

FUJITSU Software BS2000

SESAM-DBAccess V9.1 Server-Installation, Administration

Administratorhandbuch

Oktober 2018

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

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

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

Zertifizierte Dokumentation nach DIN EN ISO 9001:2015

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

Copyright und Handelsmarken

Copyright © 2018 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

Inhaltsverzeichnis

SESAM-DBAccess	4
1 Einleitung	5
1.1 Zielsetzung und Zielgruppen des Handbuchs	7
1.2 Konzept des Handbuchs	8
1.3 Änderungen gegenüber dem Vorgänger-Handbuch	9
1.4 Darstellungsmittel	10
2 Einsatz des SESAM DBAccess-Servers	11
3 Benutzerverwaltung	13
3.1 Benutzerprüfung	14
3.2 Die Erfassungsprozedur SYSPRC.SESDBA.091.USER	15
4 Starten und Beenden	17
4.1 Voraussetzungen zum Starten	18
4.2 Starten des DBAccess-Servers	20
4.3 Beenden des DBAccess-Servers	21
4.4 Jobvariablen des DBAccess-Servers	22
5 Betrieb des DBAccess-Servers	23
5.1 Initialisierungsdatei (Startparameter)	24
5.2 EBCDIC-ASCII-Konvertierung	28
5.3 Informationsfunktionen	30
5.3.1 Protokollierung in der Session Log Datei	31
5.3.2 Auskunft über aktuelle Verbindungen	34
5.3.3 Auskunft über aktuelle Parameter	36
5.4 Sichere Verbindungen über SSL	37
6 Verhalten im Fehlerfall	39
6.1 SYSOUT-Protokoll	40
6.2 Benutzer-Protokolldatei	41
7 Anhang	43

SESAM-DBAccess

Deutsch

1 Einleitung

SESAM-DBAccess besteht aus Client-Komponenten und einer Server-Komponente.

JDBC-Client

Der JDBC-Client bietet dem Benutzer jeder Plattform, auf der eine Java Virtual Machine vorhanden ist (z.B. Windows, Unix-Systeme, BS2000), den Zugriff aus Java-Anwendungen, Java Servlets und Java Applets über die JDBC-Schnittstelle auf eine SESAM/SQL-Datenbank.

JDBC ist dabei eine Java-API zur Ausführung von SQL-Anweisungen. Es besteht aus einer Menge von in Java geschriebenen Klassen und Schnittstellen. JDBC bietet eine StandardAPI, die es Entwicklern von Java-Anwendungen, Java Servlets und Java Applets ermöglicht, eine reine Java-Schnittstelle für den Zugriff auf Datenbanken zu verwenden.

Der JDBC-Client hat unabhängig von der Systemplattform, auf der er läuft, folgende Aufgaben:

- Abwickeln der Verbindung zu einem DBAccess-Server
- Weiterleiten der Anforderungen an den DBAccess-Server
- Entgegennehmen der Antworten des DBAccess-Servers

ADO.NET-Client

ADO.NET ist die von Microsoft im Rahmen der ADO-Technologie (ActiveX Data Objects) definierte Schnittstelle für .NET-Clients zur Kommunikation von Windows-Systemen mit Datenbanksystemen über das Internet.

Der ADO.NET-Client hat folgende Aufgaben:

- Abwickeln der Verbindung zu einem DBAccess-Server
- Weiterleiten der Anforderungen an den DBAccess-Server
- Entgegennehmen der Antworten des DBAccess-Servers

PDO-Client für PHP-Anwendungen

PHP (**PHP: Hypertext Preprocessor**) ist eine Skriptsprache mit einer an C und Perl angelehnten Syntax, die hauptsächlich zur Erstellung dynamischer Webseiten oder Webanwendungen verwendet wird. PHP-Anwendungen dienen u.a. der Kommunikation mit Datenbanksystemen über das Internet.

Der PDO-Client hat folgende Aufgaben:

- Abwickeln der Verbindung zu einem DBAccess-Server
- Weiterleiten der Anforderungen an den DBAccess-Server
- Entgegennehmen der Antworten des DBAccess-Servers

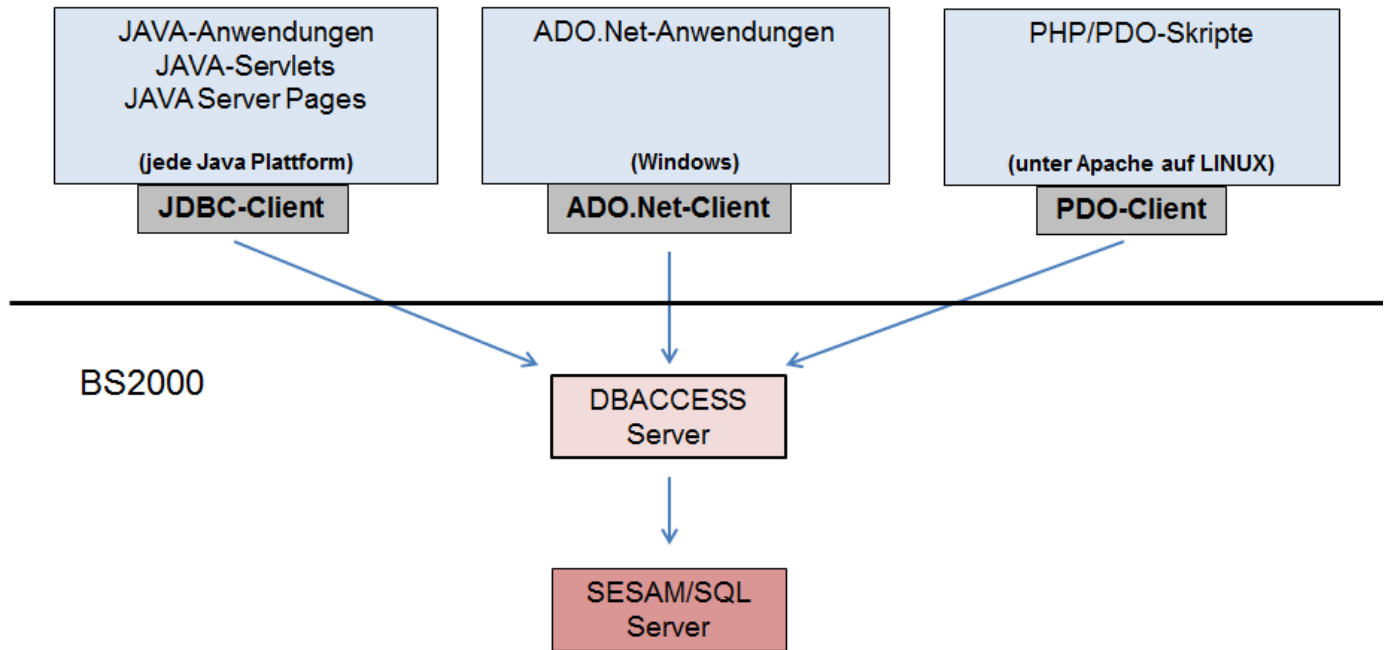
DBAccess-Server

Der DBAccess-Server (BS2000) bietet JDBC-, .NET- und PDO-Clients den Zugriff auf das Datenbanksystem SESAM/SQL.

Der DBAccess-Server hat folgende Aufgaben:

- Überprüfen der Zugriffsberechtigung
- Entgegennehmen von Aufträgen der Clients und Weiterleiten an das Datenbanksystem
- Entgegennehmen der Antworten des Datenbanksystems und Weiterleiten an den Client

- Abwickeln der Kommunikation mit dem Datenbanksystem



Überblick über die Konfiguration des Gesamtsystems

1.1 Zielsetzung und Zielgruppen des Handbuchs

Das Handbuch richtet sich an den Systembetreuer des DBAccess-Servers in BS2000. Erforderlich sind Kenntnisse des Datenbanksystems SESAM/SQL.

1.2 Konzept des Handbuchs

Das vorliegende Handbuch beschreibt die Server-Komponente von SESAM-DBAccess im BS2000.

