

---

# 1 Einleitung

Das Software-Produkt SDF (**S**ystem **D**ialog **F**acility) realisiert die Benutzeroberfläche des BS2000. Es unterstützt die Eingabe von Kommandos und Programmanweisungen im Dialog, aus Prozeduren und im Stapelbetrieb. Zu SDF gehört auch eine eigene Kommandosprache (Kommandos bzw. Anweisungen im SDF-Format). Zur internen Verarbeitung von Eingaben benötigt SDF Syntaxbeschreibungen, die in eigenen Dateien, den Syntaxdateien, abgelegt werden.

SDF V4.5 ist ab BS2000/OSD-BC V3.0 ablauffähig.

Das vorliegende Handbuch „SDF-Verwaltung“ beschreibt, wie die Systembetreuung das Software-Produkt SDF installieren und die Benutzeroberfläche mit Hilfe von SDF-Kommandos und den Dienstprogrammen SDF-I, SDF-U und SDF-PAR verwalten kann. Im Einzelnen beschreibt das Handbuch:

- die von SDF verwendeten Syntaxdateien
- die Kommandos zum Steuern von SDF
- die Installation und Verwaltung von Syntaxdateien
- das Mischen von Syntaxdateien mit Hilfe von SDF-I
- das Modifizieren von Syntaxdateien mit Hilfe von SDF-U
- das Erzeugen und Bearbeiten der SDF-Parameterdatei mit Hilfe von SDF-PAR

## 1.1 Zielsetzung und Zielgruppen des Handbuchs

Das Handbuch richtet sich an die Systembetreuung und an erfahrene BS2000-Benutzer. Kenntnisse des Betriebssystems und der Dialogschnittstelle SDF werden vorausgesetzt.

Als grundlegende Lektüre zur Installation von Softwareprodukten sind die Handbücher „Einführung in die Systembetreuung“ [2], „IMON“ [6] sowie „Verwaltung von Subsystemen“ [16] empfehlenswert.

## 1.2 Konzept des Handbuchs

Das Handbuch zeigt in den einführenden Abschnitten zu SDF, SDF-I und SDF-U, wie SDF installiert, Syntaxdateien gemischt und verändert werden können und bietet darüber hinaus eine Kurzeinführung in die Dialogschnittstelle SDF.

Die Kommandos zum Steuern von SDF und die Anweisungen der Dienstprogramme SDF-I, SDF-U und SDF-PAR sind jeweils in eigenen Kapiteln alphabetisch geordnet beschrieben.

Allgemeine Informationen über die Dialogschnittstelle SDF gibt Ihnen das Handbuch „Einführung in die Dialogschnittstelle SDF“ [1].

Die BS2000-Kommandos einschließlich der privilegierten Kommandos sind in den Handbüchern „Kommandos“ [3], Band 1 bis 5 beschrieben.

Die Maßnahmen zur Systemverwaltung und Systembedienung finden Sie im Handbuch „Einführung in die Systembetreuung“ [2].

### Readme-Datei

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

Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen `SYSRME.produkt.version.sprache`, für SDF V4.5 z.B. unter `SYSRME.SDF.045.D`.

Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Den vollständigen Pfadnamen der deutschen Readme-Datei gibt auch das folgende Kommando aus:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=produkt, LOGICAL-IDENTIFIER=SYSRME.D
```

Die Readme-Datei können Sie mit dem Kommando `SHOW-FILE` oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-DOCUMENT dateiname, LINE-SPACING=*BY-EBCDIC-CONTROL
```

## 1.3 Änderungen gegenüber der vorigen Ausgabe

Das Handbuch zu SDF V4.5A enthält gegenüber dem Vorgängerhandbuch (SDF-Verwaltung für SDF V4.1A, Ausgabe Juni 1996) folgende Änderungen:

- Für die Beendigung einer Task können automatisch zu startende LOGOFF-Prozeduren vereinbart werden. Die Vereinbarung ist sowohl systemweit als auch benutzerspezifisch möglich. Als systemweite LOGOFF-Prozedur kann je eine Call- und eine Include-Prozedur in der SDF-Parameterdatei vereinbart werden. Eine benutzerspezifische Call- und/oder Include-Prozedur wird, wie bei der LOGON-Prozedur, über den Standarddateinamen bereitgestellt.
- Im [Kapitel „Installation von SDF“](#) werden die Installationsdateien für den Einsatz ab BS2000/OSD-BC V3.0 beschrieben.
- Bei folgenden SDF-Kommandos gibt es Änderungen:

Kommando	Änderungen
MODIFY-SDF-PARAMETERS	neuer Operand SYSTEM-LOGOFF-INCL neuer Operand SYSTEM-LOGOFF-PROC
SHOW-SDF-PARAMETERS	neuer Operand SYSTEM-LOGOFF-INCL neuer Operand SYSTEM-LOGOFF-PROC Erweiterung der S-Variablen

- Die Standardanweisung SHOW-STMT zeigt die Syntax einer Anweisung (Funktionsweise analog zum Kommando SHOW-CMD).
- Die Beispiele wurden auf einem System mit BS2000/OSD V5.0 erstellt.

## 1.4 Hinweise zur Benutzeroberfläche

Ausführliche Informationen zu den Möglichkeiten, die SDF bietet, finden Sie im Handbuch „Einführung in die Dialogschnittstelle SDF“ [1].

### Abkürzungsmöglichkeiten von Namen

SDF ermöglicht, Eingaben sowohl im Dialog- und Stapelbetrieb als auch in Prozeduren abzukürzen. Diese Abkürzungen müssen jedoch in der zugehörigen Syntax-Umgebung eindeutig sein. Besonders bei Funktionserweiterungen kann eine heute bestehende Eindeutigkeit wieder aufgehoben werden. Es wird deshalb empfohlen, gerade in Prozeduren keine oder allenfalls die garantierten Abkürzungen zu verwenden, die in den Anweisungsformaten durch Fettdruck hervorgehoben sind.

Kommando- und Anweisungsnamen, Operanden und Schlüsselwortwerte können wie folgt abgekürzt werden:

- von rechts nach links können ganze Namensteile weggelassen werden. Mit einem Namensteil entfällt auch der vorangehende Bindestrich.
- bei jedem Namensteil können von rechts nach links einzelne Zeichen weggelassen werden.
- ein Stern vor einem Schlüsselwortwert als Abkürzung für diesen Operandenwert ist keine zulässige Abkürzung. Ab SDF V4.0A werden Schlüsselwortwerte immer mit führendem Stern dargestellt. Der Stern kann weggelassen werden, wenn alternativ kein variabler Operandenwert möglich ist, dessen Wertebereich den Namen des Schlüsselwortwerts beinhaltet. Diese Abkürzungsmöglichkeit kann durch Erweiterungen in einer Folgeversion eingeschränkt werden. Aus Kompatibilitätsgründen werden Operandenwerte, die bisher ohne führenden Stern dargestellt wurden, auch ohne Stern akzeptiert.

#### *Beispiel für die Eingabe*

Kommando in der Langform:

```
/MODIFY-SDF-OPTIONS SYNTAX-FILE=*NONE ,GUIDANCE=*MINIMUM
```

Kommando in Kurzform:

```
/MOD-SDF-OPT SYN-F=*NO ,GUID=*MIN
```

Die garantierten Abkürzungen sind als Vorschlag für eine verkürzte Eingabe zu verstehen, sie sind aber nicht unbedingt die in Ihrer Syntax-Umgebung kürzest möglichen. Sie sind jedoch aussagefähig und werden langfristig eindeutig gehalten.

Neben dem Kommando- oder Anweisungsnamen kann im Handbuch zusätzlich ein Kurzname dokumentiert sein. Der Kurzname ist als Aliasname des Kommandos oder der Anweisung implementiert und wird langfristig garantiert. Der Kurzname besteht aus maximal 8 Buchstaben (A...Z), die aus dem Kommando- oder Anweisungsnamen abgeleitet sind. Ein Kurzname kann nicht abgekürzt werden.

## Default-Werte

Die Angabe der meisten Operanden ist wahlfrei. Wahlfreie Operanden sind bereits mit einem Operandenwert vorbesetzt, dem so genannten Default-Wert. Die Default-Werte sind in der Syntaxdarstellung durch Unterstreichung gekennzeichnet. Wird der wahlfreie Operand nicht explizit angegeben, so wird zur Ausführung des Kommandos oder der Anweisung für den Operanden der Default-Wert eingesetzt.

## Stellungsoperanden

SDF erlaubt die wahlweise Angabe von Operanden als Schlüsselwort- oder als Stellungsoperanden. Es kann jedoch nicht völlig ausgeschlossen werden, dass sich bei einem Versionswechsel eine Operandenposition ändert. Insbesondere in Prozeduren sollten deshalb Stellungsoperanden vermieden werden.

## XHCS-Unterstützung

Jedes Terminal arbeitet mit einem bestimmten Zeichensatz. Ein codierter Zeichensatz (CCS, Coded Character Set) ist die eindeutige Darstellung der Zeichen eines Zeichensatzes in binärer Form. Bei Einsatz des Softwareproduktes XHCS<sup>1</sup> können erweiterte Zeichensätze verwendet werden. Eingaben werden von SDF gemäß der Standard-Codetabelle **EBCDIC.DF.03** interpretiert (z.B. bei der Umsetzung Groß-/Kleinschreibung). Die Codierung folgender Zeichen darf in einem erweiterten Zeichensatz nicht von der Codierung der Standard-Codetabelle abweichen:

\$, #, @, !, ", ?, ^, =, :, /, \*, -, (, ), [, ], <, >, Komma, Semikolon, Hochkomma

Zusätzliche Zeichen aus einem erweiterten Zeichensatz werden von SDF nur innerhalb der Datentypen <c-string> und <text> ausgewertet, d.h. zur Umsetzung von Groß-/Kleinschreibung wird eine von XHCS für den erweiterten Zeichensatz bereitgestellte Codetabelle verwendet. Zusätzliche Zeichen innerhalb der anderen Datentypen werden mit Syntaxfehler abgewiesen.

Bei der Eingabe von Anweisungen wird das CCS verwendet, das im jeweiligen SDF-Makroaufruf (RDSTMT/CMDRST, TRSTMT/CMDTST oder CORSTMT/CMDCST) angegeben wurde. Wurde kein CCS angegeben, wird das CCS wie bei der Eingabe von Kommandos verwendet.

Weitere Informationen zur XHCS-Unterstützung finden Sie in den Handbüchern „Einführung in die Dialogschnittstelle SDF“ [1] und „XHCS“ [15].

<sup>1</sup> Das Subsystem XHCS (Extended Host Code Support) ermöglicht die 8-Bit-Code-Verarbeitung.

## 1.5 Verwendete Metasprache

### 1.5.1 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:



für Hinweise



für Warnhinweise

Ablaufbeispiele sind mit `dicktengleicher` Schrift dargestellt. Eingaben des Benutzers sind zusätzlich `halfett` hervorgehoben.

Literaturhinweise werden im Text in Kurztiteln und eckigen Klammern [ ] angegeben. Alle folgenden Literaturhinweise beziehen sich auf BS2000/OSD-BC V5.0. Der vollständige Titel jeder Druckschrift, auf die verwiesen wird, ist im Literaturverzeichnis aufgeführt.

### 1.5.2 SDF-Syntaxdarstellung

Diese Syntaxbeschreibung basiert auf der SDF-Version 4.5A. Die Syntax der SDF-Kommando-/Anweisungssprache wird im Folgenden in drei Tabellen erklärt.

*Zu [Tabelle 1](#): Metasyntax*

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet, deren Bedeutung in [Tabelle 1](#) erläutert wird.

*Zu [Tabelle 2](#): Datentypen*

Variable Operandenwerte werden in SDF durch Datentypen dargestellt. Jeder Datentyp repräsentiert einen bestimmten Wertevorrat. Die Anzahl der Datentypen ist beschränkt auf die in [Tabelle 2](#) beschriebenen Datentypen.

Die Beschreibung der Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 2](#) erläutert.

Zu [Tabelle 3](#): Zusätze zu Datentypen

Zusätze zu Datentypen kennzeichnen weitere Eingabevorschriften für Datentypen. Die Zusätze enthalten eine Längen- bzw. Intervallangabe, schränken den Wertevorrat ein (Zusatz beginnt mit *without*), erweitern ihn (Zusatz beginnt mit *with*) oder erklären eine bestimmte Angabe zur Pflichtangabe (Zusatz beginnt mit *mandatory*). Im Handbuch werden folgende Zusätze in gekürzter Form dargestellt:

cat-id	cat
completion	compl
correction-state	corr
generation	gen
lower-case	low
manual-release	man
odd-possible	odd
path-completion	path-compl
separators	sep
temporary-file	temp-file
underscore	under
user-id	user
version	vers
wildcard-constr	wild-constr
wildcards	wild

Für den Datentyp `integer` enthält [Tabelle 3](#) außerdem kursiv gesetzte Einheiten, die nicht Bestandteil der Syntax sind. Sie dienen lediglich als Lesehilfe.

Für Sonderdatentypen, die durch die Implementierung geprüft werden, enthält [Tabelle 3](#) kursiv gesetzte Zusätze (siehe Zusatz *special*), die nicht Bestandteil der Syntax sind.

Die Beschreibung der Zusätze zu den Datentypen gilt für alle Kommandos und Anweisungen. Deshalb werden bei den entsprechenden Operandenbeschreibungen nur noch Abweichungen von [Tabelle 3](#) erläutert.

**Metasyntax**

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter (Kommando-, Anweisungs-, Operandennamen, Schlüsselwortwerte) und konstante Operandenwerte. Schlüsselwortwerte beginnen mit *.	<b>HELP-SDF</b>  <b>SCREEN-STEPS = *NO</b>
<b>GROSSBUCHSTABEN in Halbfett</b>	Großbuchstaben in Halbfett kennzeichnen garantierte bzw. vorgeschlagene Abkürzungen der Schlüsselwörter.	<b>GUIDANCE-MODE = *YES</b>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörigen Operandenwerten.	<b>GUIDANCE-MODE = *NO</b>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird (siehe <a href="#">Tabelle 2</a> und <a href="#">Tabelle 3</a> ).	<b>SYNTAX-FILE = &lt;filename 1..54&gt;</b>
<u>Unterstreichung</u>	Der Unterstrich kennzeichnet den Default-Wert eines Operanden.	<b>GUIDANCE-MODE = *NO</b>
/	Der Schrägstrich trennt alternative Operandenwerte.	<b>NEXT-FIELD = *NO / *YES</b>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	<b>,UNGUIDED-DIALOG = *YES (...)/ *NO</b>
[ ]	Eckige Klammern kennzeichnen struktureinleitende Operandenwerte, deren Angabe optional ist. Die nachfolgende Struktur kann ohne den einleitenden Operandenwert angegeben werden.	<b>SELECT = [*BY-ATTRIBUTES](...)</b>
Einrückung	Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.	<b>,GUIDED-DIALOG = *YES (...)</b> <b>*YES(...)</b> <b>SCREEN-STEPS = *NO / *YES</b>

Tabelle 1: Metasyntax (Teil 1 von 2)



Kennzeichnung	Bedeutung	Beispiele
<p style="text-align: center;"> </p> <p>,</p> <p>list-poss(n):</p>	<p>Der Strich kennzeichnet zusammengehörende Operanden einer Struktur. Sein Verlauf zeigt Anfang und Ende einer Struktur an. Innerhalb einer Struktur können weitere Strukturen auftreten. Die Anzahl senkrechter Striche vor einem Operanden entspricht der Struktur-tiefe.</p> <p>Das Komma steht vor weiteren Operanden der gleichen Struktur-stufe.</p> <p>Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muss sie in runde Klammern eingeschlossen werden.</p>	<p><b>SUPPORT = *TAPE(...)</b></p> <pre> *TAPE(...)          VOLUME = *ANY(...)        *ANY(...)                       ...                     </pre> <p><b>GUIDANCE-MODE = *NO / *YES</b></p> <p><b>,SDF-COMMANDS = *NO / *YES</b></p> <p>list-poss: <b>*SAM / *ISAM</b></p> <p>list-poss(40): &lt;structured-name 1..30&gt;</p> <p>list-poss(256): <b>*OMF / *SYSLST(...)</b> / &lt;filename 1..54&gt;</p>
<p>Kurzname:</p>	<p>Der darauf folgende Name ist ein garantierter Aliasname des Kommando- bzw. Anweisungsnamens.</p>	<p><b>HELP-SDF</b>      Kurzname: <b>HPSDF</b></p>

Tabelle 1: Metasyntax (Teil 2 von 2)

## Datentypen

Datentyp	Zeichenvorrat	Besonderheiten
alphanum-name	A...Z 0...9 \$, #, @	
cat-id	A...Z 0...9	maximal 4 Zeichen; darf nicht mit der Zeichenfolge PUB beginnen
command-rest	beliebig	
composed-name	A...Z 0...9 \$, #, @ Bindestrich Punkt Katalogkennung	alphanumerische Zeichenfolge, die in mehrere durch Punkt oder Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann. Ist auch die Angabe eines Dateinamens möglich, so kann die Zeichenfolge mit einer Katalogkennung im Format :cat: beginnen (siehe Datentyp filename).
c-string	EBCDIC-Zeichen	ist in Hochkommata einzuschließen; der Buchstabe C kann vorangestellt werden; Hochkommata innerhalb des c-string müssen verdoppelt werden
date	0...9 Strukturkennzeichen: Bindestrich	Eingabeformat: jjjj-mm-tt  jjjj: Jahr; wahlweise 2- oder 4-stellig mm: Monat tt: Tag
device	A...Z 0...9 Bindestrich	Zeichenfolge, die maximal 8 Zeichen lang ist und einem im System verfügbaren Gerät entspricht. In der Dialogführung zeigt SDF die zulässigen Operandenwerte an. Hinweise zu möglichen Geräten sind der jeweiligen Operandenbeschreibung zu entnehmen.
fixed	+, - 0...9 Punkt	Eingabeformat: [zeichen][ziffern].[ziffern]  [zeichen]: + oder - [ziffern]: 0...9  muss mindestens eine Ziffer, darf aber außer dem Vorzeichen maximal 10 Zeichen (0...9, Punkt) enthalten

Tabelle 2: Datentypen (Teil 1 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
filename	A...Z 0...9 \$, #, @ Bindestrich Punkt	<p>Eingabeformat:</p> $[:cat:][\$user.] \left\{ \begin{array}{l} \text{datei} \\ \text{datei(nr)} \\ \text{gruppe} \end{array} \right\}$ $\text{gruppe} \left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\}$ <p>:cat:                      wahlfreie Angabe der Katalogkennung;                      Zeichenvorrat auf A...Z und 0...9 eingeschränkt; max. 4 Zeichen; ist in Doppelpunkte einzuschließen;                      voreingestellt ist die Katalogkennung, die der Benutzerkennung laut Eintrag im Benutzerkatalog zugeordnet ist.</p> <p>\$user.                      wahlfreie Angabe der Benutzerkennung;                      Zeichenvorrat ist A...Z, 0...9, \$, #, @;                      max. 8 Zeichen; darf nicht mit einer Ziffer beginnen; \$ und Punkt müssen angegeben werden; voreingestellt ist die eigene Benutzerkennung.</p> <p>\$. (Sonderfall)                      System-Standardkennung</p> <p>datei                      Datei- oder Jobvariablenname;                      kann durch Punkt in mehrere Teilnamen gegliedert sein: name<sub>1</sub>[.name<sub>2</sub>[...]]                      name<sub>i</sub> enthält keinen Punkt und darf nicht mit Bindestrich beginnen oder enden;                      datei ist max. 41 Zeichen lang, darf nicht mit \$ beginnen und muss mindestens ein Zeichen aus A...Z enthalten.</p>

Tabelle 2: Datentypen (Teil 2 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
filename (Forts.)		<p>#datei (Sonderfall) @datei (Sonderfall) # oder @ als erstes Zeichen kennzeichnet je nach Systemparameter temporäre Dateien und Jobvariablen.</p> <p>datei(nr) Banddateiname nr: Versionsnummer; Zeichenvorrat ist A...Z, 0...9, \$, #, @. Klammern müssen angegeben werden.</p> <p>gruppe Name einer Dateigenerationsgruppe (Zeichenvorrat siehe unter „datei“)</p> <p>gruppe <math>\left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\}</math></p> <p>(*abs) absolute Generationsnummer (1..9999); * und Klammern müssen angegeben werden.</p> <p>(+rel) (-rel) relative Generationsnummer (0..99); Vorzeichen und Klammern müssen angegeben werden.</p>
integer	0...9, +, -	+ bzw. - kann nur erstes Zeichen (Vorzeichen) sein.
name	A...Z 0...9 \$, #, @	darf nicht mit einer Ziffer beginnen.

Tabelle 2: Datentypen (Teil 3 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
partial-filename	A...Z 0...9 \$, #, @ Bindestrich Punkt	Eingabeformat: [:cat:][\$user.][teilname.]  :cat: siehe filename \$user. siehe filename  teilname wahlfreie Angabe des gemeinsamen ersten Namensteils von Dateien und Dateigenerationsgruppen in der Form: name <sub>1</sub> . [name <sub>2</sub> . [...]] name <sub>i</sub> ; siehe filename. Das letzte Zeichen von teilname muss ein Punkt sein. Es muss mindestens einer der Teile :cat:, \$user. oder teilname angegeben werden.
posix-filename	A...Z 0...9 Sonderzeichen	Zeichenfolge, die maximal 255 Zeichen lang ist. Besteht entweder aus einem oder zwei Punkten, oder aus alphanumerischen Zeichen und Sonderzeichen; Sonderzeichen sind mit dem Zeichen \ zu entwerten. Nicht erlaubt ist das Zeichen /.  Muss in Hochkommata eingeschlossen werden, wenn alternative Datentypen zulässig sind, Separatoren verwendet werden oder das erste Zeichen ?, ! bzw. ^ ist. Zwischen Groß- und Kleinschreibung wird unterschieden.
posix-pathname	A...Z 0...9 Sonderzeichen Strukturkennzeichen: Schrägstrich	Eingabeformat: [/]part <sub>1</sub> [/.../part <sub>n</sub> ] wobei part <sub>i</sub> ein posix-filename ist; maximal 1023 Zeichen; muss in Hochkommata eingeschlossen werden, wenn alternative Datentypen zulässig sind, Separatoren verwendet werden oder das erste Zeichen ?, ! bzw. ^ ist.

Tabelle 2: Datentypen (Teil 4 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
product-version	A...Z 0...9 Punkt Hochkomma	<p>Eingabeformat: <math>[[C]][V][m]m.naso[']</math></p> <div style="text-align: right; margin-right: 20px;"> <math>\begin{array}{c}   \\   \\ \text{Korrekturstand} \\ \text{Freigabestand} \end{array}</math> </div> <p>wobei m, n, s und o jeweils eine Ziffer und a ein Buchstabe ist.                      Ob Freigabe- und/oder Korrekturstand angegeben werden dürfen oder ob sie angegeben werden müssen, bestimmen Zusätze zu dem Datentyp (siehe <a href="#">Tabelle 3</a>, Zusätze without-corr, without-man, mandatory-man und mandatory-corr).                      product-version kann in Hochkommata eingeschlossen werden, wobei der Buchstabe C vorangestellt werden kann. Die Versionsangabe kann mit dem Buchstaben V beginnen.</p>
structured-name	A...Z 0...9 \$, #, @ Bindestrich	<p>alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann;                      erstes Zeichen: A...Z oder \$, #, @</p>
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
time	0...9 Strukturkennzeichen: Doppelpunkt	<p>Angabe einer Tageszeit</p> <p>Eingabeformat: <math>\left. \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}</math></p> <p>hh: Stunden }                      mm: Minuten } führende Nullen können                      ss: Sekunden } weggelassen werden</p>
vsn	a) A...Z 0...9  b) A...Z 0...9 \$, #, @	<p>a) Eingabeformat: pvsid.folgenummer                      max. 6 Zeichen;                      pvsid: 2-4 Zeichen; Eingabe von PUB nicht erlaubt                      folgenummer: 1-3 Zeichen</p> <p>b) max. 6 Zeichen;                      PUB darf vorangestellt werden, dann dürfen jedoch nicht \$, #, @ folgen.</p>

Tabelle 2: Datentypen (Teil 5 von 6)

<b>Datentyp</b>	<b>Zeichenvorrat</b>	<b>Besonderheiten</b>
x-string	Sedezimal: 00...FF	ist in Hochkommata einzuschließen; der Buchstabe X muss vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.
x-text	Sedezimal: 00...FF	ist nicht in Hochkommata einzuschließen; der Buchstabe X darf nicht vorangestellt werden; die Anzahl der Zeichen darf ungerade sein.

Tabelle 2: Datentypen (Teil 6 von 6)

## Zusätze zu Datentypen

Zusatz	Bedeutung
<i>x..y unit</i>	<p>beim Datentyp integer: Intervallangabe</p> <p><i>x</i>      Mindestwert, der für integer erlaubt ist. <i>x</i> ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>y</i>      Maximalwert, der für integer erlaubt ist. <i>y</i> ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>unit</i>    Dimension. Folgende Angaben werden verwendet:</p> <p><i>days</i>      <i>byte</i></p> <p><i>hours</i>      <i>2Kbyte</i></p> <p><i>minutes</i>    <i>4Kbyte</i></p> <p><i>seconds</i>    <i>Mbyte</i></p> <p><i>milliseconds</i></p>
<i>x..y special</i>	<p>bei den übrigen Datentypen: Längenangabe</p> <p>Bei den Datentypen <i>catid</i>, <i>date</i>, <i>device</i>, <i>product-version</i>, <i>time</i> und <i>vsn</i> wird die Längenangabe nicht angezeigt.</p> <p><i>x</i>      Mindestlänge für den Operandenwert; <i>x</i> ist eine ganze Zahl.</p> <p><i>y</i>      Maximallänge für den Operandenwert; <i>y</i> ist eine ganze Zahl.</p> <p><i>x=y</i>    Der Operandenwert muss genau die Länge <i>x</i> haben.</p> <p><i>special</i> Zusatzangabe zur Beschreibung eines Sonderdatentyps, der durch die Implementierung geprüft wird. Vor <i>special</i> können weitere Zusätze stehen. Folgende Angaben werden verwendet:</p> <p><i>arithm-expr</i>    arithmetischer Ausdruck (SDF-P)</p> <p><i>bool-expr</i>      logischer Ausdruck (SDF-P)</p> <p><i>string-expr</i>    String-Ausdruck (SDF-P)</p> <p><i>expr</i>            beliebiger Ausdruck (SDF-P)</p> <p><i>cond-expr</i>     bedingter Ausdruck (JV)</p>
<i>with</i>	Erweitert die Angabemöglichkeiten für einen Datentyp.
<i>-compl</i>	<p>Bei Angaben zu dem Datentyp <i>date</i> ergänzt SDF zweistellige Jahresangaben der Form <i>jj-mm-tt</i> zu:</p> <p>    20<i>jj-mm-tt</i>    falls <i>jj</i> &lt; 60</p> <p>    19<i>jj-mm-tt</i>    falls <i>jj</i> ≥ 60</p>
<i>-low</i>	Groß- und Kleinschreibung wird unterschieden.
<i>-path-compl</i>	Bei Angaben zu dem Datentyp <i>filename</i> ergänzt SDF die Katalog- und/oder die Benutzerkennung, falls diese nicht angegeben werden.
<i>-under</i>	Erlaubt Unterstriche ' _ ' bei den Datentypen <i>name</i> und <i>composed-name</i> .

Tabelle 3: Zusätze zu Datentypen (Teil 1 von 7)



Zusatz	Bedeutung
with (Forts.) -wild(n)	<p>Teile eines Namens dürfen durch die folgenden Platzhalter ersetzt werden. n bezeichnet die maximale Eingabelänge bei Verwendung von Platzhaltern. Mit Einführung der Datentypen posix-filename und posix-pathname akzeptiert SDF neben den bisher im BS2000 üblichen Platzhaltern auch Platzhalter aus der UNIX-Welt (nachfolgend POSIX-Platzhalter genannt). Da derzeit nicht alle Kommandos POSIX-Platzhalter unterstützen, kann ihre Verwendung bei Datentypen ungleich posix-filename und posix-pathname zu Semantikfehlern führen.</p> <p>Innerhalb einer Musterzeichenfolge sollten entweder nur BS2000- oder nur POSIX-Platzhalter verwendet werden. Bei den Datentypen posix-filename und posix-pathname sind nur POSIX-Platzhalter erlaubt. Ist eine Musterzeichenfolge mehrdeutig auf einen String abbildbar, gilt der erste Treffer.</p>
BS2000-Platzhalter	Bedeutung
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.
Punkt am Ende	Teilqualifizierte Angabe eines Namens. Entspricht implizit der Zeichenfolge „/*“, d.h. nach dem Punkt folgt mindestens ein beliebiges Zeichen.
/	Ersetzt genau ein beliebiges Zeichen.
<s <sub>x</sub> :s <sub>y</sub> >	Ersetzt eine Zeichenfolge, für die gilt: <ul style="list-style-type: none"> <li>– sie ist mindestens so lang wie die kürzeste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie ist höchstens so lang wie die längste Zeichenfolge (s<sub>x</sub> oder s<sub>y</sub>)</li> <li>– sie liegt in der alphabetischen Sortierung zwischen s<sub>x</sub> und s<sub>y</sub>; Zahlen werden hinter Buchstaben sortiert (A...Z, 0...9)</li> <li>– s<sub>x</sub> darf auch die leere Zeichenfolge sein, die in der alphabetischen Sortierung an erster Stelle steht</li> <li>– s<sub>y</sub> darf auch die leere Zeichenfolge sein, die an dieser Stelle für die Zeichenfolge mit der höchst möglichen Codierung steht (enthält nur die Zeichen X'FF' )</li> </ul>

Tabelle 3: Zusätze zu Datentypen (Teil 2 von 7)

Zusatz	Bedeutung	
with-wild(n) (Forts.)	<s <sub>1</sub> ,...>	Ersetzt alle Zeichenfolgen, auf die eine der mit s angegebenen Zeichenkombinationen zutrifft. s kann auch die leere Zeichenfolge sein. Jede Zeichenfolge s kann auch eine Bereichsangabe „s <sub>x</sub> :s <sub>y</sub> “ sein (siehe Seite 17).
	-s	Ersetzt alle Zeichenfolgen, die der angegebenen Zeichenfolge s nicht entsprechen. Das Minuszeichen darf nur am Beginn der Zeichenfolge stehen. Innerhalb der Datentypen filename bzw. partial-filename kann die negierte Zeichenfolge -s genau einmal verwendet werden, d.h., -s kann einen der drei Namens-teile cat, user oder datei ersetzen.
Platzhalter sind in Generations- und Versionsangaben von Dateinamen nicht erlaubt. In Benutzerkennungen ist die Angabe von Platzhaltern der Systemverwaltung vorbehalten. Platzhalter können nicht die Begrenzer der Namensteile cat (Doppelpunkte) und user (\$ und Punkt) ersetzen.		
POSIX- Platzhalter	Bedeutung	
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.	
?	Ersetzt genau ein beliebiges Zeichen. Ist als erstes Zeichen außerhalb von Hochkommata nicht zulässig.	
[c <sub>x</sub> -c <sub>y</sub> ]	Ersetzt genau ein Zeichen aus dem Bereich c <sub>x</sub> und c <sub>y</sub> einschließlich der Bereichsgrenzen. c <sub>x</sub> und c <sub>y</sub> müssen einfache Zeichen sein.	
[s]	Ersetzt genau ein Zeichen aus der Zeichenfolge s. Die Ausdrücke [c <sub>x</sub> -c <sub>y</sub> ] und [s] können kombiniert werden zu [s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	
[!c <sub>x</sub> -c <sub>y</sub> ]	Ersetzt genau ein Zeichen, das nicht in dem Bereich c <sub>x</sub> und c <sub>y</sub> einschließlich der Bereichsgrenzen enthalten ist. c <sub>x</sub> und c <sub>y</sub> müssen einfache Zeichen sein. Die Ausdrücke [!c <sub>x</sub> -c <sub>y</sub> ] und [!s] können kombiniert werden zu [!s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	
[!s]	Ersetzt genau ein Zeichen, das nicht in der Zeichenfolge s enthalten ist. Die Ausdrücke [!s] und [!c <sub>x</sub> -c <sub>y</sub> ] können kombiniert werden zu [!s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	

Tabelle 3: Zusätze zu Datentypen (Teil 3 von 7)

Zusatz	Bedeutung										
<p>with (Forts.)</p> <p>-wild-constr(n)</p>	<p>Angabe einer Konstruktionszeichenfolge, die angibt, wie aus einer zuvor angegebenen Auswahlzeichenfolge mit Musterzeichen (siehe with-wild) neue Namen zu bilden sind. n bezeichnet die maximale Eingabelänge bei Verwendung von Platzhaltern.</p> <p>Die Konstruktionszeichenfolge kann aus konstanten Zeichenfolgen und Musterzeichen bestehen. Ein Musterzeichen wird durch diejenige Zeichenfolge ersetzt, die durch das entsprechende Musterzeichen in der Auswahlzeichenfolge ausgewählt wird.</p> <p>Folgende Platzhalter können zur Konstruktionsangabe verwendet werden:</p> <table border="1" data-bbox="341 541 1227 948"> <thead> <tr> <th data-bbox="341 541 488 578">Platzhalter</th> <th data-bbox="488 541 1227 578">Bedeutung</th> </tr> </thead> <tbody> <tr> <td data-bbox="341 578 488 657">*</td> <td data-bbox="488 578 1227 657">Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td data-bbox="341 657 488 799">Punkt am Ende</td> <td data-bbox="488 657 1227 799">Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td data-bbox="341 799 488 878">/ oder ?</td> <td data-bbox="488 799 1227 878">Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td data-bbox="341 878 488 948">&lt;n&gt;</td> <td data-bbox="488 878 1227 948">Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = &lt;integer&gt;</td> </tr> </tbody> </table> <p>Zuordnung der Platzhalter zu entsprechenden Platzhaltern in der Auswahlzeichenfolge:</p> <p>In der Auswahlzeichenfolge werden alle Platzhalter von links nach rechts aufsteigend nummeriert (globaler Index).</p> <p>Gleiche Platzhalter in der Auswahlzeichenfolge werden zusätzlich von links nach rechts aufsteigend nummeriert (platzhalter-spezifischer Index).</p> <p>In der Konstruktionsangabe können Platzhalter auf zwei, sich gegenseitig ausschließende Arten angegeben werden:</p> <ol style="list-style-type: none"> <li>1. Platzhalter werden über den globalen Index angegeben: &lt;n&gt;</li> <li>2. Angabe desselben Platzhalters, wobei die Ersetzung gemäß dem platzhalter-spezifischen Index entsprechend erfolgt: z.B. der zweite „/“ entspricht der Zeichenfolge, die durch den zweiten „/“ in der Auswahlzeichenfolge ausgewählt wird.</li> </ol>	Platzhalter	Bedeutung	*	Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.	Punkt am Ende	Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.	/ oder ?	Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.	<n>	Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = <integer>
Platzhalter	Bedeutung										
*	Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.										
Punkt am Ende	Entspricht der teilqualifizierten Angabe eines Namens in der Auswahlzeichenfolge. Entspricht der Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird.										
/ oder ?	Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.										
<n>	Entspricht der Zeichenfolge, die durch den n-ten Platzhalter in der Auswahlzeichenfolge ausgewählt wird; n = <integer>										

Tabelle 3: Zusätze zu Datentypen (Teil 4 von 7)

Zusatz	Bedeutung
with-wild-constr(n) (Forts.)	<p>Bei Konstruktionsangaben sind folgende Regeln zu beachten:</p> <ul style="list-style-type: none"> <li>– Die Konstruktionsangabe kann nur Platzhalter der Auswahlzeichenfolge enthalten.</li> <li>– Soll die Zeichenkette, die der Platzhalter &lt;...&gt; bzw. [...] auswählt, in der Konstruktionsangabe verwendet werden, muss die Index-Schreibweise gewählt werden.</li> <li>– Die Index-Schreibweise muss gewählt werden, wenn die Zeichenkette, die einen Platzhalter der Auswahlzeichenfolge bezeichnet, in der Konstruktionsangabe mehrfach verwendet werden soll: Bei der Auswahlangabe „A/“ muss z.B. statt „A//“ die Konstruktionszeichenfolge „A&lt;n&gt;&lt;n&gt;“ angegeben werden.</li> <li>– Der Platzhalter * kann auch die leere Zeichenkette sein. Insbesondere ist zu beachten, dass bei mehreren Sternen in Folge (auch mit weiteren Platzhaltern) nur der letzte Stern eine nicht leere Zeichenfolge sein kann: z.B. bei „****“ oder „*/*“.</li> <li>– Aus der Konstruktionsangabe sollten gültige Namen entstehen. Darauf ist sowohl bei der Auswahlangabe als auch bei der Konstruktionsangabe zu achten.</li> <li>– Abhängig von der Konstruktionsangabe können aus unterschiedlichen Namen, die in der Auswahlangabe ausgewählt werden, identische Namen gebildet werden: z.B. „A/*“ wählt die Namen „A1“ und „A2“ aus; die Konstruktionsangabe „B*“ erzeugt für beide Namen denselben neuen Namen „B“. Um dies zu vermeiden, sollten in der Konstruktionsangabe alle Platzhalter der Auswahlangabe mindestens einmal verwendet werden.</li> <li>– Wird die Konstruktionsangabe mit einem Punkt abgeschlossen, so muss auch die Auswahlzeichenfolge mit einem Punkt enden. Die Zeichenfolge, die durch den Punkt am Ende der Auswahlzeichenfolge ausgewählt wird, kann in der Konstruktionsangabe nicht über den globalen Index angegeben werden.</li> </ul>

Tabelle 3: Zusätze zu Datentypen (Teil 5 von 7)

Zusatz	Bedeutung																				
with-wild-constr(n) (Forts.)	Beispiele:																				
	<table border="1"> <thead> <tr> <th>Auswahlmuster</th> <th>Auswahl</th> <th>Konstruktionsmuster</th> <th>neuer Name</th> </tr> </thead> <tbody> <tr> <td>A/*</td> <td>AB1 AB2 A.B.C</td> <td>D&lt;3&gt;&lt;2&gt;</td> <td>D1 D2 D.CB</td> </tr> <tr> <td>C.&lt;A:C&gt;/&lt;D,F&gt;</td> <td>C.AAD C.ABD C.BAF C.BBF</td> <td>G.&lt;1&gt;.&lt;3&gt;.XY&lt;2&gt;</td> <td>G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB</td> </tr> <tr> <td>C.&lt;A:C&gt;/&lt;D,F&gt;</td> <td>C.AAD C.ABD C.BAF C.BBF</td> <td>G.&lt;1&gt;.&lt;2&gt;.XY&lt;2&gt;</td> <td>G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB</td> </tr> <tr> <td>A/B</td> <td>ACDB ACEB AC.B A.CB</td> <td>G/XY/</td> <td>GCXYD GCXYE GCXY.<sup>1)</sup> G.XYC</td> </tr> </tbody> </table>	Auswahlmuster	Auswahl	Konstruktionsmuster	neuer Name	A/*	AB1 AB2 A.B.C	D<3><2>	D1 D2 D.CB	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<3>.XY<2>	G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<2>.XY<2>	G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB	A/B	ACDB ACEB AC.B A.CB	G/XY/	GCXYD GCXYE GCXY. <sup>1)</sup> G.XYC
	Auswahlmuster	Auswahl	Konstruktionsmuster	neuer Name																	
	A/*	AB1 AB2 A.B.C	D<3><2>	D1 D2 D.CB																	
	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<3>.XY<2>	G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB																	
C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<2>.XY<2>	G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB																		
A/B	ACDB ACEB AC.B A.CB	G/XY/	GCXYD GCXYE GCXY. <sup>1)</sup> G.XYC																		
<sup>1)</sup> Punkt am Ende des Namens kann Namenskonvention widersprechen (z.B bei vollqualifizierten Dateinamen)																					
without	Schränkt die Angabemöglichkeiten für einen Datentyp ein.																				
-cat	Die Angabe einer Katalogkennung ist nicht erlaubt.																				
-corr	Eingabeformat: [[C]']V[m].na['] Angaben zum Datentyp product-version dürfen den Korrekturstand nicht enthalten.																				
-gen	Die Angabe einer Dateigeneration oder Dateigenerationsgruppe ist nicht erlaubt.																				
-man	Eingabeformat: [[C]']V[m].n['] Angaben zum Datentyp product-version dürfen weder Freigabe- noch Korrekturstand enthalten.																				
-odd	Der Datentyp x-text erlaubt nur eine gerade Anzahl von Zeichen.																				
-sep	Beim Datentyp text ist die Angabe der folgenden Trennzeichen nicht erlaubt: ; = ( ) < > _ (also Strichpunkt, Gleichheitszeichen, runde Klammer auf und zu, Größerzeichen, Kleinerzeichen und Leerzeichen)																				
-temp-file	Die Angabe einer temporären Datei ist nicht erlaubt (siehe #datei bzw. @datei bei filename).																				

