ss	main number
uu	subnumber of the undesired status

See also the “ Messages” manual.

You can partially qualify SQLSTATE by entering “***” as a subclass. You can partially qualify STATUS by entering “**” as a subnumber.

The DIAG-DUMP parameter is not available with the connection module of an application program working with the linked-in DBH (DBCONL).

Requested dump of the DBH or SESDCN task

If You need another dump at any time during the DBH or SESDCN session in addition to the automatically generated or conditional dumps, You can request it by means of the CREATE-DUMP administration statement (see "CREATE-DUMP").

The name of a requested dump file of the DBH or SESDCN task is the same as that of an automatically generated dump file:

$userid.DUMP[.*jobname*].*tsn.i*

After the DBH administration statement CREATE-DUMP CANCEL-PROGRAM=NO is issued, the DBH outputs the consistency check SESADXS: XX/05/99. After CREATE-DUMP CANCEL-PROGRAM=YES is issued, the DBH outputs the consistency check SESADXS: XX/40/99. The DBH then creates the dump file.

The CREATE-DUMP administration statement directed at SESDCN does not generate a consistency check.

Requested dump of a service task

If You need a dump of a service task You can request it by means of the CREATE-SERVICE-TASK-DUMP administration statement. The name of a requested dump file is:

$userid.DUMP[.*jobname*].*tsn.i*

Other dumps

Dumps can also be generated when running SESAM/SQL that are not announced by messages **SEN0001**, **SEN0003**, **SEK0001**, **SED9001** and **SES6000** .

The reason for such dumps may be:

- Job switch13
  If job switch 13 is set, most of the programs belonging to SESAM/SQL will automatically generate a dump when they are terminated.
- Consistency checks by the SQL runtime system

  All consistency checks issued by the SQL runtime system in the course of an application program cause the SQLSTATE 40SC1 state with message SEW40C1 and a subsequent dump. Relevant diagnostic documentation: SYSOUT log, dump of the application task.

9.5.2 Diagnostics traces

For the purposes of debugging and diagnostics, SESAM/SQL supports the following trace functions:

- the call trace
- the message trace.

The call trace logs SQL DML and CALL DML statements of the application program and the associated SESAM /SQL responses.
The connection module of the application program and the ESQL runtime system log the statement immediately after their transfer from the application program and the responses immediately before their transfer to the application program.

The message trace logs SESAM/SQL messages exchanged between the task of an application program and the task of the SESAM/SQL DBH. In addition to applicationprogram statements and the associated SESAM/SQL replies, the message trace also receives internal SESAM/SQL administration information.
The application program's connection module logs the statement message just before it is sent to the DBH and the reply message shortly after it is received.
The communication component of the DBH logs the messages of all application programs that communicate with the DBH. It logs the statement message immediately after its receipt and the reply message immediately before it is sent.

You can activate the trace as required. There are a number of options:

- TRACE statement as a connection-module parameter or UTM start parameter
- TRACE statement at the CALL DML program interface
- DBH administration statement
- SESDCN administration statement

The activated traces are logged to SYSOUT and/or SYSLST.

TRACE statement as a connection-module parameter

You can cause the call trace and/or message trace to begin logging as soon as a TIAM, DCAM or UTM application program starts. To do this, you must enter the following TRACE statement as a connection-module parameter in the configuration file of the application program:

```
TRACE,TYPE ={CALL/MSG/({CALL/MSG},...)}
[,OUTPUT={SYSOUT/SYSLST/({SYSOUT/SYSLST},...)}]
```

Meanings of the parameters:

TYPE	Selects the type of logging CALL: activates the call trace MSG: activates the message trace
OUTPUT	Selects an output medium SYSOUT: writes the log information to SYSOUT SYSLST: writes the log information to SYSLST (default)

In some cases, logging is activated with a delay and the first statements of the program run are not logged. This happens, for example, when statements lead to errors before the configuration file can be evaluated, or when SQL statements issued at the beginning of the program run are processed completely by the ESQL runtime system.

TRACE statement at the CALL DML program interface

In TIAM and DCAM mode, you can enter the trace statement at the CALL DML program interface to turn call-trace and/or message-trace logging on or off during the session (see the “ CALL-DM Applications” manual). In UTM mode, the TRACE statement is not permitted at the CALL DML program interface.

DBH administration statement

You can use the SET-DBH-MSG-TRACE administration statement (see "SET-DBH-MSG-TRACE") to turn the message trace on or off during the DBH session. This administration statement also allows you to select an output file for the log information.

SESDCN administration statement

During an SESDCN session you can turn the message trace and call trace on or off for specific users. Two administration statements are available for this:

- The SET-USER-MSG-TRACE statement activates the message trace (see "SET-USER-MSG-TRACE").
- The SET-USER-CALL-TRACE statement activates the call trace (see "SET-USER-CALL-TRACE").

Both these statements also let you select an output file for the log information.

9.5.3 Diagnostic information in the DB-DIAGAREA of openUTM

The DB-DIAGAREA is a task-specific trace area into which openUTM cyclically writes trace records. You can find more details on the DB-DIAGAREA in the openUTM manual “ Messages, Debugging and Diagnostics”.

At each call, SESAM/SQL supplies return information to openUTM which is transferred to a trace record in the DB-DIAGAREA. This return information is stored as “DB trace information” in bytes 20-51 of the trace record.

Layout of the DB trace information for SQL requests

Byte in the trace record DB-DIAGAREA	Byte in the DB trace information	Length	Meaning
20	0	1	Identifier “S” for SQL request
21	1	2	Representation of the SQL request (coded)
22	3	1	UTM operation code: • 0x00 = start parameter input • 0x04 = connection • 0x08 = disconnection • 0x0C = asynchronous disconnection • 0x10 = user call • 0x14 = finish DB transaction • 0x18 = cancel transaction • 0x1C = interrupt transaction • 0x20 = continue transaction • 0x24 = status of transaction • 0x28 = preliminary end of transaction • 0x2C = end of process • 0x30 = interrupt process • 0x34 = continue process • 0x38 = restart process
24	4	2	SQL state: • 00 = successful completion • 01 = warning • 02 = no data • 07 = error in dynamic SQL • 21 = cardinality violation • 22 = data exception • 23 = integrity constraint violation • 24 = invalid cursor state • 25 = invalid transaction state

			• 26 = invalid SQL statement name • 28 = invalid authorization specification • 2C = invalid transaction termination • 33 = invalid SQL descriptor name • 34 = invalid cursor name • 3D = invalid catalog name • 3F = invalid schema name • 40 = transaction rollback • 42 = syntax error or access rule violation • 51 = recovery errors • 52 = errors concerning HSMS archive • 55 = errors reported by BS2000 • 56 = BS2000 limits exceeded • 57 = status information • 58 = syntax error in input file • 59 = errors found by check utility • 81 = error in SQL environment • 91 = resource limit exceeded • 95 = invalid foreign transaction state • SB = CLI specific condition
26	6	1	reserved
27	8	1	type of SESAM connection: • 0x01 = with distributed processing • 0x02 = local processing
28	12	4	Message number
32	16	4	transaction serial number
36	20	4	pointer to actual UTAB
40	24	3	target as per distribution rule
43	27	1	DBH configuration name
44	28	4	TSN of the DBH
48	32	1	last digit of the serial number in the CO-LOG file
49	33	3	block number in the CO-LOG file

Table 38: Layout of the DB trace information for SQL requests

Layout ot the DB trace information for CALL-DM requests

Byte in the trace record DB-DIAGAREA	Byte in the DB trace information	Length	Meaning
20	0	3	begin of CALL DML statement
23	3	1	UTM operation code: • as in the table on "Diagnostic information in the DB-DIAGAREA of openUTM"
24	4	2	CALL DML state
26	6	1	status subcode
27	8	1	type of SESAM connection: • 0x01 = with distributed processing • 0x02 = local processing
28	12	4	Message number
32	16	4	transaction serial number
36	20	4	pointer to actual UTAB
40	24	3	target as per distribution rule
43	27	1	DBH configuration name
44	28	4	TSN of the DBH
48	32	1	last digit of the serial number in the CO-LOG file
49	33	3	block number in the CO-LOG file

Table 39: Layout ot the DB trace information for CALL-DM requests

Layout of the DB trace information for UTM system requests

Byte in the trace record DB-DIAGAREA	Byte in the DB trace information	Length	Meaning
20	0	3	identifier “UTM” for UTM system order
23	3	1	UTM operation code: • as in the table on "Diagnostic information in the DB-DIAGAREA of openUTM"
24	4	2	CALL DML state

26	6	1	status subcode
27	8	1	type of SESAM connection: • 0x01 = with distributed processing • 0x02 = local processing
28	12	4	Message number
32	16	4	transaction serial number
36	20	4	pointer to actual UTAB
40	24	3	target as per distribution rule
43	27	1	DBH configuration name
44	28	4	TSN of the DBH
48	32	1	last digit of the serial number in the CO-LOG file
49	33	3	block number in the CO-LOG file

Table 40: Layout of the DB trace information for UTM system requests

9.5.4 DBH-specific log files

CO-LOG file

In addition to the diagnostics traces, SESAM/SQL also provides the CO-LOG file for collecting information on a DBH session. This log file is DBH-specific (see the “ Core manual”). The DBH generates them when logging is turned on.