Das Kapitel „Einsatz des SESAM DBAccess-Servers“ gibt Hinweise zum Einsatz des DBAccess-Servers (Software-Voraussetzungen und Architektur).

Das Kapitel „Benutzerverwaltung“ beschreibt die Erfassung der zugelassenen DBAccessBenutzer.

Das Kapitel „Starten und Beenden“ beschreibt den Start und das Beenden des DBAccessServers.

Das Kapitel „Betrieb des DBAccess-Servers“ beschreibt Einstellungen, die den Betrieb des DBAccess-Servers beeinflussen. Ferner sind in diesem Kapitel die Funktionen beschrieben, die Auskunft über die Client-Verbindungen und den DBAccess-Server geben.

Das Kapitel „Verhalten im Fehlerfall“ enthält Hinweise, die für den Administrator im Fehlerfall von Bedeutung sind.

Im Anhang finden Sie eine Übersicht über die vom DBAccess-Server erzeugten und benötigten Dateien, Prozeduren, Bibliotheken und Programme.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Folgende wichtigen Änderungen haben sich gegenüber dem Vorgänger-Handbuch ergeben::

- Unterstützung von sicheren Verbindungen über SSL

1.4 Darstellungsmittel

In diesem Handbuch wird kurz der Begriff **Clients** für die JDBC-, .NET- und PDO-Clients verwendet.

In diesem Handbuch werden folgende Darstellungsmittel verwendet:

i für Hinweise auf besonders wichtige Informationen

2 Einsatz des SESAM DBAccess-Servers

Hardware- und Software-Voraussetzungen

Der DBAccess-Server V9.1 kann auf allen Servern unter BS2000/OSD-BC ab V9.0 ablaufen..

Die aktuell erforderlichen Softwareprodukte sind der Freigabemitteilung zu entnehmen.

Architektur des DBAccess-Servers

Der DBAccess-Server wird mit `/START-SESDBA-SERVER` im Allgemeinen in einer Batchtask gestartet. Diese Task wird im Folgenden **Admintask** genannt. Die Admintask erzeugt nach dem Start eine weitere Task, die im Folgenden **Verteilertask** genannt wird.

Die Verteilertask erzeugt weitere Tasks, die sogenannten **Servertasks**.

Servertasks unterteilen sich in zwei Gruppen:

- **Freie Servertasks** heißen Tasks, die gestartet und initialisiert wurden, aber noch keine Verbindung angenommen haben.
- **Aktive Servertasks** heißen Tasks, die schon eine Verbindung angenommen haben.

Admintask

An die Admintask werden Administrationsaufträge (z.B. Beenden des Servers) gerichtet, die auch innerhalb dieser Task bearbeitet werden.

Verteilertask

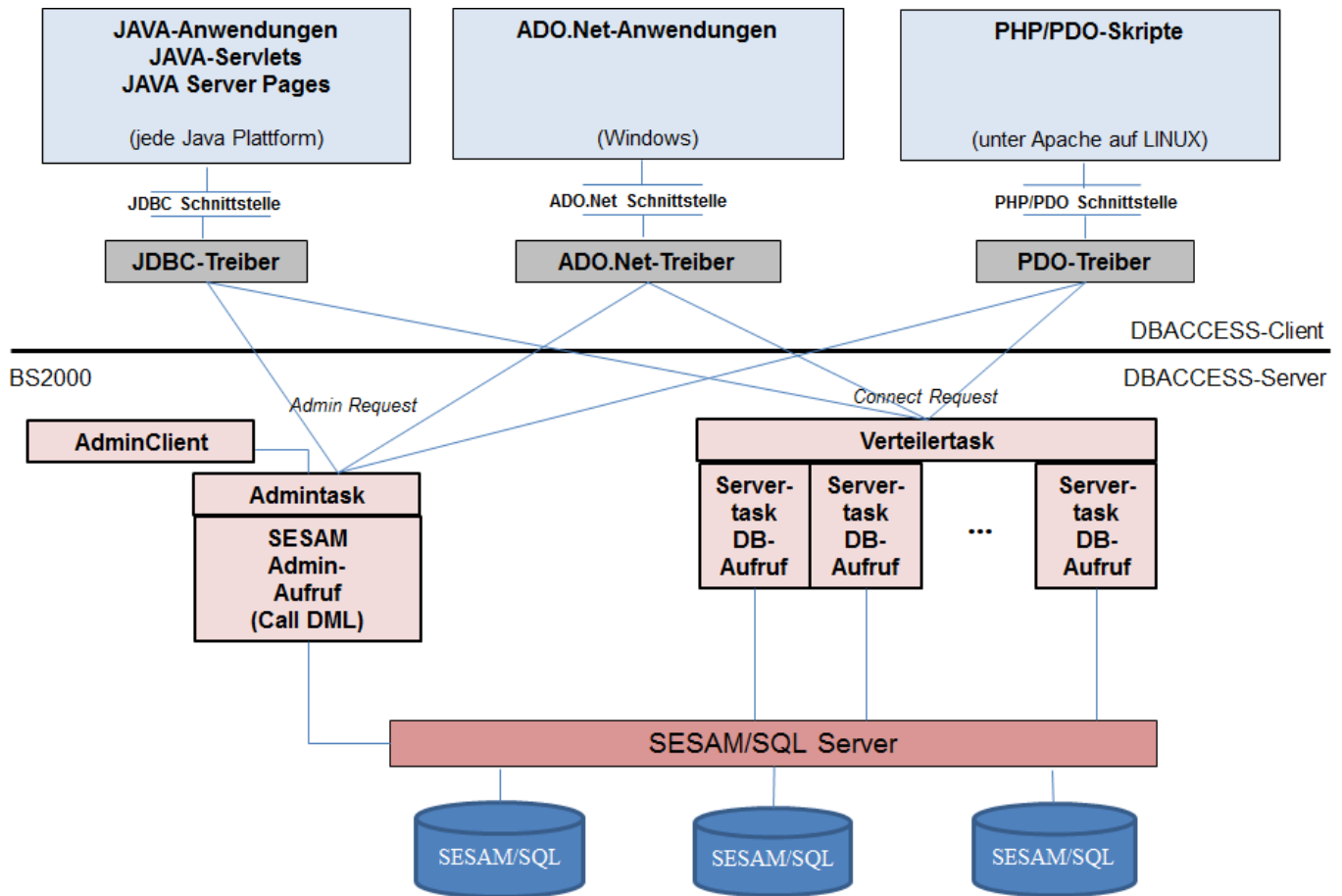
Nach dem Start erzeugt die Verteilertask mehrere Servertasks entsprechend der Anzahl der vorzuhaltenden freien Tasks (Einstellung in der Initialisierungsdatei, siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)).

Verbindungsanforderungen der Clients für SESAM/SQL-Sessions werden an die Verteilertask gerichtet. Die Verteilertask leitet eine solche Verbindungsanforderung an eine freie Servertask weiter.

Ist die Gesamtzahl der freien und aktiven Tasks noch kleiner als die Anzahl der maximalen Tasks (Einstellung in der Initialisierungsdatei, siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)), so wird anschließend eine neue Servertask gestartet.

Servertask

Eine Servertask übernimmt die Kommunikation mit dem Auftraggeber (Client). SESAM/SQL-Aufträge werden innerhalb dieser Task an das Datenbanksystem weitergeleitet und bearbeitet.



Architektur von SESAM-DBAccess

3 Benutzerverwaltung

- Benutzerprüfung
- Die Erfassungsprozedur SYSPRC.SESDBA.091.USER

3.1 Benutzerprüfung

Der DBAccess-Server kann wahlweise mit oder ohne Benutzerprüfung eingesetzt werden. Dies wird in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) eingestellt.

Arbeiten ohne Benutzerprüfung

Falls ohne Benutzerprüfung gearbeitet wird, ist keine Erfassung von DBAccess-Benutzern notwendig. Die Clients können jeden Benutzernamen verwenden, der in SESAM/SQL existiert und mit einem Systemzugang für den DBAccess-Server verknüpft ist (siehe "[Voraussetzungen zum Starten](#)"). In diesem Fall findet keine Kennwortprüfung statt.