Tabelle 3: Zusätze zu Datentypen (Teil 6 von 7)

Zusatz	Bedeutung
without (Forts.)	
-user	Die Angabe einer Benutzerkennung ist nicht erlaubt.
-vers	Die Angabe der Version (siehe „datei(nr)“) ist bei Banddateien nicht erlaubt.
-wild	Die Datentypen posix-filename bzw. posix-pathname dürfen keine Musterzeichen enthalten.
mandatory	Bestimmte Angaben sind für einen Datentyp zwingend erforderlich.
-corr	Eingabeformat: <code>[[C]][V][m].na[so][']</code> Angaben zum Datentyp product-version müssen den Korrekturstand (und damit auch den Freigabestand) enthalten.
-man	Eingabeformat: <code>[[C]][V][m].na[so][']</code> Angaben zum Datentyp product-version müssen den Freigabestand enthalten. Die Angabe des Korrekturstands ist optional möglich, wenn dies nicht durch den Zusatz without-corr untersagt wird.
-quotes	Angaben zu den Datentypen posix-filename bzw. posix-pathname müssen in Hochkommata eingeschlossen werden.

Tabelle 3: Zusätze zu Datentypen (Teil 7 von 7)

### 1.5.3 Metasyntax für die Beschreibung der S-Variablen-Ausgabe

Die Ausgabe der SHOW-Kommandos kann ab SDF V4.0 und SDF-P V2.0 in S-Variablen umgelenkt werden. Für die Beschreibung dieser Form der Ausgabe wird folgendes Tabellenformat verwendet:

Ausgabe-Information	Name der S-Variablen	T	Inhalt
(Kurzbeschreibung der Ausgabe-Information)	(z.B. var.MENU-LOG, wobei var eine deklarierte S-Variable sein muss)	Typ der S-Variablen: S: STRING I: INTEGER	(mögliche Werte, mit der die S-Variable belegt sein kann)

Der Name der S-Variablen kann den Zusatz (\*LIST) enthalten. Dieser Zusatz zeigt an, dass es sich um eine Listenvariable handelt. (\*LIST) erscheint, wenn der Inhalt der S-Variablen mit dem Kommando SHOW-VAR ausgegeben wird. Bei Zuweisungen darf der Benutzer (\*LIST) nicht angeben.

Besonderheiten und Hinweise zu den einzelnen Ausgabe-Informationen werden im Anschluss an die Tabelle erklärt.

Hinweise für die Arbeit mit der S-Variablenausgabe enthält das Handbuch „Kommandos, Band 6“ [4]. Dort sind auch die S-Variablen der SHOW-Kommandos aus „Kommandos, Band 4 und 5“ [3] beschrieben.





---

## 2 Arbeitsweise von SDF

Der Kommandoprozessor SDF (System Dialog Facility) unterstützt die Eingabe von BS2000-Kommandos und Programmanweisungen im SDF-Format. Das SDF-Format steht ab BS2000 V10.0 immer als Kommandosprache zur Verfügung. SDF kann nicht deaktiviert werden.

### Syntaxdateien

Zur Verarbeitung von Kommandos und Anweisungen benötigt SDF Informationen, die in Syntaxdateien zur Verfügung gestellt werden:

- Syntaxbeschreibung von Kommandos
- Informationen über die Implementierung der BS2000-Kommandos, z.B.
  - die Namen der System-Entries oder Prozeduren, über die die Kommandoausführung veranlasst wird
  - die Festlegungen für die Parameterübergaben an die ausführenden System-Module
- Informationen über Privilegien, die der aufrufende Benutzer besitzen muss
- Informationen über die Programme, die ihre Anweisungen über SDF einlesen, und die Syntaxbeschreibungen dieser Anweisungen
- allgemeine und kommandospezifische Festlegungen für die Dialogführung
- erläuternde Texte zu den Kommandos bzw. Anweisungen und ihren Operanden (Hilfexte).

Bei der Verarbeitung einer Kommando- oder Anweisungseingabe holt sich SDF die dazu benötigten Informationen anhand einer festgelegten Hierarchie aus den aktivierten Syntaxdateien. Syntaxdateien werden mit dem Softwareprodukt SDF-A erstellt (siehe Benutzerhandbuch „SDF-A“ [7]).

### *Syntaxdatei-Arten*

Es gibt drei Arten von Syntaxdateien:

- Systemsyntaxdateien enthalten die Syntax der allen Benutzern zur Verfügung stehenden Kommandos und Anweisungen. Ab SDF V3.0 können mehrere Systemsyntaxdateien gleichzeitig aktiviert sein. Dabei muss mindestens die Basis-Systemsyntaxdatei vorhanden sein und es kann mehrere Subsystem-Syntaxdateien geben.
- Gruppensyntaxdateien werden Benutzerkennungen zugeordnet. Sie können den Funktionsumfang der Systemsyntaxdateien einschränken oder, besonders für die privilegierte Benutzerkennung TSOS, erweitern.
- Benutzersyntaxdateien kann sich der Benutzer selbst erstellen und zuweisen. Sie können den Funktionsumfang der Systemsyntaxdateien und der Gruppensyntaxdatei (falls vorhanden) einschränken. Erweiterungen und sonstige funktionelle Änderungen sind beschränkt auf Anweisungen an Benutzerprogramme und auf Kommandos, die durch Prozeduren implementiert sind. Ab SDF V4.1 können pro Benutzerauftrag mehrere Benutzersyntaxdateien gleichzeitig aktiviert sein.

Bei System- und Gruppensyntaxdateien wird ferner folgende Unterscheidung vorgenommen:

- Softwareeinheitssyntaxdateien (SESDs) enthalten die Syntax eines bestimmten Produktes, wie SDF selbst oder BS2CP für den Grundausbau des BS2000. Softwareeinheitssyntaxdateien werden von Fujitsu Siemens Computers ausgeliefert.
- Installationssyntaxdateien (INSDs) sind aus SESDs und benutzereigenen Syntaxdateien gemischte Dateien, die von SDF als Syntaxdateien im laufenden System benutzt werden können. Eine Systemsyntaxdatei (INSD), die individuell auf ein Rechenzentrum zugeschnitten ist, kann vor der ersten Installation von SDF mit Hilfe des Dienstprogrammes SDF-I erzeugt werden.

Die Syntax der zum BS2000 gehörenden Kommandos und Programme wird von Fujitsu Siemens Computers produktspezifisch in Softwareeinheitssyntaxdateien (SESDs) ausgeliefert.

### *Sicherheitsaspekte*

Den Objekten in allen 3 Arten von Syntaxdateien können Privilegien zugeordnet werden, die den Zugriff auf diese Objekte einschränken. Eine Benutzerkennung kann nur dann auf ein Objekt zugreifen, wenn sie mindestens eines der dem Objekt zugeordneten Privilegien besitzt. Eine Benutzerkennung kann auch mehrere Privilegien haben. Durch die Auswertung der Privilegien wird es möglich, alle Objekte in einer Syntaxdatei zu definieren, auf die alle Benutzerkennungen Zugriff haben.

Wenn das Softwarepaket SECOS im Einsatz ist, können bestimmte Syntaxdatei-Ereignisse wie z.B das Aktivieren von Syntaxdateien protokolliert werden (siehe Benutzerhandbuch „SECOS“ [13]).

## Dienstprogramme für die Verwaltung bzw. Bearbeitung von Syntaxdateien

Mit dem Dienstprogramm **SDF-I** können die mit den Softwareprodukten ausgelieferten und auch die mit SDF-A (siehe [7]) selbst erstellten System- bzw. Gruppensyntaxdateien in eine System- oder Gruppensyntaxdatei eingemischt werden. Syntaxdateien von nicht mehr benötigten Produkten können auch wieder entmischt werden. Zusätzlich können mit SDF-I Syntaxdateien mit unterschiedlichen Dateiformaten konvertiert und Informationen über Dateiformat und Inhalt ausgegeben werden.

Das Dienstprogramm **SDF-U** bietet Möglichkeiten zur Modifikation von Syntaxdateien. Es bedient sich dabei einer Untermenge der Funktionen und Anweisungen des Softwareproduktes SDF-A. Folgende Modifikationsmöglichkeiten bestehen:

- neue bzw. geänderte Kommando- und Programmdefinitionen in eine Syntaxdatei einbringen (Anweisung COPY)
- Standardwerte (Vorbereitung) der SDF-Optionen in einer Syntaxdatei ändern (Anweisung SET-GLOBALS)
- Inhalte einer Syntaxdatei auf SYSLSST oder SYSOUT ausgeben (Anweisung SHOW)
- Austauschen der Prozedur, durch die ein Kommando implementiert ist (Anweisung MODIFY-CMD)

Weitergehende Möglichkeiten zur Bearbeitung von Syntaxdateien und zur Erstellung benutzereigener Syntaxdateien bietet das Softwareprodukt SDF-A (siehe gleichnamiges Handbuch [7]).

Mit dem Dienstprogramm **SDF-PAR** kann eine SDF-Parameterdatei erzeugt, geändert und angezeigt werden.

System- und Gruppensyntaxdateien, die mit SDF-A, SDF-I oder SDF-U erzeugt wurden, können entweder mit dem Kommando MODIFY-SDF-PARAMETERS oder mit dem Dienstprogramm SDF-PAR aktiviert bzw. deaktiviert werden. Benutzersyntaxdateien werden mit dem Kommando MODIFY-SDF-OPTIONS aktiviert bzw. deaktiviert.

## Kommando-Returncodes

SDF liefert dem Benutzer Informationen über die Analyse der Kommandoeingabe und -ausführung in einem Kommando-Returncode. Dieser Kommando-Returncode ist vergleichbar mit dem Returncode auf Programmebene und ermöglicht es dem Benutzer, auf bestimmte Fehlersituationen gezielt zu reagieren. Der Kommando-Returncode besteht aus drei Teilen:

- dem Maincode, der einem Meldungsschlüssel entspricht, über den mit dem Kommando HELP-MSG-INFORMATION detaillierte Informationen abgefragt werden können.
- dem Subcode1, der die aufgetretene Fehlersituation in eine Fehlerklasse einordnet, aus der abgeleitet werden kann, wie schwerwiegend ein Fehler ist. Der Wert von Subcode1 wird dezimal ausgegeben.
- dem Subcode2, der Zusatzinformationen (Wert ungleich Null) enthalten kann. Der Wert von Subcode2 wird dezimal ausgegeben. Für den Fehlerfall (also Subcode1 ungleich Null) ist die Verwendung von Subcode2 keinen Regeln unterworfen. Subcode2 kann außer Null die Werte 1 oder 2 annehmen, wenn kein Fehler aufgetreten ist. Subcode2 gleich 1 zeigt an, dass die angeforderte Leistung bereits vor Absetzen des Kommandos bestand. Subcode2 gleich 2 zeigt eine Sondersituation an und ist als Warnung einzustufen.

Eine ausführliche Beschreibung der Kommando-Returncodes finden Sie im Handbuch „Einführung in die Dialogschnittstelle SDF“ [1] bzw. „Kommandos Band 1“ [3]. Der Kommando-Returncode kann mit SDF-P-Mitteln nur in S-Prozeduren und Dialogblöcken ausgewertet werden (siehe Handbuch „SDF-P“ [14] bzw. Beschreibung des Kommandos IF-BLOCK-ERROR im Handbuch „Kommandos Band 2“ [3]).

### *Darstellung des Kommando-Returncodes*

Kommando-Returncodes werden in Tabellenform in der Reihenfolge Subcode2, Subcode1, Maincode und Bedeutungstext dargestellt. Ein Subcode2 mit dem Wert gleich Null (d.h. keine Zusatzinformation vorhanden) wird nicht in der Tabelle aufgeführt.

Sofern in einer Tabelle für einen bestimmten Kommando-Returncode Meldungen garantiert sind, enthält die Überschrift „Bedeutung“ den Zusatz „ / garantierte Meldungen“ und die Meldungsschlüssel der garantierten Meldungen werden nach dem Bedeutungstext aufgezählt. Die Bedeutung dieser Meldungsschlüssel kann sich der Benutzer mit dem Kommando HELP-MSG ausgeben lassen.

Beispiel:

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
1	0	CMD0001	Kommando fehlerfrei beendet
	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
	64	CMD0554	Kommando ohne Erfolg ausgeführt garantierte Meldungen: CMD0300, CMD0302, CMD0490, CMD0508, CMD0509, CMD0552, CMD0554, CMD0555, CMD0579

### *Allgemeine Kommando-Returncodes*

Kommando-Returncodes, die ein BS2000-Kommando bei der Ausführung zurückgeben kann, sind Bestandteil der jeweiligen Kommandobeschreibung. In der Kommandobeschreibung nicht enthalten sind allgemeine Kommando-Returncodes, die automatisch von SDF zurückgegeben werden:

- wenn ein Fehler bereits vor der Kommandoausführung von SDF erkannt wird (z.B. Syntaxfehler).
- wenn das Kommando nicht ausgeführt werden kann.
- wenn das Kommando, d.h. das Ausführungsmodul, selbst keinen Kommando-Returncode zurückgibt. In diesem Fall sind in der Kommandobeschreibung keine spezifischen Kommando-Returncodes enthalten.

Die Beschreibung der allgemeinen Kommando-Returncodes finden Sie im Handbuch „Einführung in die Dialogschnittstelle SDF“ [1] bzw. „Kommandos Band 1“ [3].

## 2.1 SDF-Parameter

Die Namen für die System- und Gruppensyntaxdateien werden nur beim ersten Aktivieren festgelegt. In einer SDF-Parameterdatei sind die Namen der zu aktivierenden System- und Gruppensyntaxdateien (siehe [Seite 31f](#)) sowie die Namen der automatisch abzuarbeitenden LOGON- bzw. LOGOFF-Prozeduren (siehe [Seite 34](#)) hinterlegt. Änderungen in der Parameterdatei können mit dem Kommando MODIFY-SDF-PARAMETERS oder mit dem Dienstprogramm SDF-PAR vorgenommen werden.

Beim Starten von SDF werden die Informationen aus der Parameterdatei in den Speicher gelesen. Anhand dieser Informationen werden die Syntaxdateien aktiviert. Mit dem Kommando MODIFY-SDF-PARAMETERS kann die Systembetreuung sowohl die Namen in der Parameterdatei als auch im Speicher ändern und damit festlegen, wie lang die Änderungen Gültigkeit haben, d.h. ob sie permanent oder nur für die aktuelle Session gelten.

Bei der Installation von SDF wird eine Parameterdatei mit dem Standardnamen SYSPAR.SDF erzeugt. Eine Parameterdatei beliebigen Namens kann mit dem Kommando MODIFY-SDF-PARAMETERS oder mit dem Dienstprogramm SDF-PAR erzeugt werden.

Zur Steuerung von SDF und zur Zuweisung von Syntaxdateien stehen zwei SDF-Kommandos zur Verfügung:

- **MODIFY-SDF-OPTIONS**
  - trifft task-lokale Festlegungen für die Eingabe und Verarbeitung von Kommandos und Anweisungen
  - aktiviert oder deaktiviert Benutzersyntaxdateien
  - gilt nur für den jeweiligen Benutzerauftrag
- **MODIFY-SDF-PARAMETERS**
  - ändert die aktuellen und/oder permanenten Festlegungen für System- und Gruppensyntaxdateien
  - ändert die aktuellen und/oder permanenten Festlegungen für systemweite LOGON- und LOGOFF-Prozeduren
  - gilt systemweit

Beim Wechsel von Syntaxdateien ist Folgendes zu beachten:

- Die Systemsyntaxdateien werden sofort systemweit für alle Benutzeraufträge gewechselt.
- Die Gruppensyntaxdatei, die beim Starten des Auftrags gültig war, bleibt bis zur Beendigung des Auftrags gültig. Aufträge, die nach dem Wechsel gestartet werden, arbeiten mit der neuen Gruppensyntaxdatei.
- Eine Benutzersyntaxdatei wird sofort gewechselt.

## 2.2 Systemsyntaxdateien

### 2.2.1 Basis-Systemsyntaxdatei

Die Basis-Systemsyntaxdatei (INSD) wird nach dem Laden von SDF automatisch aktiviert. Dabei wird die in der Parameterdatei festgelegte Datei aktiviert. Ist in den DSSM-Deklarationen für SDF keine Parameterdatei festgelegt, wird \$TSOS.SYSPAR.SDF als Parameterdatei verwendet. Hat die Parameterdatei keinen gültigen Inhalt, so wird über die Konsolmeldung CMD0691 ein neuer Name angefordert. Bei Antwort „\*STD“ wird \$TSOS.SYSSDF.SDF.045 als Basis-Systemsyntaxdatei aktiviert (und zusätzlich \$TSOS.SYSSDF.BS2CP.<bs2vers> als Subsystem-Syntaxdatei).

Während des Systemlaufs kann die Systembetreuung die aktivierte Basis-Systemsyntaxdatei wechseln (siehe Kommando MODIFY-SDF-PARAMETERS). Die während des Systemlaufs mit MODIFY-SDF-PARAMETERS aktivierte Basis-Systemsyntaxdatei kann einen beliebigen Namen haben.

### 2.2.2 Subsystem-Syntaxdateien

Grundsätzlich sind folgende Aktivierungsarten der Subsystem-Syntaxdateien zu unterscheiden:

1. Subsystem-Syntaxdateien, die beim Erzeugen eines Subsystems (START-SUBSYSTEM, siehe Handbuch „Verwaltung von Subsystemen“ [16]) implizit aktiviert werden.
2. Lokale Subsystem-Syntaxdateien, die beim Starten eines lokalen Subsystems implizit aktiviert werden (siehe Handbuch „Verwaltung von Subsystemen“ [16], Kommando START-LOCAL-SUBSYSTEM).
3. Subsystem-Syntaxdateien, die explizit aktiviert werden. Sie werden wie die Basis-Systemsyntaxdatei automatisch aktiviert, falls ihre Namen in der SDF-Parameterdatei eingetragen sind. Die Systemverwaltung kann aber auch während des Systemlaufs mit dem Kommando MODIFY-SDF-PARAMETERS Subsystem-Syntaxdateien aktivieren oder austauschen. Dieser Wechsel der Subsystem-Syntaxdateien wirkt in Abhängigkeit des Operanden SCOPE für alle bestehenden oder künftigen Tasks.



SESDs für Subsysteme, die vor „SYSTEM READY“ geladen werden (Subsystem-Startattribut CREATION-TIME=\*BEFORE-DSSM-LOAD/\*AT-DSSM-LOAD), müssen als Subsystem-Syntaxdatei in die SDF-Parameterdatei eingetragen sein. Dabei muss darauf geachtet werden, dass die Subsystem-Syntaxdatei bereits bei der Systemeinführung zugreifbar ist (liegt z.B. auf dem Home-Pubset). Subsystem-Syntaxdateien müssen mit USER-ACCESS=\*SPECIAL katalogisiert sein. Bei der Installation sorgt IMON automatisch für die korrekten Einträge.

## 2.3 Gruppensyntaxdatei

Eine Gruppensyntaxdatei kann nach der LOGON-Verarbeitung automatisch aktiviert werden. Dazu muss diese Gruppensyntaxdatei der Benutzererkennung über eine PROFILE-ID zugewiesen werden.

Dies geschieht folgendermaßen:

- Mit dem Operanden PROFILE-ID der Kommandos MODIFY-USER-ATTRIBUTES oder ADD-USER wird für die entsprechende Benutzererkennung eine PROFILE-ID vereinbart.
- Mit dem Kommando MODIFY-SDF-PARAMETERS wird der PROFILE-ID eine Gruppensyntaxdatei zugeordnet.

Nach dem LOGON sucht SDF zuerst die PROFILE-ID, die der Benutzererkennung im Benutzerkatalog-Eintrag zugeordnet ist. Gibt es eine PROFILE-ID, so verwendet SDF die Gruppensyntaxdatei, die dieser PROFILE-ID aktuell zugeordnet ist.

Ist der PROFILE-ID keine Datei zugeordnet, so wird der Benutzerauftrag ohne Gruppensyntaxdatei gestartet.

Ist der Benutzererkennung keine PROFILE-ID zugeordnet, wird keine Gruppensyntaxdatei aktiviert.

Das oben beschriebene Verfahren hat folgenden Vorteil: Ist eine Gruppensyntaxdatei mehreren Kennungen zugeordnet, so müssen beim Wechsel der Gruppensyntaxdatei nicht mehrere Benutzerkatalog-Einträge geändert werden; jetzt wird diesen Kennungen dieselbe PROFILE-ID zugeordnet und deren Zuweisung zentral in der SDF-Parameterdatei geändert.

Mit Ausnahme der Gruppensyntaxdateien für die Systemverwaltung müssen die Gruppensyntaxdateien mehrbenutzbar sein. Bei Schutz über Basic-ACL, ACL oder GUARDS müssen die zugeordneten Kennungen mindestens das Ausführungsrecht besitzen. Eine eventuell vorhandene Gruppensyntaxdatei für die Wartungskennung muss mit USER-ACCESS=\*SPECIAL katalogisiert sein.

Die Benutzerkatalog-Einträge einiger Systemkennungen (z.B. TSOS) sind standardmäßig mit bestimmten PROFILE-IDs vorbesetzt. Eine entsprechende Vorbesetzung gilt auch für die Standard-Parameterdatei. Zur Vermeidung von Namenskonflikten dürfen vom Benutzer vereinbarte PROFILE-IDs nicht mit der Zeichenfolge „SYS-“ beginnen.

Eine spezielle Gruppensyntaxdatei für die Systembetreuung (Benutzererkennung TSOS) wird nicht benötigt, da in den Systemsyntaxdateien der erweiterte Kommandovorrat bereits durch die Zuordnung des Systemprivilegs TSOS festgelegt ist. Die privilegierten Kommandos bzw. Operanden stehen damit nur der Benutzererkennung zur Verfügung, die das Systemprivileg TSOS besitzt. SDF prüft automatisch anhand der Einträge im Benutzerkatalog, welches Systemprivileg einer Benutzererkennung zugeordnet ist.



Falls eine TSOS-Gruppensyntaxdatei verwendet wird, können folgende Situationen auftreten:

- Die TSOS-Gruppensyntaxdatei ist nicht in der SDF-Parameterdatei definiert. Mit dem Kommando `MODIFY-SDF-PARAMETERS` kann die TSOS-Gruppensyntaxdatei jederzeit in der SDF-Parameterdatei deklariert werden.
- Die TSOS-Gruppensyntaxdatei ist in der SDF-Parameterdatei definiert. Wenn diese Gruppensyntaxdatei nicht existiert, erhält die Systembetreuung die Meldung `CMD0300` und die Verarbeitung wird fortgesetzt. Die Deklaration der TSOS-Gruppensyntaxdatei kann mit dem Kommando `MODIFY-SDF-PARAMETERS` jederzeit aus der SDF-Parameterdatei entfernt werden.

## 2.4 LOGON- und LOGOFF-Prozeduren

Bei der LOGON- und LOGOFF-Verarbeitung können im Dialog- und Stapelbetrieb automatisch Prozeduren zum Ablauf gebracht werden. LOGON- bzw. LOGOFF-Prozeduren können jeweils als Call- und Include-Prozedur sowohl systemweit als auch benutzerspezifisch bereitgestellt werden. LOGON-Prozeduren werden ab SDF V1.4, LOGOFF-Prozeduren ab SDF V4.4 unterstützt.

- Die benutzerspezifischen Prozedurdateien legt der Benutzer selbst an. Er kann dabei für die LOGON- und LOGOFF-Verarbeitung jeweils eine Call- und eine Include-Prozedur anlegen. Der automatische Start erfolgt, wenn die Prozeduren folgende Standardnamen besitzen:
  - \$userid.SYS.SDF.LOGON.USERPROC (Call-Prozedur für LOGON)
  - \$userid.SYS.SDF.LOGON.USERINCL (Include-Prozedur für LOGON)
  - \$userid.SYS.SDF.LOGOFF.USERPROC (Call-Prozedur für LOGOFF)
  - \$userid.SYS.SDF.LOGOFF.USERINCL (Include-Prozedur für LOGOFF)
- Die systemweiten Prozedurdateien legt die Systembetreuung an. Sie kann dabei für die LOGON- und LOGOFF-Verarbeitung ebenfalls jeweils eine Call- und eine Include-Prozedur bereitstellen. Mit dem Kommando MODIFY-SDF-PARAMETERS werden LOGON- und LOGOFF-Prozeduren aktiviert bzw. deaktiviert. Die Aktivierung bzw. Deaktivierung kann über den Operanden SCOPE für die aktuelle Session temporär oder permanent ab sofort bzw. ab der folgenden Session vereinbart werden. Permanente Änderungen werden auch in der SDF-Parameterdatei eingetragen. Alle Vereinbarungen können mit dem Kommando SHOW-SDF-PARAMETERS angezeigt werden. Bei der Aktivierung wird für den gewünschten Prozedurtyp (z.B. im Operanden SYSTEM-LOGON-PROC bei der Call-Prozedur für LOGON) explizit der Dateiname angegeben oder, falls die Prozedur unter einem Standardnamen abgelegt ist, der Wert \*STD vereinbart. Folgende Standardnamen können verwendet werden:
  - \$TSOS.SYS.SDF.LOGON.SYSPROC (Call-Prozedur für LOGON)
  - \$TSOS.SYS.SDF.LOGON.SYSINCL (Include-Prozedur für LOGON)
  - \$TSOS.SYS.SDF.LOGOFF.USERPROC (Call-Prozedur für LOGOFF)
  - \$TSOS.SYS.SDF.LOGOFF.USERINCL (Include-Prozedur für LOGOFF)

Die systemweiten Prozeduren müssen die Schutzattribute USER-ACCESS=\*ALL-USERS und ACCESS=\*READ besitzen. Falls für eine Prozedurdatei eine BASIC-ACL eingerichtet ist, müssen alle Benutzer das Ausführungsrecht besitzen.

Die Namen der systemweiten Prozedurdateien können mit dem Kommando MODIFY-SDF-PARAMETERS in die SDF-Parameterdatei eingetragen werden. Mit diesem Kommando ist es auch möglich, die systemweiten LOGON-Prozeduren für den nächsten LOGON-Vorgang der aktuellen Session zu deaktivieren, ohne dass die Einträge in der Parameterdatei geändert werden.

### *Aufrufreihenfolge*

Sind für einen Benutzer sowohl systemweite als auch benutzerspezifische Prozedurdateien angelegt, so laufen bei der LOGON-Verarbeitung zuerst die systemweiten LOGON-Prozeduren ab, bei der LOGOFF-Verarbeitung zuerst die benutzerspezifischen LOGOFF-Prozeduren.

Eine Include-Prozedur wird immer vor der entsprechenden Call-Prozedur gestartet.

### *Einschränkungen*

Kann eine benutzerspezifische oder systemweite LOGON- bzw. LOGOFF-Prozedurdatei nicht eröffnet werden, wird eine Fehlermeldung ausgegeben. Die LOGON- bzw. LOGOFF-Verarbeitung wird dann ohne Aktivierung dieser Prozedur fortgesetzt.

Ist der Benutzerkennung eines Benutzerauftrags eine mit HIERARCHY=\*NO (Kommando MODIFY-SDF-PARAMETERS) definierte PROFILE-ID zugeordnet, so werden für diesen Benutzerauftrag keine systemweiten LOGON- bzw. LOGOFF-Prozeduren aufgerufen; nur die benutzerspezifischen LOGON- bzw. LOGOFF-Prozeduren, falls vorhanden, werden dann aufgerufen.

LOGON- bzw. LOGOFF-Prozeduren werden in folgenden Fällen ohne Warnung ignoriert:

- Die Prozedurdatei ist nur katalogisiert, belegt aber keinen Speicherplatz.
- Die Task ist eine RFA-Tasks.
- Die Task besitzt außer HARDWARE-MAINTENANCE, SECURITY-ADMINISTRATION, SAT-FILE-MANAGEMENT und SAT-FILE-EVALUATION kein anderes Privileg.
- Nur bei LOGOFF-Prozedur:  
Die Task wurde mit CANCEL-JOB bzw. FORCE-JOB-CANCEL abgebrochen.

Für einige Systemtasks werden ebenfalls keine LOGON- bzw. LOGOFF-Prozeduren gestartet (z.B. bei der Systemeinleitung).

### *Unterbrechbarkeit*

Die LOGON-Prozeduren sollten mit dem Operanden INTERRUPT-ALLOWED=\*NO des Kommandos BEGIN-PROCEDURE bzw. SET-PROCEDURE-OPTIONS als nicht unterbrechbar deklariert werden. Diese Vorgehensweise ist besonders bei Verwendung geschachtelter Prozeduren empfehlenswert. Das Kommando DO wird innerhalb von LOGON-Prozeduren nicht ausgeführt. LOGON-Prozeduren können auch nicht mit dem Kommando ENDP-RESUME beendet werden.



# 3 Installation von SDF

Detaillierte Informationen zur Installation von SDF und zu Hard- und Softwarevoraussetzungen finden Sie in der Freigabemitteilung zum Softwareprodukt SDF.

## 3.1 SDF-Installationsdateien

SDF V4.5 ist ab BS2000/OSD-BC V3.0 ablauffähig.

Die folgenden Dateien sind Bestandteil der Standardlieferung und werden mit SDF V4.5A ausgeliefert:

Datei	Bemerkung
SIPLIB.SDF.045	TPR-Makrobibliothek
SPMLNK.SDF.045	Ladebibliothek für SDF V4.5A (für Anlagen mit SPARC-Architektur)
SRMLNK.SDF.045	Ladebibliothek für SDF V4.5A (für Anlagen mit RISC-Architektur)
SYSFGM.SDF.045.D	Freigabemitteilung für SDF (deutsch)
SYSFGM.SDF.045.E	Freigabemitteilung für SDF (englisch)
SYSLIB.SDF.045	PLAM-Bibliothek, die die SDF-Makros, die Definitionen der Schnittstellen-Funktionen für COBOL, FORTRAN und C, das Bindemodul CMDCSTM und die Schnittstellen-Module für die höheren Programmiersprachen enthält.
SYSLNK.SDF.045	Ladebibliothek für SDF V4.5A (für Anlagen mit /390-Architektur)
SYSMES.SDF.045	SDF-Meldungsdatei
SYSREP.SDF.045	Korrekturen für SDF V4.5A (Dummy-Item)
SYSRMS.SDF.045	RMS-Liefermenge für SDF V4.5A
SYSSDF.SDF.045	Syntaxdatei für die Kommandos aus dem Anwendungsbereich SDF
SYSSII.SDF.045 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON
SYSSSC.SDF.045.120	SSCM-Katalog-Deklarationen für das Subsystem SDF (BS2000/OSD V3.0 und Versionen ≥ V4.0)
SYSSSC.SDF.045.121	SSCM-Katalog-Deklarationen für das Subsystem SDF (BS2000/OSD V3.1)

Datei	Bemerkung
SYSMSV.SDF-I.041	Vollständige Meldungs-Primärdatei für SDF-I
SYSMES.SDF-I.041	Meldungsdatei für SDF-I
SYSPRG.SDF-I.041	Programm zum Mischen von Softwareeinheitssyntaxdateien (siehe <a href="#">Kapitel „SDF-I“ auf Seite 75</a> )
SYSSDF.SDF-I.041	Syntaxdatei, die das Kommando START-SDF-I enthält
SYSSII.SDF-I.041 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON
SYSPRG.SDF-U.041	Programm, mit dem die Systembetreuung Syntaxdateien modifizieren kann (siehe <a href="#">Kapitel „SDF-U“ auf Seite 97</a> )
SYSSDF.SDF-U.041	Syntaxdatei, die die SDF-U-Anweisungen enthält
SYSLNK.SDF-U.041	Programmodule von SDF-U
SYSMSV.SDF-U.041	Vollständige Meldungs-Primärdatei für SDF-U
SYSMES.SDF-U.041	Name der Meldungsdatei für SDF-U
SYSREP.SDF-U.041	Korrekturen für SDF-U (Dummy-Item)
SYSRMS.SDF-U.041	RMS-Liefermenge für SDF-U V4.1
SYSSII.SDF-U.041 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON
SYSPRG.SDF-PAR.011	Programm zu Erzeugen, Ändern und Anzeigen der SDF-Parameterdatei (siehe <a href="#">Kapitel „SDF-PAR“ auf Seite 127</a> )
SYSMSV.SDF-PAR.011	Vollständige Meldungs-Primärdatei für SDF-PAR
SYSMES.SDF-PAR.011	Meldungsdatei für SDF-PAR
SYSPRC. SDF-PAR.011.100	Startprozedur für SDF-PAR (BS2000/OSD < V2.0)
SYSPRC. SDF-PAR.011.112	Startprozedur für SDF-PAR (BS2000/OSD ≥ V2.0)
SYSSDF.SDF-PAR.010.USER	Benutzersyntaxdatei, die die SDF-PAR-Anweisungen enthält
SYSSII.SDF-PAR.011 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON

<sup>1)</sup> SYSSII-Dateien werden nach erfolgreicher Installation im Zielsystem gelöscht. Sie sind allerdings als X-Elemente in der Bibliothek \$SYSSAG.SOLLIB.IMON.SYSSII gesichert.