The CO-LOG file is a log file for request logging. You can use the SET-TUNING-TRACE administration statement to turn request logging on and specify the data volume for the CO-LOG file (see "SET-TUNING-TRACE"):

```
SET-TUNING-TRACE TRACE=*ON(...)
```

You can use the same statement to terminate request logging to the CO-LOG file:

```
SET-TUNING-TRACE TRACE=*OFF
```

TA-LOG and WA-LOG files

The transaction log files (TA-LOG files) and the restart log file (WA-LOG file) are DBHspecific files created by the DBH at the beginning of the session (see the “ Core manual”).

Consistency checks within the transaction restart require the TA-LOG files and the WA-LOG file of the relevant session as additional diagnostics documents along with the dump file.

9.5.5 Overview of helpful administration statements

The table below lists all administration statements that are helpful for generating diagnostics documents. The last column in the table refers you to detailed descriptions of the administration statements and commands in the alphabetical reference section of the manual.

Administration statement (SDF format)	**Administration command (ISP format)**	**Administration of**	**Brief description**	**See**
CREATE-DUMP	DUMP	DBH and SESDCN	Dump created	link
SET-DBH-MSG-TRACE	TRACE	DBH	Controls the message trace.	link
SET-DIAGNOSIS-DUMP-PARAMETER	OPT,DIAG,diag-spec	DBH	Controls the creation of a user dump file.	link
SET-SESSION-DIAGNOSIS	-	DBH	Switches deadlock analysis on and off	link
SET-TUNING-TRACE	COS	DBH	Controls request logging	link
SET-USER-CALL-TRACE	TRACE=PAR, TYPE=CALL... TRACE={ON|OFF...}	SESDCN	Controls the logging of the statements of selected users	link
SET-USER-MSG-TRACE	TRACE=PAR, TYPE=MSG... TRACE={ON|OFF...}	SESDCN	Controls the message trace of selected users	link

Table 41: Administration statements for creating diagnostics documents

9.5.6 Diagnostics documents for service tasks

The SESAM/SQL DBH relocates CPU-intensive activities, such as the sorting of intermediate-result sets or some utility functions, as service tasks (see the “ Core manual”).

The DBH start task logs activities that occur in connection with service tasks, such as the starting and termination of service tasks, to SYSOUT and SYSLST. Service-task-control calls are logged in the DBH stack.

As for the DBH task, SESAM/SQL creates a dump file for service tasks when necessary (see "Main-memory dump").

When SQL messages (SQLSTATEs) occur in service tasks, after certain utility statements, for example, diagnostics information is also stored in the following files:

File name	Meaning
catid.*userid*.DUMP[.*jobname*].*tsn.n*	Dump file of the DBH task
catid.*userid*.DUMP[.*jobname*].*tsn.n*	Dump file of the service task
catid.*userid*.SESAM*cn*.SYSLST. *yyy.mm.ddhh:mm:ss.tsn*	SYSLST log file of the DBH task
catid.*userid*.SESLK*cn*.SYSOUT. *yyy.mm.ddhh:mm:ss.tsn*	SYSOUT log file of the linked-in DBH task created during the RECOVER.
archive-filename.SYSLST	ARCHIVE log file: for SQLSTATEs after a COPY to tape with ARCHIVE or after a RECOVER from tape with ARCHIVE
hsms_archive-filename.SYSLST	HSMS log file: for SQLSTATEs after a COPY to tape with HSMS or after a RECOVER from tape with HSMS
catid.*userid*.*catalog*.*space*.EXC.C	Error file for the CHECK FORMAL utility statement (see also "Error files for SQL and utility statements")
catid.*userid*.*catalog*.*space*.EXC.L	Error file for the LOAD utility statement (see also "Error files for SQL and utility statements")
catid.*userid*.*catalog*.*space*.EXC.U	Error file for the UNLOAD utility statement (see also "Error files for SQL and utility statements")
catid.*userid*.SESAM*kn*.SYSLST.SESST*xx*.*tsn*	SYSLST log file of the service task: for SQLSTATEs after a COPY to tape or a RECOVER from tape
catid.*userid*.SESAM*kn*.SYSOUT.SESST*xx*.*tsn*	SYSOUT log file of the service task: for SQLSTATEs after a COPY to tape or a RECOVER from tape

Table 42: Files with diagnostic information for service tasks

File names

The table 42 contains the default names of all files containing diagnostics information for service tasks. The file names contain the following components:

jobname	Job name of the service task
tsn	Task sequence number of the service task or DBH task
i	Consecutive number identifying the nth dump in the task
c	Configuration name (one byte)
n	DBH name (one byte)
xx	Version number
yyyy.mm.ddhh:mm:ss	Timestamp

9.5.7 EXPLAIN pragma

An SQL access plan is an evaluation rule for an SQL statement (see the “ Core manual”). In certain cases, in the case of unexpectedly long response times, for example, you may need to know the SQL access plan. For diagnostic purposes, you can use the EXPLAIN pragma to generate a readable version of an SQL access plan (see the “ SQL Reference Manual Part 1: SQL Statements”).

9.5.8 Error files for SQL and utility statements

After SQL messages occur (SQLSTATEs), diagnostic information is stored in error files (exception files) after the following UTILITY statements:

- ALTER TABLEThe error file is defined in the SQL statement ALTER TABLE via the USING FILE parameter. Amongst other things, this file contains column values for which conversion errors resulted in data loss due to a change in data type (see the “ SQL Reference Manual Part 1: SQL Statements”).
- CHECK FORMAL {SPACE | TABLE}The error file is created by SESAM/SQL when required. Amongst other things, it contains the respective SQLSTATE which is set on detection of the first inconsistency of the object to be checked (see the “ SQL Reference Manual Part 2: Utilities”).
- LOAD The error file is defined in the utility statement LOAD via the USING FILE parameter. It contains information on the erroneous records of the input file used, and indicates the cause of the error (see the “ SQL Reference Manual Part 2: Utilities”).
- UNLOADThe is created by SESAM/SQL as required (see the “ SQL Reference Manual Part 2: Utilities”).

The error files are SAM files whose data block length is four PAM pages. You can view the error files using the BS2000 command SHOW-FILE or load them using the EDT.

If an SQLSTATE occurs after the CHECK FORMAL utility statement (see the “ SQL Reference Manual Part 2: Utilities”), the copy of the checked space can help in the error analysis in addition to the error file.

10 Appendix

This chapter describes:

- Specifying the sort sequence for CALL DML
- Password protection with SEPA
- Layout of the accounting information
- Layout of the log records for SAT
- Using FASTPAM for SESAM/SQL
- Starting SESAM programs with /START-PROGRAM

10.1 Specifying the sort sequence for CALL DML

SESAM/SQL normally sorts intermediate-result sets of retrieval statements on the basis of the EBCDI code. You can change this sort sequence.

