

---

# 1 Einleitung

Dieses Kapitel gibt eine Übersicht über den Leistungsumfang des Software-Produkts SDF. Daneben enthält es Hinweise für die Benutzung des Handbuchs.

## 1.1 Kurzbeschreibung des Produkts

Das Software-Produkt SDF (**S**ystem **D**ialog **F**acility) unterstützt die Eingabe von Kommandos und Programmanweisungen im Dialog, aus Prozeduren und im Stapelbetrieb.

Wie bereits der Produktname „System Dialog Facility“ andeutet, unterstützt SDF den Benutzer vor allem bei der Eingabe im Dialog. Der Benutzer kann für die Eingabe von der Datensichtstation zwischen drei Stufen des geführten Dialogs und zwei Stufen des ungeführten Dialogs wählen.

Der Benutzer kann temporär vom ungeführten in den geführten Dialog wechseln. Arbeitet er im Dialog mit minimaler oder mittlerer Führung, so kann er temporär in die nächsthöhere Führungsstufe wechseln.

Die Eingaben aus einer Prozedurdatei und die im Stapelbetrieb entsprechen denen im ungeführten Dialog.

Erfolgt die Eingabe im Stapelbetrieb, aus einer Prozedurdatei oder im ungeführten Dialog, so kann ein Ausdruck an Stelle eines Kommando- bzw. Anweisungsteils angegeben werden. SDF ersetzt diese Angabe durch den Inhalt der angegebenen Jobvariablen, den Wert eines S-Variablenausdrucks oder Prozedurparameters.

Sicherheitsrelevante Eingaben wie z.B. Kennwörter schützt SDF durch Dunkelsteuerung bei der Eingabe und durch Ausblendung im Protokoll.

SDF prüft eingegebene Kommandos und Anweisungen sowie deren Operandenwerte auf formale Richtigkeit, z.B. ob ein eingegebener Operandenwert die BS2000-Konvention für Dateinamen erfüllt. Wenn SDF formale Fehler feststellt, führt es mit dem Benutzer einen Korrekturdialog.

SDF unterstützt auch einen Dialog zur Korrektur formal richtiger, aber inhaltlich unzulässiger Eingaben. Voraussetzung dafür ist, dass dieser Dialog in der Kommando- bzw. Anweisungsimplementierung vorgesehen ist.

Syntaktisch richtige Eingabe werden von SDF gespeichert. Der Benutzer kann sich die gespeicherten Eingaben ansehen oder eine bestimmte Eingabe erneut ausgeben lassen, um sie unverändert oder auch modifiziert erneut einzugeben.

Für die Eingabe im Dialog kann sich der Benutzer häufig benötigte Operandenwerte, die nicht als Default-Wert eingestellt sind, als task-spezifische Default-Werte definieren, die danach nicht mehr explizit angegeben werden müssen.

SDF bietet zusammen mit SDF-A (**S**ystem **D**ialog **F**acility-**A**dmistration) die Möglichkeit, den Funktionsumfang von Kommandos und Anweisungen zu beeinflussen:

- Sperren von Kommandos und Anweisungen
- Einschränken des Funktionsvorrates, z.B. durch Sperren von Operanden oder von Operandenwerten
- Ändern von Kommandos und Anweisungen, z.B. durch Ändern des Default-Werts für einen wahlweisen Operanden
- Definition eigener Kommandos, bei deren Aufruf eine Prozedur aufgerufen wird (implementierte Prozedur)

Die Modifizierung der Kommando-/Anweisungsmenge kann so vorgenommen werden, dass sie systemweit oder nur für einzelne Benutzer wirksam ist.

Kommandos, Anweisungen, Operanden und Operandenwerte lassen sich generell für alle Betriebsarten oder gezielt für eine einzelne Betriebsart, z.B. den Dialogbetrieb, sperren.

SDF bietet die Basis für eine mehrsprachige Kommandosprache.

Die SDF-Dialogführung erfolgt in der Sprache, die auch für die Ausgabe von Systemmeldungen gilt. Eingabeaufforderungen, Hilfetexte, Meldungen von SDF und die Kommentierung der Menüs werden in der jeweils eingestellten Sprache ausgegeben.

Standardmäßig sind diese Texte in deutscher und in englischer Sprache in den ausgelieferten Syntaxdateien enthalten. Zusätzlich besteht die Möglichkeit, die englischen Schlüsselwörter der Kommandosprache in die jeweilige Landessprache zu übertragen. Die Schlüsselwörter werden dabei in der Syntaxdatei umbenannt (mit SDF-A). So kann z.B. das Kommando CREATE-FILE in ERZEUGE-DATEI umbenannt werden.

Zu SDF gehört eine eigene Kommandosprache des BS2000 (Kommandos in SDF-Format). Diese ersetzt die bisherige Kommandosprache ISP, die bis einschließlich BS2000 V9.5A verwendet wurde. Kommandos im ISP-Format werden von SDF aus Kompatibilitätsgründen ebenfalls noch unterstützt. Der volle Leistungsumfang von SDF, z.B. die Benutzerführung, ist allerdings nur bei Benutzung der neuen Kommandosprache verfügbar. Die Kommandos im SDF-Format zeichnen sich dadurch aus, dass sie leicht lesbar und leicht verständlich sind. Sie lassen sich in stark verkürzter Form eingeben.

Mit dem Dienstprogramm SDF-CONV kann der Benutzer Prozeduren, die noch ISP-Kommandos enthalten, in Prozeduren mit den entsprechenden SDF-Kommandos umsetzen (siehe Handbuch „SDF-CONV“ [7]).

## 1.2 Zielgruppe des Handbuchs

Das vorliegende Handbuch wendet sich an alle Benutzer der Dialogschnittstelle sowie an DV-Manager und Systembetreuer, die über den Einsatz von SDF zu entscheiden haben. Es vermittelt dem Benutzer, welche Möglichkeiten ihm SDF für die Eingabe von Kommandos und Anweisungen bietet und gibt einen Überblick über alle SDF-Leistungen. Zur Benutzung des Handbuchs werden Grundkenntnisse des BS2000 vorausgesetzt. Weitergehende SDF-Leistungen, die für die Zwecke der Systembetreuung genutzt werden können, sind in den Handbüchern „SDF-Verwaltung“ [5] und „SDF-A“ [4] beschrieben. Benutzer, die SDF-Leistungen, wie selbstdefinierte Kommandos oder die Programm-Schnittstelle, nutzen wollen, finden dies im Handbuch „SDF-A“ [4] beschrieben.

## 1.3 Konzept des Handbuchs

Dieses Handbuch stellt die mit SDF (**S**ystem **D**ialog **F**acility) realisierte Dialogschnittstelle des BS2000 vor und beschreibt, wie der Benutzer über sie Kommandos und Anweisungen eingibt. Das Handbuch hat folgenden Aufbau:

- Kapitel 1 informiert über den Leistungsumfang von SDF und gibt Hinweise zur Benutzung des Handbuchs.
- Kapitel 2 bietet einen Schnelleinstieg für das Arbeiten in SDF-Umgebung. In kurzen Beispielen werden Eingabemöglichkeiten dargestellt.
- Kapitel 3 stellt die für SDF entwickelte Kommandosprache des BS2000 vor.
- Kapitel 4 gibt eine Zusammenstellung der Regeln und Hinweise, die bei der Kommando- und Anweisungseingabe zu beachten sind.
- Kapitel 5 zeigt anhand einer Beispielsitzung, welche Eingabemöglichkeiten der Benutzer im geführten und ungeführten Dialog besitzt.
- Kapitel 6 beschreibt die Syntaxdateien, auf die SDF bei der Verarbeitung einer Eingabe zugreift. Anschließend werden Informationen über den Leistungsumfang von Softwareprodukten zur Veränderung und Erstellung von Syntaxdateien gegeben.
- Kapitel 7 enthält die SDF-spezifischen Kommandos, Standardanweisungen und eine Übersicht der Programmschnittstellen, die dem Benutzer zur Verfügung stehen.
- Kapitel 8 beschreibt die Ausgabe der SHOW-Kommandos in strukturierte S-Variablen.
- Kapitel 9 gibt Hinweise für OMNIS und CHECKPOINT/RESTART bei Einsatz von SDF.

Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die verwiesen wird, ist im Literaturverzeichnis aufgeführt.

## 1.4 Änderungen gegenüber der vorherigen Ausgabe

### *Stand der Beschreibung*

SDF V4.5A kann ab BS2000/OSD-BC V3.0 eingesetzt werden. Die Beschreibung des vorliegenden Handbuchs bezieht sich auf den Einsatz von SDF V4.5A in einem System mit BS2000/OSD-BC V5.0. Vor allem wurden auch die Ablaufbeispiele und Beispielsitzungen in einem System mit BS2000/OSD-BC V5.0 erstellt.

### *LOGOFF-Prozedur*

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 die Systembetreuung je eine Call- und eine Include-Prozedur in der SDF-Parameterdatei vereinbaren (Kommando MODIFY-SDF-PARAMETERS). Eine benutzerspezifische Call- und/oder Include-Prozedur kann der Benutzer, wie bei der LOGON-Prozedur, über den Standarddateinamen bereitstellen.

### *Kommando-Returncodes*

Mit dem Kommando SHOW-RETURNCODE kann der Kommando-Returncode des zuletzt ausgeführten Kommandos abgefragt werden.

Die Kommando-Returncodes der Kommandos RESET-/SHOW-INPUT-DEFAULTS, RESTORE-SDF-INPUT und SHOW-INPUT-HISTORY wurden geändert bzw. erweitert.

### *Syntax-Information*

Bei der Ausgabe des Kommandos SHOW-CMD werden auch Aliasnamen angezeigt.

Die Funktionalität des Kommandos SHOW-CMD wird über die neue Standardanweisung SHOW-STMT auch auf Programmebene zur Verfügung gestellt.

### *Beispiele*

Die Beispiele wurden unter BS2000/OSD V5.0 erstellt. Bei den Beispielen mit Menü-Führung (vor allem in Kapitel 2 und 5) ist für die Funktionstastenbelegung der Style-Guide-Modus (FUNCTION-KEYS=\*STYLE-GUIDE-MODE) eingestellt. Benutzer, die im Old-Modus (weil z.B. das Terminal den Style-Guide-Modus nicht unterstützt) arbeiten, sollten beachten, dass im Old-Modus zum Teil andere Tasten zugeordnet sind und insgesamt weniger Funktionstasten zur Verfügung stehen. Die aktuell möglichen Funktionen (und die entsprechenden Funktionentasten) werden jedoch immer unter der NEXT-Zeile angezeigt.

### Übersicht der Änderungen der Kommandoschnittstelle:

Kommando	Änderung
RESET-INPUT-DEFAULTS	– Kommando-Returncode geändert/erweitert
RESTORE-SDF-INPUT	– Kommando-Returncode geändert/erweitert
SHOW-CMD	– CMD-NAME= zeigt zusätzlich Aliasnamen an
SHOW-INPUT-DEFAULTS	– Kommando-Returncode geändert/erweitert
SHOW-RETURNCODE	– Neues Kommando; zeigt den Kommando-Returncode des letzten Kommandos an

### Readme-Datei

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

Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter den Dateinamen *SYSRME.produkt.version.sprache*, für SDF V4.5 also 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 für SDF gibt auch das folgende Kommando aus:

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

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

```
/PRINT-DOCUMENT <Pfadname>, LINE-SPACING=*BY-EBCDIC-CONTROL
```

### Begriffe

Im vorliegenden Handbuch werden die Begriffe S-Prozedur, Nicht-S-Prozedur, Dialogblock, S-Variable, vordefinierte Funktion (Builtin-Funktion) sowie S-Marke und Nicht-S-Marke verwendet:

- Die S-Prozedur ist eine Prozedur, deren Prozedur- und Parametervereinbarungen nach SDF-P-Regeln gebildet sind. Der Ablauf kann mit SDF-P-Sprachmitteln gesteuert werden. Bei Fehlern wird die SDF-P-Fehlerbehandlung angestoßen.
- Die Nicht-S-Prozedur ist eine Prozedur, die nach den bisherigen Regeln erstellt wurde (z.B. erstes Kommando BEGIN-PROCEDURE). Bei Fehler wird Spin-Off ausgelöst.
- Der Dialogblock ist eine besondere Form der Dialogeingabe bei SDF-P.
- S-Variablen gehören zum Variablenkonzept von SDF-P. Parameter einer S-Prozedur sind S-Variablen.

- Die Builtin-Funktionen gehören ebenfalls zum Variablenkonzept von SDF-P. Mit ihrer Hilfe können Inhalte von S-Variablen verarbeitet oder Umgebungsinformationen beschafft werden.
- Nicht-S-Marken sind die bisherigen Marken, die vor dem Kommandonamen stehen können.
- In S-Prozeduren können zusätzlich S-Marken verwendet werden. Diese Marken werden nur von SDF-P ausgewertet.

---

## 2 Schnelleinstieg in die SDF-Anwendung

SDF ist eine Kommandosprache, die komfortable Möglichkeiten für die Dialogeingabe bietet. Die Dialogführung erleichtert die Kommandoingabe, informiert den Anwender bei Bedarf über die Funktion einzelner Operanden und ermöglicht einen Fehlerkorrekturdialog bei falschen Eingaben. Die Eingabe von SDF-Kommandos erfordert nur geringe Vorkenntnisse; die nachfolgenden kleinen Beispiele sollen Sie davon überzeugen. Abhängig von der Konfiguration Ihres Rechners können Ihre Ausgabebildschirme sich von denen in den Beispielen unterscheiden (z.B. Umfang der angebotenen Kommandos).

Zuerst aber noch einige SDF-Highlights:

### *Mnemotechnische Namen*

Bereits die Kommandosprache will Ihnen den Gebrauch von Kommandos einfach machen: Namen von Kommandos, Operanden und konstanten Operandenwerten sind so gewählt, dass sie die Funktion oder den Zweck ihrer Verwendung erkennen lassen. Darüber hinaus sind in den Kommandos analoge Sachverhalte gleich benannt.

### *Abkürzungsmöglichkeiten*

Lassen Sie sich nicht von langen Eingaben abschrecken. Im Dialog bietet Ihnen SDF viele Möglichkeiten, Ihre Eingabe stark abzukürzen. So besitzen die meisten Operanden einen voreingestellten Operandenwert, den Default-Wert. Sie müssen diesen Operanden nur angeben, wenn Sie einen anderen Wert wünschen. Sollten einige Default-Werte den von Ihnen häufig genutzten Werten nicht entsprechen, können Sie sich für Ihre Task eigene Default-Werte definieren.

Namen können Sie ebenfalls abkürzen: Innerhalb von Namen bzw. Teilnamen lassen Sie von rechts nach links so viele Zeichen weg, dass der Name in seinem Kontext gerade noch eindeutig ist. Dabei müssen Sie beachten, dass ein Kommandoname eindeutig innerhalb aller Kommandonamen ist, ein Operandenname eindeutig innerhalb aller Operandennamen des Kommandos auf der betreffenden Strukturstufe.

### *Zeilen- oder Menümodus*

Sie können Ihre Eingaben direkt im Zeilenmodus (ungeführter Dialog, [Seite 51](#)) vornehmen. Sie können aber auch in einen menügesteuerten Eingabemodus (geführter Dialog, [Seite 57](#)) umschalten. Der geführte Dialog bietet drei Führungsstufen, die sich in der Ausführlichkeit der Menüs unterscheiden. Sie können jederzeit den Eingabemodus wechseln.

### *Fehlerkorrekturdialog*

Ihre Eingaben prüft SDF bereits vor der Kommandoausführung auf Syntaxfehler und schaltet abhängig von der eingestellten Führungsstufe automatisch in einen Fehlerkorrekturdialog um.

### *Kommando rekonstruieren*

Sie können Ihre Eingaben von SDF in einem Puffer speichern lassen. Eine so gespeicherte Eingabe können Sie zurückholen und zur erneuten Eingabe verwenden (siehe [Seite 104](#)).

### *Auskunftsysteem SDF*

Mit der Eingabe eines Fragezeichens fordern Sie Hilfe von SDF an. In Abhängigkeit von der Position des Fragezeichens innerhalb der Eingabe und der eingestellten Führungsstufe bietet Ihnen SDF unterschiedliche Hilfestellung bei der Kommandoeingabe. Das kann ein Auswahlmenü für bestimmte Kommandos sein, ein Operandenfragebogen für ein bestimmtes Kommando oder auch Informationen zu einem bestimmten Operanden.

Die einfachste Form der Hilfestellung bietet Ihnen SDF im ungeführten Dialog:

Mit der Eingabe eines Fragezeichens wechseln Sie für die Eingabe genau eines Kommandos in den geführten Dialog. Nach Abschluss der Kommandoeingabe befinden Sie sich wieder im ursprünglichen Eingabemodus, dem ungeführten Dialog. Diese Art der Eingabe wird temporär geführter Dialog (siehe [Seite 76](#)) genannt.



## Anwendungsbereiche ausgeben

Mit der Eingabe eines Fragezeichens und **[DUE]** liefert Ihnen SDF eine Auswahl von Anwendungsbereichen (siehe Anwendungsbereichs-Menü, [Seite 70](#)). Für das nächste Beispiel wählen Sie den Anwendungsbereich USER-ADMINISTRATION. Dazu geben Sie in das Eingabefeld NEXT (siehe NEXT-Zeile, [Seite 63](#)) die Nummer des Anwendungsbereichs (hier die 26) ein und schicken das Menü mit **[DUE]** weg.

/?

-----

AVAILABLE APPLICATION DOMAINS:

1 ACCOUNTING	14 MULTI-CATALOG-AND-PUBSET-MGMT
2 ALL-COMMANDS	15 NETWORK-MANAGEMENT
3 CONSOLE-MANAGEMENT	16 PROCEDURE
4 DATABASE	17 PROGRAM
5 DCAM	18 PROGRAMMING-SUPPORT
6 DEVICE	19 SDF
7 FILE	20 SECURITY-ADMINISTRATION
8 FILE-GENERATION-GROUP	21 SPOOL-PRINT-ADMINISTRATION
9 FILE-TRANSFER	22 SPOOL-PRINT-SERVICES
10 IDOM	23 STORAGE-MANAGEMENT
11 JOB	24 SYSTEM-MANAGEMENT
12 JOB-VARIABLES	25 SYSTEM-TUNING
13 MESSAGE-PROCESSING	26 USER-ADMINISTRATION

-----

NEXT = + ←

KEYS : F1=? F3=\*EXIT F5=\*REFRESH F6=\*EXIT-ALL F8=+ F9=REST-SDF-IN F12=\*CANCEL

## Benutzerkennwort ändern

Sie wollen Ihr Benutzerkennwort ändern. Dazu haben Sie bereits den Anwendungsbereich USER-ADMINISTRATION ausgewählt. Sie erhalten ein Auswahlmenü mit allen Kommandos dieses Anwendungsbereichs (siehe Kommando-Menü, [Seite 71](#)). Wählen Sie das Kommando MODIFY-USER-PROTECTION durch Eingabe der Nummer 13 in das Eingabefeld NEXT.

```
DOMAIN      : USER-ADMINISTRATION
-----
AVAILABLE COMMANDS:

 1  ADD-USER
 2  ADD-USER-GROUP
 3  COPY-TERMINAL-SET
 4  CREATE-TERMINAL-SET
 5  DELETE-TERMINAL-SET
 6  LOCK-USER
 7  MODIFY-LOGON-PROTECTION
 8  MODIFY-POSIX-USER-ATTRIBUTES
 9  MODIFY-POSIX-USER-DEFAULTS
10  MODIFY-TERMINAL-SET
11  MODIFY-USER-ATTRIBUTES
12  MODIFY-USER-GROUP
13  MODIFY-USER-PROTECTION
15  MODIFY-USER-SWITCHES
16  REMOVE-USER
17  REMOVE-USER-GROUP
18  SET-LOGON-PROTECTION
19  SHOW-LOGON-PROTECTION
20  SHOW-OPERATOR-ATTRIBUTES
21  SHOW-OPERATOR-ROLE
22  SHOW-PERSONAL-LOGON-ADMISSION
23  SHOW-POSIX-USER-ATTRIBUTES
24  SHOW-POSIX-USER-DEFAULTS
25  SHOW-PRIVILEGE
26  SHOW-TERMINAL-SET
27  SHOW-USER-ATTRIBUTES
-----
NEXT = 13←
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F12=*CANCEL
```

Tragen Sie im Operandenfragebogen Ihr bisheriges und Ihr neues Benutzerkennwort in die geschützten Eingabefelder LOGON-PASSWORD und NEW-LOGON-PASSWORD ein und schicken Sie ihn mit **[DUE]** weg.

```
DOMAIN      : USER-ADMINISTRATION          COMMAND: MODIFY-USER-PROTECTION

-----
LOGON-PASSWORD      = 'wo%t$aut'  → Eingabe nicht sichtbar
NEW-LOGON-PASSWORD = 'auf54ht$'  → Eingabe nicht sichtbar
CONFIRM-NEW-PASSWORD =
PUBSET              = *HOME
USER-IDENTIFICATION = *STD

-----
NEXT = *EXECUTE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL
```

Der temporär geführte Dialog ist beendet und Sie erhalten wieder den Schrägstrich als Aufforderung zur Kommandoingabe.

## Task-lokale SDF-Optionen abfragen und ändern

Informieren Sie sich über die aktuellen SDF-Optionen für Ihre Task, wie z.B. Eingabemodus, Umfang der Protokollierung oder Kommandogedächtnis (History). Mit der Eingabe eines Fragezeichens und **[DUE]** erhalten Sie wieder das Auswahlmenü mit den Anwendungsbereichen. Unser Kommando gehört zu dem Anwendungsbereich mit der Nummer 19 (SDF). Dort wählen Sie das Kommando SHOW-SDF-OPTIONS aus. Geben Sie dazu die Nummer 10 in das Eingabefeld NEXT ein und schicken das Menü mit **[DUE]** weg.

```
DOMAIN : SDF
-----
AVAILABLE COMMANDS:

 1  HELP-SDF
 2  MODIFY-SDF-OPTIONS
 3  REMARK
 4  RESET-INPUT-DEFAULTS
 5  RESTORE-SDF-INPUT
 6  SHOW-CMD
 7  SHOW-INPUT-DEFAULTS
 8  SHOW-INPUT-HISTORY
 9  SHOW-RETURNCODE

10  SHOW-SDF-OPTIONS
11  SHOW-SYNTAX-VERSIONS
12  START-SDF-A
13  START-SDF-CONV
14  START-SDF-I
15  START-SDF-SIM
16  START-SDF-U
17  WRITE-TEXT

( ! )

-----
NEXT = 10 ←
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F9=REST-SDF-IN F12=*CANCEL
```

Sie erhalten den Operandenfragebogen des Kommandos. In das Eingabefeld **INFORMATION**, das mit dem Default-Wert **\*ALL** vorbelegt ist, tragen Sie **\*user** ein und schicken den Fragebogen mit **[DUE]** weg.

```

DOMAIN      : SDF                                COMMAND: SHOW-SDF-OPTIONS

-----

INFORMATION      = *user

-----

NEXT = *EXECUTE
KEYS : F1=?      F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

SDF gibt Ihnen die SDF-Optionen Ihrer Task aus. Durch **INFORMATION=\*USER** erhalten Sie nur SDF-Optionen, die Sie selbst mit dem Kommando **MODIFY-SDF-OPTIONS** ändern können.

```

% USER      : *NONE
%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 : *OLD-MODE
% INPUT-HISTORY : *ON
% NUMBER-OF-INPUTS : 20
% PASSWORD-PROTECTION: *YES
/

```

Das Ausgabefeld INPUT-HISTORY mit dem Wert \*ON zeigt an, dass das Kommandogedächtnis eingeschaltet ist. Das Ausgabefeld NUMBER-OF-INPUTS zeigt an, dass SDF sich maximal die letzten 20 Eingaben merkt.

In einem der nachfolgenden Beispiele wird auf das Kommandogedächtnis zurückgegriffen. Sollte INPUT-HISTORY bei Ihnen den Wert \*OFF enthalten, schalten Sie bitte das Kommandogedächtnis ein. Dazu geben Sie z.B. direkt unter Nutzung der Abkürzungsmöglichkeiten „mod-sdf-opt input=\*on“ ein oder fordern Sie im temporär geführten Dialog den Operandenfragebogen an und ändern den Wert für INPUT-HISTORY entsprechend.

## Erstellen einer LOGON-Prozedur

Schreiben Sie sich eine kleine Prozedur, die Ihnen nach dem „Einloggen“ automatisch alle Dateien anzeigt, auf die seit mindestens 20 Tagen nicht mehr zugegriffen wurde. Die Kommandos, aus denen die Prozedur besteht, müssen Sie in einer Datei mit dem Standardnamen SYS.SDF.LOGON.USERPROC unter Ihrer Benutzerkennung abspeichern. Der Standardname bewirkt, dass SDF die Prozedur sofort nach dem Einloggen automatisch aufruft (siehe LOGON-Prozeduren, [Seite 81](#)). Rufen Sie zum Erstellen der Prozedurdatei das (Editor-)Programm EDT<sup>1)</sup> auf. Mit „?“ und **[DUE]** erhalten Sie wieder das Auswahlmenü mit den Anwendungsbereichen. Mit „17“ und **[DUE]** erhalten Sie die Kommandos des Anwendungsbereichs PROGRAM. Mit **[DUE]** blättern Sie einen Bildschirm weiter (die NEXT-Zeile ist bereits mit „+“ vorbesetzt). Zum Starten des Programms wählen Sie im darauf folgenden Bildschirm mit „34“ das Kommando START-EXECUTABLE-PROGRAM<sup>2)</sup> und schicken das Menü mit **[DUE]** weg.

```

DOMAIN      : PROGRAM
-----
AVAILABLE COMMANDS:

 1 ASSIGN-SYSDTA           15 LOAD-EXECUTABLE-PROGRAM
 2 ASSIGN-SYSIPT          16 LOAD-PROGRAM
 3 ASSIGN-SYSLST          17 MODIFY-DBL-DEFAULT
 4 ASSIGN-SYSOPT           18 MODIFY-TEST-OPTIONS
 5 ASSIGN-SYSOUT           19 REMOVE-TASKLIB           (!)
 6 CANCEL-PROGRAM         (!) 20 RESET-DBL-DEFAULT
 7 COPY-SYSTEM-FILE       21 RESUME-HARDWARE-AUDIT
 8 CREATE-DUMP             (!) 22 RESUME-LINKAGE-AUDIT
 9 DELETE-SYSTEM-FILE     23 RESUME-PROGRAM           (!)
10 EOF                     (!) 24 SELECT-PRODUCT-VERSION
11 HOLD-HARDWARE-AUDIT    25 SELECT-PROGRAM-VERSION
12 HOLD-LINKAGE-AUDIT     26 SET-TASK-CLOCK
13 INFORM-OPERATOR        27 SET-TASKLIB

-----
NEXT = +
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F12=*CANCEL

```

<sup>1)</sup> Das Dienstprogramm EDT kann auch mit dem Kommando START-EDT aufgerufen werden.

<sup>2)</sup> Das Kommando ist ab BLSSERV V2.3 verfügbar und ersetzt langfristig das Kommando START-PROGRAM.

```

DOMAIN      : PROGRAM
-----
AVAILABLE COMMANDS:

 16 LOAD-PROGRAM                29 SHOW-HARDWARE-AUDIT
 17 MODIFY-DBL-DEFAULT          30 SHOW-LINKAGE-AUDIT
 18 MODIFY-TEST-OPTIONS        31 SHOW-SELECTED-PRODUCT-VERSION
 19 REMOVE-TASKLIB              (!) 32 SHOW-SYSTEM-FILE-ASSIGNMENTS
 20 RESET-DBL-DEFAULT           33 SHOW-TASK-CLOCK
 21 RESUME-HARDWARE-AUDIT       34 START-EXECUTABLE-PROGRAM
 22 RESUME-LINKAGE-AUDIT        35 START-HARDWARE-AUDIT
 23 RESUME-PROGRAM              (!) 36 START-LINKAGE-AUDIT
 24 SELECT-PRODUCT-VERSION      37 START-PROGRAM
 25 SELECT-PROGRAM-VERSION      38 START-TASK-MEASUREMENT
 26 SET-TASK-CLOCK              39 STOP-HARDWARE-AUDIT
 27 SET-TASKLIB                 40 STOP-LINKAGE-AUDIT
 28 SHOW-DBL-DEFAULT            (!) 41 STOP-TASK-MEASUREMENT
-----
NEXT = 34 ←
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F9=REST-SDF-IN F12=*CANCEL

```

Als Alternative können Sie statt mit „?“ z.B. auch mit der Eingabe „\*prog?“ beginnen. Sie erhalten ein Auswahlm Menü mit allen Kommandos, die PROGRAM im Namen enthalten (siehe [Abschnitt „Auswahlm Menü für Kommandos und Anweisungen“ auf Seite 102](#)). Hier wählen Sie das Kommando START-EXECUTABLE-PROGRAM mit „8“ aus.

```

OPERATIONS CORRESPONDING TO: *PROG*
-----
AVAILABLE COMMANDS:

 1 CANCEL-PROGRAM                (!) 6 RESUME-PROGRAM                (!)
 2 INFORM-PROGRAM                7 SELECT-PROGRAM-VERSION
 3 LOAD-EXECUTABLE-PROGRAM        8 START-EXECUTABLE-PROGRAM
 4 LOAD-PROGRAM                  9 START-FORI-PROGRAM
 5 RESTART-PROGRAM               10 START-PROGRAM
-----
NEXT = 8 ←
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F9=REST-SDF-IN F12=*CANCEL

```

Sie erhalten in beiden Fällen den Operandenfragebogen des Kommandos START-EXECUTABLE-PROGRAM. In das Eingabefeld FROM-FILE tragen Sie den Programmnamen \$.EDT ein. Das ist der Name der Datei, aus der das Programm EDT geladen wird (in diesem Fall die Datei EDT in der System-Standardkennung; einen davon abweichenden Namen erfahren Sie von Ihrer Systembetreuung).

```

COMMAND : START-EXECUTABLE-PROGRAM

-----
FROM-FILE           = $.edt
PROGRAM-PARAMETERS = *NONE
DBL-PARAMETERS     = *STD
MONJV              = *NONE
CPU-LIMIT          = *JOB-REST
TEST-OPTIONS       = *DBL-DEFAULT
RESIDENT-PAGES     = *PARAMETERS(MINIMUM=*STD,MAXIMUM=*STD)
VIRTUAL-PAGES      = *STD

-----
NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8=+   F9=REST-SDF-IN
       F11=*EXECUTE   F12=*CANCEL

```

Nach dem Laden des EDT wird die EDT-Bildschirmmaske ausgegeben.

Tragen Sie in die Eingabefelder 1.00 bis 2.00 der EDT-Maske die angegebenen Kommandos ein. Das WRITE-TEXT-Kommando gibt den angegebenen Text nach SYSOUT aus. Es kommentiert die Dateiauswahl des SHOW-FILE-ATTRIBUTES-Kommandos.

Um die Eingabe als Datei SYS.SDF.LOGON.USERPROC abzuspeichern, geben Sie in die letzte Zeile der EDT-Maske den Befehl „write'sys.sdf.logon.userproc“ ein und schicken die Maske mit **DUE** weg.



```

1.00 /WRITE-TEXT '*** MINDESTENS 20 TAGE KEIN DATEIZUGRIFF AUF: '
2.00 /SHOW-FILE-ATTR SELECT=(LAST-ACCESS-DATE=*INTERVAL(TO=-20))
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00
17.00
18.00
19.00
20.00
21.00
22.00
23.00
w'sys.logon.userproc'                                0001.00:001(0)

```

Sie erhalten erneut die EDT-Maske, die mit der Meldung EDT0173 das Erstellen der Datei SYS.SDF.LOGON.USERPROC bestätigt. War die Datei bereits vorhanden, fragt Sie der EDT zuvor, ob die Datei überschrieben werden soll.

Mit „halt“ in der letzten Zeile und **[DUE]** beenden Sie den EDT.

Rufen Sie zur Kontrolle die Prozedur mit „call-proc sys.sdf.logon.userproc“ auf.

```

% EDT8000 EDT TERMINATED
/call-proc sys.sdf.logon.userproc
*** MINDESTENS 20 TAGE KEIN DATEIZUGRIFF AUF:
%      6 :20SG:$SDFUSER1.ABK.ABKLLST
%      9 :20SG:$SDFUSER1.ABK.KDOLST
%      6 :20SG:$SDFUSER1.ABK.NEU
%     24 :20SG:$SDFUSER1.ABK.NEU.V11.01
%     48 :20SG:$SDFUSER1.ABK.V101
%     99 :20SG:$SDFUSER1.ABK.V110
%    126 :20SG:$SDFUSER1.ABK.V110.ISAM
%      6 :20SG:$SDFUSER1.ABK.V110.SORT
%    537 :20SG:$SDFUSER1.SDFFRAME.SF.ALF.SYSSDF.SPOOL.041.120
%    537 :20SG:$SDFUSER1.SDFFRAME.SF.ALF.SYSSDF.SPOOL.042.120
%   537#:20SG:$SDFUSER1.SDFFRAME.SF.HSMS.060
%   279#:20SG:$SDFUSER1.SDFFRAME.SF.MAR1.SYSSDF.MAREN.090
%   537#:20SG:$SDFUSER1.SDFFRAME.SF.SPOOL.043.120
%:20SG: PUBLIC:      10 FILES RES=      1398 FRE=      246 REL=      207 PAGES
%:20SG: PUB/S2:       3 FILES RES=      1353 FRE=      286 REL=      265 PAGES
/

```

Diese direkte Eingabe nutzt die Abkürzungsmöglichkeiten (Abkürzung des Kommandonamens CALL-PROCEDURE; Angabe der Prozedurdatei als erster Stellungsoperand, da NAME der erste Operand im Kommando ist). Alternativ können Sie auch die bisher genutzte Methode verwenden: Eingabe „?“ → „16“ (PROCEDURE) → Auswahl CALL-PROCEDURE → Dateiname bei FROM-FILE eintragen.

### Kommando rekonstruieren

Haben Sie sich beim Aufruf der Prozedur SYS.SDF.LOGON.USERPROC (siehe vorhergehendes Beispiel) verschrieben, z.B. einen fehlerhaften Dateinamen angegeben? In diesem Fall lassen Sie sich mit „restore-sdf“, einer möglichen Abkürzung des Kommandos RESTORE-SDF-INPUT, die fehlerhafte Eingabe erneut ausgeben.

```

/call-proc sys.sdf.logn.userproc
% SSM2052 PROCEDURE FILE 'SYS.SDF.LOGN.USERPROC' CANNOT BE OPENED. DMS ERROR CO
DE '0D33'. COMMAND TERMINATED. DMS ERROR: /HELP-MSG-INFORMATION DMS0D33
/restore-sdf
/call-proc sys.sdf.logn.userproc
/

```

↑ Schreibmarke positionieren und "o" einfügen

Korrigieren Sie den Dateinamen in der Kommandozeile indem Sie z.B. die Schreibmarke hinter „sys.sdf.log“ positionieren und nach Drücken der **[EFG]**-Taste den fehlenden Buchstaben „o“ einfügen. Die somit korrigierte Kommandozeile schicken Sie mit **[DUE]** weg.

## Dialogtask beenden

Beenden Sie die Dialogtask mit dem Kommando EXIT-JOB. Nach der Eingabe von „exit-job?“ wechselt SDF in den temporär geführten Dialog und gibt Ihnen direkt den Operandenfragebogen des Kommandos aus.

```

COMMAND  : EXIT-JOB

-----
MODE                = *NORMAL
SYSTEM-OUTPUT      = *ALL
KEEP-CONNECTION    = *NO  ← hier bitte *YES eintragen

-----
NEXT = *CONTINUE
KEYS : F1=?  F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F8=+  F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

Den Operandenbogen erhalten Sie auch mit folgenden Eingaben:

„?“ → „11“ (JOB) → Auswahl EXIT-JOB.

Da wir uns im nachfolgenden Beispiel wieder einloggen wollen, ändern Sie bitte im Eingabefeld KEEP-CONNECTION den Wert auf \*YES, bevor Sie das Menü mit **DUE** wegschicken. Damit bleibt nach Taskende die Rechnerverbindung erhalten und Sie können sich sofort wieder einloggen.

## Dialogtask starten

Nach Beendigung Ihrer Dialogtask (siehe vorhergehendes Beispiel) fordert Sie das System auf, sich erneut einzuloggen. Geben Sie „?“ und **DUE** ein.

```

% EXC0419 /LOGOFF AT 1713 ON 01-10-04 FOR TSN '1PYT'
% EXC0421 CPU TIME USED: 0.8174
% JMS0150 INSTALLATION ' S150-40', BS2000 VERSION 'V140', HOST 'D016ZE04': PLEA
SE ENTER '/SET-LOGON-PARAMETERS' OR '?'
/

```

Da Sie sich vor Eingabe weiterer Kommandos zuerst im System mit SET-LOGON-PARAMETERS anmelden müssen, gibt Ihnen SDF sofort den Operandenfragebogen aus. Die Eingabe „set-logon-parameters?“ hat dieselbe Wirkung. Geben Sie Ihre Benutzerkennung, Abrechnungsnummer und Kennwort ein. Die Angabe eines Auftragsnamens ist optional. In der letzten Zeile gibt Ihnen SDF den Hinweis, dass Sie mit \*EXIT eine Übersicht weiterer Kommandos erhalten. \*EXIT können Sie entweder durch Drücken der **[F2]**-Taste (siehe Funktionstastenbelegung in der KEYS-Zeile<sup>1)</sup>) auslösen oder Sie geben \*EXIT in der NEXT-Zeile ein und drücken anschließend die **[DUE]**-Taste. \*EXIT bedeutet: den Operandenfragebogen verlassen und in ein Auswahlmenü wechseln, das alle Kommandos enthält, die Sie vor dem Einloggen verwenden dürfen. Das wären jedoch nur die Kommandos EXIT-JOB und LOGOFF zum Beenden der Task und das Kommando SET-LOGON-PARAMETERS zum Einloggen.

```

COMMAND : SET-LOGON-PARAMETERS

-----
USER-IDENTIFICATION = sdfuser1
ACCOUNT              = acc0815a
PASSWORD             = *auf54ht$*  → Eingabe nicht sichtbar
JOB-CLASS            = *STD
JOB-NAME              = sdftest
MONJV                = *NONE
JV-PASSWORD          =
SCHEDULING-TIME     = *STD
LIMIT                = *STD
RESOURCES            = *PARAMETERS(RUN-PRIORITY=*STD,CPU-LIMIT=*STD,SYSLST-LIMIT
                        =*STD,SYSOPT-LIMIT=*STD)
LOGGING              = *PARAMETERS(LISTING=*NO,HARDCOPY=*NO)
JOB-PARAMETER        = *NO
JOB-PARAMETER        = *NO

-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8=+   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

MESSAGE:  CMD0175 OTHER OPERATIONS DESIRED? PRESS *EXIT KEY

```

Nach Prüfung der LOGON-Angaben startet SDF Ihre LOGON-Prozedur.

<sup>1)</sup> Der Feldname „KEYS“ wird nur im Style-Guide-Modus angezeigt. Haben Sie den Old-Modus eingestellt (weil z.B. Ihr Terminal den Style-Guide-Modus nicht unterstützt), werden Ihnen ebenfalls die möglichen Eingaben und entsprechende Funktionstasten angezeigt, es sind aber insgesamt weniger Funktionstasten verfügbar. In dem Old-Modus entspricht \*EXIT aber der **[K1]**-Taste.

```

% JMS0066 JOB 'SDFTEST' ACCEPTED ON 01-10-04 AT 17:14, TSN = 1PY8
*** MINDESTENS 20 TAGE KEIN DATEIZUGRIFF AUF:
%      6 :20SG:$SDFUSER1.ABK.ABKLLST
%      9 :20SG:$SDFUSER1.ABK.KDOLST
%      6 :20SG:$SDFUSER1.ABK.NEU
%     24 :20SG:$SDFUSER1.ABK.NEU.V11.01
%     48 :20SG:$SDFUSER1.ABK.V101
%     99 :20SG:$SDFUSER1.ABK.V110
%    126 :20SG:$SDFUSER1.ABK.V110.ISAM
%      6 :20SG:$SDFUSER1.ABK.V110.SORT
%    537 :20SG:$SDFUSER1.SDFFRAME.SF.ALF.SYSSDF.SPOOL.041.120
%    537 :20SG:$SDFUSER1.SDFFRAME.SF.ALF.SYSSDF.SPOOL.042.120
%    537#:20SG:$SDFUSER1.SDFFRAME.SF.HSMS.060
%    279#:20SG:$SDFUSER1.SDFFRAME.SF.MAR1.SYSSDF.MAREN.090
%    537#:20SG:$SDFUSER1.SDFFRAME.SF.SPOOL.043.120
%:20SG: PUBLIC:      10 FILES RES=      1398 FRE=      246 REL=      207 PAGES
%:20SG: PUB/S2:       3 FILES RES=      1353 FRE=      286 REL=      265 PAGES
/

```

Sind Sie mit dem Kommando SET-LOGON-PARAMETERS bereits vertraut, können Sie sich auch direkt im System anmelden. Geben Sie dazu das Kommando mit den erforderlichen Angaben ein und schicken es mit **[DUE]** weg. Aus Sicherheitsgründen sollten Sie statt Ihres Benutzerkennwortes den Operandenwert \*SECRET eingeben. SDF gibt Ihnen dann zur Kennworteingabe ein geschütztes Eingabefeld aus.

```

/set-logon-par user=sdfuser1,account=acc0815a,password=*secret,job-name=sdf test
ENTER SECRET OPERAND (PASSWORD): 'auf54ht$' → Eingabe nicht sichtbar
/

```

## Hinweise

Die vorliegenden Beispiele konnten Ihnen nur einige der wichtigsten Eingabemöglichkeiten zeigen. Es wurden nur die Eingabe im ungeführten Dialog und der Wechsel in den temporär geführten Dialog gezeigt.