Arbeiten mit Benutzerprüfung

Wenn mit Benutzerprüfung gearbeitet wird, dann muss vor dem Starten des DBAccessServers eine Benutzerdatei erstellt werden.

In der Benutzerdatei müssen alle Benutzer, die über Clients einen Zugriff auf SESAM/SQL erhalten sollen, mit einem Kennwort erfasst werden. Beim Verbindungsaufbau eines Clients wird der übergebene Benutzername und das übergebene Kennwort gegen diese Daten geprüft. So wird entschieden, ob der Client mit dem DBAccess-Server arbeiten darf.

Die Benutzernamen müssen auch in SESAM/SQL existieren und mit einem Systemzugang für den DBAccess-Server verknüpft werden (siehe "[Voraussetzungen zum Starten](#)").

Die Benutzer werden menügesteuert mit der Prozedur `SYSPRC.SESDBA.091.USER` erfasst.

Gestartet wird die Erstellung der Benutzerdatei mit dem Kommando:

```
/CALL-PROCEDURE FROM-FILE=SYSPRC.SESDBA.091.USER, -  
/  
          PROC-PARAMETERS=(USERFILE=SESDBA.USERFILE)
```

Der Parameter `USERFILE` bezeichnet den Namen der zu bearbeitenden Benutzerdatei für die Benutzer. In die Prozedur ist bereits der Standardname `SESDBA.USERFILE` eingetragen. Es darf für die DBAccess-Komponente **nicht** die gleiche Benutzerdatei verwendet werden wie für die anderen Komponenten von DBAccess.

Weitere Prozedurparameter (`PROCLIB`, `INIFILE`, `CRTELIB`) enthalten die Standardnamen für die DBAccess-Prozedurbibliothek, die Initialisierungsdatei und die Bibliothek des CRTE-Laufzeitsystems (siehe [Abschnitt „Dateien, Prozeduren, Bibliotheken, Programme“](#)).

i Änderungen in der Benutzerdatei während des Betriebs des DBAccess-Servers sind erst beim nächsten Verbindungsaufbau gültig.

3.2 Die Erfassungsprozedur SYSPRC.SESDBA.091.USER

Nach dem Start der Prozedur SYSPRC.SESDBA.091.USER erscheint ein Bildschirm zum Erfassen der Benutzer.

```
DBS200 Definition of the service users
-----
+-X(Change)/D(Delete)
!
v
>.< 00001 >.....< >.....<
>.< 00002 >.....< >.....<
>.< 00003 >.....< >.....<
>.< 00004 >.....< >.....<
>.< 00005 >.....< >.....<
>.< 00006 >.....< >.....<
>.< 00007 >.....< >.....<
>.< 00008 >.....< >.....<
>.< 00009 >.....< >.....<
>.< 00010 >.....< >.....<
>.< 00011 >.....< >.....<
>.< 00012 >.....< >.....<
>.< 00013 >.....< >.....<
>.< 00014 >.....< >.....<
>.< 00015 >.....< >.....<
-----
(.....) (position to an entry)
(..) +/+/++/--/RE/EN (scroll/quit data entry/conclude data entry)
-----
LTG TAST
```

In diesem Bildschirm werden die Benutzernamen und die zugehörigen Kennwörter erfasst. Sie werden für die spätere Überprüfung beim Verbindungsaufbau eines Clients hinterlegt. Der Benutzername entspricht dem SESAM/SQL Authorization Identifier.

Benutzer (*User*)

Der Namen des Benutzers kann in Groß-/Kleinschreibung eingegeben werden, sie ist jedoch für die Anmeldung nicht relevant. Leerzeichen im Benutzernamen sind nicht erlaubt.

Kennwort (*Password*)

Das Kennwort des Benutzers kann ebenfalls in Groß-/Kleinschreibung eingegeben werden, sie ist jedoch für die Anmeldung nicht relevant. Leerzeichen sind im Kennwort erlaubt. Beim Erfassen eines neuen Benutzers ist die Angabe eines Kennworts zwingend erforderlich. Nach Taste DUE werden alle erfassten Kennwörter unsichtbar. Wird der Benutzername verändert, so muss das Kennwort ebenfalls neu eingegeben werden.

Markierspalte (*X (Change)* / *D (Delete)*)

Neue Benutzereinträge werden in Leerzeilen erfasst, d.h. es ist kein Eintrag in der Markierspalte notwendig.

Nach Taste DUE werden alle erfassten Kennwörter unsichtbar und alle Einträge werden auf nicht überschreibbar gesetzt.

Folgende Angaben können in der Markierspalte eingetragen werden:

X	Ein existierender Benutzereintrag wird überschreibbar und kann geändert werden. Das Kennwort bleibt unsichtbar. Wird der Benutzername verändert, muss das Kennwort neu eingegeben werden.
D	Ein Benutzereintrag wird gelöscht.

Positionieren (*position to an entry*)

Auf den in diesem Feld eingetragenen Benutzer wird innerhalb der Tabelle positioniert. Wird kein vollständiger Benutzername eingegeben, so wird auf den ersten Benutzereintrag positioniert, der mit dieser Zeichenfolge beginnt.

Blättern/Erfassung abbrechen/Erfassung beenden (*scroll/quit data entry/conclude data entry*)

Folgende Angaben können gemacht werden:

+	Um eine Seite weiter positionieren
-	Um eine Seite zurück positionieren
++	An das Ende der Tabelle positionieren
-	An den Anfang der Tabelle positionieren
RE	Die Erfassung beenden, ohne die gemachten Angaben in die Benutzerdatei zu übernehmen
EN	Die Erfassung beenden, die gemachten Angaben werden sortiert in die Benutzerdatei übernommen

i Die Kennwörter werden verschlüsselt gespeichert und auch verschlüsselt vom Client übertragen. Dadurch wirkt sich die ASCII-EBCDIC-Konvertierung auf die Kennwörter aus. Das bedeutet, wenn Kennwörter Sonderzeichen enthalten und die Konvertierungstabelle geändert wird, müssen diese Kennwörter neu eingegeben und verschlüsselt werden (siehe auch [Abschnitt „EBCDIC-ASCII-Konvertierung“](#)).

4 Starten und Beenden

- Voraussetzungen zum Starten
- Starten des DBAccess-Servers
- Beenden des DBAccess-Servers
- Jobvariablen des DBAccess-Servers

4.1 Voraussetzungen zum Starten

DBH

Vor dem Start des DBAccess-Servers muss der DBH von SESAM/SQL-Server hochgefahren sein.

SESAM/SQL-Konfigurationsdatei

Zum Betrieb des DBAccess-Servers wird eine SESAM-Konfigurationsdatei benötigt. Die Datei muss vom Administrator erstellt werden. Der Standardname ist `SESDBA.SESCONF`. Bei Abweichung muss der Name in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) eingetragen werden. Die SESAM/SQL-Konfigurationsdatei muss die folgenden Angaben enthalten:

```
CNF=<c>
NAM=<n>
NOTYPE
PREFETCH-BUFFER=<pb>
CCSN=<ccs_name>
```

Mit `CNF` und `NAM` wird der SESAM/SQL-DBH adressiert. Mit `NOTYPE` wird die Ausgabe auf Console unterdrückt. Mit `PREFETCH-BUFFER` wird ein Buffer in 4KB Blöcken festgelegt, der bei nicht änderbaren Ergebnissen zwischen DBH und Server verwendet wird.

Der Operand `CCSN` legt fest, mit welchem codierten Zeichensatz die Werte für Spalten, die in der Datenbank vom Typ `NATIONAL CHARACTER` bzw. `NATIONAL CHARACTER VARYING` vereinbart sind, interpretiert werden sollen. Wenn die angesprochene Datenbank einen signifikanten CCSN (d.h. ungleich `*NONE`) hat, dann muss hier der CCSN der Datenbank angegeben werden. Andernfalls kann mit der Datenbank nicht kommuniziert werden.