## 3.2 Subsystemkatalog generieren

SDF wird über DSSM geladen. Dazu muss das Subsystem SDF im Subsystemkatalog deklariert werden. Die Generierung des Subsystemkatalogs mit SSCM ist im Handbuch „Einführung in die Systembetreuung“ [2] beschrieben.

Die Deklarationen dazu werden in der Datei **SYSSSC.SDF.045.120** (bzw. mit dem Suffix 121 für BS2000/OSD V3.1) ausgeliefert.

Der Name der SDF-Parameterdatei ist in den ausgelieferten Deklarationen als Subsystem-Attribut mit \$TSOS.SYSPAR.SDF festgelegt.

Im Normalfall erfolgt die Generierung des Subsystemkatalogs durch IMON.

## 3.3 SDF bereitstellen

### 3.3.1 Dateien für SDF bereitstellen

Die Bereitstellung der Dateien für SDF und die Aktivierung der benötigten Systemsyntax- und Meldungsdateien erfolgen durch den Installationsmonitor IMON.



IMON aktiviert die Syntaxdatei von SDF (d.h. die Datei SYSSDF.SDF.045) als Basis-Systemsyntaxdatei. Alle anderen Systemsyntaxdateien einschließlich SYSSDF.BS2CP.<bs2vers> werden als Subsystem-Syntaxdateien aktiviert.

### 3.3.2 SDF starten

SDF wird automatisch beim Startup geladen. Danach können mit dem Kommando MODIFY-SDF-PARAMETERS andere Namen für die Syntaxdateien vereinbart werden.





---

## 4 Kommandos zur SDF-Verwaltung

### 4.1 Funktionelle Übersicht

#### Syntaxdateien aktivieren und deaktivieren

MODIFY-SDF-OPTIONS	SDF-Einstellungen ändern und Benutzersyntaxdateien aktivieren oder deaktivieren
MODIFY-SDF-PARAMETERS	System-, Subsystem- und Gruppensyntaxdateien zuweisen und systemweite Logon-/Logoff-Prozeduren aktivieren oder deaktivieren

#### Anzeigefunktionen

SHOW-SDF-OPTIONS	Taskspezifische Information über aktivierte Syntaxdateien und die Festlegungen für die Anweisungseingabe und -verarbeitung ausgeben
SHOW-SDF-PARAMETERS	Zugewiesene Gruppen- und Systemsyntaxdateien anzeigen
SHOW-SYNTAX-VERSIONS	Ausgeben, welche Syntaxdateien in die aktiven Syntaxdateien integriert sind

Ist das Subsystem SDF-P geladen, kann die Ausgabe der SHOW-Kommandos in S-Variablen umgelenkt werden. Die Umlenkung der Ausgaben ist z.B. möglich, wenn Sie das jeweilige SHOW-Kommando mit dem Kommando EXECUTE-CMD (siehe Handbuch „SDF-P“ [14]) oder mit dem CMD-Makro (siehe Handbuch „Makroaufrufe an den Ablaufteil“ [9]) ausführen. Informationen zur Ausgabe in S-Variablen finden Sie auch im Benutzerhandbuch „Kommandos, Band 6“ [4].

Die Beschreibung weiterer Kommandos des Anwendungsbereiches SDF (mit Ausnahme der START-Kommandos) finden Sie im Handbuch „Einführung in die Dialogschnittstelle SDF“ [1].

## 4.2 Beschreibung der Kommandos

### MODIFY-SDF-OPTIONS

#### Benutzersyntaxdateien aktivieren und SDF-Optionen ändern

<b>Anwendungsbereich:</b>	SDF
<b>Privilegierung:</b>	STD-PROCESSING TSOS HARDWARE-MAINTENANCE SAT-FILE-EVALUATION SAT-FILE-MANAGEMENT SECURITY-ADMINISTRATION

Mit dem Kommando MODIFY-SDF-OPTIONS

- aktivieren oder deaktivieren Sie Benutzersyntaxdateien
- ändern Sie folgende für Ihren Benutzerauftrag geltende Festlegungen für die Kommando-/Anweisungseingabe und -verarbeitung:
  - Art der Dialogführung (Operand GUIDANCE)
  - Protokollierungsumfang (Operand LOGGING)
  - Eingabeschnittstelle für Utilities (Operand UTILITY-INTERFACE)
  - Syntaxfehlerdialog in Prozeduren (Operand PROCEDURE-DIALOGUE)
  - Lage des Fortsetzungszeichens (Operand CONTINUATION)
  - Syntaxprüfung von Kommandos (Operand MODE)
  - Syntaxprüfung von Anweisungen (Operand DEFAULT-PROGRAM-NAME)
  - Wahl der Funktionstastenbelegung (Operand FUNCTION-KEYS)
  - Speicherung von Eingaben (Operand INPUT-HISTORY)

Bei Beginn der Task werden die Einstellungen der SDF-Optionen aus den Global-Informationen der aktivierten System- bzw. Gruppensyntaxdatei übernommen. Beim Aktivieren einer Benutzersyntaxdatei werden die Einstellungen entsprechend den darin enthaltenen Global-Informationen geändert. Unter dem Dienstprogramm OMNIS wird jedoch beim Wechsel der Benutzersyntaxdatei GUIDANCE=\*EXPERT eingestellt, unabhängig davon, was in der Global-Information der aktivierten Benutzersyntaxdatei festgelegt ist. Wenn eine Benutzersyntaxdatei den Namen \$<userid>.SDF.USER.SYNTAX hat und eine Task unter der Benutzerkennung <userid> gestartet wird, dann wird diese Benutzersyntaxdatei nach der LOGON-Verarbeitung automatisch aktiviert.

Der Operandenfragebogen des Kommandos MODIFY-SDF-OPTIONS enthält im (temporär) geführten Dialog als voreingestellte Werte die aktuellen Einstellungen, die Sie auch mit dem Kommando SHOW-SDF-OPTIONS abfragen können.

MODIFY-SDF-OPTIONS	Kurzname: MDSDFO
<b>SYNTAX-FILE = *UNCHANGED / *ADD(...) / *REMOVE(...) / *NONE</b>	
*ADD(...)	
<b>ADD-NAME = *STD / list-poss(2000): *STD / &lt;filename 1..54&gt;</b>	
*REMOVE(...)	
<b>REMOVE-NAME = *LAST / *ALL / *BY-SELECTION / list-poss(2000): &lt;filename 1..54&gt; / *STD</b>	
<b>,GUIDANCE = *UNCHANGED / *EXPERT / *NO / *MAXIMUM / *MEDIUM / *MINIMUM</b>	
<b>,LOGGING = *UNCHANGED / *INPUT-FORM / *ACCEPTED-FORM / *INVARIANT-FORM</b>	
<b>,UTILITY-INTERFACE = *UNCHANGED / *OLD-MODE / *NEW-MODE</b>	
<b>,PROCEDURE-DIALOGUE = *UNCHANGED / *YES / *NO</b>	
<b>,CONTINUATION = *UNCHANGED / *OLD-MODE / *NEW-MODE</b>	
<b>,MENU-LOGGING = *UNCHANGED / *NO / *YES</b>	
<b>,CMD-STATISTICS = *UNCHANGED / *NO / *YES</b>	
<b>,MODE = *UNCHANGED / *EXECUTION / *TEST(...)</b>	
*TEST(...)	
<b>CHECK-PRIVILEGES = *UNCHANGED / *NO / *YES</b>	
<b>,DEFAULT-PROGRAM-NAME = *UNCHANGED / *NONE / &lt;structured-name 1..30&gt;</b>	
<b>,FUNCTION-KEYS = *UNCHANGED / *STYLE-GUIDE-MODE / *BY-TERMINAL-TYPE / *OLD-MODE</b>	
<b>,INPUT-HISTORY = *UNCHANGED / *ON(...) / *OFF / *RESET</b>	
*ON(...)	
<b>NUMBER-OF-INPUTS = *UNCHANGED / &lt;integer 1..100&gt;</b>	
<b>PASSWORD-PROTECTION = *UNCHANGED / *NO / *YES</b>	

**SYNTAX-FILE =**

Gibt an, ob eine Benutzersyntaxdatei aktiviert oder deaktiviert werden soll. Mehrere Benutzersyntaxdateien können gleichzeitig aktiviert sein.

Eine Benutzersyntaxdatei, die aktiviert werden soll, muss zugreifbar sein. Ist der Auftraggeber nicht Eigentümer der Syntaxdatei, muss sie mehrbenutzbar sein. Ist die Syntaxdatei mit Basic-ACL, ACL oder GUARDS geschützt, muss für den Auftraggeber in jedem Fall mindestens das Ausführungsrecht gesetzt sein (siehe Handbuch „SECOS“ [13]).

Ist die Datei mit einem Kennwort gegen Ausführung geschützt, so muss das Kennwort in der Kennwort-Tabelle des Auftrags eingetragen sein (siehe Kommando ADD-PASSWORD, Handbuch „Kommandos“ [3]).

**SYNTAX-FILE = \*UNCHANGED**

Es gilt die bisherige Vereinbarung.

**SYNTAX-FILE = \*ADD(...)**

Eine bzw. mehrere Benutzersyntaxdateien sollen zusätzlich aktiviert werden.

**ADD-NAME = list-poss(2000): \*STD / <filename 1..54>**

Bestimmt die zu aktivierende Benutzersyntaxdatei.

In einer Liste können auch mehrere Syntaxdateien angegeben werden.

**ADD-NAME = \*STD**

Die Benutzersyntaxdatei mit dem Standardnamen \$<user-id>.SDF.USER.SYNTAX wird aktiviert. Ist diese nicht vorhanden, so wird keine zusätzliche Benutzersyntaxdatei aktiviert.

**ADD-NAME = <filename 1..54>**

Die angegebene Benutzersyntaxdatei wird aktiviert.

**SYNTAX-FILE = \*REMOVE(...)**

Eine bzw. mehrere Benutzersyntaxdateien werden deaktiviert.

**REMOVE-NAME = \*LAST / \*ALL / list-poss(2000): <filename 1..54> / \*STD**

Bestimmt die zu deaktivierende Benutzersyntaxdatei.

In einer Liste können auch mehrere Syntaxdateien angegeben werden.

**REMOVE-NAME = \*LAST**

Die zuletzt aktivierte Benutzersyntaxdatei wird deaktiviert.

**REMOVE-NAME = \*ALL**

Alle aktivierten Benutzersyntaxdateien werden deaktiviert.

**REMOVE-NAME = \*BY-SELECTION**

SDF bietet in einer Auswahlmaske alle aktivierten Benutzersyntaxdateien an. Die in der Maske ausgewählten Benutzersyntaxdateien werden deaktiviert.

**REMOVE-NAME = <filename 1..54>**

Die angegebene Benutzersyntaxdatei wird deaktiviert.

**REMOVE-NAME = \*STD**

Die Benutzersyntaxdatei mit dem Standardnamen \$<user-id>.SDF.USER.SYNTAX wird deaktiviert.

**SYNTAX-FILE = \*NONE**

Für den Auftrag ist keine Benutzersyntaxdatei aktiviert.

Alle momentan aktiven Benutzersyntaxdateien werden deaktiviert.

**GUIDANCE =**

Bestimmt die Art der Dialogführung.

**GUIDANCE = \*UNCHANGED**

Es gilt die bisherige Vereinbarung.

**GUIDANCE = \*EXPERT**

Eingestellt wird die EXPERT-Form des ungeführten Dialogs:

Das System fordert mit / zur Kommandoingabe bzw. mit %// zur Anweisungseingabe auf. Ein Syntaxfehlerdialog wird nicht geführt. Die geblockte Eingabe von Kommandos und Anweisungen ist im Dialog möglich, d.h., mehrere Kommandos bzw. Anweisungen können, getrennt durch das logische Zeilenende-Zeichen, gleichzeitig eingegeben werden.

**GUIDANCE = \*NO**

Eingestellt wird die NO-Form des ungeführten Dialogs:

Das System fordert, abhängig von der eingestellten Sprache (E oder D), mit „%CMD:“ oder „%KDO:“ zur Kommandoingabe bzw. mit „%STMT:“ oder „%ANW:“ zur Anweisungseingabe auf. Ein Syntaxfehlerdialog erlaubt die Korrektur einer fehlerhaften Eingabe ohne Wiederholung des gesamten Kommandos bzw. der Anweisung. Die geblockte Eingabe von Kommandos und Anweisungen ist im Dialog möglich, d.h. mehrere Kommandos bzw. Anweisungen können, getrennt durch das logische Zeilenende-Zeichen, gleichzeitig eingegeben werden.

**GUIDANCE = \*MAXIMUM**

Eingestellt wird der geführte Dialog mit maximaler Hilfestufe:

Angezeigt werden alle alternativen Operandenwerte mit Zusätzen und alle Hilfetexte der angezeigten Anwendungsbereiche, Kommandos, Anweisungen und Operanden.

**GUIDANCE = \*MEDIUM**

Eingestellt wird der geführte Dialog mit mittlerer Hilfestufe:

Angezeigt werden alle alternativen Operandenwerte ohne Zusätze und alle Hilfetexte der angezeigten Anwendungsbereiche, Kommandos, Anweisungen.

**GUIDANCE = \*MINIMUM**

Eingestellt wird der geführte Dialog mit minimaler Hilfestufe:

Angezeigt werden nur voreingestellte Operandenwerte ohne Zusätze.

**LOGGING =**

Bestimmt, wie die Eingaben protokolliert werden. Kennwörter und Angaben zu geheimen Operanden werden in Protokollen immer ausgeblendet.

**LOGGING = \*UNCHANGED**

Die geltende Festlegung für die Protokollierung bleibt unverändert.

**LOGGING = \*INPUT-FORM**

Im ungeführten Dialog werden die Eingabezeichenfolgen exakt protokolliert. Im geführten Dialog oder beim Fehlerdialog wird wie bei \*ACCEPTED-FORM protokolliert.

**LOGGING = \*ACCEPTED-FORM**

Protokolliert werden:

- alle Namen in ihrer Langform,
- jeder in der Eingabe vorkommende Operand mit seinem Namen und dem angegebenen Wert,
- die ggf. durch Korrekturen erzeugten Enddarstellungen.

Eingaben aus dem geführten Dialog werden zu einem String verkettet.

**LOGGING = \*INVARIANT-FORM**

Protokolliert werden:

- alle Namen mit dem in der Syntaxdatei festgelegten Standardnamen,
- jeder angegebene Operand mit seinem Namen und dem angegebenen Wert,
- jeder optionale Operand, der implizit in der Eingabe enthalten ist, mit seinem Namen und dem voreingestellten Default-Wert,
- die ggf. durch einen Korrekturdialog erzeugte Enddarstellung.

Eingaben aus dem geführten Dialog werden zu einem String verkettet.

**UTILITY-INTERFACE = \*UNCHANGED / \*OLD-MODE / \*NEW-MODE**

Steuert die Eingabeschnittstelle von Utilities, die parallel eine alte und eine neue (SDF-) Schnittstelle anbieten.

**PROCEDURE-DIALOGUE = \*UNCHANGED / YES / \*NO**

Gibt an, ob der Benutzer während eines Prozedurablaufs zur Korrektur fehlerhafter Prozedurkommandos aufgefordert werden soll (Syntaxfehlerdialog in Prozeduren). Der Operand steuert gleichzeitig die Hilfeanforderung („?“ in der Eingabe) innerhalb von Prozeduren.

**CONTINUATION = \*UNCHANGED / \*OLD-MODE / \*NEW-MODE**

Bestimmt, in welcher Spalte in Prozedur- und ENTER-Dateien das Fortsetzungszeichen für Kommandos anzugeben ist. Bei der Einstellung \*OLD-MODE muss das Fortsetzungszeichen „-“ genau in Spalte 72 angegeben werden. Bei \*NEW-MODE kann das Fortsetzungszeichen in den Spalten 2 bis 72 angegeben werden.

**MENU-LOGGING = \*UNCHANGED / \*NO / \*YES**

Gibt an, ob die Menüs des geführten Dialogs komplett protokolliert werden sollen. Dieser Operand ist nur für Diagnosezwecke vorgesehen.

**CMD-STATISTICS = \*UNCHANGED / \*NO / \*YES**

Bestimmt, ob eine Statistik der abgesetzten Kommandos aus der System-Syntaxdatei angelegt wird. Der Operand ist der Systembetreuung vorbehalten.

Ist die Statistik-Routine aktiv (CMD-STATISTICS=\*YES), zählt SDF jedes Kommando, das abgesetzt wird. Berücksichtigt werden alle Kommandos, die in der System- oder den Subsystem-Syntaxdateien enthalten sind.

**CMD-STATISTICS = \*NO**

Die Statistik-Routine wird ausgeschaltet. SDF wertet den Zählerstand aus und schreibt die aufbereitete Statistik in die Datei \$.SYS.SDF.CMD.STATISTICS.

Die aufbereitete Statistikdatei enthält:

- Eine Kopfzeile mit Datum und Uhrzeit, zu dem die Statistik-Routine aktiviert und wieder deaktiviert wurde.
- Für jedes Subsystem eine Überschrift mit dem Namen des Subsystems.
- Für jedes Kommando eines Subsystems den Kommandonamen und den Zählerstand.

**CMD-STATISTICS = \*YES**

Eine Kommandostatistik wird geführt. In der zentralen SDF-Tabelle wird bei jedem Aufruf eines Kommandos, das in der Basis-Systemsyntaxdatei oder in einer Subsystem-Syntaxdatei definiert ist, ein Zähler um eins erhöht. Beim Führen der Statistik muss mit erhöhtem Zeitbedarf gerechnet werden, da der Zugriff auf die zentrale SDF-Tabelle über einen Lock-Manager erfolgt. In den Systemsyntaxdateien definierte Alias-Namen werden wie Kommandos mitgezählt, da sie separat in der SDF-Kommandotabelle eingetragen sind.

**MODE =**

Bestimmt, ob die folgende Eingabe ausgeführt oder syntaktisch analysiert werden muss.

**MODE = \*UNCHANGED**

Die aktuelle Festlegung des TEST-Modus bleibt bestehen.

**MODE = \*EXECUTION**

Der Testmodus wird ausgeschaltet, d.h. Eingaben werden normal ausgeführt.

**MODE = \*TEST(...)**

Der Testmodus wird eingeschaltet. Das bedeutet, dass die nachfolgenden Kommandos nicht ausgeführt, sondern nur syntaktisch analysiert werden. Die Kommandos MODIFY-SDF-OPTIONS und SHOW-SDF-OPTIONS werden jedoch immer ausgeführt.

In S-Prozeduren werden auch SDF-P-Kontrollflusskommandos ausgeführt. Dies kann zu Fehlern führen, da die Kommandos, die S-Variablen deklarieren oder setzen, im Testmodus nicht ausgeführt werden. Zum Testen von S-Prozeduren stellt das kostenpflichtige Subsystem SDF-P eine eigene Testhilfe zur Verfügung (siehe Benutzerhandbuch „SDF-P“ [14]). Die Behandlung von Anweisungen im Testmodus kann im Operanden DEFAULT-PROGRAM-NAME festgelegt werden.

**CHECK-PRIVILEGES = \*UNCHANGED / \*NO / \*YES**

Bestimmt, ob zusätzlich zur Syntax geprüft wird, ob der Benutzer die für die Eingabe erforderlichen Privilegien besitzt.

**CHECK-PRIVILEGES = \*NO**

Die Privilegien des Benutzers werden nicht geprüft. Diese Einstellung kann z.B. dann erforderlich sein, wenn die zu testende Prozedur für andere Benutzer erstellt wird.

**CHECK-PRIVILEGES = \*YES**

Zusätzlich zur Syntax werden die Privilegien des Benutzers geprüft.

**DEFAULT-PROGRAM-NAME =**

Wird nur verwendet, wenn MODE=\*TEST(...) angegeben wurde und legt den Programmnamen für die Syntaxanalyse von Anweisungen fest. Alle Anweisungen, die unter diesem Programmnamen definiert wurden, werden anhand der Syntaxbeschreibungen analysiert. In S-Prozeduren können Anweisungen nicht überprüft werden.

**DEFAULT-PROGRAM-NAME = \*UNCHANGED**

Die aktuelle Festlegung bezüglich des Programmnamens bleibt unverändert.

**DEFAULT-PROGRAM-NAME = \*NONE**

Bei MODE=\*TEST(...) werden alle Anweisungen ignoriert.

**DEFAULT-PROGRAM-NAME = <structured-name 1..30>**

Programmname (ADD-PROGRAM NAME= ..., siehe „SDF-A“ [7]). Bei MODE=\*TEST(...) wird die Syntaxanalyse für das Programm mit dem angegebenen Namen ausgeführt. Die Syntaxanalyse ist unabhängig vom Programmaufruf, da die Kommandos LOAD-/START-EXECUTABLE-PROGRAM (bzw. LOAD-/START-PROGRAM) im Testmodus nicht ausgeführt werden.

**FUNCTION-KEYS =**

Legt die Belegung der Funktionstasten fest. Die verschiedenen Modi sind ausführlich im Benutzerhandbuch „Einführung in die Dialogschnittstelle SDF“ [1] beschrieben. Nicht unterstützte Funktionstasten wirken nicht wie [DUE], sondern bewirken die leere Leistung.

**FUNCTION-KEYS = \*UNCHANGED**

Die Belegung der Funktionstasten wird nicht geändert.

**FUNCTION-KEYS = \*STYLE-GUIDE-MODE**

Die Belegung der Funktionstasten entspricht dem Fujitsu-Siemens-Styleguide. In diesem Modus gelten folgende Tastenbelegungen:

<u>[K2]</u>	Unterbrechungs-Funktion
<u>[F1]</u>	Hilfe-Funktion
<u>[F3]</u>	Exit-Funktion
<u>[F5]</u>	Refresh-Funktion (nur im geführten Dialog)
<u>[F6]</u>	Exit-all-Funktion
<u>[F7]</u>	Rückwärts blättern (nur im geführten Dialog)
<u>[F8]</u>	Vorwärts blättern (nur im geführten Dialog)
<u>[F9]</u>	RESTORE-SDF-INPUT INPUT=*LAST ausführen
<u>[F11]</u>	Execute-Funktion (nur im geführten Dialog)
<u>[F12]</u>	Cancel-Funktion



**FUNCTION-KEYS = \*BY-TERMINAL-TYPE**

Die Belegung der Funktionstasten ist abhängig von dem Terminaltyp. Unterstützt der Terminaltyp die umfangreichere Styleguide-Funktionalität, wählt SDF die Einstellung \*STYLE-GUIDE-MODE. Anderenfalls wählt SDF die Einstellung \*OLD-MODE.



Bei der Einstellung wird ausgewertet, mit welchem Typ das Terminal im System generiert wurde. Weicht der generierte Terminaltyp vom tatsächlichen Terminaltyp ab, ist nicht gewährleistet, dass die Einstellung dem tatsächlich unterstützten Funktionsumfang entspricht. Bei einer Terminalemulation kann der erkannte Terminaltyp sowohl von der Generierung als auch von Umgebungsvariablen abhängen. Genaueres ist der Beschreibung des Emulationsprogrammes zu entnehmen.

**FUNCTION-KEYS = \*OLD-MODE**

Die Belegung der Funktionstasten erfolgt im bisherigen Modus, der von allen Terminaltypen unterstützt wird. In diesem Modus gelten folgende Tastenbelegungen:

- K1**      Exit-Funktion
- K2**      Unterbrechungs-Funktion
- K3**      Refresh-Funktion (nur im geführten Dialog)
- F1**      Exit-all-Funktion
- F2**      Test-Funktion (nur im geführten Dialog)
- F3**      Execute-Funktion (nur im geführten Dialog)

**INPUT-HISTORY = \*UNCHANGED / \*ON(...) / \*OFF / \*RESET**

Bestimmt, ob der Eingabepuffer eingeschaltet, ausgeschaltet oder zurückgesetzt wird.

**INPUT-HISTORY = \*ON(...)**

Der Eingabepuffer wird eingeschaltet. SDF speichert syntaktisch richtige Eingaben (Kommandos und Anweisungen) im Eingabepuffer. Nicht gespeichert werden SET-LOGON-PARAMETERS, RESTORE-SDF-INPUT und SHOW-INPUT-HISTORY.

Ob ISP-Kommandos gespeichert werden, ist abhängig von der Angabe im Operanden PASSWORD-PROTECTION.

Mit SHOW-INPUT-HISTORY kann sich der Benutzer die gespeicherten Eingaben ausgeben lassen. Mit RESTORE-SDF-INPUT kann der Benutzer eine bestimmte Eingabe ausgeben lassen, um sie dann unverändert oder auch modifiziert erneut abzuschicken.



Wertangaben für „geheime“ Operanden, die weder dem Default-Wert noch einem mit SECRET=\*NO definierten Wert entsprechen, werden im Eingabepuffer abhängig vom Operanden PASSWORD-PROTECTION mit „^“ oder im Klartext gespeichert.

Wertangaben zu nicht „geheimen“ Operanden werden im Eingabepuffer im Klartext gespeichert. Im Einzelfall können diese Eingaben aus Benutzersicht auch schützenswerte Informationen darstellen (z.B. Prozedurparameter). Soll verhindert werden, dass diese Eingaben mit SHOW-INPUT-HISTORY bzw. RESTORE-SDF-INPUT erneut am Bildschirm sichtbar gemacht werden können, so kann der

Benutzer wie folgt vorgehen:

Vor sicherheitsrelevanten Eingaben den Eingabepuffer ausschalten und danach wieder einschalten. Wurden die Eingaben bereits gespeichert, kann der Eingabepuffer mit \*RESET zurückgesetzt werden. In diesem Fall werden jedoch alle gespeicherten Eingaben gelöscht.

**NUMBER-OF-INPUTS = \*UNCHANGED / <integer 1..100>**

Legt die maximale Anzahl der Eingaben fest, die in dem Puffer gespeichert werden. Bei Erreichen der maximalen Anzahl gespeicherter Eingaben wird die jeweils älteste Eingabe gelöscht.

**PASSWORD-PROTECTION = \*UNCHANGED / \*NO / \*YES**

Bestimmt, ob Wertangaben für „geheime“ Operanden und ISP-Kommandos im Eingabepuffer gespeichert werden.

**PASSWORD-PROTECTION = \*NO**

Wertangaben für „geheime“ Operanden werden im Klartext gespeichert. ISP-Kommandos werden ebenfalls im Eingabepuffer gespeichert.

**PASSWORD-PROTECTION = \*YES**

Wertangaben für „geheime“ Operanden werden mit „^“ (entspricht dem Operandenwert \*SECRET) gespeichert. ISP-Kommandos werden nicht im Eingabepuffer gespeichert.

**INPUT-HISTORY = \*OFF**

Der Eingabepuffer wird ausgeschaltet. Die nachfolgenden Eingaben werden nicht gespeichert. Alle bisher gespeicherten Eingaben sind jedoch weiterhin zugreifbar.

**INPUT-HISTORY = \*RESET**

Der Eingabepuffer wird zurückgesetzt. Die bereits gespeicherten Eingaben werden gelöscht und sind nicht mehr zugreifbar. Die nachfolgenden Eingaben werden wieder gespeichert.

**Kommando-Returncodes**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei beendet
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
	64	CMD0554	Kommando ohne Erfolg ausgeführt garantierte Meldungen: CMD0300, CMD0302, CMD0490, CMD0508, CMD0509, CMD0552, CMD0554, CMD0555, CMD0579

**Beispiel**

```

/show-sdf-opt _____ (1)
%SYNTAX FILES CURRENTLY ACTIVATED :
% SYSTEM      : :20SH:$TSOS.SYSSDF.SDF.045
%             VERSION : SESD04.5A300
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.ACS.140
%             VERSION : SESD14.0B100
.
.
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.SDF-A.041
%             VERSION : SESD04.1E10
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.TASKDATE.140
%             VERSION : SESD14.0A100
% GROUP       : *NONE
% USER       : :20SG:$USER1.SDF.USER.SYNTAX
%             VERSION : 13.11.2000
%CURRENT SDF OPTIONS :
% GUIDANCE    : *EXPERT
% LOGGING     : *INPUT-FORM
% CONTINUATION : *NEW-MODE
% UTILITY-INTERFACE : *NEW-MODE
% PROCEDURE-DIALOGUE : *NO
% MENU-LOGGING : *NO
% MODE        : *EXECUTION
% CHECK-PRIVILEGES : *YES
% DEFAULT-PROGRAM-NAME : *NONE
% FUNCTION-KEYS : *STYLE-GUIDE-MODE
% INPUT-HISTORY : *ON
% NUMBER-OF-INPUTS : 20
% PASSWORD-PROTECTION: *YES
/modify-sdf-opt syntax-file=*add(syssdf.example.03) _____ (2)
/show-sdf-opt information=*user _____ (3)
% USER       : :20SG:$USER1.SDF.USER.SYNTAX
%             VERSION : 13.11.2000
% USER       : :20SG:$USER1.SYSSDF.EXAMPLE.03
%             VERSION : 16.10.2001
%CURRENT SDF OPTIONS :
% GUIDANCE    : *EXPERT
% LOGGING     : *INPUT-FORM
% CONTINUATION : *NEW-MODE
% UTILITY-INTERFACE : *NEW-MODE
% PROCEDURE-DIALOGUE : *NO
% MENU-LOGGING : *NO
% MODE        : *EXECUTION
% CHECK-PRIVILEGES : *YES
% DEFAULT-PROGRAM-NAME : *NONE
% FUNCTION-KEYS : *STYLE-GUIDE-MODE

```

```

% INPUT-HISTORY      : *ON
%   NUMBER-OF-INPUTS : 20
%   PASSWORD-PROTECTION: *YES
/modify-sdf-opt guid=*max _____ (4)
.
.
(im geführten Dialog:)
NEXT = mod-sdf-opt synt-file=*remove,guid=*expert _____ (5)
.
/show-sdf-opt information=*user _____ (6)
% USER      : :20SG:$USER1.SDF.USER.SYNTAX
%           VERSION : 13.11.2000
%CURRENT SDF OPTIONS :
% GUIDANCE      : *EXPERT
% LOGGING       : *INPUT-FORM
% CONTINUATION  : *NEW-MODE
% UTILITY-INTERFACE : *NEW-MODE
% PROCEDURE-DIALOGUE : *NO
% MENU-LOGGING  : *NO
% MODE          : *EXECUTION
% CHECK-PRIVILEGES : *YES
% DEFAULT-PROGRAM-NAME : *NONE
% FUNCTION-KEYS  : *STYLE-GUIDE-MODE
% INPUT-HISTORY  : *ON
%   NUMBER-OF-INPUTS : 20
%   PASSWORD-PROTECTION: *YES

```

- (1) Mit SHOW-SDF-OPTIONS werden Informationen über die für diesen Benutzerauftrag aktivierten Syntaxdateien und die geltenden Festlegungen für die Kommandoverarbeitung angefordert: Neben der Basis-Systemsyntaxdatei und mehreren Subsystem-Syntaxdateien ist die Benutzersyntaxdatei SDF.USER.SYNTAX aktiviert. Die Benutzerführung ist auf den Expertenmodus eingestellt.
- (2) Zusätzlich wird eine Benutzersyntaxdatei SYSSDF.EXAMPLE.03 aktiviert.
- (3) Die benutzerspezifischen Informationen werden angezeigt. Die alte Benutzersyntaxdatei SDF.USER.SYNTAX ist weiterhin aktiviert. Zusätzlich ist auch die Benutzersyntaxdatei SYSSDF.EXAMPLE.03 aktiviert. Für Kommandos, die in beiden Benutzersyntaxdateien definiert sind, wird die Kommandodefinition aus SYSSDF.EXAMPLE.03 verwendet, da diese zuletzt aktiviert wurde.
- (4) Es wird in den geführten Dialog umgeschaltet.
- (5) Die zuletzt aktivierte Benutzersyntaxdatei wird deaktiviert und wieder in den ungeführten Dialog gewechselt.
- (6) Die erneute Ausgabe der benutzerspezifischen SDF-Optionen zeigt, dass tatsächlich nur die Benutzersyntaxdatei SYSSDF.EXAMPLE.03 deaktiviert wurde.

## MODIFY-SDF-PARAMETERS

### SDF-Parameter ändern

**Anwendungsbereich:** SDF

**Privilegierung:** TSOS

Mit dem Kommando MODIFY-SDF-PARAMETERS erzeugt bzw. ändert die Systembetreuung die Einträge in der SDF-Parameterdatei und kann damit die Systemsyntaxdateien austauschen, die Zuordnung von Gruppensyntaxdateien zu Benutzern bestimmen und systemweite LOGON- bzw. LOGOFF-Prozeduren (je eine Call- und eine Include-Prozedur) definieren.

#### MODIFY-SDF-PARAMETERS

**SCOPE** = \*TEMPORARY / \*PERMANENT / \*NEXT-SESSION(...)

\*NEXT-SESSION(...)

| **PARAMETER-FILE-NAME** = \*CURRENT / <filename 1..54>

,**SYNTAX-FILE-TYPE** = \*UNCHANGED / \*SYSTEM(...) / \*GROUP(...) / \*SUBSYSTEM(...)

\*SYSTEM(...)

| **NAME** = <filename 1..54>

\*GROUP(...)

| **NAME** = \*NONE / <filename 1..54>

| ,**PROFILE-ID** = <structured-name 1..30> / <filename 1..54>

| ,**HIERARCHY** = \*YES / \*NO

\*SUBSYSTEM(...)

| **NAME** = \*NONE / <filename 1..54>

| ,**SUBSYSTEM-NAME** = <structured-name 1..8>

,**SYSTEM-LOGON-PROC** = \*UNCHANGED / \*NO / \*STD / <filename 1..54>

,**SYSTEM-LOGON-INCL** = \*UNCHANGED / \*NO / \*STD / <filename 1..54>

,**SYSTEM-LOGOFF-PROC** = \*UNCHANGED / \*NO / \*STD / <filename 1..54>

,**SYSTEM-LOGOFF-INCL** = \*UNCHANGED / \*NO / \*STD / <filename 1..54>

**SCOPE =**

Bestimmt den Umfang der Aktivierung. Bei SCOPE=\*TEMPORARY oder \*PERMANENT kann die Gruppensyntaxdatei, die zur PROFILE-ID der Benutzerkennung TSOS gehört, nicht deaktiviert werden. Bei SCOPE=\*PERMANENT oder \*NEXT-SESSION wird eine SDF-Parameterdatei erstellt, falls sie noch nicht existiert.

**SCOPE = \*TEMPORARY**

*Typ Systemsyntaxdatei:*

Die angegebene Systemsyntaxdatei wird systemweit für alle Benutzeraufträge aktiviert. Sie wird jedoch nicht in der SDF-Parameterdatei gespeichert und auch nicht für die nächste Session berücksichtigt.

*Typ Subsystem-Syntaxdatei:*

Die angegebene Subsystem-Syntaxdatei wird systemweit für alle Benutzeraufträge aktiviert oder deaktiviert. Sie wird jedoch nicht in der SDF-Parameterdatei gespeichert und auch nicht für die nächste Session berücksichtigt.