SDF bietet Ihnen aber auch die Möglichkeit, permanent in einem geführten Dialog zu arbeiten. Das [Kapitel „Beispiele für die Kommandoingabe im Dialogbetrieb“ auf Seite 117](#) beschreibt eine Beispielsitzung mit allen Führungsmodi.

Die Eingabemöglichkeiten, insbesondere Abkürzungsregeln, Führungsstufen, Menüsteuerung und die Definition von task-spezifischen Default-Werten finden Sie im [Kapitel „Kommando- und Anweisungseingabe“ auf Seite 51 ff](#) beschrieben.

Der Funktionsumfang (erlaubte Kommandos, Operanden und Operandenwerte), der Ihnen als Benutzer zur Verfügung steht, ist in Syntaxdateien festgelegt. Die Bedeutung der Syntaxdateien ist im [Kapitel „SDF-Syntaxdateien“ auf Seite 157](#) beschrieben.

Sind in Ihren Prozeduren noch Kommandos der Vorgänger-Kommandosprache ISP enthalten, unterstützt Sie das Dienstprogramm SDF-CONV bei der Umsetzung in SDF-Kommandos. Bei Eingabe eines ISP-Kommandos mit abschließenden Fragezeichen im Dialog, hilft SDF das entsprechende SDF-Kommando zu finden: Gibt es genau ein entsprechendes SDF-Kommando erhalten Sie dessen Operandenfragebogen; ist keine eindeutige Abbildung möglich, gibt SDF die Liste der möglichen SDF-Kommandos aus.

---

## 3 Die Kommandosprache SDF

Für das Software-Produkt SDF wurde eine eigene Kommandosprache entwickelt: die Kommandosprache SDF. Sie ersetzt ab BS2000 V10.0A die Kommandosprache ISP. Kommandos der Kommandosprache ISP werden weiter kompatibel unterstützt. Die volle SDF-Funktionalität kann jedoch nur für Kommandos und Anweisungen mit SDF-Syntax genutzt werden. ISP-Kommandos werden nur auf korrekte Eingabe des Kommandonamens geprüft.

### 3.1 Aufbau der Kommandosprache SDF

Die Syntaxbeschreibung von SDF-Kommandos und -Anweisungen ist in den Syntaxdateien enthalten (siehe [Kapitel „SDF-Syntaxdateien“ auf Seite 157ff](#)):

- Kommandos beginnen mit führendem Schrägstrich, der im Dialog vorgegeben ist. Kommandos erwartet SDF aus der logischen Systemdatei SYSCMD.
- SDF-Anweisungen, d.h. Anweisungen an ein Programm mit SDF-Schnittstelle, beginnen mit führendem Doppelschrägstrich, der im Dialog vorgegeben ist. Anweisungen erwartet SDF aus der logischen Systemdatei SYSSTMT, für die dieselbe Zuweisung gilt, die für die Systemdatei SYSDTA getroffen wurde.

Eingabedaten, d.h. Daten, Parameter und Anweisungen, die ein Programm ohne SDF-Schnittstelle einliest, werden von SDF nicht analysiert. Eingabedaten erwarten Programme aus der logischen Systemdatei SYSDTA.

Ein Kommando bzw. eine Anweisung besteht aus folgenden Elementen:

- Kommando- bzw. Anweisungsname
- Operandennamen
- mögliche Operandenwerte (Schlüsselwortwerte und variable Werte)
- zusätzliche Hilfetexte zur näheren Erläuterung des Kommandos bzw. der Anweisung und den Operanden

Kommando- bzw. Anweisungsnamen, Operandennamen und Schlüsselwortwerte sind als Schlüsselwörter in der Syntaxbeschreibung enthalten. Schlüsselwörter werden als solche angegeben (siehe [Abschnitt „Verkürzte Eingabe“ auf Seite 94 ff](#)). Variable Operandenwerte werden durch Datentypen beschrieben (siehe [Tabelle 2 auf Seite 33ff](#)). Mit dem Datentyp ist festgelegt, aus welchem Zeichenvorrat und nach welchen Vorschriften ein Operandenwert gebildet wird. SDF prüft, ob die Angabe für einen variablen Operandenwert innerhalb

der Menge der zulässigen Operandenwerte liegt, die durch den Datentyp festgelegt ist. Ein variabler Operandenwert kann auch als Einzelwert (konstanter Wert) eines Datentyps definiert sein. Statt des Datentyps enthält die Syntaxbeschreibung dann den jeweiligen Einzelwert (z.B. die Namen der S-Variablenströme im Kommando ASSIGN-STREAM: STREAM-NAME = SYSVAR / SYSMMSG / SYSINF).

Schlüsselwörter bestehen meist aus mehreren Teilnamen, die mit Bindestrichen verbunden sind. Die Namen sind englischsprachig und entstammen meist dem allgemeinen Sprachgebrauch. Sie sind so gewählt, dass in Bezug auf den gesamten Kommandovorrat analoge Sachverhalte durch dieselben Schlüsselwörter beschrieben werden. Kommandos beginnen stets mit einem Verb. Danach folgt das Objekt, auf das die Tätigkeit anzuwenden ist; z.B. MODIFY-FILE-ATTRIBUTES für das Verändern von Dateieigenschaften.

Es gibt auch Paare von gegensätzlichen Tätigkeiten:

CREATE-XY und DELETE-XY für das Erzeugen und Löschen eines Objekts XY.

Kommandos werden im Hinblick auf Sachgebiete in Anwendungsbereichen zusammengefasst. Ein Kommando kann in mehreren Anwendungsbereichen vorkommen.

Jeder Operand besitzt einen Namen und mindestens einen möglichen Operandenwert. Einem Operandenwert können Operanden hierarchisch untergeordnet sein. In diesem Fall leitet der Operandenwert eine Struktur ein, die alle ihm untergeordneten Operanden enthält. Die Operanden der Struktur werden in Klammern eingeschlossen. Innerhalb einer Struktur können wiederum Strukturen eingeleitet werden, was Strukturschachtelung genannt wird.

*Beispiel für eine Liste:*

Format (Ausschnitt von SHOW-FILE-ATTRIBUTES):

```
,SELECT = *ALL / [*BY-ATTRIBUTES](...)  
      [*BY-ATTRIBUTES](...  
      | ,FILE-STRUCTURE = *ANY / list-poss(5): *PAM / *SAM / *ISAM / *BTAM / *NONE
```

Eingabe:

```
/show-file-attributes select=*by-attributes(file-structure=(*sam,*isam))
```

Der Operand FILE-STRUCTURE bezeichnet eine bestimmte Dateieigenschaft und ist dem Operandenwert (der Struktur) \*BY-ATTRIBUTES untergeordnet.

Die gleichzeitige Angabe mehrerer Operandenwerte (Liste) kann zugelassen sein. Operandenwerte, die als Listenelement zugelassen sind, werden im Format nach „list-poss(n):“ (siehe SDF-Syntaxdarstellung, [Seite 32](#)) aufgezählt, wobei n die maximal zulässige Anzahl von Listenelementen angibt.



*Beispiel für eine Liste:*

Format (Ausschnitt von SHOW-FILE-ATTRIBUTES):

**FILE-STRUCTURE = \*ANY / list-poss(5): \*PAM / \*SAM / \*ISAM / \*BTAM / \*NONE**

Eingabe:

```
/show-file-attributes select=*by-attributes(file-structure=(*sam,*isam))
```

Der Operand FILE-STRUCTURE bezeichnet die Dateieigenschaften SAM und ISAM.

Die meisten Operanden sind bereits mit einem Operandenwert vorbesetzt, dem Default-Wert. Dieser wird angenommen, falls keine explizite Angabe gemacht wird (siehe Default-Werte, [Seite 96](#)).

## Konventionen für die Eingabe

Bei der Eingabe von Kommandos und Anweisungen ist Folgendes zu beachten:

- Eingaben werden von SDF gemäß der Standard-Codetabelle **EBCDIC.DF.03** interpretiert (z.B. bei der Umsetzung Groß-/Kleinschreibung).

Bei Einsatz von XHCS<sup>1</sup> können erweiterte Zeichensätze (CCS<sup>2</sup>) verwendet werden. Die Codierung der Zeichen Dollar (\$), Nummernzeichen (#), Klammeraffe (@), Ausrufezeichen (!), Zirkumflex (^), runde und eckige Klammern, Größer- und Kleinerzeichen, Stern (\*), Schrägstrich (/), Doppelpunkt, Komma, Semikolon, Hochkomma und Anführungszeichen darf in einem erweiterten Zeichensatz nicht von der Codierung der Standard-Codetabelle abweichen.

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 Verwendung eines erweiterten Zeichensatzes sind folgende Fälle zu unterscheiden:

- Eingabe von der Datensichtstation:

Bei der Eingabe von Kommandos wird das CCS aus dem Benutzereintrag verwendet, wenn mit dem Kommando MODIFY-TERMINAL-OPTIONS der 8-Bit-Modus eingeschaltet wurde.

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

- Eingabe aus Datei oder PLAM-Bibliothekselement:

Bei der Eingabe von Kommandos wird das CCS der Eingabedatei bzw. der PLAM-Bibliothek verwendet. Das CCS des PLAM-Bibliothekselements wird verwendet, wenn SYSDTA mit dem Kommando ASSIGN-SYSDTA zugewiesen wurde.

Bei der Eingabe von Anweisungen muss das im jeweiligen Makroaufruf angegebene CCS mit dem CCS für SYSDTA-Eingaben übereinstimmen, sonst wird die Anweisung abgewiesen.

Während einer Prozedurunterbrechung wird das CCS verwendet, das für die Eingabe von der Datensichtstation eingestellt ist.

---

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

<sup>2</sup> Coded Character Set

- Eingabe aus einer S-Variablen:  
Ist SYSDTA einer zusammengesetzten S-Variablen zugewiesen, wird im Dialogbetrieb das CCS verwendet, das für die Eingabe von der Datensichtstation eingestellt ist. Im Stapelbetrieb wird das CCS **EDF03IRV** verwendet.
- In den Makroaufrufen CMD und TRCMD werden erweiterte Zeichen **nicht** unterstützt.
- Erstes Zeichen einer Kommandoingabe ist der Schrägstrich „/“; vor einer Anweisung stehen zwei Schrägstriche „//“.  
Bei Eingabe von der Datensichtstation werden diese Kennzeichen als Eingabeaufforderung vom System vorgegeben. Bei der Eingabe aus Prozedurdateien müssen diese Kennzeichen in den Eingabesätzen enthalten sein.
- Zwischen Schrägstrich und dem Kommandonamen kann eine Marke stehen. Sie kennzeichnet die Kommandozeile als Sprungziel innerhalb von Prozeduren. Mindestens ein Leerzeichen trennt die Marke von dem Kommandonamen. Eine Marke hat in S- und Nicht-S-Prozeduren unterschiedliche Formate:
  - Die S-Marke besteht aus einem bis zu 255 Zeichen langen Namen (entspricht <structured-name 1..255>), dem ein Doppelpunkt folgt. S-Marken gehören nicht zum Kommando. Sie werden nur von SDF-P ausgewertet und können nur in S-Prozeduren verwendet werden.
  - Die Nicht-S-Marke besteht aus einem führenden Punkt und aus einem bis zu acht Zeichen langen Namen (entspricht <name 1..8>).
- Der Kommando- bzw. Anweisungsname wird von nachfolgenden Operanden mit mindestens einem Leerzeichen getrennt.
- Ein Operand wird vom nächsten Operanden mit einem Komma getrennt.
- Operandenwerte innerhalb einer Liste werden mit einem Komma getrennt. Wird mehr als ein Listenelement angegeben, so ist die Aufzählung in runde Klammern einzuschließen.
- Operanden können entweder als Schlüsselwortoperanden angegeben werden oder als Stellungsoperanden. Bei der Angabe als Schlüsselwortoperand wird dem Operandenamen mit einem Gleichheitszeichen der Operandenwert zugeordnet.  
Bei der Angabe als Stellungsoperand wird nur der Operandenwert angegeben. Der Operand wird dann über die relative Position des Wertes innerhalb der Eingabe zugeordnet (siehe [Seite 97](#)).
- Weitere Leerzeichen zwischen Schlüsselwörtern, variablen Operandenwerten, Kommata und Gleichheitszeichen sind zu Dokumentationszwecken möglich und werden ignoriert.
- Zeichenfolgen, die in Anführungszeichen eingeschlossen sind, werden als Kommentare interpretiert und ignoriert. Kommentare können wie weitere Leerzeichen verwendet werden, vor Marken sind sie jedoch nicht erlaubt.

- Zeilenende-Kommentar:  
In S-Prozeduren leitet die Zeichenfolge &\* einen Zeilenende-Kommentar ein, d.h. alle nachfolgenden Zeichen bis zum Zeilenende werden ignoriert. Insbesondere werden Fortsetzungszeichen, Semikolon und &-Ausdrücke ignoriert.
- Fortsetzungszeilen:  
Ein Bindestrich als letztes Zeichen eines Eingabesatzes bzw. auch mit nachfolgenden Leerzeichen wird als Fortsetzungszeichen interpretiert. Die Folgezeile ist die Fortsetzungszeile des begonnenen Kommandos bzw. der Anweisung.  
Bei Eingabe von der Datensichtstation fordert das System mit „/“ bzw. „//“ zur Fortsetzung der Eingabe auf. Bei Eingabe aus Prozedur- oder ENTER-Dateien muss die Fortsetzungszeile mit „/“ bzw. „//“ beginnen.  
Das Fortsetzungszeichen für Kommandos muss innerhalb der Spalten 2 und 72 stehen, wenn CONTINUATION=\*NEW-MODE eingestellt ist (s. SHOW-SDF-OPTIONS bzw. MODIFY-SDF-OPTIONS). Mit CONTINUATION=\*OLD-MODE muss das Fortsetzungszeichen genau auf Spalte 72 stehen.  
Auf Spalte 72 nachfolgende Zeichen werden ignoriert.  
Die Gesamtlänge eines Kommandos (mit seinen Folgezeilen) beträgt 16364 Byte einschließlich Leerzeichen und Kommentaren. Für die noch unterstützten ISP-Kommandos beträgt die Gesamtlänge 4096 Byte.  
  
Eingabesätze für eine Anweisung können länger als 72 Zeichen sein, wobei das Fortsetzungszeichen ab Spalte 2 angegeben werden kann. Die Länge einer Anweisung kann maximal 16364 Byte betragen.  
  
Bei Eingabe von der Dialogstation ist die Länge der Eingabesätze und die Position eines Fortsetzungszeichens nur von der Größe des Eingabepuffers der Dialogstation abhängig, sie beträgt jedoch maximal 16364 Byte.  
  
In S-Prozeduren kann mit dem Kommando SET-PROCEDURE-OPTIONS INPUT-FORMAT=\*FREE-RECORD-LENGTH die Länge eines Eingabesatzes unter Berücksichtigung der Maximallänge von 16364 Byte frei vereinbart werden. Ein Fortsetzungszeichen kann ab Spalte 2 angegeben werden.
- Bei Eingabe aus Nicht-S-Prozedur- bzw. ENTER-Dateien müssen Schlüsselwörter großgeschrieben werden. Dies gilt auch für Werte von Prozedurparametern, S-Variablen und Jobvariablen, wenn sie Teile von Kommandos bzw. Anweisungen ersetzen sollen.
- Bei Verwendung eines Strichpunkts zwischen zwei Kommandos können innerhalb eines Eingabesatzes (einschl. Folgezeilen) mehrere Kommandos angegeben werden. Diese Eingabemöglichkeit ist nur im Dialog und innerhalb von *S-Prozeduren* erlaubt.

## 3.2 SDF-Syntaxdarstellung

Nachfolgendes Beispiel zeigt die Syntaxdarstellung eines Kommandos in einem Handbuch. Das Kommandoformat besteht aus einem Feld mit dem Kommandonamen. Anschließend werden alle Operanden mit den zulässigen Operandenwerten aufgelistet. Struktureinleitende Operandenwerte und die von ihnen abhängigen Operanden werden zusätzlich aufgelistet.

HELP-SDF	Kurzname: HPSDF
<b>GUIDANCE-MODE = *NO / *YES</b> <b>,SDF-COMMANDS = *NO / *YES</b> <b>,ABBREVIATION-RULES = *NO / *YES</b> <b>,GUIDED-DIALOG = *YES (...)</b> <b>*YES(...)</b>   <b>SCREEN-STEPS = *NO / *YES</b> <b>,SPECIAL-FUNCTIONS = *NO / *YES</b> <b>,FUNCTION-KEYS = *NO / *YES</b> <b>,NEXT-FIELD = *NO / *YES</b> <b>,UNGUIDED-DIALOG = *YES (...) / *NO</b> <b>*YES(...)</b>   <b>SPECIAL-FUNCTIONS = *NO / *YES</b> <b>,FUNCTION-KEYS = *NO / *YES</b>	

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> $[:cat:][\$user.] \left\{ \begin{array}{l} \text{datei} \\ \text{datei(nr)} \\ \text{gruppe} \\ \text{gruppe} \left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\} \end{array} \right\}$ <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)

<b>Datentyp</b>	<b>Zeichenvorrat</b>	<b>Besonderheiten</b>
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	Eingabeformat: $[[C]][V][m]m.naso[']$ <div style="text-align: right; margin-right: 20px;"> <math>\left. \begin{array}{l}   \\   \\   \end{array} \right\} \begin{array}{l} \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	alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
time	0...9 Strukturkennzeichen: Doppelpunkt	Angabe einer Tageszeit Eingabeformat: $\left. \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}$ hh: Stunden } führende Nullen können mm: Minuten } weggelassen werden ss: Sekunden }
vsn	a) A...Z 0...9  b) A...Z 0...9 \$, #, @	a) Eingabeformat: pvsid.folgenummer max. 6 Zeichen; pvsid: 2-4 Zeichen; Eingabe von PUB nicht erlaubt folgenummer: 1-3 Zeichen  b) max. 6 Zeichen; PUB darf vorangestellt werden, dann dürfen jedoch nicht \$, #, @ folgen.

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>	<p>Erweitert die Angabemöglichkeiten für einen Datentyp.</p> <p><i>-compl</i>    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> <p><i>-low</i>      Groß- und Kleinschreibung wird unterschieden.</p> <p><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.</p> <p><i>-under</i>    Erlaubt Unterstriche ' _ ' bei den Datentypen <i>name</i> und <i>composed-name</i>.</p>

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> ,...>  -s	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 <a href="#">Seite 40</a> ).  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										
with (Forts.) -wild- constr(n)	<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"> <thead> <tr> <th>Platzhalter</th> <th>Bedeutung</th> </tr> </thead> <tbody> <tr> <td>*</td> <td>Entspricht der Zeichenfolge, die durch den Platzhalter * in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td>Punkt am Ende</td> <td>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>/ oder ?</td> <td>Entspricht dem Zeichen, das durch den Platzhalter / oder ? in der Auswahlzeichenfolge ausgewählt wird.</td> </tr> <tr> <td>&lt;n&gt;</td> <td>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: In der Auswahlzeichenfolge werden alle Platzhalter von links nach rechts aufsteigend nummeriert (globaler Index). Gleiche Platzhalter in der Auswahlzeichenfolge werden zusätzlich von links nach rechts aufsteigend nummeriert (platzhalter-spezifischer Index). 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]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]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]m.naso[']</code> Angaben zum Datentyp product-version müssen den Korrekturstand (und damit auch den Freigabestand) enthalten.
-man	Eingabeformat: <code>[[C]][V][m]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)

### 3.3 Sprache für Ausgaben von SDF

Alle Ausgaben von SDF werden in der Sprache vorgenommen, die auch für die Ausgabe von System-Meldungen eingestellt ist. In den von Fujitsu Siemens Computers ausgelieferten Syntaxdateien sind Texte für die Sprachschlüssel D (deutsch) und E (englisch) enthalten. Diese Texte sind:

- kommando- bzw. anweisungs-spezifische Hilfetexte
- Hilfetexte zu Datentypen
- SDF-spezifische Meldungen (z.B Aufforderung zur Kommandoeingabe „%KDO:“ bzw. „%CMD:“)
- Bezeichner, die in der Dialogführung verwendet werden (Namen der Eingabefelder, Bezeichnung der Datentypen und ihrer Zusätze)

Die Systemsprache wird allgemein bei Systemgenerierung festgelegt. Zusätzlich kann die Systemverwaltung für jede Benutzererkennung die Sprache im Benutzerkatalog festlegen. Dieser Eintrag hat Vorrang vor der allgemeinen Einstellung der Systemgenerierung. Mit dem Kommando SHOW-USER-ATTRIBUTES kann sich der Benutzer über die für ihn geltende Einstellung informieren.

Der Benutzer kann die Systemsprache auch taskspezifisch mit dem Kommando MODIFY-MSG-ATTRIBUTES einstellen.

Ist der eingestellte Sprachschlüssel in den aktivierten Syntaxdateien der Task nicht definiert, d.h., die Globalinformationen der aktivierten Syntaxdateien enthalten keine allgemeinen Texte für diesen Sprachschlüssel, verwendet SDF die Sprache, die bei der Systemgenerierung festgelegt wurde. Mit der Meldung CMD0159 wird der Benutzer darauf hingewiesen.

Wurde ein Sprachschlüssel eingestellt, für den Globalinformationen in den aktivierten Syntaxdateien enthalten sind, gibt SDF die kommando- und anweisungs-spezifischen Hilfetexte aus, die für diesen Sprachschlüssel definiert wurden. Fehlende Hilfetexte für diesen Sprachschlüssel gibt SDF, soweit vorhanden, in der bei der Systemgenerierung festgelegten Sprache aus.

## 3.4 Kommando-Returncode

SDF liefert dem Benutzer Informationen über die Analyse der Kommandoeingabe und die -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.

### Aufbau des Kommando-Returncodes

Der Kommando-Returncode besteht aus drei Teilen:

- Maincode, der einem Meldungsschlüssel entspricht, über den mit dem Kommando HELP-MSG-INFORMATION detaillierte Informationen abgefragt werden können.
- Subcode1, der die aufgetretene Fehlersituation in eine Fehlerklasse einordnet, aus der abgeleitet werden kann, wie schwer wiegend ein Fehler ist. Der Wert von Subcode1 wird *dezimal* ausgegeben. Folgende fünf Fehlerklassen sind im BS2000 definiert:
  - Klasse A: kein Fehler  
Der Wert ist null. Es kann normal weitergearbeitet werden.
  - Klasse B: Syntaxfehler  
Der Wert ist eine Zahl zwischen 1 und 31. Das Kommando wurde syntaktisch falsch eingegeben. Eine Wiederholung der Eingabe ist nur nach Korrektur des Syntaxfehlers sinnvoll.
  - Klasse C: interner Fehler (Systemfehler)  
Der Wert ist 32. Eine Wiederholung der Eingabe ist nur sinnvoll, wenn der interne Fehler behoben wurde.
  - Klasse D: Fehler, die keiner anderen Fehlerklasse zuzuordnen sind  
Der Wert ist eine Zahl zwischen 64 und 127. Zur Bestimmung der weiteren Vorgehensweise sollte der Maincode ausgewertet werden.
  - Klasse E: Kommando kann vorübergehend nicht ausgeführt werden  
Der Wert ist eine Zahl zwischen 128 und 130. Die Eingabe kann unverändert wiederholt werden. Das Kommando kann nach einer Wartezeit wieder ausgeführt werden. Die Dauer der Wartezeit wird eingeteilt in kurzfristig, langfristig und unbefristet. Kurzfristig entspricht dem Wert 128 und bedeutet, dass ein Warten im Dialog als sinnvoll angesehen wird. Langfristig entspricht dem Wert 129 und bedeutet, dass ein Warten im Stapelbetrieb als sinnvoll angesehen wird.  
  
Unbefristet entspricht dem Wert 130 und bedeutet, dass es unsicher ist, ob der Fehler überhaupt behoben wird.

- 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 mit Wert 1 zeigt an, dass die angeforderte Leistung bereits vor Absetzen des Kommandos bestand. Subcode2 mit Wert 2 zeigt eine Sondersituation an und ist als Warnung einzustufen.

Der Kommando-Returncode kann nur mit SDF-P-Mitteln in S-Prozeduren und Dialogblöcken ausgewertet werden (siehe Handbuch „SDF-P“ [6]). Zur Auswertung siehe auch Beschreibung des Kommandos IF-BLOCK-ERROR und der Builtin-Funktionen MAINCODE, SUBCODE1 und SUBCODE2 im Handbuch „Kommandos, Band 6“ [2].

### Darstellung von 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. es ist 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 es werden die Meldungsschlüssel der garantierten Meldungen nach dem Bedeutungstext aufgezählt. Die Bedeutung dieser Meldungsschlüssel kann sich der Benutzer mit dem Kommando HELP-MSG-INFORMATION 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.

Die nachfolgende Tabelle enthält 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. der Ausführungsmodul selbst keinen Kommando-Returncode zurückgibt. In diesem Fall sind in der Kommandobeschreibung keine spezifischen Kommando-Returncodes enthalten.

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	0	CMD0002	Kommandoausführung mit Warnung <sup>4)</sup>
1	0	CMD0026	Kommando nicht mehr verfügbar
2	0	CMD0093	Prozedur im Testmodus beendet <sup>1)</sup>
2	0	CMD0201	Dateiende erreicht <sup>1)</sup>
2	0	CMD0214	Programmende <sup>3)</sup>
	1	CMD0211	Übergabebereich von SDF zu klein
	1	CMD0202	Syntaxfehler <sup>5) 6)</sup>
	1	CMD0205	SPIN-OFF <sup>1) 2)</sup>
	1	CMD2201	interner Parameterfehler
	2	CMD0200	Kommando zurzeit nicht verfügbar
	2	CMD2202	Subsystem nicht definiert
	3	CMD2203	Fehler bei Installation
	32	CMD0221	interner SDF-Fehler
	32	CMD2009	Fehler bei Ausgabe in Variablen
	64	CMD0216	Erforderliches Privileg fehlt
	65	CMD2241	Subsystem nicht geladen
	66	CMD2242	keine Verbindung zum Subsystem
	128	CMD2280	Subsystem kurzfristig nicht verfügbar
	129	CMD2281	Subsystem langfristig nicht verfügbar
	130	CMD2282	Subsystem für unbestimmte Dauer nicht verfügbar
	130	NBR0748	Nach /SHUTDOWN ist an der Bedienstation keine Kommandoeingabe außer /SHOW-PENDING-MSG möglich

- 1) Die Kommando-Returncodes CMD0093, CMD0201 und CMD0205 sind nicht möglich, wenn das Kommando über den CMD-Makro abgesetzt wird.
- 2) Der Kommando-Returncode CMD0205 wird zurückgegeben, wenn Spin-Off ausgelöst wurde. Beispiel: Programmende mit dem Makroaufruf TERM mit UNIT=STEP und während des Programmlaufs wurde kein Kommando-Returncode mit dem Makroaufruf CMDRC gesetzt.

- 3) Der Kommando-Returncode CMD0214 kann von dem CMD-Makro nicht mehr an das aufrufende Programm zurückgegeben werden, da das Programm durch CMD-Makro-Ausführung bereits beendet wurde.
- 4) Der Kommando-Returncode CMD0002 wird von SDF zurückgegeben, wenn ein Kommando trotz fehlerfreier Ausführung Spin-Off auslöst. Das bisherige Spin-Off-Verhalten von Kommandos, und damit die Fehlerbehandlung in Prozeduren, werden dadurch kompatibel unterstützt. In S-Prozeduren, in denen ERROR-MECHANISM=\*BY-RETURNCODE gilt (siehe Kommando SET- bzw. MODIFY-PROCEDURE-OPTIONS), wird die Fehlerbehandlung nicht eingeleitet, da der zugehörige Subcode1 den Wert null besitzt.
- 5) Der Kommando-Returncode CMD0202 wird von SDF zurückgegeben, wenn ein Kommando bei fehlerhafter Ausführung keinen Spin-Off auslöst. CMD0202 zeigt Syntax- und Semantikfehler an, die bei der Kommandoausführung erkannt werden. Das bisherige Spin-Off-Verhalten von Kommandos und damit die Fehlerbehandlung in Prozeduren werden dadurch kompatibel unterstützt. In S-Prozeduren, in denen ERROR-MECHANISM=\*BY-RETURNCODE gilt (siehe Kommando SET- bzw. MODIFY-PROCEDURE-OPTIONS), wird die Fehlerbehandlung eingeleitet, da der zugehörige Subcode1 einen Wert ungleich null besitzt.
- 6) Der Kommando-Returncode CMD0202 wird von SDF sowohl bei Syntaxfehler als auch bei Semantikfehler von dem Ausführungsmodul zurückgegeben.

#### *Hinweise*

- In Benutzerprogrammen kann mit dem Makroaufruf CMDRC ein Kommando-Returncode gesetzt werden. Dieser bleibt erhalten bis zum nächsten CMDRC-Makroaufruf. Bei Programmbeendigung wird der aktuelle Kommando-Returncode aus dem Programm an den Aufrufer zurückgegeben und löst in S-Prozeduren bei Subcode1 ungleich null die Fehlerbehandlung aus, wenn ERROR-MECHANISM=\*BY-RETURNCODE vereinbart wurde (siehe Kommando SET- bzw. MODIFY-PROCEDURE-OPTIONS).
- Ab SDF-P V2.1A können im Programm gesetzte Returncodes nach jeder Anweisung wie Kommando-Returncodes ausgewertet werden (siehe Kommando BEGIN-BLOCK im Handbuch „SDF-P“ [6]).

---

## 4 Kommando- und Anweisungseingabe

Kommandos und Anweisungen können von der Datensichtstation im ungeführten oder geführten Dialog eingegeben werden. Aus dem ungeführten Dialog kann der Benutzer für eine Eingabe temporär in den geführten Dialog wechseln.

Kommandos und Anweisungen können auch in einer Datei oder einer zusammengesetzten S-Variablen abgespeichert werden. Die Eingabe erfolgt dann im Prozedur- oder Stapelbetrieb.

Innerhalb der Eingabe kann der Benutzer Ausdrücke verwenden, die SDF durch den Inhalt einer Jobvariablen oder S-Variablen oder in Prozeduren durch den Wert eines Prozedurparameters ersetzt (siehe [Abschnitt „Ausdrücke in der Eingabe ersetzen“ auf Seite 87](#)).

Die meisten der in diesem Kapitel gegebenen Informationen kann sich der Benutzer mit dem Kommando HELP-SDF ausgeben lassen (siehe [Seite 166](#)).

### 4.1 Eingabe im ungeführten Dialog

Es gibt zwei Formen des ungeführten Dialogs, die EXPERT-Form und die NO-Form. Die Einstellung erfolgt über die entsprechende Angabe im Operanden GUIDANCE des Kommandos MODIFY-SDF-OPTIONS. Ein Wechsel in eine andere Form des Dialogs ist möglich (siehe [Seite 57](#) und [Seite 76](#)).

Sie unterscheiden sich vor allem dadurch, ob SDF einen Dialog zur Korrektur fehlerhafter Eingaben führt oder nicht. Sie werden ausgewählt mit dem Operanden GUIDANCE des Kommandos MODIFY-SDF-OPTIONS.

### 4.1.1 EXPERT-Form

Nach der LOGON-Verarbeitung ist die Benutzerführung auf die EXPERT-Form voreingestellt, falls in den Globalinformationen der aktivierten Syntaxdateien keine andere Festlegung getroffen wurde. Aus einer anderen Dialogart kann der Benutzer mit MODIFY-SDF-OPTIONS GUIDANCE=\*EXPERT in die EXPERT-Form wechseln.

SDF fordert mit „/“ zur Kommandoeingabe auf (bzw. mit „//“ zur Anweisungseingabe). Bei fehlerhaften Eingaben werden nur Fehlermeldungen ausgegeben. Einen Korrekturdialog führt SDF nicht. Die Wiedererlangung der Eingabe mit RESTORE-SDF-INPUT ist möglich. Der Benutzer kann zur Eingabe in den temporär geführten Dialog wechseln (siehe [Seite 76](#)).

### 4.1.2 NO-Form

In die NO-Form wird mit MODIFY-SDF-OPTIONS GUIDANCE=\*NO gewechselt.

SDF fordert mit „%KDO:“ zur Kommandoeingabe auf (bzw. mit „%ANW:“ zur Anweisungseingabe). Bei fehlerhafter Eingabe gibt SDF Fehlermeldungen aus und beginnt einen Korrekturdialog. Mit der Eingabe eines Fragezeichens kann der Benutzer für die aktuelle Eingabe in den temporär geführten Dialog wechseln (siehe [Seite 76](#)). Mit dem Kommando RESTORE-SDF-INPUT kann die letzte Eingabe angezeigt werden (siehe [Seite 199](#)).

### 4.1.3 Spezielle Eingaben

- ! als Operandenwert setzt für diesen Operanden den Default-Wert. Nachfolgende Zeichen brauchen nicht gelöscht zu werden.
- ! als erstes Zeichen des Kommando- bzw. Anweisungsnamens definiert die angegebenen Operandenwerte als task-spezifische Default-Werte des Kommandos bzw. der Anweisung. Das Kommando bzw. die Anweisung wird nicht ausgeführt.
- ? als Operandenwert gibt Informationen über die möglichen Operandenwerte aus. In der NO-Form bekommt der Benutzer die Eingabe wieder angezeigt und wird zur Eingabe der Operanden aufgefordert. „?“ nach einem Kommandonamen wechselt in den temporär geführten Dialog (siehe [Seite 76](#)).
- ?? als Operandenwert liefert Informationen zu Datentypen, die als Operandenwert zugelassen sind.
- ^ als Operandenwert eines „geheimen“ Operanden fordert ein dunkelgesteuertes Eingabefeld für die verdeckte Eingabe des Operandenwertes an.
- als letztes Zeichen zeigt an, dass das Kommando bzw. die Anweisung in der nächsten Eingabezeile fortgesetzt wird.

**LZF**-Taste

löscht ab der Schreibmarke alle Zeichen der Eingabezeile.

**LZE**-Taste

ermöglicht geblockte Eingabe (siehe [Abschnitt „Geblockte Eingaben“ auf Seite 55](#)).

In ein und demselben Kommando kann der gleiche Operand mehrmals angegeben werden. Es wird jedoch nur die letzte Angabe zu diesem Operanden akzeptiert.

Bei Angabe des Operandenwertes \*SECRET (sowohl explizit als auch implizit durch Vorbesetzung) gibt SDF ein dunkelgesteuertes Eingabefeld für die verdeckte Eingabe des Operandenwertes aus (siehe [Seite 76](#)).

### 4.1.4 Funktionstasten

Bestimmte Funktionen kann der Benutzer durch Drücken einer Funktionstaste auslösen. Die Wirkung der Funktionstasten ist abhängig von der Einstellung der SDF-Option FUNCTION-KEYS.

Es existieren zwei Belegungsmodi: der bisherige Old-Modus (\*OLD-MODE) und der Style-Guide-Modus (\*STYLE-GUIDE-MODE), der mehr Funktionalität bietet. Das Terminal oder die Terminalemulation müssen jedoch den Style-Guide-Modus unterstützen und entsprechend generiert sein.

<b>*OLD-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
[K1]	Exit-Funktion Beendet das aktuell laufende Programm. In einer Meldung wird der Benutzer gefragt, ob das Programm beendet werden soll. Während eines Korrekturdialogs bei der Kommando- bzw. Anweisungseingabe (siehe NO-Form) beendet die Funktion nur den Korrekturdialog. Entspricht [F3] im Style-Guide-Modus.
[K2]	Unterbrechungs-Funktion Unterbricht ein aktuelles Programm oder eine Prozedur, oder bricht die Ausgabe eines Kommandos ab.
[F1]	Exit-all-Funktion Beendet das aktuell laufende Programm. In einer Meldung wird der Benutzer gefragt, ob das Programm beendet werden soll. Während eines Korrekturdialogs bei der Kommando- bzw. Anweisungseingabe (siehe NO-Form) beendet die Funktion nur den Korrekturdialog. Entspricht [F6] im Style-Guide-Modus.

Tabelle 4: Funktionstastenbelegung (Old-Modus) im ungeführten Dialog

<b>*STYLE-GUIDE-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
[K2]	Unterbrechungs-Funktion Unterbricht ein aktuelles Programm oder eine Prozedur, oder bricht die Ausgabe eines Kommandos ab.
[F1]	Hilfe-Funktion Wechselt in den temporär geführten Dialog. Entspricht der Eingabe von „?“.

Tabelle 5: Funktionstastenbelegung (Style-Guide-Modus) im ungeführten Dialog (Teil 1 von 2)

*STYLE-GUIDE-MODE	
Funktionstaste	Wirkung
[F3]	Exit-Funktion Beendet das aktuell laufende Programm. In einer Meldung wird der Benutzer gefragt, ob das Programm beendet werden soll. Während eines Korrekturdialogs bei der Kommando- bzw. Anweisungseingabe (siehe NO-Form) beendet die Funktion nur den Korrekturdialog. Entspricht [K1] im Old-Modus.
[F6]	Exit-all-Funktion Beendet das aktuell laufende Programm. In einer Meldung wird der Benutzer gefragt, ob das Programm beendet werden soll. Während eines Korrekturdialogs bei der Kommando- bzw. Anweisungseingabe (siehe NO-Form) beendet die Funktion nur den Korrekturdialog. Entspricht [F1] im Old-Modus.
[F9]	Gibt das zuletzt eingegebene Kommando bzw. Anweisung erneut aus. Entspricht der Eingabe von „RESTORE-SDF-INPUT“ ohne Operanden (voreingestellt ist INPUT=*LAST-CMD bzw. *LAST-STMT).
[F12]	Cancel-Funktion Beendet das aktuell laufende Programm. In einer Meldung wird der Benutzer gefragt, ob das Programm beendet werden soll. Während eines Korrekturdialogs bei der Kommando- bzw. Anweisungseingabe (siehe NO-Form) beendet die Funktion nur den Korrekturdialog.

Tabelle 5: Funktionstastenbelegung (Style-Guide-Modus) im ungeführten Dialog (Teil 2 von 2)

## Hinweis

Im Style-Guide-Modus werden die Funktionstasten [K1], [K3], [F2], [F4], [F5], [F7], [F8], [F10], [F11] und [F13] bis [F24] nicht unterstützt. Das Drücken einer nicht unterstützten Funktionstaste bewirkt die Ausgabe einer Fehlermeldung.

## 4.1.5 Geblockte Eingaben

In der EXPERT- und NO-Form des ungeführten Dialogs können mehrere Kommandos bzw. Anweisungen gleichzeitig abgeschickt werden (geblockte Eingabe). Bei der geblockten Eingabe werden eine oder mehrere logische Zeilen eingegeben, die mit dem logischen Zeilenende ([LZE]-Taste) abgeschlossen sind. Jede logische Eingabezeile enthält ein Kommando bzw. eine Anweisung. Die geblockte Eingabe kann mit der Endemarke ([EM]-Taste) abgeschlossen werden.

Nach Abschicken des Eingabeblockes gibt SDF jedes analysierte Kommando bzw. Anweisung vor der Ausführung noch einmal **unverändert** am Bildschirm aus. Dabei wird die Zeichenfolge „(BL)“ vorangestellt.

Ist ein Kommando bzw. eine Anweisung fehlerhaft und wird der Fehler nicht im Dialog kor-

rigiert, so werden nachfolgende Kommandos bzw. Anweisungen nicht mehr analysiert. Die verbleibenden logischen Eingabezeilen gibt SDF mit der Meldung CMD0122 **unverändert** am Bildschirm aus. Kennwörter werden dabei im Protokoll nicht ausgeblendet.

Innerhalb einer logischen Eingabezeile kann der Benutzer auch mehrere Kommandos oder Anweisungen angeben, die mit einem Strichpunkt von einander zu trennen sind. Der Inhalt einer logischen Eingabezeile kann nur einem logischen Eingabestrom zugeordnet sein: Kommandos, die von SYSCMD gelesen werden, können nicht mit Anweisungen, die von SYSSTMT gelesen werden, innerhalb einer logischen Eingabezeile mit Strichpunkt verketten werden.

## Beispiel

```

/add-file-link link=sortin01,file-name=datei.1<
add-file-link link=sortin02,file-name=datei.2<
add-file-link link=sortout,file-name=out.sort1-2<
start-sort;sort-record;end<
show-file-attr out.sort1-2;show-file-attr datei.;show-file-link link=sort*
(BL) /add-file-link link=sortin01,file-name=datei.1
(BL) add-file-link link=sortin02,file-name=datei.2
(BL) add-file-link link=sortout,file-name=out.sort1-2
(BL) start-sort
% BLS0523 ELEMENT 'SRT80', VERSION '078' FROM LIBRARY ':20SH:$TSOS.SYSLNK.SORT.
078' IN PROCESS
% BLS0524 LLM 'SRT80', VERSION '078' OF '2001-08-24 15:20:53' LOADED
% BLS0551 COPYRIGHT (C) FUJITSU SIEMENS COMPUTERS GMBH 2001.ALL RIGHTS RESERVED
% SRT1001 18:49:13/000000.00 SORT/MERGE STARTED, VERSION 07.8A00/BS2000V14.0
% SRT1130 PLEASE ENTER SORT STATEMENTS
(BL) sort-record
(BL) end
% SRT1016 SORT/MERGE INPUT RECORDS:.....688 (FROM 01)
% SRT1016 SORT/MERGE INPUT RECORDS:.....2.003 (FROM 02)
% SRT1017 RECORDS TO BE SORTED/MERGED:.....2.691
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....2.691
% SRT1002 18:49:14/000000.19 SORT/MERGE COMPLETED
(BL) show-file-attr out.sort1-2
% 66 :20SG:$USER1.OUT.SORT1-2
%:20SG: PUBLIC: 1 FILE RES= 66 FRE= 2 REL= 0 PAGES
(BL) show-file-attr datei.
%PLEASE ACKNOWLEDGE
% 21 :20SG:$USER1.DATEI.1
% 48 :20SG:$USER1.DATEI.2
%:20SG: PUBLIC: 2 FILES RES= 69 FRE= 4 REL= 0 PAGES
(BL) show-file-link link=sort*
%-- LINK-NAME ----- FILE-NAME -----
% SORTOUT :20SG:$USER1.OUT.SORT1-2
% SORT01 :20SG:$USER1.DATEI.1
% SORT02 :20SG:$USER1.DATEI.2
/

```



## 4.2 Eingabe im geführten Dialog

Der geführte Dialog ist in vollem Umfang nur für Kommandos und Anweisungen möglich, die in den zugewiesenen Syntaxdateien definiert sind und die Operanden besitzen, für die die Dialogführung zugelassen ist.