Der Inhalt der SESAM-Konfigurationsdatei kann während des Betriebes geändert werden. Änderungen in der SESAM-Konfigurationsdatei gelten für neu gestartete Tasks. Sie gelten nicht für bereits gestartete Tasks.

SESAM/SQL-Datenbankzugriffsrechte für Clients

Benutzer, die über Clients zugreifen dürfen, müssen vom SESAM/SQL-Administrator eingerichtet (Anweisung `CREATE USER`) werden, und sie müssen die entsprechenden Rechte erhalten (Anweisung `GRANT`).

Für die Benutzer muss ein Systemzugang definiert werden:

```
CREATE SYSTEM_USER ({rechnername | *}, [ *], {bs2000_benutzerkennung | *}) FOR
benutzer AT CATALOG catalog
```

wobei `rechnername` und `bs2000_benutzerkennung` den Namen des Rechners und die Kennung bezeichnen, wo der DBAccess-Server gestartet wird.

Falls Clients nicht den Berechtigungsschlüssel des *Universal User* verwenden, muss den Tabellen `SYS_UNIQUE_CONSTRAINTS`, `SYS_REFERENTIAL_CONSTRAINTS` und `SYS_INDEXES` im Schema `SYS_INFO_SCHEMA` für DBAccess-Benutzer ein Leserecht erteilt werden, z.B. mit den folgenden Anweisungen:

<code>GRANT SELECT ON TABLE SYS_UNIQUE_CONSTRAINTS</code>	<code>TO PUBLIC</code>
<code>GRANT SELECT ON TABLE SYS_REFERENTIAL_CONSTRAINTS</code>	<code>TO PUBLIC</code>
<code>GRANT SELECT ON TABLE SYS_INDEXES</code>	<code>TO PUBLIC</code>

Cancel-Berechtigung für Clients

Clients haben die Möglichkeit, eigene SESAM/SQL-Aufträge abubrechen. Hierzu benötigt der DBAccess-Server die Berechtigung, SESAM-Administrationsanweisungen über die SESAM-CALL-DML Schnittstelle einzugeben. Diese Berechtigung muss für den Rechner und die BS2000-Kennung, unter der der DBAccess-Server läuft, eingetragen werden.

Die Berechtigung erfolgt mit der SESAM-DBH-Option

```
ADMINISTRATOR = *ANY(...)
```

bzw.

```
ADMINISTRATOR = *TIME-SHARING-USER(...)
```

Das dabei vergebene Administrationskennwort muss in der SESAM-DBAccess INI-Datei, Abschnitt [Connect], Schlüssel „SESAMAdmPass“, eingetragen werden.

BS2000-Systembetreuung

Für die BS2000-Benutzerkennung und die Jobklasse, unter der der DBAccess-Server läuft, sollten TP-Prozesse zugelassen werden. Außerdem sollte die Ablaufpriorität (RUN-PRIO) zulässig sein, die in der SESAM-DBAccess INI-Datei, Abschnitt [MainTask] und Schlüssel „RunPrio“, für mit fork() erzeugte Batch-Jobs eingetragen wird.

4.2 Starten des DBAccess-Servers

Shared Code für den DBAccess-Server

Teile des DBAccess-Servers enthalten gemeinsam benutzte (shared) Module. Diese Module sind im Subsystem SESDBA zusammengefasst und können vor dem Start des DBAccess-Servers mit dem Kommando /START-SUBSYSTEM in den Klasse-4-Speicher geladen werden.

Da der DBAccess-Server mit mehreren Servertasks gestartet wird, verringert dies die Belastung des Hauptspeichers und verkürzt die Ladezeit der Servertasks.

Beispiel

```
/START-SUBSYSTEM SUBSYSTEM-NAME=SESDBA,VERSION=9.1A00,  
                VERSION-PARALLELISM=*COEXISTENCE_MODE
```

Der Wert *COEXISTENCE_MODE wird für den DBAccess-Server empfohlen.

Starten des DBAccess-Servers

Der DBAccess-Server wird im Allgemeinen als Batch-Job gestartet mit dem Kommando

```
/START-SESDBA-SERVER
```

Der Server verwendet eine Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)), deren Name über SYSDTA eingelesen wird. Standardmäßig heißt diese Datei SYSDAT.SESDBA.091.INIT. Falls der Standardname verwendet werden soll, genügt die Eingabe von DUE bzw. einer Leerzeile in Prozeduren.

Beispiel für die Kommandofolge in einer Prozedur

```
(IN)  /ASSIGN-SYSDTA TO-FILE=*SYSCMD  
(IN)  /START-SESDBA-SERVER  
(OUT) Enter Name of INI-file (default is SYSDAT.SESDBA.091.INIT):  
(IN)  <INI-Dateiname> oder Leerzeichen
```

4.3 Beenden des DBAccess-Servers

Zum Beenden des DBAccess-Servers wird das Administrationsprogramm gestartet mit dem Kommando

```
/START-SESDBA-ADMINISTRATION
```

Wie der Server verwendet das Administrationsprogramm eine Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)), deren Name über SYSDTA eingelesen wird. Standardmäßig heißt sie SYSDAT.SESDBA.091.INIT. Falls der Standardname verwendet werden soll, genügt die Eingabe von DUE bzw. einer Leerzeile in Prozeduren..

Anschließend fordert das Programm über SYSDTA ein Kommando an. Alle Kommandos und Parameter können auch in Kleinbuchstaben eingegeben werden. Zum Beenden gibt es folgende Kommandos:

CLOSE	Der DBAccess-Server (Verteilertask und freie Servertasks) wird beendet. Aktive Servertasks laufen weiter. Aktive Clients behalten ihre Verbindung. Es können sich jedoch keine weiteren Clients mehr anmelden. Die Administration ist weiterhin möglich. Achtung: Bevor der Server neu gestartet werden kann, muss auch die Administration mit CLOSE ALL beendet werden.
CLOSE ALL	Der DBAccess-Server wird komplett beendet (einschließlich Administration). Auch aktive Servertasks werden beendet, d.h. offene Transaktionen werden zurückgesetzt. Das Administrationsprogramm beendet sich anschließend.
CLOSE_ALL	Dieses Kommando wird aus Kompatibilitätsgründen weiterhin unterstützt. Es kann nicht abgekürzt werden. Es hat dieselbe Wirkung wie CLOSE ALL.
END	Das Administrationsprogramm wird beendet.

Das Administrationsprogramm kann im Dialog, in einer Prozedur oder als Batch-Job gestartet werden. In Prozeduren und Batch-Jobs kann das auszuführende Kommando in der Prozedur angegeben werden.

i Das Beenden des Servers kann bei eingeschalteter Option SessionLog=True einige Minuten dauern, da in diesem Fall die vorhandenen Servertasks (serialisiert) eine Ende-Meldung in das SessionLog schreiben. Ein nachfolgender Neustart des Servers sollte erst dann erfolgen, wenn alle Servertasks des vorangehenden Laufs beendet sind.

Beispiel für die Kommandofolge in einer Prozedur

```
(IN) /ASSIGN-SYSDTA TO-FILE=*SYSCMD
(IN) /START-SESDBA-ADMINISTRATION
(OUT) Enter Name of INI-file (default is SYSDAT.SESDBA.091.INIT):
(IN) <INI-Dateiname> oder Leerzeichen
(OUT) Enter administration command:
(IN) CLOSE ALL
```

4.4 Jobvariablen des DBAccess-Servers

Der DBAccess-Server zeigt seine Betriebsbereitschaft durch die Jobvariable:

SESAM.DBA-SERVER.<admin-port>

<admin-port> ist die Portnummer für die Administration des DBAccess-Servers.