The DBH module SESFS*xx* (*xx*= version number, for SESAM/SQL V9.1 the DBH module is called SESFS91) contains a table that specifies the sort sequence. The entries in this table have the format XL2'*nnmm*', where *nn* and *mm* are characters in hexadecimal notation. An entry in the XL2'*nnmm*' format specifies that the second character (*mm*) follows the first character (*nn*) in the sort sequence. The table has a total of 255 entries, which specify the sort sequence for all 256 characters of the EBCDI code.

If you want to specify a sort sequence other than the default, you can change the table entries in the source file of SESFS*xx*. In doing this, you must note the following:

- In the FSORT macro, you can change only entries that adhere to the format XL2'*nnmm*'.
- You cannot change the number of entries in the table.
- Each character must be assigned a unique place in the sort sequence.

Example

The character “a” (X'81') is to come after “A”(X'C1') in the sort sequence:

Default table	Changed table
.	.
.	.
XL2'7F80'	XL2'7F80'
XL2'8081	XL2'8082'
XL2'8182'	XL2'8283'
XL2'8283'	.
.	.
.	XL2'C0C1'
XL2'C0C1'	XL2'C181'
XL2'C1C2'	XL2'81C2'
XL2'C2C3'	XL2'C2C3'
.	.
.	.

After changing the table, you have to compile the SESFS*xx* module and enter it in the SESAM/SQL module library. For a retrieval statement issued with sorting, the module library must contain a single module called SESFS*xx* at execution time.

> Do not make any other changes to the table. The table is only taken into consideration during the sort if its sort sequence differs from that of the default table. Making changes that do not affect the sort sequence is therefore a waste of time.

10.2 Password protection with SEPA

The SESAM/SQL utility SEPA allows you to use passwords to protect a CALL DML table against unauthorized access. The password catalog is in the same user space as the CALL DML table. Access via the SQL interface is subject to SQL access protection.

In a table that is already protected by passwords, you can use SEPA to add, change or delete passwords.

> If the user space is password protected, the password for the file must be made known to the task by means of the SDF command ADD-PASSWORD before SEPA is called.
> You back up the password catalog by backing up the user space: When recovering the space, you can use the log files created since the last backup to apply the modifications to the space up to the current status, but not the password catalog. Changes to the password catalog are not recorded in the log files. It is therefore advisable to create a SESAM backup copy of the relevant space after each change to the password catalog.
>
> A CALL DML table defined with SQL or migrated from a V1 database without password protection can be accessed via the CALL interface by anyone, provided a SEPA run has not been executed.
>
> A CALL DML table migrated from a protected V1 database is protected by the migrated password catalog.

10.2.1 The password-protection concept

It is possible to set up password protection for the records and attributes of CALL DML tables. You can use a password to limit access to:

- specific records in a table
- specific attributes
- a specific access mode.

The password and information on its access authorization are contained in the password catalog. In the application program, you enter the password in the first three bytes of CALL DML statements.

The CALL DML password-protection concept makes it possible to have hierarchically structured password protection.

It makes sense to assign passwords that apply to a related set of activities: A single password should allow users to carry out all the operations involved in a UTM conversation or a transaction, for example.

You should also note that to execute JOIN operations, read authorization is required for the JOIN attributes of the relevant tables. Access is only possible via the passwords of the tables involved.

Access protection for the records in a table

Each record in a table is identified by a primary-key value.

The access authorization of a password can be assigned for the access modes read, update and delete.

Each access mode is exclusive. Nevertheless, all possible combinations of the different access modes are permitted.

The authorization provided by a password can apply to:

- specific records or groups of records in a table
- all the records in a table
- all the records in a table with the exception of specific records or groups of records

If, in the “read/update” or “read and update” access modes, the password is to provide full access to all the table records to which it applies (i.e. to all the attributes of each record), it must be assigned global attribute authorization for these access modes. Delete authorization refers only to table records. It also applies in the case of attribute restrictions.

Examples

Record access authorization for the read access mode:

The table COMPANY contains the attributes PERSNO, PNAME, PFNAM, PSTR, PZIP, PCITY, PBTHDAT, PDEPT, PLANG and PSAL.

The PERSNO attribute has been defined as the primary key.

The password XX1 has read authorization for the table records with the primary-key values 013751 and 234781. The password XX1 also has global read authorization for all the table's attributes.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
018392									
111111									
234781	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
333333									
. nnnnnn . .									
673241									

Table 43: Read authorization for the password XX1 (read/bold).

The password XX2 has read authorization for all the table records except those with the primary-key values 111111 and 333333. The password XX2 also has global read authorization for all the table's attributes.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
018392	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
111111									
234781	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
333333									
. nnnnnn . .	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**
673241	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**	**read**

Table 44: Read authorization for the password XX2 (read/bold).

Access protection for attributes

The access authorization of a password can be restricted to the access modes read and update.

There is no hierarchical relationship between these two access modes; in other words, update authorization does not include read authorization, and vice versa. However, you can assign a password both types of authorization at the same time.

The authorization provided by a password can apply to:

- specific attributes or ranges of attributes
- all the attributes in the table
- all the attributes in the table with the exception of selected attributes or ranges of attributes

The primary key has a special role in the assignment of attribute authorization. Since primary-key values identify table records, the primary key is not subject to access protection for attributes. To ensure that the attribute authorization of a password also applies to the primary key in a specific access mode, the password must also be assigned global record authorization for this access mode.

Examples

The access authorization for attributes in the read access mode:

The password XX3 has read authorization for the attributes PNAME, PFNAM, PZIP and PCITY and for all the table's records.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**		**read**	**read**				
018392	**read**	**read**		**read**	**read**				
111111	**read**	**read**		**read**	**read**				
234781	**read**	**read**		**read**	**read**				
333333	**read**	**read**		**read**	**read**				
. nnnnnn . .	**read**	**read**		**read**	**read**				
673241	**read**	**read**		**read**	**read**				

Table 45: Read authorization for the password XX3 (read/bold)

The password XX4 has read authorization for all attributes except PBTHDAT and PSAL. The password XX4 also has global read authorization for all the table's records.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
018392	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
111111	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
234781	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
333333	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
. nnnnnn . .	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
673241	**read**	**read**	**read**	**read**	**read**		**read**	**read**	

Table 46: Read authorization for the password XX4 (read/bold)

Combined access protection for records and attributes of a table

You can use a single password to assign access authorization for both records and attributes. You can impose restrictions on record and attribute access that apply to the individual access modes independently of each other. Note, however, that restrictions on record and attribute access in the same access mode overlap.

Examples

Combined access protection in the read access mode:

The password XX5 has read authorization for the attributes PNAME, PFNAM, PZIP and PCITY. The password XX5 also has read authorization for table records with the primary-key values 013751 and 234781.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**		**read**	**read**				
018392									
111111									
234781	**read**	**read**		**read**	**read**				
333333									
. nnnnnn . .									
673241									

Table 47: Read authorization for the password XX5 (read/bold)

The password XX6 has read authorization for all attributes in the table except for PBTHDAT and PSAL. The password XX6 also has read authorization for all the records in the table except for the records with the primary-key values 111111 and 333333.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
018392	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
111111									
234781	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
333333									
. nnnnnn . .	**read**	**read**	**read**	**read**	**read**		**read**	**read**	
673241	**read**	**read**	**read**	**read**	**read**		**read**	**read**	

Table 48: Read authorization for the password XX6 ((read/bold)

The password XX7 has read authorization for the attributes PNAME, PFNAM, PZIP and PCITY. The password XX7 also has read authorization for all the table's records except the one with the primary-key value 111111.