Für die Eingabe sicherheitsrelevanter Operandenwerte (wie Kennwörter) werden dunkelgesteuerte Eingabefelder ausgegeben.

Im Dialog mit minimaler oder mittlerer Führung kann der Benutzer weitere Erläuterungen über einen Operanden anfordern.

Gibt der Benutzer ein Kommando bzw. eine Anweisung mit allen erforderlichen Operandenwerten in der NEXT-Zeile eines Fragebogens oder Menüs an, so wird dieses ohne weiteren Dialog ausgeführt.

### 4.2.1 Stufen der Dialogführung

Es gibt drei Stufen des geführten Dialogs. Sie unterscheiden sich im Umfang der von SDF ausgegebenen Informationen. Sie werden ausgewählt mit dem Operanden GUIDANCE des Kommandos MODIFY-SDF-OPTIONS. Durch Angabe eines Fragezeichens in der NEXT-Zeile kann der Benutzer temporär in die nächsthöhere Führungsstufe wechseln.

#### Maximale Benutzerführung

Mit MODIFY-SDF-OPTIONS GUIDANCE=\*MAXIMUM wird in die maximale Benutzerführung gewechselt.

SDF gibt die Default-Werte der wahlweisen Operanden, Aufzählung sämtlicher Operandenwerte mit Zusatzerläuterungen und Hilfetexte für Anwendungsbereiche, Kommandos, Anweisungen und Operanden aus.

## Mittlere Benutzerführung

Mit `MODIFY-SDF-OPTIONS GUIDANCE=*MEDIUM` wird in die mittlere Benutzerführung gewechselt.

SDF gibt die Default-Werte der wahlweisen Operanden und Aufzählungen sämtlicher zulässiger Operandenwerte ohne Zusatzerläuterungen aus. Hilfetexte gibt SDF nur für Anwendungsbereiche, Kommandos und Anweisungen aus.

## Minimale Benutzerführung

Mit `MODIFY-SDF-OPTIONS GUIDANCE=*MINIMUM` wird in die minimale Benutzerführung gewechselt.

SDF gibt nur die Default-Werte der wahlweisen Operanden aus. Aufzählungen sämtlicher zulässiger Operandenwerte und Hilfetexte gibt SDF nicht aus.

## 4.2.2 Aufbau der Bildschirmmasken

Im geführten Dialog bietet SDF dem Benutzer eine Bildschirmmaske mit gewünschten Informationen bzw. mit Eingabefeldern an. Der Bildschirm ist dabei immer in drei Bereiche unterteilt, die je nach Stand der Dialogführung unterschiedliche Inhalte haben können.

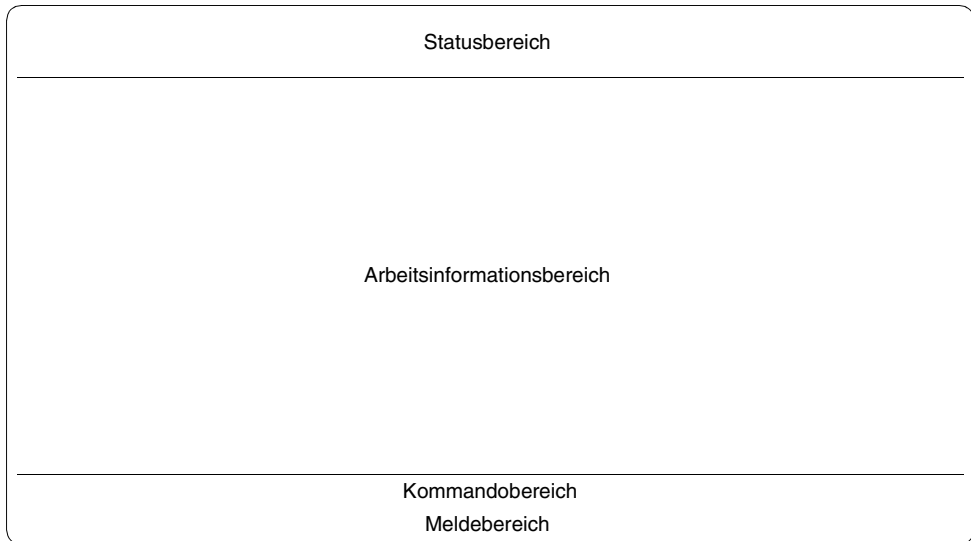


Bild 1: Bildschirmaufbau

Der Statusbereich kann enthalten:

- den Namen des ausgewählten Anwendungsbereiches.
- zusätzlich den Namen des ausgewählten Kommandos.
- zusätzlich den Namen eines struktureinleitenden Operandenwertes, falls die zugehörigen Operanden angezeigt werden.
- die bisher akzeptierten Operanden; verkettet zu einem String, der linksbündig abgeschnitten wird, falls er länger als zwei Anzeigezeilen wird.

Der Arbeitsinformationsbereich kann enthalten:

- die Auflistung aller zugelassenen Anwendungsbereiche.
- die Auflistung aller zugelassenen Kommandos eines ausgewählten Anwendungsbereiches.
- die Namen der möglichen Operanden eines ausgewählten Kommandos mit einer anschließenden Eingabezeile. Die Eingabezeile ist bei wahlfreien Operanden mit dem Default-Wert vorbesetzt. Je nach Stufe der Benutzerführung werden zusätzlich Informationen über den Operanden angezeigt.

Der Kommandobereich enthält:

- die so genannte NEXT-Zeile. Nach dem Operanden NEXT wird eine Eingabezeile für zulässige Anweisungen zur Steuerung der Menüführung angezeigt. Die Eingabezeile ist bereits mit einer Anweisung vorbesetzt.
- Zusatzinformationen werden in den nachfolgenden Zeilen in Abhängigkeit von der SDF-Option FUNCTIONS-KEYS angezeigt.
  - Im Old-Modus (FUNCTIONS-KEYS=\*OLD-MODUS) werden alle zulässigen Steueranweisungen angezeigt. Funktionstasten, die einer zulässigen Steueranweisung entsprechen werden ebenfalls angezeigt.
  - Im Style-Guide-Modus (FUNCTIONS-KEYS=\*STYLE-GUIDE-MODE) werden die zulässigen Steueranweisungen nur in der Führungsstufe GUIDANCE=\*MEDIUM bzw. \*MAXIMUM angezeigt. Die zulässigen Funktionstasten werden in allen Führungsstufen in einer eigenen Zeile nach dem Schlüsselwort KEYS angezeigt.

Der Meldungsbereich enthält gegebenenfalls zusätzlich die von SDF zuletzt ausgegebene Fehlermeldung.

Die ausgegebenen Bildschirme werden im Folgenden als Menüs bezeichnet, wenn im Arbeitsinformationsbereich nur Informationen über auszuwählende Anwendungsbereiche, Kommandos oder Anweisungen enthalten sind. Sie werden als Operandenfragebögen bezeichnet, wenn im Arbeitsinformationsbereich Operanden aufgelistet werden, für die die Angabe eines Wertes möglich ist. Angezeigte Menüs oder Operandenfragebögen können auf mehrere Bildschirme aufgeteilt sein (abhängig vom auszugebenden Umfang). Der Bildschirmwechsel erfolgt mit der Steueranweisung „+“ bzw. „-“ in der NEXT-Zeile des Kommandobereichs.

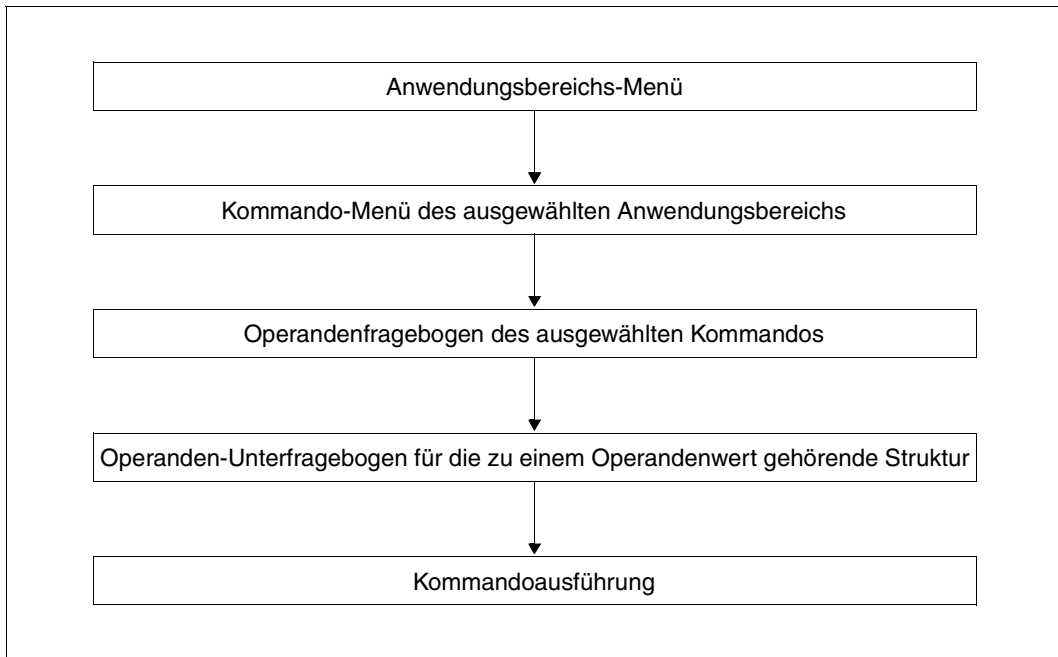


Bild 2: Weg vom Anwendungsbereichs-Menü zur Kommandoausführung

### 4.2.3 Spezielle Eingaben

- ! als Operandenwert setzt für diesen Operanden den Default-Wert. Nachfolgende Zeichen brauchen nicht gelöscht zu werden.
- ? als Operandenwert gibt Informationen über die möglichen Operandenwerte aus. In der NO-Form bekommt der Benutzer die Eingabe wieder angezeigt und wird zur Eingabe der Operanden aufgefordert. „?“ nach einem Kommandonamen wechselt in den temporär geführten Dialog (siehe [Seite 76](#)).
- ?? als Operandenwert liefert Informationen zu Datentypen, die als Operandenwert zugelassen sind.
- ^ als Operandenwert eines „geheimen“ Operanden fordert ein dunkelgesteuertes Eingabefeld für die verdeckte Eingabe des Operandenwertes an.
- ([<operand>,...]  
nach einem struktureinleitenden Operanden veranlasst die Ausgabe des Unterfragebogens für die Struktur. Angegebene Operandenwerte werden in den Unterfragebogen übernommen.

([<operand>,...])

nach einem struktureinleitenden Operanden unterdrückt die Ausgabe des Unterfragebogens für die Struktur. Die angegebenen Werte werden übernommen; für nicht angegebene Werte werden die jeweiligen Default-Wert gesetzt.

- als letztes Zeichen zeigt an, dass das Kommando bzw. die Anweisung in der nächsten Eingabezeile fortgesetzt wird.

**LZF**-Taste

löscht ab der Schreibmarke alle Zeichen der Eingabezeile.

## 4.2.4 Eingaben in die NEXT-Zeile

In die NEXT-Zeile können Anweisungen zur Steuerung der Menüführung oder ein Kommando bzw. eine Anweisung entsprechend der EXPERT-Form des ungeführten Dialogs erfolgen. Das Zeichen für das logische Zeilenende (LZE-Taste) ist als Eingabe nicht zulässig. Das Löschen von Zeichenfolgen hinter der gewünschten Eingabe ist mit der LZF-Taste möglich.

Die folgende Aufstellung enthält alle möglichen Eingaben. Die zulässige Menge ist abhängig vom angezeigten Bildschirminhalt; eine Auswahl der wichtigsten zulässigen Eingaben wird unter der NEXT-Zeile angezeigt.

+ , ++ , - , --	blättert im Menü (bzw. Fragebogen) vor und zurück. Dabei entsprechen „-“ der <b>[F7]</b> -Taste und „+“ der <b>[F8]</b> -Taste im Style-Guide-Modus.
*EXECUTE	führt die aktuelle Operation (Kommando bzw. Anweisung) aus. Entspricht der <b>[F3]</b> -Taste im Old-Modus bzw. der <b>[F11]</b> -Taste im Style-Guide-Modus.
*CONTINUE	gibt einen Unterfragebogen aus, falls der aktuelle Fragebogen mindestens einen struktureinleitenden Operandenwert enthält. Andernfalls wird die aktuelle Operation ausgeführt.
*TEST	prüft Eingaben auf Syntaxfehler. Entspricht der <b>[F2]</b> -Taste im Old-Modus.
*CANCEL	bricht das aktuell angezeigte Menü (bzw. den Operandenfragebogen) ab und wechselt zum übergeordneten Menü bzw. im Falle eines Unterfragebogens in den übergeordneten Operandenfragebogen. Die bisher im Unterfragebogen gemachten Angaben werden verworfen. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet *CANCEL nach Rückfrage das Programm. Entspricht der <b>[F12]</b> -Taste im Style-Guide-Modus.
*EXIT	bricht das aktuell angezeigte Menü (bzw. den Operandenfragebogen) ab und wechselt zum übergeordneten Menü. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet *EXIT nach Rückfrage das Programm. Entspricht der <b>[K1]</b> -Taste im Old-Modus bzw. der <b>[F3]</b> -Taste im Style-Guide-Modus.
*EXIT-ALL	bricht das aktuelle Menü (bzw. den Operandenfragebogen) ab und wechselt im geführten Dialog zur höchsten Menüstufe bzw. im temporär geführten Dialog zur Kommando- bzw. Anweisungeingabe des ungeführten Dialogs. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet

	*EXIT-ALL nach Rückfrage das Programm. Entspricht der <b>[F1]</b> -Taste im Old-Modus bzw. der <b>[F6]</b> -Taste im Style-Guide-Modus.
*REFRESH	wiederholt den zuletzt ausgegebenen Bildschirm mit den ursprünglichen Werten. Entspricht der <b>[K3]</b> -Taste im Old-Modus bzw. der <b>[F5]</b> -Taste im Style-Guide-Modus.
*DOM-MENU	wechselt in das Anwendungsbereichs-Menü. Wird *DOM-MENU in einem Operandenfragebogen angegeben, so wird vor dem Wechsel die aktuelle Operation ausgeführt.
<number>	Bei Eingabe im Anwendungsbereichs-Menü wird das Kommando-Menü des entsprechenden Anwendungsbereichs ausgegeben. Bei Eingabe im Kommando- bzw. Anweisungsmenü wird der Operandenfragebogen des entsprechenden Kommandos bzw. der entsprechenden Anweisung ausgegeben.
(<domain>)	gibt das Kommando-Menü des Anwendungsbereichs <domain> aus. Wird (<domain>) in einem Operandenfragebogen angegeben, so wird vor dem Wechsel die aktuelle Operation ausgeführt.
<command>?	führt die aktuelle Operation aus und gibt anschließend den Operandenfragebogen des Kommandos <command> aus. Bereits angegebene Operandenwerte werden in den Fragebogen übernommen.
<command>	führt die aktuelle Operation aus und anschließend auch das Kommando <command>. <command> schließt die Angabe der zugehörigen Operanden mit ein.
!<command>	führt die aktuelle Operation aus und definiert anschließend für das Kommando <command> task-spezifische Default-Werte. <command> schließt die Angabe der zugehörigen Operanden mit ein.
<statement>?	führt die aktuelle Operation aus und gibt anschließend den Operandenfragebogen der Anweisung <statement> aus. Bereits angegebene Operandenwerte werden in den Fragebogen übernommen.
<statement>	führt die aktuelle Operation aus und anschließend auch die Anweisung <statement>. <statement> schließt die Angabe der zugehörigen Operanden mit ein.
!<statement>	führt die aktuelle Operation aus und definiert anschließend für die Anweisung <statement> task-spezifische Default-Werte. <statement> schließt die Angabe der zugehörigen Operanden mit ein.



- ? schaltet für die aktuelle Eingabe in die nächsthöhere Führungsstufe. Entspricht der **[F1]**-Taste im Style-Guide-Modus.
- \*DOWN(<operand>) gibt den Unterfragebogen für den angegebenen Operandenwert <operand> aus, wenn dieser eine Struktur einleitet.
- \*UP wechselt vom Unterfragebogen zurück in den übergeordneten Operandenfragebogen. Die bisher im Unterfragebogen gemachten Angaben werden, im Gegensatz zu \*CANCEL, ausgewertet.

## 4.2.5 Funktionstasten

Bestimmte Funktionen kann der Benutzer durch Drücken einer Funktionstaste auslösen. Einige Eingaben in die NEXT-Zeile können dabei auch über die Funktionstasten erfolgen. Die Wirkung der Funktionstasten ist abhängig von der Einstellung der SDF-Option FUNCTION-KEYS; Einstellung mit dem Kommando MODIFY-SDF-OPTIONS. Es existieren zwei Belegungsmodi: der bisherige Old-Modus (\*OLD-MODE) und der Style-Guide-Modus (\*STYLE-GUIDE-MODE), der mehr Funktionalität bietet. Das Terminal oder die Terminalemulation müssen jedoch den Style-Guide-Modus unterstützen und entsprechend generiert sein. Die nachfolgende Aufstellung zeigt die Belegung der Funktionstasten:

<b>*OLD-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
<b>[K1]</b>	Exit-Funktion bricht das aktuell angezeigte Menü (bzw. den Operandenfragebogen) ab und wechselt zum übergeordneten Menü. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet K1 nach Rückfrage das Programm. Entspricht *EXIT in der NEXT-Zeile bzw. <b>[F3]</b> im Style-Guide-Modus.
<b>[K2]</b>	Unterbrechungs-Funktion Unterbricht ein aktuelles Programm oder eine Prozedur, oder bricht die Ausgabe eines Kommandos ab.
<b>[K3]</b>	Refresh-Funktion Wiederholt den zuletzt ausgegebenen Bildschirm mit den ursprünglichen Werten. Entspricht *REFRESH in der NEXT-Zeile bzw. <b>[F5]</b> im Style-Guide-Modus.
<b>[F1]</b>	Exit-all-Funktion Bricht das aktuelle Menü (bzw. den Operandenfragebogen) ab und wechselt im geführten Dialog zur höchsten Menüstufe bzw. im temporär geführten Dialog zur Kommando- bzw. Anweisungeingabe des ungeführten Dialogs. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet F1 nach Rückfrage das Programm. Entspricht *EXIT-ALL in der NEXT-Zeile bzw. <b>[F6]</b> im Style-Guide-Modus.

Tabelle 6: Funktionstastenbelegung (Old-Modus) im geführten Dialog (Teil 1 von 2)

<b>*OLD-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
[F2]	Test-Funktion Prüft Eingaben auf Syntaxfehler. Entspricht *TEST in der NEXT-Zeile.
[F3]	Execute-Funktion Führt die aktuelle Operation (Kommando bzw. Anweisung) aus. Entspricht *EXECUTE in der NEXT-Zeile bzw. [F11] im Style-Guide-Modus.

Tabelle 6: Funktionstastenbelegung (Old-Modus) im geführten Dialog (Teil 2 von 2)

<b>*STYLE-GUIDE-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
[K2]	Unterbrechungs-Funktion Unterbricht ein aktuelles Programm oder eine Prozedur, oder bricht die Ausgabe eines Kommandos ab.
[F1]	Hilfe-Funktion Wechselt in den temporär geführten Dialog. Entspricht der Eingabe von „?“ in der NEXT-Zeile.
[F3]	Exit-Funktion Bricht das aktuell angezeigte Menü (bzw. den Operandenfragebogen) ab und wechselt zum übergeordneten Menü. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet F3 nach Rückfrage das Programm. Entspricht *EXIT in der NEXT-Zeile bzw. [K1] im Old-Modus.
[F5]	Refresh-Funktion Wiederholt den zuletzt ausgegebenen Bildschirm mit den ursprünglichen Werten. Entspricht *REFRESH in der NEXT-Zeile bzw. [K3] im Old-Modus.
[F6]	Exit-all-Funktion Bricht das aktuelle Menü (bzw. den Operandenfragebogen) ab und wechselt im geführten Dialog zur höchsten Menüstufe bzw. im temporär geführten Dialog zur Kommando- bzw. Anweisungeingabe des ungeführten Dialogs. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet F6 nach Rückfrage das Programm. Entspricht *EXIT-ALL in der NEXT-Zeile bzw. [F1] im Old-Modus.
[F7]	Rückwärts Blättern Blättert in Menüs bzw. Fragebogen, die sich über mehrere Bildschirme erstrecken, zurück. Entspricht „-“ in der NEXT-Zeile.

Tabelle 7: Funktionstastenbelegung (Style-Guide-Modus) im geführten Dialog (Teil 1 von 2)

<b>*STYLE-GUIDE-MODE</b>	
<b>Funktionstaste</b>	<b>Wirkung</b>
<b>F8</b>	Vorwärts Blättern Blättert in Menüs bzw. Fragebogen, die sich über mehrere Bildschirme erstrecken, vorwärts. Entspricht „+“ in der NEXT-Zeile.
<b>F9</b>	Gibt das zuletzt eingegebene Kommando bzw. Anweisung erneut aus. Entspricht der Eingabe von „RESTORE-SDF-INPUT“ ohne Operanden (voreingestellt ist INPUT=*LAST-CMD bzw. *LAST-STMT).
<b>F11</b>	Execute-Funktion Führt die aktuelle Operation (Kommando bzw. Anweisung) aus. Entspricht *EXECUTE in der NEXT-Zeile bzw. <b>F3</b> im OLD-Modus.
<b>F12</b>	Cancel-Funktion Bricht das aktuell angezeigte Menü (bzw. den Operandenfragebogen) ab und wechselt zum übergeordneten Menü bzw. im Falle eines Unterfragebogens in den übergeordneten Operandenfragebogen. Die bisher im Unterfragebogen gemachten Angaben werden verworfen. Innerhalb eines Anweisungsmenüs im geführten Dialog beendet F12 nach Rückfrage das Programm. Entspricht *CANCEL in der NEXT-Zeile.

Tabelle 7: Funktionstastenbelegung (Style-Guide-Modus) im geführten Dialog (Teil 2 von 2)

### Hinweis

Im Style-Guide-Modus werden die Funktionstasten **K1**, **K3**, **F2**, **F4**, **F10**, **F11** und **F13** bis **F24** nicht unterstützt. Das Drücken einer nicht unterstützten Funktionstaste bewirkt die Ausgabe einer Fehlermeldung.

Situationsbedingt sind die Funktionstasten **F7**, **F8** und **F11** nicht immer verfügbar.

### 4.2.6 Wirkung der Rückzugsfunktionen

Die zwei nachfolgenden Bilder zeigen, wie die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL in verschiedenen Bildschirmmasken im Systemmodus und im Programmodus wirken. Die Rückzugsfunktionen können sowohl über die Eingabe der gleichnamigen Steueranweisung in die NEXT-Zeile, als auch durch Drücken der entsprechenden Funktionstaste ausgelöst werden.

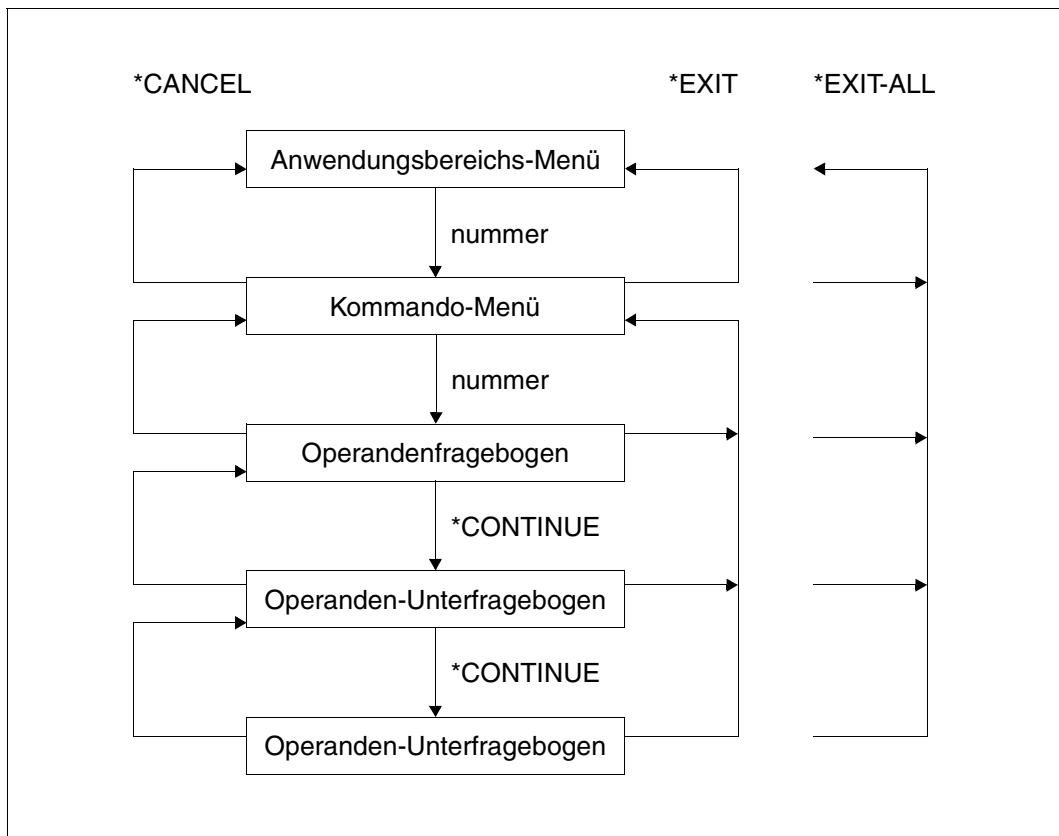


Bild 3: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Systemmodus (geführter Dialog)

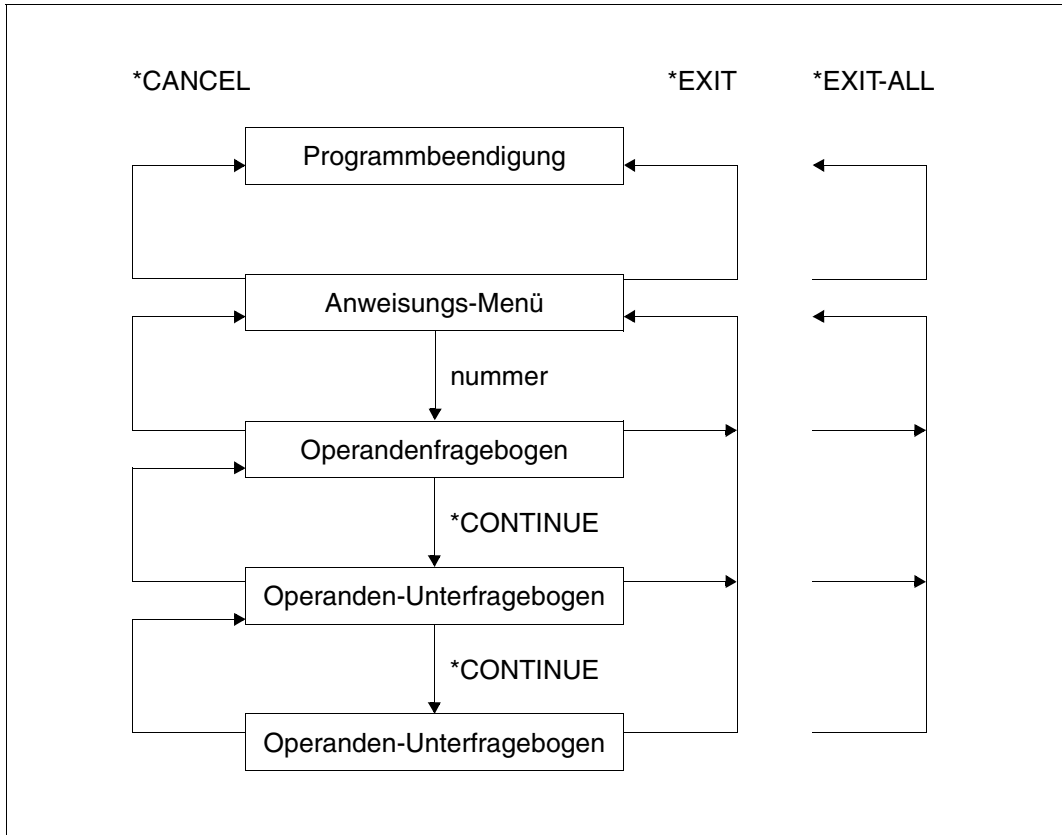


Bild 4: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Programmodus (geführter Dialog)

## 4.2.7 Anwendungsbereichs-Menü

Jedes BS2000-Kommando ist mindestens einem der folgenden Anwendungsbereiche zugeordnet:

ACCOUNTING	PREVENTIVE-DIAGNOSTIC-SUPPORT
ALL-COMMANDS <sup>1)</sup>	PROCEDURE
AUTOMATED OPERATING	PROGRAM
CONDITIONAL-JOB-CONTROL	PROGRAMMING-SUPPORT
CONSOLE-MANAGEMENT	RSO-SPOOL-ADMINISTRATION <sup>3)</sup>
DATA-BASE	SDF
DCAM	SECURITY-ADMINISTRATION
DEVICE	SPOOL <sup>4)</sup>
ERROR-LOGGING	SPOOL-PRINT-ADMINISTRATION
FILE	SPOOL-PRINT-SERVICES
FILE-GENERATION-GROUP	STORAGE-MANAGEMENT
FILE-TRANSFER	SUBSYSTEM-MANAGEMENT <sup>5)</sup>
JOB	SYSTEM-MANAGEMENT
JOB-VARIABLES	SYSTEM-TUNING
MESSAGE-PROCESSING	USER-ADMINISTRATION
MULTI-CATALOG-AND-PUBSET-MGMT <sup>2)</sup>	UTILITIES
NETWORK-MANAGEMENT	VM2000-VIRTUAL-MACHINE

Die hier aufgelisteten Anwendungsbereiche entsprechen dem Stand von BS2000/OSD-BC V5.0A. Anwendungsbereiche ohne Inhalt (Kommandos gesperrt oder nicht implementiert) werden nicht angezeigt.

*Anmerkungen:*

- 1) Der Anwendungsbereich ALL-COMMANDS enthält alle Kommandos, deren Eingabe im Dialog erlaubt ist.
- 2) Der Anwendungsbereich MULTI-CATALOG-AND-PUBSET-MGMT ersetzt langfristig den Anwendungsbereich MULTI-CATALOG.
- 3) Der Anwendungsbereich SPOOL-PRINT-ADMINISTRATION ersetzt langfristig den Anwendungsbereich RSO-SPOOL-ADMINISTRATION.
- 4) Der Anwendungsbereich SPOOL-PRINT-SERVICES ersetzt langfristig den Anwendungsbereich SPOOL.
- 5) Der Anwendungsbereich SYSTEM-MANAGEMENT ersetzt langfristig den Anwendungsbereich SUBSYSTEM-MANAGEMENT

SDF bietet dem Benutzer ein Menü an, in dem diese Anwendungsbereiche aufgelistet sind. Der Benutzer wählt den für seine Aufgabe relevanten Arbeitsbereich aus. SDF bietet ihm dann anschließend in einem Kommando-Menü die Kommandos dieses Anwendungsbereichs zur Auswahl an.

Das Anwendungsbereichs-Menü gibt SDF nur im Kommandomodus aus. Der Benutzer erhält es in folgenden Fällen:

- nach Einschalten des geführten Dialogs, sofern noch kein Anwendungsbereich eingestellt ist
- nach Eingabe eines Fragezeichens im ungeführten Dialog
- nach Angabe von \*DOM-MENU in der NEXT-Zeile eines Menüs oder Fragebogens
- nach Angabe von \*CANCEL in der NEXT-Zeile eines Kommando-Menüs oder nach Betätigen der **[F12]**-Taste (nur im Style-Guide-Modus)
- nach Angabe von \*EXIT in der NEXT-Zeile eines Kommando-Menüs oder nach Betätigen der **[K1]**- bzw. der **[F3]**-Taste (abhängig von der Einstellung FUNCTION-KEYS)
- nach Angabe von \*EXIT-ALL in der NEXT-Zeile bzw. nach Betätigen der **[F1]**-Taste bzw. der **[F6]**-Taste (abhängig von der Einstellung FUNCTION-KEYS) im geführten Dialog. Ein temporär geführter Dialog wird mit dieser Angabe bzw. Funktionstaste beendet.

## 4.2.8 Kommando- und Anweisungs-Menü

Für jeden Anwendungsbereich gibt es ein Menü, in dem die ihm zugeordneten Kommandos aufgelistet sind. Für jedes Programm mit SDF-Oberfläche gibt es ein Menü mit den zulässigen Anweisungen.

SDF bietet ein solches Menü dem Benutzer an. Der Benutzer wählt das für seine Aufgabe relevante Kommando (bzw. die Anweisung) aus. SDF gibt dann anschließend einen Operandenfragebogen für das ausgewählte Kommando (bzw. die Anweisung) aus. Besitzt das ausgewählte Kommando keine Operanden, so ist es im Kommando-Menü mit „(!)“ oder „(SOFORTIGE AUSFUEHRUNG!)“ gekennzeichnet und wird unmittelbar ohne vorherige Ausgabe eines Fragebogens ausgeführt.

Ein Kommando- bzw. Anweisungs-Menü erhält der Benutzer in folgenden Fällen:

- nach Auswahl eines Anwendungsbereichs im Anwendungsbereichs-Menü
- nach Start eines Programms, sofern direkt nach Programmstart mehrere Anweisungen zugelassen werden
- nach Angabe von (<domain>) in der NEXT-Zeile eines Menüs oder Fragebogens (Kommando-Menü für den Anwendungsbereich <domain>)

- nach Angabe von \*CANCEL in der NEXT-Zeile eines Operandenfragebogens oder nach Betätigen der **[F12]**-Taste im Style-Guide-Modus)
- nach Angabe von \*EXIT in der NEXT-Zeile eines Operandenfragebogens oder nach Betätigen der **[K1]**-Taste bzw. der **[F3]**-Taste (abhängig von der Einstellung FUNCTION-KEYS)
- nach Ausführung eines Kommandos bzw. einer Anweisung im geführten Dialog (Kommandomodus: Menü für den Anwendungsbereich, der zuletzt eingestellt war; Programmmodus: Voraussetzung ist, dass für den folgenden Bearbeitungsschritt mehrere Anweisungen zugelassen werden.)
- im Kommandomodus nach Einschalten des geführten Dialogs, wenn bereits ein Anwendungsbereich eingestellt ist
- im Programmmodus nach Einschalten des geführten Dialogs während des Programmablaufs
- im Programmmodus nach Eingabe eines Fragezeichens im ungeführten Dialog, sofern für den folgenden Bearbeitungsschritt mehrere Anweisungen zugelassen werden

## 4.2.9 Operandenfragebogen

Für jedes Kommando und jede Anweisung mit Operanden gibt es einen Operandenfragebogen. In ihm sind die zugehörigen Operanden aufgelistet. Je nach Definition in der Syntaxdatei können bei minimaler und mittlerer Führung die Operanden einer Struktur (siehe [Seite 24](#)) in den Fragebogen integriert oder in einem eigenem Unterfragebogen aufgelistet sein. Wenn SDF im Fragebogen einen Operandenwert angibt, für dessen Struktur es einen eigenen Unterfragebogen gibt, so weist ein hinter dem Wert stehendes leeres Klammerpaar auf den Unterfragebogen hin. Auch wenn ein Unterfragebogen existiert, kann der Benutzer die Operanden der Struktur im übergeordneten Fragebogen angeben. Sie sind in runde Klammern einzuschließen und unmittelbar an den struktureinleitenden Operandenwert anzuhängen.

SDF bietet dem Benutzer einen Fragebogen an, in dem dieser die Werte für die Operanden eines Kommandos bzw. einer Anweisung angibt. Der Fragebogen ist mit den Default-Werten der wahlweisen Operanden vorbesetzt. Nach dem Absenden eines Fragebogens gibt SDF entweder einen weiteren Fragebogen aus (Unterfragebogen für eine Struktur) oder SDF veranlasst die Ausführung des Kommandos bzw. der Anweisung. Einen Operandenfragebogen erhält der Benutzer in folgenden Fällen:

- nach Auswahl des Kommandos bzw. der Anweisung im Menü
- nach Eingabe von <command>? bzw. <statement>? im ungeführten Dialog oder in der NEXT-Zeile eines Fragebogens oder Menüs



- nach Eingabe von !<command>? bzw. !<statement>? im ungeführten Dialog oder in der NEXT-Zeile eines Fragebogens oder Menüs
- wenn im geführten Dialog ein Programm eine bestimmte Anweisung erwartet (Operandenfragebogen für eine Anweisung)
- im Programmmodus nach Eingabe eines Fragezeichens im ungeführten Dialog, wenn das Programm für den folgenden Bearbeitungsschritt eine bestimmte Anweisung erwartet
- wenn im ungeführten Dialog (NO-Form) die Aufforderung zur Korrektur einer fehlerhaften Eingabe mit einem Fragezeichen beantwortet wird
- wenn im geführten Dialog eine fehlerhafte Eingabe gemacht wurde

Nach Absenden eines Operandenfragebogens, in dem statt eines Operandenwerts ein Fragezeichen steht, bietet SDF den gleichen Fragebogen mit zusätzlichen Informationen an.

Der Benutzer wechselt in folgenden Fällen in einen untergeordneten Operandenfragebogen:

- nach Angabe mindestens eines struktureinleitenden Operandenwerts im Fragebogen und Angabe von \*CONTINUE in der NEXT-Zeile (Ausgabe eines Unterfragebogens)
- nach Angabe eines struktureinleitenden Operandenwerts im Fragebogen und Angabe von \*DOWN(<operand>) in der NEXT-Zeile eines Fragebogens (Wechsel in den Unterfragebogen für eine einzelne Struktur)

In einen übergeordneten Operandenfragebogen wechselt der Benutzer in folgenden Fällen:

- nach Angabe von \*UP in der NEXT-Zeile eines Unterfragebogens
- nach Angabe von „-“ in der NEXT-Zeile der Anfangsseite des ersten einer Folge von Unterfragebögen (bzw. Betätigen der **F7**-Taste im Style-Guide-Modus)
- nach Angabe von \*CANCEL in der NEXT-Zeile eines Unterfragebogens (bzw. Betätigen der **F12**-Taste im Style-Guide-Modus)

Die bisher im Unterfragebogen gemachten Angaben bleiben beim Wechsel mit \*UP oder „-“ erhalten, beim Wechsel mit \*CANCEL werden sie jedoch verworfen.








## 4.2.10 Eingaben in den Operandenfragebogen

In die Eingabezeilen des Operandenfragebogens können entsprechende Operandenwerte eingetragen bzw. die vorbesetzten Default-Werte übernommen werden. Beim Eintragen eines neuen Operandenwertes muss ein bereits vorhandener Wert restlos überschrieben werden. Verbleibende Zeichen können mit der **[LZF]**-Taste gelöscht oder mit Leerzeichen überschrieben werden. Bei der Eingabe einer Liste von Operandenwerten (list-possible) können die Klammern entfallen. Weitere Angaben sind möglich:

- |                     |   |
|---------------------|---|
| ?                   | als Operandenwert liefert Hilfetext und Angabe des Wertebereichs für diesen Operanden. Hat SDF nach vorheriger fehlerhafter Eingabe die Meldung „BITTE KORRIGIEREN SIE DEN ODER DIE FEHLERHAFTEN OPERANDEN“ gebracht, liefert das Fragezeichen zusätzlich detaillierte Fehlermeldungen.<br>Der Zeilenrest muss nicht gelöscht werden. |
| ??                  | als Operandenwert liefert Informationen zu Datentypen, die als Operandenwert zugelassen sind.<br>Der Zeilenrest muss nicht gelöscht werden.   |
| !                   | als Operandenwert setzt für diesen Operanden den Default-Wert ein (von Bedeutung, wenn der als Vorbesetzung im Fragebogen stehende Default-Wert vorher überschrieben wurde).<br>Der Zeilenrest muss nicht gelöscht werden.  |
| ^                   | als Operandenwert eines „geheimen“ Operanden fordert ein dunkelgesteuertes Eingabefeld für die verdeckte Eingabe des Operandenwertes an.  |
| ([<operand,...]     | hinter einem struktureinleitenden Operandenwert gibt den Unterfragebogen für die zugehörige Struktur aus. Dieser ist mit den nach der geöffneten Klammer eventuell angegebenen Operanden vorbesetzt.  |
| ([<operand>,...])   | hinter einem struktureinleitenden Operandenwert unterdrückt den Unterfragebogen und setzt für die nicht angegebenen Operanden der Struktur Default-Werte ein.   |
| -                   | als letztes Zeichen in einer Eingabezeile bewirkt die Ausgabe einer Fortsetzungszeile.  |
| <b>[LZF]</b> -Taste | löscht ab Schreibmarke alle Zeichen der Eingabezeile.   |

## 4.2.11 Positionieren im Operandenfragebogen

Zu Beginn eines Operandenfragebogens befindet sich die Schreibmarke am Anfang der ersten Eingabezeile. Einen Wechsel der Eingabezeile nimmt der Benutzer durch Bewegen der Schreibmarke vor:

-  Die Schreibmarke wechselt in dieselbe Position der darunter liegenden Eingabezeile. Aus der NEXT-Zeile wird wieder in die erste Eingabezeile gewechselt.
-  Die Schreibmarke wechselt in dieselbe Position der darüberliegenden Eingabezeile. Befindet sich die Schreibmarke in der ersten Eingabezeile, so wechselt sie an den Zeilenbeginn. Befindet sich die Schreibmarke in der NEXT-Zeile wechselt sie an deren Beginn.
-  Die Schreibmarke wechselt an den Beginn der darunter liegenden Eingabezeile. Folgezeilen werden dabei übersprungen. Aus der NEXT-Zeile wird wieder in die erste Eingabezeile gewechselt.
-  Die Schreibmarke wechselt an den Beginn der darüberliegenden Eingabezeile, wenn sich die Schreibmarke am Beginn der aktuellen Eingabezeile befindet. Andernfalls wechselt die Schreibmarke an den Beginn der aktuellen Eingabezeile.
-  Die Schreibmarke wechselt an den Beginn der ersten Eingabezeile.
-  Die Schreibmarke wechselt an den Beginn einer Eingabezeile bzw. Folgezeile. Befindet sich die Schreibmarke in der ersten Eingabezeile, so wechselt sie an den Zeilenbeginn. Befindet sich die Schreibmarke in der NEXT-Zeile wechselt sie an deren Beginn.
-  Die Schreibmarke wechselt an den Beginn der darunter liegenden Eingabezeile bzw. Folgezeile. Aus der NEXT-Zeile wird wieder in die erste Eingabezeile gewechselt.