Der Inhalt der Jobvariable ist:

Date	<i>jjj.mm.tt</i>
Time	<i>hh.mm.ss</i>
<text>	

<text> enthält:

DBA-SERVER_STARTED	nach erfolgreichem Start des DBAccess-Servers
DBA-SERVER_NORMAL_END	nach erfolgreicher Beendigung der Server-Session
DBA-SERVER_ABNORMAL_END	nach Abbruch der Server-Session

5 Betrieb des DBAccess-Servers

- Initialisierungsdatei (Startparameter)
- EBCDIC-ASCII-Konvertierung
- Informationsfunktionen
 - Protokollierung in der Session Log Datei
 - Auskunft über aktuelle Verbindungen
 - Auskunft über aktuelle Parameter
- Sichere Verbindungen über SSL

5.1 Initialisierungsdatei (Startparameter)

Der Betrieb des DBAccess-Servers wird durch Startparameter gesteuert. Diese Startparameter sind in einer Initialisierungsdatei `SYSDAT.SESDBA.091.INIT` abgelegt. Diese Initialisierungsdatei hat die folgenden Einträge:

```
[DBMS]
Product=SESAM/SQL
Version=9.1A

[Connect]
SESCONF=<ses_conf>
SESAMAdmPass=<SESAM_administration-password>
DefaultCatalog=<default-catalog>
DefaultSchema=<default-schema>
DefaultUser=<default-user>
UserIdentification=<True/False>

[SSLConnect]
SSLUsage=NO
SSLCertificate=<Certificate-file>
SSLKey=<Key-file>

[MainTask]
ServerPort=2112
AdminPort=2113
UserFile=<user-filename>
ConvTable=<EBCDIC-ASCII-conversion-filename>
MsgFile=<message-filename>
SessionLogFile=<True/False>
RunPrio=210
TasksPrestart=3
TasksMaximum=100
ConnectQueue=5
PackageLen=8192
OptLogMsgLen=8192
MaxPreparableStatements=64

[ServerTask]
SESAM_Prefetch=20

[Trace]
AdminTask=0
MainTask=0
ServerTask=0

[AdminClient]
ServerHost=127.0.0.1
```

Zur Steuerung des DBAccess-Servers können bzw. müssen diese Einträge vom Administrator angepasst werden. Dies gilt insbesondere, wenn neben der JDBC-Schnittstelle parallel auch die ADO.NET-Schnittstelle bedient wird.

Erläuterung der Einträge

Die Einträge in der Initialisierungsdatei haben die folgende Bedeutung:

Abschnitt [DBMS]

Schlüssel	Standard	Erläuterung	A
Product	SESAM/SQL	DBMS-Produktname	N
Version	9.1A	DBMS-Version	N

Abschnitt [Connect]

Schlüssel	Standard	Erläuterung	A
SESCONF	SESDBA. SESCONF	SESAM-Konfigurationsdatei	Y
SESAMAdmPass	(leer)	Kennwort für die SESAM/SQL-Administration	Y
DefaultCatalog	D0CATALOG	Default-Katalogname, wird als Default für die Client-Operation CONNECT verwendet ¹	Y
DefaultSchema	D0SCHEMA	Default-Schemaname, wird als Default für die Client-Operation CONNECT verwendet ¹	Y
DefaultUser	D0USER	Default-Benutzername, wird als Default für die Client-Operation CONNECT verwendet. Groß-/Kleinschreibung ist nicht relevant.	Y
UserIdentification	True	Schalter bestimmt, ob der Server mit (True) oder ohne (False) Benutzeridentifikation arbeitet.	Y

¹Reguläre Namen müssen in Großbuchstaben angegeben werden. Nicht reguläre Namen dürfen nicht in Anführungszeichen eingeschlossen werden. D.h, die Namen müssen so angegeben werden, wie sie im INFORMATION_SCHEMA (z.B. Tabelle TABLES) angezeigt werden.

Abschnitt [SSLConnect]

Schlüssel	Standard	Erläuterung	A
SSLUsage	NO	NO: d.h. eine unverschlüsselte Kommunikation(Defaultverbindungstyp). PREFER: d.h. der Server akzeptiert sowohl Connects <u>mit</u> als auch <u>ohne</u> Verschlüsselung. REQUIRE: d.h. der Server akzeptiert nur noch Connects mit Verschlüsselung. Connect-Versuche ohne Verschlüsselung werden abgewiesen.	N
SSLCertificate	(leer)	Zertifikatsangabe (Datei mit Zertifikat) für SSL Verbindungen	N

SSLKey	(leer)	Datei mit dem privaten Schlüssel für die Nachrichtenverschlüsselung für SSL Verbindungen	N
--------	--------	--	---

Abschnitt [MainTask]

Schlüssel	Standard	Erläuterung	A
ServerPort	2112	Portnummer der Verteilertask, diese Portnummer muss auch am Clientrechner administriert werden.	N
AdminPort	2113	Portnummer der Admintask.	N
UserFile	SESDBA. USERFILE	Name der Benutzerdatei des Servers.	N
ConvTable	SYSDAT. SESDBA. 091. EBCASC	Dateiname der Konvertierungstabelle des Servers. *SESAM: mit dieser Angabe wird über XHCS der CCSN zur Konvertierung benutzt, der als CCSN in der SESAM-Konfigurationsdatei eingetragen ist.	N
MsgFile	SYSMES. SESDBA. 091	Name der SESAM-DBAccess Meldungsdatei.	N
SessionLogFile	False	True: Informationen über Verbindungen werden in eine Session Log Datei geschrieben.	D
RunPrio	210	Ablaufpriorität der Tasks (BS2000-Parameter RUN-PRIO, 128 - 255).	P
TasksPrestart	3	Anzahl Tasks, die der Server vorweg startet, um einen schnellen Verbindungsaufbau zu ermöglichen ¹ (1 - 500).	P
TasksMaximum	100	Maximale Anzahl Servertasks, die der Server parallel starten darf ¹ (1 - 500).	P
ConnectQueue	5	Anzahl von Verbindungsanforderungen, die noch in einer Warteschlange gehalten werden, wenn der Server die maximale Anzahl Servertasks erreicht hat (0 - 100).	P
PackageLen	8192	Größe für Nachrichtenpakete (1024 - 32768).	P
OptLogMsgLen	8192	Optimale Größe für logische Nachrichten (PackageLen - 65536).	P
MaxPreparable Statements	64	max. Anzahl vorbereiteter Anweisungen (64 - 1000).	P

¹Die Anzahl Tasks, die der Server starten kann, wird begrenzt durch obige Parameter. Wenn die angegebenen Werte für TasksPrestart bzw. TasksMaximum nicht erreicht werden, sollten Sie die Werte MAXUP bzw. NPROC der Posix-Installation prüfen, die die Anzahl Prozesse pro User bzw. im System beschränken.

Abschnitt [ServerTask]

Schlüssel	Standard	Erläuterung	A
SESAM_Prefetch	20	Anzahl von Sätzen, die im Prefetch Modus von SESAM angefordert werden.	P

Abschnitt [Trace]

Schlüssel	Standard	Erläuterung	A
AdminTask	0	Trace Level der Admintask	D
MainTask	0	Trace Level der Verteilertask	D
ServerTask	0	Trace Level der Servertask	D

Abschnitt [AdminClient]

Schlüssel	Standard	Erläuterung	A
ServerHost	127.0.0.1	IP-Adresse des Servers (127.0.0.1 bezeichnet den lokalen Rechner)	N

Die Spalte **A** bezeichnet die Notwendigkeit für den Administrator, die Standardwerte zu ändern:

N	Eine Änderung ist im allgemeinen nicht notwendig.
Y	Anpassung an die SESAM-Installation und Datenbank ist notwendig.
P	Anpassungen zur Verbesserung der Performance können erfolgen, wenn das Anwendungsprofil bekannt ist.
D	Änderungen sind nur zur Erstellung von Diagnoseunterlagen notwendig.

Zeitpunkt der Auswertung

Alle Schlüssel werden beim Start des Servers gelesen. Sie werden für alle Tasks in einem Memory-Pool hinterlegt. Änderungen während des laufenden Betriebes sind zunächst wirkungslos. Sie gelten erst nach einem neuerlichen Start der Servers.

Geänderte Einträge im Abschnitt [Trace] werden für alle Tasks durch das Administrationskommando `REUSE-INIT-FILE` neu übernommen.

5.2 EBCDIC-ASCII-Konvertierung

Bei der Übertragung der Daten vom SESAM-DBAccess-Server an den Client-Rechner konvertiert DBAccess diese Daten von EBCDIC.DF4.04 nach ISO8859-1 (8 Bit ASCII) und umgekehrt.