PERSNO	PNAME	PFNAM	PSTR	PZIP	PCITY	PBTHDAT	PDEPT	PLANG	PSAL
013751	**read**	**read**		**read**	**read**				
018392	**read**	**read**		**read**	**read**				
111111									
234781	**read**	**read**		**read**	**read**				
333333	**read**	**read**		**read**	**read**				
. nnnnnn . .	**read**	**read**		**read**	**read**				
673241	**read**	**read**		**read**	**read**				

Table 49: Read authorization for the password XX7 (read/bold)

10.2.2 The SEPA statements

Overview of the statements

1P0 statement:	Controls program run
1P1 statement:	Generates, changes, deletes password
1P2 statement:	Access authorization for attributes
1P2N statement:	Access authorization for all attributes except those in the 1P2N statement
1P3 statement:	Access authorization for a primary-key value or primary-key group
1P3N statement:	Access authorization for all primary-key values except the one specified or for all primary-key groups except the one specified
1P4 statement:	Last value in a range
1P5 statement:	Outputs information on passwords
END statement:	End of the sequence of statements

Overview of the possible SEPA statement sequences

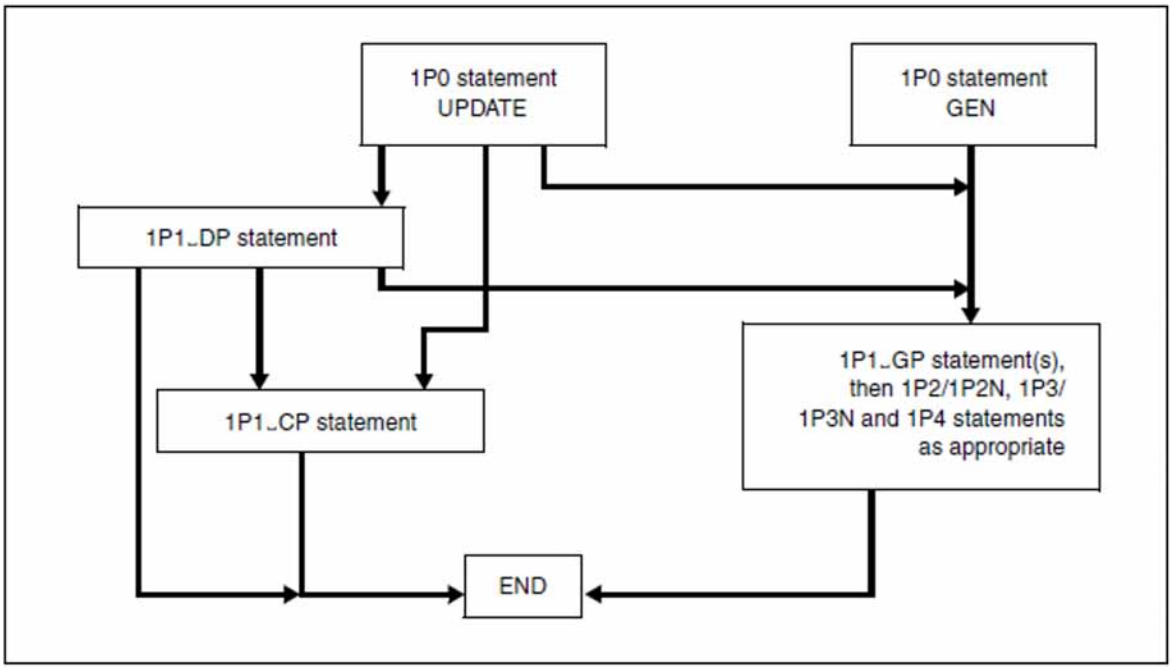


Figure 11: Possible SEPA statement sequences after the 1P0 statement (format 1)

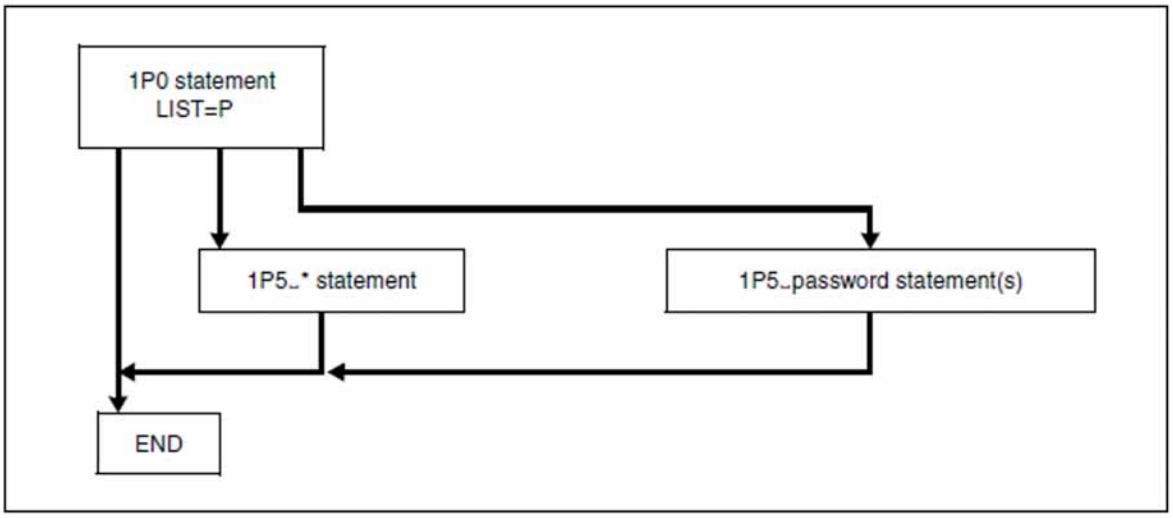


Figure 12: Possible SEPA statement sequences after the 1P0 statement (format 2)

Statement descriptions

This section contains detailed descriptions of the statements of the SESAM/SQL utility SEPA.

10.2.2.1 Statement 1P0 (format 1)

The 1P0 statement specifies which functions SEPA is to execute:

```
1P0'BLANK'NAME='schema;table',SPACE='filename',{GEN | UPDATE}[,LIST=Y]
```

schema	Name of the schema
table	Table name
filename	File name of the space in which the table is located. The password catalog is stored in the same space as the table.
GEN	Generates the password catalog.
UPDATE	Updates the password catalog.
LIST=Y	Lists the generated or updated passwords.

10.2.2.2 Statement 1P0 (format 2)

Lists the passwords for which information is to be output (1P5 statement). Only 1P5 statements are permissible after this statement.

```
1P0'BLANK'NAME='schema;table',SPACE='filename',LIST=P
```

schema	Name of the schema
table	Table name
filename	File name of the space in which the table is located: the password catalog is stored in the same space as the table.
LIST=P	Lists the passwords in the 1P5 statements.

10.2.2.3 Statement 1P1

The 1P1 statement allows you to generate, delete and rename passwords. Please note:

- The 1P1 entry must begin in column 1.
- A statement can contain a maximum of 130 characters.
- Several 1P1'BLANK'GP, 1P1'BLANK'DP and 1P1'BLANK'CP statements can be issued.
- Legal combinations:
  - 1P1'BLANK'DP and 1P1'BLANK'CP statements
    All 1P1'BLANK'DP statements must come before the first 1P1'BLANK'CP statement.
  - 1P1'BLANK'DP and 1P1'BLANK'GP statements
    All 1P1'BLANK'DP statements must come before the first 1P1'BLANK'GP statement.
- Illegal combinations:
  - 1P1'BLANK'CP, 1P1'BLANK'DP and 1P1'BLANK'GP statements
  - 1P1'BLANK'CP and 1P1'BLANK'GP statements

1P1 statements must be entered in ascending order by password.

Generating a password and assigning global authorization

```
1P1'BLANK'GP'BLANK'password[,AR={ E | Y | N }][,OR={ E | Y | N }][,AU={ E | Y | N }]
                                                [,OU={ E | Y | N }][,OD={ E | Y | N }]
```

password

The password is a three-character combination of letters, digits and special characters; it can be specified in printable form or hexadecimal notation. For example, the password AAA in hexadecimal notation is X'C1C1C1'.

When specifying the password in printable form, note the following:

- The password must not contain any NIL characters.
- The first character of the password must not be a blank.
- The password must not begin with the string X'.

When specifying the password in hexadecimal notation, all combinations except X'000000', X'FFFFFF' and X'404040' are permitted.

In the LIST log, every password is listed in both printable form and hexadecimal notation.

The following operands specify the type of access authorization:

AR	Reads attributes in the table records specified by OR.
OR	Reads table records; the attributes read are specified by AR.
AU	Updates attributes in the table records specified by OU.
OU	Updates table records as specified by AU.

OD Deletes records from a table.