## 4.3 Eingabe im temporär geführten Dialog

Sowohl aus der EXPERT- als auch NO-Form des ungeführten Dialogs kann der Benutzer für die Eingabe eines Kommandos oder einer Anweisung in den temporär geführten Dialog wechseln. Der Umfang der Benutzerführung des temporär geführten Dialogs entspricht dem Umfang bei minimaler Benutzerführung im geführten Dialog (siehe [Seite 58](#)). Nach Beendigung der Eingabe arbeitet der Benutzer wieder in der ursprünglich eingestellten Form des ungeführten Dialogs.

Im Kommandomodus des ungeführten Dialogs kann der Benutzer mit der Eingabe „?“ ein Anwendungsbereichs-Menü anfordern. Aus diesem wählt er den Anwendungsbereich aus, in dem er arbeiten will. SDF gibt dann das Kommando-Menü dieses Anwendungsbereichs aus.

Im Programmodus des ungeführten Dialogs erhält der Benutzer nach Eingabe von „?“ das Anweisungs-Menü. Ist zum Zeitpunkt der Eingabe nur eine einzige Anweisung sinnvoll, so gibt SDF statt des Anweisungs-Menüs den Operandenfragebogen dieser Anweisung aus.

Den Operandenfragebogen für das Kommando `<command>` bzw. die Anweisung `<statement>` fordert der Benutzer mit „`<command>? [<operand>,...]`“ bzw. mit „`<statement>? [<operand>,...]`“ an.

Der Fragebogen ist mit den ggf. angegebenen Operandenwerten sowie mit den Default-Werten der wahlweisen Operanden vorbelegt. Durch Drücken der **[K1]**-Taste im Old-Modus bzw. **[F3]**-Taste im Style-Guide-Modus oder durch Angabe von \*EXIT in der NEXT-Zeile des Fragebogens erhält der Benutzer das Anwendungsbereichs-Menü (bzw. im Programmodus das Anweisungs-Menü).

Ist `<command>` ein Kommando im ISP-Format, so sind drei Fälle zu unterscheiden:

- Das Kommando wird in der Form `<command>?` eingegeben und lässt sich genau einem Kommando im SDF-Format zuordnen. SDF gibt den Operandenfragebogen des entsprechenden Kommandos im SDF-Format aus.
- Das Kommando wird in der Form `<command>?` eingegeben und lässt sich mehreren Kommandos im SDF-Format zuordnen. SDF gibt ein Menü mit den entsprechenden Kommandos aus.
- Das Kommando wird in der Form `<command>? <operand>,...` eingegeben. SDF gibt den Fragebogen für das eingegebene ISP-Kommando aus. Dieser besitzt nur den Operanden OPERANDS, der mit der bereits eingegebenen Operandenzeichenfolge vorbelegt ist. In das Eingabefeld können weitere Operanden als Schlüsselwort- oder auch als Stellungsoperand eingetragen werden. Für ISP-Kommandos ist jeweils nur ein Operand definiert, der die Aufzählung aller ISP-Operanden zulässt (definiert als `<command-rest>`). Die Eingabe wird auf Operandenebene nicht von SDF analysiert (d.h. Syntaxfehler werden erst vom ausführenden Modul abgewiesen).

In der NO-Form des ungeführten Dialogs erhält der Benutzer nach einer fehlerhaften Eingabe die Aufforderung zur Korrektur der angezeigten fehlerhaften Operanden. Bei Eingabe von „?“ gibt SDF den Operandenfragebogen für das zu korrigierende Kommando (bzw. Anweisung) aus.

Will der Benutzer den temporär geführten Dialog abbrechen und in den ungeführten Dialog zurückkehren, so muss er entweder die **[F1]**-Taste (Old-Modus) bzw. die **[F6]**-Taste (Style-Guide-Modus) drücken oder in der NEXT-Zeile des Menüs bzw. Fragebogens \*EXIT-ALL eingeben.

Die letzte Eingabe kann mit dem Kommando bzw. der Anweisung RESTORE-SDF-INPUT oder durch Betätigen der **[F9]**-Taste (Style-Guide-Modus) wiederhergestellt werden (siehe [Seite 199](#)).

### 4.3.1 Wirkung der Rückzugsfunktionen

Die zwei nachfolgenden Bilder zeigen, wie die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL in verschiedenen Bildschirmmasken im Systemmodus und im Programmodus wirken. Die Rückzugsfunktionen können sowohl über die Eingabe der gleichnamigen Steueranweisung in die NEXT-Zeile, als auch durch Drücken der entsprechenden Funktionstaste ausgelöst werden.

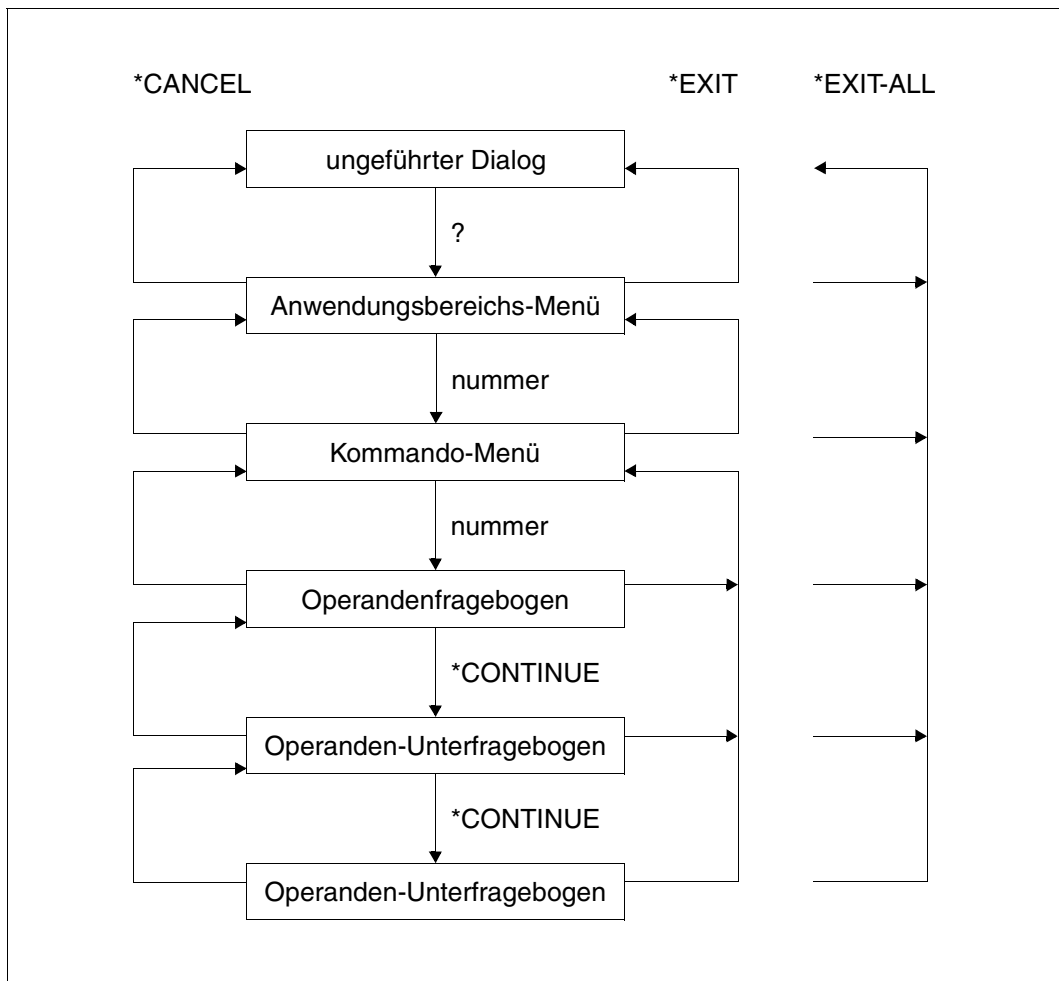


Bild 5: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Systemmodus (temporär geführter Dialog)

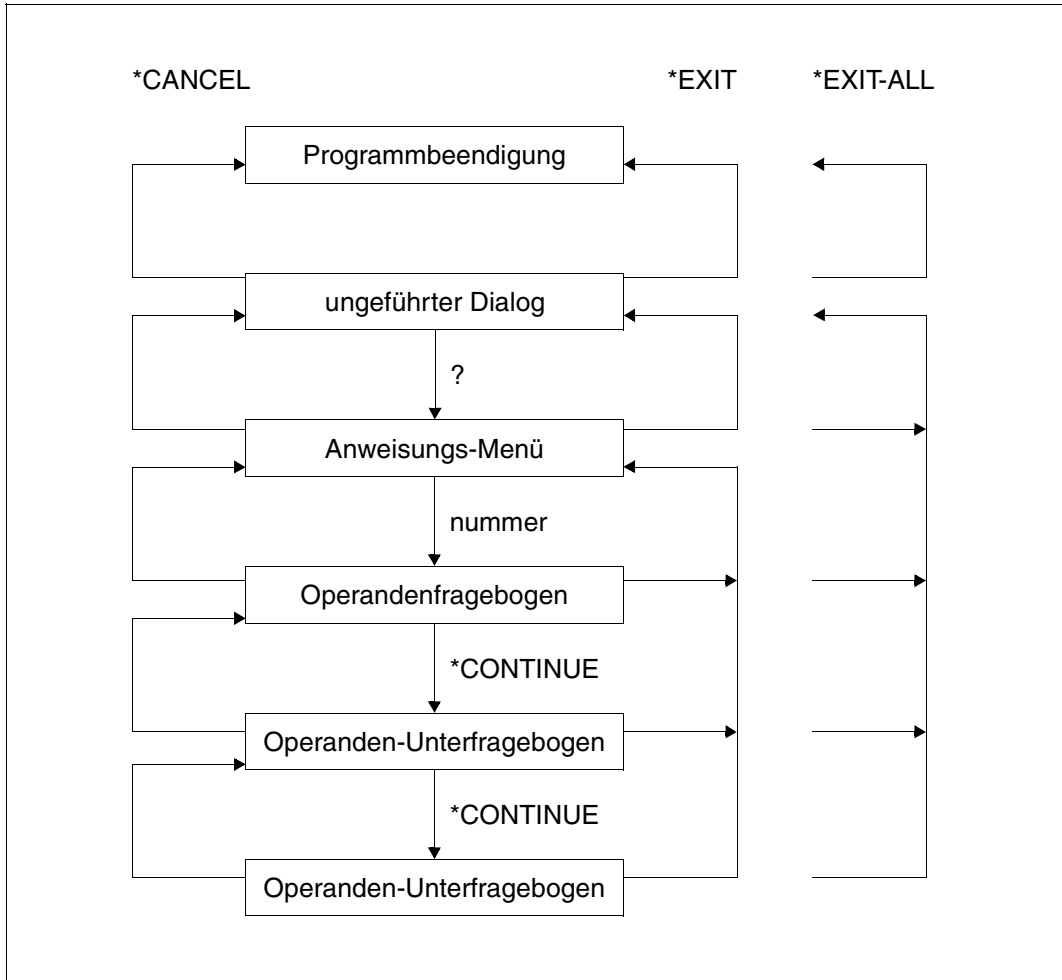


Bild 6: Wirkung der Funktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Programmodus (temporär geführter Dialog)

## 4.4 Weitere Eingabemöglichkeiten

Neben der Eingabe von der Datensichtstation ist auch die Eingabe im Prozedur- und im Stapelbetrieb möglich. Die Kommandos und Anweisungen speichert der Benutzer in einer Datei oder zusammengesetzten S-Variablen ab, wobei die formalen Regeln für das Erstellen von Prozeduren bzw. ENTER-Dateien zu beachten sind.

### 4.4.1 Eingabe aus Prozeduren

Die Eingabe aus Prozeduren entspricht der im ungeführten Dialog. Buchstaben sind als Großbuchstaben einzugeben. Kommandos ist ein Schrägstrich voranzustellen; Anweisungen werden zwei Schrägstriche vorangestellt. Ein Kommando bzw. eine Anweisung kann aus einer oder mehreren Zeilen bestehen. Als Fortsetzungszeichen gilt der Bindestrich. Er zeigt an, dass eine Folgezeile vorhanden ist. Jede Folgezeile beginnt mit einem Schrägstrich (bzw. zwei Schrägstrichen bei Anweisungen).

Je nach Einstellung muss das Fortsetzungszeichen bei Kommandos genau in Spalte 72 (Old-Modus) oder in einer der Spalten 2 bis 72 (New-Modus) stehen. Diese Einstellung lässt sich mit dem Kommando `SHOW-SDF-OPTIONS` abfragen. Die Einstellung wird durch die entsprechende Angabe (`*OLD-MODE` oder `*NEW-MODE`) im Operanden `CONTINUATION` des Kommandos `MODIFY-SDF-OPTIONS` vorgenommen. In S-Prozeduren kann die Zeilenlänge mit dem Kommando `SET-PROCEDURE-OPTIONS INPUT-FORMAT=*FREE-RECORD-LENGTH` vereinbart werden, d.h., das Fortsetzungszeichen kann an beliebiger Stelle ab Spalte 2 stehen und die Zeilenlänge kann somit 72 Zeichen überschreiten.

Bei Anweisungen kann das Fortsetzungszeichen in jedem Fall an beliebiger Stelle ab Spalte 2 stehen, und die Zeilenlänge kann mehr als 72 Zeichen betragen.

Eingabesätze, die im Widerspruch zur SDF-Syntaxbeschreibung stehen, lösen in Nicht-S-Prozeduren Spin-Off, in S-Prozeduren die SDF-P-Fehlerbehandlung aus. Es besteht jedoch die Möglichkeit, während des Prozedurablaufs im Dialog, einen Korrekturdialog für fehlerhafte Prozedurkommandos bzw. -anweisungen zu erlauben. Dies kann durch die entsprechende Angabe im Operanden `PROCEDURE-DIALOG` des Kommandos `MODIFY-SDF-OPTIONS` eingestellt werden. Die aktuelle Einstellung kann mit dem Kommando `SHOW-SDF-OPTIONS` erfragt werden.

Die Definition von „Task-spezifischen Default-Werten“ ist ab [Seite 110](#) beschrieben.

### Implementierte Prozeduren

Der Benutzer kann eigene Kommandos in einer Benutzersyntaxdatei definieren (siehe [Seite 157](#) und [Seite 163](#)). Er erstellt dabei für das Kommando eine Syntaxbeschreibung, die den Kommandonamen, die Operandennamen und die jeweils möglichen Operanden-



werte enthält. Der Kommandoname wird bei der Definition mit dem Namen einer Prozedur verknüpft, die bei Eingabe des „neuen“ Kommandos aufgerufen wird. In der Syntaxbeschreibung ist festgelegt, wie die Prozedurparameter bei Aufruf durch SDF zu versorgen sind. Parameterwerte, die der Kommandoaufrufer bestimmen soll, werden über die angegebenen Operandenwerte bestimmt.

Der Benutzer kann somit einer Prozedur eine SDF-Oberfläche geben und damit alle Funktionen der SDF-Schnittstelle nutzen. Insbesondere sind alle Arten der Dialogführung, die Prüfung und Korrektur der angegebenen Operandenwerte, die Information über zulässige Werte mit Erläuterungen und die verdeckte Eingabe von sicherheitsrelevanten Werten möglich.

Die Definition eigener Kommandos ist mit dem Softwareprodukt SDF-A vorzunehmen (Beispiel siehe Handbuch „SDF-A“ [4]).

### LOGON-Prozeduren

Bei der LOGON-Verarbeitung besteht die Möglichkeit, LOGON-Prozeduren automatisch ablaufen zu lassen. LOGON-Prozeduren können sowohl systemweit als auch benutzerspezifisch eingesetzt werden. Der Aufruf kann in beiden Fällen als Call- oder Include-Prozedur erfolgen (entspricht dem Aufruf mit dem Kommando CALL- bzw. INCLUDE-PROCEDURE; siehe auch Handbuch „SDF-P“ [6]).

LOGON-Prozeduren dürfen kein DO-Kommando enthalten und dürfen nicht mit dem Kommando ENDP-RESUME beendet werden. Außerdem sollten sie nicht unterbrechbar sein.

Der Benutzer kann für alle Aufträge, die unter seiner Benutzerkennung laufen, sowohl eine Call- als auch eine Include-Prozedur als Benutzer-LOGON-Prozeduren anlegen, wobei eine Include-Prozedur vor der entsprechenden Call-Prozedur gestartet wird.

Für den automatischen Aufruf muss die Call-Prozedur unter dem Standardnamen \$userid.SYS.SDF.LOGON.USERPROC, die Include-Prozedur unter dem Standardnamen \$userid.SYS.SDF.LOGON.USERINCL katalogisiert werden.

Systemweit kann die Systembetreuung durch einen entsprechenden Eintrag in der SDF-Parameterdatei System-LOGON-Prozeduren (Call- und Include-Prozedur) für alle Benutzer bereitstellen (siehe Handbuch „SDF Verwaltung“ [5]).

Für Task, bei denen eine Gruppensyntaxdatei ohne Auswertung der Systemsyntaxdatei zugewiesen ist, werden keine System-LOGON-Prozeduren gestartet.

Benutzer-LOGON-Prozeduren werden erst nach Ablauf der vereinbarten System-LOGON-Prozeduren gestartet. Erst nach Ablauf der LOGON-Prozeduren können Kommandos eingegeben werden.

LOGON-Prozeduren werden in folgenden Fällen ohne Warnung ignoriert:

- Die Prozedurdatei ist nur katalogisiert, belegt aber keinen Speicherplatz.
- Die Task ist eine RFA-Task.
- Die Task besitzt außer HARDWARE-MAINTENANCE, SECURITY-ADMINISTRATION, SAT-FILE-MANAGEMENT und SAT-FILE-EVALUATION kein anderes Privileg.

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

### LOGOFF-Prozeduren

Bei der LOGOFF-Verarbeitung besteht die Möglichkeit, LOGOFF-Prozeduren automatisch ablaufen zu lassen. LOGOFF-Prozeduren können sowohl systemweit als auch benutzer-spezifisch eingesetzt werden. Der Aufruf kann in beiden Fällen als Call- oder Include-Prozedur erfolgen (entspricht dem Aufruf mit dem Kommando CALL- bzw. INCLUDE-PROCEDURE; siehe auch Handbuch „SDF-P“ [6]).

LOGOFF-Prozeduren dürfen kein DO-Kommando enthalten und dürfen nicht mit dem Kommando ENDP-RESUME beendet werden. Außerdem sollten sie nicht unterbrechbar sein.

Der Benutzer kann für alle Aufträge, die unter seiner Benutzerkennung laufen, sowohl eine Call- als auch eine Include-Prozedur als Benutzer-LOGOFF-Prozeduren anlegen, wobei eine Include-Prozedur vor der entsprechenden Call-Prozedur gestartet wird.

Für den automatischen Aufruf muss die Call-Prozedur unter dem Standardnamen \$userid.SYS.SDF.LOGOFF.USERPROC, die Include-Prozedur unter dem Standardnamen \$userid.SYS.SDF.LOGOFF.USERINCL katalogisiert werden.

Wird in der Prozedur ein EXIT-JOB- bzw. LOGOFF-Kommando aufgerufen, endet die Bearbeitung der Benutzer-LOGOFF-Prozeduren und es werden zunächst ggf. vereinbarte System-LOGOFF-Prozeduren gestartet.

Systemweit kann die Systembetreuung System-LOGOFF-Prozeduren (Call- und Include-Prozedur) für alle Benutzer durch einen entsprechenden Eintrag in der SDF-Parameterdatei bereitstellen (siehe Handbuch „SDF Verwaltung“ [5]). System-LOGOFF-Prozeduren werden erst nach Ablauf der vereinbarten Benutzer-LOGON-Prozeduren gestartet. Benutzer- und Gruppensyntaxdateien werden vor ihrem Start deaktiviert.

Für Task, bei denen eine Gruppensyntaxdatei ohne Auswertung der Systemsyntaxdatei zugewiesen ist, werden keine System-LOGOFF-Prozeduren gestartet.

LOGOFF-Prozeduren werden in folgenden Fällen ohne Warnung ignoriert:

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

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

### Prozedurdialog

In Prozeduren kann die SDF-Dialogführung zugelassen werden, d.h. Prozedureingaben werden von SDF wie Dialogeingaben behandelt. So kann z.B. der Korrekturdialog bei syntaktisch falschen Eingaben genutzt werden. Außerdem kann in der Prozedur erzwungen werden, dass Kennwörter verdeckt eingegeben werden (Operandenwert \*SECRET in der Prozedur).

Der Prozedurdialog kann genutzt werden, wenn vor Prozeduraufruf bzw. in der Prozedur die SDF-Option PROCEDURE-DIALOG auf \*YES gesetzt wird.

Zur Nutzung des Korrekturdialogs ist außerdem eine entsprechende Führungsstufe einzustellen (z.B. GUIDANCE=\*NO).

### Unterbrechungsschutz von Prozeduren

Die Möglichkeit, Prozeduren mit der **[K2]**-Funktionstaste zu unterbrechen, kann durch entsprechende Angabe im Operanden INTERRUPT-ALLOWED im Kommando BEGIN-PROCEDURE einer Nicht-S-Prozedur bzw. SET-PROCEDURE-OPTIONS einer S-Prozedur untersagt werden. Damit können Prozeduren gegen eine unerwünschte Unterbrechung geschützt werden (z.B. innerhalb der Prozedur werden Zugriffe auf geschützte Dateien erlaubt). Die Eigenschaft „nicht unterbrechbar“ bleibt bei Prozedurschachtelung erhalten.

### Unterbrechungsschutz von Programmen in Prozeduren

Ein Programm, das innerhalb einer nicht unterbrechbaren Prozedur geladen wird, ist nur in bestimmten Fällen gegen Unterbrechung (implizit) geschützt. Dabei kann eine Programmunterbrechung durch die Prozedur selbst als beabsichtigt angesehen werden. Eine Unterbrechung durch den Prozeduraufrufer ist aber nicht erwünscht.

Andererseits kann es während des Programmlaufs Verarbeitungszustände geben, in denen keinerlei Unterbrechung (auch nicht aus einer Prozedur) erwünscht ist (z.B. Verarbeitung sicherheitsrelevanter Daten). In diesem Fall muss sich das Programm explizit gegen eine Unterbrechung schützen.

Innerhalb einer Prozedur oder eines Programmes können folgende Ereignisse (Events) eine Unterbrechung anfordern:

Kontrollinstanz	Ereignis	vor Ereignis		nach Ereignis	
		Modus	Eingabestufe	Modus	Eingabestufe
Prozedur	<b>[K2]</b> -Taste	proc	cmd	dia	cmd
Programm	<b>[K2]</b> -Taste	proc	prog	dia	cmd
		dia	prog	dia	cmd
	BKPT-Makro	any	prog	same	cmd

Fortsetzung ➡

Kontroll- instanz	Ereignis	vor Ereignis		nach Ereignis	
		Modus	Eingabe- stufe	Modus	Eingabe- stufe
Programm (Forts.)	CMD-Makro	any	prog	mclp	cmd
	//HOLD-PROGRAM	any	prog	same	cmd
	//EXECUTE-SYSTEM-CMD	any	prog	mclp	cmd
	/HOLD-PROGRAM	proc	prog	proc	cmd

Bedeutung: proc Prozedur  
 dia Dialog  
 cmd Kommandoeingabe  
 prog Programmeingabe  
 mclp Kommandoeingabe über CMD-Makro  
 any Kommando- oder Programmeingabe  
 same derselbe Modus wie vor Eintritt des Ereignisses

Die Unterbrechungsanforderung für ein in einer Prozedur laufendes Programm wird dabei wie folgt behandelt:

Unterbrechung durch	Unterbrechung erlaubt			
	Programm	nein	ja	
	Prozedur	ja / nein	nein	ja
[K2]-Taste		R	R	A
[K2]-STXIT-Routine		A,RE	A,RI	A
BKPT-Makro		A,RE	A,RI	A
CMD-Makro		A,RE	A,RI	A
andere Makroaufrufe		A,RE	A,RI	A
//HOLD-PROGRAM		R	C	C
//EXECUTE-SYSTEM-CMD		R	C	A
/HOLD-PROGRAM		R	C	C

Bedeutung: A Die Unterbrechung wird vom System akzeptiert.  
 RE Die Unterbrechung sollte von einem Programm, das explizit nicht unterbrechbar ist (CLISSET), zurückgewiesen werden.  
 RI Die Unterbrechung sollte von einem Programm, das implizit nicht unterbrechbar ist, zurückgewiesen werden.  
 R Die Unterbrechung wird vom System zurückgewiesen.  
 C Die Unterbrechung wird vom System zurückgewiesen, wenn die Anweisungen nicht aus der Prozedurdatei gelesen werden (SYSSTMT ≠ SYSCMD).

Zurückweisen bedeutet:

- `[K2]` wird ignoriert und die Verarbeitung wird fortgesetzt.
- Das Kommando bzw. die Standardanweisung `HOLD-PROGRAM` oder das Kommando `BEGIN-BLOCK PROGRAM-INPUT=*MIXED-WITH-CMD` erzeugen im Programm die EOF-Bedingung.
- Die Standardanweisung `EXECUTE-SYSTEM-CMD` wird zurückgewiesen und aktiviert Spin-Off auf Anweisungsebene.
- Die Standardanweisung `HOLD-PROGRAM` wird in jedem Fall zurückgewiesen, wenn sie nicht aus der Prozedurdatei gelesen wird. Anderenfalls könnte eine unterschiedliche Eingabequelle von Anweisungen und Kommandos zu Inkonsistenzen im Prozedurablauf führen.

Folgende Funktionalität wird mit `BS2000/OSD-BC ≥ V2.0` unterstützt:

- die Standardanweisungen `EXECUTE-SYSTEM-CMD` und `HOLD-PROGRAM`
- der Makroaufruf `CLISSET`, der ein Programm explizit für nicht unterbrechbar erklärt
- der Makroaufruf `CLIGET`, mit dem abgefragt wird, ob die programmaufrufende Prozedur nichtunterbrechbar ist

## 4.4.2 Eingabe im Stapelbetrieb

Die Eingabe im Stapelbetrieb entspricht der im ungeführten Dialog. Buchstaben sind als Großbuchstaben einzugeben. Kommandos ist ein Schrägstrich voranzustellen, Anweisungen zwei Schrägstriche. Ein Kommando bzw. eine Anweisung kann aus einer oder mehreren Zeilen bestehen. Als Fortsetzungszeichen gilt der Bindestrich. Er zeigt an, dass eine Folgezeile vorhanden ist. Jede Folgezeile beginnt mit einem Schrägstrich (bzw. zwei Schrägstrichen bei Anweisungen).

Je nach Einstellung der SDF-Option CONTINUATION muss das Fortsetzungszeichen bei Kommandos genau in Spalte 72 (Old-Modus) oder in einer der Spalten 2 bis 72 (New-Modus) stehen. Diese Einstellung lässt sich mit dem Kommando SHOW-SDF-OPTIONS abfragen und mit der entsprechenden Angabe im Operanden CONTINUATION des Kommandos MODIFY-SDF-OPTIONS ändern. Bei Anweisungen kann das Fortsetzungszeichen in jedem Fall an beliebiger Stelle ab Spalte 2 stehen, und die Zeilenlänge kann mehr als 72 Zeichen betragen. Eingabesätze, die im Widerspruch zur SDF-Syntaxbeschreibung stehen, lösen Spin-Off aus, da im Stapelbetrieb kein Korrekturdialog möglich ist.

Die Definition task-spezifischer Default-Werte ist nicht möglich.

## 4.5 Ausdrücke in der Eingabe ersetzen

Der Wert eines Prozedurparameters, eines S-Variablenausdrucks oder einer Jobvariablen kann Kommando- oder Anweisungsteile ersetzen. Der zu ersetzende Ausdruck ist gekennzeichnet durch ein &-Zeichen, dem unmittelbar der Name des Prozedurparameters bzw. der Jobvariablen oder ein in Klammern gesetzter S-Variablenausdruck folgt. SDF ersetzt den Ausdruck vor der Ausführung durch den entsprechenden Wert. Die daraus resultierende Eingabe wird von SDF syntaktisch geprüft. Die Ersetzung von Ausdrücken ist im ungeführten Dialog, in Prozeduren und im Stapelbetrieb möglich. Im (temporär) geführten Dialog ist sie nur in der NEXT-Zeile und in der Eingabe für Operandenwerte erlaubt.

### Prozedurparameter ersetzen

Ausdrücke der Form „&parameter“ werden in Nicht-S-Prozeduren durch den Wert ersetzt, der *parameter* im Kommando BEGIN-PROCEDURE, CALL-PROCEDURE bzw. beim Prompting zugewiesen wurde. Ersetzt werden Ausdrücke in Kommandos. Innerhalb von Eingabedaten (werden von SYSDTA gelesen) werden Ausdrücke nur ersetzt, wenn ein Escape-Zeichen im Operanden ESCAPE-CHARACTER des BEGIN-PROCEDURE-Kommandos vereinbart wurde (#, @, & bzw. \*) und der Ausdruck mit diesem Zeichen beginnt. Innerhalb von Anweisungen (von SYSSTMT gelesen) werden Ausdrücke derzeit wie in Kommandos ersetzt. Es wird jedoch empfohlen ESCAPE-CHARACTER='&' zu vereinbaren.

Für die Ersetzung von Prozedurparametern bestehen folgende Einschränkungen:

- Innerhalb von CJC-Kommandofolgen ist das Ersetzen nicht möglich.
- Prozedurparameter können in Prozeduren oder ENTER-Dateien den führenden Schrägstrich vor Kommandos bzw. die zwei führenden Schrägstriche vor Anweisungen, den führenden Punkt von Nicht-S-Marken, den Strichpunkt als Kommandotrenner und das Fortsetzungszeichen nicht ersetzen.
- Ausdrücke können nicht geschachtelt werden.
- Bei Verdoppelung des &-Zeichens bzw. des Escape-Zeichens wird keine Ersetzung durchgeführt und das zweite &- bzw. Escape-Zeichen entfällt.

## Jobvariablen ersetzen

Diese Funktion ist nur verfügbar, wenn das kostenpflichtige Substem JV geladen ist.

Ausdrücke zur Ersetzung von Jobvariablen werden angegeben:

- direkt über Jobvariablennamen in der Form „&(jv-name)“.
- indirekt über ihren Jobvariablenkettungsnamen in der Form „&>(\*jv-link)“. Die Verknüpfung von Kettungsname mit der Jobvariablen erfolgt vorher durch das Kommando SET-JV-LINK LINK-NAME = jv-link, JV-NAME = jv-name.

Für die Ersetzung von Jobvariablen bestehen folgende Einschränkungen:

- Eine Jobvariable kann nur in ihrer gesamten Länge einen Ausdruck ersetzen.
- Für die einzusetzende Jobvariable muss Lesezugriff bestehen, sonst wird die Eingabe als Syntaxfehler abgewiesen.
- Innerhalb von CJC-Kommandofolgen ist das Ersetzen nicht möglich.
- Jobvariablen können keine Eingabedaten ersetzen. Anweisungen an Programme mit SDF-Oberfläche werden von SDF nicht wie Eingabedaten, sondern wie Kommandos behandelt.
- Jobvariablen können in Prozeduren oder ENTER-Dateien den führenden Schrägstrich vor Kommandos bzw. die zwei führenden Schrägstriche vor Anweisungen, den führenden Punkt von Nicht-S-Marken, den Strichpunkt als Kommandotrenner und das Fortsetzungszeichen nicht ersetzen.
- Jobvariablen können nicht als Prozedurparameter eingesetzt werden. Diese Einschränkung kann z.B. durch Verwendung eines Kettungsnamens umgangen werden (siehe Beispiel 3 [auf Seite 92](#)).
- Ausdrücke können nicht geschachtelt werden.
- Im Dialog und in S-Prozeduren wird eine Jobvariablen-Ersetzung in der oben gezeigten Form nur durchgeführt, wenn keine gleich lautende S-Variable bzw. Builtin-Funktion bekannt ist. Diese Form kann aber ersetzt werden durch die Builtin-Funktion JV( ). Die Angabe besitzt dann die Form „&(JV(JV-NAME=<c-string 1..54>))“. Siehe auch die vordefinierte Funktion (Builtin-Funktion) JV( ) im Handbuch „SDF-P“ [6] bzw. im Handbuch „Kommandos, Band 6“ [2]. Enthält der Jobvariablenname die Katalog- und/oder die Benutzerkennung, kann nur eine Jobvariablen-Ersetzung durchgeführt werden.



## S-Variablenausdrücke ersetzen

Ausdrücke der Form „&(ausdruck)“ werden im Dialog und in S-Prozeduren ersetzt durch den Wert von *ausdruck*, wobei *ausdruck* der Name eine S-Variablen, einer vordefinierten Funktion (Builtin-Funktion) oder ein gültiger S-Variablenausdruck sein kann. Für eine S-Variablen, deren Name keinen Punkt enthält, kann der Ausdruck auch in der Form „&s-variable“ (also ohne runde Klammern) angegeben werden.

Prozedurparameter einer S-Prozedur sind S-Variablen und werden ebenfalls in Ausdrücken ersetzt.

Ersetzt werden Ausdrücke in Kommandos. Innerhalb von Eingabedaten (werden von SYSDTA gelesen) werden Ausdrücke nur ersetzt, wenn ein Escape-Zeichen im Operanden DATA-ESCAPE-CHARACTER des Kommandos SET-PROCEDURE-OPTIONS vereinbart wurde (#, @, & bzw. \*) und der Ausdruck mit diesem Zeichen beginnt.

Innerhalb von Anweisungen (von SYSSTMT gelesen) werden Ausdrücke derzeit wie in Kommandos ersetzt.

Ausdrücke können geschachtelt werden.

Für Ersetzung von S-Variablenausdrücken bestehen folgende Einschränkungen:

- Kontrollflusskommandos können nicht erzeugt werden.
- Der zu ersetzende S-Variablenausdruck wird in den Typ STRING konvertiert.
- S-Marken können nicht erzeugt werden.
- Innerhalb von CJC-Kommandofolgen ist das Ersetzen nicht möglich.
- S-Variablenausdrücke können in Prozeduren und ENTER-Dateien den führenden Schrägstrich vor Kommandos bzw. die zwei führenden Schrägstriche vor Anweisungen, den führenden Punkt von Nicht-S-Marken, den Strichpunkt als Kommandotrenner und das Fortsetzungszeichen nicht ersetzen.
- Bei Verdoppelung des &-Zeichens bzw. des Escape-Zeichens wird keine Ersetzung durchgeführt und das zweite &- bzw. Escape-Zeichen entfällt.

## Beispiele

### 1. Ersetzen von Jobvariablen im Dialog:

```
/cre-jv jv=cmd _____ (1)
```

```
/mod-jv jv=cmd,set-value='SHOW-FILE-ATTR' _____ (2)
```

```
/&(cmd) _____ (3)
```

```
%      3 :20SG:$USER1.ALT.SYS.LOGON.USERPROC.X1
%     51 :20SG:$USER1.ALT.SYSSDF.USER.EXAMPLE.1
%     21 :20SG:$USER1.DATEI.1
%     48 :20SG:$USER1.DATEI.2
%     84 :20SG:$USER1.DATEI.3
%     66 :20SG:$USER1.OUT.SORT1-2
%      3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      7 FILES RES=      276 FRE=      39 REL=      21 PAGES
```

```
/mod-jv jv==egon,set-value='-FILE-ATTR F-NAME=PROC.' _____ (4)
```

```
/set-jv-link link-name=walter,jv-name=egon _____ (5)
```

```
/sh&(*walter) _____ (6)
```

```
%      3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
```

- (1) Es wird der Jobvariablenname 'CMD' vereinbart.
- (2) Die Jobvariable CMD erhält den Wert 'SHOW-FILE-ATTR' (Abkürzung des Kommandonamens SHOW-FILE-ATTRIBUTES).
- (3) Nach Absenden des Kommandos wird der Variablenstring durch das in der Jobvariablen definierte Kommando ersetzt und das Kommando ausgeführt.
- (4) Der Wert der Jobvariablen CMD wird geändert. Er enthält jetzt nur noch einen Teil des Kommandonamens ('-FILE-ATTRIBUTES') und den teilqualifizierten Dateinamen 'PROC.'.
- (5) Der Jobvariablen CMD wird der Linkname 'WALTER' zugeordnet.
- (6) Nach Absenden des Kommandos wird der Variablenstring durch den der Jobvariablen zugeordneten Kommandoteil ersetzt und das Kommando ausgeführt. Der Bezug zur Jobvariablen wird durch den Linknamen hergestellt.

## 2. Ersetzen von Jobvariablen und S-Variablen im Dialog:

```

/sh-jv cmd ----- (1)
%SHOW-FILE-ATTR
/&(amp)cmd) proc.jv ----- (2)
%      3 :2OSG:$USER1.PROC.JV
%:2OSG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
/cmd='PRINT-DOCUMENT' ----- (3)
/sh-var cmd
CMD = PRINT-DOCUMENT
/&(amp)cmd) proc.jv ----- (4)
% SCP0810 SPOOLOUT FOR FILE ':2OSG:$USER1.PROC.JV' ACCEPTED. TSN: '1FAL', SPOOL
OUT-NAME: 'SDFTEST', MONJV: '*NONE'
% SCP1025 PRINT JOB ACCEPTED BY SERVER 'GH5090Y0' WITH TSN '5BXC'
/(&(:2osg:cmd) proc.jv ----- (5)
%      3 :2OSG:$USER1.PROC.JV
%:2OSG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
/(&(jv(jv-name='CMD')) proc.jv ----- (6)
%      3 :2OSG:$USER1.PROC.JV
%:2OSG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
/

```

- (1) Die Jobvariable `CMD` besitzt den Wert `SHOW-FILE-ATTR`.
- (2) Der Kommandoname wird aus der Jobvariablen `CMD` ersetzt, d.h., es wird `SHOW-FILE-ATTRIBUTES` für die Datei `PROC.JV` ausgeführt.
- (3) Die S-Variable `CMD` wird implizit durch Zuweisen des Wertes `PRINT-DOCUMENT` eingerichtet.
- (4) In der Eingabe wird der Ausdruck `&(CMD)` jetzt durch den Inhalt der S-Variablen `CMD` ersetzt, d.h., es wird das Kommando `PRINT-DOCUMENT` für die Datei `PROC.JV` ausgeführt.
- (5) Um bei der Ersetzung den Inhalt der Jobvariablen `CMD` zu erhalten, wird dem Namen der Jobvariablen die Katalogkennung vorangestellt.
- (6) Die Ersetzung der Jobvariableninhalts kann auch über die Builtin-Funktion `JV()` erreicht werden.

3. Den Namen einer zu ersetzenden Jobvariablen als Prozedurparameter einer Nicht-S-Prozedur übergeben:

```

/BEG-PROC PAR=*YES(PROC-PAR=( &PARAM1 )) _____ (1)
.
.
.
/SET-JV-LINK LINK-NAME=PARAM1 ,JV-NAME=&PARAM1 _____ (2)
/&(*PARAM1) FILE-NAME=LST.JOB _____ (3)
.
.
.
/END-PROC

```

- (1) Die durch den Prozedurparameter PARAM1 spezifizierte Jobvariable soll das jeweils auszuführende Kommando enthalten.  
Da die Angabe „&(&PARAM1)“ nicht zulässig ist, wird im Folgenden der Umweg über einen Kettungsnamen beschritten.  
Zur Möglichkeit eines verschachtelten Ausdrucks siehe Beispiel 4, Punkt (3).
- (2) Der aktuelle Jobvariablenname wird für den Prozedurparameter PARAM1 eingesetzt und mit dem Kettungsnamen PARAM1 verknüpft.
- (3) Für den Kettungsnamen PARAM1 wird der Inhalt der vereinbarten Jobvariablen eingesetzt. PRINT-DOCUMENT DOCUMENT-FORMAT=\*TEXT(LINE-SPACING=\*BY-EBCDIC-CONTROL), LAYOUT-CONTROL=\*PAR(ROTATION=90, LEFT-MARGIN=10) als Jobvariablenwert bewirkt z.B., dass die Datei LST.JOB wie angegeben ausgedruckt wird. Enthält die Jobvariable z.B. SHOW-FILE-ATTRIBUTES INFORMATION=\*PAR(HISTORY=\*YES, SECURITY=\*YES), werden die gewünschten Dateiattribute der Datei LST.JOB ausgegeben.