Sollte jedoch aufgrund nationaler Zeichensätze oder sonstiger Eigenheiten der Daten eine andere Konvertierung benötigt werden, so bietet DBAccess die Möglichkeit, in der Konvertierungstabelle `SYSDAT.SESDBA.091.EBCASC` für jedes EBCDIC-Zeichen die gewünschte ASCII-Konvertierung individuell festzulegen. Geändert werden darf die Notation jedoch nur für Sonderzeichen.

Standardmäßig hat diese Datei den folgenden Inhalt:

```
# Converting EBCDIC.DF.04 to ISO8859-1 (ASCII 8 bit)
# This conversion table differs from standard conversion in following items:
# - EBCDIC 0x15 Newline -> ASCII 0x0a Newline instead of 0x1a
# - EBCDIC 0x25          -> ASCII 0x85          instead of 0x0a
# - The conversion is extended to a bijective conversion
#
EBCDIC->ASCII
#   -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -a -b -c -d -e -f
0-   00 01 02 03 9c 09 86 7f 97 8d 8e 0b 0c 0d 0e 0f    0
01-  10 11 12 13 9d 0a 08 87 18 19 92 8f 1c 1d 1e 1f    1
12-  80 81 82 83 84 85 17 1b 88 89 8a 8b 8c 05 06 07    2
23-  90 91 16 93 94 95 96 04 98 99 9a 9b 14 15 9e 1a    3
34-  20 a0 e2 e4 e0 e1 e3 e5 e7 f1 60 2e 3c 28 2b 7c    4
45-  26 e9 ea eb e8 ed ee ef ec df 21 24 2a 29 3b 9f    5
56-  2d 2f c2 c4 c0 c1 c3 c5 c7 d1 5e 2c 25 5f 3e 3f    6
67-  f8 c9 ca cb c8 cd ce cf cc a8 3a 23 40 27 3d 22    7
78-  d8 61 62 63 64 65 66 67 68 69 ab bb f0 fd de b1    8
89-  b0 6a 6b 6c 6d 6e 6f 70 71 72 aa ba e6 b8 c6 a4    9
9a-  b5 af 73 74 75 76 77 78 79 7a a1 bf d0 dd fe ae    a
ab-  a2 a3 a5 b7 a9 a7 b6 bc bd be ac 5b 5c 5d b4 d7    b
bc-  f9 41 42 43 44 45 46 47 48 49 ad f4 f6 f2 f3 f5    c
cd-  a6 4a 4b 4c 4d 4e 4f 50 51 52 b9 fb fc db fa ff    d
de-  d9 f7 53 54 55 56 57 58 59 5a b2 d4 d6 d2 d3 d5    e
ef-  30 31 32 33 34 35 36 37 38 39 b3 7b dc 7d da 7e    f#
#   -0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -a -b -c -d -e -f
#
```

Zu jedem EBCDIC-Zeichen, das durch die Randbeschriftung gekennzeichnet wird, ist im Tabellenkörper die entsprechende ASCII-Darstellung angegeben. Die Konvertierung in der umgekehrten Richtung (z.B. SQL Anweisungen und Eingabeparameter) erfolgt von DBAccess automatisch durch Invertierung dieser Tabelle.

Hinweise

- Kommentarzeilen sind in der Datei mit # an der 1. Stelle gekennzeichnet.
- Die Konvertierungstabelle EBCDIC-ASCII besagt, dass z.B. EBCDIC 0xF0 nach ASCII 0x30 konvertiert wird.
- Die Abbildung ist bijektiv, d.h. jedem Hexadezimalwert im EBCDIC wird eindeutig ein Hexadezimalwert im ASCII zugeordnet und umgekehrt.
- Bei einer Änderung muss darauf geachtet werden, dass eine bijektive Abbildung entsteht und keine Zeichen, die in SQL-Schlüsselwörtern vorkommen, geändert werden.

-
- Die Kennwörter werden verschlüsselt gespeichert und auch verschlüsselt vom Client übertragen. Dadurch wirkt sich die ASCII-EBCDIC-Konvertierung auf die Kennwörter aus. Wenn Kennwörter Sonderzeichen enthalten und die Konvertierungstabelle geändert wird, müssen dementsprechend diese Kennwörter neu eingegeben und verschlüsselt werden.

Erzeugen einer neuen Konvertierungstabelle

Wenn in der Initialisierungsdatei für die Konvertierungstabelle des Servers der StandardDateiname durch *SESAM ersetzt wird, dann wird über XHCS der CCSN zur Konvertierung benutzt, der als CCSN in der SESAM-Konfigurationsdatei eingetragen ist.

Das Administrationskommando WRITE_NEW_CONVTABLE <dateiname> ermöglicht das Speichern dieser neuen Konvertierungstabelle, die dann bei Bedarf als neue Konvertierungstabelle des Servers benutzt werden kann.

Eine Kennwortdatei ist nur mit der Konvertierungstabelle nutzbar, mit der sie erzeugt wurde, siehe "Die Erfassungsprozedur SYSPRC.SESDBA.090.USER". Daher sollte die neue Konvertierungstabelle zum Erzeugen einer neuen Kennwortdatei benutzt werden.

5.3 Informationsfunktionen

Der DBAccess-Server kann Daten historischer Client-Verbindungen in einer Session Log Datei protokollieren und Auskunft über aktuelle Verbindungen geben.

5.3.1 Protokollierung in der Session Log Datei

Der DBAccess-Server kann Daten historischer Client-Verbindungen in einer Session Log Datei protokollieren. Hierzu muss in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) im Abschnitt [MainTask] der Schlüssel SessionLogFile=True gesetzt werden.

Der Name der Datei ist SESDBA.SESSIONLOG.<ServerPort>, wobei ServerPort die Portnummer der Verteilertask ist.

Die Datei wird beim Start des Servers nicht gelöscht, falls sie schon existiert. Sie wird weitergeschrieben.

i Die Session Log Datei sollte während des Serverlaufes nicht mit /SHOW-FILE angeschaut werden, da dadurch die Datei gesperrt wird. Die Datei kann jedoch mit EDT angeschaut werden, ohne die Ausgaben des Servers zu stören.

Folgende Informationen werden in die Datei geschrieben:

- Beim Start des Servers werden die Meldungen

```
Start of SESAM-DBAccess-Server (<ServerPort>) (Version <Version>)  
(TSN = <Tsn>, PID = <Pid>) Administration Task at YYYY-MM-DD hh:mm:ss  
Start of SESAM-DBAccess-Server (<ServerPort>) (Version <Version>)  
(TSN = <Tsn>, PID = <Pid>) Distributor Task at YYYY-MM-DD hh:mm:ss
```

geschrieben, wobei ServerPort die Portnummer der Verteilertask, Version die Version des DBAccess-Servers, Tsn die Task Sequence Number und Pid die Prozess ID der jeweiligen Task bezeichnet.

- Während der Server läuft, protokollieren die Servertasks die Verbindungen:

Die Ausgaben haben folgendes Format:

Position	Inhalt
1 – 19	Zeitstempel der Initialisierung der Servertask.
21 – 24	TSN (Task sequence number) der Servertask.
26 –30	PID (Prozess ID) der Servertask.
32	Status: Leer = bisher keine Verbindung aufgebaut. F = Verbindung mit Fehler beendet. C = nach erfolgreichem Connect verbunden. S = nach erfolgreichem Connect über SSL verbunden. P = nach erfolgreichem Connect verbunden, aber momentan suspended. Q = noch erfolgreichem Connect über SSL verbunden, aber momentan suspended. D = Verbindung durch Disconnect beendet. A = Verbindung durch Verlust der Netzverbindung oder wegen Administrationskommando CLOSE ALL abgebaut.
34 – 47	Zeitstempel des Verbindungsaufbaus, nur wenn Status = F/D/A.
49 – 66	Benutzername, nur wenn Status = F/D/A.
Ab 68	Rechnername oder IP-Adresse des Client Rechners, nur wenn Status = F/D/A.
Ab 89	Zeitstempel des Verbindungsabbaus, nur wenn Status = D/A.