*Typ Gruppensyntaxdatei:*

Die angegebene Gruppensyntaxdatei wird der angegebenen PROFILE-ID zugeordnet. Diese Zuordnung wirkt sich nur für zukünftige Benutzeraufträge aus (folgende LOGON-Verarbeitung in der aktuellen Session). Bereits existierende Benutzeraufträge werden davon nicht berührt. Der angegebene Gruppensyntaxdateiname wird nicht in der SDF-Parameterdatei gespeichert und deshalb in der nächsten Session nicht berücksichtigt. Im Falle einer Deaktivierung (\*NONE) werden die bestehende PROFILE-ID und der zugehörige Gruppensyntaxdateiname *nicht* aus der SDF-Parameterdatei gelöscht.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = \*NO:*

Die systemweite LOGON-Prozedur wird für die nächsten LOGON-Verarbeitungen in der aktuellen Session deaktiviert. Die SDF-Parameterdatei wird nicht verändert.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = <filename 1..54> oder \*STD:*

Die angegebene systemweite LOGON-Prozedur wird für alle weiteren LOGON-Verarbeitungen der aktuellen Session aktiviert. Da die SDF-Parameterdatei nicht verändert wird, gilt diese Festlegung nur für die aktuelle Session.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = \*NO:*

Die systemweite LOGOFF-Prozedur wird für alle folgenden LOGOFF-Verarbeitungen in der aktuellen Session deaktiviert. Die SDF-Parameterdatei wird nicht verändert.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = <filename 1..54> oder \*STD:*

Die angegebene systemweite LOGOFF-Prozedur wird für alle folgenden LOGOFF-Verarbeitungen der aktuellen Session aktiviert. Da die SDF-Parameterdatei nicht verändert wird, gilt diese Festlegung nur für die aktuelle Session.

**SCOPE = \*PERMANENT**

*Typ Systemsyntaxdatei:*

Die angegebene Systemsyntaxdatei wird systemweit für alle Benutzeraufträge aktiviert. Zusätzlich wird dieser Syntaxdateiname in der SDF-Parameterdatei gespeichert. Er wird daher in der nächsten Session berücksichtigt.

*Typ Subsystem-Syntaxdatei:*

Die angegebene Subsystem-Syntaxdatei wird systemweit für alle Benutzeraufträge aktiviert. Zusätzlich wird dieser Syntaxdateiname in der SDF-Parameterdatei gespeichert. Er wird daher in der nächsten Session berücksichtigt. Im Falle der Deaktivierung der Subsystem-Syntaxdatei werden die Namen des Subsystems und der zugehörigen Subsystem-Syntaxdatei aus der SDF-Parameterdatei gelöscht.

*Typ Gruppensyntaxdatei:*

Die angegebene Gruppensyntaxdatei wird der angegebenen PROFILE-ID zugeordnet. Diese Zuordnung wirkt sich nur für zukünftige Benutzeraufträge aus (folgendes LOGON in der aktuellen Session). Bereits existierende Benutzeraufträge werden davon nicht berührt. Die angegebene Gruppensyntaxdatei wird in der SDF-Parameterdatei gespeichert und daher in der nächsten Session berücksichtigt. Im Falle einer Deaktivierung (\*NONE) werden die bestehende PROFILE-ID und der zugehörige Gruppensyntaxdateiname aus der SDF-Parameterdatei gelöscht.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = \*NO:*

Die systemweite LOGON-Prozedur wird für die nächsten LOGON-Verarbeitungen in der aktuellen Session deaktiviert. Ihr Name wird aus der SDF-Parameterdatei gelöscht. Wenn vorhanden, wird ab der nächsten Session die systemweite LOGON-Prozedur mit dem Standardnamen verwendet.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = <filename 1..54> oder \*STD:*

Die angegebene systemweite LOGON-Prozedur wird für alle weiteren LOGON-Verarbeitungen der aktuellen Session aktiviert. Ihr Name wird in der SDF-Parameterdatei gespeichert und daher auch in der nächsten Session berücksichtigt.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = \*NO:*

Die systemweite LOGOFF-Prozedur wird für alle folgenden LOGOFF-Verarbeitungen in der aktuellen Session deaktiviert. Ihr Name wird aus der SDF-Parameterdatei gelöscht. Wenn vorhanden, wird ab der nächsten Session die systemweite LOGOFF-Prozedur mit dem Standardnamen verwendet.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = <filename 1..54> oder \*STD:*

Die angegebene systemweite LOGOFF-Prozedur wird für alle folgenden LOGOFF-Verarbeitungen der aktuellen Session aktiviert. Ihr Name wird in der SDF-Parameterdatei gespeichert und daher auch in der nächsten Session berücksichtigt.

**SCOPE = \*NEXT-SESSION(...)**

*Typ Systemsyntaxdatei:*

Der angegebene Systemsyntaxdateiname wird ohne Überprüfung in der SDF-Parameterdatei gespeichert und in der nächsten Session berücksichtigt. Die Überprüfung erfolgt in der nächsten Session. Die laufende Session wird davon nicht berührt.

*Typ Subsystem-Syntaxdatei:*

Der angegebene Subsystem-Syntaxdateiname wird ohne Überprüfung in der SDF-Parameterdatei gespeichert und in der nächsten Session berücksichtigt. Die Überprüfung erfolgt in der nächsten Session. Die laufende Session wird davon nicht berührt. Im Falle der Deaktivierung der Subsystem-Syntaxdatei werden die Namen des Subsystems und der zugehörigen Subsystem-Syntaxdatei aus der SDF-Parameterdatei gelöscht (die laufende Session wird davon nicht berührt).

*Typ Gruppensyntaxdatei:*

Die angegebene Gruppensyntaxdatei wird der angegebenen PROFILE-ID in der SDF-Parameterdatei zugeordnet. Die Zuordnung wird erst in der nächsten Session berücksichtigt. Bereits existierende Benutzeraufträge und folgende Benutzeraufträge der gegenwärtigen Session werden davon nicht berührt. Der angegebene Syntaxdateiname wird ohne Überprüfung in der SDF-Parameterdatei gespeichert. Die Überprüfung erfolgt in der nächsten Session. Im Falle einer Deaktivierung (\*NONE) werden die bestehende PROFILE-ID und der zugehörige Gruppensyntaxdateiname aus der SDF-Parameterdatei entfernt.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = \*NO:*

Der Name der systemweiten LOGON-Prozedur wird aus der SDF-Parameterdatei gelöscht. Wenn vorhanden, wird ab der nächsten Session die systemweite LOGON-Prozedur mit dem Standardnamen verwendet. Auf die aktuelle Session hat das keinen Einfluss.

*Mit SYSTEM-LOGON-PROC bzw. SYSTEM-LOGON-INCL = <filename 1..54> oder \*STD:*

Der Name der angegebenen systemweiten LOGON-Prozedur wird in der SDF-Parameterdatei gespeichert. Die Festlegung gilt ab der nächsten Session und hat keine Auswirkung auf die aktuelle Session.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = \*NO:*

Der Name der systemweiten LOGOFF-Prozedur wird aus der SDF-Parameterdatei gelöscht. Wenn vorhanden, wird ab der nächsten Session die systemweite LOGOFF-Prozedur mit dem Standardnamen verwendet. Auf die aktuelle Session hat das keinen Einfluss.

*Mit SYSTEM-LOGOFF-PROC bzw. SYSTEM-LOGOFF-INCL = <filename 1..54> oder \*STD:*

Der Name der angegebenen systemweiten LOGOFF-Prozedur wird in der SDF-Parameterdatei gespeichert. Die Festlegung gilt ab der nächsten Session und hat keine Auswirkung auf die aktuelle Session.

**PARAMETER-FILE-NAME = \*CURRENT / <filename 1..54>**

Bestimmt den Namen der SDF-Parameterdatei, die erstellt oder aktualisiert werden soll. Voreingestellt ist \*CURRENT, d.h. die SDF-Parameterdatei der laufenden Session.



**SYNTAX-FILE-TYPE =**

Legt den Typ der Syntaxdatei fest.

**SYNTAX-FILE-TYPE = \*SYSTEM(...)**

Die Syntaxdatei ist vom Typ System.

**NAME = <filename 1..54>**

Bestimmt den Namen der Datei, die als Basis-Systemsyntaxdatei verwendet werden soll.

**SYNTAX-FILE-TYPE = \*GROUP(...)**

Die Syntaxdatei ist vom Typ Group.

**NAME =**

Bestimmt den Namen der Datei, die als Gruppensyntaxdatei verwendet werden soll.

**NAME = \*NONE**

Die Zuordnung der Gruppensyntaxdatei zur festgelegten PROFILE-ID wird gelöscht.

**NAME = <filename 1..54>**

Die angegebene Gruppensyntaxdatei wird der festgelegten PROFILE-ID zugeordnet.

**PROFILE-ID = <structured-name 1..30>**

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet war oder werden soll.

**HIERARCHY =**

Bestimmt, ob für die Syntaxanalyse der Kommandos/Anweisungen eines Benutzerauftrags mit der festgelegten PROFILE-ID die SDF-Dateihierarchie bestehen bleibt, d.h. ob die Systemsyntaxdateien zur Syntaxanalyse herangezogen werden.

**HIERARCHY = \*YES**

Die Systemsyntaxdateien werden beim Einrichten des Benutzerauftrags standardmäßig aktiviert. Die SDF-Dateihierarchie bleibt bestehen.

**HIERARCHY = \*NO**

Die Systemsyntaxdateien werden direkt nach der LOGON-Verarbeitung deaktiviert. Für Benutzer mit der festgelegten PROFILE-ID sind die Definitionen in den Systemsyntaxdateien somit ohne Bedeutung. Es gelten ausschließlich die in der zugeordneten Gruppensyntaxdatei hinterlegten Definitionen. Eine Gruppensyntaxdatei, die mit HIERARCHY=\*NO definiert ist, **muss** außer der Globalinformation auch mindestens das EXIT-JOB- bzw. LOGOFF-Kommando enthalten. Nur so ist es möglich, einen Benutzerauftrag, dem die festgelegte PROFILE-ID zugeordnet ist, zu beenden.

**SYNTAX-FILE-TYPE = \*SUBSYSTEM(...)**

Die Syntaxdatei ist vom Typ System und wird zusätzlich zu den bereits aktivierten System-syntaxdateien verwendet. Sie erscheint als Subsystem-Syntaxdatei in der Parameterdatei. Die Globalinformationen und Anwendungsbereiche in dieser Syntaxdatei werden nicht beachtet.

**NAME = \*NONE / <filename 1..54>**

Bestimmt den Namen der Datei, die als Subsystem-Syntaxdatei verwendet werden soll. Bei Angabe von \*NONE wird die zu SUBSYSTEM-NAME gehörende Subsystem-Syntaxdatei deaktiviert. \*NONE darf nur angegeben werden, wenn es sich nicht um ein DSSM-Subsystem handelt.

**SUBSYSTEM-NAME = <structured-name 1..8>**

Name des Subsystems, zu dem die Subsystem-Syntaxdatei gehört.

Das Subsystem muss nicht unbedingt ein von DSSM erzeugtes Subsystem sein.

**SYSTEM-LOGON-PROC =**

Legt fest, ob eine systemweite LOGON-Prozedur als Call-Prozedur ablaufen soll. In Abhängigkeit vom Operanden SCOPE haben die Angaben für die systemweite LOGON-Call-Prozedur unterschiedliche Auswirkungen.

**SYSTEM-LOGON-PROC = \*UNCHANGED**

Kein Wechsel der systemweiten LOGON-Call-Prozedur.

**SYSTEM-LOGON-PROC = \*NO**

Die systemweite LOGON-Call-Prozedur wird deaktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGON-PROC = \*STD**

Die systemweite LOGON-Call-Prozedur mit dem Standardnamen \$TSOS.SYS.SDF.LOGON.SYSPROC wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGON-PROC = <filename 1..54>**

Die angegebene systemweite LOGON-Call-Prozedur wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGON-INCL =**

Legt fest, ob eine systemweite LOGON-Prozedur als Include-Prozedur ablaufen soll. In Abhängigkeit vom Operanden SCOPE haben die Angaben für die systemweite LOGON-Include-Prozedur unterschiedliche Auswirkungen.

**SYSTEM-LOGON-INCL = \*UNCHANGED**

Kein Wechsel der systemweiten LOGON-Include-Prozedur.

**SYSTEM-LOGON-INCL = \*NO**

Die systemweite LOGON-Include-Prozedur wird deaktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGON-INCL = \*STD**

Die systemweite LOGON-Include-Prozedur mit dem Standardnamen \$TSOS.SYS.SDF.LOGON.SYSINCL wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGON-INCL = <filename 1..54>**

Die angegebene systemweite LOGON-Include-Prozedur wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-PROC =**

Legt fest, ob eine systemweite LOGOFF-Prozedur als Call-Prozedur ablaufen soll. In Abhängigkeit vom Operanden SCOPE haben die Angaben für die systemweite LOGOFF-Call-Prozedur unterschiedliche Auswirkungen.

**SYSTEM-LOGOFF-PROC = \*UNCHANGED**

Kein Wechsel der systemweiten LOGOFF-Call-Prozedur.

**SYSTEM-LOGOFF-PROC = \*NO**

Die systemweite LOGOFF-Call-Prozedur wird deaktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-PROC = \*STD**

Die systemweite LOGOFF-Call-Prozedur mit dem Standardnamen \$TSOS.SYS.SDF.LOGOFF.SYSPROC wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-PROC = <filename 1..54>**

Die angegebene systemweite LOGOFF-Call-Prozedur wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-INCL =**

Legt fest, ob eine systemweite LOGOFF-Prozedur als Include-Prozedur ablaufen soll. In Abhängigkeit vom Operanden SCOPE haben die Angaben für die systemweite LOGOFF-Include-Prozedur unterschiedliche Auswirkungen.

**SYSTEM-LOGOFF-INCL = \*UNCHANGED**

Kein Wechsel der systemweiten LOGOFF-Include-Prozedur.

**SYSTEM-LOGOFF-INCL = \*NO**

Die systemweite LOGOFF-Include-Prozedur wird deaktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-INCL = \*STD**

Die systemweite LOGOFF-Include-Prozedur mit dem Standardnamen \$TSOS.SYS.SDF.LOGOFF.SYSINCL wird aktiviert (siehe auch Operand SCOPE).

**SYSTEM-LOGOFF-INCL = <filename 1..54>**

Die angegebene systemweite LOGOFF-Include-Prozedur wird aktiviert (siehe auch Operand SCOPE).

## Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei beendet
2	0	CMD0677	Zugriffsrechte für die Systemsyntaxdatei wurden geändert (Warnung) garantierte Meldung: CMD0677
2	0	CMD0689	Angabe zum Operanden HIERARCHY ignoriert (Warnung) garantierte Meldung: CMD0689
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
	64	CMD0556	Kommando ohne Erfolg ausgeführt garantierte Meldungen: CMD0300, CMD0302, CMD0490, CMD0508, CMD0509, CMD0556, CMD0557, CMD0671, CMD0672, CMD0674, CMD0678; CMD0679, CMD0681, CMD0682, CMD0687, CMD0688, CMD0690, CMD0814, CMD0815
1	64	CMD0601	Kommando ist der Systembetreuung vorbehalten garantierte Meldung: CMD0601

## Hinweise

- Der Zugriff auf die SDF-Parameterdatei (Kommando MODIFY-SDF-PARAMETERS) ist nur einer Task unter der Kennung TSOS möglich. Gleichzeitige Zugriffe von weiteren Tasks werden mit einer Fehlermeldung abgewiesen.
- Die Gruppensyntaxdatei, die der PROFILE-ID der Systembetreuung (TSOS) zugeordnet ist, muss nur mehrbenutzbar sein, wenn sie auch von anderen Benutzerkennungen verwendet wird. Bei Dateischutz durch Basic-ACL, ACL oder GUARDS müssen die betreffenden Kennungen mindestens das Ausführungsrecht besitzen.
- Bei Angabe von SCOPE=\*TEMPORARY bzw. \*PERMANENT kann die TSOS-Gruppensyntaxdatei nicht deaktiviert werden.
- Bei Angabe von SCOPE=\*NEXT-SESSION werden angegebene Syntaxdateinamen und die Namen der systemweiten LOGON- bzw. LOGOFF-Prozeduren (Call- und Include-Prozedur) ohne Überprüfung in der Parameterdatei hinterlegt und erst mit Beginn der nächsten Session überprüft.
- Dateinamen, die ohne Benutzerkennung angegeben werden, erhalten die aktuelle Benutzerkennung der Task.
- Bei einer Aliasnamenersetzung werden nur die realen Dateinamen eingetragen.
- Globalinformationen aus Subsystem-Syntaxdateien werden ignoriert.

- Eine Subsystem-Syntaxdatei kann auch beim Starten des Subsystems automatisch von DSSM aktiviert werden (festgelegt in den Subsystem-Deklarationen). Ist für das Subsystem bereits eine Subsystem-Syntaxdatei in der SDF-Parameterdatei vereinbart und damit aktiviert, wird nur die für das Subsystem zuletzt aktivierte Syntaxdatei ausgewertet.  
Eine von DSSM automatisch aktivierte Syntaxdatei kann für das betreffende Subsystem mit dem Kommando MODIFY-SDF-PARAMETERS nicht entfernt, sondern nur ausgetauscht werden.  
Von DSSM automatisch aktivierte Syntaxdateien werden bei Beendigung des Subsystems von DSSM wieder deaktiviert.  
Subsystem-Syntaxdateien, die über die SDF-Parameterdatei aktiviert wurden, können nur mit einem entsprechenden MODIFY-SDF-PARAMETERS-Kommando deaktiviert werden.

## Beispiele

1. *Eine Subsystem-Syntaxdatei für die nächste BS2000-Sitzung definieren*

```
/MODIFY-SDF-PARAMETERS SYNTAX-FILE-TYPE=*SUBSYSTEM(-
/                               NAME=SYSSDF.RZ-TOOLS.010,SUBSYSTEM=RZTOOLS),-
/                               SCOPE=*NEXT-SESSION
```

Der Name der Subsystem-Syntaxdatei SYSSDF.RZ-TOOLS.010 für das RZ-eigene Subsystem RZTOOLS wird in der SDF-Parameterdatei gespeichert. Der Name der SDF-Parameterdatei ist entweder im aktuellen DSSM-Katalog definiert oder entspricht dem Standardnamen \$.SYSPAR.SDF.

Die angegebene Subsystem-Syntaxdatei wird aktiviert, wenn SDF beim nächsten BS2000-Startup geladen wird (NEXT-SESSION).

2. *Eine neue Gruppensyntaxdatei für die privilegierte Benutzerkennung TSOS definieren*

Die Benutzerkennung TSOS ist mit der PROFILE-ID SYS-TSOS vorbesetzt. Mit dem Kommando MODIFY-SDF-PARAMETERS können Sie eine neue Gruppensyntaxdatei definieren.

```
/MODIFY-SDF-PARAMETERS SYNTAX-FILE-TYPE=*GROUP(NAME=SYS.SDF.TSOS.0002,-
/                                               PROFILE-ID=SYS-TSOS),-
/                                               SCOPE=*PERMANENT
```

Die angegebene Gruppensyntaxdatei wird der PROFILE-ID SYS-TSOS zugewiesen. Diese Gruppensyntaxdatei wird beim nächsten LOGON unter der Benutzerkennung TSOS aktiviert. Die Zuweisung der Gruppensyntaxdatei ist dauerhaft (\*PERMANENT) und bleibt daher über die Dauer eines Systemlaufs hinaus erhalten. In die aktuelle SDF-Parameterdatei wird der Name der Gruppensyntaxdatei eingetragen.

### 3. *Einer Gruppe von Benutzern eine Gruppensyntaxdatei zuweisen*

Bevor Sie der Gruppe von Benutzern eine Gruppensyntaxdatei zuweisen können, müssen Sie für jede der Benutzerkennungen die gleiche PROFILE-ID definieren.

```
/MODIFY-USER-ATTRIBUTES USER-ID=A,PROFILE-ID=SDF-A-ADM
/MODIFY-USER-ATTRIBUTES USER-ID=B,PROFILE-ID=SDF-A-ADM
/MODIFY-USER-ATTRIBUTES USER-ID=C,PROFILE-ID=SDF-A-ADM
/MODIFY-SDF-PARAMETERS SYNTAX-FILE-TYPE=*GROUP(NAME=SYS.SDF-A-GROUP,-
/                                     PROFILE-ID=SDF-A-ADM),-
/                                     SCOPE=*PERMANENT
```

Die Gruppensyntaxdatei SYS.SDF-A-GROUP wird den Benutzerkennungen A, B und C über die gemeinsame PROFILE-ID SDF-A-ADM zugewiesen. In die aktuelle SDF-Parameterdatei wird der Name der Gruppensyntaxdatei und die zugehörige PROFILE-ID eingetragen.

## SHOW-SDF-OPTIONS

### Aktive Syntaxdateien und SDF-Optionen anzeigen

**Anwendungsbereich:** SDF

**Privilegierung:** STD-PROCESSING  
TSOS  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

Das Kommando SHOW-SDF-OPTIONS gibt die Namen und Versionen der momentan aktiven Syntaxdateien sowie die aktuellen Einstellungen der SDF-Optionen aus. Für Syntaxdateien wird immer der reale Dateiname angezeigt, auch wenn der Benutzer im Kommando MODIFY-SDF-OPTIONS eine Benutzersyntaxdatei über ihren im Aliaskatalog vereinbarten Aliasnamen bezeichnet hat (siehe Kommando ADD-ALIAS-CATALOG-ENTRY).

Der Operand INFORMATION steuert den Umfang der auszugebenden Informationen. Mit INFORMATION=\*USER erhalten Sie nur Informationen über Einstellungen, die Sie selbst mit MODIFY-SDF-OPTIONS tasklokal ändern können.

Das Kommando unterstützt die strukturierte Ausgabe in S-Variable (siehe [Seite 64](#) bzw. Handbuch „Kommandos, Band 6“ [4]).

<b>SHOW-SDF-OPTIONS</b>	Kurzname: <b>SHSDFO</b>
<b>INFORMATION = *ALL / *USER</b>	

#### **INFORMATION =**

Legt fest, ob alle Informationen oder nur benutzerspezifische Informationen ausgegeben werden.

#### **INFORMATION = \*ALL**

Alle Informationen über aktivierte Syntaxdateien und SDF-Einstellungen werden ausgegeben. Dazu gehören auch die benutzerspezifischen Informationen.

#### **INFORMATION = \*USER**

Nur benutzerspezifische Informationen werden ausgegeben. Dazu gehören Name und Version der aktivierten Benutzersyntaxdatei und die mit MODIFY-SDF-OPTIONS festgelegten Einstellungen.

## Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

## Ausgabe in S-Variablen

Die Ausgabe des Kommandos SHOW-SDF-OPTIONS kann in eine S-Variablen umgelenkt werden, wenn das Subsystem SDF-P geladen ist. Die S-Variablen muss als Listen-Variablen vom Typ Struktur deklariert sein (Kommando DECLARE-VARIABLE, siehe Handbuch „SDF-P“ [14]).

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Überprüfung der Benutzerprivilegien	var(*LIST).CHECK-PRIV	S	*YES *NO
statistische Erfassung von Kommandos aus der Systemsyntaxdatei	var(*LIST).CMD-STATIS	S	" (leere Zeichenkette) *NO *YES
Position des Fortsetzungszeichens bei Kommandoeingabe	var(*LIST).CONTI	S	*NEW-MODE *OLD-MODE
Funktionstastenbelegung	var(*LIST).FUNC-KEY	S	*OLD-MODE *STYLE-GUIDE-MODE
Art der Dialogführung	var(*LIST).GUIDE	S	*EXPERT *MAX *MED *MIN *NO
Aufzeichnung von Eingaben	var(*LIST).INPUT-HIST	S	*OFF *ON
Protokollierungs-Form für Eingaben	var(*LIST).LOG	S	*ACCEPT-FORM *INPUT-FORM *INVARIANT-FORM
Menüprotokollierung	var(*LIST).MENU-LOG	S	*NO *YES
Syntaxprüfung von Kommandos	var(*LIST).MODE	S	*EXEC *TEST
Größe des Eingabepuffers	var(*LIST).NUM-OF-INPUT	I	<integer 0..100>
Schutz von geheimen Operandenwerten im Eingabepuffer	var(*LIST).PASSWORD-PROT	S	*YES *NO
Syntaxfehlerdialog	var(*LIST).PROC-DIALOG	S	*NO *YES
Name der Syntaxdatei	var(*LIST).SF(*LIST).F-NAME	S	<filename 1..54>

Fortsetzung ➔



Ausgabe-Information	Name der S-Variablen	T	Inhalt
Art der Syntaxdatei	var(*LIST).SF(*LIST).TYPE	S	*GROUP *SUBSYS *SYS *USER
Version der Syntaxdatei	var(*LIST).SF(*LIST).VERSION	S	<text 1..12>
Name des Testprogrammes zur Syntaxprüfung von SDF-Anweisungen	var(*LIST).TEST-PROG-NAME	S	*NONE <structured-name 1..30>
Eingabeschnittstelle für Utilities	var(*LIST).UTILITY-INTERF	S	*NEW-MODE *OLD-MODE

### *Hinweise zum Inhalt der Variablen*

**CMD-STATIS:** Für Benutzeraufträge ohne TSOS-Privileg enthält die Variable die leere Zeichenkette " " .

**F-NAME:** Name einer Syntaxdatei mit vollständiger Pfadangabe

**TYPE:** Es werden nur Informationen zu aktivierten Syntaxdateien ausgegeben (sind z.B. keine Subsystem-Syntaxdateien aktiviert, dann wird kein Listenelement vom Typ \*SUBSYS ausgegeben). Ausnahme: Ist keine Benutzersyntaxdatei aktiviert, wird das Listenelement vom Typ \*USER trotzdem einmal angelegt (F-NAME und VERSION enthalten dabei nur die leere Zeichenkette).

Wurde INFORMATION=\*USER angegeben, dann enthält die Variable keine Informationen zu System-, Subsystem- und Gruppensyntaxdateien.

**VERSION:** Die in der Globalinformation der Syntaxdatei festgelegte Version der Syntaxdatei. Wurde dort keine Version festgelegt, enthält die Variable den Wert UNDEFINED.

Alle anderen Felder entsprechen den Einstellungen, die mit dem Kommando MODIFY-SDF-OPTIONS gesetzt werden können.

## SHOW-SDF-PARAMETERS

### SDF-Parameter anzeigen

**Anwendungsbereich:** SDF

**Privilegierung:** TSOS

Das Kommando SHOW-SDF-PARAMETERS gibt Auskunft über die Einträge einer SDF-Parameterdatei. Dabei werden die Namen der Systemsyntaxdateien sowie alle Zuordnungen von PROFILE-IDs zu einer Gruppensyntaxdatei ausgegeben. Wahlweise können die Namen der systemweiten LOGON- bzw. LOGOFF-Prozeduren (Call- und Include-Prozedur) mit ausgegeben werden.

Das Kommando unterstützt die strukturierte Ausgabe in S-Variable (siehe [Seite 69](#) bzw. Handbuch „Kommandos, Band 6“ [4]).

#### SHOW-SDF-PARAMETERS

```

SCOPE = *TEMPORARY / *NEXT-SESSION(...)
  *NEXT-SESSION(...)
    | PARAMETER-FILE-NAME = *CURRENT / <filename 1..54>
,SYNTAX-FILE-TYPE = *ALL / *SYSTEM / *GROUP(...) / *SUBSYSTEM(...) / *NONE
  *GROUP(...)
    | PROFILE-ID = *ALL / <structured-name 1..30> / <filename 1..54>
  *SUBSYSTEM(...)
    | SUBSYSTEM-NAME = *ALL / <structured-name 1..8>
,SYSTEM-LOGON-PROC = *YES / *NO
,SYSTEM-LOGON-INCL = *YES / *NO
,SYSTEM-LOGOFF-PROC = *YES / *NO
,SYSTEM-LOGOFF-INCL = *YES / *NO

```

#### **SCOPE =**

Bestimmt den Umfang der Ausgabe in Abhängigkeit von den aktivierten Syntaxdateien.

#### **SCOPE = \*TEMPORARY**

Die in der laufenden Sitzung aktivierten Syntaxdateien und/oder systemweiten LOGON- bzw. LOGOFF-Prozeduren werden ausgegeben.

**SCOPE = \*NEXT-SESSION(...)**

Die Syntaxdateien und/oder die systemweiten LOGON- bzw. LOGOFF-Prozeduren, die in einer SDF-Parameterdatei stehen, werden ausgegeben.

**PARAMETER-FILE-NAME = \*CURRENT / <filename 1..54>**

Bestimmt den Namen der SDF-Parameterdatei, aus der die anzuzeigende Information stammt.



Der angezeigte Name der SDF-Parameterdatei kann im Falle der Verwendung von Aliasnamen durch ACS als Ergebnis einer Dateinamensersetzung interpretiert werden. Nur wenn kein Aliasname für die SDF-Parameterdatei vereinbart wurde, wird der vollqualifizierte, reale Dateiname von SDF gespeichert und ausgegeben.

**PARAMETER-FILE-NAME = \*CURRENT**

Die anzuzeigende Information stammt aus der SDF-Parameterdatei, die beim Start der laufenden Session verwendet wurde.

**SYNTAX-FILE-TYPE =**

Bestimmt, für welche Syntaxdatei die Information angezeigt werden soll.

**SYNTAX-FILE-TYPE = \*ALL**

Der (Basis-)Systemsyntaxdateiname, die Namen der Subsystem-Syntaxdateien und die Gruppensyntaxdateinamen (mit der zugehörigen PROFILE-ID) werden angezeigt.

**SYNTAX-FILE-TYPE = \*SYSTEM**

Der Name der Basis-Systemsyntaxdatei wird angezeigt.

**SYNTAX-FILE-TYPE = \*GROUP(...)**

Bestimmt, dass Gruppensyntaxdateinamen angezeigt werden.

**PROFILE-ID =**

Bestimmt, welche Gruppensyntaxdateinamen angezeigt werden.

**PROFILE-ID = \*ALL**

Alle Gruppensyntaxdateien (mit zugehöriger PROFILE-ID) werden angezeigt.

**PROFILE-ID = <structured-name 1..30> / <filename 1..54>**

Bestimmt die PROFILE-ID, für die der zugehörige Gruppensyntaxdateiname angezeigt wird. Der Wert <filename> ist nur aus Kompatibilitätsgründen zugelassen und ist im geführten Dialog nicht möglich.

**SYNTAX-FILE-TYPE = \*SUBSYSTEM(...)**

Die Namen der Subsystem-Syntaxdateien werden angezeigt.

**SUBSYSTEM-NAME =**

Gibt an, zu welchen Subsystemen die Namen der Subsystem-Syntaxdateien angezeigt werden.

**SUBSYSTEM-NAME = \*ALL**

Die Namen aller Subsystem-Syntaxdateien werden angezeigt.

**SUBSYSTEM-NAME =**

Es wird nur der Name der Subsystem-Syntaxdatei angezeigt, die zum angegebenen Subsystem gehört.

**SYNTAX-FILE-TYPE = \*NONE**

Keine Syntaxdateinamen werden angezeigt.

**SYSTEM-LOGON-PROC = \*YES / \*NO**

Gibt an, ob der Name der systemweiten LOGON-Call-Prozedur (Aufruf mit dem Kommando CALL-PROCEDURE) angezeigt werden soll.

**SYSTEM-LOGON-INCL = \*YES / \*NO**

Gibt an, ob der Name der systemweiten LOGON-Include-Prozedur (Aufruf mit dem Kommando INCLUDE-PROCEDURE) angezeigt werden soll.

**SYSTEM-LOGOFF-PROC = \*YES / \*NO**

Gibt an, ob der Name der systemweiten LOGOFF-Call-Prozedur (Aufruf mit dem Kommando CALL-PROCEDURE) angezeigt werden soll.

**SYSTEM-LOGOFF-INCL = \*YES / \*NO**

Gibt an, ob der Name der systemweiten LOGOFF-Include-Prozedur (Aufruf mit dem Kommando INCLUDE-PROCEDURE) angezeigt werden soll.

**Kommando-Returncodes**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei beendet
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
1	64	CMD0601	Kommando ist der Systembetreuung vorbehalten garantierte Meldung: CMD0601
1	64	CMD0680	SDF-Parameterdatei fehlerhaft garantierte Meldungen: CMD0300, CMD0680, CMD0687

**Hinweis**

Subsystem-Syntaxdateien, die sich auf einem Pubset befinden, der während der System-einleitung nicht verfügbar war, werden nicht angezeigt. Diese Syntaxdateien können nur beim Importieren des Pubsets aktiviert werden.

## Ausgabe in S-Variablen

Die Ausgabe des Kommandos SHOW-SDF-PARAMETERS kann in eine S-Variable umgelenkt werden, wenn das Subsystem SDF-P geladen ist. Die S-Variable muss als Listen-Variable vom Typ Struktur deklariert sein (Kommando DECLARE-VARIABLE, siehe Handbuch „SDF-P“ [14]).

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Umfang der aktivierten Syntaxdateien	var(*LIST).SCOPE	S	*TEMP <filename 1..54>
Name der Syntaxdatei	var(*LIST).SF(*LIST).F-NAME	S	<filename 1..54>
Syntaxanalyse inklusive Systemsyntaxdatei	var(*LIST).SF(*LIST).HIERARCHY	S	(leere Zeichenkette) *NO *YES
Name des Subsystems	var(*LIST).SF(*LIST).NAME	S	" (leere Zeichenkette) <structured-name 1..8> BS2000
Profile-Id für die Gruppensyntaxdatei	var(*LIST).SF(*LIST).PROF-ID	S	" (leere Zeichenkette) <structured-name 1..30>
Zustand des Subsystems	var(*LIST).SF(*LIST).STATE	S	" (leere Zeichenkette) ACTIVE COEXISTENT DEACTIVATED HOLD IN-EXCHA IN-HOLD TO-IMPORT
Typ der Syntaxdatei	var(*LIST).SF(*LIST).TYPE	S	*GROUP *SUBSYS *SYS
Version der Syntaxdatei	var(*LIST).SF(*LIST).VERSION	S	<text 1..12>
Name der systemweiten LOGON-Call-Prozedur	var(*LIST).SYS-LOGON-PROC	S	*STD <filename 1..54>
Name der systemweiten LOGON-Include-Prozedur	var(*LIST).SYS-LOGON-INCL	S	*STD <filename 1..54>
Name der systemweiten LOGOFF-Call-Prozedur	var(*LIST).SYS-LOGOFF-PROC	S	*STD <filename 1..54>
Name der systemweiten LOGOFF-Include-Prozedur	var(*LIST).SYS-LOGOFF-INCL	S	*STD <filename 1..54>

### Hinweise zum Inhalt der Variablen

**SCOPE:** Bei Angabe von SCOPE=\*NEXT-SESSION(...) wird der Name der Parameterdatei als <filename 1..54> ausgegeben.

- F-NAME:** Name einer Syntaxdatei mit vollständiger Pfadangabe
- HIERARCHY:** Information entsprechend der Einstellung im Kommando MODIFY-SDF-PARAMETERS. Bei TYPE=\*SYS und TYPE=\*SUBSYS enthält das Feld die leere Zeichenkette.
- NAME:** Name des Subsystems bei TYPE=\*SUBSYS. Bei TYPE=\*SYS enthält das Feld den Wert 'BS2000'. Bei TYPE=\*GROUP enthält das Feld die leere Zeichenkette.
- PROF-ID:** Bei TYPE=\*SYS und TYPE=\*SUBSYS enthält das Feld die leere Zeichenkette.
- STATE:** Beschreibt den Status eines Subsystems, wenn es durch DSSM aktiviert wurde. Das Feld enthält in folgenden Fällen die leere Zeichenkette:
- bei TYPE=\*SYS
  - bei TYPE=\*GROUP
  - wenn im Kommando mit SCOPE=\*NEXT-SESSION(...) der Pfadname der Parameterdatei angegeben wurde.
- Für Subsysteme, die nicht durch DSSM aktiviert wurden, wird mit folgenden Ausnahmen immer der Status mit ACTIVE angezeigt:
- Konnte die Subsystem-Syntaxdatei bei der Systemeinführung nicht aktiviert werden, wird TO-IMPORT angezeigt.
  - Wurde die Subsystem-Syntaxdatei durch IMON deaktiviert, wird DEACTIVATED angezeigt.
- TYPE:** Die Ausgabe ist abhängig davon, welche Typen von Syntaxdateien mit dem Operanden SYNTAX-FILE-TYPE angefordert wurden:

Operand SYNTAX-FILE-TYPE=	Ausgabe
*ALL	Listenelemente zu allen Syntaxdatei-Typen
*SYSTEM	nur TYPE=*SYS
*GROUP(*ALL)	nur TYPE=*GROUP, für alle PROFILE-IDs
*GROUP(<name>)	nur TYPE=*GROUP, für eine bestimmte PROFILE-ID
*SUBSYSTEM(*ALL)	nur TYPE=*SUBSYS, für alle Subsysteme
*SUBSYSTEM(<name>)	nur TYPE=*SUBSYS, für ein bestimmtes Subsystem

**VERSION:** Version der Syntaxdatei

**SYS-LOGON-PROC:**

Das Listenelement erscheint nur, wenn im Kommando SYSTEM-LOGON-PROC=\*YES angegeben wurde (Default-Wert).