4. Ersetzung von Jobvariablen und S-Variablen in einer S-Prozedur, wobei der Jobvariablenname als Prozedurparameter übergeben wird:

*Inhalt der Prozedurdatei DO.JVTEST:*

```

/          SET-PROC-OPT  JV-REPLACE=*AFTER-BUILTIN
/          DECL-PAR      JV-1(INIT=*PROMPT)
/          &(JV(JV-NAME=JV-1))  FILE-NAME=LST.JOB
/FEHL:    IF-BLOCK-ERROR
/          WRITE-TEXT   C '** Fehler &MC **'
/
/          ELSE
/          WRITE-TEXT   C '** Kommando &(&(JV-1)) ausgefuehrt **'
/          END-IF
/ENDE:    EXIT-PROC

```

*Prozedurablauf:*

```

/sh-jv jv(cmd) ----- (1)
%SHOW-FILE-ATTR
/cal-proc do.jvtest,log=*yes
%          1 1 /SET-PROC-OPT  JV-REPLACE=*AFTER-BUILTIN
%          2 1 /DECL-PAR      JV-1(INIT=*PROMPT)
%JV-1: cmd ----- (2)
%          3 1 /SHOW-FILE-ATTR  FILE-NAME=LST.JOB ----- (3)
%          3 :N:$USER0001.LST.JOB
%:N:      PUBLIC:      1 FILE RES=          3 FREE=          3 REL=          3 PAGES
%          4 1 /FEHL:
%          4 1 / IF-BLOCK-ERROR
%          6 1 /ELSE
%          7 1 /WRITE-TEXT   C '** Kommando SHOW-FILE-ATTR ausgefuehrt **'
** Kommando SHOW-FILE-ATTR ausgefuehrt ** ----- (4)
%          8 1 /END-IF
%          9 1 /ENDE:
%          9 1 /  EXIT-PROC
/

```

- (1) Als Inhalt der Jobvariablen CMD wird SHOW-FILE-ATTR angezeigt.
- (2) Nach Aufruf der Prozedur DO.JVTEST wird die Prozedurparameter JV-1 gepromptet und der Wert CMD zugewiesen. Der Ausdruck wird ersetzt durch den Wert der Jobvariablen, den die Builtin-Funktion JV( ) ermittelt. Der gesuchte Jobvariablenname wird über die S-Variable JV-1 an die Builtin-Funktion übergeben.
- (3) Es wird das Kommando SHOW-FILE-ATTRIBUTES ausgeführt.
- (4) An dieser Stelle wurde ein verschachtelter Ausdruck verwendet: Eine Jobvariable soll ersetzt werden, wobei sich der Name aus dem zu ersetzenden Ausdruck für die S-Variable JV-1 ergibt. Es wird also der Inhalt der Jobvariablen CMD ersetzt. Voraussetzung ist, dass keine gleichnamige S-Variable oder Builtin-Funktion existiert und im SET-PROCEDURE-OPTION-Kommando die Jobvariablen-Ersetzung explizit eingestellt ist.

## 4.6 Verkürzte Eingabe

SDF bietet die Möglichkeit, die Kommando- bzw. Anweisungseingabe sowohl im Dialog- als auch im Stapelbetrieb abzukürzen.

Eine heute bestehende Eindeutigkeit kann allerdings bei Funktionserweiterungen in einer neuen BS2000-Version wieder aufgehoben sein. In automatisierten Kommandofolgen sollten deshalb nur eingeschränkt Abkürzungen verwendet werden.

### 4.6.1 Namen abkürzen

Die Abkürzungsmöglichkeit besteht für alle verwendeten Namen (Schlüsselwörter):

- Kommando- bzw. Anweisungsnamen
- Operandennamen
- Schlüsselwortwerte

Die Namen können wie folgt abgekürzt werden:

- Bei mehrteiligen Namen (Teilnamen mit Bindestrich verbunden) können von rechts nach links Teilnamen weggelassen werden. Mit einem Teilnamen entfällt auch der zugehörige Bindestrich.
- Innerhalb eines Teilnamens oder eines einteiligen Namens können von rechts nach links Zeichen weggelassen werden.
- Ein Stern zu Beginn eines Namens gehört nicht zum Namen. Er unterscheidet nur den konstanten Operandenwert von dem variablen Operandenwert, dessen Wertebereich die Zeichenfolge des konstanten Wertes beinhaltet. Der Stern allein stellt selbst bei gegebener Eindeutigkeit keine gültige Abkürzung dar.
- Ab SDF V4.0A werden Schlüsselwortwerte in der Dialogführung und in der Syntaxdarstellung immer mit führendem Stern dargestellt. Der führende Stern eines Schlüsselwortwertes kann weggelassen werden, wenn alternativ kein variabler Operandenwert möglich ist, dessen Wertebereich den Namen des Schlüsselwortwertes 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.
- Ab SDF V4.1A kann der Name bzw. ein Teilname eines Schlüsselwortwertes auch das Zeichen Punkt enthalten (z.B. \*V4.5 oder \*OSD-V5.0). Der Punkt ist Bestandteil des Namens bzw. des Teilnamens. Bei der Abkürzung des Namens darf der Punkt nicht am Ende des Namens bzw. des Teilnamens stehen.

Damit SDF die abgekürzten Namen richtig interpretieren kann, müssen die gewählten Abkürzungen innerhalb der jeweils unmittelbaren Syntaxumgebung eindeutig sein. Für bestimmte Namen kann jedoch in der Syntaxdatei eine Minimalabkürzung festgelegt sein. SDF akzeptiert dann auch bei gegebener Eindeutigkeit keine kürzere Eingabe.

Eine eindeutige Zuordnung ist in folgenden Fällen gegeben:

- ein Kommandoname ist eindeutig innerhalb aller möglichen Kommandonamen. Wird ein Teilname eines Kommandos vollständig angegeben, so ist dieses Kommando eindeutig gegenüber einem zweiten Kommando, bei dem der angegebene Teilname eine Abkürzung desselben Namentails darstellt. Existieren z.B. die Kommandos START-C-COMPILER und START-COBOL-COMPILER, dann bezeichnet die Eingabe START-C-COMP eindeutig das Kommando START-C-COMPILER.
- ein Anweisungsname ist eindeutig innerhalb aller möglichen Anweisungen eines geladenen Programmes.
- ein Operandenname ist eindeutig innerhalb aller möglichen Operanden des angegebenen Kommandos bzw. der Anweisung auf der gleichen Strukturstufe, für einen Operandenname in einer untergeordneten Struktur werden nur alle möglichen Operandennamen dieser Struktur betrachtet.
- ein Schlüsselwortwert ist eindeutig innerhalb der Menge aller für den angegebenen Operanden möglichen Operandenwerte.

Beispielsweise ist die Eingabe `/MOD-SDF-OPT SYN-F=*NONE , GUI=*MIN` eine mögliche Abkürzung für `/MODIFY-SDF-OPTIONS SYNTAX-FILE=*NONE , GUIDANCE=*MINIMUM`

Die Handbücher führen so genannte garantierte Abkürzungen auf (Hervorhebung durch Fettdruck). Die garantierten Abkürzungen sind nicht die kürzest möglichen. Sie sind jedoch noch aussagefähig und werden langfristig eindeutig gehalten.

Für alle anderen möglichen Abkürzungen kann eine solche Garantie nicht gegeben werden. In Prozeduren sollen deshalb nur die Langformen oder die garantierten Abkürzungen verwendet werden (verbessert auch die Lesbarkeit der Prozeduren!).

Neben dem Kommando- bzw. Anweisungsnamen kann im Handbuch zusätzlich ein Kurzname dokumentiert sein. Der Kurzname ist als Aliasname des Kommandos bzw. der Anweisung implementiert und wird langfristig garantiert. Der Kurzname besteht aus maximal 8 Zeichen (A...Z), die aus dem Kommando- bzw. Anweisungsnamen abgeleitet sind. Eine Abkürzung des Kurznamens ist nicht möglich.

Beispiel: **MDSDFO** statt **MODIFY-SDF-OPTIONS**

Die in den Handbüchern angegebenen Namen sind in den Syntaxdateien zusätzlich als so genannte Standardnamen definiert. Auch nach einer möglichen Umbenennung der Kommandonamen werden die Standardnamen weiterhin akzeptiert, allerdings nur in der Langform. Beispielsweise wird nach einer Umbenennung des Kommandos **CREATE-FILE** in **ERZEUGE-DATEI** weiterhin die Eingabe `/CREATE-FILE` akzeptiert, jedoch nicht mehr die Eingabe `/CR-F`. Sollen Prozeduren gegen Umbenennungen invariant bleiben so, müssen alle Namen in der Langform angegeben werden.

## 4.6.2 Default-Wert übernehmen

Die Angabe der meisten Operanden ist wahlfrei. Wahlfreie Operanden sind bereits mit einem Operandenwert vorbesetzt, dem so genannten Default-Wert (Standardwert). Erfolgt keine explizite Angabe zu einem wahlfreien Operanden, so wird zur Ausführung des Kommandos bzw. der Anweisung für diesen Operanden der Default-Wert gesetzt. Die Operandenwerte \*UNCHANGED bzw. \*CURRENT stehen für die bereits vorgenommene Einstellung, d.h. eine Einstellung wird als Wert unverändert übernommen.

Da nur Angaben zu Operanden, die keine Default-Werte besitzen sollen, gemacht werden müssen, kann die Eingabe in den meisten Fällen wesentlich verkürzt werden.

Soweit Operanden keine bzw. nicht die gewünschten Default-Werte besitzen kann der Benutzer für die Eingabe im Dialog task-spezifisch eigene Default-Werte festlegen (siehe [Abschnitt „Task-spezifische Default-Werte“ auf Seite 110](#)).

Beispielsweise ist die Angabe `MOD-SDF-OPT SYN=*N, GUID=*MIN` eine mögliche Abkürzung für:

```
MOD-SDF-OPT SYN=*NONE, GUID=*MIN, LOG=*UNCH, UT=*UNCH, PROC=*UNCH,  
CONT=*UNCH, MENU=*UNCH, MODE=*UNCH, DEFAULT-PROG=*UNCH, FUNCTION-  
KEYS=*UNCH, INPUT-HISTORY=*UNCH
```

### Hinweis

Der Default-Wert eines Operanden ist nicht zu verwechseln mit dem Operandenwert \*STD. Der Operandenwert \*STD muss nicht der voreingestellte Default-Wert sein. Die Bedeutung von \*STD ist im Einzelfall der jeweiligen Operandenbeschreibung zu entnehmen. \*STD kann beispielsweise für einen bei der Systeminstallation eingestellten Wert stehen (z.B. `SPACE=*STD` im Kommando `CREATE-FILE`) oder für eine vom Taskmodus abhängige Einstellung (z.B. `DIALOG-CONTROL=*STD` im Kommando `DELETE-FILE`).



### 4.6.3 Stellungsoperanden angeben

Jeder Operand kann wahlweise als Schlüsselwort- oder als Stellungsoperand angegeben werden. Die Angabe als Schlüsselwortoperand bedeutet, dass der Operandenname und der gewünschte Wert zusammen in der Form <operandenname>=<operandenwert> angegeben werden. Bei Angabe als Stellungsoperand wird nur der Operandenwert angegeben. Die Zuordnung erfolgt allein über die Stellung innerhalb der Eingabe im Vergleich zur Kommando- bzw. Anweisungsdefinition. Bei der Eingabe von Stellungsoperanden ist zu beachten:

- Für jeden weggelassenen Operanden vor einem Stellungsoperanden ist ein Komma einzugeben.
- Nach einem Schlüsselwortoperanden können auf der gleichen Strukturebene keine Stellungsoperanden angegeben werden.

Beispielsweise ist die Eingabe `MOD-SDF-OPT *NONE, *MIN` eine mögliche Abkürzung für `MOD-SDF-OPT SYN-F=*NONE, GUID=*MIN`.

Es kann nicht völlig ausgeschlossen werden, dass sich bei einem Versionswechsel eine Operandenposition ändert. In Prozeduren sollten deshalb nur Schlüsselwortoperanden verwendet werden.

## 4.6.4 Strukturen abkürzen

Bei der Angabe von Strukturen bestehen folgende Möglichkeiten, die Eingabe zu verkürzen:

### STRUCTURE-IMPLICIT-Schreibweise

Die Angabe des struktureinleitenden Operanden unterbleibt und der untergeordnete Operand wird außerhalb der Strukturklammern angegeben.

Voraussetzung ist im Allgemeinen, dass der untergeordnete Operand kommando- bzw. anweisungsglobal oder in Bezug auf eine übergeordnete Struktur eindeutig ist. Operanden, für die die STRUCTURE-IMPLICIT-Schreibweise langfristig garantiert ist, werden in der entsprechenden Kommando- bzw. Anweisungsbeschreibung explizit aufgeführt.

*Beispiel*

```
SHOW-FILE-ATTR ACCESS-METHOD=*ISAM
```

ist die verkürzte Schreibweise von

```
SHOW-FILE-ATTR SEL=*BY-ATTR(ACCESS-METHOD=*ISAM)
```

### Hinweis

In vielen Fällen, in denen die STRUCTURE-IMPLICIT-Schreibweise nicht möglich ist, kann die nachfolgend beschriebene „Flache Schreibweise“ angewendet werden.

### Flache Schreibweise

Der struktureinleitende Operand wird angegeben. Die Angabe des untergeordneten Operanden erfolgt jedoch außerhalb der Strukturklammern.

Der untergeordnete Operand muss nicht kommando- bzw. anweisungsglobal eindeutig definiert sein.

Er darf jedoch nicht in mehreren aktivierten Strukturen vorkommen. Die Aktivierung einer Struktur kann sowohl durch die explizite Angabe als auch implizit über den Default-Wert, falls ein Operand nicht angegeben wird, erfolgen!

Die flache Schreibweise ist nicht langfristig garantiert!

*Beispiel*

```
CRE-FILE DATEI1 , SUP=*PRIV-DISK , VOL=ABC123 , DEV-TYPE=D3475
```

ist die verkürzte Schreibweise von

```
CRE-FILE DATEI1 , SUP=*PRIV-DISK(VOL=ABC123 , DEV-TYPE=D3475)
```

**Schreibweise bei Attribut NULL-ABBREVIATION=\*YES**

Der untergeordnete Operand wird innerhalb der Strukturklammern angegeben, der struktureinleitende Operandenwert wird jedoch bis auf die leere Zeichenfolge abgekürzt, d.h. er wird nicht angegeben.

Voraussetzung ist, dass der struktureinleitende Operandenwert in der Syntaxdatei mit dem Attribut NULL-ABBREVIATION=\*YES definiert ist. Dieses Attribut kann innerhalb der Menge der möglichen Operandenwerte nur einmal vergeben werden (falls mehrere struktureinleitende Operandenwerte zugelassen sind; siehe Handbuch „SDF-A“ [4]).

Operandenwerte, für die das Attribut NULL-ABBREVIATION langfristig garantiert ist, sind in der Syntaxdarstellung in eckige Klammern eingeschlossen (siehe „SDF-Syntaxdarstellung“, [Seite 29](#)).

*Beispiel*

```
MOD-FILE-ATTR DATEI1 , PROTECTION=(ACCESS=*READ)
```

ist die verkürzte Schreibweise von

```
MOD-FILE-ATTR DATEI1 , PROTECTION=*PARAMETERS(ACCESS=*READ)
```

## 4.7 Protokollierung der Eingabe

Die Protokollierung des Ablaufs eines Dialogauftrags auf SYSLST wird mit dem Operanden LOGGING des Kommandos SET-LOGON-PARAMETERS (bei Stapelauftrag ENTER-JOB) bzw. MODIFY-JOB-OPTIONS eingestellt. Bei eingeschalteter Protokollierung wird der Protokollierungsumfang von Eingaben an SDF und die Ausgaben von SDF über die Einstellung der SDF-Parameter LOGGING und MENU-LOGGING gesteuert. Die Veränderung dieser Parametereinstellung kann durch explizite Angabe der gewünschten Einstellung bei den entsprechenden Operanden des Kommandos MODIFY-SDF-OPTIONS erfolgen.

### 4.7.1 LOGGING-Einstellung

Bei LOGGING=\*INVARIANT-FORM werden protokolliert:

- alle Namen in der Fassung, die in den Handbüchern angegeben ist
- alle in der Eingabe vorkommenden Operanden mit ihren Namen und den angegebenen Werten
- alle in der Eingabe implizit enthaltenen wahlweisen Operanden mit ihren Default-Werten
- benutzerspezifische Kommando- bzw. Anweisungsdefinitionen werden durch die entsprechenden systemspezifischen ersetzt
- nur das von SDF akzeptierte Endresultat eines durchgeführten Korrekturdialogs. Im geführten Dialog erfolgte Eingaben werden zu einem String verkettet.

Als SECRET definierte Operandenwerte, wie auch verdeckt erfolgte Eingaben werden im Protokoll durch „P“ ersetzt. In der INVARIANT-Form bleibt der Auftragsablauf anhand des Protokolls nachvollziehbar und verständlich.

Bei LOGGING=\*ACCEPTED-FORM werden protokolliert:

- alle Namen in ihrer Langform (Falls mit SDF-A ein Name in der Syntaxdatei geändert wurde, so wird der neue Name protokolliert und nicht der in den Handbüchern angegebene.)
- jeder in der Eingabe vorkommende Operand mit seinem Namen und dem angegebenen Wert
- nur das von SDF akzeptierte Endresultat eines durchgeführten Korrekturdialogs. Im geführten Dialog erfolgte Eingaben werden zu einem String verkettet.

Als SECRET definierte Operandenwerte, wie auch verdeckt erfolgte Eingaben werden im Protokoll durch „P“ ersetzt. In der ACCEPTED-Form bleibt der Auftragsablauf anhand des Protokolls ebenfalls noch nachvollziehbar und verständlich.

Bei LOGGING=\*INPUT-FORM werden im ungeführten Dialog die Eingabezeichenfolgen exakt protokolliert. SECRET-Operanden werden ausgeblendet. Im geführten Dialog oder beim Korrekturdialog wird wie bei ACCEPTED-FORM protokolliert.

## 4.7.2 MENU-LOGGING-Einstellung

Bei Angabe von \*YES (im Gegensatz zu \*NO) werden alle ausgegebenen Bildschirme (Menü- und Operandenfragebogen) exakt mitprotokolliert. Diese Funktion ist für Diagnosezwecke vorgesehen. Zu beachten ist, dass in diesem Fall die Menge der zu protokollierenden Datensätze auf SYSLST sehr große Ausmaße annehmen kann.

## 4.7.3 Einschränkungen

In folgenden Fällen kann der Benutzer die Protokollierung nicht beeinflussen:

- Die Protokollierung ist in der Kommandodefinition untersagt worden. Ob bzw. wie protokolliert wird ist abhängig von dem ausführenden Modul.
- Das Kommando bzw. die Anweisung enthalten Operanden, die als „geheim“ definiert wurden (z.B. Kennwörter). Kann die Angabe zu einem „geheimen“ Operanden diesem eindeutig zugeordnet werden, wird als Operandenwert der Wert „P“ im Protokoll eingesetzt. Kann keine eindeutige Zuordnung getroffen werden, wird nur der Kommando- bzw. Anweisungsname ins Protokoll übernommen. Keine eindeutige Zuordnung bedeutet, dass der Kommando- bzw. Anweisungsname erkannt werden, aber die nachfolgenden Operanden einen Syntaxfehler enthalten.
- Der Kommando- bzw. Anweisungsname können nicht erkannt werden. In diesem Fall wird nur der fehlerhafte Kommando- bzw. Anweisungsname in das Protokoll übernommen, da Angaben zu geheimen Operanden enthalten sein könnten.
- Prozeduren können nur protokolliert werden, wenn dies innerhalb der Prozedur zugelassen wurde. Darüber hinaus werden S-Prozeduren nur dann protokolliert, wenn der Aufrufer auch die Berechtigung zum lesenden Zugriff auf die Prozedurdatei besitzt.

## 4.8 Auswahlmenü für Kommandos und Anweisungen

Die Auswahl von Kommandos kann der Benutzer über ein Anwendungsbereichs-Menü vornehmen. Darin sind alle Kommandos zu einem Themengebiet enthalten. Innerhalb eines Programms kann der Benutzer die Auswahl über das Anweisungs-menü vornehmen. Eine weitere Auswahlmöglichkeit besteht bei der Verwendung von Musterzeichen oder mehrdeutigen Abkürzungen in Kommando- bzw. Anweisungsnamen.

### 4.8.1 Musterzeichen in Kommando- oder Anweisungsnamen

Bei Eingabe eines Kommando- bzw. Anweisungsnamens mit Musterzeichen und einem abschließenden Fragezeichen erhält der Benutzer ein Auswahlmenü aller Kommandos bzw. Anweisungen, die der angegebenen Musterzeichenfolge entsprechen. Das Auswahlmenü wird auch ausgegeben, wenn genau ein Kommando oder eine Anweisung getroffen wird. Existiert kein entsprechendes Kommando oder Anweisung, erhält der Benutzer eine Fehlermeldung.

Das Auswahlmenü existiert nur bis zur Ausführung des Kommandos oder der Anweisung. Danach befindet sich der Benutzer im ursprünglichen Eingabemodus. Ein Auswahlmenü mit gleichem Inhalt kann mit derselben Musterzeichenfolge wieder angefordert werden.

Mithilfe von Musterzeichen kann der Benutzer ein Kommando bzw. eine Anweisung ohne genaue Kenntnis des Namens finden. Er kann auch ein Auswahlmenü nach eigenen Auswahlkriterien anfordern. Beispielsweise liefert die Eingabe „\*job\*?“ alle Kommandos, deren Name die Zeichenfolge JOB enthält.

Bei der Verwendung von Musterzeichen ist Folgendes zu beachten:

- Alle Zeichen, die auf das Fragezeichen folgen, werden ignoriert.
- Das Musterzeichen „?“ wird nicht erkannt. Es wird als abschließendes Fragezeichen interpretiert, d.h. nachfolgende Zeichen werden ignoriert.
- Eine Musterzeichenfolge für einen Kommando- bzw. Anweisungsnamen darf maximal 30 Zeichen lang sein.
- Das Musterzeichen „/“ zu Beginn eines Kommandonamens wird als Kommando-Prompt interpretiert.
- Die Zeichenfolge „/“ zu Beginn eines Anweisungsnamens wird als Anweisungs-Prompt interpretiert.
- Abkürzungen innerhalb einer Musterzeichenfolge sind nicht möglich. Beispielsweise wird durch die Eingabe „m-sdf-o\*?“ nicht das Kommando MODIFY-SDF-OPTIONS gefunden. Richtig wäre die Eingabe „m\*-sdf-o\*?“.

Abbruch des Auswahlmenüs mit \*EXIT, \*EXIT-ALL, \*CANCEL oder Drücken der entsprechenden Funktionstasten bewirkt im geführten Dialog die Rückkehr in das Anwendungsbereichs- bzw. Anweisungs-menü. Ein temporärer Dialog wird beendet.

## 4.8.2 Mehrdeutige Abkürzung von Kommando- oder Anweisungsnamen

Bei Eingabe eines abgekürzten Kommando- bzw. Anweisungsnamens mit abschließendem Fragezeichen erhält der Benutzer ein Auswahlmenü mit allen betroffenen Kommandos bzw. Anweisungen.

Zeichen nach dem Fragezeichen werden ignoriert.

Entspricht die Abkürzung genau einem Kommando bzw. einer Anweisung erhält der Benutzer den Operandenfragebogen, wobei Zeichen nach dem Fragezeichen als Operandenangaben übernommen werden. Existiert kein Kommando oder keine Anweisung, erhält der Benutzer eine Fehlermeldung.

Abbruch des Auswahlmenüs mit \*EXIT, \*EXIT-ALL, \*CANCEL oder Drücken der entsprechenden Funktionstasten bewirkt im geführten Dialog die Rückkehr in das Anwendungsbeereichs- bzw. Anweisungsmenü. Ein temporärer Dialog wird beendet.

## 4.9 Vorangegangene Eingaben wieder verwenden

Syntaktisch richtige Kommandos oder Anweisungen werden von SDF task-lokal gespeichert. Mit dem Operanden INPUT-HISTORY im Kommando MODIFY-SDF-OPTIONS wird der Eingabepuffer ein- bzw. ausgeschaltet oder zurückgesetzt. Außerdem legt er fest, wie viele Eingaben maximal gespeichert werden sollen. Bei Erreichen der maximalen Anzahl gespeicherter Eingaben wird die jeweils älteste Eingabe gelöscht. Die gespeicherten Eingaben werden automatisch nummeriert (Eingabeseriennummer).

Eingaben im geführten Dialog werden in der ACCEPTED-Form gespeichert, Eingaben im ungeführten Dialog in der INPUT-Form. Nicht gespeichert werden die Kommandos bzw. Standardanweisungen SHOW-INPUT-HISTORY und RESTORE-SDF-INPUT. ISP-Kommandos werden nur gespeichert, wenn die Einstellung PASSWORD-PROTECTION auf \*NO gesetzt wurde.

Mit dem Kommando (bzw. der Standardanweisung) SHOW-INPUT-HISTORY kann sich der Benutzer über gespeicherte Eingaben informieren.

Das Kommando RESTORE-SDF-INPUT gibt eine bestimmte gespeicherte Eingabe erneut am Bildschirm aus. Der Benutzer kann das ausgegebene Kommando bzw. die Anweisung unverändert oder auch geändert als Eingabe verwenden. Er erspart sich dabei die Eingabe des gesamten Kommandos bzw. der Anweisung.

Mit INPUT-HISTORY=\*OFF schaltet der Benutzer den Eingabepuffer aus. Bereits gespeicherte Eingaben bleiben erhalten.

Nach INPUT-HISTORY=\*ON werden Eingaben wieder gespeichert.

Mit INPUT-HISTORY=\*RESET kann der Eingabepuffer zurückgesetzt werden, d.h. bereits gespeicherte Eingaben werden gelöscht und nachfolgende Eingaben weiter gespeichert.

### Schutz von „geheimen“ Operanden

Wertangaben für „geheime“ Operanden, die weder dem Default-Wert noch einem mit SECRET=\*NO definierten Wert entsprechen, werden im Eingabepuffer mit „^“ gespeichert. Im ungeführten Dialog hat der Benutzer bei der Wiederanzeige mit RESTORE-SDF-INPUT folgende Möglichkeiten:

- Er kann das Kommando bzw. die Anweisung unverändert abschicken. In diesem Fall gibt SDF für jeden geheimen Operanden ein dunkelgesteuertes Eingabefeld aus, in das der Benutzer den gewünschten Wert eingeben kann.
- Er kann das „^“ löschen, dafür den gewünschten Wert direkt einfügen und danach das Kommando bzw. die Anweisung abschicken.



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.

### **Schutz von „geheimen“ Operanden aufheben**

Nur im interaktiven Dialog kann der Benutzer mit INPUT-HISTORY=\*ON(PASSWORD-PROTECTION=\*NO) den Schutz von „geheimen“ Operanden aufheben. Danach werden Wertangaben für „geheime“ Operanden im Klartext gespeichert. ISP-Kommandos werden danach ebenfalls im Eingabepuffer gespeichert.

Bei dieser Einstellung werden Kennwörter mit SHOW-INPUT-HISTORY bzw. RESTORE-SDF-INPUT am Bildschirm im Klartext ausgegeben und können dadurch ggf. unbefugten Personen bekannt werden. Vor allem sollte der Benutzer vor jedem Verlassen des Bildschirmarbeitsplatzes dafür sorgen, dass sich keine unbefugte Person den Inhalt des Eingabepuffers ausgeben lassen kann. Besitzt der Bildschirmarbeitsplatz keine entsprechenden Sicherungsfunktionen (z.B. Chipkarten-Terminal), sollte vorher mindestens der Eingabepuffer gelöscht werden.

## 4.10 Testmodus

Mit der Angabe \*TEST bzw. \*EXECUTE im Operanden MODE des Kommandos MODIFY-SDF-OPTIONS kann der Testmodus ein- bzw. ausgeschaltet werden. Bei eingeschaltetem Testmodus werden die nachfolgenden Kommandos von SDF nur syntaktisch abgeprüft, aber nicht ausgeführt. Zusätzlich kann in dem untergeordneten Operanden CHECK-PRIVILEGES eingestellt werden, ob die Privilegien der Task bei der Syntaxanalyse berücksichtigt werden sollen.

Die Kommandos MODIFY-SDF-OPTIONS und SHOW-SDF-OPTIONS werden als einzige ausgeführt. Damit können die SDF-Einstellungen während des Tests verändert bzw. der Testmodus wieder ausgeschaltet und Informationen über aktuelle Einstellungen eingeholt werden.

### *Hinweis*

Das kostenpflichtige Subsystem SDF-P stellt dem Benutzer für S-Prozeduren geeignete Testhilfen zur Verfügung (siehe Handbuch „SDF-P“ [6]). Der hier beschriebene Testmodus wird in S-Prozeduren mit folgender Einschränkungen unterstützt:

Eingaben werden immer zuerst von SDF-P analysiert. Anweisungen können nicht getestet werden. SDF-P-Kontrollflusskommandos werden ausgeführt. Dies kann zu Fehlern führen, da z.B. Kommandos zum Setzen bzw. Deklarieren von S-Variablen im Testmodus nicht ausgeführt werden.

Jobvariablen, S-Variablen und Prozedurparameter werden vor der Syntaxanalyse ersetzt.

Im Prozedur- und Stapelbetrieb werden Eingabezeilen ohne führenden „/“ als Datensätze interpretiert und ignoriert. Ein Datensatz bzw. eine Folge von Datensätzen wird von den Meldungen CMD0091 und CMD0092 umschlossen, um nicht ausgewertete Eingabesätze kenntlich zu machen.

Die Angabe eines Programmnamens im Operanden DEFAULT-PROGRAM-NAME im Kommando MODIFY-SDF-OPTIONS bewirkt die Erweiterung der Syntaxanalyse auf Anweisungen an ein Programm mit SDF-Oberfläche. Sie wirkt jedoch nur im Prozedur- oder Stapelbetrieb und bei eingeschaltetem Testmodus.

Eingabezeilen mit führenden „/“ werden als Programmanweisungen interpretiert und ebenfalls syntaktisch analysiert. Die Syntaxanalyse der Anweisungen erfolgt anhand der Anweisungen, die für den angegebenen Programmnamen in der Syntaxdatei definiert wurden (**nicht** abhängig vom Dateinamen, der im Kommando START-PROGRAM verwendet wird). Sollen Anweisungen mehrerer Programme ausgewertet werden, so ist die Einstellung DEFAULT-PROGRAM-NAME vor dem jeweiligen Anweisungsblock entsprechend zu ändern.

Die Beschränkung der Syntaxanalyse auf Kommandos kann mit DEFAULT-PROGRAM-NAME=\*NONE wieder erreicht werden.

Die Unterbrechung einer Prozedur mit der K2-Funktionstaste wird auch bei eingeschaltetem Testmodus ausgeführt. Die Rückkehr mit dem Kommando RESUME-PROCEDURE ist bei eingeschaltetem Testmodus nicht möglich.

Bei Beendigung einer Prozedur im Testmodus wird der Benutzer mit der Meldung CMD0093 darauf hingewiesen, dass der Testmodus noch eingeschaltet ist.

### Beispiel

Inhalt der Prozedurdatei „PROC.SELECT“:

```

/BEG-PROC          LOG=*ALL -
/                  ,PAR=*YES(      -
/                  PROC-PAR=(&MODE=*EXEC,-
/                  &TESTPROGRAM=PERCON,-
/                  &OUTFILE, &ELEM)-
/                  ,ESC-CHAR=C' '&' -
/                  )
/ASS-SYSDTA        TO=*SYSCMD
/MOD-SDF-OPT       MODE=&MODE -
/                  ,DEFAULT-PROG-NAME=&TESTPROGRAM -
/                  ,LOG=*INPUT-FORM
/SHOW-SDF-OPT      INF=*USER
/SHOW-FILE-ATTR   F-NAME=&(JV.INFILE)
/START-PERCON
//SHOW-SDF-OPT
//ASS-INPUT-F      FILE=*DISK-F(NAME=&(JV.INFILE))
//ASS-OUT-F        FILE=*DISK-F(NAME=&OUTFILE)
//SEL-INPUT-REC   COND= ((20,4)=NUMERIC AND (25,5)=ALPHA)
//START-CONV
//END
/SHOW-FILE-ATTR   F-NAME=&OUTFILE
/MOD-JOB-SW        ON=(1)
/START-PROG        FROM-FILE=$LMS
LIB FILE=USER.LIB,USAGE=BOTH
TOCX *
ADDX &OUTFILE>&ELEM
END
/SET-JOB-STEP
/MOD-JOB-SW        OFF=(1)
/MODIFY-SDF-OPT   MODE=*EXEC -
/                  ,DEFAULT-PROG-NAME=*NONE
/END-PROC

```

Die Prozedur PROC.SELECT wählt mithilfe des Dienstprogrammes PERCON Datensätze, die bestimmte Kriterien erfüllen (ein Feld darf nur Ziffern, ein anderes Feld darf keine Ziffern enthalten), aus der Datei TEST aus (Name ist der Wert der Jobvariablen JV.INFILE), und schreibt sie in die Datei, deren Name über den Prozedurparameter &OUTFILE angegeben

wird. Anschließend wird die Ergebnisdatei mit dem Dienstprogramm LMS in die Bibliothek USER.LIB übernommen. Der Name des Bibliothekelements wird über den Prozedurparameter &ELEM abgefragt (vorher werden die bereits existierenden Elemente angezeigt).

*Ablaufprotokoll:*

```

/c1p proc.select,proc-par=(mode=*test) _____ (1)
%/BEGIN-PROC      LOG=*ALL                ,PAR=*YES(
      PROC-PAR=(&MODE=*EXEC,&TESTPROGRAM=PERCON,&OUTFILE, &ELEM)
      ,ESC-CHAR=C'&'                      )
%/ASS-SYSDTA      TO=*SYSCMD              _____ (2)
%/MOD-SDF-OPT     MODE=*TEST              ,DEFAULT-PROG-NAME=PERCON
      ,LOG=*INPUT-FORM                    _____ (3)
%/SHOW-SDF-OPT   INFORMATION=*USER       _____ (4)
%  USER          : :20SG:$USER1.SYSSDF.USER.EXAMPLE.1
%                VERSION : UNDEFINED
%CURRENT SDF OPTIONS :
%  GUIDANCE       : *EXPERT
%  LOGGING        : *INPUT-FORM
%  CONTINUATION   : *NEW-MODE
%  UTILITY-INTERFACE : *NEW-MODE
%  PROCEDURE-DIALOGUE : *NO
%  MENU-LOGGING   : *NO
%  MODE           : *TEST
%  CHECK-PRIVILEGES : *YES
%  DEFAULT-PROGRAM-NAME : PERCON
%  FUNCTION-KEYS   : *STYLE-GUIDE-MODE
%  INPUT-HISTORY   : *ON
%  NUMBER-OF-INPUTS : 20
%  PASSWORD-PROTECTION: *YES
%/SHOW-FILE-ATTR F-NAME=TEST             _____ (5)
%/START-PERCON
%///SHOW-SDF-OPT _____ (6)
%///ASS-INPUT-F   FILE=*DISK-F(NAME=TEST) _____ (7)
%///ASS-OUT-F     FILE=*DISK-F(NAME=&OUTFILE)
%&OUTFILE=test4 _____ (8)
%///ASS-OUT-F     FILE=*DISK-F(NAME=TEST4) _____ (9)
%///SEL-INPUT-REC COND= ((20,4)=NUMERIC AND (25,5)=ALPHA)
%///START-CONV
%///END
%/SHOW-FILE-ATTR F-NAME=TEST4
%/MOD-JOB-SW      ON=(1)
%/START-PROG     FROM-FILE=$LMS
%  CMD0091 NEXT INPUT(S) IGNORED UNTIL MESSAGE CMD0092 _____ (10)
%LIB FILE=USER.LIB,USAGE=BOTH
%TOCX *
%ADDX &OUTFILE>&ELEM
%END

```

```

% CMD0092 END OF IGNORED INPUT OR INPUTS _____ (11)
%/SET-JOB-STEP
%/MOD-JOB-SW      OFF=(1)
%/MODIFY-SDF-OPT  MODE=*EXEC      ,DEFAULT-PROG-NAME=*NONE _____ (12)
%/END-PROC

```

- (1) Aufruf der Prozedur PROC.SELECT mit Angabe des Prozedurparameters MODE=\*TEST.
- (2) Der Systemdatei SYSDTA wird als Eingabequelle die Systemdatei SYSCMD zugewiesen, d.h. Datensätze werden ebenfalls aus der Prozedurdatei gelesen.
- (3) Die SDF-Einstellungen werden geändert: Die Angabe MODE=\*TEST bewirkt das Einschalten des Testmodus. Der Testmodus wird auf Anweisungen mit führenden „/“ erweitert. Die Protokollierung erfolgt wie eingegeben.
- (4) SHOW-SDF-OPTIONS wird ausgeführt und die Ausgabe der aktuellen SDF-Einstellungen.
- (5) Die Jobvariablenersetzung wurde durchgeführt. Für &(JV.INFILE) wurde der Inhalt der Jobvariablen JV.INFILE, „TEST“ eingesetzt.
- (6) Anweisungen mit führenden „/“ werden nur syntaktisch geprüft. (SHOW-SDF-OPTIONS ist hier eine Anweisung!)
- (7) Die Jobvariablenersetzung wurde durchgeführt (wie bei Punkt 5).
- (8) Der Prozedurparameter OUTFILE wurde bei Prozeduraufruf nicht angegeben. Deshalb wird er durch Prompting abgefragt.
- (9) Für &OUTFILE wird der angegebene Wert „TEST4“ gesetzt.
- (10) Die nachfolgenden Anweisungen an das Dienstprogramm LMS werden ignoriert, da sie nicht mit „/“ eingeleitet werden.  
Anweisungen mit führenden „/“ werden laut Vereinbarung nur anhand der für PERCON in der Syntaxdatei definierten Anweisungen syntaktisch geprüft. Werden weitere Programme mit SDF-Oberfläche verwendet, muss die Vereinbarung vorher entsprechend geändert werden.
- (11) Der letzte zu ignorierende Datensatz wird erkannt. Die nächste Eingabezeile beginnt mit „/“ bzw. „/“.
- (12) MODIFY-SDF-OPTIONS wird ausgeführt und bewirkt hier das Ausschalten des Testmodus.

## 4.11 Task-spezifische Default-Werte

Bei der Eingabe von Kommandos bzw. Anweisungen kann der Benutzer Operanden weglassen, deren Default-Werte den von ihm gewünschten Werten entsprechen. Für häufig benutzte Kommandos bzw. Anweisungen kann der Benutzer innerhalb seiner Dialog-Task Operandenwerte festlegen, die seinen Anforderungen entsprechen. Diese task-spezifischen Default-Werte werden bei interaktiver Eingabe an Stelle der in den Syntaxdateien festgelegten Default-Werte eingesetzt.

### Task-spezifische Default-Werte definieren

Task-spezifische Default-Werte können nur innerhalb einer Dialog-Task durch interaktive Eingabe oder in einer Prozedur definiert werden. Zur Definition wird das Kommando bzw. die Anweisung mit allen Operandenwerten, die als Default-Wert definiert werden sollen, eingegeben. Der Kommando- bzw. Anweisungsname muss mit einem Ausrufezeichen (!) beginnen. Die Eingabe wird nur syntaktisch geprüft und als eine Definition von task-spezifischen Default-Werten gespeichert. Das Fehlen von Pflichtoperanden wird ignoriert. Bei allen späteren Eingaben des Kommandos bzw. der Anweisung werden die task-spezifischen Default-Werte an Stelle der ursprünglichen Default-Werte verwendet.

Bei der Definition bestehen folgende Einschränkungen:

- Die Definition im Stapelbetrieb ist nicht möglich.
- Für einen geheimen Operanden kann kein task-spezifischer Default-Wert definiert werden; außer, es soll einer der folgenden Operandenwerte als task-spezifischer Default-Wert definiert werden:
  - ein mit OUTPUT=\*SECRET-PROMPT definierter Operandenwert (im Allgemeinen der Operandenwert \*SECRET)
  - ein mit SECRET=\*NO definierter Operandenwert.
- Für SDF-P-Kontrollflusskommandos (z.B. IF, FOR) und für ISP-Kommandos kann ebenfalls kein task-spezifischer Default-Wert definiert werden.
- Für ein Kommando bzw. eine Anweisung können maximal 10 Definitionen gespeichert werden.
- Insgesamt können maximal 100 Definitionen gespeichert werden.

## Definition im Dialog

Der Benutzer definiert im Kommandomodus mit der Eingabe `!<command> <operands>` task-spezifische Default-Werte für die mit `<operands>` angegebenen Operanden des Kommandos `<command>`. Analog dazu kann er im Programmmodus mit der Eingabe `!<statement> <operands>` task-spezifische Default-Werte für die Anweisung `<statement>` des geladenen Programms definieren.

Zur Definition der Default-Werte kann der Benutzer auch in den temporär geführten Dialog verzweigen. Hierzu muss der Kommando- bzw. Anweisungsname mit einem Fragezeichen enden. Es werden nur die tatsächlich geänderten Operandenwerte als task-spezifische Default-Werte gespeichert.

Abkürzungen sind wie bei der normalen Eingabe von Kommandos bzw. Anweisungen erlaubt.

## Definition in Prozeduren

Der Benutzer kann task-spezifische Default-Werte auch in Prozeduren definieren. Die Definition wirkt jedoch nur für die Dialogeingabe. Um unbeabsichtigte Seiteneffekte zu vermeiden wirken sich im Prozedurmodus nur die in den Syntaxdateien festgelegten Default-Werte aus.

Über Prozeduren kann der Benutzer die Definition task-spezifischer Default-Werte automatisieren bzw. vereinfachen. Bei Definition in einer LOGON-Prozedur sind die definierten Default-Werte bereits bei Beginn der Dialogtask verfügbar. Der Benutzer kann für verschiedene Aufgabenstellungen Prozeduren vorbereiten, die bei Bedarf aufgerufen werden können und die gewünschte Menge von Default-Werten definieren.

Die Definition für ein Kommando beginnt mit „/“, die für eine Anweisung mit „//“. Die Definition erfolgt sonst wie bei der Dialogeingabe. Vor einem Kommandonamen kann eine Marke stehen. Eine Definition innerhalb des Prozedurkopfes ist nicht erlaubt.

Die Definition für eine Anweisung kann auch im Kommandomodus erfolgen, d.h. das Programm muss nicht geladen sein. Zur Syntaxanalyse wird das Programm verwendet, das mit der SDF-Option DEFAULT-PROGRAM-NAME eingetragen ist (siehe Kommando MODIFY-SDF-OPTIONS). Die Definition wird auch zur Nutzung mit diesem Programm gespeichert.