The following operands specify authorization values:

E Access authorization is assigned by subsequent 1P2 or 1P3 statements for individual attributes or records of a table.

Y Access authorization applies to all attributes or records of a table.

N Access authorization for all attributes and records of a table is withdrawn from the password for the specified access mode (default).

Default function:

If no authorization value is specified for any access mode, all access authorization is withdrawn from the password.

Example

The password ABC allows all attributes to be read but only some to be updated: `1P1'BLANK'GP'BLANK'ABC,AR=Y,AU=E`

Deleting a password

```
1P1'BLANK'DP'BLANK'password['BLANK'password]...
```

password

Specify the password to be deleted. If you want to delete several passwords, you must separate them with blanks. You can enter the passwords in printable form or hexadecimal notation (see the description of the GP operand).

Example

The passwords AXZ and TTP are to be deleted:

```
1P1'BLANK' DP'BLANK' AXZ'BLANK' TTP
```

Changing a password

```
1P1'BLANK'CP'BLANK'password-old1,password-new1['BLANK'password-old2,password-new2]...
```

You can enter the passwords in printable form or hexadecimal notation (see the description of 1P1'BLANK'GP).

You can use a single 1P1 statement to enter several changes, separated by blanks.

A change cannot be specified separately in two different 1P1 statements.

Example

The password ABB is to be changed to BBB:

```
1P1'BLANK' CP'BLANK' ABB,BBB
```

10.2.2.4 Statements 1P2 and 1P2N

You use these statements to assign access authorization for specific attributes.

Since the primary key is not subject to access protection for attributes, the associated symbolic attribute name AAA cannot be specified in 1P2 or 1P2N statements. In the case of a compound key, however, it is possible to use 1P2 /1P2N statements to assign access authorization to a password for the individual compound-key attributes (symbolic attribute names: AAB, AAC, AAD,...).
After a 1P1 statement you can specify either 1P2 statements or 1P2N statements, but not both.
You can also use a 1P2/1P2N statement to assign authorization for individual attributes that the table does not yet contain but that are to be added subsequently.

The 1P2 statement allows you to assign access authorization for specific attributes or ranges of attributes.

```
1P2'BLANK''BLANK'b{san | san1san2 }['BLANK'b{san | san1san2 }]...
```

This statement assigns access authorization for all attributes in the table except for those attributes or ranges of attributes whose symbolic attribute names are specified in the statement.

```
1P2N'BLANK'[b]{san | san1san2 }['BLANK'{san | san1san2 }]...
```

b Type of authorization:
1 for read
2 for update
3 for read and update

In the 1P2N statement, you must always specify the type of authorization.
If several 1P2N statements are entered for one 1P1 statement, only the first 1P2N statement can contain the authorization type. It is specified before the first attribute name or range. This then applies to all subsequent attributes in the 1P2N statement and all subsequent 1P2N statements that belong to the same 1P1 statement.

san

Symbolic attribute name

san1san2

For san1, specify the name of the first attribute in a range of attributes. For san2, specify the name of the last attribute in the range.

If you specify several symbolic attribute names, they must be in ascending order. Ranges of attributes must not overlap.

Example 1

The attribute with the symbolic name ABC can only be read, the range from CCB to DFG can only be updated, and FGH can be read and updated:

```
1P2'BLANK''BLANK' 1ABC'BLANK' 2CCBDFG'BLANK' 3FGH
```

Example 2

All attributes can be read except ABC, CAF, CAK to DKV, MAN, VWX and XX1. Access authorization for this is specified in two 1P2N statements:

```
1P2N'BLANK' 1ABC'BLANK' CAF'BLANK' CAKDKV1P2N'BLANK' MAN'BLANK' VWX'BLANK' XX1
```

10.2.2.5 Statements 1P3 and 1P3N

You use these statements to assign access authorization for individual records or groups of records in a table.

The records of a table are identified by primary-key values or primary-key-group values.

After a 1P1 statement you can specify either 1P3 statements or 1P3N statements, but not both. You can also use a 1P3/1P3N statement to assign authorization for the primary-key values of records the table does not yet contain but that are to be added subsequently.

The 1P3 statement allows you to assign authorization for an individual record or a group of records in a table.

```
1P3'BLANK''BLANK'b'BLANK'{ primary-key-value | Gprimary-key-group-value}[x]
```

The 1P3N statement assigns the access authorization b for all the records in a table except for those specified in the statement.

```
1P3N'BLANK'[b]{'BLANK'primary-key-value | Gprimary-key-group-value }[x]
```

b Type of authorization:
1 for read
2 for update
3 for read and update
4 for delete
5 for read and update
6 for update and delete
7 for read, update and delete

The type of authorization must always be specified for the 1P3 statement.
If you specify several 1P3N statements after a 1P1 statement, only the first can contain the authorization type. This authorization type then applies to all subsequent 1P3N statements belonging to the same 1P1 statement.

primary-key-value

Specify a primary-key value (up to 64 characters). You can specify it in printable form or hexadecimal notation.
For example, the primary-key value P19350 is X'D7F1F9F3F5F0' in hexadecimal notation.
If the primary-key value begins with the string X', it must be specified in hexadecimal notation.
If the primary key is longer than 64 characters, a follow-up statement is required (see x below). In this case, the primary-key value can only be specified in printable form.

G The operand G applies access authorization to a primary-key group.

primary-key-group-value

Specify a primary-key-group value (up to 64 characters). It can be entered in printable form or hexadecimal notation. If it is longer than 64 characters, a followup statement is required. In this case, it must be specified in printable form.

x Flag indicating there is a follow-up statement:
You can use a maximum of 64 characters in a single 1P3/1P3N statement for the primary-key value. If the primary-key value is longer, you must enter a random character “x” as the 65th character. Enter the rest of the primary-key value in one or more follow-up statements.

Structure of a 1P3 follow-up statement, for example: 1P3'BLANK''BLANK''BLANK''BLANK'psw[x]

Each 1P3/1P3N follow-up statement can contain up to 64 characters. If the primarykey value is longer, you again enter the character “x” as the 65th character to indicate that there is a further follow-up statement.

Example

A primary-key value has 200 characters. The character @ is used to indicate there is a 1P3 follow-up statement:

Statement	1P3'BLANK''BLANK'b'BLANK'z...z@	z...z are the first 64 characters of the primary-key value.
1st follow-up statement	1P3'BLANK''BLANK''BLANK''BLANK'z...z@	z...z are the next 64 characters of the primary-key value.
2nd follow-up statement	1P3'BLANK''BLANK''BLANK''BLANK'z...z@	z...z are the next 64 characters of the primary-key value.
3rd follow-up statement	1P3'BLANK''BLANK''BLANK''BLANK'zzzzzzzz	zzzzzzzz are the last 8 characters of the primary-key value.

10.2.2.6 Statement 1P4

You use this statement to specify the last value in the range of primary keys or primary-key groups whose first value was specified in a preceding 1P3 statement.

You can specify the primary-key value in printable form or hexadecimal notation (see the 1P3 statement). Ranges of primary keys or primary-key groups in 1P3/1P4 statements must not overlap.

```
1P4'BLANK''BLANK''BLANK'{'BLANK'primary-key-value | Gprimary-key-group-value }[x]
```

primary-key-value

Last value in the range of primary keys

primary-key-group-value

Last value in the range of primary-key groups

x

Flag indicating there is a follow-up statement (analogous to the 1P3 follow-up statement)

Example

The primary-key values in the range from ALPHA to BETA can only be read:

```
1P3'BLANK''BLANK'1'BLANK'ANTON
1P4'BLANK''BLANK''BLANK''BLANK' BERTA
```

10.2.2.7 Statement 1P5

You use this statement to output information on passwords.

The 1P5 statement is only permissible if you have specified the 1P0 statement in format 2.

```
1P5'BLANK'{ password['BLANK'password]... | * }
```

password

You can enter the password in printable form or hexadecimal notation (see the 1P1 statement). Up to 17 passwords can be specified.

* Outputs information on all passwords.

The following information is listed for each password specified in a 1P5 statement:

- name (printable and hexadecimal)
- information on global authorization for primary-key values (primary-key values are listed in printable form and in hexadecimal notation)
- information on global authorization for attributes
- information on any individual restrictions imposed on specific attributes
- information on any individual restrictions imposed on specific primary-key values.