Die Ausgaben erfolgen, wenn

- ein Verbindungsaufbau, z.B. wegen falscher Benutzeridentifikation, fehlschlägt (Status = F).
- eine Verbindung kontrolliert vom Client beendet wird (Status = D).
- eine Verbindung aus anderen Gründen beendet wird (z.B. Administrationskommando CLOSE ALL, Abbruch der TCP/IP-Verbindung) (Status = A).
- eine initialisierte Task, die noch keine Verbindung hatte, beendet wird (Administrationskommando CLOSE ALL oder CLOSE) (Status = Leer).

-
- Beim Beenden des Servers werden folgende Meldungen geschrieben:

```
Maximum number of simultaneous connections was <n1>  
Maximum number of simultaneous connections allowed by configuration: <n2>  
Shutdown of SESAM-DBAccess-Server (<ServerPort>) Administration Task at YYYY-MM-DD hh:mm:ss  
Shutdown of SESAM-DBAccess-Server (<ServerPort>) Distributor Task at YYYY-MM-DD hh:mm:ss
```

n1 bezeichnet die maximale Anzahl paralleler Client-Verbindungen während des Serverlaufes.

n2 bezeichnet die in der Initialisierungsdatei im Abschnitt [MainTask] mit dem Schlüssel TasksMaximum festgelegte, maximal erlaubte Anzahl paralleler Client-Verbindungen.

ServerPort bezeichnet die Portnummer der Verteilertask.

5.3.2 Auskunft über aktuelle Verbindungen

Der DBAccess-Server liefert Informationen über aktuelle Verbindungen. Hierzu wird das Administrationsprogramm SESDBAADM, wie in [Abschnitt „Beenden des DBAccess-Servers“](#) beschrieben, gestartet. Nun können die folgenden Administrationskommandos eingegeben werden:

GET_INFO	ACTIVE [number]	Auflistung der momentan aktiven Tasks und TCP/IP Verbindungen. Number > 0 gibt die Zahl der auf einmal auszugebenden Sätze an. Stehen noch weitere Sätze zur Verfügung, folgt eine Meldung „Press ENTER to continue“. Number = 0 (Default) bedeutet keine Begrenzung.
GET_INFO	STATISTICS	Ausgabe von Statistik-Informationen.

i Die Kommandos und Parameter können auch in Kleinschreibung eingegeben werden.

Die Ausgaben des Kommandos GET_INFO ACTIVE haben folgendes Format:

Position	Inhalt
1 – 19	Zeitstempel der Initialisierung der Servertask.
21 – 24	TSN (Task sequence number) der Servertask.
26 – 30	PID (Prozess ID) der Servertask.
32	Status: Leer = bisher kein Connect gelaufen. F = Connect mit Fehler beendet. C = es besteht eine aktuelle unverschlüsselte Verbindung. S = es besteht eine aktuelle verschlüsselte Verbindung. P = es besteht eine aktuelle unverschlüsselte Verbindung, aber momentan suspended. Q = es besteht eine aktuelle verschlüsselte Verbindung, aber momentan suspended. D = die Verbindung wurde logisch beendet.
34 – 47	Zeitstempel des Verbindungsaufbaus, nur wenn Status = F/C/S/P/Q/D.
49 – 66	Benutzername, nur wenn Status = F/C/S/P/Q/D.
Ab 68	Rechnername oder IP-Adresse des Client-Rechners, nur wenn Status = F/C/S/P/Q/D.

Das Kommando GET_INFO STATISTICS erzeugt folgende Ausgaben:

```
Number of current connections: <n1>  
Maximum number of simultaneous connections up to now: <n2>  
Maximum number of simultaneous connections allowed by configuration: <n3>
```

n1 bezeichnet die aktuelle Anzahl von Verbindungen.

n2 bezeichnet die max. Anzahl paralleler Client-Verbindungen während des Serverlaufes.

n3 bezeichnet die in der Initialisierungsdatei im Abschnitt [MainTask] mit dem Schlüssel TasksMaximum festgelegte, maximal erlaubte Anzahl paralleler Client-Verbindungen.

5.3.3 Auskunft über aktuelle Parameter

Der DBAccess-Server liefert Informationen über die aktuellen Parameter der laufenden Session. Diese entsprechen normalerweise den Einstellungen in der ini-Datei, können aber auch andere Werte erhalten, z.B. bei Fehlern in der ini-Datei, die nicht zum Abbruch der Initialisierung führen.

Hierzu wird das Administrationsprogramm SESDBAADM, wie in [Abschnitt „Beenden des DBAccess-Servers“](#) beschrieben, gestartet. Nun kann das folgende Administrationskommando eingegeben werden:

```
GET_INFO PARAMETERS
```

Auflistung der momentan gültigen Parameter der Session

i Das Kommando kann auch in Kleinschreibung eingegeben werden.

Beispiel für die Ausgabe

```
IniFileName:          SYSDAT.SESDBA.091.INIT
Product:              SESAM/SQL
Version:              9.1A
SESCONF:              SESDBA.SESCONF
SESAMAdmPass:         ***
DefaultCatalog:      DDCATALOG
DefaultSchema:        DOSHEMA
DefaultUser:          DOUSER
UserIdentification:   True
SSLUsage:             NO
SSLCertificate:
SSLKey:
ServerPort:           2112
AdminPort:            2113
UserFile:              SESDBA.USERFILE
ConvTable:            SYSDAT.SESDBA.091.EBCASC
MsgFile:              SYSMES.SESDBA.091
SessionLogFile:       False
RunPrio:              210
TasksPrestart:        3
TasksMaximum:         100
ConnectQueue:         5
PackageLen:           8192
OptLogMsgLen:         8192
MaxPreparableStatements: 64
SESAM_Prefetch:       20
AdminTask:            0
MainTask:             0
ServerTask:           0
ServerHost:           127.0.0.1
```

5.4 Sichere Verbindungen über SSL

Manche Applikationen erfordern aus Sicherheitsgründen eine höhere Sicherheitsstufe bei der Netzwerk-Kommunikation eines Clients mit dem DBAccess-Server. Daher bietet der DBAccess Server die Möglichkeit, sogenannte "Sichere Verbindungen" durch Nutzung von SSL zu verwenden.

Dabei hat man folgende Konfigurationsmöglichkeiten (s. Initialisierungsdatei):

- Der Server kann wie bisher als Server ohne die Möglichkeit für SSL Verbindungen konfiguriert werden (im weiteren abkürzend als "**NoSSL Server**" bezeichnet).
- Der Server kann so konfiguriert werden, dass er ausschliesslich SSL Verbindungen akzeptiert (im weiteren abkürzend als "**SSL Server**" bezeichnet).
- Der Server kann so konfiguriert werden, dass er beide Verbindungstypen akzeptiert; in diesem Fall wird der Server immer eine SSL Verbindung wählen, wenn ihm vom Client die Wahl des Verbindungstyps überlassen wird (im weiteren abkürzend als "**SSL/NoSSL Server**" bezeichnet).

Der DBACCESS-Server benötigt ein Server-Zertifikat (s. Initialisierungsdatei); über dieses kann bei sicheren Verbindungen der Algorithmus für die Verschlüsselung des Netzverkehrs zwischen Client und Server ausgehandelt werden (schliesslich müssen beide Partner einen gemeinsamen Verschlüsselungsalgorithmus beherrschen).

Zertifikate kann man sich über offizielle Zertifikatsstellen beschaffen (genauer: man schickt ein Zertifikat und ein CSR, d.h. ein Certificate Signing Request an eine derartige Stelle, die das Zertifikat dann offiziell signiert); in diesem Fall werden sie dort hinterlegt und können über eine URL, d.h über einen Webzugriff, verwendet werden oder sie sind als lokale Dateien verfügbar.

Man kann sich aber auch selbst(-signierte) Zertifikate erzeugen; dafür wird im BS2000 die Prozedur make.cert in der Bibliothek \$TSOS.SYSSPR.TCP-IP-AP.0nn ausgeliefert.