Syntaxfehler bei der Definition von Default-Werten lösen die Fehlerbehandlung aus.

## Definition an der Programmschnittstelle

Die Makroaufrufe CMDTST und TRCMD können die Definition von task-spezifischen Default-Werten zulassen. Bei allen anderen SDF-Makroaufrufen wird die Definition von Default-Werten als Syntaxfehler abgewiesen.

## Auswertung der Definitionen

Bei der Eingabe eines Kommandos bzw. einer Anweisung prüft SDF, ob für dieses Kommando bzw. für diese Anweisung task-spezifische Default-Werte gespeichert sind und setzt ggf. die gespeicherten Default-Werte in die eingegebene Zeichenfolge ein. Danach erfolgt die Syntaxanalyse der Eingabe.

Durch den Wechsel von Syntaxdateien (z.B. Versionswechsel eines Subsystems) können bereits gespeicherte Default-Werte der aktuellen Syntax widersprechen und zum Zurückweisen der Eingabe führen. In diesem Fall muss der Benutzer die fehlerhafte Definition löschen.

Werden bei der Definition untergeordnete Operanden außerhalb der Struktur angegeben (STRUCTURE-IMPLICIT-Schreibweise), so wird der struktureinleitende Operandenwert nicht automatisch zum task-spezifischen Default-Wert.

Beispielsweise wird durch die Definition `!CREATE-FILE ACCESS=*READ` der übergeordnete Operand `PROTECTION=*PARAMETERS` nicht ebenfalls zum Default-Wert.

Für ein Kommando bzw. eine Anweisung können mehrere Definitionen gespeichert werden. Es werden alle für das Kommando bzw. die Anweisung gespeicherten Definitionen in der Reihenfolge, in der sie definiert wurden, ausgewertet.

Für ein und denselben Operanden wird jeweils der zuletzt definierte Default-Wert eingesetzt.

Um der Definition von Operanden einer Strukturstufe einen neuen Default-Wert hinzuzufügen bzw. einen einzelnen Default-Wert zu ändern, genügt es, eine neue Definition mit diesem Default-Wert abzuspeichern.

## Task-spezifische Default-Werte verwalten

Mit dem Kommando `SHOW-INPUT-DEFAULTS` kann sich der Benutzer über alle aktuell definierten task-spezifischen Default-Werte informieren. Die Ausgabe erfolgt nach `SYSOUT` oder `SYSLST` und kann auf bestimmte Kommandos, Anweisungen oder Programme beschränkt werden. Die Definitionen können mit einer Eingabeseriennummer ausgegeben werden, die ein gezieltes Löschen von einzelnen Definitionen erlaubt. Auf Programmebene ist die gleichnamige Standardanweisung mit derselben Funktionalität verfügbar.

Mit dem Kommando `RESET-INPUT-DEFAULTS` kann der Benutzer task-spezifische Default-Werte löschen. Ohne Angabe von Operanden werden alle für Kommandos definierten Default-Werte gelöscht. Darüberhinaus kann der Benutzer auch die Definitionen ausgewählter Kommandos bzw. Anweisungen löschen. Über die Eingabeseriennummer kann auch eine einzelne Definition gezielt gelöscht werden.



## Beispiele

### Beispiel 1

```

/sdf-a
% BLS0517 MODULE 'SDAMAIN' LOADED (1)
% SDA0001 'SDF-A' VERSION '04.1E10' STARTED (1)
%//open mode=? _____ (1)
% CMD0090 EXPLANATION OF OPERAND 'MODE':
*UPDATE() or *CREATE() or *READ or *INIT() -DEFAULT-: *UPDATE
%//!open mode=*read _____ (2)
%//open mode=? _____ (3)
% CMD0090 EXPLANATION OF OPERAND 'MODE':
*UPDATE() or *CREATE() or *READ or *INIT() -DEFAULT-: *READ
%//

```

- (1) Das Dienstprogramm SDF-A wird gestartet. Anschließend werden die Operandenwerte des Operanden MODE der Anweisung OPEN-SYNTAX-FILE abgefragt. Als Default-Wert wird \*UPDATE angezeigt.
- (2) Der Default-Wert wird task-spezifisch auf \*READ geändert.
- (3) Bei der erneuten Abfrage der Operandenwerte wird jetzt \*READ als Default-Wert angezeigt.

### Beispiel 2

```

/show-file-attr creation-date=*today _____ (1)
%      84 :20SG:$USER1.DATEI.3
%      3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      2 FILES RES=      87 FRE=      21 REL=      18 PAGES
/reset-input-defaults *all _____ (2)
/call-proc proc.default-1,log=*yes _____ (3)
%      1 1 /CMD-DEF-1:
%      1 1 / !SHOW-FILE-ATTR INF=*MIN
%      2 1 /!CRE-FILE      SUP=*PRIV(VOL=WORK01,DEV-TYPE=D3490-30)
%      3 1 /!CRE-FILE      SUP=*PUBLIC
%      4 1 /CMD-DEF-2:
%      4 1 / !PRINT-DOC      LINE-SPACING=*BY-EBCDIC-CONTR
%      5 1 /PROG-DEF-1:
%      5 1 / MOD-SDF-OPT      DEFAULT-PROG=SDF-A
%//!OPEN-SYNTAX-FILE MODE=*READ
%//!SHOW      OBJ=*CMD(NAME=*ALL),ATT-INFO=*NO
%      8 1 /PROG-DEF-2:
%      8 1 / MOD-SDF-OPT      DEFAULT-PROG=SORT

```

```

%//!SORT-RECORDS      KEEP-EQUAL-SEQUENCES=*YES
%          10 1 /END:
%          10 1 / IF-BLOCK-ERROR
%          12 1 /end-if
%          13 1 /MOD-SDF-OPT      DEFAULT-PROG=*NONE
%          14 1 /SHOW-INPUT-DEFAULTS *CMD

/!SHOW-FILE-ATTR INFORMATION=*MINIMUM
/!CRE-FILE SUPPORT=*PRIVATE-DISK(VOLUME=WORK01,DEVICE-TYPE=D3490-30)
/!CRE-FILE SUPPORT=*PUBLIC-DISK
/!PRINT-DOC LINE-SPACING=*BY-EBCDIC-CONTROL
%          15 1 /WRITE-TEXT 'SDF-A Default-Werte:'
SDF-A Default-Werte:
%          16 1 /SHOW-INPUT-DEFAULTS *STMT(PROG=SDF-A)
//!OPEN-SYNTAX-FILE MODE=*READ
//!SHOW OBJECT=*COMMAND(NAME=*ALL),ATTACHED-INFORMATION=*NO
//!OPEN-SYNTAX-FILE MODE=*READ
//!SHOW OBJECT=*COMMAND(NAME=*ALL),ATTACHED-INFORMATION=*NO
%          17 1 /WRITE-TEXT 'SORT Default-Werte:'
SORT Default-Werte:
%          18 1 /SHOW-INPUT-DEFAULTS *STMT(PROG=SORT)
//!SORT-RECORDS KEEP-EQUAL-SEQUENCES=*YES
//!SORT-RECORDS KEEP-EQUAL-SEQUENCES=*YES
%          1  /EXIT-PROCEDURE ERROR=*NO

```

**/shid \*all, input-serial-number=\*yes** \_\_\_\_\_ (4)

```

/" 8 : " !SHOW-FILE-ATTR INFORMATION=*MINIMUM
/" 9 : " !CRE-FILE SUPPORT=*PRIVATE-DISK(VOLUME=WORK01,DEVICE-TYPE=D3490-30)
/" 10 : " !CRE-FILE SUPPORT=*PUBLIC-DISK
/" 11 : " !PRINT-DOC LINE-SPACING=*BY-EBCDIC-CONTROL
// " 12 : " !OPEN-SYNTAX-FILE MODE=*READ
// " 13 : " !SHOW OBJECT=*COMMAND(NAME=*ALL),ATTACHED-INFORMATION=*NO
// " 14 : " !SORT-RECORDS KEEP-EQUAL-SEQUENCES=*YES
/show-file-attr creation-date=*today _____ (5)
%S NNN NW          84 :20SG:$USER1.DATEI.3
%S NNN NW          3  :20SG:$USER1.PROC.JV

```

- (1) Mit **SHOW-FILE-ATTRIBUTES** werden alle zum aktuellen Tagesdatum erstellten Dateien angezeigt. Die Ausgabe erfolgt implizit mit **INFORMATION=\*NAME-AND-SPACE** (Default-Wert des Kommandos).
- (2) Mit **RESET-INPUT-DEFAULTS** werden alle bisher gesetzten task-spezifischen Default-Werte zurückgesetzt.

- (3) Aufruf der Prozedur PROC.DEFAULT-1. In der Prozedur werden task-spezifische Default-Werte für Kommandos und für Anweisungen der Programme SDF-A und SORT gesetzt (vgl. Prozedurablauf). Die Prozedur endet mit der Ausgabe der task-spezifischen Default-Werte.  
Dabei ist zu beachten:  
Die Programme werden nicht aufgerufen. Die Default-Werte der Anweisungen werden im Kommandomodus gesetzt. Vor dem Setzen der Default-Werte wird mit dem Kommando MODIFY-SDF-OPTION das entsprechende Programm eingestellt (Operand DEFAULT-PROGRAM-NAME).
- (4) Anschließend werden mit SHOW-INPUT-DEFAULTS (Kurzname SHID) noch einmal alle task-spezifischen Default-Werte mit der Eingabeseriennummer ausgegeben.
- (5) Bei der erneuten Eingabe des Kommandos SHOW-FILE-ATTRIBUTES werden alle zum aktuellen Tagesdatum erstellten Dateien angezeigt. Die Ausgabe erfolgt jetzt mit dem task-spezifischen Default-Wert INFORMATION=\*MINIMUM (vgl. Default-Wert mit der Eingabeseriennummer 8).



---

## 5 Beispiele für die Kommandoeingabe im Dialogbetrieb

Die nachfolgende Beispielsitzung zeigt die Eingabe in allen Formen des ungeführten und geführten Dialogs. Unter anderem wird gezeigt,

- welche Führung SDF bei seiner Eingabe in den drei Stufen des geführten Dialogs bietet,
- wie der Benutzer die Eingabe komprimieren kann,
- welche Unterstützung SDF bei der Korrektur fehlerhafter Eingaben bietet,
- wie der Benutzer in den temporär geführten Dialog wechseln kann,
- wie der Benutzer die Dialogführung wechselt,
- wie der Benutzer Informationen über die Eingabe mit SDF abfragt.

Die Beispielsitzung wurde an einem Rechner mit BS2000/OSD-BC V5.0A durchgeführt. Bei Systemgenerierung wurde mit dem Sprachschlüssel D die deutsche Sprache eingestellt. Im Beispiel meldet sich der Benutzer unter der Benutzerkennung **USER1** im System an. Für die Benutzerkennung USER1 ist im Katalogeintrag ebenfalls der Sprachschlüssel D eingetragen. Der SDF-Dialog und die Meldungsausgabe erfolgen deshalb in deutscher Sprache (zur Spracheinstellung siehe „Sprache für Ausgaben von SDF“ [Seite 46](#)). Zur Demonstration der Eingabemöglichkeiten wurde das häufig verwendete Benutzer-Kommando SHOW-FILE-ATTRIBUTES verwendet. Eine vollständige Beschreibung des Kommandos enthält das Handbuch „Kommandos, Band 3“ [1]. Unter der Benutzerkennung USER1 sind vier Dateien katalogisiert:

- die ISAM-Dateien TEST.BEISPIEL.1 und TEST.BEISPIEL.2
- die SAM-Dateien TEST.BEISPIEL.3 und TEST.BEISPIEL.4

Wollen Sie sich mit den Eingabemöglichkeiten von SDF vertraut machen, können Sie die Beispielsitzung nachvollziehen. Dabei müssen Sie die LOGON-Angaben (z.B. Benutzerkennung) und die Dateinamen Ihren Gegebenheiten anpassen.

Nachfolgend ist die Funktionstastenbelegung des Style-Guide-Modus eingestellt. Nach der NEXT-Zeile werden abhängig vom Führungsmodus die möglichen Eingaben angezeigt, wobei die möglichen Funktionen und Funktionstasten in einer eigener Zeile (KEYS:) angezeigt werden. Bei eingestelltem Old-Modus sind statt der genannten Funktionstasten die entsprechenden Tasten des Old-Modus zu verwenden. Existiert im Old-Modus keine entspre-

chende Funktionstaste, muss die entsprechende Steueranweisung bzw. das entsprechende Kommando in der NEXT-Zeile eingegeben werden (siehe [Abschnitt „Funktionstasten“ auf Seite 65](#)).

Wenn Sie die Beispielsitzung abbrechen wollen, können Sie den geführten Dialog jederzeit wieder beenden. Geben Sie dazu in der NEXT-Zeile das Kommando MODIFY-SDF-OPTIONS GUIDANCE=\*EXPERT (bzw. \*NO) ein.

## 5.1 HELP-SDF

Nach der Verarbeitung des Kommandos LOGON ist die Benutzerführung auf GUIDANCE=\*EXPERT (EXPERT-Form) voreingestellt. Nach Verbindungsaufbau fordert das System zum LOGON auf:

```
% JMS0150 ANLAGE ' S150-40', BS2000 VERSION 'V140', RECHNER 'D016ZE04':
BITTE '/SET-LOGON-PARAMETERS' ODER '?' EINGEBEN
/?
```

Die LOGON-Aufforderung wird mit „?“ beantwortet. Für das Kommando SET-LOGON-PARAMETERS wechselt SDF in einen temporär geführten Dialog:

```
KOMMANDO : SET-LOGON-PARAMETERS

-----
USER-IDENTIFICATION = user1
ACCOUNT             = acc0001
PASSWORD            = c'auf54ht$' → Eingabe nicht sichtbar
JOB-CLASS           = *STD
JOB-NAME            = testsdf
MONJV               = *NONE
JV-PASSWORD        =
SCHEDULING-TIME    = *STD
LIMIT               = *STD
RESOURCES           = *PARAMETERS(RUN-PRIORITY=*STD,CPU-LIMIT=*STD,SYSLST-LIMIT
                    =*STD,SYSOPT-LIMIT=*STD)
LOGGING             = *PARAMETERS(LISTING=*NO,HARDCOPY=*NO)
JOB-PARAMETER      = *NO

-----
NEXT = +
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN
      F11=*EXECUTE F12=*CANCEL

MESSAGE: CMD0175 ANDERE ANGABEN GEWUENSCHT? *EXIT-TASTE BETAETIGEN
```

Bild 7: Kommando SET-LOGON-PARAMETERS

Der Benutzer gibt seine Benutzerkennung (USER1), die Abrechnungsnummer (ACC0001), das Kennwort (C'AUF54HT\$', nicht sichtbar, da das Eingabefeld für den „geheimen“ Operanden PASSWORD dunkelgesteuert ist) und den Auftragsnamen (TESTSDF) an. Da die

NEXT-Zeile bereits mit „+“ vorbesetzt ist, kann nach Absenden des Bildschirms (**[DUE]**-Taste) die nächste Seite des Operandenfragebogens ausgegeben werden. Da in diesem Beispiel keine weiteren Angaben erforderlich sind, wird das Kommando sofort abgeschickt (mit der **[F11]**-Taste bzw. mit Eingabe von \*EXECUTE in der NEXT-Zeile und **[DUE]**-Taste). Nach Verarbeitung des Kommandos SET-LOGON-PARAMETERS ist die Benutzerführung auf GUIDANCE=\*EXPERT (EXPERT-Form) eingestellt. Das Kommando HELP-SDF ohne Angabe von Operanden gibt allgemeine Informationen über SDF aus. Die nochmalige Eingabe des Kommandos mit einem Fragezeichen bewirkt den Wechsel in den temporär geführten Dialog.

```
% JMS0066 JOB 'SDFTEST' ACCEPTED ON 01-10-04 AT 17:14, TSN = 1PY8
/help-sdf
%
% Einfuehrung:
%
% SDF ist ein komfortabler Kommando-Interpreter und Dialog-
% Manager. Sie koennen folgende Vorteile von SDF nutzen:
%
% - gefuehrter Dialog
% - Abkuerzbarkeit
% - Blockeingabe
% - Kommandogedaechtnis (History)
% - Definieren task-spezifischer Defaultwerte
%
% Ausserdem wurde eine verstaendlichere Kommando-Sprache
% entwickelt:
% Namen von Kommandos, Operanden und Schluesselwortwerten
% sind so gewaehlt, dass sie die Funktion oder den Zweck
% erkennen lassen. Analoge Sachverhalte (z.B. der Dateiname)
% sind gleich benannt(z.B. FILE-NAME=, FROM-FILE= oder TO-FILE=).
% Bei variablen Operandenwerten beschreiben Grunddatentypen
% den zulaessigen Wertebereich.
% Fehlerhafte Eingaben werden mit einer Meldung zur Korrektur
% angeboten.
% Anwender der alten Kommandosprache ISP erhalten mit der
% Eingabe "<altes-Kdo>?" den Operandenfragebogen in der neuen
% Syntax bzw. ein Auswahlmeneue der entsprechenden SDF-Kommandos.
%
% Fuer weitere Informationen geben Sie bitte "HELP-SDF?" ein.
% Sie erhalten dann den Operandenfragebogen dieses Kdos. Bei den
% Operanden, die Sie auf "YES" setzen, erhalten Sie weitere
% Informationen.
% Im Handbuch "Master Index" finden Sie Verweise auf Handbuecher,
% die die Beschreibung der einzelnen Kommandos enthalten.
%
```

```
/help-sdf?
```

Nach der Eingabe „help-sdf?“ gibt SDF den Operandenfragebogen für das Kommando HELP-SDF aus.

```
KOMMANDO   : HELP-SDF

-----
GUIDANCE-MODE      = *NO
SDF-COMMANDS      = *NO
ABBREVIATION-RULES = *NO
GUIDED-DIALOG     = *YES(SCREEN-STEPS=*NO,SPECIAL-FUNCTIONS=*ye,FUNCTION-KEYS
                    =*NO,NEXT-FIELD=*NO)
UNGUIDED-DIALOG   = *YES(SPECIAL-FUNCTIONS=*NO,FUNCTION-KEYS=*NO)

-----
NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8+=   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL
```

Bild 8: Operandenfragebogen für das Kommando HELP-SDF

Bei dem Operanden GUIDED-DIALOG wird in der Struktur \*YES für den Operanden SPECIAL-FUNCTIONS die Vorbesetzung \*NO (Default-Wert) in „\*ye“ geändert (entspricht \*YES). Damit werden Informationen über die Funktionstasten angefordert. Über die Angabe in der NEXT-Zeile wird gesteuert, was das Absenden des Bildschirms (DUE-Taste) bewirkt. Die Vorbesetzung \*CONTINUE steht für \*EXECUTE oder + und bewirkt in diesem Fall die Kommandoausführung. Folgender Text wird ausgegeben:

```
%
% Gefuehrter Dialog : Spezialfunktionen
%
% ? als Operandenwert:
%   Liefert Hilfetext und Informationen ueber moegliche
%   Operandenwerte (Wertebereiche, usw.); bei fehlerhafter
%   Eingabe zusaetzlich detaillierte Fehlermeldungen.
% ?? als Operandenwert:
%   Liefert Hilfetexte, eine Uebersicht der moeglichen
%   Operandenwerte und einen Hilfetext mit Informationen
%   ueber Datentypen, die dem Operanden zugeordnet sind.
% ! als Operandenwert:
%   Setzt fuer diesen Operanden den Defaultwert ein.
```



```
% ( hinter einem struktureinleitenden Operandenwert:  
%     Gibt den Unterfragebogen fuer die Struktur aus, die  
%     diesem Operandenwert zugeordnet ist.  
% () hinter einem struktureinleitenden Operandenwert:  
%     Unterdrueckt den Unterfragebogen und setzt fuer die  
%     Operanden der Struktur die jeweiligen Defaultwerte ein.  
% - als letztes Zeichen in einer Eingabezeile:  
%     Eine Fortsetzungszeile wird ausgegeben.  
% LZF-Taste:  
%     Loescht ab der Schreibmarke alle Zeichen der Eingabezeile.  
%
```

## 5.2 Ungeführter Dialog

### 5.2.1 EXPERT-Form

Nachdem SDF die Informationen über die Funktionstasten ausgegeben hat, wird der Dialog folgendermaßen fortgesetzt:

```

/sh-f-attr test.beispiel.,file-struct=i _____ (1)
%      33 :20SG:$USER1.TEST.BEISPIEL.1 _____ (2)
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%:20SG: PUBLIC:      2 FILES RES=      105 FRE=      26 REL=      9 PAGES
/sh-f-attr test,beispiel.,file-struct=(i,s) _____ (3)
% CMD0051 OPERAND 'INFORMATION' FEHLERHAFT
% CMD0064 OPERANDEN-WERT 'BEISPIEL.' MIT DATEN-TYP '*NAME-AND-SPACE ODER
*SPACE-SUMMARY ODER *ALL-ATTRIBUTES ODER *PARAMETERS() ODER *STATISTICS ODER
*MINIMUM' UNVEREINBAR _____ (4)

/sh-f-attr inf=? _____ (5)
% CMD0090 ERLAEUTERUNG ZU OPERAND 'INFORMATION':
*NAME-AND-SPACE oder *SPACE-SUMMARY oder *ALL-ATTRIBUTES oder *PARAMETERS()
oder *STATISTICS oder *MINIMUM -Default-: *NAME-AND-SPACE

/sh-f-attr test.beispiel.,file-struct=(i,s) _____ (6)
%      33 :20SG:$USER1.TEST.BEISPIEL.1 _____ (7)
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%      48 :20SG:$USER1.TEST.BEISPIEL.3
%      3  :20SG:$USER1.TEST.BEISPIEL.4
%:20SG: PUBLIC:      4 FILES RES=      156 FRE=      30 REL=      9 PAGES
/mod-sdf-opt guid=n _____ (8)
%KDO:

```

- (1) Mit dem Kommando SHOW-FILE-ATTRIBUTES werden über die ISAM-Dateien („file-struct=i“), deren Name mit „test.beispiel.“ beginnt, Standardinformationen (Größe und Name) angefordert.
- (2) Das System gibt die gewünschten Informationen aus.
- (3) Mit dem Kommando SHOW-FILE-ATTRIBUTES werden über die ISAM- und SAM-Dateien („file-struct=(i,s)“), deren Name mit „test.beispiel.“ beginnt, die Standardinformationen (Größe und Name) angefordert. Dem Operanden FILE-STRUCTURE werden zwei Werte zugewiesen „(i,s)“, die in Klammern eingeschlossen sind (Listeneingabe). Der Dateiname „test,beispiel.“ ist fehlerhaft (Komma statt Punkt).
- (4) SDF interpretiert „beispiel.“ als Wert für den Operanden INFORMATION (Eingabe als zweiter Stellungsoperand) und weist ihn als fehlerhaft zurück.

- (5) SDF führt in der EXPERT-Form keinen Korrekturdialog sondern fordert zur Kommandoeingabe auf. Eine Erläuterung des Operanden INFORMATION wird angefordert. Das erreicht der Benutzer durch Eingabe eines Fragezeichens an Stelle des Operandenwerts. Das hinter dem Wert \*PARAMETERS stehende Klammerpaar weist darauf hin, dass \*PARAMETERS eine Struktur einleitet.
- (6) Das in Arbeitsschritt (3) eingegebene fehlerhafte Kommando wird fehlerfrei eingegeben.
- (7) Das System gibt die gewünschten Informationen aus.
- (8) Mit dem Kommando MODIFY-SDF-OPTIONS wird in die NO-Form des geführten Dialogs gewechselt. Alle Operanden dieses Kommandos sind wahlweise und haben den Default-Wert \*UNCHANGED. Das hat zur Folge, dass nur die explizit angegebenen SDF-Optionen, in diesem Fall die Benutzerführung, geändert werden.

## 5.2.2 NO-Form

Der Dialog wird folgendermaßen fortgesetzt:

```
%KDO:sh-f-a test.,file-struct=i _____ (1)
% CMD0187 DIE ABKUERZUNG 'SH-F-A' IST MEHRDEUTIG BEZUEGLICH DER KOMMANDOS
'SHOW-FILE-ATTRIBUTES,SHOW-FT-ADMISSION-SET'
%BITTE ANWEISUNGSNAMEN KORRIGIEREN:sh-f-attr _____ (2)
%      33 :20SG:$USER1.TEST.BEISPIEL.1
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%:20SG: PUBLIC:      2 FILES RES=      105 FRE=      26 REL=      9 PAGES
%KDO:sh-f-attr test,beispiel.,file-struct=(i,s) _____ (3)
% CMD0051 OPERAND 'INFORMATION' FEHLERHAFT _____ (4)
% CMD0064 OPERANDEN-WERT 'BEISPIEL.' MIT DATEN-TYP '*NAME-AND-SPACE ODER
*SPACE-SUMMARY ODER *ALL-ATTRIBUTES ODER *PARAMETERS() ODER *STATISTICS ODER
*MINIMUM' UNVEREINBAR
%BITTE OPERANDEN EINGEBEN: _____ (5)
%test,beispiel.,file-struct=(i,s) _____ (6)
test.beispiel.,inf=name _____ (7)
%      33 :20SG:$USER1.TEST.BEISPIEL.1 _____ (8)
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%      48 :20SG:$USER1.TEST.BEISPIEL.3
%      3  :20SG:$USER1.TEST.BEISPIEL.4
%:20SG: PUBLIC:      4 FILES RES=      156 FRE=      30 REL=      9 PAGES
%KDO:
```

- (1) Mit dem Kommando SHOW-FILE-ATTRIBUTES werden über die ISAM-Dateien („file-struct“), deren Name mit „test.beispiel.“ beginnt, die Standardinformationen (Größe und Name) angefordert. Das Kommando wird mit „sh-f-a“ abgekürzt.
- (2) SDF fordert zur Korrektur des Kommandonamens auf, da die angegebene Abkürzung nicht eindeutig war. Nach Eingabe der eindeutigen Abkürzung „sh-f-attr“ wird das Kommando SHOW-FILE-ATTRIBUTES mit den vorher angegebenen Auswahlkriterien ausgeführt.
- (3) Das System gibt die gewünschten Informationen aus.
- (4) Mit dem Kommando SHOW-FILE-ATTRIBUTES werden über die ISAM- und SAM-Dateien („file-struct=(i,s)“), deren Name mit „test,beispiel.“ beginnt, die Standardinformationen (Größe und Name) angefordert. Dem Operanden FILE-STRUCTURE werden zwei Werte zugewiesen „(i,s)“, die in Klammern eingeschlossen sind (Listeneingabe). Der Dateiname „test,beispiel.“ ist fehlerhaft (Komma statt Punkt).
- (5) SDF interpretiert „beispiel.“ als Wert für den Operanden INFORMATION (Eingabe als zweiter Stellungsoperand) und weist ihn als fehlerhaft zurück. Das hinter dem Wert \*PARAMETERS stehende Klammernpaar weist darauf hin, dass \*PARAMETERS eine Struktur einleitet.

- (6) SDF fordert zur Korrektur der Operanden auf.
- (7) SDF bietet die eingegebenen Operanden zur Korrektur an.
- (8) Die Operanden werden korrigiert. Es ist nicht nur der Dateiname zu korrigieren sondern auch der Operand INFORMATION. Für die Korrektur kann der Benutzer auch temporär in den geführten Dialog wechseln.
- (9) Das System gibt die gewünschten Informationen aus.

## 5.3 Temporär geführter Dialog

Aus dem ungeführten Dialog kann für die Eingabe eines Kommandos (bzw. einer Anweisung) temporär in den geführten Dialog gewechselt werden. Das wird mit dem hinter dem Kommandonamen stehenden Fragezeichen erreicht.

```
%KDO:sh-f-attr? test.beispiel.
```

Nach der Eingabe von „sh-f-attr? test.beispiel.“ gibt SDF den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES aus. Der Fragebogen ist vorbesetzt mit dem für den Operanden FILE-NAME angegebenen Dateinamen und den Default-Werten der wahlweisen Operanden INFORMATION, SELECT und OUTPUT.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.

-----

FILE-NAME      = TEST.BEISPIEL.
INFORMATION    = *NAME-AND-SPACE
SELECT         = ?ALL
OUTPUT         = *SYSOUT
OUTPUT-OPTIONS = *PARAMETERS(SORT-LIST=*BY-FILENAME)

-----

NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8+=   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

Bild 9: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Über den Operanden SELECT werden mit „?“ weitere Informationen angefordert. Der Zeilenrest „ALL“ darf in diesem Fall stehen bleiben. Die NEXT-Zeile ist mit \*CONTINUE vorbesetzt (steht für \*EXECUTE oder +) und bewirkt in diesem Fall, dass SDF den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES mit einer Erläuterung des Operanden SELECT ausgibt.

SDF gibt weitere Informationen über den Operanden SELECT aus. Das hinter dem Wert \*BY-ATTRIBUTES stehende Klammernpaar weist darauf hin, dass \*BY-ATTRIBUTES eine Struktur einleitet.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*ALL

-----

FILE-NAME      = TEST.BEISPIEL.
INFORMATION    = *NAME-AND-SPACE
SELECT        = *by
               *ALL oder *BY-ATTRIBUTES()
               Kriterien fuer die Dateiauswahl (ALL: alle Dateien;
               BY-ATTRIBUTES: Dateien mit den angegebenen Merkmalen)
OUTPUT        = *SYSOUT
OUTPUT-OPTIONS = *PARAMETERS(SORT-LIST=*BY-FILENAME)

-----

NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8==   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

Bild 10: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Der für den Operanden SELECT vorgegebene Default-Wert \*ALL wird mit „\*by“ (\*BY-ATTRIBUTES) überschrieben. Das Absenden des Fragebogens bewirkt die Ausgabe eines Unterfragebogens für die Struktur \*BY-ATTRIBUTES.

SDF gibt den Unterfragebogen für die Struktur \*BY-ATTRIBUTES aus.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES

-----

CREATION-DATE      = *ANY
EXPIRATION-DATE    = *ANY
LAST-ACCESS-DATE   = *ANY
LAST-CHANGE-DATE   = *ANY
SUPPORT            = *ANY
VOLUME             = *ANY
SIZE               = ??NY
NUMBER-OF-EXTENTS  = *ANY
NUMBER-OF-FREE-PAGES = *ANY
HIGHEST-USED-PAGE  = *ANY
BLOCK-COUNTER      = *ANY
ACCESS             = *ANY
PASSWORD           = *ANY

-----

NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8==
      F9=REST-SDF-IN  F11=*EXECUTE  F12=*CANCEL

```

Bild 11: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Mit „??“ werden weitere Informationen über den Operanden SIZE angefordert. Der Zeilenrest „NY“ darf in diesem Fall stehen bleiben. Das Absenden des Fragebogens bewirkt, dass SDF weitere Informationen über den Operanden SIZE ausgibt. Zusätzlich zu möglichen Operandenwerten und dem Hilfetext (bei Anforderung mit „?“) werden auch Informationen über Datentypen ausgegeben, die als Eingabe zugelassen sind.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES(SIZE=*ANY)
-----
SIZE              = 33
                  *ANY oder *FREESIZE oder Ganzzahl_0..2147483647 oder
                  *INTERVAL(FROM=0,TO=2147483647)
                  Anzahl Seiten. Ausgewählt werden alle Dateien mit der
                  angegebenen Speicherbelegung
-----
                  <Ganzzahl> : Folge von Ziffern (0..9), denen ein
                  Vorzeichen (+ oder -) vorangehen kann
NUMBER-OF-EXTENTS = *ANY
NUMBER-OF-FREE-PAGES = *ANY
HIGHEST-USED-PAGE = *ANY
BLOCK-COUNTER     = *ANY
ACCESS            = *ANY
-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE   F12=*CANCEL

```

Bild 12: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Der für den Operanden SIZE vorgegebene Operandenwert ANY wird mit „33“ überschrieben (Auswahl von Dateien, die genau 33 PAM-Seiten belegen). Da die NEXT-Zeile bereits mit „+“ vorbesetzt ist, wird nach Absenden des Bildschirms (DUE-Taste) die nächste Seite des Operandenfragebogens ausgegeben.



```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES(SIZE=33)
-----
PASSWORD           = *ANY
USER-ACCESS        = *ANY
BASIC-ACL           = *ANY
ACL                 = *ANY
GUARDS              = *ANY
PROTECTION-ACTIVE  = *ANY
STATUS              = *ANY
FILE-STRUCTURE     = ?ANY
BLOCK-CONTROL-INFO = *ANY
BACKUP-CLASS       = *ANY
MIGRATE             = *ANY
STORAGE-LEVEL      = *ANY
GENERATIONS         = *NO
-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE  F12=*CANCEL

```

Bild 13: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Mit „?“ werden weitere Informationen über den Operanden FILE-STRUCTURE angefordert. Der Zeilenrest „ANY“ darf in diesem Fall stehen bleiben. Das Absenden des Fragebogens bewirkt, dass SDF weitere Informationen über den Operanden FILE-STRUCTURE ausgibt.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES(SIZE=33,FILE-STRUCTURE=*ANY)
-----
FILE-STRUCTURE     = iANY
                   *ANY oder -Liste erlaubt (5)-: *PAM oder *SAM oder
                   *ISAM oder *BTAM oder *NONE
                   Dateistruktur. Ausgewählt werden alle Dateien mit der
                   angegebenen Dateistruktur
BLOCK-CONTROL-INFO = *ANY
BACKUP-CLASS       = *ANY
MIGRATE             = *ANY
STORAGE-LEVEL      = *ANY
GENERATIONS         = *NO
TYPE-OF-FILES      = *ANY
FROM-CATALOG       = *STD
-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE  F12=*CANCEL

```

Bild 14: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Als Zugriffsmethode wird ISAM ausgewählt. Die Angabe „i“ ist eine eindeutige Abkürzung für \*ISAM. Der führende Stern kann entfallen, da als alternative Operandenwerte nur Schlüsselwörter zulässig sind. Die Schreibweise ohne Stern ist nicht langfristig garantiert und sollte deshalb ausschließlich im Dialog verwendet werden. Fälschlicherweise wird der Zeilenrest „ANY“ nicht gelöscht. Der Fragebogen wird abgesandt.

SDF interpretiert auch den verbliebenen Zeilenrest „ANY“ als Teil des Operandenwerts für FILE-STRUCTURE. SDF lehnt den Wert „iANY“ als fehlerhaft ab und fordert zur Korrektur auf (der fehlerhafte Operand ist hervorgehoben; im Beispiel unterstrichen).

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES(SIZE=33,FILE-STRUCTURE=iANY)
-----
FILE-STRUCTURE = i
BLOCK-CONTROL-INFO = *ANY
BACKUP-CLASS     = *ANY
MIGRATE          = *ANY
STORAGE-LEVEL    = *ANY
GENERATIONS      = *NO
TYPE-OF-FILES    = *ANY
FROM-CATALOG     = *STD
IO-ATTRIBUTES    = *ANY
DISK-WRITE       = *ANY
FREE-FOR-DELETION = *ANY
-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE  F12=*CANCEL
ERROR: CMD0081 WERT 'IANY' NICHT IN SCHLUESSELWORT-LISTE DES WERTE-BEREICHS
      '*ANY ODER -LISTE ERLAUBT (5)-: *PAM ODER *SAM ODER *ISAM ODER *BTAM
      ODER *NONE' ENTHALTEN

```

Bild 15: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Der Operandenwert für FILE-STRUCTURE wird durch Löschen der Zeichen „ANY“ korrigiert. Nach Absenden des Fragebogens wird der Fragebogen weitergeblättert (Vorbelegung „+“ in der NEXT-Zeile):

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES(SIZE=33,FILE-STRUCTU
              RE=*ISAM)
-----
STORAGE-CLASS      = *ANY
MANAGEMENT-CLASS   = *ANY
ADM-INFORMATION    = *ANY
USER-INFORMATION   = *ANY
VOLUME-SET         = *ANY
AVAILABILITY       = *ANY
SO-MIGRATION       = *ANY
WORK-FILE          = *ANY
FILE-PREFORMAT     = *ANY
ACCESS-COUNTER     = *ANY
CODED-CHARACTER-SET = *ANY
SPACE-RELEASE-LOCK = *ANY
-----
NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE  F12=*CANCEL

```

Bild 16: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Das Absenden des Fragebogens bewirkt die Kommandoausführung. Das System gibt die gewünschten Informationen aus.

```

%          33 :20SG:$USER1.TEST.BEISPIEL.1
%:20SG: PUBLIC:          1 FILE RES=          33 FRE=          24 REL=          9 PAGES

```

Der Dialog wird folgendermaßen fortgesetzt:

```
%CMD:fstat?
```

Für die Eingabe des Kommandos FSTATUS (ISP-Format) wird temporär in den geführten Dialog gewechselt (Fragezeichen hinter dem Kommandonamen). Die Kommandos FSTATUS (ISP-Format) und SHOW-FILE-ATTRIBUTES (SDF-Format) haben genau dieselbe Funktion. Da das Kommando FSTATUS ohne Operanden eingegeben wurde, gibt SDF den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES aus. Der Fragebogen ist mit den Default-Werten der wahlweisen Operanden FILE-NAME, INFORMATION, SELECT und OUTPUT vorbesetzt.

```

KOMMANDO      : SHOW-FILE-ATTRIBUTES

-----
FILE-NAME      = test.beispiel.
INFORMATION    = *NAME-AND-SPACE
SELECT         = *by(file-struct=(i,s))
OUTPUT        = *SYSOUT
OUTPUT-OPTIONS = *PARAMETERS(SORT-LIST=*BY-FILENAME)

-----

NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8+=   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

Bild 17: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Der Default-Wert `*ALL` des Operanden `FILE-NAME` wird mit „test.beispiel.“ überschrieben, der Default-Wert `*ALL` des Operanden `SELECT` mit „by“ (`*BY-ATTRIBUTES`). Die Ausgabe eines Unterfragebogens für die daran hängende Struktur wird durch die direkte Angabe von untergeordneten Operanden „(file-struct=i)“ unterdrückt. Die Angabe „file-struct=i“ ordnet dem Operanden `FILE-STRUCTURE` den Wert `*ISAM` zu. Das Absenden des Fragebogens bewirkt die Ausführung des Kommandos. Das System gibt die gewünschten Informationen aus.

```

%          33 :20SG:$USER1.TEST.BEISPIEL.1
%          72 :20SG:$USER1.TEST.BEISPIEL.2
%:20SG: PUBLIC:          2 FILES RES=          105 FRE=          26 REL=          9 PAGES

```

Der Dialog wird folgendermaßen fortgesetzt:

```
%KDO:sh-f-attr test,beispiel.,select=(file-struct=i) _____ (1)
% CMD0051 OPERAND 'INFORMATION' FEHLERHAFT _____ (2)
% CMD0064 OPERANDEN-WERT 'BEISPIEL.' MIT DATEN-TYP '*NAME-AND-SPACE ODER
*SPACE-SUMMARY ODER *ALL-ATTRIBUTES ODER *PARAMETERS() ODER *STATISTICS ODER
*MINIMUM' UNVEREINBAR
%BITTE OPERANDEN EINGEBEN: _____ (3)
%test,beispiel.,select=(file-struct=i) _____ (4)
? _____ (5)
```

- (1) Mit dem Kommando SHOW-FILE-ATTRIBUTES werden über die ISAM-Dateien („select=(file-struct=i)“ entspricht SELECT=\*BY-ATTRIBUTES(FILE-STRUCTURE=\*ISAM)), deren Name mit „test,beispiel.“ beginnt, die Standardinformationen (Größe und Name) angefordert. Fälschlicherweise wird bei der Angabe des teilqualifizierten Dateinamens ein Komma statt eines Punktes eingegeben.
- (2) SDF interpretiert „beispiel.“ als Wert für den Operanden INFORMATION (Eingabe als zweiter Stellungsoperand) und weist ihn als fehlerhaft zurück.
- (3) SDF fordert zur Korrektur der Operanden auf.
- (4) SDF bietet die eingegebenen Operanden zur Korrektur an.
- (5) Die Eingabe des Fragezeichens bewirkt, dass SDF für die Korrektur in den temporär geführten Dialog wechselt.

```
KOMMANDO : SHOW-FILE-ATTRIBUTES
OPERANDEN : FILE-NAME=TEST,INFORMATION=beispiel.,SELECT=*BY-ATTRIBUTES(FILE-STR
           UCTURE=*ISAM)
-----
INFORMATION = !eispiel.
              *NAME-AND-SPACE oder *SPACE-SUMMARY oder *ALL-ATTRIBUTES
              oder *PARAMETERS(ALLOCATION=*NO,BACKUP=*NO,HISTORY=*NO,
              ORGANIZATION=*NO,PASSWORDS=*NO,SECURITY=*NO) oder
              *STATISTICS oder *MINIMUM
-----
NEXT = -
       Folge-Kdo / *CONTINUE / *TEST
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
       F9=REST-SDF-IN   F11=*EXECUTE   F12=*CANCEL
ERROR:  CMD0054 NOCH NICHT KORRIGIERTER FEHLERHAFTER WERT 'BEISPIEL.'
```

Bild 18: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Mit „!“ wird dem Operanden INFORMATION der Default-Wert (NAME-AND-SPACE) zugeordnet. Der Zeilenrest „EISPIEL.“ darf in diesem Fall stehen bleiben. Zur Korrektur des ersten Operanden (FILE-NAME) muss mit der [F7]-Taste (entspricht der Eingabe von „!“ in der NEXT-Zeile, wobei der Zeilenrest „CONTINUE“ gelöscht werden muss, z.B. mit der [LZF]-Taste) im Operandenfragebogen zurückgeblättert werden. Nach Absenden mit der [DUE]-Taste gibt SDF den Beginn des Operandenfragebogens aus.

```

KOMMANDO   : SHOW-FILE-ATTRIBUTES
OPERANDEN  : FILE-NAME=TEST,SELECT=*BY-ATTRIBUTES(FILE-STRUCTURE=*ISAM)

-----
FILE-NAME      = TEST.beispiel.
                *ALL oder Dat.Name
INFORMATION    = *NAME-AND-SPACE
                *NAME-AND-SPACE oder *SPACE-SUMMARY oder *ALL-ATTRIBUTES
                oder *PARAMETERS(ALLOCATION=*NO,BACKUP=*NO,HISTORY=*NO,
                ORGANIZATION=*NO,PASSWORDS=*NO,SECURITY=*NO) oder
                *STATISTICS oder *MINIMUM

-----
NEXT = +
      Folge-Kdo / *CONTINUE / *TEST
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8+=   F9=REST-SDF-IN
      F11=*EXECUTE   F12=*CANCEL

```

Bild 19: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Der Wert des Operanden FILE-NAME wird korrigiert. Das Absenden des Fragebogens mit der [F11]-Taste (entspricht der Eingabe von \*EXECUTE in der NEXT-Zeile) bewirkt die Kommandoausführung.

```

%          33 :20SG:$USER1.TEST.BEISPIEL.1
%          72 :20SG:$USER1.TEST.BEISPIEL.2
%:20SG: PUBLIC:          2 FILES RES=          105 FRE=          26 REL=          9 PAGES

```

Der Dialog wird folgendermaßen fortgesetzt:

```
%KDO:?
```

Die Eingabe eines Fragezeichens statt eines Kommandos bewirkt, dass SDF das Anwendungsbereichs-Menü ausgibt.

```
-----
VERFUEGBARE ANWENDUNGSBEREICHE:
```

1 ACCOUNTING	14 MULTI-CATALOG-AND-PUBSET-MGMT
2 ALL-COMMANDS	15 NETWORK-MANAGEMENT
3 CONSOLE-MANAGEMENT	16 PROCEDURE
4 DATABASE	17 PROGRAM
5 DCAM	18 PROGRAMMING-SUPPORT
6 DEVICE	19 SDF
7 FILE	20 SECURITY-ADMINISTRATION
8 FILE-GENERATION-GROUP	21 SPOOL-PRINT-ADMINISTRATION
9 FILE-TRANSFER	22 SPOOL-PRINT-SERVICES
10 IDOM	23 STORAGE-MANAGEMENT
11 JOB	24 SYSTEM-MANAGEMENT
12 JOB-VARIABLES	25 SYSTEM-TUNING
13 MESSAGE-PROCESSING	26 USER-ADMINISTRATION

```
-----
NEXT = 19
```

```
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F12=*CANCEL
```

Bild 20: Anwendungsbereich-Menü

Mit „19“ wird die Ausgabe des Kommando-Menüs für den Anwendungsbereich SDF veranlasst. Um dies zu erreichen, könnte der Benutzer statt „19“ auch „(sdf)“ angeben.