**SYS-LOGON-INCL:**

Das Listenelement erscheint nur, wenn im Kommando SYSTEM-LOGON-INCL=\*YES angegeben wurde (Default-Wert).

**SYS-LOGOFF-PROC:**

Das Listenelement erscheint nur, wenn im Kommando SYSTEM-LOGOFF-PROC=\*YES angegeben wurde (Default-Wert).

**SYS-LOGOFF-INCL:**

Das Listenelement erscheint nur, wenn im Kommando SYSTEM-LOGOFF-INCL=\*YES angegeben wurde (Default-Wert).

## SHOW-SYNTAX-VERSIONS

### Versionsstände von Syntaxdateien anzeigen

**Anwendungsbereich:** SDF

**Privilegierung:** STD-PROCESSING  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

Das Kommando SHOW-SYNTAX-VERSIONS gibt Informationen der momentan für die Task aktivierten System-, Subsystem- und Gruppensyntaxdateien aus. Die Ausgabe enthält die Namen und Versionsstände aller zurzeit eingesetzten Softwareprodukte und Komponenten. Am Anfang der Liste werden die Versionsstände der Syntaxdateien ausgegeben, die in die Basis-Systemsyntaxdatei integriert sind.

Sie können die Ausgabe auch für ein bestimmtes bzw. eine Liste von Softwareprodukten und Komponenten anfordern.

Das Kommando unterstützt die strukturierte Ausgabe in S-Variable (siehe [Seite 73](#) bzw. Handbuch „Kommandos, Band 6“ [4]).

<b>SHOW-SYNTAX-VERSIONS</b>
<b>SOFTWARE-UNIT-NAME</b> = <u>*ALL</u> / list-poss(2000): <structured-name 1..16>

#### **SOFTWARE-UNIT-NAME =**

Legt den Umfang der Ausgabe fest.

#### **SOFTWARE-UNIT-NAME = \*ALL / list-poss(2000): <structured-name 1..16>**

Die Versionsstände und Namen aller produktspezifischen Syntaxdateien, die in die momentan für die Task aktivierten System-, Subsystem- und Gruppensyntaxdateien integriert sind, werden ausgegeben.

#### **SOFTWARE-UNIT-NAME = list-poss(2000): <structured-name 1..16>**

Die Versionsstände der angegebenen Softwareprodukte werden ausgegeben. Ist ein angegebenes Produkt in mehreren aktivierten Syntaxdateien integriert, dann ist es auch in der ausgegebenen Liste mehrfach vorhanden.



### Kommando-Returncodes

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei ausgeführt
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft
1	64	CMD0811	Kommandoausführung nicht erfolgreich garantierte Meldungen: CMD0300, CMD0500, CMD0811

### Ausgabe in S-Variablen

Die Ausgabe des Kommandos SHOW-SYNTAX-VERSIONS kann in eine S-Variable umgelenkt werden, wenn das Subsystem SDF-P geladen ist. Die S-Variable muss als Listen-Variable vom Typ Struktur deklariert sein (Kommando DECLARE-VARIABLE, siehe Handbuch „SDF-P“ [14]).

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Pfadname der Syntaxdatei	var(*LIST).F-NAME	S	<filename 1..54>
Name der Softwarekomponente	var(*LIST).SW-UNIT(*LIST). COMPONENT(*LIST).NAME	S	<structured-name 1..13>
Version der Softwarekomponente	var(*LIST).SW-UNIT(*LIST). COMPONENT(*LIST).VERSION	S	<text 3..3> / <text 8..8>
Name der Softwareeinheit	var(*LIST).SW-UNIT(*LIST).NAME	S	<structured-name 1..16>
Version der Softwareeinheit	var(*LIST).SW-UNIT(*LIST).VERSION	S	<text 7..8>
Art der Syntaxdatei	var(*LIST).TYPE	S	*GROUP *SYS

#### *Hinweise zum Inhalt der Variablen*

VERSION: Version einer Softwareeinheits- oder Komponenten-Syntaxdatei

TYPE: Zwischen System- und Subsystem-Syntaxdateien wird nicht unterschieden. Beide werden mit TYPE=\*SYS ausgegeben.



---

## 5 SDF-I

Das Dienstprogramm SDF-I mischt Gruppen- oder Systemsyntaxdateien und führt Konsistenzprüfungen durch. Dabei können von Fujitsu Siemens Computers ausgelieferte und/oder vom Benutzer selbst (mit SDF-A) erstellte Syntaxdateien gemischt werden. Die mit SDF-I eingemischten Syntaxdateien können mit SDF-I wieder entmischt werden. SDF-I unterstützt Syntaxdateien mit oder ohne PAM-Key.



Die Installation erfolgt im Normalfall über den Installationsmonitor IMON. Da IMON die SESDs (Softwareeinheitssyntaxdateien) ab BS2000/OSD V2.0 nicht mehr mischt, sondern als Basis-Systemsyntaxdatei und Subsystem-Syntaxdateien aktiviert, findet SDF-I nur noch in Sonderfällen Verwendung (z.B. Mischen von kundeneigenen Subsystem- oder Gruppensyntaxdateien, Anzeigen von Syntaxdatei-Informationen).

### 5.1 Ein-/Ausgaben von SDF-I

SDF-I mischt eine oder mehrere Syntaxdateien mit einer Eingabesyntaxdatei in eine Ausgabesyntaxdatei. Dabei müssen die Eingabedatei und die zu mischenden Syntaxdateien vom gleichen Typ sein (System- oder Gruppensyntaxdateien). Die Ausgabesyntaxdatei erhält den gleichen Typ wie die Eingabesyntaxdatei.

Folgende Dateien können Eingabedateien sein:

- SESDs (Softwareeinheitssyntaxdateien), die von Fujitsu Siemens Computers ausgeliefert werden (z.B. die SESD für den Grundausbau des BS2000).
- die aktuelle INSD (Installationssyntaxdatei), die gerade vom System verwendet wird.

Folgende Dateien können in die Eingabesyntaxdatei eingemischt werden:

- SESDs.
- jede andere, vom Benutzer mit SDF-A erstellte Syntaxdatei (siehe Handbuch „SDF-A“ [7]).

Mit SDF V2.0 wurde ein neues Syntaxdatei-Format eingeführt. Ältere Syntaxdateien beliebigen Typs (Benutzer, Gruppe, System) können mit der Anweisung CONVERT-SYNTAX-FILE in das neue Format konvertiert werden. CONVERT-SYNTAX-FILE kann unabhängig von einer OPEN-Anweisung ausgeführt werden.



**Achtung!**

Es ist nicht möglich, eine Konvertierung vom alten in das neue Format wieder rückgängig zu machen. Bewahren Sie deshalb die alte Syntaxdatei auf, um im Problemfall darauf zugreifen zu können. Sie sollten auch eine Prozedur mit den SDF-A-Anweisungen haben, mit der die Syntaxdatei wieder erstellt werden kann.

Weitere Informationen dazu finden Sie im [Abschnitt „Hinweise“ ab Seite 79](#).

## 5.2 Aufruf und Ablauf von SDF-I

SDF-I liest seine Anweisungen nicht über SDF und benötigt deshalb zu seinem Ablauf keine aktivierten Syntaxdateien. Es gibt daher für SDF-I-Anweisungen auch keinen geführten Dialog. In der Syntaxdatei SYSSDF.SDF-I.041, die zusammen mit SDF ausgeliefert wird, ist lediglich das Kommando START-SDF-I definiert.

Das Dienstprogramm SDF-I kann unter jeder Benutzerkennung mit folgendem Kommando gestartet werden:

<b>START-SDF-I</b>	Kurzname: <b>SDF-I</b>
<p><b>VERSION</b> = <u>*STD</u> / &lt;product-version&gt;</p> <p><b>,MONJV</b> = <u>*NONE</u> / &lt;filename 1..54 without-gen-vers&gt;</p> <p><b>,CPU-LIMIT</b> = <u>*JOB-REST</u> / &lt;integer 1..32767 seconds&gt;</p>	

**VERSION =**

Sind mehrere Versionen von SDF-I mit IMON installiert, so kann der Benutzer die Version auswählen, mit der er arbeiten möchte. Bei Angabe der Versionsbezeichnung mit Hochkommas kann der Buchstabe C vorangestellt werden (C-STRING-Syntax).

Falls das Produkt nicht mit IMON installiert wurde oder die angegebene Version nicht existiert, gilt VERSION=\*STD.

**VERSION = \*STD**

Aufruf der SDF-I-Version mit der höchsten Versionsnummer.

**VERSION = <product-version>**

Angabe der SDF-I-Version im Format mm.n[a[so]] (vgl. „[product-version](#)“ auf Seite 14).

**MONJV =**

Angabe einer Monitor-Jobvariablen zur Überwachung des SDF-I-Laufs.

**MONJV = \*NONE**

Es wird keine Monitor-Jobvariable verwendet.

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

Name der zu verwendenden Jobvariablen.

**CPU-LIMIT =**

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf.

**CPU-LIMIT = \*JOB-REST**

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

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

Es soll nur die angegebene Zeit verwendet werden.

SDF-I kann im Dialog oder auch als Stapelauftrag laufen. Wenn mehrere Syntaxdateien gemischt werden sollen, ist es sinnvoll, SDF-I als Stapelauftrag laufen zu lassen.

Die Anweisungen CONVERT-SYNTAX-FILE und SHOW-SYNTAX-FILE können unabhängig vom Mischvorgang auch ohne vorhergehende OPEN-Anweisung eingegeben werden.

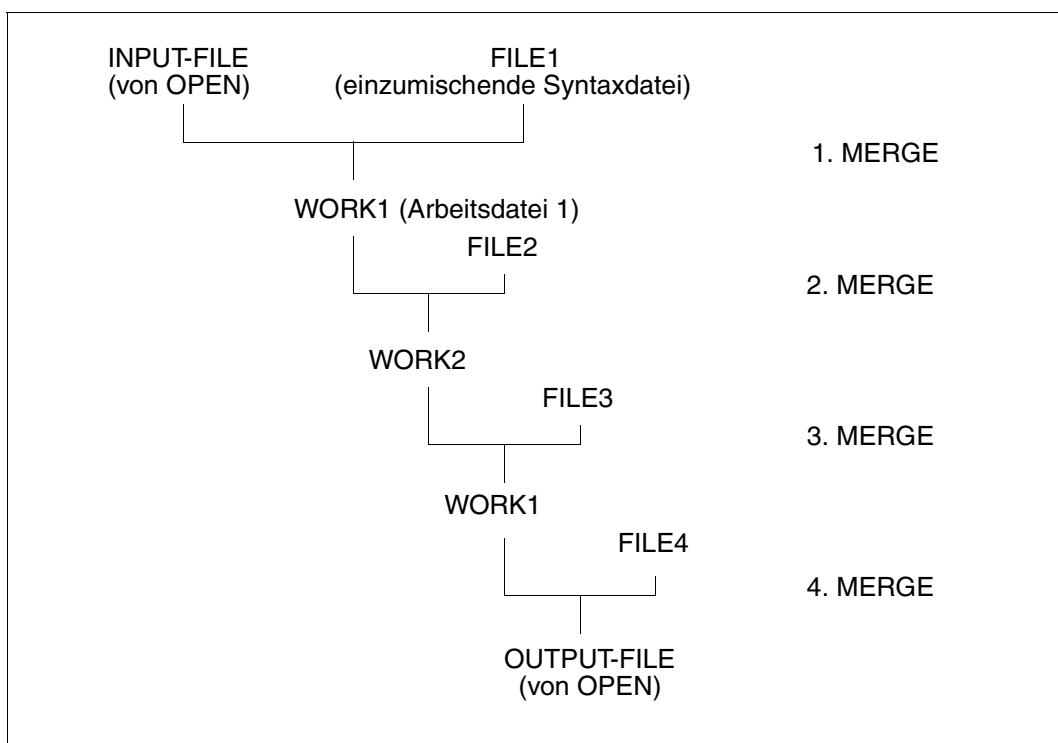
Vor jedem Mischvorgang müssen zuerst mit der OPEN-Anweisung die Eingabesyntaxdatei und die Ausgabesyntaxdatei zugewiesen werden. Anschließend werden mit MERGE-Anweisungen die Syntaxdateien angegeben, die dazugemischt werden sollen. Für jede Syntaxdatei, die dazugemischt werden soll, muss eine MERGE-Anweisung gegeben werden.

Nach jeder MERGE-Anweisung wird eine Arbeitsdatei erzeugt. Diese Arbeitsdatei enthält alle bis dahin eingemischten Syntaxdateien einschließlich der Eingabesyntaxdatei. Mit der END-Anweisung oder der OPEN-Anweisung für die nächste Syntaxdatei wird die Arbeitsdatei in die Ausgabedatei umgewandelt. Mit der END-Anweisung wird SDF-I beendet.

Mit der Anweisung REMOVE können zuvor mit SDF-I eingemischte Syntaxdateien wieder entmischt werden. Der REMOVE-Anweisung muss wie bei MERGE eine OPEN-Anweisung vorangehen, in der Eingabe- und Ausgabesyntaxdatei festgelegt werden.

## 5.3 Behandlung von Arbeitsdateien

SDF-I benutzt maximal zwei Arbeitsdateien, die von SDF-I angelegt und wieder gelöscht werden. Sie werden unter dem Namen SYSSDF.SDF-I.version.tsn.n (n=1 oder 2, version: SDF-I-Version, z.B. 041 bei SDF-I V4.1) angelegt. Bis SDF-I V3.0 wurde dafür der Name SDF-I.MERGE.WORK.tsn.n verwendet. Der Benutzer sollte darauf achten, dass eigene Dateinamen nicht mit diesen Namensteilen beginnen. Beim Einmischen von mehr als einer Syntaxdatei in die Eingabesyntaxdatei (Operand INPUT-FILE in der OPEN-Anweisung) werden zwei Arbeitsdateien im Wechsel verwendet. Sie sehen im Bild, wie die beiden Arbeitsdateien im Mischvorgang verwendet werden.



Verwendung von zwei Arbeitsdateien beim Mischen

Die Arbeitsdateien erreichen die Größe der vollständig gemischten Datei. Aus diesem Grund muss bei Ablauf von SDF-I eine entsprechende Reserve an Plattenspeicherplatz zur Verfügung stehen. Steht nicht genug Speicherplatz zur Verfügung, so wird die Bearbeitung der MERGE-Anweisung abgebrochen.

## 5.4 SDF-Standardanweisungen und Anwendungsbereiche

Die SDF-Standardanweisungen sind ab SDF V4.1 in der Syntaxdatei von SDF enthalten. Für SDF wird eine eigene SESD ausgeliefert.

SDF-I mischt die Standardanweisungen ein, wenn die Syntaxdatei von SDF einzumischen ist. Dann stehen automatisch allen Programmen in der Ausgabesyntaxdatei die Standardanweisungen zur Verfügung. Beim Einmischen einer beliebigen SESD in eine Syntaxdatei, in der die SDF-Syntaxdatei schon vorhanden ist, erhalten die Programme der neu dazugekommenen Syntaxdatei ebenfalls Zugriff auf die Standardanweisungen.

Ab SDF-I V4.0 werden alle Anwendungsbereiche aus den Eingabesyntaxdateien in die Liste der Anwendungsbereiche der Ausgabesyntaxdatei eingemischt. Anwendungsbereiche, zu denen keine Kommandos gehören, werden von SDF-I nicht entfernt. Sie können nicht benötigte Anwendungsbereiche mit der SDF-A/SDF-U-Anweisung REMOVE entfernen.

## 5.5 Hinweise

### zu Eingaben:

- Die Anweisungen und ihre Operanden können bis zur Eindeutigkeit abgekürzt werden. Von dieser Möglichkeit sollte nur im Dialog Gebrauch gemacht werden.
- Fortsetzungszeilen sind möglich. Die maximale Länge einer Anweisung beträgt 2044 Zeichen.
- Für den Mischvorgang ist folgende Reihenfolge der Anweisungen einzuhalten (es können mehrere MERGE-Anweisungen aufeinander folgen):

```
OPEN
MERGE bzw. REMOVE
END
```

### zur Kompatibilität der Syntaxdatei-Formate:

- Syntaxdateien haben ab SDF V2.0 ein neues Format. SDF-I verarbeitet dieses neue Format ab Version 2.0. Ältere Syntaxdateien müssen in das neue Format konvertiert werden, wenn sie in SDF  $\geq$  V2.0 eingesetzt werden sollen. Mit SDF-I können aber auch Ausgabesyntaxdateien mit alter Struktur (für SDF < V2.0A) erstellt werden.
- Syntaxdateien, die mit SDF-I V4.1 und V4.1-Format generiert wurden, können nicht mehr mit SDF-I-Versionen < 4.1 verarbeitet werden. Sie können jedoch problemlos in SDF ab Version 3.0 eingesetzt werden.

- Beim Mischen hängt das Format der Ausgabesyntaxdatei vom Format der Eingabesyntaxdateien ab. SDF-I erzeugt die Ausgabesyntaxdatei mit dem höchsten Format, das unter den Eingabesyntaxdateien vorkommt. Wenn z.B. nur Syntaxdateien mit V2- und V3-Format eingemischt werden, erhält die Ausgabesyntaxdatei das Format V3. Diese Syntaxdatei kann dann auch mit SDF-I V3.0 bearbeitet werden.
- Ab SDF-I V3.0 wird als Name des Produktes SDF-A „SDF-A“ in die Ausgabesyntaxdatei eingetragen. In älteren Syntaxdatei-Versionen (für  $SDF \leq V2.0$ ) war der Name von SDF-A noch mit „SDFA“ festgelegt. Wird eine solche Syntaxdatei mit SDF-I  $\geq V3.0$  bearbeitet, sollten Sie sich mit der Anweisung SHOW-SYNTAX-FILE den tatsächlich eingetragenen Namen anzeigen lassen und nur diesen Namen in den SDF-I-Anweisungen verwenden.
- Während einer OPEN/MERGE-Sequenz werden Produkte, die schon in der Syntaxdatei enthalten sind, beim Mischen durch die neuen Versionen ersetzt. Eine Ausnahme bildet das Produkt SDF < V4.0.



## 5.6 SDF-I-Anweisungen

### 5.6.1 Funktionelle Übersicht

#### Anzeigefunktion

SHOW-SYNTAX-FILE                      Gibt Informationen über Gruppen- und Systemsyntaxdateien aus

#### Syntaxdateien konvertieren

CONVERT-SYNTAX-FILE                  Wandelt eine beliebige Syntaxdatei (System, Gruppe, Benutzer) mit alter Struktur in eine Syntaxdatei mit neuer Struktur um

#### Syntaxdateien mischen und entmischen

MERGE                      Legt die Syntaxdatei fest, die in die Eingabesyntaxdatei gemischt werden soll und prüft eine SESD ggf. auf Konsistenz mit einem vorgegebenen Produktnamen oder einer Produktversion

OPEN                      Legt Folgendes fest:  
– die Eingabesyntaxdatei, in die eine Syntaxdatei eingemischt oder aus der eine Syntaxdatei entmischt werden soll  
– die Ausgabesyntaxdatei, in der das Ergebnis des Mischens/Entmischens stehen soll

REMOVE                    entfernt zuvor mit SDF-I eingemischte Syntaxdateien aus der Eingabesyntaxdatei

#### SDF-I beenden

END                      beendet den SDF-I-Lauf

## 5.6.2 Beschreibung der Anweisungen

### CONVERT-SYNTAX-FILE Syntaxdatei mit neuem Format versehen

Die Anweisung CONVERT-SYNTAX-FILE ermöglicht die Umwandlung einer beliebigen Syntaxdatei (Gruppe, System, Benutzer) mit altem Format (SDF < V2.0A) in eine Syntaxdatei mit neuem Format, das mit SDF ab V4.1 kompatibel ist.

<b>CONVERT-SYNTAX-FILE</b>
<b>INPUT-FILE</b> = <filename 1..54> , <b>OUTPUT-FILE</b> = <filename 1..54>

#### **INPUT-FILE = <filename 1..54>**

Bestimmt die Syntaxdatei mit alter Struktur, die umgewandelt werden soll.

#### **OUTPUT-FILE = <filename 1..54>**

Bestimmt den Namen der Ausgabesyntaxdatei mit der neuen Struktur.

#### **Hinweise**

- Es ist nicht möglich, eine Konvertierung vom alten in das neue Format wieder rückgängig zu machen. Bewahren Sie deshalb die alte Syntaxdatei auf, um im Problemfall darauf zugreifen zu können. Sie sollten auch eine Prozedur mit den SDF-A-Anweisungen haben, mit der die Syntaxdatei wieder erstellt werden kann.
- Mit SDF-I V4.1 konvertierte Syntaxdateien dürfen nicht mit SDF-I-Versionen < 4.1 verarbeitet werden. Sie können jedoch problemlos in SDF ab Version 3.0 eingesetzt werden.

**END**  
**SDF-I beenden**

Die Anweisung beendet den SDF-I-Lauf und erzeugt die Ausgabedatei, deren Name im Operanden OUTPUT-FILE der Anweisung OPEN festgelegt wurde.

END

Diese Anweisung hat keine Operanden.

## MERGE

### Syntaxdateien mischen

Mischt die Eingabesyntaxdatei mit einer anderen, hier angegebenen Syntaxdatei. Die hier angegebene Syntaxdatei muss vom gleichen Typ (System- oder Gruppensyntaxdatei) sein wie die Eingabesyntaxdatei.

Beim Mischen werden Programme mit den dazugehörigen Anweisungen als Einheit übertragen, ersetzen also auch ein ganzes Programm, das in der Eingabedatei schon vorhanden ist. Es ist nicht möglich, mehrere Versionen eines Produktes einzumischen.

**Format 1:** Einmischen von Softwareeinheitssyntaxdateien

<b>MERGE</b>
<b>FILE</b> = <filename 1..54> <b>,CHECK-PRODUCT</b> = <u>*NO</u> / <structured-name 1..15> <b>,CHECK-VERSION</b> = <u>*NO</u> / <c-string 1..7> <b>,REPLACE-PRODUCT</b> = <u>YES</u> / <u>NO</u> <b>,UPDATE-SDF-GLOBALS</b> = <u>YES</u> / <u>NO</u>

#### **FILE = <filename 1..54>**

Gibt die Syntaxdatei an, die mit der unter OPEN angegebenen Eingabedatei gemischt werden soll. Wurden nach dem OPEN schon MERGE-Anweisungen ausgeführt, so wird die Syntaxdatei in die gerade verwendete Arbeitsdatei eingemischt, in der die vorher mit MERGE eingemischten Syntaxdateien schon vorhanden sind (siehe [Seite 78](#)).

#### **CHECK-PRODUCT =**

Bestimmt, ob die einzumischende Syntaxdatei auf Namenskonsistenz mit einem vorgegebenen Produktnamen geprüft wird. Enthält die angegebene Syntaxdatei mehr als eine Softwareeinheit, wird die Meldung SDI0023 ausgegeben. Der Mischlauf wird jedoch trotzdem durchgeführt. Es ist dann aber nach dem Mischen nicht mehr möglich, eine oder alle der in dieser Syntaxdatei enthaltenen SESDs mit REMOVE zu entfernen.

#### **CHECK-PRODUCT = \*NO**

Eine Prüfung auf Namenskonsistenz findet nicht statt.

#### **CHECK-PRODUCT = <structured-name 1..15>**

Gibt den Produktnamen der Softwareeinheit an, die in der angegebenen Syntaxdatei enthalten ist (z.B. SPOOL, TIAM, etc.).

**CHECK-VERSION =**

Bestimmt, ob die einzumischende Syntaxdatei auf Versionskonsistenz mit einer vorgegebenen Versionsnummer geprüft wird.

**CHECK-VERSION = \*NO**

Eine Prüfung auf Versionskonsistenz findet nicht statt.

**CHECK-VERSION = <c-string 1..7>**

Versionsnummer einer Softwareeinheit. Die Versionsnummern der von Fujitsu Siemens Computers gelieferten SEDs entsprechen dem SDF-Datentyp <product-version> (siehe „[product-version](#)“ auf Seite 14).

Format: 'nn.naxx'.

nn.na = Versionsnummer der Softwareeinheit

xx = Änderungsstand der Syntaxdatei

Es ist möglich, nur den linken Teil der Versionsnummer (nn.n) anzugeben. Die Überprüfung erfolgt dann in der angegebenen Länge.

**REPLACE-PRODUCT =**

Gibt an, ob bereits vorhandene Objekte in der Eingabesyntaxdatei von Objekten mit gleichem internen oder externen Namen aus der hier angegebenen Syntaxdatei überschrieben werden sollen.

**REPLACE-PRODUCT = YES**

Objekte der Eingabesyntaxdatei werden überschrieben von Objekten mit gleichem internen oder externen Namen. Dies ermöglicht eine Umbenennung von internen oder externen Namen.

**REPLACE-PRODUCT = NO**

Bei Objekten mit gleichem internen oder externen Namen wird der Mischlauf abgebrochen.

**UPDATE-SDF-GLOBALS =**

Gibt an, ob die Globalinformation der Eingabesyntaxdatei überschrieben werden soll (Update).

**UPDATE-SDF-GLOBALS = YES**

Die Globalinformation der Eingabesyntaxdatei wird mit der Globalinformation der hier angegebenen Syntaxdatei überschrieben. Die Versionsnummer der Eingabesyntaxdatei wird dabei nicht überschrieben.

**UPDATE-SDF-GLOBALS = NO**

Die Globalinformation der Eingabesyntaxdatei wird nicht überschrieben.

**Format 2:** Einmischen von selbsterstellten Syntaxdateien

<b>MERGE</b>
<b>FILE</b> = <filename 1..54> <b>,REMOVE-ID</b> = <structured-name 1..15> <b>,REPLACE-PRODUCT</b> = <u>YES</u> / NO

**FILE = <filename 1..54>**

Gibt die Syntaxdatei an, die mit der unter OPEN angegebenen Eingabedatei gemischt werden soll. Wurden nach dem OPEN schon MERGE-Anweisungen ausgeführt, so wird die Syntaxdatei in die gerade verwendete Arbeitsdatei eingemischt, in der die vorher mit MERGE eingemischten Syntaxdateien schon vorhanden sind (siehe [Seite 78](#)).

**REMOVE-ID = <structured-name 1..15>**

Kurzname der Syntaxdatei (notwendig für späteres Löschen).

**REPLACE-PRODUCT =**

Gibt an, ob bereits vorhandene Objekte in der Eingabesyntaxdatei von Objekten mit gleichem internen oder externen Namen aus der hier angegebenen Syntaxdatei überschrieben werden sollen.

**REPLACE-PRODUCT = YES**

Objekte der Eingabesyntaxdatei werden überschrieben von Objekten mit gleichem internen oder externen Namen. Dies ermöglicht eine Umbenennung von internen oder externen Namen.

**REPLACE-PRODUCT = NO**

Bei Objekten mit gleichem internen oder externen Namen wird der Mischlauf abgebrochen.

**Hinweise**

- Tritt bei der Behandlung von MERGE ein Fehler auf, so wird die Bearbeitung mit der nächsten MERGE-Anweisung fortgesetzt. Dabei setzt SDF-I auf dem Stand des letzten erfolgreichen MERGE auf. Nach Programmende wird der Spin-off-Mechanismus ausgelöst. Können auch nachfolgende MERGE-Anweisungen nicht ausgeführt werden, so enthält die Ausgabedatei den Stand des zuletzt erfolgreichen MERGE. Die Ausgabedatei enthält in diesem Fall nicht alle gewünschten Produkte, ist aber eine gültige Syntaxdatei.
- Erscheint nach einer MERGE-Anweisung für eine von Fujitsu Siemens Computers gelieferte Syntaxdatei die Meldung SDI0035, dann muss diese ausgelieferte Syntaxdatei nicht wie eine SESD (siehe Format 1), sondern wie eine selbsterstellte Syntaxdatei behandelt werden (siehe Format 2).

- Wenn mehrere Syntaxdateien für mehrere Produkte eingemischt werden müssen, ist es besser, die System- und Gruppensyntaxdateien aus den Original-Syntaxdateien mit Hilfe eines Batch-Auftrages neu zu generieren. Der Nachteil des erhöhten Speicherbedarfs wird durch eine höhere Sicherheit der generierten Syntaxdateien ausgeglichen.

## OPEN

### Eingabe- und Ausgabesyntaxdatei zuweisen

Die Anweisung OPEN weist die Eingabesyntaxdatei und die Ausgabesyntaxdatei zu. Die Eingabesyntaxdatei wird mit den neuen Syntaxdateien gemischt (MERGE-Anweisung). Das Ergebnis dieses Vorgangs legt SDF-I in der Ausgabesyntaxdatei ab. Die Eingabesyntaxdatei wird von SDF-I nicht verändert. Die Angabe der Eingabe- und Ausgabesyntaxdatei ist zwingend sowohl beim Mischen als auch beim Entmischen.

Eingabesyntaxdatei kann nur eine System- oder Gruppensyntaxdatei sein. Die Ausgabesyntaxdatei erhält automatisch denselben Typ wie die Eingabesyntaxdatei.

Für die Ausgabedatei kann wahlweise eine Versionsnummer angegeben werden. Wenn die Ausgabedatei mit SDF (z.B. mit MODIFY-SDF-OPTIONS) aktiviert wurde, kann mit SHOW-SDF-OPTIONS diese Versionsnummer ausgegeben werden.

Außerdem kann angegeben werden, ob die Ausgabedatei die alte Syntaxdateistruktur (für SDF < V2.0A) oder die neue Syntaxdateistruktur (für SDF ≥ V2.0A) haben soll.

OPEN
<pre> <b>INPUT-FILE</b> = &lt;filename 1..54&gt; , <b>OUTPUT-FILE</b> = &lt;filename 1..54&gt; , <b>VERSION</b> = 'UNDEFINED' / &lt;c-string 1..12&gt; / &lt;alphanum-name 1..12&gt; , <b>FORMAT</b> = *STD / *OLD / *NEW , <b>BLKCTRL</b> = *STD / *PAMKEY / *DATA , <b>WRITE-MODE</b> = *NEW / *REPLACE </pre>

#### **INPUT-FILE = <filename 1..54>**

Name der Syntaxdatei, die als Eingabedatei benutzt werden soll. Diese Datei wird mit der bei der Anweisung MERGE angegebenen Datei gemischt. Die logische Struktur von Eingabesyntaxdateien wird von SDF-I nicht überprüft. Es wird angenommen, dass diese mit SDF-A korrekt aufgebaut wurden. In der Regel ist die Eingabedatei entweder eine von Fujitsu Siemens Computers ausgelieferte oder bereits in einem früheren SDF-I-Lauf erzeugte INSD. Steht eine solche nicht zur Verfügung, so sollte die Softwareeinheitssyntaxdatei von SDF als Eingabedatei angegeben werden.

#### **OUTPUT-FILE = <filename 1..54>**

Name der Ausgabedatei nach Ausführung aller MERGE- und REMOVE-Anweisungen. Sie erhält denselben Typ (System- oder Gruppensyntaxdatei) wie die Eingabedatei. Auf einer fremden Benutzerkennung kann eine OUTPUT-Datei nur von der Systemverwaltung angelegt werden.



**VERSION = 'UNDEFINED' / <c-string 1..12> / <alphanumeric-name 1..12>**

Versionsnummer, die in der Globalinformation der Ausgabedatei hinterlegt und nach Aktivierung dieser Syntaxdatei beim Kommando SHOW-SDF-OPTIONS angezeigt wird. Wenn kein besonderer Wert angegeben wird, wird als Wert 'UNDEFINED' in der Globalinformation abgelegt.

**FORMAT =**

Bestimmt, welches Format die Ausgabesyntaxdatei erhalten soll. Die Anweisung SHOW-SYNTAX-FILE (siehe [Seite 93](#)) gibt als Formatangabe einer Syntaxdatei V1, V2, V3, V4 oder V4.1 aus. Für Syntaxdateien mit V1-Format muss FORMAT=\*OLD angegeben werden. Die Formate V2 bis V4.1 entsprechen FORMAT=\*NEW.

**FORMAT = \*STD**

Die Ausgabesyntaxdatei wird mit der alten Syntaxdateistruktur (SDF < V2.0A) erzeugt, wenn alle Syntaxdateien, die für diese Ausgabesyntaxdatei verarbeitet werden, die alte Syntaxdateistruktur haben. Falls auch Syntaxdateien mit neuer Struktur dabei sind, wirkt \*STD wie FORMAT=\*NEW.

**FORMAT = \*OLD**

Bestimmt, dass die Ausgabesyntaxdatei mit der alten Syntaxdateistruktur erzeugt wird (SDF < V2.0A). Alle Syntaxdateien, die für diese Ausgabesyntaxdatei verarbeitet werden, müssen dann die alte Syntaxdateistruktur haben. Syntaxdateien, die Elemente mit neuer Struktur beinhalten, werden zurückgewiesen.

**FORMAT = \*NEW**

Bestimmt, dass die Ausgabesyntaxdatei mit der neuen Syntaxdateistruktur erzeugt wird (SDF ≥ V2.0A). Für die Erstellung der Ausgabesyntaxdatei mit neuer Syntaxdateistruktur können Syntaxdateien alter und neuer Struktur beliebig verwendet werden. FORMAT=\*NEW wirkt nur, wenn anschließend mindestens eine MERGE-Anweisung ausgeführt wird. Eine nachfolgende REMOVE-Anweisung konvertiert nicht die gesamte Syntaxdatei, sondern nur die von REMOVE betroffenen Dateiinhalte (z.B. Globalinformationen, Kommandolisten, Programmlisten, ...).

**BLKCTRL =**

Bestimmt, ob die Ausgabesyntaxdatei und die SDF-I-Arbeitsdateien mit oder ohne PAM-Key erzeugt werden.

**BLKCTRL = \*STD**

Die Block-Control-Einstellung, die für die aktuelle Task gilt, wird übernommen.

**BLKCTRL = \*DATA**

Die Ausgabesyntaxdatei und die SDF-I-Arbeitsdateien werden ohne PAM-Key erzeugt.

**BLKCTRL = \*PAMKEY**

Die Ausgabesyntaxdatei und die SDF-I-Arbeitsdateien werden mit PAM-Key erzeugt. Die Dateien können dann nur auf Platten gespeichert werden, die PAM-Key unterstützen.

**WRITE-MODE =**

Legt fest, ob die Ausgabesyntaxdatei schon existiert oder nicht.

SDF-I arbeitet schneller, wenn es bereits eine Ausgabedatei mit annähernd der benötigten Größe gibt. Z.B. kann der Benutzer mit dem Kommando CREATE-FILE vor dem SDF-I-Lauf eine Datei mit den notwendigen Datei-Attributen (primary/secondary allocation) erzeugen.

**WRITE-MODE = \*NEW**

Die Ausgabesyntaxdatei darf noch nicht existieren, andernfalls wird eine Fehlermeldung ausgegeben. SDF-I berechnet die Größe der Ausgabesyntaxdatei aus der Größe der Eingabesyntaxdateien.

**WRITE-MODE = \*REPLACE**

Die Ausgabesyntaxdatei muss bereits existieren, andernfalls wird eine Fehlermeldung ausgegeben. SDF-I verwendet die Datei-Attribute der katalogisierten Datei (primary/secondary allocation) als Attribute für die Ausgabesyntaxdatei und die SDF-I-Arbeitsdateien. Die bereits existierende Ausgabedatei wird überschrieben.

**Hinweis**

Tritt bei der Behandlung von OPEN ein Fehler auf, so werden nachfolgende MERGE-Anweisungen bis zu einer neuen OPEN-Anweisung abgewiesen.

## REMOVE

### Syntaxdateien entmischen

Eine Syntaxdatei, die zuvor mit SDF-I eingemischt wurde, kann mit REMOVE wieder entmischt werden.

**Format 1:** Entmischen von Softwareeinheitssyntaxdateien

<b>REMOVE</b>
<b>PRODUCT-NAME</b> = <structured-name 1..15>

#### **PRODUCT-NAME = <structured-name 1..15>**

Bestimmt den Namen des Produkts, dessen Syntaxdatei aus der Eingabedatei entfernt werden soll, die in der vorhergehenden OPEN-Anweisung festgelegt wurde. Dieser Operand darf nur für Produkte von Fujitsu Siemens Computers verwendet werden. Er stimmt mit dem Namen des Produkts überein (SESD-Name), der mit der Anweisung SHOW-SYNTAX-FILE ausgegeben wird.