Die Verwendung dieser Prozedur ist im Manual "interNet Services User Guide" beschrieben.

Man beachte, dass ein Zertifikat mindestens aus zwei Teilen besteht: Der eigentlichen Zertifikats-Datei und einer Datei mit dem privaten Schlüssel für die Verschlüsselung (das natürlich nicht allgemein zugänglich gemacht werden sollte). Beide Teile werden für die Konfiguration eines SSL bzw. SSL/NoSSL Servers benötigt (s. Initialisierungsdatei).

Ein Client kann bei einer Verbindungsanforderung an den DBACCESS-Server fordern, dass

- eine gewöhnliche, also unsichere Verbindung aufgebaut wird (unsecure)
- eine sichere Verbindung aufgebaut wird (secure)
- eine sichere Verbindung aufgebaut wird, wenn der Server dies unterstützt, oder eine unsichere, wenn der Server keine sicheren Verbindungen unterstützt (prefer secure)

Siehe dazu auch die readme-Datei des jeweiligen Clients (JDBC, ADO.Net, PHP/PDO).

Dann ergibt sich folgende Matrix für die Art der tatsächlich aufgebauten Verbindung zwischen Client und Server:

Servertyp	NoSSL	SSL	SSL/NoSSL
Clientanforderung			
unsecure	unsecure	error	unsecure
secure	error	secure	secure
prefer secure	unsecure	secure	secure

Verschlüsselung und Entschlüsselung kosten CPU sowohl auf Client- als auch auf Server-Seite; insbesondere ist der Aufbau einer sicheren Verbindung wesentlich teurer (bzgl. CPU und v.a. auch bzgl. Netzverkehr) als der Aufbau einer gewöhnlichen, d.h. unsicheren Verbindung.

Aus diesem Grund wird die Verwendung von Connection Pooling für sichere Verbindungen empfohlen, sofern die Client Schnittstelle dies anbietet. Bei den Sesam Clients wird Connection Pooling nur für die JDBC Schnittstelle angeboten (s. readme-Datei des JDBC Clients).

6 Verhalten im Fehlerfall

Fehler und Traceinformationen erscheinen auf den folgenden Medien:

- SYSOUT-Protokoll
- Benutzer-Protokolldatei

6.1 SYSOUT-Protokoll

Schwerwiegende Fehler, die ein Ende des Serverlaufes bedingen, wie z.B. Fehler beim Anmelden an den sockets oder beim Anfordern von Ressourcen, werden auf SYSOUT protokolliert. Zusätzlich werden bei Trace Level 1 diese Meldungen in die entsprechende TraceDatei ausgegeben.

In den folgenden Fällen erfolgt eine Warnung bzw. Information auf SYSOUT:

- *Verteilertask*: Bei Ende wird die maximale Anzahl von parallelen Verbindungen bzw. Tasks protokolliert.
- *Verteilertask*: Falls die maximale Anzahl von parallelen Verbindungen bzw. Tasks erreicht wird, erfolgt eine Warnung.
- *Admintask*: Falls die EBCDIC-ASCII-Konvertierungstabelle nicht bijektiv ist, erfolgt eine Warnung.
- *Admintask*: Falls in der EBCDIC-ASCII-Konvertierungstabelle der IA5 Kern geändert wurde, erfolgt eine Warnung.

i SYSOUT Ausgaben von Sohnprozessen werden nur unter ganz speziellen Bedingungen sichtbar: Die Shellvariable PROGRAM-ENVIRONMENT='SHELL' muss gesetzt sein und das Serverprogramm muss im Dialog - nicht in einer Prozedur - gestartet werden. Daher sind Ausgaben von der Verteilertask und von den Servertasks im allgemeinen nur in der entsprechenden Trace-Datei mit Trace Level 1 zu finden.

6.2 Benutzer-Protokolldatei

In jeder Task können Traceinformationen, abhängig von der eingestellten Protokollierungsstufe, in eine Benutzer-Protokolldatei geschrieben werden.

Trace der Admintask

Durch den Schlüssel AdminTask im Abschnitt [Trace] in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) kann ein Trace für die Admintask eingeschaltet werden.

Für das Trace Level sind die folgenden Werte möglich. Die höheren Levels schließen die kleineren Levels ein:

Wert	Umfang des Trace
0	Trace ausgeschaltet
1	Fehlersituationen
3	Administrationsanforderungen
7	SESAM-Administrationschnittstelle bzw. Administrationsauftrags-Ausführung
15	Socket-Schnittstelle
31	Wichtige Funktionen

Die Trace-Datei der Admintask erhält den Namen TRC.SESDBA.<port>.ADM.<tsn>.<n>, wobei <port> die Portnummer für SESAM/SQL-Anforderungen ist, <tsn> die TSN der Task und <n> gleich 1 bis 5. Die Dateien mit Suffix 1 bis 5 werden im WrapAround-Modus angelegt und überschrieben.

Trace der Verteilertask

Durch den Schlüssel MainTask im Abschnitt [Trace] in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) kann ein Trace für die Verteilertask eingeschaltet werden.

Für das Trace Level sind die folgenden Werte möglich. Die höheren Levels schließen die kleineren Levels ein:

Wert	Umfang des Trace
0	Trace ausgeschaltet
1	Fehlersituationen
3	Verbindungsaufbau und Servertask aktivieren
7	Servertask-Verwaltung
15	Socket-Schnittstelle
31	Wichtige Funktionen

Die Trace-Datei des Verteilers erhält den Namen TRC.SESDBA.<port>.DST.<tsn>.<n>, wobei <port> die Portnummer für SESAM/SQL-Anforderungen ist, <tsn> die TSN der Task und <n> gleich 1 bis 5. Die Dateien mit Suffix 1 bis 5 werden im WrapAround-Modus angelegt.

Trace der Servertasks

Durch den Schlüssel ServerTask im Abschnitt [Trace] in der Initialisierungsdatei (siehe [Abschnitt „Initialisierungsdatei \(Startparameter\)“](#)) kann ein Trace für die Servertasks eingeschaltet werden.

Für das Trace Level sind die folgenden Werte möglich. Die höheren Levels schließen die kleineren Levels ein:

Wert	Umfang des Trace
0	Trace ausgeschaltet
1	Fehlersituationen
3	An SESAM/SQL übergebene SQL-Anweisungen
7	Komplette SESAM/SQL-Schnittstelle
15	Client-Protokollschnittstelle
31	Wichtige Funktionen

Die Trace-Datei des Servers erhält den Namen TRC.SESDBA.<port>.SRV.<tsn>.<n>, wobei <port> die Portnummer für SESAM/SQL-Anforderungen ist, <tsn> die TSN der Task und <n> gleich 1 bis 5. Die Dateien mit Suffix 1 bis 5 werden im WrapAround Modus angelegt.

7 Anhang

Dateien, Prozeduren, Bibliotheken, Programme

Die Namen sind alphabetisch sortiert.

SESAM.DBA-SERVER.<admin-port>	Jobvariable des DBAccess-Servers
SESDBA.SESCONF	SESAM-Konfigurationsdatei
SESDBA.USERFILE	SESAM-DBAccess-Benutzerdatei
SYSDAT.SESDBA.091.EBCASC	Konvertierungstabelle
SYSDAT.SESDBA.091.INIT	Initialisierungsdatei
SYSLIB.SESDBA.091.PROCLIB	SESAM-DBAccess-Bibliothek mit Prozeduren
SYSTEMS.SESDBA.091	SESAM-DBAccess-Meldungsdatei
SYSOML.SESDBA.091	SESAM-DBAccess-Modulbibliothek
SYSPRC.SESDBA.091.USER	Prozedur zum Erstellen der Benutzerdatei
TRC.SESDBA.<port>.ADM.<tsn>.<n>	Trace-Datei für die Admintask
TRC.SESDBA.<port>.SRV.<tsn>.<n>	Trace-Datei für die Servertasks
TRC.SESDBA.<port>.DST.<tsn>.<n>	Trace-Datei für die Verteilertask