SDF gibt das Kommando-Menü für den Anwendungsbereich SDF aus.

```
-----
ANW.BEREICH: SDF
```

```
VERFUEGBARE KOMMANDOS:
```

1 HELP-SDF	10 SHOW-SDF-OPTIONS
2 MODIFY-SDF-OPTIONS	11 SHOW-SYNTAX-VERSIONS
3 REMARK	12 START-SDF-A
4 RESET-INPUT-DEFAULTS	13 START-SDF-CONV
5 RESTORE-SDF-INPUT	14 START-SDF-I
6 SHOW-CMD	15 START-SDF-SIM
7 SHOW-INPUT-DEFAULTS	16 START-SDF-U
8 SHOW-INPUT-HISTORY	17 WRITE-TEXT
9 SHOW-RETURNCODE	(!)

```
-----
NEXT = 2
```

```
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F9=REST-SDF-IN F12=*CANCEL
```

Bild 21: Kommando-Menü für den Anwendungsbereich SDF

Mit „2“ wird das Kommando MODIFY-SDF-OPTIONS ausgewählt.

SDF gibt den Operandenfragebogen für das Kommando MODIFY-SDF-OPTIONS aus.

```

ANW.BEREICH: SDF                                KOMMANDO: MODIFY-SDF-OPTIONS
-----
SYNTAX-FILE          = *UNCHANGED
GUIDANCE             = min
LOGGING              = *INPUT-FORM
UTILITY-INTERFACE    = *NEW-MODE
PROCEDURE-DIALOGUE  = *NO
CONTINUATION         = *NEW-MODE
MENU-LOGGING        = *NO
MODE                 = *EXECUTION
DEFAULT-PROGRAM-NAME = *NONE
FUNCTION-KEYS       = *STYLE-GUIDE-MODE
INPUT-HISTORY       = *ON(

-----
NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8+=   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

```

Bild 22: Operandenfragebogen für das Kommando MODIFY-SDF-OPTIONS

Bei dem Operanden GUIDANCE wird der aktuelle Wert (Vorbesetzung) \*NO in \*MINIMUM geändert. Das geschieht mit der Eingabe „min“.

Über die Angabe in der NEXT-Zeile wird gesteuert, wie sich das Abschicken des Bildschirms (DUE-Taste) auswirkt. Die Vorbesetzung \*CONTINUE wirkt wie \*EXECUTE, wenn der aktuelle Fragebogen keinen struktureinleitenden Operandenwert enthält. In diesem Fall ist Operand INPUT-HISTORY mit „ON(“ vorbesetzt. Deshalb wird nach DUE der Unterfragebogen der Struktur \*ON ausgegeben. Die Kommandoausführung ohne Wechsel in den Operanden-Unterfragebogen kann durch Drücken der F11-Taste bzw. mit \*EXECUTE in der NEXT-Zeile erreicht werden.



```
ANW.BEREICH: SDF                                KOMMANDO: MODIFY-SDF-OPTIONS
STRUKTUR   : INPUT-HISTORY=*ON
OPERANDEN  : GUIDANCE=*MINIMUM,INPUT-HISTORY=*ON
-----
NUMBER-OF-INPUTS      = 20
PASSWORD-PROTECTION  = *YES

-----
NEXT = *EXECUTE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL
```

Bild 23: Operanden-Unterfragebogen der Struktur \*ON des MODIFY-SDF-OPTIONS-Kommandos

Mit dem Absenden des Fragebogens wird in den Dialog mit minimaler Benutzerführung gewechselt.

## 5.4 Geführter Dialog

### 5.4.1 Minimale Benutzerführung

Im temporär geführten Dialog wurde der Anwendungsbereich SDF ausgewählt. Nach Ausführung des Kommandos MODIFY-SDF-OPTIONS bleibt diese Einstellung bestehen. SDF setzt deshalb den Dialog mit der Ausgabe des Kommando-Menüs für den Anwendungsbereich SDF fort.

```

ANW.BEREICH: SDF
-----
VERFUEGBARE KOMMANDOS:

 1  HELP-SDF                               10  SHOW-SDF-OPTIONS
 2  MODIFY-SDF-OPTIONS                     11  SHOW-SYNTAX-VERSIONS
 3  REMARK                                  12  START-SDF-A
 4  RESET-INPUT-DEFAULTS                   13  START-SDF-CONV
 5  RESTORE-SDF-INPUT                       14  START-SDF-I
 6  SHOW-CMD                                15  START-SDF-SIM
 7  SHOW-INPUT-DEFAULTS                     16  START-SDF-U
 8  SHOW-INPUT-HISTORY                       17  WRITE-TEXT
 9  SHOW-RETURNCODE                          (!)

-----
NEXT = (file)
KEYS : F1=?  F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 24: Kommando-Menü für den Anwendungsbereich SDF

Die Angabe „(file)“ in der NEXT-Zeile ändert die Einstellung des Anwendungsbereichs. Ab jetzt ist der Anwendungsbereich FILE eingestellt.

SDF gibt das Kommando-Menü für den Anwendungsbereich FILE aus.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

  1  ADD-ALIAS-CATALOG-ENTRY          14  CREATE-TAPE-SET
  2  ADD-FILE-LINK                    15  DELETE-ALTERNATE-INDEX
  3  ADD-ISAM-POOL-LINK                16  DELETE-FILE
  4  ADD-PASSWORD                      17  DELETE-ISAM-POOL
  5  CHANGE-FILE-LINK                  18  DELETE-SYSTEM-FILE
  6  CHECK-FILE-CONSISTENCY           19  DELETE-TAPE-SET
  7  CHECK-IMPORT-DISK-FILE            20  EDIT-FILE-ATTRIBUTES
  8  CONCATENATE-DISK-FILES            21  EDIT-FILE-LINK
  9  COPY-FILE                          22  EXPORT-FILE
 10  COPY-SYSTEM-FILE                  23  EXTEND-TAPE-SET
 11  CREATE-ALTERNATE-INDEX            24  HOLD-ALIAS-SUBSTITUTION      (!)
 12  CREATE-FILE                       25  IMPORT-FILE
 13  CREATE-ISAM-POOL                  26  LOAD-ALIAS-CATALOG

-----
NEXT = +
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F12=*CANCEL

```

Bild 25: Kommando-Menü für den Anwendungsbereich FILE

Da die NEXT-Zeile bereits mit „+“ vorbesetzt ist, bewirkt das Absenden des Bildschirms ([DUE]-Taste) die Ausgabe der nächsten Seite des Kommando-Menüs.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

 27  LOCK-FILE-LINK                    40  SET-FILE-NAME-PREFIX
 28  MODIFY-ACS-OPTIONS                 41  SHOW-ACS-OPTIONS
 29  MODIFY-ALIAS-CATALOG-ENTRY          42  SHOW-ACS-SYSTEM-FILES
 30  MODIFY-FILE-ATTRIBUTES              43  SHOW-ALIAS-CATALOG-ENTRY
 31  PURGE-ALIAS-CATALOG                  (!) 44  SHOW-CE-LOCK
 32  REMOVE-ALIAS-CATALOG-ENTRY          45  SHOW-FILE
 33  REMOVE-FILE-ALLOCATION-LOCKS          46  SHOW-FILE-ATTRIBUTES
 34  REMOVE-FILE-LINK                    47  SHOW-FILE-LINK
 35  REMOVE-ISAM-POOL-LINK                48  SHOW-FILE-LOCKS
 36  REMOVE-PASSWORD                      49  SHOW-FILE-NAME-PREFIX      (!)
 37  REPAIR-DISK-FILES                    50  SHOW-INDEX-ATTRIBUTES
 38  REPAIR-FILE-LOCKS                    51  SHOW-ISAM-POOL-ATTRIBUTES
 39  RESUME-ALIAS-SUBSTITUTION            (!) 52  SHOW-ISAM-POOL-LINK

-----
NEXT = 46
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F8=+
      F9=REST-SDF-IN F12=*CANCEL

```

Bild 26: Kommando-Menü für den Anwendungsbereich FILE

Mit „46“ wird die Ausgabe des Operandenfragebogens für das Kommando SHOW-FILE-ATTRIBUTES veranlasst.

SDF gibt den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES aus. Der Fragebogen ist vorbesetzt mit den Default-Werten der wahlweisen Operanden FILE-NAME, INFORMATION, SELECT und OUTPUT.

ANW.BEREICH: FILE	KOMMANDO: SHOW-FILE-ATTRIBUTES
-----	
FILE-NAME	= test,beispiel.
INFORMATION	= *NAME-AND-SPACE
SELECT	= ?ALL
OUTPUT	= *SYSOUT
OUTPUT-OPTIONS	= *PARAMETERS(SORT-LIST=*BY-FILENAME)
-----	
NEXT =	*CONTINUE
KEYS :	F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8+= F9=REST-SDF-IN
	F11=*EXECUTE F12=*CANCEL

Bild 27: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Als Wert für den Operanden FILE-NAME wird fälschlicherweise „test,beispiel.“ angegeben (Komma statt Punkt). Über den Operanden SELECT werden mit „?“ weitere Informationen angefordert. Der Zeilenrest „ALL“ darf in diesem Fall stehen bleiben. Über die Angabe in der NEXT-Zeile wird gesteuert, was das Absenden des Bildschirms (DUE-Taste) bewirkt. Die Vorbesetzung \*CONTINUE steht für \*EXECUTE oder + und bewirkt in diesem Fall, dass SDF den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES mit einer Erläuterung des Operanden SELECT ausgibt.

SDF gibt den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES erneut aus. Er enthält die Aufforderung, den Wert des Operanden FILE-NAME zu korrigieren und zusätzliche Informationen über den Operanden SELECT. Das hinter dem Operandenwert \*BY-ATTRIBUTES stehende Klammernpaar weist darauf hin, dass \*BY-ATTRIBUTES eine Struktur einleitet.

```

ANW.BEREICH: FILE                                KOMMANDO: SHOW-FILE-ATTRIBUTES
OPERANDEN  : SELECT=*ALL

-----
FILE-NAME   = test.beispiel.
INFORMATION = *NAME-AND-SPACE
SELECT      = *by
            *ALL oder *BY-ATTRIBUTES()
            Kriterien fuer die Dateiauswahl (ALL: alle Dateien;
            BY-ATTRIBUTES: Dateien mit den angegebenen Merkmalen)
OUTPUT      = *SYSOUT
OUTPUT-OPTIONS = *PARAMETERS(SORT-LIST=*BY-FILENAME)

-----
NEXT = *CONTINUE
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F8=+   F9=REST-SDF-IN
      F11=*EXECUTE  F12=*CANCEL

ERROR:  CMD0147 FEHLERHAFT(E)N OPERANDEN KORRIGIEREN

```

Bild 28: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Der Operandenwert für FILE-NAME wird korrigiert. Der Default-Wert \*ALL des Operanden SELECT wird mit „by“ (\*BY-ATTRIBUTES) überschrieben. Das Absenden des Fragebogens bewirkt die Ausgabe eines Unterfragebogens für die Struktur \*BY-ATTRIBUTES.

SDF gibt den Unterfragebogen für die Struktur \*BY-ATTRIBUTES aus.

```

ANW.BEREICH: FILE                                KOMMANDO: SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES

-----
CREATION-DATE      = *ANY
EXPIRATION-DATE    = *ANY
LAST-ACCESS-DATE   = *ANY
LAST-CHANGE-DATE   = *ANY
SUPPORT            = *ANY
VOLUME             = *ANY
SIZE               = *ANY
NUMBER-OF-EXTENTS  = *ANY
NUMBER-OF-FREE-PAGES = *ANY
HIGHEST-USED-PAGE  = *ANY
BLOCK-COUNTER      = *ANY
ACCESS             = *ANY
PASSWORD           = *ANY

-----
NEXT = +
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN  F11=*EXECUTE  F12=*CANCEL

```

Bild 29: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Da die NEXT-Zeile bereits mit „+“ vorbesetzt ist, bewirkt das Absenden des Bildschirms (**DUE**-Taste) die Ausgabe der nächsten Seite des Operandenfragebogens.

```

ANW.BEREICH: FILE                                KOMMANDO: SHOW-FILE-ATTRIBUTES
STRUKTUR   : SELECT=*BY
OPERANDEN  : FILE-NAME=TEST.BEISPIEL.,SELECT=*BY-ATTRIBUTES
-----
USER-ACCESS      = *ANY
BASIC-ACL        = *ANY
ACL              = *ANY
GUARDS           = *ANY
PROTECTION-ACTIVE = *ANY
STATUS           = *ANY
FILE-STRUCTURE   = i,s
BLOCK-CONTROL-INFO = *ANY
BACKUP-CLASS     = *ANY
MIGRATE          = *ANY
STORAGE-LEVEL    = *ANY
GENERATIONS      = *NO
TYPE-OF-FILES    = *ANY
-----
NEXT = mod-sdf-opt guid=med
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F11=*EXECUTE   F12=*CANCEL

```

Bild 30: Operanden-Unterfragebogen für die Struktur \*BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES

Die Angabe „i,s“ (entspricht „\*isam,\*sam“) ordnet den Operanden FILE-STRUCTURE die Werte \*ISAM und \*SAM zu. Die sonst bei Listenangabe erforderlichen Klammern können hier entfallen. In der NEXT-Zeile kann der Benutzer unabhängig vom Anwendungsbereich ein beliebiges Kommando als Folge-Kommando angeben. Dies wird nach Ausführung des Kommandos SHOW-FILE-ATTRIBUTES ausgeführt. Als Folge-Kommando wird das Kommando MODIFY-SDF-OPTIONS angegeben („mod-sdf-opt guid=med“). Damit wird in den Dialog mit mittlerer Benutzerführung gewechselt. Das Absenden des Fragebogens bewirkt zuerst die Ausführung des Kommandos SHOW-FILE-ATTRIBUTES. Das System gibt die gewünschten Informationen aus. Danach wird die Benutzerführung gewechselt. Zur Fortsetzung des Dialogs wird die **DUE**-Taste gedrückt.

```

%      33 :20SG:$USER1.TEST.BEISPIEL.1
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%      48 :20SG:$USER1.TEST.BEISPIEL.3
%       3 :20SG:$USER1.TEST.BEISPIEL.4
%:20SG: PUBLIC:      4 FILES RES=      156 FRE=      30 REL=      9 PAGES
%PLEASE ACKNOWLEDGE

```

## 5.4.2 Mittlere Benutzerführung

Nach dem Drücken der **[DUE]**-Taste gibt SDF erneut das Kommando-Menü für den Anwendungsbereich FILE aus. Dieses besteht bei mittlerer Benutzerführung aus mehreren Seiten. Die erste Seite wird angezeigt. Im Gegensatz zum Menü, das SDF bei minimaler Führung ausgibt, enthält es Erläuterungen der Kommandos (Hilfe-Texte). Ab dieser Führungsstufe werden außerdem in dem hier eingestellten Style-Guide-Modus nach der NEXT-Zeile die möglichen Steueranweisungen angezeigt.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:
  1  ADD-ALIAS-CATALOG-ENTRY      : Fuegt dem aktuellen Aliaskatalog einen
                                     Eintrag hinzu.
  2  ADD-FILE-LINK                : Speichert Angaben ueber die Merkmale einer
                                     Datei unter einem bestimmten Dateikettungs-
                                     amen. Diese Angaben werden beim Eroeffnen
                                     der Datei anstelle der entsprechenden
                                     Angaben im Programm verwendet
  3  ADD-ISAM-POOL-LINK           : Ordnet einen Kettungs-Namen einem
                                     vorhandenen ISAM-Pool zu
  4  ADD-PASSWORD                 : Traegt Kennwoerter fuer Dateien oder JV's
                                     in die Kennwortliste des Auftrags ein
  5  CHANGE-FILE-LINK            : Aendert den Kettungsnamen einer Datei
  6  CHECK-FILE-CONSISTENCY       : Ueberprueft die Konsistenz einer ISAM Datei
-----
NEXT = ++
      Nummer / Folge-Kdo / (Folge-Ber) / *DOM-MENU
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F12=*CANCEL

```

Bild 31: Kommando-Menü für den Anwendungsbereich FILE

Mit der Angabe „++“ in der NEXT-Zeile wird zum Ende des Menüs geblättert.

SDF gibt die letzte Seite des Kommando-Menüs für den Anwendungsbereich FILE aus.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

57  START-SORT                : Startkommando fuer das Sortieren mit der
                                Anforderung von Folgeanweisungen.
58  STOP-FILE-CACHING        : Beendet das Caching einer gerade offenen
                                und gecachten Datei oder einer Datei in
                                der sich noch      Daten im Cache
                                befinden
59  STORE-ALIAS-CATALOG      : Speichert den Aliaskatalog in eine Datei.
60  UNLOCK-FILE-LINK        : Hebt die Sperre fuer einen Dateikettungsname
                                en auf. Der Dateikettungsname kann wieder
                                geloescht werden; waehrend der Sperre
                                versuchtes Loeschen wird jetzt wirksam
-----
NEXT = -
      Nummer / Folge-Kdo / (Folge-Ber) / *DOM-MENU
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F9=REST-SDF-IN F12=*CANCEL

```

Bild 32: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **[F7]**-Taste) wird um einen Bildschirm im Menü zurückgeblättert.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

55  START-FILE-CACHING      : Startet Caching fuer eine gegenwaertig
                                offene, aber nicht gecachte Datei
56  START-PERCON           : Unterstuetzt Dateiumsetzungen:
                                GRUNDFUNKTIONEN:
                                - Kopieren von Dateien,
                                - Editieren eines Bandinhaltes,
                                - Duplizieren von Baendern,
                                NEBENFUNKTIONEN:
                                - Satzauswahl,
                                - Satzstruktur aendern,
                                - Seitenaufbereitung,
                                - Gruppenbildung,
                                - Eingabeband positionieren.
-----
NEXT = -
      Nummer / Folge-Kdo / (Folge-Ber) / *DOM-MENU
KEYS : F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F8=+
      F9=REST-SDF-IN F12=*CANCEL

```

Bild 33: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **[F7]**-Taste) wird um einen Bildschirm im Menü zurückgeblättert.



```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

49 SHOW-FILE-NAME-PREFIX      : Zeigt das eingestellte Dateinamenspraefix.
   (SOFORTIGE AUSFUEHRUNG!)
50 SHOW-INDEX-ATTRIBUTES     : Informiert ueber die Sekundaerindizes
   einer NK-ISAM Datei
51 SHOW-ISAM-POOL-ATTRIBUTES  : Gibt Attribute und Belegungs-Zustaende
   von ISAM-Pools aus
52 SHOW-ISAM-POOL-LINK       : Gibt die Zuordnung von ISAM-Pools zu
   Pool-Link-Namen aus
53 SHOW-SYSTEM-FILE-ASSIGNMENTS : Gibt Informationen ueber Systemdateien aus
54 SORT-FILE                  : Kommando fuer das Sortieren mit einfacher
   Versorgung (Ein- und Ausgabedatei,
   Sortierfeld-Beschreibung und der Angabe
   von Dateien mit Sortier-Anweisungen).
-----
NEXT = -
      Nummer / Folge-Kdo / (Folge-Ber) / *DOM-MENU
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F12=*CANCEL

```

Bild 34: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **[F7]**-Taste) wird um einen Bildschirm im Menü zurückgeblättert.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

43 SHOW-ALIAS-CATALOG-ENTRY  : Gibt Eintraege des Aliaskatalogs aus.
44 SHOW-CE-LOCK              : Gibt aus, wer den CE-Lock fuer diese
   Datei/JV haelt.
45 SHOW-FILE                  : Ermoeoglicht im Dialogbetrieb das Lesen
   einer Datei ohne Aufruf eines Dateibearbeit
   ers
46 SHOW-FILE-ATTRIBUTES      : Gibt Dateierkmale aus, die im Katalog
   gespeichert sind
47 SHOW-FILE-LINK            : Gibt Dateierkmale aus, die unter einem
   bestimmten Dateikettungsnamen gespeichert
   sind
48 SHOW-FILE-LOCKS           : Zeigt die gerade gehaltenen Dateisperren
   fuer eine Datei an
-----
NEXT = 46
      Nummer / Folge-Kdo / (Folge-Ber) / *DOM-MENU
KEYS : F1=?   F3=*EXIT   F5=*REFRESH   F6=*EXIT-ALL   F7=-   F8=+
      F9=REST-SDF-IN   F12=*CANCEL

```

Bild 35: Kommando-Menü für den Anwendungsbereich FILE

Mit „46“ wird der Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES angefordert.

SDF gibt den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES aus. Im Gegensatz zu dem Fragebogen bei minimaler Benutzerführung enthält dieser Angaben über die zulässigen Operandenwerte. Das hinter dem Wert \*BY-ATTRIBUTES stehende Klammernpaar weist darauf hin, dass \*BY-ATTRIBUTES eine Struktur einleitet, für die es einen eigenen Fragebogen gibt.

ANW.BEREICH: FILE		KOMMANDO: SHOW-FILE-ATTRIBUTES
FILE-NAME	=	<b>test.beispiel.</b> *ALL oder Dat.Name
INFORMATION	=	*NAME-AND-SPACE *NAME-AND-SPACE oder *SPACE-SUMMARY oder *ALL-ATTRIBUTES oder *PARAMETERS(ALLOCATION=*NO, BACKUP=*NO, HISTORY=*NO, ORGANIZATION=*NO, PASSWORDS=*NO, SECURITY=*NO) oder *STATISTICS oder *MINIMUM
SELECT	=	<b>*by(file-struct=(i,s))</b> *ALL oder *BY-ATTRIBUTES()
OUTPUT	=	*SYSOUT *NONE oder *SYSOUT oder *SYSLIST() oder *PRINTER() oder Dat.Name(FORM-NAME=*STD)
-----		
NEXT	=	<b>mod-sdf-opt guid=max</b> Folge-Kdo / (Folge-Ber) / *CONTINUE / *DOM-MENU / *TEST
KEYS	:	F1=? F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8== F9=REST-SDF-IN F11=*EXECUTE F12=*CANCEL

Bild 36: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Als Operandenwert für FILE-NAME wird „test.beispiel.“ angegeben. Der Default-Wert \*ALL des Operanden SELECT wird mit „by“ (\*BY-ATTRIBUTES) überschrieben. Die Ausgabe eines Unterfragebogen für die daran hängende Struktur wird durch das Klammernpaar unterdrückt. Die Angabe „file-struct=(i,s)“ ordnet dem Operanden FILE-STRUCTURE die Werte \*ISAM und \*SAM zu. Hier ist es erforderlich, die beiden Werte in Klammern einzuschließen „(i,s)“. In der NEXT-Zeile kann der Benutzer unabhängig vom Anwendungsbe-  
reich ein beliebiges Kommando als Folge-Kommando angeben. Dieses wird nach Ausführung des Kommandos SHOW-FILE-ATTRIBUTES ausgeführt. Als Folge-Kommando wird das Kommando MODIFY-SDF-OPTIONS angegeben („mod-sdf-opt guid=max“). Damit wird in den Dialog mit maximaler Benutzerführung gewechselt. Das Absenden des Fragebogens bewirkt zuerst die Ausführung des Kommandos SHOW-FILE-ATTRIBUTES. Das System gibt die gewünschten Informationen aus. Danach wird die Benutzerführung gewechselt. Zur Fortsetzung des Dialogs wird die **[DUE]**-Taste gedrückt.

```
%      33 :20SG:$USER1.TEST.BEISPIEL.1
%      72 :20SG:$USER1.TEST.BEISPIEL.2
%      48 :20SG:$USER1.TEST.BEISPIEL.3
%       3 :20SG:$USER1.TEST.BEISPIEL.4
%:20SG: PUBLIC:      4 FILES RES=      156 FRE=      30 REL=      9 PAGES
%PLEASE ACKNOWLEDGE
```

### 5.4.3 Maximale Benutzerführung

Nach dem Drücken der **[DUE]**-Taste gibt SDF erneut das Kommando-Menü für den Anwendungsbereich FILE aus. Dieses unterscheidet sich von dem bei mittlerer Führung nur in unwesentlichen Details.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

 1 ADD-ALIAS-CATALOG-ENTRY      : Fuegt dem aktuellen Aliaskatalog einen
                                Eintrag hinzu.
 2 ADD-FILE-LINK                 : Speichert Angaben ueber die Merkmale einer
                                Datei unter einem bestimmten Dateikettungs-
                                amen. Diese Angaben werden beim Eroeffnen
                                der Datei anstelle der entsprechenden
                                Angaben im Programm verwendet
 3 ADD-ISAM-POOL-LINK           : Ordnet einen Kettungs-Namen einem
                                vorhandenen ISAM-Pool zu
 4 ADD-PASSWORD                  : Traegt Kennwoerter fuer Dateien oder JV's
                                in die Kennwortliste des Auftrags ein
 5 CHANGE-FILE-LINK             : Aendert den Kettungsnamen einer Datei
 6 CHECK-FILE-CONSISTENCY       : Ueberprueft die Konsistenz einer ISAM Datei
-----
NEXT = ++
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH F6=*EXIT-ALL F8=+  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 37: Kommando-Menü für den Anwendungsbereich FILE

Mit der Angabe „++“ in der NEXT-Zeile wird zum Ende des Menüs geblättert. SDF gibt die letzte Seite des Kommando-Menüs für den Anwendungsbereich FILE aus.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

57  START-SORT                : Startkommando fuer das Sortieren mit der
                                Anforderung von Folgeanweisungen.
58  STOP-FILE-CACHING         : Beendet das Caching einer gerade offenen
                                und gecachten Datei oder einer Datei von
                                der sich noch      Daten im Cache
                                befinden
59  STORE-ALIAS-CATALOG       : Speichert den Aliaskatalog in eine Datei.
60  UNLOCK-FILE-LINK          : Hebt die Sperre fuer einen Dateikettungsname
                                auf. Der Dateikettungsname kann wieder
                                geloescht werden; waehrend der Sperre
                                versuchtes Loeschen wird jetzt wirksam
-----

NEXT = -
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F7=-  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 38: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **F7**-Taste) wird um einen Bildschirm im Kommando-Menü zurückgeblättert.

SDF gibt die vorletzte Seite des Kommando-Menüs für den Anwendungsbereich FILE aus.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

55  START-FILE-CACHING        : Startet Caching fuer eine gegenwaertig
                                offene, aber nicht gecachte Datei
56  START-PERCON              : Unterstuetzt Dateiumsetzungen:
                                GRUNDFUNKTIONEN:
                                - Kopieren von Dateien,
                                - Editieren eines Bandinhaltes,
                                - Duplizieren von Baendern,
                                NEBENFUNKTIONEN:
                                - Satzauswahl,
                                - Satzstruktur aendern,
                                - Seitenaufbereitung,
                                - Gruppenbildung,
                                - Eingabeband positionieren.
-----

NEXT = -
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F7=-  F8+=  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 39: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **[F7]**-Taste) wird um einen Bildschirm im Kommando-Menü zurückgeblättert. SDF gibt die gewünschte Seite des Kommando-Menüs für den Anwendungsbereich FILE aus.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

 49 SHOW-FILE-NAME-PREFIX      : Zeigt das eingestellte Dateinamenspraefix.
    (SOFORTIGE AUSFUEHRUNG!)
 50 SHOW-INDEX-ATTRIBUTES     : Informiert ueber die Sekundaerindizes
                               einer NK-ISAM Datei
 51 SHOW-ISAM-POOL-ATTRIBUTES  : Gibt Attribute und Belegungs-Zustaende
                               von ISAM-Pools aus
 52 SHOW-ISAM-POOL-LINK        : Gibt die Zuordnung von ISAM-Pools zu
                               Pool-Link-Namen aus
 53 SHOW-SYSTEM-FILE-ASSIGNMENTS : Gibt Informationen ueber Systemdateien aus
 54 SORT-FILE                   : Kommando fuer das Sortieren mit einfacher
                               Versorgung (Ein- und Ausgabedatei,
                               Sortierfeld-Beschreibung und der Angabe
                               von Dateien mit Sortier-Anweisungen).
-----
NEXT = -
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F8+= F9=REST-SDF-IN F12=*CANCEL

```

Bild 40: Kommando-Menü für den Anwendungsbereich FILE

Mit „-“ (alternativ **[F7]**-Taste) wird um einen weiteren Bildschirm im Kommando-Menü zurückgeblättert.

```

ANW.BEREICH: FILE
-----
VERFUEGBARE KOMMANDOS:

 43 SHOW-ALIAS-CATALOG-ENTRY  : Gibt Eintraege des Aliaskatalogs aus.
 44 SHOW-CE-LOCK               : Gibt aus, wer den CE-Lock fuer diese
                               Datei/JV haelt.
 45 SHOW-FILE                   : Ermoeeglicht im Dialogbetrieb das Lesen
                               einer Datei ohne Aufruf eines Dateibearbeit
                               ers
 46 SHOW-FILE-ATTRIBUTES       : Gibt Dateimerkmale aus, die im Katalog
                               gespeichert sind
 47 SHOW-FILE-LINK             : Gibt Dateimerkmale aus, die unter einem
                               bestimmten Dateikettungsnamen gespeichert
                               sind
 48 SHOW-FILE-LOCKS            : Zeigt die gerade gehaltenen Dateisperren
                               fuer eine Datei an
-----
NEXT = 46
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F7=- F8+= F9=REST-SDF-IN F12=*CANCEL

```

Bild 41: Kommando-Menü für den Anwendungsbereich FILE

Mit der Eingabe „46“ wird der Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES angefordert.

SDF gibt den Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES aus. Im Gegensatz zu dem Fragebogen bei mittlerer Benutzerführung enthält dieser Zusatzangaben zu den zulässigen Operandenwerten, z.B. minimale und maximale Länge, und Erläuterungen der Operanden (Hilfe-Texte). Da es bei maximaler Führung auch für die Struktur \*PARAMETERS einen eigenen Unterfragebogen gibt, sind die Operanden dieser Struktur in dem Fragebogen nicht aufgelistet. Die Klammernpaare hinter den Werten \*PARAMETERS und \*BY-ATTRIBUTES weisen darauf hin, dass diese eine Struktur einleiten. Da der Fragebogen bei maximaler Führung umfangreicher ist, besteht er aus zwei Seiten.

ANW.BEREICH: FILE		KOMMANDO: SHOW-FILE-ATTRIBUTES
FILE-NAME	= <b>test.beispiel.</b>	*ALL oder Dateiname_1..54 mit-Musterzeichen(80) Namen der Dateien, ueber die Informationen gewünscht sind
INFORMATION	= *NAME-AND-SPACE	*NAME-AND-SPACE oder *SPACE-SUMMARY oder *ALL-ATTRIBUTES oder *PARAMETERS() oder *STATISTICS oder *MINIMUM
SELECT	= <b>*by(file-struct=(i,s))</b>	Art der auszugebenden Information *ALL oder *BY-ATTRIBUTES() Kriterien fuer die Dateiauswahl (ALL: alle Dateien; BY-ATTRIBUTES: Dateien mit den angegebenen Merkmalen)
NEXT	= <b>*dom</b>	Folge-Kommando / (Folge-Bereich) / *CONTINUE / *DOMAIN-MENU / *TEST
KEYS	:	F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F8=+ F9=REST-SDF-IN F11=*EXECUTE F12=*CANCEL

Bild 42: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES

Als Operandenwert für FILE-NAME wird „test.beispiel.“ angegeben. Der Default-Wert \*ALL des Operanden SELECT wird mit „\*by“ (\*BY-ATTRIBUTES) überschrieben. Die Ausgabe eines Unterfragebogens für die daran hängende Struktur wird durch das Klammernpaar unterdrückt. Die Angabe „file-struct=(i,s)“ ordnet dem Operanden FILE-STRUCTURE den Wert \*ISAM und \*SAM zu. In der NEXT-Zeile kann der Benutzer mit der Angabe \*DOMAIN-MENU das Anwendungsbereichs-Menü anfordern. Dazu reicht die Eingabe „\*dom“ und [LZF]-Taste aus. „\*dom“ ist eine eindeutige Abkürzung. Die [LZF]-Taste bewirkt, dass die von der Vorbesetzung „\*CONTINUE“ verbliebenen Zeichen „\*TINUE“ gelöscht werden. Würde dort „\*domTINUE“ stehen, so würde SDF das als fehlerhafte Angabe betrachten. Mit dem Wegschicken des Fragebogens wird die Ausführung des Kommandos

SHOW-FILE-ATTRIBUTES veranlasst. Das Absenden des Fragebogens bewirkt zuerst die Ausführung des Kommandos. Das System gibt die gewünschten Informationen aus. Nach dem Drücken der **[DUE]**-Taste gibt SDF das Anwendungsbereichs-Menü aus.

```
%          33 :20SG:$USER1.TEST.BEISPIEL.1
%          72 :20SG:$USER1.TEST.BEISPIEL.2
%          48 :20SG:$USER1.TEST.BEISPIEL.3
%           3 :20SG:$USER1.TEST.BEISPIEL.4
%:20SG: PUBLIC:          4 FILES RES=          156 FRE=          30 REL=          9 PAGES
%PLEASE ACKNOWLEDGE
```

SDF gibt das Anwendungsbereichs-Menü aus. Dieses unterscheidet sich von dem bei mittlerer Benutzerführung nur in unwesentlichen Details. Im Gegensatz zu dem bei minimaler Führung enthält es Erläuterungen der Anwendungsbereiche (Hilfe-Texte).

```
-----
VERFUEGBARE ANWENDUNGSBEREICHE:

 1 ACCOUNTING           : Ausgabe von Informationen ueber die
                        : Benutzerkennung und Einfuegen von Daten in
                        : Abrechnungssaetze
 2 ALL-COMMANDS        : Ausgabe aller Kommandonamem in
                        : alphabetischer Reihenfolge
 3 CONSOLE-MANAGEMENT  : Steuerung Operator-Arbeitsplatz
 4 DATABASE            : Management und Administration von
                        : Datenbanken
 5 DCAM                : Steuern des Teilhaberbetriebs (DCAM)
 6 DEVICE              : Information ueber Geraete und Datentraeger
 7 FILE                : Verwalten von Dateien
 8 FILE-GENERATION-GROUP : Verwalten von Dateigenerationsgruppen

-----
NEXT = +
      Nummer / Folge-Kommando / (Folge-Bereich)
KEYS : F5=*REFRESH   F8=+   F9=REST-SDF-IN
```

Bild 43: Anwendungsbereichs-Menü

Da die NEXT-Zeile bereits mit „+“ vorbesetzt ist, bewirkt das Abschicken des Bildschirms (**[DUE]**-Taste) die Ausgabe der nächsten Seite des Anwendungsbereichs-Menüs.

```

-----
VERFUEGBARE ANWENDUNGSBEREICHE:

  9  FILE-TRANSFER           : Uebertragen von Dateien zwischen Rechnern
                                ueber Netzverbindungen
 10  IDOM                     : 
 11  JOB                       : Auftragssteuerung
 12  JOB-VARIABLES           : Verwalten von Jobvariablen
 13  MESSAGE-PROCESSING      : Verwalten von Meldungsdateien
 14  MULTI-CATALOG-AND-PUBSET-MGMT : Steuern des Zugriffs auf Dateien im
                                lokalen BS2000-Rechnerverbund
 15  NETWORK-MANAGEMENT      : Steuerung von DCM Anwendungen und
                                Verbindungen
 16  PROCEDURE               : Steuern von Kommandoprozeduren
 17  PROGRAM                  : Steuern von Programmlaeufen

-----
NEXT = +
      Nummer / Folge-Kommando / (Folge-Bereich)
KEYS : F5=*REFRESH  F7=-  F8+=  F9=REST-SDF-IN

```

Bild 44: Anwendungsbereichs-Menü

Mit „+“ (bereits vorbesetzt) wird im Anwendungsbereichs-Menü weitergeblättert.

```

-----
VERFUEGBARE ANWENDUNGSBEREICHE:

 18  PROGRAMMING-SUPPORT     : Aufruf von Compilern und Programmierwerkzeu
                                gen
 19  SDF                     : Steuern der Dialogschnittstelle
 20  SECURITY-ADMINISTRATION : Verwaltung von Zugriffsberechtigungen und
                                Beweissicherung
 21  SPOOL-PRINT-ADMINISTRATION : Spool & Print Verwaltung Oberflaeche
 22  SPOOL-PRINT-SERVICES    : Spool & Print Benutzer Oberflaeche
 23  STORAGE-MANAGEMENT      : Verwalten von externen Speichern
 24  SYSTEM-MANAGEMENT       : Dynamische Kontrolle der Subsystemkonfigura
                                tion

-----
NEXT = 19
      Nummer / Folge-Kommando / (Folge-Bereich)
KEYS : F5=*REFRESH  F7=-  F8+=  F9=REST-SDF-IN

```

Bild 45: Anwendungsbereichs-Menü

Mit „19“ wird der Anwendungsbereich SDF ausgewählt.



SDF gibt das Kommando-Menü für den Anwendungsbereich SDF aus.

```

ANW.BEREICH: SDF
-----
VERFUEGBARE KOMMANDOS:

 1  HELP-SDF                : Gibt Auskunft ueber SDF
 2  MODIFY-SDF-OPTIONS     : Aktiviert oder deaktiviert eine
                             Benutzersyntaxdatei und aendert die
                             Einstellungen der SDF-Optionen
 3  REMARK                  : Schreibt Bemerkungen in eine Kommandodatei
 4  RESET-INPUT-DEFAULTS   : Setzt einige oder alle taskspezifischen
                             Default-Werte zurueck.
 5  RESTORE-SDF-INPUT      : Ermoeoglicht die erneute Eingabe einer
                             bereits erfolgten Eingabe. Die
                             Kommandos/Anweisungen werden im aktuell
                             Dialogfuehrungsmodus angezeigt
 6  SHOW-CMD                : Gibt die Syntaxbeschreibung eines
                             Kommandos nach SYSOUT oder SYSLST aus.
-----
NEXT = +
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F8+=+  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 46: Kommando-Menü für den Anwendungsbereich SDF

Mit „+“ (alternativ **[F8]**-Taste) wird im Kommando-Menü um einen Bildschirm weitergeblättert.

```

ANW.BEREICH: SDF
-----
VERFUEGBARE KOMMANDOS:

 7  SHOW-INPUT-DEFAULTS    : Zeigt die Liste der taskspezifischen
                             Default-Werte an.
 8  SHOW-INPUT-HISTORY     : Zeigt eine Liste der gespeicherten
                             Eingaben.
 9  SHOW-RETURNCODE        : Zeigt den Kommando-Returncode des letzten
                             Kommandos.
    (SOFORTIGE AUSFUEHRUNG!)
10  SHOW-SDF-OPTIONS       : Gibt die Namen der momentan aktiven
                             SDF-Syntaxdateien und die Einstellungen
                             der SDF-Options aus
11  SHOW-SYNTAX-VERSIONS   : Zeigt die Versionen der Syntax-beschreibung
                             en der eingesetzten Produkte und
                             Komponenten an
-----
NEXT = 10
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F7=-  F8+=+  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 47: Kommando-Menü für den Anwendungsbereich SDF

Mit „10“ wird Operandenfragebogen für das Kommando SHOW-SDF-OPTIONS angefordert.

```

ANW.BEREICH: SDF                                KOMMANDO: SHOW-SDF-OPTIONS
-----
INFORMATION      = *user
                  *ALL oder *USER
                  Gibt an, ob alle Informationen oder nur Informationen
                  ueber die taskspezifischen SDF-Optionen ausgegeben
                  werden sollen.
-----
NEXT = *EXECUTE
      Folge-Kommando / (Folge-Bereich) / *CONTINUE / *DOMAIN-MENU / *TEST
KEYS : F3=*EXIT F5=*REFRESH F6=*EXIT-ALL F9=REST-SDF-IN F11=*EXECUTE F12=*CANCEL

```

Bild 48: Operandenfragebogen für das Kommando SHOW-SDF-OPTIONS

Der Default-Wert \*ALL des Operanden INFORMATION wird mit „\*user“ überschrieben.