10.2.2.8 END statement

You use this statement to terminate the sequence of statements.

```
END
```

10.2.3 Example of password generation

All types of access are to be permitted to the CALL TABLE table COMPANY provided the correct password (ppp) is entered.
If the SEPA statements are entered in a separate file, this file and a start procedure could have the following structure:

Input file

```
1P0 NAME='COMPANY_SCH;COMPANY',SPACE='CALL_COMPANY.COMPANY',GEN
1P1 GP ppp,AR=Y,OR=Y,AU=Y,OU=Y,OD=Y
END
```

Start procedure

```
/BEGIN-PROCEDURE LOGGING=ALL
/OPTION MSG=FHL
/ASSIGN-SYSLST TO-FILE=LST.SEPA
/MOD-MSG-ATTRIBUTES TASK-LANGUAGE=D
/ASSIGN-SYSDTA TO-FILE=input-file
/START-SESAM-CDML-AUTHORIZATION
/SET-JOB-STEP
/ASSIGN-SYSLST TO-FILE=*PRIMARY
/END-PROCEDURE
```

where:

ppp	is the password
input-file	is the name of the input file for the SEPA statements

10.3 Layout of the accounting information

SESAM/SQL creates an accounting record (DBS1 record) for each user session. At the end of the user session, the accounting record is made available in the accounting buffer:

The contents of this buffer are written to the BS2000 accounting file and can be further processed by the RAV accounting procedure.

DBS1 record

Field	Length	Meaning				
1	2	Total record length				
2	2	Free				
3	4	Record ID: “DBS1”				
4	8	Timestamp (STCK format)				
5	2	Length of the user information				
6	2	Length of the basic information: 80 bytes				
7	4	Free				
User information						
8	8	TIAM: accounting number				
9	8	TIAM: BS2000 ID				
10	32	Requesting user: UTM R/L [1] DCAM R/L [1] TIAM R [1] TIAM L [1]	1-8	9-16	17-24	25-32
			host	application	user	TAC name
			host	application	user	program name
			host	TSN=tsn	program name	userid
			request name	TSN=tsn	program name	userid
Basic information						
11	8	Source: q = {R|L} [1]	UTM	SESAMUq		
			DCAM	SESAMDq		
			TIAM	SESAMTq		
12	4	Number of physical file accesses				
13	4	Number of logical file accesses				
14	4	Number of DML statements				

Field	Length	Meaning

15	4 4	Elapsed time in the following format: • seconds • nanoseconds
16	4 2 2 2 2 2 1	Beginning of user session: local time, printable • Year • Month • Day • Hour • Minute • Second • Season: C'W': winter time, C'S': daylight saving time
17	1	Free
18	4 2 2 2 2 2 1	End of user session: local time, printable • Year • Month • Day • Hour • Minute • Second • Season: C'W': winter time, C'S': daylight saving time
19	1	Free
Variable part		
20	8	Host ID of the DBH
21	8	User ID of the DBH
22	8	Account # of the DBH
23	1	Configuration identifier
24	1	Name identifier
25	2	Number of variable fields: 1
26	2	Field distance: X'0000'
27	2	Field identification
28	2	Length of the variable field

29	4 4	CPU time in the following formats: • seconds • nanoseconds

Table 50: DBS1 record

[1]R = remote, L = local

10.4 Layout of the log records for SAT

SESAM/SQL logs security relevant events with the help of the component SAT (Security Audit Trail) of the software product SECOS.
An overview of this function is given in the “ Core manual”, in the section “Logging securityrelevant events with SAT”. For more information about SAT and the SAT evaluation routine SATUT, refer to the “ Security Control System - Audit” manual.

For each security relevant event, a log record (SATLOG record) is written in the protected SATLOG file. The SATLOG file can be analyzed later using the SATUT evaluation routine.

SATLOG records consist of a fixed part which is constantly supplied, and a variable part which contains information about the security relevant event from SESAM/SQL.

The following fields in the fixed part of every SATLOG record are always assigned values:

- user ID and TSN of the DBH task (user-id, tsn)
- logging time (timestp)
- abbreviated name of the event (evt) and result of the event (res)
- chipcard ID if a chipcard is used

Fieldname	**al/** **fil**	**exit**	**Meaning and values of information:** **SDF data type or keywords**
auditid	*	0001	Audit subject identification type: x-string 2..32
evt		00F3	Abbreviated name of the event type: c-string 1..3
res		00F5	Result of the event keywords: F/S for Failure/Success
timestp		00F1	Logging time format: yyyy-mm-dd/hh:mm:ss
tsn		00F4	TSN of the DBH task type: c-string 1..4
user-id		00F6	User ID of the DBH task type: c-string 1..8

Table 51: Field names and values for every SATLOG record

These fields are therefore no longer mentioned in the following SESAM-specific tables. For the meaning of the column headers see "Auditable information for every SESAM object event".

10.4.1 SESAM object events

The table in the following shows the auditable events of the SAT object SESAM, the abbreviated names of the events, the subcodes and indication of their audit attributes.The /MODIFY-SAT-PRESELECTION command enables the security administrator to modify the audit attributes for most events.

The individual columns have the following meanings:

- **OBJECT event** column
  Specification of the object, accompanied by the operations which result in auditable events.
- **Abbreviated name of the event** column
  Each event has a 3-character event name which may be used as a keyword in the commands /SHOW-SAT-STATUS and /MODIFY-SAT-PRESELECTION as well as when analyzing with SATUT (statements //ADD-SELECTION-CONDITIONS and //SELECT-RECORDS).
- **Subcode** column
  Specification of an event. Forms part of the logged information.
- **Audit attribute Chg** column
  Indication of whether the SAT audit attribute for the event can be changed. Y (YES): SAT audit attribute can be changed
- **Audit attribute Dft** column
  Shows the default setting for the SAT audit attribute of the event:
  N: SAT audit attribute NONE, i.e. the event is not logged

OBJECT Event	**Event name evt**	**Subcode for event sessubc**	**Audit attribute Chg**	**Audit attribute Dft**
SESAM Administer DBH session	SEA	ADM	Y	N
Grant/revoke rights	SEP	PRI	Y	N
Add/remove/modify users	SEP	USR	Y	N
DDL statement	SES	DDL	Y	N
SSL statement	SES	SSL	Y	N
Utility statement	SES	UTI	Y	N
Start SESAM-DBH or service task	SET	STRT	Y	N
Stop SESAM-DBH or service task	SET	END	Y	N
Stop process	SEU	END	Y	N

Table 52: Object-related events, abbreviated names, and audit attributes of the SAT object SESAM

10.4.2 Auditable information for every SESAM object event

The table in the following shows a list of the events for the SAT object SESAM, with the associated information fields and their type of output:

M = Mandatory (is always output)

O = Optional (may be output)

- = is not output

Event of the object SESAM	evt	SAT-Information													
		appl nam	appl uid	dbh conf	dbh nam	dbn ame	dbt abl e	hos tna m	sch ema	ses sub c	ses ext	stm tcf	stm tcs	utm sct	utm user
Administer DBH session	SEA	M	O	M	M	-	-	M	-	M	M	-	-	M	M
Change access rights and user accesses	SEP	M	O	M	M	M	-	M	-	M	O	-	-	M	M
DDL, SSL, utility statement	SES	M	O	M	M	M	O	M	O	M	O	-	-	M	M
Start or stop SESAM task (DBH or Service-Task)	SET	-	-	M	M	-	-	-	-	M	M	-	-	-	-
Stop process	SEU	M	O	M	M	-	-	M	-	M	-	M	M	M	M

Table 53: Auditable information of the events of the SAT object SESAM

The table below shows the values present in the various fields.

The field names which can also be monitored via the alarm function of SAT or for which a filter condition can be defined are identified in the second column (al/fil) by means of an asterisk (*). The asterisk (*) mark means that aside from the existence of the field the contents of the field can also be checked.

The identifier in the third column (exit) can be used to edit the SAT information in the audit records when using exit routine 110. The identifiers are specified in hexadecimal notation.