**Format 2:** Entmischen selbsterstellter Syntaxdateien

<b>REMOVE</b>
<b>REMOVE-ID</b> = <structured-name 1..15>

#### **REMOVE-ID = <structured-name 1..15>**

Bestimmt den Namen des Produkts, dessen Syntaxdatei aus der Eingabedatei entfernt werden soll, die in der vorhergehenden OPEN-Anweisung festgelegt wurde. Dieser Operand darf nur für vom Benutzer selbst erstellte Syntaxdateien verwendet werden. Er stimmt mit der REMOVE-ID überein, die mit MERGE eingeführt wurde.

## Beispiel

*Softwareeinheits-Syntaxdatei entmischen*

Die Eingabesyntaxdatei *SF.base* enthält folgende SESDs: L1 Version 1, L2 Version 1, L3 Version 1, L4 Version 1

Die einzumischende Syntaxdatei *SF.in* enthält folgende SESD: L2 Version 2

```
OPEN INPUT-FILE=SF.base,OUTPUT-FILE=SF.out _____ (1)
REMOVE PRODUCT-NAME=L2 _____ (2)
MERGE FILE=SF.in _____ (3)
END _____ (4)
```

- (1) Die Eingabesyntaxdatei *SF.base* und der Name der Ausgabesyntaxdatei *SF.out* werden festgelegt.
- (2) Das Produkt L2 wird aus der Eingabesyntaxdatei *SF.base* entfernt, da im nächsten Schritt die Syntaxdatei *SF.in* eingemischt wird, die eine neue Version des Produkts L2 enthält. So wird sichergestellt, dass in der Syntaxdatei keine Reste der Version 1 verbleiben (wenn in der Version 2 z.B. ein Kommando entfallen ist, würde dieses durch bloßes Einmischen der Version 2 nicht entfernt werden).
- (3) Die Syntaxdatei *SF.in* wird mit der Eingabesyntaxdatei *SF.base* in die interne SDF-I-Arbeitsdatei (siehe [Seite 78](#)) gemischt.
- (4) SDF-I wird beendet, die Arbeitsdatei erhält den mit OPEN festgelegten Namen *SF.out* und enthält folgende SESDs: L1 Version 1, L2 Version 2, L3 Version 1, L4 Version 1. Die SESD L2 ist nun in der neuen Version vorhanden.

## Hinweise

- Es können nur die Syntaxdateien entmischt werden, die vorher mit der SDF-I-Anweisung MERGE eingemischt wurden.
- Objekte, die mit MERGE ... REPLACE-PRODUCT=YES überschrieben wurden, werden nicht wiederhergestellt.
- Wenn mehrere Syntaxdateien für mehrere Produkte entmischt werden müssen, ist es besser, die System- und Gruppensyntaxdateien aus den Original-Syntaxdateien mit Hilfe eines Batch-Auftrages neu zu generieren. Der Nachteil des erhöhten Speicherbedarfs wird durch eine höhere Sicherheit der generierten Syntaxdateien ausgeglichen.

## SHOW-SYNTAX-FILE

### Informationen über eine Syntaxdatei anzeigen

Die Anweisung SHOW-SYNTAX-FILE gibt Informationen über eine System- oder Gruppensyntaxdatei aus. Diese kann entweder die gerade bearbeitete oder eine explizit bestimmte Syntaxdatei sein.

Die Anweisung gibt folgende Informationen aus:

- Name, Typ, Version und Format der Syntaxdatei
- Versionsinformationen zu Produkten, die in der Syntaxdatei enthalten sind
- Globalinformationen aus der Syntaxdatei (ausschnittsweise)
- Privilegierung der Kommandos (nur bei INFORMATION=CMD-INTERFACE)

#### SHOW-SYNTAX-FILE

```
FILE = *CURRENT / *INPUT-FILE / <filename 1..54>
,INFORMATION = ALL-ATTRIBUTES / VERSIONS / GLOBALS / CMD-INTERFACE
,PRODUCT-NAME = *ALL / <structured-name 1..15>
```

#### FILE =

Bestimmt die System- oder Gruppensyntaxdatei, über die Informationen ausgegeben werden.

#### FILE = \*CURRENT

Der Operandenwert darf ausschließlich nach (mindestens) einer vorausgegangenen OPEN-Anweisung angegeben werden. Die Informationen beziehen sich

- unmittelbar nach einer OPEN-Anweisung auf die unter INPUT-FILE angegebene Syntaxdatei.
- nach einer MERGE- oder REMOVE-Anweisung auf die aktuelle temporäre SDF-I-Arbeitsdatei.

#### FILE = \*INPUT-FILE

Der Operandenwert darf ausschließlich nach (mindestens) einer vorausgegangenen OPEN-Anweisung angegeben werden. Die Informationen beziehen sich auf die Eingabesyntaxdatei, die bei der Anweisung OPEN unter INPUT-FILE angegeben wurde.

#### FILE = <filename 1..54>

Die Informationen beziehen sich auf die explizit angegebene System- oder Gruppensyntaxdatei.

**INFORMATION =**

Bestimmt die Art von Informationen, die ausgegeben werden. Alle Ausgaben enthalten den Syntaxdatei-Typ (GROUP oder SYSTEM).

**INFORMATION = ALL-ATTRIBUTES**

Folgende Informationen werden ausgegeben:

- Name, Typ, Version und Format der Syntaxdatei
- Versionen aller SEDDs, die in der angegebenen Syntaxdatei enthalten sind (für selbsterstellte Syntaxdateien wird als Version \*CUSTOM\* ausgegeben)
- SDF-Globalinformation (ausschnittsweise)

**INFORMATION = VERSIONS**

Folgende Informationen werden ausgegeben:

- Name, Typ, Version und Format der Syntaxdatei
- Versionen aller SEDDs, die in der angegebenen Syntaxdatei enthalten sind (für selbsterstellte Syntaxdateien wird als Version \*CUSTOM\* ausgegeben)

**INFORMATION = GLOBALS**

Folgende Informationen werden ausgegeben:

- Name, Typ, Version und Format der Syntaxdatei
- SDF-Globalinformation (ausschnittsweise)

**INFORMATION = CMD-INTERFACE**

Ausgegeben wird Name, Typ, Version und Format der Syntaxdatei und eine Liste der Kommandos, die in der Syntaxdatei definiert sind. Ab BS2000/OSD-BC V1.0 wird auch die Privilegierung der Kommandos ausgegeben (siehe Beispiel, [Seite 95](#)).

**PRODUCT-NAME =**

Dieser Operand wirkt nur bei Angabe von INFORMATION=CMD-INTERFACE und bestimmt, ob nur die Kommandos zu einem bestimmten Produkt ausgegeben werden.

**PRODUCT-NAME = \*ALL**

Alle Kommandos der angegebenen Syntaxdatei werden ausgegeben.

Bei der Ausgabe aus einer von Fujitsu Siemens Computers ausgelieferten SEDD (Softwareeinheitssyntaxdatei) oder aus einer INSD (Installationssyntaxdatei), werden selbsterstellte Kommandos, die eingemischt wurden, nicht angezeigt.

**PRODUCT-NAME = <structured-name 1..16>**

Alle Kommandos des angegebenen Produkts werden ausgegeben.

Sollen selbsterstellte Kommandos, die in eine von Fujitsu Siemens Computers ausgelieferte SEDD (Softwareeinheitssyntaxdatei) oder eine INSD (Installationssyntaxdatei) eingemischt wurden, angezeigt werden, muss die beim Einmischen verwendete REMOVE-ID als Produktname angegeben werden.



- (1) SDF-I wird gestartet.
- (2) Für die Systemsyntaxdatei des BS2000-Grundausbau werden Informationen zu allen enthaltenen Kommandos und deren Privilegierung angefordert. Im Beispiel wird die Ausgabe nur ausschnittsweise wiedergegeben.
- (3) Die Syntaxdatei hat Format V4.1. Sie wurde also bereits mit SDF-I V4.1 bearbeitet. Neben den Formaten V1 bis V4.1, die den SDF-I-Versionen V1.0 bis V4.1 entsprechen, kann auch OLD oder NEW ausgegeben werden. OLD entspricht dabei V1, NEW sind alle Formate V2 bis V4.1.
- (4) Unter anderem wird für das Kommando SHOW-FILE die Privilegierung für das Kommando (ALL-Zeile) und für mögliche Eingabemodi ausgegeben. Die Eingabemodi entsprechen den Modi, die mit der Anweisung ADD-CMD (siehe Handbuch „SDF-A“ [7]) bei der Kommando-Definition festgelegt wurden:

```
BA:  BATCH-ALLOWED
BPA: BATCH-PROC-ALLOWED
DA:  DIALOG-ALLOWED
DPA: DIALOG-PROC-ALLOWED
GA:  GUIDED-ALLOWED
CA:  CMD-ALLOWED
```

Nach dem Eingabemodus werden im Rest der Zeile die zugehörigen Privilegien ausgegeben. Jedes Zeichen (0 oder 1) steht für ein Privileg. 1 bedeutet, dass für einen Benutzerauftrag mit diesem Privileg die Ausführung des Kommandos in diesem Modus erlaubt ist. Die Reihenfolge der Privilegien in dieser Zeichenkette entspricht der Reihenfolge, in der die Privilegien aktuell im System definiert sind. Wird für einen bestimmten Eingabemodus keine Zeile ausgegeben, so ist für diesen Eingabemodus PRIVILEGE=\*SAME definiert, d.h. die Privilegien entsprechen genau denen, die für das Kommando definiert sind.

- (5) SDF-I fordert mit nach Beendigung der Ausgabe mit '\*' zu weiteren Eingaben auf.



## 6 SDF-U

SDF-U ist ein Programm, mit dem die Systemverwaltung Gruppen- und Systemsyntaxdateien gemäß den Belangen des Rechenzentrums modifizieren kann. Es ist vorrangig für die Installation neuer Syntaxdateiversionen gedacht und bedient sich dabei einer Untermenge an Funktionen und Anweisungen des Softwareprodukts SDF-A (siehe Handbuch „SDF-A“ [7]).

Die Anweisungen an SDF-U sind in einer separaten SDF-U-Syntaxdatei definiert.

### Starten von SDF-U

SDF-U kann unter jeder Benutzerkennung mit folgendem Kommando gestartet werden:

<b>START-SDF-U</b>	Kurzname: <b>SDF-U</b>
<b>VERSION</b> = <u>*STD</u> / <product-version> <b>,MONJV</b> = <u>*NONE</u> / <filename 1..54 without-gen-vers> <b>,CPU-LIMIT</b> = <u>*JOB-REST</u> / <integer 1..32767 <i>seconds</i> >	

### **VERSION =**

Sind mehrere Versionen von SDF-U mit IMON installiert, so kann der Benutzer die Version auswählen, mit der er arbeiten möchte. Bei Angabe der Versionsbezeichnung mit Hochkommas kann der Buchstabe C vorangestellt werden (C-STRING-Syntax).

Falls das Produkt nicht mit IMON installiert wurde oder die angegebene Version nicht existiert, gilt VERSION=\*STD.

### **VERSION = \*STD**

Aufruf der SDF-U-Version mit der höchsten Versionsnummer.

### **VERSION = <product-version>**

Angabe der SDF-U-Version im Format mm.n[a[so]] (vgl. „[product-version](#)“ auf Seite 14).

### **MONJV =**

Angabe einer Monitor-Jobvariablen zur Überwachung des SDF-U-Laufs.

### **MONJV = \*NONE**

Es wird keine Monitor-Jobvariable verwendet.

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

Name der zu verwendenden Jobvariablen.

**CPU-LIMIT =**

Maximale CPU-Zeit in Sekunden, die das Programm beim Ablauf verbrauchen darf.

**CPU-LIMIT = \*JOB-REST**

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

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

Es soll nur die angegebene Zeit verwendet werden.

### SDF-Standardanweisungen

Die SDF-Standardanweisungen (siehe [Seite 99](#)) sind ab SDF V4.1 und SDF-U V4.1 nicht mehr in der SDF-U-Syntaxdatei, sondern in der Syntaxdatei von SDF definiert. Sie werden systemweit für Benutzerprogramme angeboten, die ihre Anweisungen über SDF einlesen.

### Unterbrechen und Fortsetzen von SDF-U

Wird SDF-U während der Bearbeitung einer Anweisung mit [\[K2\]](#) unterbrochen, so kann es auf zwei Arten fortgesetzt werden.

- Wird SDF-U mit dem Kommando RESUME-PROGRAM fortgesetzt, so wird die unterbrochene Anweisung weiter bearbeitet.
- Wird SDF-U mit dem Kommando INFORM-PROGRAM wieder fortgesetzt, so wird die unterbrochene Anweisung abgebrochen und SDF-U steht nach Ausgabe der Meldung SDU0443 für eine neue Anweisung bereit.

## 6.1 SDF-U-Anweisungen

### 6.1.1 Funktionelle Übersicht

#### SDF-Standardanweisungen

Die ausführliche Beschreibung der SDF-Standardanweisungen finden Sie im Benutzerhandbuch „Einführung in die Dialogschnittstelle SDF“ [1].

END	SDF-U-Lauf beenden
EXECUTE-SYSTEM-CMD	Systemkommando während des SDF-U-Laufs ausführen
HELP-MSG-INFORMATION	Text einer Systemmeldung auf SYSOUT ausgeben
HOLD-PROGRAM	SDF-U-Lauf anhalten
MODIFY-SDF-OPTIONS	Benutzersyntaxdatei aktivieren oder deaktivieren und SDF-Einstellungen ändern
REMARK	Kommentare in die Anweisungsfolge schreiben und die Ausführung eines Programms dokumentieren
RESET-INPUT-DEFAULTS	Taskspezifische Default-Werte zurücksetzen
RESTORE-SDF-INPUT	Vorher eingegebene Anweisungen oder Kommandos am Bildschirm wiederanzeigen
SHOW-INPUT-DEFAULTS	Taskspezifische Default-Werte anzeigen
SHOW-INPUT-HISTORY	Inhalt des Eingabepuffers anzeigen
SHOW-SDF-OPTIONS	Taskspezifische Information über aktivierte Syntaxdateien und die Festlegungen für die Anweisungseingabe und -verarbeitung anzeigen
SHOW-STMT	Syntax einer Anweisung anzeigen
STEP	Wiederaufsetzpunkt in einer Folge von SDF-U-Anweisungen definieren
WRITE-TEXT	Einen bestimmten Text nach SYSOUT oder SYSLST ausgeben

**Syntaxdateien bearbeiten und erstellen**

COPY	Inhalte einer Syntaxdatei in die geöffnete Syntaxdatei kopieren
EDIT	Auf ein Kommando der geöffneten Syntaxdatei positionieren
MODIFY-CMD	Definition eines Kommandos in der geöffneten Syntaxdatei ändern
MODIFY-VALUE	Definition eines Operandenwerts in der geöffneten Syntaxdatei ändern
OPEN-SYNTAX-FILE	Syntaxdatei zur Bearbeitung mit SDF-U öffnen. Erste Anweisung nach dem Aufruf von SDF-U
REMOVE	Objekte in der geöffneten Syntaxdatei löschen
SET-GLOBALS	Allgemeine Festlegungen über die Kommandoeingabe und -verarbeitung ändern

**Anzeigefunktion**

SHOW	Inhalte einer geöffneten Syntaxdatei auf SYSOUT oder SYSLST anzeigen
SHOW-CORRECTION- INFORMATION	Korrekturinformationen aus der geöffneten Syntaxdatei ausgeben (nur für Diagnosezwecke)
SHOW-STATUS	Name der geöffneten Syntaxdatei ausgeben

## 6.1.2 Beschreibung der Anweisungen

### COPY

#### Inhalte einer Syntaxdatei kopieren

Mit der Anweisung COPY kopieren Sie Inhalte einer Syntaxdatei. Die Kopien fügt SDF-U in die geöffnete Syntaxdatei ein. Die geöffnete Syntaxdatei und die Syntaxdatei, deren Inhalt kopiert wird, müssen vom gleichen Typ sein.

COPY
<p><b>OBJECT</b> = *<b>DOMAIN</b>(...) / *<b>COMMAND</b>(...) / *<b>PROGRAM</b>(...)</p> <p>*<b>DOMAIN</b>(...)</p> <p>    <b>NAME</b> = *<b>ALL</b>(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>        *<b>ALL</b>(...)</p> <p>            <b>EXCEPT</b> = *<b>NONE</b> / &lt;structured-name 1..30 with-wild&gt; /</p> <p>                list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>*<b>COMMAND</b>(...)</p> <p>    <b>NAME</b> = *<b>ALL</b>(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>        *<b>ALL</b>(...)</p> <p>            <b>EXCEPT</b> = *<b>NONE</b> / &lt;structured-name 1..30 with-wild&gt; /</p> <p>                list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>*<b>PROGRAM</b>(...)</p> <p>    <b>NAME</b> = *<b>ALL</b>(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>        *<b>ALL</b>(...)</p> <p>            <b>EXCEPT</b> = *<b>NONE</b> / &lt;structured-name 1..30 with-wild&gt; /</p> <p>                list-poss(2000): &lt;structured-name 1..30&gt;</p> <p>,<b>FROM-FILE</b> = &lt;filename 1..54 without-gen-vers&gt;</p> <p>,<b>ATTACHED-INFO</b> = *<b>YES</b> / *<b>NO</b> / *<b>ONLY</b></p> <p>,<b>OVERWRITE-POSSIBLE</b> = *<b>NO</b> / *<b>EXTERNAL-ATTRIBUTES</b> / *<b>YES</b></p>

**OBJECT =**

Art des Objekts, dessen Definition zu kopieren ist.

**OBJECT = \*DOMAIN(...)**

Bestimmt, dass die Definitionen von Anwendungsbereichen kopiert werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Anwendungsbereichen werden kopiert.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Anwendungsbereiche werden nicht kopiert.

**NAME = <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Anwendungsbereiche werden kopiert, bzw. die Definitionen der Anwendungsbereiche, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*COMMAND(...)**

Bestimmt, dass die Definitionen von Kommandos kopiert werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Kommandos werden kopiert.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Kommandos werden nicht kopiert.

**NAME = <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Kommandos werden kopiert, bzw. die Definitionen der Kommandos, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*PROGRAM(...)**

Bestimmt, dass die Definitionen von Programmen kopiert werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Programmen werden kopiert.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Programme werden nicht kopiert.

**NAME = <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Programme werden kopiert, bzw. die Definitionen der Programme, deren Name zum Wildcard-Suchmuster passt.

**FROM-FILE = <filename 1..54>**

Syntaxdatei, die die zu kopierenden Definitionen enthält.

**ATTACHED-INFO =**

Bestimmt, welche der Definitionen, die zu dem spezifizierten Objekt gehören, kopiert werden.

**ATTACHED-INFO = \*YES**

Die Definition des angegebenen Objekts wird einschließlich der Definitionen aller Objekte kopiert, die dem spezifizierten zugeordnet sind. (Anwendungsbereich mit zugehörigen Kommandos, Programm mit zugehörigen Anweisungen, Kommando oder Anweisung mit zugehörigen Operanden.)

**ATTACHED-INFO = \*NO**

Die Definition des spezifizierten Objekts wird ohne die Definitionen der Objekte kopiert, die dem spezifizierten zugeordnet sind. (Anwendungsbereich ohne zugehörige Kommandos, Programm ohne zugehörige Anweisungen, Kommandos oder Anweisung ohne zugehörige Operanden.)

**ATTACHED-INFO = \*ONLY**

Ausschließlich die Definitionen der Objekte, die dem spezifizierten zugeordnet sind, werden kopiert. Die Definition des spezifizierten Objekts selbst wird nicht kopiert.

**OVERWRITE-POSSIBLE =**

Bestimmt, ob ein Objekt kopiert wird, wenn dieses in der geöffneten Syntaxdatei bereits definiert ist.

**OVERWRITE-POSSIBLE = \*NO**

SDF-U lehnt das Kopieren mit einer entsprechenden Meldung ab, wenn das Objekt bereits in der geöffneten Syntaxdatei definiert ist.

**OVERWRITE-POSSIBLE = \*EXTERNAL-ATTRIBUTES**

SDF-U kopiert nur die Objekte ohne die Definition der Objekte selbst. Die Definition des „aktuellen“ Objektes bleibt erhalten. Die Angabe dieses Operanden ist nur möglich für das Kopieren von Anwendungsbereichen und Programmen (COPY OBJ=\*DOMAIN... oder OBJ=\*PROGRAM...). Der Operand ATTACHED-INFO erhält den Wert \*NO, unabhängig davon, ob Sie einen anderen Wert angegeben haben.

**OVERWRITE-POSSIBLE = \*YES**

SDF-U kopiert unabhängig davon, ob das Objekt bereits in der geöffneten Syntaxdatei definiert ist. Dabei ersetzt SDF-U ggf. die in der geöffneten Syntaxdatei vorhandene Definition durch die zu kopierende Definition.

## EDIT

### Auf ein Kommando der Syntaxdatei positionieren

Mit der Anweisung EDIT erklären Sie ein Kommando oder einen Operandenwert zum „aktuellen“ Objekt. Wenn Sie mit der MODIFY-CMD-Anweisung die Definition eines Kommandos ändern wollen, müssen Sie zuvor dafür sorgen, dass dieses das „aktuelle“ Objekt ist.

EDIT
<pre> <b>OBJECT</b> = *<b>COMMAND</b>(...) / *<b>VALUE</b>(...)   *<b>COMMAND</b>(...)         <b>NAME</b> = &lt;structured-name 1..30&gt;   *<b>VALUE</b>(...)         <b>OPERAND-L1</b> = *<b>ABOVE-CURRENT</b> / &lt;structured-name 1..20&gt;         ,<b>VALUE-L1</b> = *<b>CURRENT</b> / &lt;structured-name 1..30&gt;         ,<b>ORIGIN</b> = *<b>CURRENT</b> / *<b>COMMAND</b>(...)             *<b>COMMAND</b>(...)                     <b>NAME</b> = &lt;structured-name 1..30&gt;           </pre>

#### **OBJECT = \*COMMAND(...)**

Ein Kommando wird aktuelles Objekt.

**NAME = <structured-name 1..30>**

Name des Kommandos.

#### **OBJECT = \*VALUE(...)**

Ein Operandenwert wird aktuelles Objekt.

**OPERAND-L1 = \*ABOVE-CURRENT / <structured-name 1..20>**

Bestimmt den Operanden, zu dem der zum aktuellen Objekt werdende Operandenwert gehört. \*ABOVE-CURRENT bedeutet, dass ein zu OPERAND-L1 gehörender Wert aktuelles Objekt ist. <structured-name 1..30> muss ein kommandoglobal eindeutiger Operandenname sein.

**VALUE-L1 = \*CURRENT / <structured-name 1..30>**

Bestimmt den Operandenwert, der aktuelles Objekt wird. \*CURRENT bedeutet, dass VALUE-L1 aktuelles Objekt ist. Ist VALUE-L1 nicht aktuelles Objekt und vom Datentyp \*KEYWORD, so ist der für ihn definierte Einzelwert anzugeben. Dabei ist zu beachten, dass dieser Einzelwert in jedem Fall ohne vorangestellten Stern anzugeben ist. Ist der Operandenwert nicht vom Typ \*KEYWORD, so ist der für ihn definierte Datentyp anzugeben.



**ORIGIN =**

Bestimmt das Kommando, in dem der angegebene Operandenwert aktuelles Objekt wird.

**ORIGIN = \*CURRENT**

Der Operandenwert gehört zu dem Kommando, das zurzeit entweder selbst aktuelles Objekt ist oder einen Operanden oder Operandenwert enthält, der aktuelles Objekt ist.

**ORIGIN = \*COMMAND(...)**

Der Operandenwert gehört zu dem bei NAME angegebenen Kommando.

**NAME=<structured-name 1..30>**

Name des Kommandos.

## **END**

### **Programmlauf beenden**

Mit der Anweisung END beenden Sie die Eingabe an SDF-U und schließen die zuletzt geöffnete Syntaxdatei.

END

Diese Anweisung hat keine Operanden.

## MODIFY-CMD

### Kommandodefinition ändern

Mit der Anweisung MODIFY-CMD ändern Sie in der bearbeiteten Syntaxdatei die Definition eines Kommandos. Dieses Kommando muss „aktuelles“ Objekt sein.

#### MODIFY-CMD

**IMPLEMENTOR** = \*UNCHANGED / \*PROCEDURE(...)

\*PROCEDURE(...)

| **NAME** = \*UNCHANGED / <c-string 1..54>

#### **IMPLEMENTOR =**

Art der Kommandoimplementierung.

#### **IMPLEMENTOR = \*UNCHANGED**

Keine Änderung bezüglich der Kommandoimplementierung.

#### **IMPLEMENTOR = \*PROCEDURE(...)**

Das Kommando ist durch eine Kommandoprozedur implementiert. Die Eingabe des Kommandos bewirkt den Aufruf der Kommandoprozedur.

**NAME = \*UNCHANGED / <c-string 1..54>**

Name der Datei, in der die Prozedur steht. Bibliothekselemente können mit 'library(element)' angegeben werden.

## MODIFY-VALUE

### Operandenwertdefinition ändern

Mit der Anweisung MODIFY-VALUE ändern Sie in der bearbeiteten Syntaxdatei die Wertdefinition eines Operandenwertes. Dieser Operandenwert muss „aktuelles“ Objekt sein.

#### MODIFY-VALUE

**VALUE** = \*UNCHANGED / list-poss(2000): <c-string 1..1800 with-low>(…)

<c-string 1..1800 with-low>(…)

| **OUTPUT** = \*UNCHANGED / <c-string 1..1800>

#### VALUE =

Bestimmt, welche Werte als Eingabe zulässig sind.

#### VALUE = \*UNCHANGED

Keine Änderung bezüglich der zulässigen Eingabewerte.

#### VALUE = list-poss(2000): <c-string 1..1800 with-low>(…)

Der Operandenwert muss einen der angegebenen Werte haben (zwingend für Werte vom Typ KEYWORD). Sofern hier keine Liste angegeben ist, kann der Benutzer den hier angegebenen Wert im Gegensatz zum STANDARD-NAME und zum ALIAS-NAME bei der Eingabe abkürzen. Ist der Operandenwert vom Typ KEYWORD, so ist Listenangabe nicht zulässig.

#### OUTPUT =

Bestimmt, welcher Wert an die Implementierung übergeben wird.

#### OUTPUT = \*UNCHANGED

Keine Änderung bezüglich des zu übergebenden Werts.

#### OUTPUT = <c-string 1..1800>

Der hier angegebene Wert wird übergeben.

## OPEN-SYNTAX-FILE

### Syntaxdatei öffnen

Mit der Anweisung OPEN-SYNTAX-FILE öffnen Sie eine Gruppen- oder Systemsyntaxdatei zur Bearbeitung mit SDF-U. Es ist die erste Anweisung (abgesehen von den Standardanweisungen), die Sie nach dem Aufruf von SDF-U geben müssen. Jede weitere OPEN-SYNTAX-FILE-Anweisung bewirkt implizit, dass SDF-U die zuvor geöffnete Syntaxdatei schließt.

#### OPEN-SYNTAX-FILE

```

FILE = <filename 1..54>
, TYPE = *GROUP(...) / *SYSTEM
    *GROUP(...)
        | SYSTEM-DESCRIPTION = *CURRENT / *NO
, MODE = *UPDATE / *READ

```

#### **FILE = <filename 1..54>**

Name der zu öffnenden Syntaxdatei.

#### **TYPE =**

Art der zu öffnenden Syntaxdatei.

#### **TYPE = \*GROUP(...)**

Eine Gruppensyntaxdatei ist zu öffnen.

#### **SYSTEM-DESCRIPTION = \*CURRENT / \*NO**

Legt fest, ob SDF-U auf die derzeit aktivierte Systemsyntaxdatei zugreift.

#### **TYPE = \*SYSTEM**

Eine Systemsyntaxdatei ist zu öffnen.

#### **MODE =**

Legt fest, wie die geöffnete Syntaxdatei bearbeitet werden darf.

#### **MODE = \*UPDATE**

Der Inhalt der Syntaxdatei kann sowohl ausgegeben als auch verändert werden. Die Syntaxdatei existiert bereits. Sie darf nicht aktiviert sein.

#### **MODE = \*READ**

Der Inhalt der Syntaxdatei kann nur ausgegeben, aber nicht verändert werden (nur Lesezugriff). Die Syntaxdatei existiert bereits. Sie darf aktiviert sein.

## REMOVE

### Objekte der Syntaxdatei löschen

Mit der Anweisung REMOVE löschen Sie Objekte in der bearbeiteten Syntaxdatei. Objekte in diesem Sinne sind Anwendungsbereiche, Kommandos und Programme.

REMOVE
<pre> <b>OBJECT</b> = *DOMAIN(...) / *COMMAND(...) / *PROGRAM(...)   *DOMAIN(...)             <b>NAME</b> = *ALL(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;             *ALL(...)                     <b>EXCEPT</b> = *NONE / &lt;structured-name 1..30 with-wild&gt; /           list-poss(2000): &lt;structured-name 1..30&gt;         *COMMAND(...)             <b>NAME</b> = *ALL(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;             *ALL(...)                     <b>EXCEPT</b> = *NONE / &lt;structured-name 1..30 with-wild&gt; /           list-poss(2000): &lt;structured-name 1..30&gt;         *PROGRAM(...)             <b>NAME</b> = *ALL(...) / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;             *ALL(...)                     <b>EXCEPT</b> = *NONE / &lt;structured-name 1..30 with-wild&gt; /           list-poss(2000): &lt;structured-name 1..30&gt;           </pre>

#### **OBJECT =**

Art des Objekts, das zu löschen ist.

#### **OBJECT = \*DOMAIN(...)**

Bestimmt, dass Anwendungsbereiche gelöscht werden. Die diesen Anwendungsbereichen zugeordneten Kommandos werden gelöscht, sofern sie nicht noch mindestens einem anderen Anwendungsbereich zugeordnet oder mit REMOVE-POSSIBLE=\*NO definiert sind.

#### **NAME = \*ALL(...)**

Alle Anwendungsbereiche werden gelöscht.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /  
list-poss(2000): <structured-name 1..30>**

Die hier angegebenen Anwendungsbereiche werden nicht gelöscht.

**NAME = <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die namentlich genannten Anwendungsbereiche werden gelöscht, bzw. die Anwendungsbereiche, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*COMMAND(...)**

Bestimmt, dass Kommandos gelöscht werden. Das schließt das Löschen der zugehörigen Operanden mit ein.

**NAME = \*ALL(...)**

Alle Kommandos werden gelöscht.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die hier angegebenen Kommandos werden nicht gelöscht.

**NAME = <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die namentlich genannten Kommandos werden gelöscht, bzw. die Kommandos, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*PROGRAM(...)**

Bestimmt, dass Programme gelöscht werden. Das schließt das Löschen der zugehörigen Anweisungen mit ein.

**NAME = \*ALL(...)**

Alle Programme werden gelöscht.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die hier angegebenen Programme werden nicht gelöscht.

**NAME = <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die namentlich genannten Programme werden gelöscht, bzw. die Programme, deren Name zum Wildcard-Suchmuster passt.

## SET-GLOBALS

### Globalinformation ändern

Mit der Anweisung SET-GLOBALS ändern Sie die allgemeinen Festlegungen über die Kommando-/Anweisungseingabe und -verarbeitung. Diese Festlegungen sind in der Globalinformation enthalten und werden wirksam, sobald die Syntaxdatei aktiviert wird.

#### SET-GLOBALS

```

VERSION = *UNCHANGED / <alphanum-name 1..12> / <c-string 1..12>
, GUIDANCE = *UNCHANGED / *STD / *MAXIMUM / *MEDIUM / *MINIMUM / *NO / *EXPERT
, LOGGING = *UNCHANGED / *STD / *INPUT-FORM / *ACCEPTED-FORM / *INVARIANT-FORM
, PROCEDURE-DIALOGUE = *UNCHANGED / *STD / *YES / *NO
, UTILITY-INTERFACE = *UNCHANGED / *STD / *OLD-MODE / *NEW-MODE
, CONTINUATION = *UNCHANGED / *STD / *OLD-MODE / *NEW-MODE
, FUNCTION-KEYS = *UNCHANGED / *STD / *OLD-MODE / *STYLE-GUIDE-MODE / *BY-TERMINAL-TYPE
, INPUT-HISTORY = *UNCHANGED / *STD / *ON / *OFF
, NUMBER-OF-INPUTS = *UNCHANGED / *STD / <integer 1..100>

```

#### VERSION =

Legt die Versionsnummer der Syntaxdatei fest. Sie dient nur zu Dokumentationszwecken.

#### VERSION = \*UNCHANGED

Die Versionsnummer bleibt unverändert.

#### VERSION = <alphanum-name 1..12> / <c-string 1..12>

Die Syntaxdatei erhält die angegebene Versionsnummer. Leerzeichen am Ende eines c-string werden nicht berücksichtigt.

#### GUIDANCE =

Bestimmt den Umfang der Benutzerführung. (Diese Festlegung gilt nicht für Aufträge, die über OMNIS gestartet wurden; für diese wird grundsätzlich \*EXPERT angenommen.)

#### GUIDANCE = \*UNCHANGED

Die in der Globalinformation stehende Festlegung über die Benutzerführung bleibt unverändert.

#### GUIDANCE = \*STD

Die Benutzerführung bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*NO.



**GUIDANCE = \*MAXIMUM**

Zur Auswahl von Anwendungsbereichen, Kommandos und Anweisungen werden Menüs mit Erläuterungstexten ausgegeben. Zur Operandenbesetzung werden Fragebögen ausgegeben. Für alle Strukturen gibt es eigene Fragebögen. Die Fragebögen enthalten Hilfetexte zu den Operanden, die Default-Werte, alle erlaubten Operandenwerte und Zusatzinformationen zu diesen.

**GUIDANCE = \*MEDIUM**

Zur Auswahl von Anwendungsbereichen, Kommandos und Anweisungen werden Menüs mit Erläuterungstexten ausgegeben. Zur Operandenbesetzung werden Fragebögen ausgegeben. Für Strukturen, deren einleitender Wert mit SIZE=\*LARGE (siehe Anweisung ADD-VALUE im Handbuch „SDF-A“ [7]) definiert ist, gibt es eigene Fragebögen. Die Fragebögen enthalten die Default-Werte und alle erlaubten Operandenwerte.

**GUIDANCE = \*MINIMUM**

Zur Auswahl von Anwendungsbereichen, Kommandos und Anweisungen werden Menüs ausgegeben. Zur Operandenbesetzung werden Fragebögen ausgegeben. Für Strukturen, deren einleitender Wert mit SIZE=\*LARGE (siehe Anweisung ADD-VALUE im Handbuch „SDF-A“ [7]) definiert ist, gibt es eigene Fragebögen. Die Fragebögen enthalten nur die Default-Werte.

**GUIDANCE = \*NO**

Mit dem Text %CMD: bzw. %KDO: oder %STMT: bzw. %ANW: wird die Eingabe angefordert. Mehrere Kommandos können, getrennt durch ein „logisches Zeilenende“, in einem Block hintereinander eingegeben werden. Bei fehlerhafter Eingabe werden Korrekturmöglichkeiten geboten.

**GUIDANCE = \*EXPERT**

Mit / oder %// wird die Eingabe angefordert. Mehrere Kommandos können, getrennt durch ein „logisches Zeilenende“, in einem Block hintereinander eingegeben werden. Korrekturmöglichkeiten werden bei fehlerhafter Eingabe nicht angeboten.

**LOGGING =**

Bestimmt, wie die Eingaben protokolliert werden.

**LOGGING = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über die Protokollierung bleibt unverändert.

**LOGGING = \*STD**

Die Protokollierung bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*INPUT-FORM.

**LOGGING = \*INPUT-FORM**

Im ungeführten Dialog werden die Eingabezeichenfolgen genau wie eingegeben protokolliert. Kennwörter werden ausgeblendet. Im geführten Dialog oder beim Fehlerdialog wird wie bei \*ACCEPTED-FORM protokolliert.

**LOGGING = \*ACCEPTED-FORM**

Protokolliert werden:

- alle Namen in ihrer Langform
- jeder in der Eingabe vorkommende Operand mit seinem Namen und dem angegebenen Wert
- die ggf. durch Korrekturen erzeugten Enddarstellungen

Kennwörter sind ausgeblendet. Eingaben im geführten Dialog sind zu einem String verkettet.

**LOGGING = \*INVARIANT-FORM**

Protokolliert werden:

- alle Namen in der Fassung, die in der Syntaxdatei als STANDARD-NAME definiert ist (d.h. die in den Handbüchern angegebenen Namen)
- jeder in der Eingabe vorkommende Operand mit seinem Namen und dem angegebenen Wert
- alle in der Eingabe implizit enthaltenen optionalen Operanden mit ihren Default-Werten
- die ggf. durch Korrekturen erzeugten Enddarstellungen

Kennwörter sind ausgeblendet. Eingaben im geführten Dialog sind zu einem String verkettet.