Fieldname	al/ fil	exit	Meaning and values of information: SDF data type or keywords
applnam	*	0025	Application name [1] type: c-string 1..8 In case of a TIAM program the field contains the string 'TSN=<tsn>' where <tsn> stands

			for the tsn of the application program. With logging of an administration statement via /INFORM-PROGRAM this field contains the string ' SEND'.
appluid	*	0162	Application user id type: c-string 1..8 With logging of an administration statement via /INFORM-PROGRAM this field is not filled.
dbhconf	*	0160	DBH configuration identifier [2] type: c-string 1..1
dbhnam	*	015F	DBH name identifier[2] type: c-string 1..1
dbname	*	0165	Name of the database type: c-string 1..18 Logical database name to which the statement relates.
dbtable	*	0167	Table name in the catalog type: c-string 1..31 Table name which is addressed in a statement. This field is only supplied if the name is known at the time of logging.
hostnam	*	0029	Name of the host of the user program [1] type: c-string 1..8 With logging of an administration statement via /INFORM-PROGRAM this field contains the string ' SESAM'.
schema	*	0166	Schema name in the catalog type: c-string 1..31 Schema which is addressed in a statement. This field is only supplied if the name is known at the time of logging.
sessubc	*	015E	Subcode for the SESAM events type: c-string 1..4 The following subcodes can be evaluated, depending on the event concerned: **Event SET** • STRT: Start of a SESAM task (DBH or service task) • END: End of a SESAM task (DBH or service task) **Event SEU** • END: End of process **Event SEA** • ADM: Administer DBH session **Event SEP** • USR: Add/remove/modify users • PRI: Grant/revoke rights **Event SES**

			• DDL: DDL statement • SSL: SSL statement • UTI: utility statement
sestext	*	0168	Additional information about the SESAM event type: c-string 1..64 Depending on the event, the following additional information is given, if it exists, at the time of logging: • 'DBH-Task' when a DBH task has been started or terminated • 'Service-Task' when a service task has been started or terminated • The first 64 characters of an adminisration statement • Brief statement texts such as 'ALTER_TABLE' or 'COPY_CATALOG' in the case of DDL, SSL or utility statements. If the statement has altered the catalog, the statement number, the internal TAID and number and subnumber of the catalog will also be issued. This information is used to understand the changes with the catalog.
stmtctf	*	0164	Number of unsuccessful statements in the process type: integer 0..2^{31}-1 A statement is unsuccessful if it is not acknowledged with “successful completion”, “no data” or “rollback”.
stmtcts	*	0163	Number of successful statements in the process type: integer 0..2^{31}-1 A statement is successful if it is acknowledged with “successful completion”, “no data” or “rollback”.
utmsct	*	0161	UTM session counter [1] type: x-string 2..16 (ALARM/FILTER: x-string 16..16)
utmuser	*	0048	UTM user name [1] type: c-string 1..8 With logging of an administration statement via /INFORM-PROGRAM this field contains the string ' MESSAGE'.

Table 54: Field names and values for the SAT object SESAM

[1]A specific process can be identified with the hostnam, applnam, utmuser and utmsct fields

[2]The SAT log records for all tasks of a database handler can be identified with the dbhnam and dbhconf fields

10.5 Using FASTPAM for SESAM/SQL

SESAM/SQL uses the FASTPAM access method for all files except for the database files that are stored on shared private disks (SPD).

These files are still processed by SESAM/SQL with the UPAM access method because the type of disk does not allow you to use FASTPAM.

Accessing the disks with FASTPAM yields higher performance than accessing them with UPAM because FASTPAM prepares certain paths and resident areas are created.

In order to create resident area the BS2000 user ID requires FASTPAM authorization. Whether or not the BS2000 user ID has the corresponding authorization is determined via the BS2000 command SHOW-USER-ATTRIBUTES. The DMS-TUNING-RESOURCES field must be set to the value *EXCLUSIVE.

FASTPAM can be used even though the BS2000 user ID does not have FASTPAM authorization, but in this case the areas are not stored in resident memory. This also applies to when the BS2000 user ID has FASTPAM authorization, but not all the required areas (I/O buffer and administration information) can be stored in resident memory because there is not enough main memory free.

If the attempt to load SESAM/SQL is rejected in spite of there being a sufficiently high value for RESIDENT PAGES in the BS2000 user ID, the BS2000 system administrator may have to increase the maximum number of resident main memory pages allowed accordingly (BS2000 command MODIFY-SYSTEM-BIAS). You can query the maximum value set (CORE) by using the BS2000 command SHOW-SYSTEM-STATUS with the INFORMATION=*SYSTEM-PARAMETERS parameter.

You can enter the required number of resident pages in the START command for the DBH start in the RESIDENT-PAGES parameter, see "Operands of the start commands". You must specify a minimum and a maximum value.

If you specify a minimum value = 0, SESAM/SQL will start in any case, although performance may suffer. If you specify a minimum value > 0, SESAM/SQL will only start if the minimum requirement can be satisfied.

The specifications for some DBH options are relevant to the calculation of the maximum value, see the formula on "Using FASTPAM for SESAM/SQL".

In addition to the size of the resident memory, the number of required FASTPAM parameter lists (FPAMACC) also represents a limit for FASTPAM. The number of parameter lists required by SESAM/SQL depends on the DBH options.

FASTPAM permits a maximum of 5000 parameter lists. This drops to 500 parameter lists if SESAM/SQL is working without resident areas. This is the case in the absence of FASTPAM authorization or when insufficient resident memory is available for the BS2000 task.

If the limits for the maximum permitted number of parameter lists are exceeded, SESAM/SQL issues a corresponding message and switches automatically to UPAM.

Additional information on the FASTPAM access method can be found in the “ Introductory Guide to DMS”.

Formula for the number of resident pages required:

Number of resident pages (rounded to 4KB units) =

Value of the option CURSOR-BUFFER=*PARAMETERS(BUFFER-SIZE) / 4+ 68 * Number of threads (see Comments on the formula)
+ 33 * Number of entries in the database catalog (option SQL-DATABASE-CATALOG)+ 305
+ Size for FPAMACCs (FASTPAM parameter lists, see Comments on the formula)

Comments on the formula

- If the value of the option CURSOR-BUFFER=*PARAMETERS(BUFFER-SIZE=) / 4) is less than (8 * Number of threads), then (8 * Number of threads) is used.

  The number of threads is calculated here using the following formula
  Number of threads =
  Value of the option THREADS
  + Value of the option SYSTEM-THREADS=*PARAMETERS(WRITE-THREADS)+ 6 + D + S + R

  where

  - the value for D is calculated according to the table below:

Value of the option THREADS	**Value for D**
up to 32	1
up to 64	2
up to 96	3
... (in increments of 32)	... (+1)
up to 1024	32

  - the value for S is calculated according to the table below:

Number of entries in the database catalog (option SQL-DATABASE-CATALOG)	**Value for S**
up to 4	1
up to 8	2
up to 12	3
... (in increments of 4)	... (+1)
up to 36	9
over 36	10

  - the value for R is calculated according to the table below:

Value of the option USERS	**Value for R**

up to 3072	3
up to 4096	4
up to 5120	5
... (in increments of 1024)	... (+1)
up to 32767	32

- The size for FPAMACCs is calculated according to the table below:

  Size of FPAMACCs (rounded to 4KB units) =
  36 bytes * Value of the option DBH-TASKS
  * (Number of threads + 3 * Value of the option SQL-DATABASE-CATALOG + 100) / 4096

10.6 Starting SESAM programs with /START-PROGRAM

All SESAM/SQL programs are started using SESAM start commands for the specific programs; see "Overview of the start commands". It is assumed that you have installed SESAM/SQL and CRTE with IMON or that the files required have been set up under their default filenames (see "Tasks of the start commands").

For reasons of compatibility, you can continue to use the command START-PROGRAM to start the SESAM programs. Additional information about the program-specific sections for DBH, SESADM and utility monitor is contained in this section.
You will also find the command sequences mentioned in the start files of the demonstration database MAN-DB; see the “ Core manual”.