**PROCEDURE-DIALOGUE =**

Bestimmt, ob der Benutzer zur interaktiven Korrektur aufgefordert werden soll, wenn bei einer im Dialogbetrieb ablaufenden Prozedur oder SYSSTMT-Datei Syntax- oder Semantikfehler auftreten.

**PROCEDURE-DIALOGUE = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über die interaktive Korrektur bleibt unverändert.

**PROCEDURE-DIALOGUE = \*STD**

Die Regelung über die interaktive Korrektur bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*NO.

**PROCEDURE-DIALOGUE = \*YES**

Der Benutzer wird zur interaktiven Korrektur aufgefordert.

**PROCEDURE-DIALOGUE = \*NO**

Der Benutzer wird zur interaktiven Korrektur nicht aufgefordert.

**UTILITY-INTERFACE =**

Setzt einen Schalter, der von einem Programm über den Makroaufruf CMDSTA abgefragt werden kann. Über diesen Schalter lässt sich bei Programmen, die ihre Anweisungen sowohl mit RDATA als auch über SDF mit RDSTMT lesen können, die Art der Anweisungseingabe steuern.

**UTILITY-INTERFACE = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über den Schalter bleibt unverändert.

**UTILITY-INTERFACE = \*STD**

Der Schalter bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*NEW.

**UTILITY-INTERFACE = \*OLD-MODE**

Programme sollen ihre Anweisungen mit RDATA lesen.

**UTILITY-INTERFACE = \*NEW-MODE**

Programme sollen ihre Anweisungen über SDF mit RDSTMT lesen.

**CONTINUATION =**

Bestimmt, in welcher Spalte bei der Kommandoingabe (SYSCMD) gegebenenfalls das Fortsetzungszeichen „-“ anzugeben ist. (Bei Anweisungseingabe (SYSSTMT) kann das Fortsetzungszeichen immer in einer beliebigen Spalte angegeben werden.)

**CONTINUATION = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über das Fortsetzungszeichen bleibt unverändert.

**CONTINUATION = \*STD**

Die Regelung über das Fortsetzungszeichen bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei bewirkt \*STD, dass die bei der Systemgenerierung getroffene Festlegung gilt.

**CONTINUATION = \*OLD-MODE**

Das Fortsetzungszeichen muss in Spalte 72 angegeben werden.

**CONTINUATION = \*NEW-MODE**

Das Fortsetzungszeichen kann in den Spalten 2 bis 72 angegeben werden.

**FUNCTION-KEYS =**

Bestimmt die Funktionstastenbelegung. Die verschiedenen Modi sind ausführlich im Benutzerhandbuch „Einführung in die Dialogschnittstelle SDF“ [1] beschrieben. Nicht unterstützte Funktionstasten wirken nicht wie DUE, sondern bewirken die leere Leistung.

**FUNCTION-KEYS = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über die Funktionstastenbelegung bleibt unverändert.

**FUNCTION-KEYS = \*STD**

Der Schalter bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*BY-TERMINAL-TYPE.

**FUNCTION-KEYS = \*OLD-MODE**

Die Belegung der Funktionstasten erfolgt im bisherigen Modus, der von allen Terminaltypen unterstützt wird. In diesem Modus gelten folgende Tastenbelegungen:

- K1** Exit-Funktion
- K2** Unterbrechungs-Funktion
- K3** Refresh-Funktion (nur im geführten Dialog)
- F1** Exit-all-Funktion
- F2** Test-Funktion (nur im geführten Dialog)
- F3** Execute-Funktion (nur im geführten Dialog)

**FUNCTION-KEYS = \*STYLE-GUIDE-MODE**

Die Belegung der Funktionstasten entspricht dem Fujitsu-Siemens-Styleguide. In diesem Modus gelten folgende Tastenbelegungen:

- K2** Unterbrechungs-Funktion
- F1** Hilfe-Funktion
- F3** Exit-Funktion
- F5** Refresh-Funktion (nur im geführten Dialog)
- F6** Exit-all-Funktion
- F7** Rückwärts blättern (nur im geführten Dialog)
- F8** Vorwärts blättern (nur im geführten Dialog)
- F9** RESTORE-SDF-INPUT INPUT=\*LAST ausführen
- F11** Execute-Funktion
- F12** Cancel-Funktion

**FUNCTION-KEYS = \*BY-TERMINAL-TYPE**

Die Belegung der Funktionstasten ist abhängig von dem Terminaltyp. Unterstützt der Terminaltyp die umfangreichere Funktionalität des Fujitsu-Siemens-Styleguide, wählt SDF die Einstellung \*STYLE-GUIDE-MODE. Anderenfalls wählt SDF die Einstellung \*OLD-MODE.



Bei der Einstellung wird ausgewertet, mit welchem Typ das Terminal im System generiert wurde. Weicht der generierte Terminaltyp vom tatsächlichen Terminaltyp ab, ist nicht gewährleistet, dass die Einstellung dem tatsächlich unterstützten Funktionsumfang entspricht. Bei einer Terminalemulation kann der erkannte Terminaltyp sowohl von der Generierung als auch von Umgebungsvariablen abhängen. Genaueres ist der Beschreibung des Emulationsprogrammes zu entnehmen.

**INPUT-HISTORY =**

Bestimmt, ob der Eingabepuffer eingeschaltet, ausgeschaltet oder zurückgesetzt wird.

**INPUT-HISTORY = \*UNCHANGED**

Die in der Globalinformation stehende Festlegung über die Speicherung der Eingaben bleibt unverändert.

**INPUT-HISTORY = \*STD**

Der Schalter bleibt unverändert, wenn die bearbeitete Gruppensyntaxdatei aktiviert wird. Für eine Systemsyntaxdatei hat \*STD die gleiche Wirkung wie \*ON.

**INPUT-HISTORY = \*ON**

Der Eingabepuffer wird eingeschaltet. SDF speichert syntaktisch richtige Eingaben (Kommandos und Anweisungen) im Eingabepuffer. Nicht gespeichert werden SET-LOGON-PARAMETERS, RESTORE-SDF-INPUT und SHOW-INPUT-HISTORY.

Ob ISP-Kommandos gespeichert werden, ist abhängig von der Angabe im Operanden PASSWORD-PROTECTION (Kommando/Anweisung MODIFY-SDF-OPTIONS).

Mit SHOW-INPUT-HISTORY kann sich der Benutzer die gespeicherten Eingaben ausgeben lassen. Mit RESTORE-SDF-INPUT kann der Benutzer eine bestimmte Eingabe ausgeben lassen, um sie dann unverändert oder auch modifiziert erneut abzuschicken.



Wertangaben für „geheime“ Operanden, die weder dem Default-Wert noch einem mit SECRET=\*NO definierten Wert entsprechen, werden im Eingabepuffer abhängig vom Operanden PASSWORD-PROTECTION mit „^“ oder im Klartext gespeichert.

Wertangaben zu nicht „geheimen“ Operanden werden im Eingabepuffer im Klartext gespeichert. Im Einzelfall können diese Eingaben aus Benutzersicht auch schützenswerte Informationen darstellen (z.B. Prozedurparameter). Soll verhindert werden, dass diese Eingaben mit SHOW-INPUT-HISTORY bzw. RESTORE-SDF-INPUT erneut am Bildschirm sichtbar gemacht werden können, so kann der Benutzer wie folgt vorgehen:

Vor sicherheitsrelevanten Eingaben den Eingabepuffer ausschalten und danach wieder einschalten. Wurden die Eingaben bereits gespeichert, kann der Eingabepuffer mit \*RESET zurückgesetzt werden. In diesem Fall werden jedoch alle gespeicherten Eingaben gelöscht.

**INPUT-HISTORY = \*OFF**

Der Eingabepuffer wird ausgeschaltet. Die nachfolgenden Eingaben werden nicht gespeichert. Alle bisher gespeicherten Eingaben sind jedoch weiterhin zugreifbar.

**NUMBER-OF-INPUTS = \*UNCHANGED / \*STD / <integer 1..100>**

Gehört zum Operanden INPUT-HISTORY und legt fest, wie viele Eingaben in dem Eingabepuffer gespeichert werden können. Die obere Grenze liegt bei 100. Bei Erreichen der maximalen Anzahl gespeicherter Eingaben wird die jeweils älteste Eingabe gelöscht.

## SHOW Objekte der Syntaxdatei anzeigen

Mit der Anweisung SHOW geben Sie Inhalte einer Syntaxdatei auf SYSOUT oder SYSLST aus. Die Ausgabe kann unterbrochen bzw. wieder gestartet oder mit **[K2]** abgebrochen werden.

(Teil 1 von 2)

### SHOW

**OBJECT** = \*GLOBAL-INFORMATION / \*DOMAIN(...) / \*COMMAND(...) / \*PROGRAM(...)

\*DOMAIN(...)

**NAME** = \*ALL(...) / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

\*ALL(...)

**EXCEPT** = \*NONE / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

\*COMMAND(...)

**NAME** = \*ALL(...) / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

\*ALL(...)

**EXCEPT** = \*NONE / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

\*PROGRAM(...)

**NAME** = \*ALL(...) / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

\*ALL(...)

**EXCEPT** = \*NONE / <structured-name 1..30 with-wild> /

list-poss(2000): <structured-name 1..30>

,ATTACHED-INFORMATION = \*YES / \*NO / \*IMMEDIATE

,SIZE = \*MINIMUM / \*MAXIMUM / \*MEDIUM

,IMPLEMENTATION-INFO = \*NO (...)

\*NO(...)

**FORM** = \*UNGUIDED / \*GUIDED

,LANGUAGE = E / <name 1..1>

Fortsetzung →

```

, LINES-PER-PAGE = *STD / *UNLIMITED(...) / <integer 1..200>
    *UNLIMITED(...)
        |   OUTPUT-FORM = *STD / *FOR-INPUT
, OUTPUT = *SYSOUT / *SYSLST(...)
    *SYSLST(...)
        |   SYSLST-NUMBER = *STD / <integer 1..99>
, LINE-LENGTH = *STD / <integer 72..132>
, PRIVILEGE = *ANY / list-poss(64): <structured-name 1..30>

```

**OBJECT =**

Art des Objekts, dessen Definition auszugeben ist.

**OBJECT = \*GLOBAL-INFORMATION**

Bestimmt, dass die Globalinformation der Syntaxdatei ausgegeben wird.

**OBJECT = \*DOMAIN(...)**

Bestimmt, dass die Definitionen von Anwendungsbereichen ausgegeben werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Anwendungsbereichen werden ausgegeben.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Anwendungsbereiche werden nicht ausgegeben.

**NAME = <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Anwendungsbereiche werden ausgegeben, bzw. die Definitionen der Anwendungsbereiche, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*COMMAND(...)**

Bestimmt, dass die Definitionen von Kommandos ausgegeben werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Kommandos werden ausgegeben.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /****list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Kommandos werden nicht ausgegeben.

**NAME = <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Kommandos werden ausgegeben, bzw. die Definitionen der Kommandos, deren Name zum Wildcard-Suchmuster passt.

**OBJECT = \*PROGRAM(...)**

Bestimmt, dass die Definitionen von Programmen ausgegeben werden.

**NAME = \*ALL(...)**

Die Definitionen von allen Programmen werden ausgegeben.

**EXCEPT = \*NONE / <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die Definitionen der hier angegebenen Programme werden nicht ausgegeben.

**NAME = <structured-name 1..30 with-wild> /**

**list-poss(2000): <structured-name 1..30>**

Die Definitionen der namentlich genannten Programme werden ausgegeben, bzw. die Definitionen der Programme, deren Name zum Wildcard-Suchmuster passt.

**ATTACHED-INFORMATION =**

Bestimmt, welche der Definitionen, die zu dem angegebenen Objekt gehören, ausgegeben werden.

**ATTACHED-INFORMATION = \*YES**

Die Definition des angegebenen Objekts wird einschließlich der Definitionen aller Objekte ausgegeben, die dem angegebenen Objekt zugeordnet sind. (Anwendungsbereich mit zugehörigen Kommandos, Programm mit zugehörigen Anweisungen, Kommando mit zugehörigen Operanden).

**ATTACHED-INFORMATION = \*NO**

Die Definition des angegebenen Objekts wird ohne die Definitionen der Objekte ausgegeben, die dem angegebenen zugeordnet sind. (Anwendungsbereich ohne zugehörige Kommandos, Programm ohne zugehörige Anweisungen, Kommando ohne zugehörige Operanden.)

**ATTACHED-INFORMATION = \*IMMEDIATE**

Die Definition des angegebenen Objekts wird mit den Definitionen der Objekte ausgegeben, die dem angegebenen Objekt unmittelbar zugeordnet sind. Anwendungsbereiche bzw. Programme werden mit den zugehörigen Kommandos und Anweisungen ausgegeben, jedoch ohne die zugehörigen Operanden und Operandenwerte. Für Kommandos wirkt \*IMMEDIATE wie \*YES.

**SIZE =**

Bestimmt den Umfang der Ausgabe. Was der Operand SIZE konkret bewirkt, hängt mit vom Operanden OBJECT ab. Der Operand hat keine Auswirkung auf die Ausgabe der Globalinformation. Für OBJECT=\*COMMAND werden folgende Informationen ausgegeben:



**SIZE = \*MINIMUM**

Der Kommandoname, alle zugehörigen Operandennamen, alle zugehörigen Default-Werte und die struktureinleitenden Operandenwerte werden ausgegeben. Zusatzinformationen zu den Operandenwerten und Hilfetexte werden nicht ausgegeben.

**SIZE = \*MAXIMUM**

Der Kommandoname, alle zugehörigen Operandennamen, alle zugehörigen Default-Werte und alle sonstigen zugehörigen Operandenwerte werden ausgegeben. Außerdem werden die Zusatzinformationen zu den Operandenwerten und die Hilfetexte ausgegeben.

**SIZE = \*MEDIUM**

Der Kommandoname, alle zugehörigen Operandennamen, alle zugehörigen Default-Werte und alle sonstigen zugehörigen Operandenwerte werden ausgegeben. Zusatzinformationen zu den Operandenwerten und Hilfetexte werden nicht ausgegeben.

**IMPLEMENTATION-INFO =**

Bestimmt, wie die Ausgabe aufbereitet wird.

**IMPLEMENTATION-INFO = \*NO(...)**

Die Definitionen der angegebenen Objekte werden in einer handbuchähnlichen Darstellung ausgegeben. Die Operanden, die mit PRESENCE=\*INTERNAL-ONLY belegt sind, werden hier nicht aufgelistet.

**FORM =**

Bestimmt, ob die Definitionen von Objekten, die für den geführten Dialog nicht zugelassen sind, ausgegeben werden.

**FORM = \*UNGUIDED**

Die Definitionen werden ausgegeben.

**FORM = \*GUIDED**

Die Definitionen werden nicht ausgegeben.

**LANGUAGE = E / <name 1..1>**

Bestimmt, in welcher Sprache die Hilfetexte ausgegeben werden (E=Englisch, D=Deutsch). Für die Globalinformation ist dieser Operand wirkungslos.

**LINES-PER-PAGE = \*STD / \*UNLIMITED(...) / <integer 1..200>**

Bestimmt die Anzahl der Zeilen pro Seite. Diese Zahl enthält nicht die beiden Zeilen für den Header, den SDF-U für jede neue Seite erzeugt. Der Header wird nur bei OUTPUT=\*SYSLST erzeugt.

**LINES-PER-PAGE = \*STD**

Der Default-Wert beträgt 24 Zeilen bei Ausgabe auf den Bildschirm und 55 Zeilen bei Ausgabe in eine Datei.

**LINES-PER-PAGE = \*UNLIMITED(...)**

Keine Kontrolle durch SDF-U (es wird kein Header erzeugt).

**OUTPUT-FORM =**

Legt fest, welche Zeichen am Zeilenanfang ausgegeben werden.

**OUTPUT-FORM = \*STD**

Das erste Zeichen jeder Zeile ist ein Leerzeichen.

**OUTPUT-FORM = \*FOR-INPUT**

Am Beginn jeder Zeile werden zwei Schrägstriche (//) ausgegeben.

Diese Angabe kann in Kombination mit IMPLEMENTATION=\*YES dazu verwendet werden, SDF-U-Anweisungen zu erzeugen, mit deren Hilfe eine Syntaxdatei oder ein Syntaxdatei-Objekt wieder hergestellt wird.

**OUTPUT =**

Bestimmt das Ausgabemedium für die gewünschten Informationen.

**OUTPUT = \*SYSOUT**

Die Ausgabe erfolgt in die logische Systemdatei, d.h. im Dialog in der Regel auf den Bildschirm.

**OUTPUT = \*SYSLST(...)**

Die Ausgabe erfolgt in die logische Systemdatei SYSLST.

**SYSLST-NUMBER = \*STD / <integer 1..99>**

Bestimmt die Nummer der logischen Systemdatei SYSLST. Bei Angabe von \*STD erhält die logische Systemdatei SYSLST keine Nummer.

**LINE-LENGTH = \*STD / <integer 72..132>**

Bestimmt die Zeilenlänge für die Ausgabe.

**LINE-LENGTH = \*STD**

Der Default-Wert beträgt 74 Zeichen bei Ausgabe auf den Bildschirm und 72 Zeichen bei Ausgabe in eine Datei.

**PRIVILEGE = \*ANY / list-poss(64): <structured-name 1..30>**

Es werden nur die Objekte ausgegeben, denen mindestens eines der in der Liste genannten Privilegien zugeordnet ist. Bei Angabe von \*ANY werden die Objekte ohne Beachtung ihrer Privilegien ausgegeben.

## SHOW-CORRECTION-INFORMATION

### Korrekturinformation der Syntaxdatei ausgeben

Die Anweisung SHOW-CORRECTION-INFORMATION gibt Informationen über in der Syntaxdatei enthaltene Korrekturen aus.

Die Anweisung ist nur für Diagnosezwecke vorgesehen.

SHOW-CORRECTION-INFORMATION
<pre> CORRECTION-ID = *SOURCE (...) / *OBJECT(...)  *SOURCE(...)     PM-NUMBER = *ALL / list-poss(100): &lt;alphanum-name 8..8&gt;  *OBJECT(...)     PM-NUMBER = *ALL / &lt;alphanum-name 8..8&gt;     ,JULIAN-DATE = *ANY / &lt;integer 1..366&gt;  ,PRODUCT-NAME = *ANY / &lt;structured-name 1..15&gt;(…) &lt;structured-name 1..15&gt;(…)     VERSION = *ANY / &lt;product-version&gt; </pre>

Die Operandenbeschreibung entfällt an dieser Stelle, da die Anweisung nur für Diagnosezwecke vorgesehen ist.

## SHOW-STATUS

### Status der geöffneten Syntaxdatei anzeigen

Mit der Anweisung SHOW-STATUS veranlassen Sie, dass SDF-U den Namen der gerade geöffneten Syntaxdatei und Informationen über eine oder mehrere zugeordnete Referenzsyntaxdateien ausgibt.

<b>SHOW-STATUS</b>

Diese Anweisung hat keine Operanden.

## STEP

### Definieren eines Wiederaufsetzpunktes

Mit der Anweisung STEP definieren Sie in einer Folge von SDF-U-Anweisungen einen Wiederaufsetzpunkt für die Fehlerbehandlung. Die Eingabe der Anweisung STEP ist nur in Prozeduren und Stapelaufträgen zulässig.

<b>STEP</b>

Diese Anweisung hat keine Operanden.

Trifft SDF-U auf einen syntaktischen oder schwerwiegenden logischen Fehler, dann leitet es folgende Schritte ein:

- es wird eine Fehlermeldung ausgegeben,
- die laufende SDF-U-Anweisung wird abgebrochen,
- die nachfolgenden Anweisungen werden bis zum Erreichen von STEP oder END überlesen.

Erreicht SDF-U als Nächstes ein END, so erzeugt es ein abnormales Programmende (TERM UNIT=STEP,MODE=ABNORMAL) und aktiviert den Spin-Off-Mechanismus. Erreicht SDF-U als Nächstes ein STEP, so fährt es mit der auf STEP folgenden Anweisung fort. Falls ein Fehler keinen Einfluss auf den ordnungsgemäßen Ablauf des laufenden Jobs hat, so muss der Benutzer von SDF-U den abnormalen Programmabbruch durch Einfügen eines STEP abfangen.

Geringfügige logische Fehler korrigiert SDF-U selbstständig unter Ausgabe einer Warnmeldung. Es wird kein Spin-Off-Mechanismus aktiviert.



---

## 7 SDF-PAR

SDF-PAR ist ein Dienstprogramm, mit dem Sie eine Parameterdatei für das Produkt SDF erzeugen, modifizieren und anzeigen können. Es bietet somit eine Alternative zu den privilegierten Kommandos MODIFY-SDF-PARAMETERS und SHOW-SDF-PARAMETERS, die im [Kapitel „Kommandos zur SDF-Verwaltung“](#) ab [Seite 41](#) beschrieben sind. Ein Vorteil von SDF-PAR besteht darin, dass Sie die Parameterdatei off-line bearbeiten können.

Das Programm SDF-PAR V1.1 kann ab SDF V1.4 eingesetzt werden. Die Anweisungen an SDF-PAR sind in einer Benutzersyntaxdatei definiert, die mit SDF-PAR ausgeliefert wird.

### Starten von SDF-PAR

Für den Start von SDF-PAR in BS2000/OSD-BC  $\geq$  V2.0 ist folgende, im Lieferumfang enthaltene Prozedur aufzurufen:

```
/CALL-PROCEDURE $.SYSPRC.SDF-PAR.011.112
```

Die Prozedur enthält neben dem eigentlichen Aufruf von SDF-PAR auch die notwendigen Vorbereitungen für den Programmaufruf (Meldungs- und Syntaxdatei von SDF-PAR werden aktiviert).

Hinweise zur Installation von SDF-PAR finden Sie im [Abschnitt „Installation von SDF-PAR“](#) auf [Seite 139](#) und in der Freigabemittelung zum Softwareprodukt SDF.

## 7.1 SDF-PAR-Anweisungen

### 7.1.1 Funktionelle Übersicht

#### SDF-Parameterdatei erzeugen, bearbeiten und anzeigen

END	SDF-PAR beenden
OPEN-PARAMETER-FILE	SDF-Parameterdatei erzeugen oder zur Bearbeitung öffnen
MODIFY-SYNTAX-FILE	Einträge für Syntaxdateien ändern
MODIFY-SYSTEM-LOGON-INCLUDE	Eintrag für die systemweite LOGON-Include-Prozedur ändern
MODIFY-SYSTEM-LOGON-PROCEDURE	Eintrag für die systemweite LOGON-Call-Prozedur ändern
MODIFY-VERSION	Format der Parameterdatei ändern
SHOW-PARAMETER-FILE	Inhalt der geöffneten SDF-Parameterdatei anzeigen

#### SDF-Standardanweisungen

Die Beschreibung dieser Anweisungen finden Sie im Benutzerhandbuch „Einführung in die Dialogschnittstelle SDF“ [1].

END	SDF-PAR beenden
EXECUTE-SYSTEM-CMD	Systemkommando während des SDF-PAR-Laufs ausführen
HELP-MSG-INFORMATION	Text einer Systemmeldung auf SYSOUT ausgeben
HOLD-PROGRAM	SDF-PAR-Lauf anhalten
MODIFY-SDF-OPTIONS	Benutzersyntaxdatei aktivieren oder deaktivieren und SDF-Einstellungen ändern
REMARK	Kommentare in die Anweisungsfolge schreiben und die Ausführung eines Programms dokumentieren
RESET-INPUT-DEFAULTS	Taskspezifische Default-Werte zurücksetzen
RESTORE-SDF-INPUT	Vorher eingegebene Anweisungen oder Kommandos am Bildschirm wiederanzeigen
SHOW-INPUT-DEFAULTS	Taskspezifische Default-Werte anzeigen
SHOW-INPUT-HISTORY	Inhalt des Eingabepuffers anzeigen



---

SHOW-SDF-OPTIONS	Taskspezifische Information über aktivierte Syntaxdateien und die Festlegungen für die Anweisungseingabe und -verarbeitung anzeigen
SHOW-STMT	Syntax einer Anweisung anzeigen
STEP	Wiederaufsetzpunkt in einer Folge von SDF-PAR-Anweisungen definieren
WRITE-TEXT	Einen bestimmten Text nach SYSOUT oder SYSLST ausgeben

## 7.1.2 Beschreibung der Anweisungen

### MODIFY-SYNTAX-FILE

#### Einträge für Syntaxdateien ändern

Mit der Anweisung MODIFY-SYNTAX-FILE können Sie Syntaxdatei-Namen in die Parameterdatei eintragen oder bestehende Syntaxdatei-Einträge ändern oder entfernen. Folgende Einträge sind möglich:

- **Systemsyntaxdatei:**  
Ein Eintrag zur Basis-Systemsyntaxdatei muss in jeder Parameterdatei vorhanden sein. Der Eintrag kann geändert, aber nicht entfernt werden. Die Basis-Systemsyntaxdatei enthält die Syntaxbeschreibung aller Systemkommandos und gilt für alle aktuell angemeldeten Benutzeraufträge.
- **Subsystem-Syntaxdateien:**  
Einträge zu Subsystem-Syntaxdateien sind optional. Subsystem-Syntaxdateien sind Systemsyntaxdateien, die zusätzlich zur Basis-Systemsyntaxdatei aktiviert werden. Eine solche Systemsyntaxdatei muss nicht unbedingt zu einem von DSSM aktivierten Subsystem gehören.
- **Gruppensyntaxdateien:**  
Einträge zu Gruppensyntaxdateien sind optional. Eine Gruppensyntaxdatei enthält die Syntaxbeschreibung aller Kommandos, Programme und Anweisungen, die nur für eine von der Systemverwaltung definierte Gruppe von Benutzern verfügbar sind. Die Gruppensyntaxdateien werden über eine PROFILE-ID zugewiesen.

#### MODIFY-SYNTAX-FILE

**TYPE** = \*SYSTEM / \*SUBSYSTEM(...) / \*GROUP(...)

\*SUBSYSTEM(...)

    |   **SUBSYSTEM-NAME** = <structured-name 1..8>

\*GROUP(...)

    |   **PROFILE-ID** = <structured-name 1..30>

    |   **,HIERARCHY** = \*YES / \*NO

**,NAME** = \*UNCHANGED / \*NONE / <filename 1..54 without-gen-vers>

**TYPE =**

Legt fest, für welchen Syntaxdatei-Typ der Eintrag geändert wird.

**TYPE = \*SYSTEM**

Der Eintrag für die Basis-Systemsyntaxdatei wird geändert.

**TYPE = \*SUBSYSTEM(...)**

Der Eintrag für eine Subsystem-Syntaxdatei wird geändert.

**SUBSYSTEM-NAME = <structured-name 1..8>**

Name des Subsystems, zu dem die Subsystem-Syntaxdatei gehört.

**TYPE = \*GROUP(...)**

Der Eintrag für eine Gruppensyntaxdatei wird geändert.

**PROFILE-ID = <structured-name 1..30>**

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet war oder werden soll.

**HIERARCHY =**

Bestimmt, ob für die Syntaxanalyse der Kommandos/Anweisungen eines Benutzerauftrags mit der festgelegten PROFILE-ID die SDF-Dateihierarchie bestehen bleibt, d.h. ob die Systemsyntaxdateien zur Syntaxanalyse herangezogen werden.

**HIERARCHY = \*YES**

Die Systemsyntaxdateien werden beim Einrichten des Benutzerauftrags standardmäßig aktiviert. Die SDF-Dateihierarchie bleibt bestehen.

**HIERARCHY = \*NO**

Die Systemsyntaxdateien werden direkt nach der LOGON-Verarbeitung deaktiviert. Für Benutzer mit der festgelegten PROFILE-ID sind die Definitionen in den Systemsyntaxdateien somit ohne Bedeutung. Es gelten ausschließlich die in der zugeordneten Gruppensyntaxdatei hinterlegten Definitionen. Eine Gruppensyntaxdatei, die mit HIERARCHY=\*NO definiert ist, **muss** außer der Globalinformation auch mindestens das LOGOFF-Kommando enthalten. Nur so ist es möglich, einen Benutzerauftrag, dem die festgelegte PROFILE-ID zugeordnet ist, zu beenden.

**NAME =**

Bestimmt den Syntaxdatei-Namen. Die angegebene Syntaxdatei muss von dem bei TYPE festgelegten Typ sein.

**NAME = \*UNCHANGED**

Der Eintrag wird nicht verändert.

**NAME = \*NONE**

Der Syntaxdatei-Eintrag wird aus der Parameterdatei gelöscht. Der Eintrag für die Systemsyntaxdatei kann nicht gelöscht werden.

**NAME = <filename 1..54 without-gen-vers>**

Bestimmt den Syntaxdatei-Namen.

## MODIFY-SYSTEM-LOGON-PROCEDURE

### Eintrag für systemweite LOGON-Call-Prozedur ändern

Mit der Anweisung MODIFY-SYSTEM-LOGON-PROCEDURE können Sie den Namen einer systemweiten LOGON-Call-Prozedur in die geöffnete Parameterdatei eintragen. Diese Prozedur wird bei jedem LOGON-Vorgang mit dem Kommando CALL-PROCEDURE aufgerufen und läuft automatisch ab.

<b>MODIFY-SYSTEM-LOGON-PROCEDURE</b>
<b>NAME</b> = * <u>UNCHANGED</u> / *NONE / *STD / <filename 1..54 without-gen-vers>

**NAME = \*UNCHANGED / \*NONE / \*STD / <filename 1..54 without-gen-vers>**

Gibt an, ob eine systemweite LOGON-Call-Prozedur in die Parameterdatei eingetragen wird. Dieser Eintrag ist optional.

**NAME = \*NONE**

Der Eintrag für eine systemweite LOGON-Call-Prozedur wird aus der Parameterdatei gelöscht.

**NAME = \*STD**

Der Standardname \$.SYS.SDF.LOGON.SYSPROC wird eingetragen.

**NAME = <filename 1..54 without-gen-vers>**

Name der neuen systemweiten LOGON-Call-Prozedur.

## MODIFY-SYSTEM-LOGON-INCLUDE

### Eintrag für systemweite LOGON-Include-Prozedur ändern

Mit der Anweisung MODIFY-SYSTEM-LOGON-INCLUDE können Sie den Namen einer systemweiten LOGON-Include-Prozedur in die geöffnete Parameterdatei eintragen. Diese Prozedur wird bei jedem LOGON-Vorgang mit dem Kommando INCLUDE-PROCEDURE aufgerufen und läuft automatisch ab.

<b>MODIFY-SYSTEM-LOGON-INCLUDE</b>
<b>NAME</b> = <u>*UNCHANGED</u> / *NONE / *STD / <filename 1..54 without-gen-vers>

**NAME = \*UNCHANGED / \*NONE / \*STD / <filename 1..54 without-gen-vers>**

Gibt an, ob eine systemweite LOGON-Include-Prozedur in die Parameterdatei eingetragen wird. Dieser Eintrag ist optional.

**NAME = \*NONE**

Der Eintrag für eine systemweite LOGON-Include-Prozedur wird aus der Parameterdatei gelöscht.

**NAME = \*STD**

Der Standardname \$.SYS.SDF.LOGON.SYSINCL wird eingetragen.

**NAME = <filename 1..54 without-gen-vers>**

Name der neuen systemweiten LOGON-Include-Prozedur.

## MODIFY-VERSION

### Format der Parameterdatei ändern

Mit der Anweisung MODIFY-VERSION können Sie das Format der geöffneten Parameterdatei ändern. Das Format einer Parameterdatei wird beim Erzeugen mit der Anweisung OPEN-PARAMETER-FILE festgelegt. Zwei Formate sind möglich:

- Format \*V1:  
Die Parameterdatei kann in SDF < V1.4A eingesetzt werden.
- Format \*V2:  
Die Parameterdatei kann in SDF ab V1.4A eingesetzt werden.

<b>MODIFY-VERSION</b>
<b>VERSION</b> = * <u>UNCHANGED</u> / *V1 / *V2

#### **VERSION = \*UNCHANGED / \*V1 / \*V2**

Legt das Format der Parameterdatei fest.

#### **VERSION = \*V1**

Die Parameterdatei wird in das \*V1-Format konvertiert. Sie ist danach in SDF < V1.4A einsetzbar.

Die Satzlänge der Parameterdatei wird bei der Konvertierung von \*V2-Format nach \*V1-Format auf 54 Byte verkürzt. Die abgeschnittenen Informationen haben für SDF < V1.4A keine Bedeutung. Zur Information wird darüber eine Meldung ausgegeben. Jedes Objekt der Parameterdatei wird rekonstruiert und wieder in die geöffnete Parameterdatei eingefügt. Für jedes Objekt wird eine Meldung ausgegeben.

#### **VERSION = \*V2**

Die Parameterdatei wird in das \*V2-Format konvertiert. Sie ist danach ab SDF V1.4A einsetzbar.

Die Satzlänge der Parameterdatei wird bei der Konvertierung von \*V1-Format nach \*V2-Format auf 67 Byte verlängert. Die zusätzlichen Bytes werden mit X'00' aufgefüllt. Die Sätze für die Gruppensyntaxdatei-Einträge werden mit 'YES' und 10mal X'00' erweitert. Jedes Objekt der Parameterdatei wird rekonstruiert und wieder in die geöffnete Parameterdatei eingefügt. Für jedes Objekt wird eine Meldung ausgegeben.

## OPEN-PARAMETER-FILE

### Parameterdatei erzeugen oder öffnen

Mit der Anweisung OPEN-PARAMETER-FILE öffnen Sie eine bereits vorhandene oder erzeugen eine neue Parameterdatei. Beim Erzeugen der Parameterdatei legen Sie auch das Format fest. Zwei Formate sind möglich:

- Format \*V1:  
Die Parameterdatei kann in SDF < V1.4A eingesetzt werden.
- Format \*V2:  
Die Parameterdatei kann in SDF ab V1.4A eingesetzt werden.

Beim Erzeugen der Parameterdatei können auch schon Syntaxdateinamen eingetragen werden. Standardmäßig wird eine Parameterdatei im \*V2-Format erzeugt, in der als Name für die Basis-Systemsyntaxdatei \$.SYS.SDF.SYSTEM.SYNTAX eingetragen ist.

#### OPEN-PARAMETER-FILE

```

NAME = <filename 1..54 without-gen-vers>
,MODE = *READ / *UPDATE / *CREATE(...)
      *CREATE(...)
          |
          | VERSION = *V2 / *V1
          |
          | ,SYSTEM-SYNTAX-FILE = *STD / <filename 1..54 without-gen-vers>
          |
          | ,GROUP-SYNTAX-FILE = *NONE / *STD(...) / <filename 1..54 without-gen-vers>(…)
          |
          | *STD(...)
          |   |
          |   | PROFILE-ID = SYS-TSOS / <structured-name 1..30>
          |   |
          |   | ,HIERARCHY = *YES / *NO
          |   |
          |   | <filename 1..54 without-gen-vers>(…)
          |   |
          |   | PROFILE-ID = SYS-TSOS / <structured-name 1..30>
          |   |
          |   | ,HIERARCHY = *YES / *NO

```

**NAME = <filename 1..54 without-gen-vers>**

Name der zu öffnenden oder zu erzeugenden Parameterdatei.

**MODE =**

Legt fest, ob die Parameterdatei nur zum Lesen oder zum Ändern geöffnet wird oder ob sie erzeugt werden soll.

**MODE = \*READ**

Die bereits vorhandene Parameterdatei wird nur zum Lesen geöffnet.

**MODE = \*UPDATE**

Die bereits vorhandene Parameterdatei wird im Lese-/Schreibmodus geöffnet.

**MODE = \*CREATE(...)**

Die Parameterdatei wird erzeugt und im Lese-/Schreibmodus geöffnet. Existiert eine Parameterdatei mit dem angegebenen Namen bereits, so wird die Anweisung abgewiesen.

**VERSION =**

Legt das Format der Parameterdatei fest.

**VERSION = \*V2**

Die Parameterdatei wird im \*V2-Format erzeugt und ist in SDF ab V1.4A einsetzbar.

**VERSION = \*V1**

Die Parameterdatei wird im \*V1-Format erzeugt und ist in SDF < V1.4A einsetzbar.

**SYSTEM-SYNTAX-FILE =**

Legt den Eintrag für die Basis-Systemsyntaxdatei fest.

**SYSTEM-SYNTAX-FILE = \*STD**

Als Name für die Basis-Systemsyntaxdatei wird \$.SYS.SDF.SYSTEM.SYNTAX in die Parameterdatei eingetragen.