> When SESAM programs are started using /START-PROGRAM the following operand must be specified:
> `RUN-MODE=*ADVANCED(UNRESOLVED-EXTRNS=*DELAY,`
> `LOAD-INFORMATION=*REFERENCES)`

Version coexistence

If SESAM programs are started by means of the /START-PROGRAM command, the most recently installed correction delivery status is loaded, if you do not specify the VERSION parameter. If you do specify the parameter, the specified version must match the version of the start LLM in the SESAM module library.

Starting the DBH

You should also note the following before starting the SESAM/SQL DBH with /START-PROGRAM:

- The SESAM/SQL DBH loads modules dynamically. When dynamically loading the first module, it searches the following module libraries in the order specified:
  1. The module library, which is assigned the link name SESAMOML by means of the /ADD-FILE-LINK command
  2. The module library assigned by means of /SET-TASKLIB LIBRARY=*modlib*.This module library is ignored if *modlib* is the TASKLIB library of the default user ID
  3. Hardware dependent module library SYSLNK.SESAM-SQL.<ver> or SKULNK.SESAM-SQL.<ver> of the calling user ID.
  4. Hardware dependent SYSLNK.SESAM-SQL.<ver> or SKULNK.SESAM-SQL.<ver> module library of the default user ID (the user ID specified with the BS2000 system parameter DEFLUID)
  5. The TASKLIB module library of the default user ID

  All subsequent modules are loaded from the same library as the first one. This means you must make all SESAM /SQL modules available in the same module library.
- SYSLNK.SESAM-SQL.<ver> is the default name of the module library of SESAM/SQL for/390 servers. SKULNK.SESAM-SQL.<ver> is the default name of the module library of SESAM/SQL for x86 servers.

  If you use another name, or if the module library is not under either the calling user ID or the default user ID, you have to assign the module library with the link name SESAMOML (ADD-FILE-LINK LINK-NAME=SESAMOML, FILE-NAME=*sesam-modlib*). If the RELOAD-DBH-SESSION statement is to be used in order to load a new correction version of the DBH modules without interrupting ongoing operation, the file thus assigned should be overwritten with the new correction version.
- The hardware dependent CRTE library is required to run the SESAM/SQL DBH. You must always assign the CRTE library under the link name

BLSLIB*xx* before starting the DBH, if you have not installed CRTE with IMON. The placeholder *xx* stands for a two-digit serial number (00 £ *xx* £?99).

Command sequence for starting the DBH (/390 servers)

```
/ADD-FILE-LINK LINK-NAME=SESAMOML, FILE-NAME=$TSOS.SYSLNK.SESAM-SQL.<ver>
/ADD-FILE-LINK LINK-NAME=BLSLIB01, FILE-NAME=$TSOS.SYSLNK.CRTE
/START-PROGRAM FROM-FILE=*MODULE -
/ (LIBRARY=$TSOS.SYSLNK.SESAM-SQL.<ver> -
/ ,ELEM=SESAM -
/ ,PROG-MODE=ANY -
/ ,RUN-MODE=ADVANCED(UNRESOLVED-EXTRNS=*DELAY,LOAD-INFORMATION=*REFERENCES))
```

Starting SESADM (/390 servers)

```
/ADD-FILE-LINK LINK-NAME=SESAMOML, FILE-NAME=$TSOS.SYSLNK.SESAM-SQL.<ver>
/ADD-FILE-LINK LINK-NAME=BLSLIB01, FILE-NAME=$TSOS.SYSLNK.CRTE
/START-PROGRAM FROM-FILE=*MODULE -
/ (LIBRARY=$TSOS.SYSLNK.SESAM-SQL.<ver> -
/ ,ELEM=SESADM -
/ ,PROG-MODE=ANY -
/ ,RUN-MODE=ADVANCED(UNRESOLVED-EXTRNS=*DELAY,LOAD-INFORMATION=*REFERENCES))
```

Starting the utility monitor (/390 servers)

```
/ADD-FILE-LINK LINK-NAME=MAPLIB,FILE-NAME=$TSOS.SYSFHS.SESAM-SQL.<ver>.UTI.D
/ADD-FILE-LINK LINK-NAME=SESAMOML,FILE-NAME=$TSOS.SYSLNK.SESAM-SQL.<ver>
/ADD-FILE-LINK LINK-NAME=BLSLIB01,FILE-NAME=$TSOS.SYSLNK.CRTE
/ADD-FILE-LINK LINK-NAME=SEEHELP,FILE-NAME=$TSOS.SYSMAN.SESAM-SQL.<ver>.UTI.D
/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=SESCONF.SESUTI.ZX -
/                               ,ACCESS-METHOD=SAM
/START-PROGRAM FROM-FILE=*MODULE(LIBRARY=$TSOS.SYSLNK.SESAM-SQL.<ver>    -
/                        ,
ELEMENT=SESUTI
-
/                        ,PROG-MODE=ANY                                  -
/                        ,RUN-MODE=ADVANCED(ALTERNATE-LIBRARIES=YES      -
/                        ,UNRESOLVED-EXTRNS=*DELAY                       -
/                        ,LOAD-INFORMATION=*REFERENCES))
```

Starting the performance monitor SESMON (SX servers)

```
/ADD-FILE-LINK LINK-NAME=MAPLIB,FILE-NAME=$TSOS.SYSFHS.SESAM-SQL.<ver>.MON.E
/ADD-FILE-LINK LINK-NAME=SESAMOML, FILE-NAME=$TSOS.SPULNK.SESAM-SQL.<ver>
/ADD-FILE-LINK LINK-NAME=BLSLIB01, FILE-NAME=$TSOS.SPULNK.CRTE
/MODIFY-DBL-DEFAULT PRIORITY=*FORCED, -
/  SCOPE=*PROGRAM-CALLS(LOADING=*PARAM,RESOLUTION=*PARAM(MMODE-CHECK=NO))
/START-PROGRAM FROM-FILE=*MODULE -
/ (LIBRARY=$TSOS.SPULNK.SESAM-SQL.<ver> -
/ ,ELEM=SESMON -
/ ,PROG-MODE=ANY -
/ ,RUN-MODE=ADVANCED(UNRESOLVED-EXTRNS=*DELAY,LOAD-INFORMATION=*REFERENCES))
```

11 Related publications

You will find the manuals on the internet at http://manuals.ts.fujitsu.com. You can order printed versions of manuals which are displayed with the order number.

SESAM/SQL-Server (BS2000)
SQL Reference Manual Part 1: SQL Statements
User Guide

SESAM/SQL-Server (BS2000)
SQL Reference Manual Part 2: Utilities
User Guide

SESAM/SQL-Server (BS2000)
CALL-DM Applications
User Guide

SESAM/SQL-Server (BS2000)
Core manual
User Guide

SESAM/SQL-Server (BS2000)
Utility Monitor
User Guide

SESAM/SQL-Server (BS2000)
Messages
User Guide

SESAM/SQL-Server (BS2000)
Performance
User Guide

WebTA access for SESAM/SQL
(Product document, also available on the manual server)

ESQL-COBOL (BS2000)
ESQL-COBOL for SESAM/SQL-Server

User Guide

SESAM-DBAccess
Server-Installation, Administration (available on the manual server only)

SDF (BS2000)
SDF Dialog Interface
User Guide

BS2000 OSD/BC
Commands
User Guide

SDF-P (BS2000)
Programming in the Command Language
User Guide

BS2000 OSD/BC
Introduction to System Administration
User Guide

BS2000 OSD/BC
Introductory Guide to DMS
User Guide

BS2000 OSD/BC
Executive Macros
User Guide

ARCHIVE (BS2000)
User Guide

HSMS (BS2000)
Hierarchical Storage Management System
User Guide

EDT (BS2000)
Statements
User Guide

BS2000 OSD/BC
Subsystem Management (DSSM/SSCM)
User Guide

JV (BS2000)
Job variables
User Guide

SECOS (BS2000)
Security Control System - Audit
User Guide

SORT (BS2000)
SDF Format
User Guide

SNMP Management
SNMP Management für BS2000
User Guide

openUTM (BS2000)
Messages, Debugging and Diagnostics
User Guide

openSM2 (BS2000)
Software Monitor
User Guide