**SYSTEM-SYNTAX-FILE = <filename 1..54 without-gen-vers>**

Name der Basis-Systemsyntaxdatei, der in die Parameterdatei eingetragen wird.

**GROUP-SYNTAX-FILE =**

Der Gruppensyntaxdatei-Eintrag für die Default-User-ID wird festgelegt.

**GROUP-SYNTAX-FILE = \*NONE**

Für die Default-User-ID wird keine Gruppensyntaxdatei eingetragen.

**GROUP-SYNTAX-FILE = \*STD**

Für die Default-User-ID wird die Gruppensyntaxdatei \$.SYS.SDF.GROUP.SYNTAX eingetragen.

**PROFILE-ID = SYS-TSOS / <structured-name 1..30>**

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet wird. Default-Wert ist SYS-TSOS, da SDF für die User-ID TSOS auch diesen Wert verwendet.

**HIERARCHY =**

Bestimmt, ob für die Syntaxanalyse der Kommandos/Anweisungen eines Benutzerauftrags mit der festgelegten PROFILE-ID die SDF-Dateihierarchie bestehen bleibt, d.h. ob die Systemsyntaxdateien zur Syntaxanalyse herangezogen werden.

**HIERARCHY = \*YES**

Die Systemsyntaxdateien werden beim Einrichten des Benutzerauftrags standardmäßig aktiviert. Die SDF-Dateihierarchie bleibt bestehen.



**HIERARCHY = \*NO**

Die Systemsyntaxdateien werden direkt nach der LOGON-Verarbeitung deaktiviert. Für Benutzer mit der festgelegten PROFILE-ID sind die Definitionen in den Systemsyntaxdateien somit ohne Bedeutung. Es gelten ausschließlich die in der zugeordneten Gruppensyntaxdatei hinterlegten Definitionen. Eine Gruppensyntaxdatei, die mit HIERARCHY=\*NO definiert ist, **muss** außer der Globalinformation auch mindestens das Kommando EXIT-JOB (bzw. LOGOFF) enthalten. Nur so ist es möglich, einen Benutzerauftrag, dem die festgelegte PROFILE-ID zugeordnet ist, zu beenden.

**GROUP-SYNTAX-FILE = <filename 1..54 without-gen-vers>**

Für die Default-User-ID wird die Gruppensyntaxdatei mit dem angegebenen Namen eingetragen.

**PROFILE-ID = SYS-TSOS / <structured-name 1..30>**

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet wird. Default-Wert ist SYS-TSOS, da SDF für die User-ID TSOS auch diesen Wert verwendet.

**HIERARCHY =**

Bestimmt, ob für die Syntaxanalyse der Kommandos/Anweisungen eines Benutzerauftrags mit der festgelegten PROFILE-ID die SDF-Dateihierarchie bestehen bleibt, d.h. ob die Systemsyntaxdateien zur Syntaxanalyse herangezogen werden.

**HIERARCHY = \*YES**

Die Systemsyntaxdateien werden beim Einrichten des Benutzerauftrags standardmäßig aktiviert. Die SDF-Dateihierarchie bleibt bestehen.

**HIERARCHY = \*NO**

Die Systemsyntaxdateien werden direkt nach der LOGON-Verarbeitung deaktiviert. Für Benutzer mit der festgelegten PROFILE-ID sind die Definitionen in den Systemsyntaxdateien somit ohne Bedeutung. Es gelten ausschließlich die in der zugeordneten Gruppensyntaxdatei hinterlegten Definitionen. Eine Gruppensyntaxdatei, die mit HIERARCHY=\*NO definiert ist, **muss** außer der Globalinformation auch mindestens das Kommando EXIT-JOB (bzw. LOGOFF) enthalten. Nur so ist es möglich, einen Benutzerauftrag, dem die festgelegte PROFILE-ID zugeordnet ist, zu beenden.

## SHOW-PARAMETER-FILE

### Parameterdatei anzeigen

Die Anweisung SHOW-PARAMETER-FILE zeigt den Inhalt der aktuell geöffneten Parameterdatei an. Folgende Einträge werden ausgegeben, sofern sie vorhanden sind:

- Versionsformat der Parameterdatei
- Name der eingetragenen systemweiten LOGON-Prozedur
- Name der eingetragenen Basis-Systemsyntaxdatei
- zu jedem eingetragenen Subsystem wird der Name der Subsystem-Syntaxdatei ausgegeben
- zu jeder eingetragenen PROFILE-ID wird der Name der zugeordneten Gruppensyntaxdatei ausgegeben. Für Parameterdateien im \*V2-Format wird auch die Hierarchie-Einstellung ausgegeben.

SHOW-PARAMETER-FILE

Diese Anweisung hat keine Operanden.

## 7.2 Installation von SDF-PAR

Folgende Dateien werden u.a. mit SDF-PAR V1.1 ausgeliefert:

- SYSPRG.SDF-PAR.011  
Programm SDF-PAR
- SYSMES.SDF-PAR.011  
Meldungsdatei für SDF-PAR ab BS2000/OSD-BC V2.0
- SYSPRC.SDF-PAR.011.112  
Start-Prozedur, die die Ablaufumgebung für SDF-PAR vorbereitet und SDF-PAR startet
- SYSSDF.SDF-PAR.011.USER  
Benutzersyntaxdatei, die die SDF-PAR-Anweisungen enthält
- SYSSII.SDF-PAR.011  
Installations-Informationsdatei für die Installation mit IMON

Vor dem Starten von SDF-PAR muss die SDF-PAR-Meldungsdatei zugewiesen und die SDF-PAR-Benutzersyntaxdatei aktiviert werden. Alle notwendigen Vorbereitungen für den Programmlauf werden auch von der mitgelieferten SYSPRC-Prozedur ausgeführt.

Da für SDF-PAR kein START-Kommando existiert, muss die SYSPRG-Datei mit dem Kommando START-PROGRAM bzw. START-EXECUTABLE-PROGRAM (ab BLSSERV V2.3) gestartet werden.

Detaillierte Informationen zur Installation von SDF-PAR und zu Hard- und Softwarevoraussetzungen entnehmen Sie bitte der Freigabemittteilung zum Softwareprodukt SDF-PAR.



---

## 8 Verhalten im Fehlerfall

### **SDF-Parameterdatei kann beim Laden nicht geöffnet werden**

Kann die Parameterdatei beim Laden von SDF nicht geöffnet werden, fordert SDF über eine Konsolmeldung eine neue Parameterdatei an. Bei der Antwort „\*STD“ aktiviert SDF die Datei \$TSOS.SYSSDF.SDF.045 als Basis-Systemsyntaxdatei und die Datei \$TSOS.SYSSDF.BS2CP<bs2vers> als Subsystem-Syntaxdatei. Nach dem Laden von SDF kann die Parameterdatei gelöscht und eine neue Parameterdatei mit MODIFY-SDF-PARAMETERS oder dem Dienstprogramm SDF-PAR erstellt werden.

### **Basis-Systemsyntaxdatei kann beim Laden nicht aktiviert werden**

Wenn die Basis-Systemsyntaxdatei beim Laden nicht aktiviert werden kann, wird eine Fehlermeldung ausgegeben und eine neue Systemsyntaxdatei wird angefordert. Gibt der Operator dann eine gültige Basis-Systemsyntaxdatei an, so wird diese aktiviert, ihr Name jedoch nicht in die Parameterdatei eingetragen. Wird keine gültige Systemsyntaxdatei angegeben, wird der Ladevorgang abgebrochen. In diesem Fall muss das System vom Backup-Pubset geladen werden. Steht ein solches nicht zur Verfügung, so ist ein Restaurieren der Platten mit dem Dienstprogramm FDDRL (siehe Handbuch „FDDRL“ [12]) oder eventuell eine Systemneugenerierung erforderlich.

### **Systemsyntaxdatei nicht mit USER-ACCESS=\*SPECIAL katalogisiert**

Wenn eine Systemsyntaxdatei zwar mehrbenutzbar, aber nicht mit USER-ACCESS=\*SPECIAL katalogisiert ist, wird eine Fehlermeldung ausgegeben und das Attribut USER-ACCESS auf \*SPECIAL gesetzt. Wenn die Systemsyntaxdatei nicht mehrbenutzbar ist, muss der Operator diese Attribut-Änderung bestätigen. Tut er das nicht, dann kann die Systemsyntaxdatei nicht aktiviert werden.

### **Basis-Systemsyntaxdatei ohne Kommando MODIFY-SDF-PARAMETERS**

Wenn die Basis-System-Syntaxdatei nicht das Kommando MODIFY-SDF-PARAMETERS enthält, wird eine Fehlermeldung ausgegeben. Der Operator kann in diesem Fall den Ladevorgang fortsetzen, abbrechen oder den Namen einer anderen Basis-Systemsyntaxdatei eingeben.



---

# Literatur

Wenden Sie sich zum Bestellen von Handbüchern bitte an Ihre zuständige Geschäftsstelle.

- [1] **SDF V4.5A (BS2000/OSD)**  
**Einführung in die Dialogschnittstelle SDF**  
Benutzerhandbuch

*Zielgruppe*

BS2000/OSD-Anwender

*Inhalt*

Das Handbuch beschreibt die Dialog-Eingabe von Kommandos und Anweisungen im SDF-Format. Ein Schnelleinstieg mit leicht nachvollziehbaren Beispielen und weitere umfangreiche Beispiele erleichtern die Anwendung. SDF-Syntaxdateien werden erklärt.

*Bestellnummer*

U2339-J-Z125-8

- [2] **BS2000/OSD-BC V5.0**  
**Einführung in die Systembetreuung**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.

*Inhalt*

Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbaus enthalten: Systemeinleitung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei-, Pubset- und Systemzeit-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

*Bestellnummer*

U2417-J-Z125-14

- [3] **BS2000/OSD-BC V5.0**  
**Kommandos Band 1 - 5**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Die Bände 1 bis 5 enthalten die Kommandos ADD-... bis WRITE-... (BS2000/OSD-Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien. Die Kommando- und Operandenfunktionen werden ausführlich beschrieben; viele Beispiele unterstützen das Verständnis. Am Anfang jedes Bandes informiert eine Übersicht über alle in den Bänden 1-5 beschriebenen Kommandos.

Der Anhang von Band 1 enthält u.a. Informationen zur Kommandoeingabe, zu bedingten Jobvariablenausdrücken, Systemdateien, Auftragschaltern, Geräte- und Volumetypen. Der Anhang der Bände 4 und 5 enthält jeweils eine Übersicht zu den Ausgabespalten der SHOW-Kommandos der Komponente NDM. Der Anhang von Band 5 enthält zusätzlich eine Übersicht aller START-Kommandos.

In jedem Band ist ein umfangreiches Stichwortverzeichnis mit allen Stichwörtern der Bände 1-5 enthalten.

*Bestellnummern*

U2338-J-Z125-15 Kommandos Band 1, A – C  
U41074-J-Z125-2 Kommandos Band 2, D – MOD-I  
U21070-J-Z125-5 Kommandos Band 3, MOD-J – R  
U41075-J-Z125-2 Kommandos Band 4, S – SH-O  
U23164-J-Z125-4 Kommandos Band 5, SH-P – Z

- [4] **BS2000/OSD-BC V5.0**  
**Kommandos Band 6, Ausgabe in S-Variablen und SDF-P-BASYS**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Programmierer und Anwender, die Prozeduren erstellen.

*Inhalt*

Band 6 enthält die tabellarische Darstellung aller S-Variablen, die von den SHOW-Kommandos bei einer strukturierten Ausgabe mit Werten versorgt werden. Weitere Kapitel:

- Einführung in das Arbeiten mit S-Variablen
- SDF-P-BASYS V2.2

*Bestellnummer*

U23165-J-Z125-4



- [5] **BS2000/OSD-BC V5.0**  
**Systeminstallation**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die BS2000/OSD-Systemverwaltung.

*Inhalt*

Beschrieben wird die Generierung der Hardware-Konfiguration mit UGEN und die Installationsdienste. Letztere beinhalten die Plattenorganisation mit MPVS, die Installation von Datenträgern mit dem Dienstprogramm SIR und das Subsystem IOCFCOPY.

*Bestellnummer*

U2505-J-Z125-15

- [6] **IMON V2.5 (BS2000/OSD)**  
**Installationsmonitor**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung des Betriebssystems BS2000/OSD.

*Inhalt*

Das Handbuch beschreibt die Installation und Verwaltung von BS2000-Software mit dem Installationsmonitor IMON und seinen drei Komponenten IMON-BAS, IMON-GPN und IMON-SIC.

In zwei Beispielkapiteln wird die Installation (standard und kundenspezifisch) mit der Komponente IMON-BAS für Systeme mit BS2000/OSD V2.0 und ab BS2000/OSD V3.0/V4.0 ausführlich dargestellt.

*Bestellnummer*

U21926-J-Z125-3

- [7] **SDF-A V4.1E (BS2000/OSD)**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an erfahrene BS2000-Benutzer und an die Systembetreuung.

*Inhalt*

Es beschreibt, wie Syntaxdateien bearbeitet werden und erklärt die SDF-A-Funktionen anhand von Beispielen. Die SDF-A-Anweisungen sind in alphabetischer Reihenfolge aufgeführt.

Über die im Handbuch beschriebene SDF-Programmschnittstelle können auch Benutzerprogramme die SDF-Benutzeroberfläche nutzen. Mit dem Dienstprogramm SDF-SIM kann die Syntax von Kommandos und Anweisungen getestet werden.

*Bestellnummer*

U2284-J-Z125-9

- [8] **SDF-A (BS2000/OSD)**  
Tabellenheft

*Zielgruppe*

Das Tabellenheft wendet sich an geübte BS2000-Benutzer mit Erfahrung bei der Anwendung von SDF-A.

*Inhalt*

Es enthält die SDF-A-Anweisungen in alphabetischer Reihenfolge. Daran anschließend sind die Makros und Funktionsaufrufe der SDF-Programmschnittstelle, die Formate des Übergabebereichs und die SDF-SIM-Anweisungen aufgeführt.

- [9] **BS2000/OSD-BC V5.0**  
**Makroaufrufe an den Ablaufteil**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an alle BS2000/OSD-Assembler-Programmierer.

*Inhalt*

Das Handbuch enthält eine Zusammenstellung der Makroaufrufe an den Ablaufteil:

- Binden und Laden
- virtueller Speicher, Memory Pool, ESA
- Task- und Programmsteuerung
- ITC, Serialisierung, Eventing, DLM, Contingencys, STXIT
- Meldungswesen, Accounting, JMS, TIAM, VTSU, ...

Ausführliche Beschreibung aller Makroaufrufe in lexikalischer Reihenfolge mit Beispielen. Allgemeiner Lernteil über ITC, Serialisierung, Eventing, DLM, Contingencies, STXIT, virtueller Speicher, Memory Pool, ESA, ...

*Bestellnummer*

U3291-J-Z125-9

- [10] **BS2000/OSD-BC V5.0**  
**Dienstprogramme**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Es beschreibt die Dienstprogramme DPAGE V14.0A, INIT V14.0A, JMP V2.0A, JMU V14.0A, LMSCONV V3.3B, PAMCONV V13.0A, PASSWORD V14.0A, PVSREN V1.4A, RMS V7.1A, SCDM V5.0A, SMPGEN V14.0A, SPCCNTRL V14.0A, TPCOMP2 V14.0A, VOLIN V14.0A.

*Bestellnummer*

U4303-J-Z125-7

- [11] **MSGMAKER V1.2A** (BS2000/OSD)  
**Erstellen und Bearbeiten von BS2000 Meldungsdateien**  
 Benutzerhandbuch
- Zielgruppe*  
 Das Handbuch wendet sich an die Systembetreuung und an den nichtprivilegierten Anwender.
- Inhalt*  
 Das Handbuch beschreibt das Dienstprogramm MSGMAKER, mit dem BS2000-Meldungsdateien erstellt und bearbeitet werden können. Beschrieben sind die Bedienung (Masken und Anweisungen) von MSGMAKER, sowie wichtige Abläufe des BS2000.
- Bestellnummer*  
 U23715-J-Z125-2
- [12] **FDDRL V14.0A** (BS2000/OSD V5.0)  
 Benutzerhandbuch
- Zielgruppe*  
 BS2000/OSD-Systembetreuer und -Operateur
- Inhalt*  
 FDDRL sichert und restauriert physikalisch den Inhalt ganzer Platten und Pubsets. Das Handbuch beschreibt die Funktionen und Anweisungen des Programms FDDRL zur physikalischen Datensicherung im Rechenzentrum.
- Bestellnummer*  
 U3251-J-Z125-7
- [13] **SECOS V4.0** (BS2000/OSD)  
**Security Control System**  
 Benutzerhandbuch
- Zielgruppe*
- BS2000-Systemverwalter
  - BS2000-Anwender, die den erweiterten Zugriffsschutz für Dateien nutzen
- Inhalt*  
 Leistung und Anwendung der Funktionseinheiten:
- SRPM (Privilegien und Betriebsmittel verwalten)
  - SRPMSSO (Single Sign On)
  - GUARDS (Zugriffsbedingungsverwaltung und -auswertung für Objekte)
  - GUARDEF (Default Protection, Standardschutz)
  - GUARDCOO (Co-owner Protection, Miteigentümerschutz)
  - SAT (Protokollierung und Auswertung sicherheitsrelevanter Daten, Ereignisüberwachung mit Alarmfunktion).
- Bestellnummer*  
 U5605-J-Z125-6

- [14] **SDF-P V2.2A (BS2000/OSD)**  
**Programmieren in der Kommandosprache**  
Benutzerhandbuch

*Zielgruppe*

BS2000-Anwender und Systembetreuung.

*Inhalt*

SDF-P ist eine strukturierte Prozedursprache im BS2000. Nach einführenden Kapiteln zum Prozedur- und Variablenkonzept werden Kommandos, Funktionen und Makros ausführlich beschrieben.

Inhaltlicher Überblick:

- Schnelleinstieg SDF-P
- Prozedurkonzept von SDF-P
- S-Prozeduren erstellen, testen, aufrufen, steuern
- S-Variablen, S-Variablenströme, Funktionen, Ausdrücke
- Nicht-S-Prozeduren umstellen
- Makros, Builtin-Funktionen, SDF-P-Kommandos

Für den Einsatz von SDF-P V2.2A werden SDF-P-BASYS  $\geq$  V2.1A, VAS  $\geq$  V2.0A und SDF  $\geq$  V4.1A vorausgesetzt.

*Bestellnummer*

U6442-J-Z125-5

- [15] **XHCS V1.3 (BS2000/OSD)**  
**8-bit-Code-Verarbeitung im BS2000/OSD**  
Benutzerhandbuch

*Zielgruppe*

Anwender der Zugriffsmethoden DCAM, TIAM und *openUTM* sowie Systembetreuer; Anwender, die von EHCS auf XHCS umstellen.

*Inhalt*

XHCS (Extended Host Code Support) ist ein Softwareprodukt des BS2000/OSD. Es ermöglicht Ihnen, erweiterte Zeichensätze bei 8-bit-Datenstationen zu nutzen. XHCS ist die zentrale Informationsquelle über die codierten Zeichensätze im BS2000/OSD. XHCS löst EHCS ab.

*Bestellnummer*

U9232-J-Z135-4

*Erweiterungen siehe Readme-Datei zu XHCS V1.4*

- [16] **BS2000/OSD-BC V5.0**  
**Verwaltung von Subsystemen (DSSM/SSCM)**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und die Softwareberatung des BS2000/OSD.

*Inhalt*

Es werden das Subsystemkonzept des BS2000/OSD, die Dynamische Subsystemverwaltung DSSM V4.0 und die Subsystemkatalog-Verwaltung SSCM V2.3 mit den dazugehörigen Kommandos und Anweisungen beschrieben.

DSSM bietet die Möglichkeit, benutzereigene Subsystem-Konfigurationen tasklokal zu erstellen und zu verwalten.

*Bestellnummer*

U23166-J-Z125-3

- [17] **BS2000/OSD**  
**Softbooks Deutsch**  
CD-ROM

*Zielgruppe*

BS2000/OSD-Anwender

*Inhalt*

Auf der CD-ROM „BS2000/OSD SoftBooks Deutsch“ sind nahezu alle deutschen Handbücher und Readme-Dateien zur BS2000-Systemsoftware der aktuellsten BS2000/OSD-Version und auch von Vorgängerversionen gespeichert, incl. der aufgeführten Handbücher. Diese Softbooks finden Sie auch im Internet auf unserem Manual Server. Sie können in den Handbüchern nachschlagen oder sich vollständige Handbücher herunterladen.

*Bestellnummer der CD-ROM*

U26175-J8-Z125-1



---

# Stichwörter

## A

- Abkürzungsmöglichkeiten bei SDF 4
- Ablauf von SDF-I 76
- aktivieren
  - Gruppensyntaxdatei 32
  - SDF 25
  - Systemsyntaxdatei 31
- aktuelles Objekt 104
- Aliasname 9
- alphanum-name (Datentyp) 10
- Ausgabe in S-Variable, Metasyntax 23
- ausgeben
  - SDF-Optionen 63
  - Syntaxdateiname 124

## B

- Basis-Systemsyntaxdatei 26
  - nicht aktivierbar 141
  - wechseln 31
- beenden
  - SDF-I 83
  - SDF-PAR 128
  - SDF-U 106
- Benutzerauftrag, Festlegungen ändern 42
- benutzerspezifische LOGON-Prozeduren 34
- Benutzersyntaxdatei 26
  - aktivieren 42
  - deaktivieren 42, 44
  - für SDF-PAR 127

## C

- cat (Zusatz zu Datentypen) 21
- cat-id (Datentyp) 10
- CCS (Coded Character Set) 5
- command-rest (Datentyp) 10

- compl (Zusatz zu Datentypen) 16
- composed-name (Datentyp) 10
- COPY (Anweisung, SDF-U) 101
- corr (Zusatz zu Datentypen) 21, 22
- c-string (Datentyp) 10

## D

- date (Datentyp) 10
- Dateihierarchie 57, 131, 136, 137
- Datentypen SDF 6, 10
  - Zusätze 7
- device (Datentyp) 10
- Dialogführung steuern 44
- Dienstprogramm
  - SDF-I 27
  - SDF-PAR 27, 30, 127
  - SDF-U 27
- DSSM 39

## E

- EDIT (Anweisung, SDF-U) 104
- einstellen, SDF-Optionen 42
- END (Anweisung, SDF-I) 83
- END (Anweisung, SDF-PAR) 128
- END (Anweisung, SDF-U) 106
- END (SDF-PAR) 128
- Expertenmodus 45

## F

- Fehlerdialog bei Syntaxfehler 46
- Fehlerfall (Verhalten von SDF) 141
- filename (Datentyp) 11
- fixed (Datentyp) 10
- full-filename siehe Datentyp filename 11
- Funktionstasten-Belegung festlegen 42

## G

- garantierte Abkürzungen 4
- gen (Zusatz zu Datentypen) 21
- generieren, Subsystemkatalog 39
- Globalinformation ändern 112
- Gruppensyntaxdatei 26, 32
  - automatisch aktivieren 32
  - dauerhaft zuweisen 55
  - für nächste Session zuweisen 56
  - temporär zuweisen 54

## I

- Index
  - global 19
  - Konstruktionszeichenfolge 19
  - platzhalter-spezifisch 19
  - Schreibweise 20
- Informationen, taskspezifische 63
- Inhalt von Syntaxdateien 25
- INSD 26
- Installation
  - SDF 37
  - SDF-PAR 127, 139
- Installationsdateien 37
- Installationssyntaxdatei 26
- integer (Datentyp) 12

## K

- Kommando
  - Definition ändern mit SDF-U 107
  - Protokollierung ändern 45
  - Returncodes 28
- Konstruktionsangabe 20
- Konstruktionszeichenfolge 19
- Kurzname 4, 9

## L

- laden, SDF 39
- LOGOFF-Prozedur
  - Aufrufreihenfolge 35
  - benutzerspezifische 34
  - Einschränkungen 35
  - systemweite 34

## LOGON-Prozedur

- Aufrufreihenfolge 35
- benutzerspezifische 34
- Einschränkungen 35
- systemweite 34
- löschen, Objekte der Syntaxdatei 110
- low (Zusatz zu Datentypen) 16

## M

- man (Zusatz zu Datentypen) 21, 22
- mandatory (Zusatz zu Datentypen) 22
- MERGE (SDF-I) 84
- Metasyntax
  - für SDF 6
  - für S-Variablen-Ausgabe 23
- MODIFY-CMD (Anweisung, SDF-U) 107
- MODIFY-SDF-OPTIONS (Kommando) 42
- MODIFY-SDF-PARAMETERS (Kommando) 53
- MODIFY-SYNTAX-FILE (Anweisung, SDF-PAR) 130
- MODIFY-SYSTEM-LOGON-INCLUDE (Anweisung, SDF-PAR) 133
- MODIFY-SYSTEM-LOGON-PROCEDURE (Anweisung, SDF-PAR) 132
- MODIFY-VALUE (Anweisung, SDF-U) 108
- MODIFY-VERSION (Anweisung, SDF-PAR) 134

## N

- Name
  - Benutzer-LOGON-Prozedur 34
  - Gruppensyntaxdatei 30
  - Systemsyntaxdatei 30
- name (Datentyp) 12

## O

- Objekt, aktuelles 104
- odd (Zusatz zu Datentypen) 21
- OPEN (Anweisung, SDF-I) 88
- OPEN-PARAMETER-FILE (Anweisung, SDF-PAR) 135
- OPEN-SYNTAX-FILE (Anweisung, SDF-U) 109
- Operandenwertdefinition ändern 108



**P**

Parameterdatei von SDF 141  
 anzeigen 138  
 Einträge ändern 130, 132, 133  
 erzeugen 30, 127, 135  
 Format ändern 134  
 modifizieren 127  
 öffnen 135  
 partial-filename (Datentyp) 13  
 path-compl (Zusatz zu Datentypen) 16  
 posix-filename (Datentyp) 13  
 posix-pathname (Datentyp) 13  
 POSIX-Platzhalter 17  
 Privilegien 26  
 product-version (Datentyp) 14  
 PROFILE-ID vereinbaren 32  
 Protokollierung ändern (für Kommando) 45  
 Prozedur mit Syntaxfehlerdialog 46  
 Prozedurdateien  
   benutzerspezifische 34  
   systemweite 34

**Q**

quotes (Zusatz zu Datentypen) 22

**R**

REMOVE (SDF-I) 91  
 REMOVE (SDF-U) 110  
 Returncodes der Kommandos 28

**S**

Schlüsselwortoperanden bei SDF 5  
 SDF  
   aktivieren 25  
   Dialogführung steuern 44  
   Fehlerdialog (Syntaxfehler) 46  
   Installation 37  
   laden 39  
   Parameter anzeigen 66  
   starten 39  
 SDF-I  
   Ablauf 76  
   Arbeitsdateien 78  
   Ausgabesyntaxdatei zuweisen 88

  beenden 83  
   Eingabesyntaxdatei zuweisen 88  
   Fortsetzungszeilen 79  
   Hinweise zu Eingaben 79  
   selbsterstellte Syntaxdateien einmischen 86  
   selbsterstellte Syntaxdateien entmischen 91  
   SESDs einmischen 84  
   SESDs entmischen 91  
   starten 76  
   Syntaxdateien entmischen 91  
   Syntaxdateien mischen 84  
   Syntaxdatei-Kompatibilität 79  
 SDF-Optionen  
   ausgeben 63  
   einstellen 42  
 SDF-PAR 127  
   beenden 128  
   Benutzersyntaxdatei 127  
   Dienstprogramm 27, 30  
   installieren 127, 139  
   starten 127  
 SDF-Parameterdatei 30, 127, 141  
 SDF-U 97  
   beenden 106  
   fortsetzen 98  
   Kommandodefinition ändern 107  
   Operandenwertdefinition ändern 108  
   starten 97  
   Syntaxdatei öffnen 109  
   Syntaxdatei-Objekte löschen 110  
   unterbrechen 98  
 sep (Zusatz zu Datentypen) 21  
 SESD 26  
 SET-GLOBALS (Anweisung, SDF-U) 112  
 SHOW (Anweisung, SDF-U) 118  
 SHOW-CORRECTION-INFORMATION (Anweisung, SDF-U) 123  
 SHOW-PARAMETER-FILE (SDF-PAR) 138  
 SHOW-SDF-OPTIONS (Kommando) 63  
 SHOW-SDF-PARAMETERS (Kommando) 66  
 SHOW-STATUS (Anweisung, SDF-U) 124  
 SHOW-SYNTAX-FILE (Anweisung, SDF-I) 93  
 SHOW-SYNTAX-VERSIONS (Kommando) 72

- Softwareeinheitssyntaxdateien 26
    - einmischen 84
    - entmischen 91
  - SSCM 39
  - Standardanweisungen von SDF 79, 99
  - starten
    - SDF-I 76
    - SDF-PAR 127
    - SDF-U 97
  - Stellungsoperanden bei SDF 5
  - STEP (Anweisung, SDF-U) 125
  - Stern vor konstantem Operandenwert 4
  - structured-name (Datentyp) 14
  - Subsystemkatalog generieren 39
  - Subsystem-Syntaxdatei 26, 31
    - dauerhaft aktivieren 55
    - für nächste Session aktivieren 56
  - S-Variablen, Metasyntax 23
  - Syntaxanalyse 57, 131, 136, 137
  - Syntaxdatei 25
    - Benutzersyntaxdatei 26
    - entmischen 75, 91
    - Gruppensyntaxdatei 26
    - Inhalt 25
    - Inhalte kopieren 101
    - mischen 75, 84
    - modifizieren 97
    - Name ausgeben 124
    - Objekte ausgeben 118
    - Objekte löschen 110
    - öffnen 109
    - positionieren auf ein Kommando 104
    - selbsterstellte einmischen 86
    - selbsterstellte entmischen 91
    - Systemsyntaxdatei 26
    - Typen 26
    - verwalten 27
    - wechseln 30
    - zuweisen 30
  - Syntaxprüfung
    - von Anweisungen 42
    - von Kommandos 42
  - Syntaxversionen anzeigen 72
  - Systemsyntaxdatei 26, 31
    - aktivieren 31, 57, 131, 136, 137
    - dauerhaft aktivieren 55
    - deaktivieren 57, 131, 137
    - für nächste Session aktivieren 56
    - Informationen ausgeben 93
    - nicht aktivierbar 141
    - temporär aktivieren 54
    - wechseln 31
  - systemweite LOGON-Prozeduren 34
- ## T
- taskspezifische Informationen 63
  - temp-file (Zusatz zu Datentypen) 21
  - text (Datentyp) 14
  - time (Datentyp) 14
  - Typen von Syntaxdateien 26
- ## U
- under (Zusatz zu Datentypen) 16
  - user (Zusatz zu Datentypen) 22
- ## V
- vereinbaren, PROFILE-ID 32
  - Verhalten von SDF im Fehlerfall 141
  - vers (Zusatz zu Datentypen) 22
  - vsn (Datentyp) 14
- ## W
- wechseln, Basis-Systemsyntaxdatei 31
  - Wiederaufsetzpunkt definieren 125
  - wild(n) (Zusatz zu Datentypen) 17
  - with (Zusatz zu Datentypen) 16
  - without (Zusatz zu Datentypen) 21
- ## X
- XHCS-Unterstützung 5
  - x-string (Datentyp) 15
  - x-text (Datentyp) 15
- ## Z
- Zielgruppe des Handbuchs 1
  - Zielsetzung des Handbuchs 1
  - Zusätze zu Datentypen 7, 16

---

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zielsetzung und Zielgruppen des Handbuchs	1
1.2	Konzept des Handbuchs	2
1.3	Änderungen gegenüber der vorigen Ausgabe	3
1.4	Hinweise zur Benutzeroberfläche	4
1.5	Verwendete Metasprache	6
1.5.1	Darstellungsmittel	6
1.5.2	SDF-Syntaxdarstellung	6
1.5.3	Metasyntax für die Beschreibung der S-Variablen-Ausgabe	23
<b>2</b>	<b>Arbeitsweise von SDF</b>	<b>25</b>
2.1	SDF-Parameter	30
2.2	Systemsyntaxdateien	31
2.2.1	Basis-Systemsyntaxdatei	31
2.2.2	Subsystem-Syntaxdateien	31
2.3	Gruppensyntaxdatei	32
2.4	LOGON- und LOGOFF-Prozeduren	34
<b>3</b>	<b>Installation von SDF</b>	<b>37</b>
3.1	SDF-Installationsdateien	37
3.2	Subsystemkatalog generieren	39
3.3	SDF bereitstellen	39
3.3.1	Dateien für SDF bereitstellen	39
3.3.2	SDF starten	39
<b>4</b>	<b>Kommandos zur SDF-Verwaltung</b>	<b>41</b>
4.1	Funktionelle Übersicht	41
4.2	Beschreibung der Kommandos	42
	MODIFY-SDF-OPTIONS	Benutzersyntaxdateien aktivieren und SDF-Optionen ändern
		42
	MODIFY-SDF-PARAMETERS	SDF-Parameter ändern
		53
	SHOW-SDF-OPTIONS	Aktive Syntaxdateien und SDF-Optionen anzeigen
		63
	SHOW-SDF-PARAMETERS	SDF-Parameter anzeigen
		66
	SHOW-SYNTAX-VERSIONS	Versionsstände von Syntaxdateien anzeigen
		72

<b>5</b>	<b>SDF-I</b> .....	<b>75</b>
5.1	Ein-/Ausgaben von SDF-I .....	75
5.2	Aufruf und Ablauf von SDF-I .....	76
5.3	Behandlung von Arbeitsdateien .....	78
5.4	SDF-Standardanweisungen und Anwendungsbereiche .....	79
5.5	Hinweise .....	79
5.6	SDF-I-Anweisungen .....	81
5.6.1	Funktionelle Übersicht .....	81
5.6.2	Beschreibung der Anweisungen .....	82
	CONVERT-SYNTAX-FILE      Syntaxdatei mit neuem Format versehen .....	82
	END                         SDF-I beenden .....	83
	MERGE                      Syntaxdateien mischen .....	84
	OPEN                       Eingabe- und Ausgabesyntaxdatei zuweisen .....	88
	REMOVE                    Syntaxdateien entmischen .....	91
	SHOW-SYNTAX-FILE        Informationen über eine Syntaxdatei anzeigen .....	93
<b>6</b>	<b>SDF-U</b> .....	<b>97</b>
6.1	SDF-U-Anweisungen .....	99
6.1.1	Funktionelle Übersicht .....	99
6.1.2	Beschreibung der Anweisungen .....	101
	COPY                       Inhalte einer Syntaxdatei kopieren .....	101
	EDIT                       Auf ein Kommando der Syntaxdatei positionieren ..	104
	END                        Programmlauf beenden .....	106
	MODIFY-CMD                Kommandodefinition ändern .....	107
	MODIFY-VALUE              Operandenwertdefinition ändern .....	108
	OPEN-SYNTAX-FILE        Syntaxdatei öffnen .....	109
	REMOVE                    Objekte der Syntaxdatei löschen .....	110
	SET-GLOBALS               Globalinformation ändern .....	112
	SHOW                       Objekte der Syntaxdatei anzeigen .....	118
	SHOW-CORRECTION-INFORMATION      Korrekturinformation der Syntaxdatei ausgeben .....	123
	SHOW-STATUS               Status der geöffneten Syntaxdatei anzeigen .....	124
	STEP                       Definieren eines Wiederaufsetzpunktes .....	125











---

# SDF V4.5A (BS2000/OSD)

## SDF-Verwaltung

Benutzerhandbuch

### *Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und an erfahrene BS2000-Benutzer.

### *Inhalt*

Es beschreibt, wie SDF mit Hilfe von SDF-Kommandos und den Dienstprogrammen SDF-I, SDF-U und SDF-PAR installiert und verwaltet wird. Die Anweisungen von SDF-I, SDF-U und SDF-PAR sind vollständig beschrieben.

**Ausgabe: März 2002**

**Datei: sdf\_vw.pdf**

Copyright © Fujitsu Siemens Computers GmbH, 2002.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

Dieses Handbuch wurde erstellt von  
cognitas. Gesellschaft für Technik-Dokumentation mbH  
[www.cognitas.de](http://www.cognitas.de)

Fujitsu Siemens Computers GmbH  
Handbuchredaktion  
81730 München

# Kritik Anregungen Korrekturen

**Fax: 0 700 / 372 00000**

e-mail: [manuals@fujitsu-siemens.com](mailto:manuals@fujitsu-siemens.com)  
<http://manuals.fujitsu-siemens.com>

---

Absender

---

Kommentar zu SDF V4.5A  
SDF-Verwaltung



## Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/) and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009