Nach Abschicken des Bildschirms mit **[DUE]** gibt SDF Informationen über alle SDF-Optionen aus, die der Benutzer selbst mit MODIFY-SDF-OPTIONS verändern kann:

```

% USER          : *NONE
%CURRENT SDF OPTIONS :
% GUIDANCE      : *MAXIMUM
% 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
%PLEASE ACKNOWLEDGE

```

Nach dem Drücken der **[DUE]**-Taste gibt SDF wieder das Kommando-Menü für den Anwendungsbereich SDF aus.

```

ANW.BEREICH: SDF
-----
VERFUEGBARE KOMMANDOS:

 1  HELP-SDF                : Gibt Auskunft ueber SDF
 2  MODIFY-SDF-OPTIONS     : Aktiviert oder deaktiviert eine
                             Benutzersyntaxdatei und aendert die
                             Einstellungen der SDF-Optionen
 3  REMARK                  : Schreibt Bemerkungen in eine Kommandodatei
 4  RESET-INPUT-DEFAULTS   : Setzt einige oder alle taskspezifischen
                             Default-Werte zurueck.
 5  RESTORE-SDF-INPUT      : Ermoeoglicht die erneute Eingabe einer
                             bereits erfolgten Eingabe. Die
                             Kommandos/Anweisungen werden im aktuell
                             Dialogfuehrungsmodus angezeigt
 6  SHOW-CMD               : Gibt die Syntaxbeschreibung eines
                             Kommandos nach SYSOUT oder SYSLST aus.
-----
NEXT = exit-job
      Nummer / Folge-Kommando / (Folge-Bereich) / *DOMAIN-MENU
KEYS : F3=*EXIT  F5=*REFRESH  F6=*EXIT-ALL  F8=+  F9=REST-SDF-IN  F12=*CANCEL

```

Bild 49: Kommando-Menü für den Anwendungsbereich SDF

Statt in der NEXT-Zeile mit einer Nummer ein Kommando aus dem Menü auszuwählen, kann der Benutzer dort auch ein beliebiges Kommando angeben. Mit dem Abschicken des Menüs wird dieses Kommando ausgeführt. Das Kommando EXIT-JOB wird in der NEXT-Zeile angegeben. Mit dem Abschicken des Fragebogens wird die Sitzung beendet.



---

## 6 SDF-Syntaxdateien

Im Lieferumfang von Produkten mit SDF-Oberfläche sind so genannte Syntaxdateien enthalten. Diese enthalten:

- Informationen über die Syntax von Kommandos und Anweisungen,
- Informationen darüber, wie diese Kommandos im BS2000 implementiert sind, z.B.
  - die Namen der Systemmodule, über die die Kommandoausführung veranlasst wird,
  - die Festlegungen für die Parameterübergaben an die ausführenden Systemmodule,
- Informationen über Privilegien, die der aufrufende Benutzer besitzen muss,
- allgemeine und kommandospezifische Festlegungen für die Dialogführung,
- erläuternde Texte zu den Kommandos und ihren Operanden (Hilfe-Texte).

Zur Verarbeitung einer Kommandoeingabe holt SDF sich die dazu benötigten Informationen aus den aktivierten Syntaxdateien.

In analoger Weise unterstützt SDF die Eingabe von Anweisungen an Programme, wenn diese in einer Syntaxdatei definiert sind, z. B. Anweisungen an SDF-A oder an SDF-U.

Mit dem Softwareprodukt SDF-A (**S**ystem **D**ialog **F**acility-**A**dmistration) sowie mit dem Dienstprogramm SDF-U (**S**ystem **D**ialog **F**acility-**U**tility) lassen sich Syntaxdateien bearbeiten. Mit dem Dienstprogramm SDF-I lassen sich Syntaxdateien in eine Syntaxdatei einmischen.

Es gibt drei Arten von Syntaxdateien:

- Systemsyntaxdatei
- Gruppensyntaxdatei
- Benutzersyntaxdatei

Für eine Task können mehrere Systemsyntaxdateien aktiviert sein: eine Basis-Systemsyntaxdatei und mehrere Subsystem-Syntaxdateien. Daneben kann für eine Task je eine Gruppen- und eine Benutzersyntaxdatei aktiviert sein. Für diesen Fall gibt es eine feste Dateihierarchie, die festlegt, wie SDF mit mehreren Syntaxdateien arbeitet.

## 6.1 Systemsyntaxdatei

Die Benutzung von SDF setzt voraus, dass eine Basis-Systemsyntaxdatei vorhanden ist, die automatisch beim Systemladen aktiviert wird. Daneben können ab SDF V3.0 weitere Systemsyntaxdateien, die Subsystem-Syntaxdateien, vorhanden sein.

Systemweit kann immer nur eine Basis-Systemsyntaxdatei aktiviert sein.

In der SDF-Parameterdatei werden die Einträge über Systemsyntaxdateien verwaltet. Die Verwaltung der Systemsyntaxdateien nimmt die Systembetreuung mit dem Kommando `MODIFY-SDF-PARAMETERS` vor. Mit dem Dienstprogramm `SDF-PAR` kann die Systembetreuung eine SDF-Parameterdatei erstellen.

### *Basis-Systemsyntaxdatei*

Die Basis-Systemsyntaxdatei enthält die Definitionen der Anwendungsbereiche, der Standardanweisungen und der SDF-spezifischen Kommandos (aus dem Anwendungsbereich SDF) sowie die systemweiten Einstellungen der Globalinformation. Standardmäßig wird hierfür die Syntaxdatei `SYSSDF.SDF.045` aktiviert. Die Systembetreuung kann mit den Dienstprogrammen `SDF-A` bzw. `SDF-U` Modifizierungen (z.B. Funktionseinschränkungen) vornehmen. Modifizierungen gelten für alle Benutzer des Systems.

Ein Wechsel der Basis-Systemsyntaxdatei während des Systemlaufs wirkt sofort für alle bestehenden und künftigen Tasks.

### *Subsystem-Syntaxdatei*

Neben der Basis-Systemsyntaxdatei können mehrere Subsystem-Syntaxdateien aktiviert sein. Eine Subsystem-Syntaxdatei enthält die Definitionen von Kommandos und Programmen, die zu einem von DSSM verwalteten Subsystem oder auch zu einer beliebigen Installation-Unit (incl. `BS2CP`) gehören. Sie kann mit `SDF-A` modifiziert werden, wobei Modifikationen, wie z.B. Funktionseinschränkungen, für alle Benutzer des Systems gelten.

Die Aktivierung einer Subsystem-Syntaxdatei während des Systemlaufs wirkt sofort für alle bestehenden und künftigen Tasks.

Eine Subsystem-Syntaxdatei kann auf zwei Arten aktiviert werden:

- Der Name der Subsystem-Syntaxdatei wurde bei Deklaration des Subsystems festgelegt. Beim Laden des Subsystems wird die Subsystem-Syntaxdatei automatisch aktiviert bzw. beim Entladen wieder deaktiviert.
- Der Name der Subsystem-Syntaxdatei ist in der SDF-Parameterdatei eingetragen. Bei der Systemeinführung wird sie automatisch aktiviert. Ist der Pubset, auf dem sie abgelegt ist, bei der Systemeinführung nicht verfügbar, kann sie erst durch das Importieren des Pubsets aktiviert werden.

Eine Änderung (Deaktivierung bzw. Austausch) kann nur über die SDF-Parameterdatei erfolgen. Die Bereitstellung der Subsystem-Syntaxdatei ist hier unabhängig von der Verfügbarkeit des zugehörigen Subsystems.

Ist eine Kommando- bzw. Anweisungsdefinition in mehreren aktivierten Subsystem-Syntaxdateien definiert, so sind folgende Fälle zu unterscheiden:

- Von dem Subsystem kann immer nur eine Version aktiviert sein. In diesem Fall wird die Kommando- bzw. Anweisungsdefinition benutzt, die in der Syntaxdatei der zuletzt aktivierten Version des Subsystems enthalten ist.
- Von dem Subsystem können mehrere Versionen aktiviert sein (Koexistenz). Dabei sind folgende Fälle zu unterscheiden:
  - Für Tasks, die einer bestimmten Subsystem-Version zugeordnet sind, wird die Kommando- bzw. Anweisungsdefinition aus der Syntaxdatei der entsprechenden Subsystem-Version benutzt.  
Die Syntexanalyse eines START-<utility>-Kommandos erfolgt immer vor dem Laden des Programms, d.h. vor der Verbindung der Task mit einer Subsystem-Version des aufgerufenen Programms. Deshalb wird dafür die Syntaxdatei der zuerst aktivierten Version des Subsystems ausgewertet.
  - Für Tasks, die keiner Subsystem-Version zugeordnet sind, wird die Kommando- bzw. Anweisungsdefinition benutzt, die in der Syntaxdatei der zuerst aktivierten Version des Subsystems enthalten ist.

## 6.2 Gruppensyntaxdatei

Eine Gruppensyntaxdatei kann, muss aber nicht aktiviert sein. Die Systembetreuung ordnet sie der Benutzererkennung zu. Die Zuordnung erfolgt über den Benutzereintrag. Im Ausgabefeld *PROFILE-ID* des Kommandos SHOW-USER-ATTRIBUTES ist der Name einer PROFILE-ID eingetragen. Über diese PROFILE-ID kann einer Benutzererkennung oder auch einer Gruppe von Benutzerkennungen eine Gruppensyntaxdatei zugeordnet werden. Die Zuordnung von PROFILE-ID und Gruppensyntaxdatei wird in der SDF-Parameterdatei verwaltet. Die Einträge nimmt die Systembetreuung mit dem Kommando MODIFY-SDF-PARAMETERS vor. Wenn eine derartige Zuordnung besteht, wird die Gruppensyntaxdatei nach LOGON-Verarbeitung automatisch aktiviert. Sie bleibt aktiviert bis zum Taskende. Je Task kann immer nur eine Gruppensyntaxdatei aktiviert sein. Die Veränderung der Zuordnung wirkt nur für zukünftige Tasks.

Eine Gruppensyntaxdatei kann Erweiterungen, Einschränkungen und sonstige Änderungen gegenüber der Systemsyntaxdatei enthalten. Die Systembetreuung hat dadurch die Möglichkeit, den Funktionsvorrat gezielt für einzelne Benutzer zu modifizieren. Hat die Systembetreuung den von Fujitsu Siemens Computers angebotenen Funktionsumfang in der Systemsyntaxdatei systemweit eingeschränkt, so kann sie über eine Gruppensyntaxdatei diese Einschränkung gezielt für eine Benutzererkennung aufheben. Andererseits kann sie systemweit angebotene Funktionen über eine Gruppensyntaxdatei gezielt für eine Benutzererkennung einschränken.

Gruppensyntaxdateien sind mit dem Softwareprodukt SDF-A selbst zu erstellen, können aber auch im Rahmen bestimmter Softwareprodukte geliefert werden.

Werden mit einer neuen Produktversion auch neue oder modifizierte Kommandos bzw. Anweisungen ausgeliefert, müssen alle Gruppensyntaxdateien, die diese Kommando- bzw. Anweisungsdefinitionen enthalten, ersetzt bzw. entsprechend modifiziert werden.

## 6.3 Benutzersyntaxdatei

Eine Benutzersyntaxdatei kann, muss aber nicht aktiviert sein. Für eine unter der Kennung <userid> gestartete Task wird die Benutzersyntaxdatei, die unter dem Namen \$<userid>.SDF.USER.SYNTAX katalogisiert ist, automatisch nach der Verarbeitung des Kommandos SET-LOGON-PARAMETERS aktiviert.

Während der Task kann der Benutzer eine bzw. mehrere Benutzersyntaxdateien mit dem Kommando MODIFY-SDF-OPTIONS aktivieren bzw. deaktivieren. Eine Benutzersyntaxdatei kann Erweiterungen, Einschränkungen und sonstige Änderungen gegenüber den Systemsyntaxdateien und ggf. gegenüber der Gruppensyntaxdatei enthalten. Mögliche Erweiterungen und sonstige funktionelle Änderungen sind beschränkt auf Anweisungen an Programme und auf Kommandos, die durch Prozeduren implementiert sind. Funktionelle Einschränkungen, die in einer System- oder Gruppensyntaxdatei für BS2000-Kommandos (durch Systemmodule implementiert) festgelegt sind, können in einer Benutzersyntaxdatei nicht aufgehoben werden.

Benutzersyntaxdateien sind mit dem Softwareprodukt SDF-A selbst zu erstellen, können aber auch im Rahmen bestimmter Softwareprodukte geliefert werden.

Werden mit einer neuen Produktversion auch neue oder modifizierte Kommandos bzw. Anweisungen ausgeliefert, müssen alle Benutzersyntaxdateien, die diese Kommando- bzw. Anweisungsdefinitionen enthalten, ersetzt bzw. entsprechend modifiziert werden.



## 6.4 Privilegien in Syntaxdateien

Ab BS2000/OSD-BC V1.0 werden Privilegien in Syntaxdateien ausgewertet. Privilegien können vergeben werden an Anwendungsbereiche, Programme, Kommandos, Anweisungen, Operanden und Operandenwerte. Der Benutzer hat nur Zugriff auf Objekte einer Syntaxdatei, die mit dem Privileg definiert wurden, das auch für seine Benutzerkennung vergeben wurde. Der Funktionsumfang kann somit innerhalb einer Syntaxdatei für Benutzer abhängig von ihrem Privileg eingeschränkt werden. Die Funktionseinschränkung durch eine spezielle Gruppensyntaxdatei ist in diesem Fall nicht mehr nötig.

Standardmäßig ist allen Benutzerkennungen mit Ausnahme der Systemkennungen das Privileg STD-PROCESSING zugeordnet. Mit Einsatz des kostenpflichtigen Software-Produkts SECOS können einer nicht-privilegierten Benutzerkennung durch Zuordnung eines Privilegs privilegierte Funktionen eingeräumt werden.

Das Privilegienkonzept im BS2000 ist in dem Handbuch „SECOS“ [10] beschrieben. Die Vergabe von Privilegien in Syntaxdateien ist in dem Handbuch „SDF-A“ [4] beschrieben.

## 6.5 Dateihierarchie

Welche Kommando- und Anweisungseingaben möglich sind, hängt vom Inhalt der zugewiesenen Syntaxdateien ab. Die aktivierten Syntaxdateien werden in der Reihenfolge Benutzersyntaxdatei → Gruppensyntaxdatei → Systemsyntaxdatei durchsucht.

Die in einer Benutzersyntaxdatei enthaltenen Definitionen haben Vorrang vor denen in der Gruppen- und Systemsyntaxdatei. Ist eine Definition in mehreren Benutzersyntaxdateien enthalten, so hat die Definition in der zuletzt aktivierten Benutzersyntaxdatei Vorrang.

Die in den Benutzersyntaxdateien enthaltenen Definitionen der BS2000-Kommandos (durch Systemmodule implementiert) müssen voll durch die Gruppen- oder Systemsyntaxdatei abgedeckt sein. Ist ein solches Kommando in allen drei Dateien definiert, so muss die Definition in der Benutzersyntaxdatei durch die Gruppensyntaxdatei voll abgedeckt sein.

Die in der Gruppensyntaxdatei enthaltenen Definitionen haben Vorrang vor denen in der Systemsyntaxdatei. In der Gruppensyntaxdatei nicht enthaltene Definitionen werden der Systemsyntaxdatei entnommen. Wurde die Dateihierarchie bei der Zuordnung einer Gruppensyntaxdatei außer Kraft gesetzt, so bestimmt sich der zur Verfügung stehende Kommando- und Anweisungsvorrat nur aus der Gruppensyntaxdatei.

Ist eine Definition in mehreren System- bzw. Subsystem-Syntaxdateien enthalten, so hat die Definition in der zuletzt aktivierten Subsystem-Syntaxdatei Vorrang.

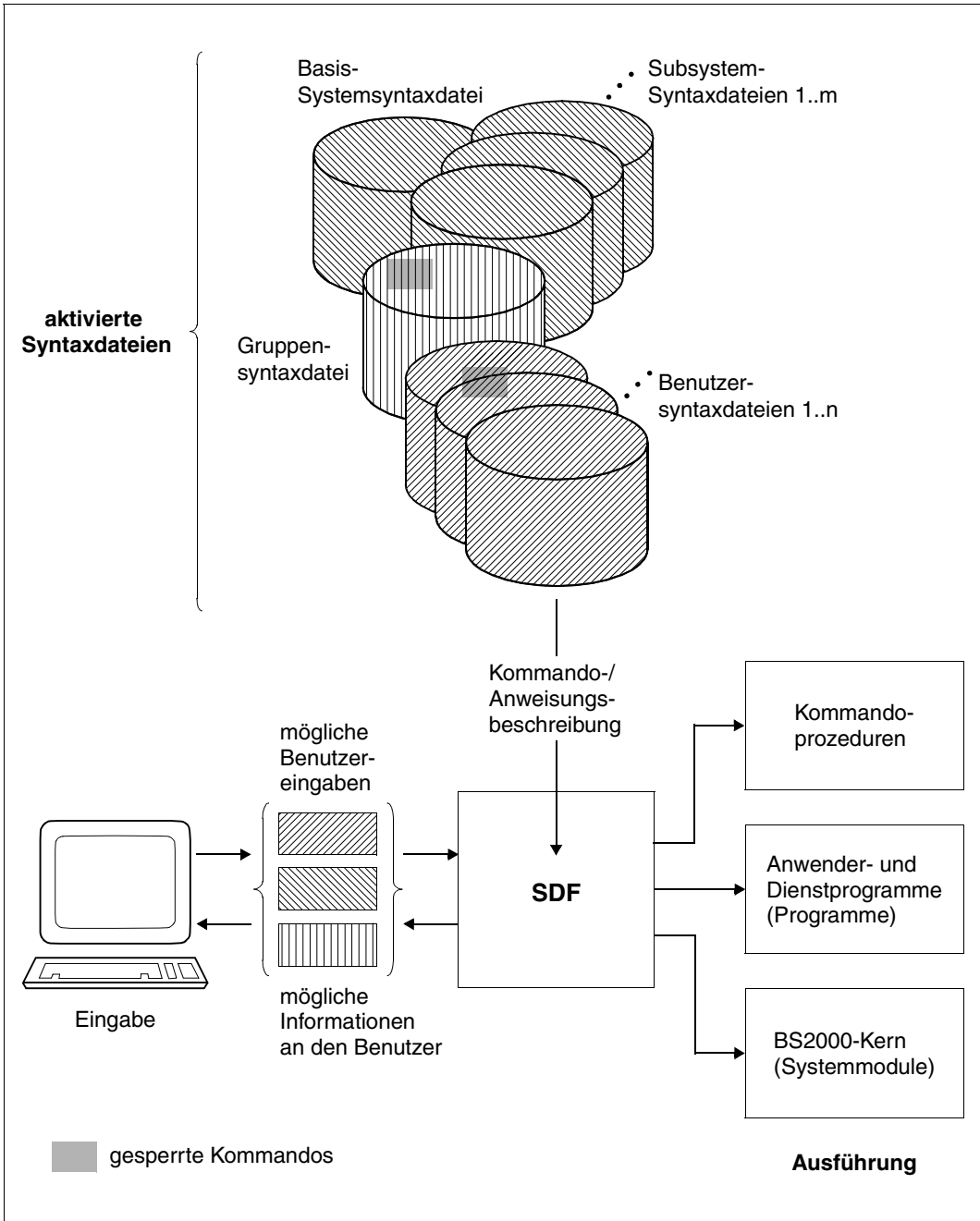


Bild 50: Arbeitsweise von SDF bei Aktivierung verschiedener Syntaxdateitypen

## 6.6 Bearbeiten von Syntaxdateien

Die Bearbeitung einer Syntaxdatei (erstellen und modifizieren) kann nur mit den Dienstprogrammen SDF-A, SDF-U oder SDF-I erfolgen. Syntaxdateien werden hierbei nur auf logischer Ebene bearbeitet. Der interne Dateiaufbau bleibt für den Benutzer unsichtbar. Deshalb kann eine Bearbeitung mit anderen Mitteln zu Fehlermeldungen beim Ansprechen von manipulierten Kommando- bzw. Anweisungsdefinitionen führen.

Mit dem Softwareprodukt SDF-A lassen sich neue Syntaxdateien erstellen und bereits vorhandene Syntaxdateien ändern.

SDF-A bietet folgende Leistungen:

- Definieren eigener Kommandos, die durch Kommandoprozeduren implementiert sind, und Ändern dieser Definitionen
- Definieren eigener Programme und der Anweisungen an diese sowie Ändern dieser Definitionen
- Sperren von Anwendungsbereichen, Kommandos, Programmen, Anweisungen, Operanden und Operandenwerten
- Vergabe von Privilegien für Anwendungsbereiche, Kommandos, Programme, Anweisungen, Operanden und Operandenwerte
- Ändern von Standardwerten
- Umbenennen von Anwendungsbereichen, Kommando-, Anweisungs- und Operandennamen sowie Operandenwerten, die Schlüsselworte sind
- Ändern oder Austauschen von Hilfetexten für den geführten Dialog, z.B. Ersetzen von deutschen Hilfetexten durch solche in der Landessprache
- Ändern der Kommando- und Anweisungsreihenfolge zu Anwendungsbereichen und Definieren neuer Anwendungsbereiche
- Ändern der Voreinstellungen der SDF-Parameter, z.B. für den geführten Dialog
- Ausgabe der Inhalte von Syntaxdateien

Die mit SDF-A vorgenommene Bearbeitung einer Syntaxdatei wird wirksam, sobald die Datei aktiviert wird.

Mit dem Dienstprogramm SDF-U kann die Systembetreuung Syntaxdateien verwalten. SDF-U bietet eine kleine Untermenge der Funktionen von SDF-A:

- Ändern der Voreinstellungen der SDF-Optionen, z.B. für den geführten Dialog
- Ändern der Definitionen von Kommandos, die durch Prozeduren implementiert sind
- Ausgabe der Inhalte von Syntaxdateien
- Kopieren von Syntaxdatei-Inhalten
- Löschen von Objekten der Syntaxdateien

Mit dem Dienstprogramm SDF-I kann eine System- oder Gruppensyntaxdatei aus mehreren Liefereinheits-Syntaxdateien erstellt werden. Die Bearbeitung erfolgt nur für Syntaxdateien gleichen Typs.

SDF-I bietet folgende Funktionen:

- Ausgeben von Informationen aus einer Syntaxdatei. Angezeigt werden:
  - Typ der Syntaxdatei
  - Format der Syntaxdatei
  - Voreinstellungen der SDF-Optionen (Globalinformation)
  - Version (Globalinformation)
  - enthaltene Kommandos und Produktnamen der Liefereinheiten

Die SDF-I-Anweisung SHOW-SYNTAX-FILE ist im Abschnitt SDF-I-Anweisungen (siehe [Seite 235](#)) beschrieben.

- Konvertieren einer Syntaxdatei in das neue Format. Mit Einsatz von SDF V2.0A wurde der interne logische Aufbau von Syntaxdateien geändert. SDF unterstützt ab V2.0A sowohl das alte als auch das neue Format, das alte Format jedoch mit verringerter Performance. Syntaxdateien, die mit einer SDF-A-Version kleiner V2.0 erstellt wurden, besitzen das alte Format und können mit der SDF-I-Anweisung CONVERT-SYNTAX-FILE auf das neue Format umgestellt werden. Die Anweisung ist im Abschnitt SDF-I-Anweisungen (siehe [Seite 235](#)) beschrieben.
- Mischen einer Syntaxdatei aus einzelnen Softwareeinheits-Syntaxdateien. Hiermit kann eine Syntaxdatei neu erstellt werden. Die Erweiterung einer Syntaxdatei um eine Softwareeinheits-Syntaxdatei für ein neu einzusetzendes Softwareprodukt ist ebenfalls möglich.
- Entmischen einer Syntaxdatei. Diese Funktion ermöglicht es, für eine neue Version eines bereits eingesetzten Softwareproduktes die Softwareeinheits-Syntaxdatei in die bestehende Syntaxdatei zu übernehmen. Hierbei werden alle Bestandteile der Syntaxdatei entmischt, die durch die ältere Softwareeinheits-Syntaxdatei eingemischt wurden.

---

## 7 Benutzerschnittstelle SDF

Nachfolgend werden alle SDF-spezifischen Kommandos, Standardanweisungen und Makroaufrufe aufgelistet, die dem Benutzer zur Verfügung stehen. Er kann damit SDF-Einstellungen taskspezifisch vornehmen, seine Benutzer-Syntaxdateien verwalten und SDF-Leistungen für eigene Benutzerprogramme nutzen.

### 7.1 Kommandos aus dem Anwendungsbereich SDF

Alle Kommandos, die der Benutzer zur Steuerung bzw. Verwaltung der taskspezifischen SDF-Schnittstelle benötigt, sind in dem Anwendungsbereich SDF enthalten (siehe auch Handbücher „Kommandos, Band 1 - 5“ [1]). Nachfolgend werden beschrieben:

- HELP-SDF
- MODIFY-SDF-OPTIONS
- REMARK
- RESET-INPUT-DEFAULTS
- RESTORE-SDF-INPUT
- SHOW-CMD
- SHOW-INPUT-DEFAULTS
- SHOW-INPUT-HISTORY
- SHOW-SDF-OPTIONS
- SHOW-SYNTAX-VERSIONS
- WRITE-TEXT

Die Kommandobeschreibung enthält den Kommandonamen mit kurzem Hilfetext, den Anwendungsbereich, die Privilegien, mit denen das Kommando ausgeführt werden kann, die Funktionsbeschreibung, die Syntaxdarstellung, die Operandenbeschreibung, Kommando-Returncodes, Darstellung von Ausgaben nach SYSOUT bzw. in S-Variablen (optional für SHOW-Kommandos), Beispiele und Hinweise.

Die Kommandos zur systemweiten Steuerung und Verwaltung von SDF sind der Systembetreuung vorbehalten und werden in den Handbüchern „Kommandos, Band 1 -5“ [1] und „SDF-Verwaltung“ [5] beschrieben.

## HELP-SDF

### Informationen über SDF ausgeben

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

### Funktionsbeschreibung

Das Kommando HELP-SDF gibt Informationen über SDF aus.

Die Gesamtinformation ist in Abschnitte unterteilt, die mit den einzelnen Operanden des Kommandos angesprochen werden können.

Gibt der Benutzer HELP-SDF ohne Operanden bzw. nur mit den voreingestellten Operandenwerten ein, so erhält er eine kurze Einführung zu SDF (Beschreibung siehe „HELP-SDF mit voreingestellten Operandenwerten“, [Seite 169](#)).

Gibt der Benutzer /HELP-SDF? ein, so erscheint der Operandenfragebogen des Kommandos (temporär geführter Dialog) und er kann die gewünschten Informationen auswählen.

## Format

HELP-SDF	Kurzname: HPSDF
<p><b>GUIDANCE-MODE</b> = <u>*NO</u> / *YES</p> <p><b>,SDF-COMMANDS</b> = <u>*NO</u> / *YES</p> <p><b>,ABBREVIATION-RULES</b> = <u>*NO</u> / *YES</p> <p><b>,GUIDED-DIALOG</b> = <u>*YES</u> (...)</p> <p>    <u>*YES</u>(...)</p> <p>          <b>SCREEN-STEPS</b> = <u>*NO</u> / *YES</p> <p>          <b>,SPECIAL-FUNCTIONS</b> = <u>*NO</u> / *YES</p> <p>          <b>,FUNCTION-KEYS</b> = <u>*NO</u> / *YES</p> <p>          <b>,NEXT-FIELD</b> = <u>*NO</u> / *YES</p> <p><b>,UNGUIDED-DIALOG</b> = <u>*YES</u> (...) / *NO</p> <p>    <u>*YES</u>(...)</p> <p>          <b>SPECIAL-FUNCTIONS</b> = <u>*NO</u> / *YES</p> <p>          <b>,FUNCTION-KEYS</b> = <u>*NO</u> / *YES</p>	

## Operandenbeschreibung

### **GUIDANCE-MODE** = \*NO / \*YES

Arten der Dialogführung (Ausgabe siehe [Seite 170](#)).

### **SDF-COMMANDS** = \*NO / \*YES

SDF-spezifische Kommandos (Ausgabe siehe [Seite 171](#)).

### **ABBREVIATION-RULES** = \*NO / \*YES

Abkürzungsregeln für die Kommandosyntax (Ausgabe siehe [Seite 172](#)).

### **GUIDED-DIALOG** = \*YES(...) / \*NO

Anwendung des geführten Dialogs (Ausgabe siehe [Seite 173](#)).

### **GUIDED-DIALOG** = \*YES(...)

Die Beschreibung des geführten Dialogs ist in 4 Informationsblöcke unterteilt, die einzeln angefordert werden können:

#### **SCREEN-STEPS** = \*NO / \*YES

Reihenfolge und Inhalt der Menüs (Anwendungsbereichs-Menü, Kommando-Menü, Operandenfragebogen, Unterfragebogen; Ausgabe siehe [Seite 173](#)).

#### **SPECIAL-FUNCTIONS** = \*NO / \*YES

Spezialfunktionen im Menü (Ausgabe siehe [Seite 173](#)).

**FUNCTION-KEYS = \*NO / \*YES**

Funktionstasten, die das Menü steuern (Ausgabe siehe [Seite 174](#)).

**NEXT-FIELD = \*NO / \*YES**

Eingabe in die NEXT-Zeile, die das Menü steuern (Ausgabe siehe [Seite 176](#)).

**UNGUIDED-DIALOG = \*NO / \*YES(...)**

Anwendung des ungeführten Dialogs (Ausgabe siehe [Seite 177](#)).

**UNGUIDED-DIALOG = \*YES(...)**

Die Beschreibung des ungeführten Dialogs ist in 2 Informationsblöcke unterteilt, die einzeln angefordert werden können:

**SPECIAL-FUNCTIONS = \*NO / \*YES**

Spezialfunktionen im Menü des ungeführten Dialogs (Ausgabe siehe [Seite 177](#)).

**FUNCTION-KEYS = \*NO / \*YES**

Funktionstasten, die das Menü des ungeführten Dialogs steuern (Ausgabe siehe [Seite 178](#)).

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ausgeführt
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft
1	64	CMD0810	Fehler während der Ausführung
			garantierte Meldungen: CMD0500, CMD0810



## Informationen, die mit HELP-SDF ausgegeben werden:

### *HELP-SDF mit den voreingestellten Operandenwerten*

```
%  
% Einfuehrung:  
%  
% SDF ist ein komfortabler Kommando-Interpreter und Dialog-  
% Manager. Sie koennen folgende Vorteile von SDF nutzen:  
%  
% - gefuehrter Dialog  
% - Abkuerzbarkeit  
% - Blockeingabe  
% - Kommandogedaechtnis (History)  
% - Definieren task-spezifischer Defaultwerte  
%  
% Ausserdem wurde eine verstaendlichere Kommando-Sprache  
% entwickelt:  
% Namen von Kommandos, Operanden und Schlueselwortwerten  
% sind so gewaehlt, dass sie die Funktion oder den Zweck  
% erkennen lassen. Analoge Sachverhalte (z.B. der Dateiname)  
% sind gleich benannt(z.B. FILE-NAME=, FROM-FILE= oder TO-FILE=).  
% Bei variablen Operandenwerten beschreiben Grunddatentypen  
% den zulaessigen Wertebereich.  
% Fehlerhafte Eingaben werden mit einer Meldung zur Korrektur  
% angeboten.  
% Anwender der alten Kommandosprache ISP erhalten mit der  
% Eingabe "<altes-Kdo>?" den Operandenfragebogen in der neuen  
% Syntax bzw. ein Auswahlmueue der entsprechenden SDF-Kommandos.  
%  
% Fuer weitere Informationen geben Sie bitte "HELP-SDF?" ein,  
% Sie erhalten dann den Operandenfragebogen dieses Kdos. Bei den  
% Operanden, die Sie auf "YES" setzen , erhalten Sie weitere  
% Informationen.  
% Im Handbuch "Master Index" finden Sie Verweise auf Handbuecher,  
% die die Beschreibung der einzelnen Kommandos enthalten.  
%
```

*HELP-SDF mit GUIDANCE-MODE=\*YES*

```

%
% Arten der Dialogfuehrung :
%
% 1. Gefuehrter Dialog
% Zur Eingabe im gefuehrten Dialog werden dem Benutzer Menues
% in folgender Reihenfolge angeboten :
% Anwendungsbereichs-Menue, Kommando-Menue, Operandenfragebogen
% und Operanden-Unterfragebogen.
% Der Umfang der im Menue angebotenen Information kann ueber
% den Operanden GUIDANCE des Kommandos MODIFY-SDF-OPTIONS
% gesteuert werden :
% - GUIDANCE=*MAXIMUM  saemtliche Operandenwerte mit Zusaetzen
%                       (Wertebereich, etc.); Hilfetext fuer
%                       Kommandos und Operanden
%
% - GUIDANCE=*MEDIUM   saemtliche Operandenwerte ohne Zusaetze;
%                       Hilfetext nur fuer Kommandos
% - GUIDANCE=*MINIMUM  nur Defaultwerte der Operanden; keine
%                       Zusaetze; keine Hilfetexte
%
% 2. Ungefuehrter Dialog
% Im ungefuehrten Dialog bietet keine Menues. Es gibt 2 Formen des
% ungefuehrten Dialogs, die ueber den Operanden GUIDANCE
% des Kommandos
% MODIFY-SDF-OPTIONS gesteuert werden:
% - GUIDANCE=*NO
%   das System fordert mit "%CMD:" zur Kommandoeingabe
%   auf; Syntaxfehler-Dialog zur Korrektur
%   fehlerhafter Eingaben ohne Wiederholung des
%   gesamten Kommandos, detaillierte Fehlermeldungen;
%   geblockte Kommandoeingabe: mehrere Kommandos,
%   die durch das logische Zeilenendezeichen (LZE)
%   getrennt sind, koennen gleichzeitig abgeschickt werden
% - GUIDANCE=*EXPERT
%   das System fordert mit "/" zur Kommandoeingabe auf;
%   kein Syntaxfehler-Dialog,detaillierte
%   Fehlermeldungen; geblockte Kommandoeingabe moeglich.
%

```

*HELP-SDF mit SDF-COMMANDS=\*YES*

```
%  
% SDF-spezifische Kommandos :  
%  
% - HELP-SDF  
%   Gibt Informationen ueber SDF aus.  
% - MODIFY-SDF-OPTIONS  
%   Aendert die task-spezifischen SDF-Optionen.  
% - REMARK  
%   Kennzeichnet Bemerkungen in einer Kommandodatei.  
% - RESET-INPUT-DEFAULTS  
%   Loescht task-spezifische Defaultwerte.  
% - RESTORE-SDF-INPUT  
%   Gibt eine vorangegangene Eingabe erneut aus.  
% - SHOW-CMD  
%   Gibt die Syntaxbeschreibung eines Kommandos nach SYSOUT/SYSL  
%   SYSLST aus.  
% - SHOW-SDF-OPTIONS  
%   Gibt Informationen ueber die Einstellung der task-lokalen  
%   SDF-Optionen aus.  
% - SHOW-INPUT-DEFAULTS  
%   Gibt die task-spezifische Defaultwerte aus.  
% - SHOW-INPUT-HISTORY  
%   Gibt eine Liste der letzten Eingaben aus.  
% - SHOW-SYNTAX-VERSIONS  
%   Gibt Informationen ueber den Inhalt der aktivierten  
%   Syntaxdateien aus.  
%  
% - WRITE-TEXT  
%   Gibt Text nach SYSOUT/SYSLST aus.  
%
```

*HELP-SDF mit ABBREVIATION-RULES=\*YES*

```

%
% Abkuerzungsregeln fuer die Kommandosyntax
%
% - Kommandonamen, Operandennamen und konstante Operandenwerte
% (keine Grunddatentypen) koennen folgendermassen abgekuerzt
% werden:
% - von rechts nach links Zeichen weglassen
%   z.B: TRANSFER-FILE abgekuerzt in TRANS
% - innerhalb von Teilfolgen von rechts nach links Zeichen
%   weglassen
%   z.B: TRANSFER-FILE abgekuerzt in T-F
%   Einschraenkung: die Abkuerzungen muessen eindeutig sein,
%   z.B: CREATE als Abkuerzung ist mehrdeutig unter CREATE-FILE
%       CREATE-FILE-GENERATION, CREATE-FILE-GROUP und CREATE-JV
%
% - Operandenwerte duerfen auch ohne den zugehoerigen
%   Operandennamen angegeben werden (Stellungsoperanden).
%   Dabei ist zu beachten:
%   - Fuer jeden weggelassenen Operanden vor einem
%     Stellungsoperanden ist ein Komma einzugeben.
%   - Wird in der Operandenfolge eines Kommandos ein Operand
%     vollstaendig (Operandenname und Operandenwert) angegeben, so
%     duerfen nachfolgende Operanden nicht mehr als
%     Stellungsoperanden angegeben werden.
%
% - Gefuehrter Dialog:
%   - Operanden einer Struktur muessen nicht im
%     Unterfragebogen, sondern koennen auch im Operandenfragebogen
%     eingegeben werden. Sie sind in Klammern einzuschliessen und
%     muessen direkt hinter dem struktureinleitenden
%     Operandenwert stehen.
%   - In der NEXT-Zeile des Menues koennen auch komplette
%     Kommandos eingegeben werden. Endet ein Kommandoname mit
%     Fragezeichen (<command>?), so erscheint der
%     Operandenfragebogen des angegebenen Kommandos.
%     Er enthaelt die bereits angegebenen Operandenwerte.
%
% - Auswahlmenue bei mehrdeutiger Eingabe:
%   Enthaeilt der Kommandoname Musterzeichen und endet mit
%   Fragezeichen, gibt SDF ein Auswahlmenue aus mit allen,
%   Kommandos, die der Musterzeichenfolge entsprechen.
%   Ist der Kommandoname eine mehrdeutige Abkuerzung und endet mit
%   Fragezeichen, gibt SDF ein Auswahlmenue mit den entsprechenden
%   Kommandos aus.

```

**HELP-SDF mit GUIDED-DIALOG=\*YES(SCREEN-STEPS=\*YES)**

```

%
% Gefuehrter Dialog: Reihenfolge und Inhalt der Menues
%
% Anwendungsbereichs-Menue:
%   Listet alle Anwendungsbereiche auf (Kommandos sind
%   entsprechend ihrer Funktion einzelnen Anwendungsbereichen
%   zugeordnet). Ein Anwendungsbereich kann ueber Nummer oder
%   Name ausgewaehlt werden. Der Anwendungsbereich
%   *ALL-COMMANDS listet alle Kommandos auf.
%
% Kommando-Menue:
%   Listet alle Kommandos des ausgewaehlten Anwendungsbereichs
%   auf. Ein Kommando kann ueber Nummer oder Name ausgewaehlt
%   werden. Ein Kommando mit nachfolgenden Ausrufezeichen
%   besitzt keine Operanden.
%
% Operandenfragebogen:
%   Listet die Operanden des ausgewaehlten Kommandos auf. Die
%   gewuenschten Operandenwerte sind anzugeben.
%
% Unterfragebogen:
%   Listet die Operanden einer Struktur auf. Bei Angabe eines
%   struktureinleitenden Operandenwertes im Operandenfragebogen
%   wird fuer die zugehoerige Struktur ein Unterfragebogen
%   ausgegeben.
%
% Kommando-Menue:
%   Wurde ein Kommando ausgefuehrt oder abgebrochen, so
%   erscheint wieder das aktuelle Kommando-Menue.
%

```

**HELP-SDF mit GUIDED-DIALOG=\*YES(SPECIAL-FUNCTIONS=\*YES)**

```

%
% Gefuehrter Dialog : Spezialfunktionen
%
% ? als Operandenwert:
%   Liefert Hilfetext und Informationen ueber moegliche
%   Operandenwerte (Wertebereiche, usw.); bei fehlerhafter
%   Eingabe zusaetzlich detaillierte Fehlermeldungen.
% ?? als Operandenwert:
%   Liefert Hilfetexte, eine Uebersicht der moeglichen
%   Operandenwerte und einen Hilfetext mit Informationen
%   ueber Datentypen , die dem Operanden zugeordnet sind.
% ! als Operandenwert:
%   Setzt fuer diesen Operanden den Defaultwert ein.

```

```

% ( hinter einem struktureinleitenden Operandenwert:
%     Gibt den Unterfragebogen fuer die Struktur aus, die
%     diesem Operandenwert zugeordnet ist.
% () hinter einem struktureinleitenden Operandenwert:
%     Unterdrueckt den Unterfragebogen und setzt fuer die
%     Operanden der Struktur die jeweiligen Defaultwerte ein.
% - als letztes Zeichen in einer Eingabezeile:
%     Eine Fortsetzungszeile wird ausgegeben
% LZF-Taste:
%     Loescht ab der Schreibmarke alle Zeichen der Eingabezeile

```

***HELP-SDF mit GUIDED-DIALOG=\*YES(FUNCTION-KEYS=\*YES)***

```

%
% Gefuehrter Dialog: Funktionstasten
%
% Die Wirkung der Funktionstasten haengt ab von Einstellung der
% SDF-Option FUNCTION-KEYS (Kommando/Standardanweisung
% MODIFY-SDF-OPTIONS).
% Es sind zwei Einstellungen moeglich: der *OLD-MODE und der
% *STYLE-GUIDE-MODE der mehr Funktionalitaet bietet.
%
% *OLD-MODE
%
% K1  EXIT-Funktion
%     Entspricht *EXIT in der NEXT-Zeile.
%     Entspricht F3 im *STYLE-GUIDE-MODE (siehe dort).
% K2  Unterbrechungs-Funktion
%     Unterbricht ein aktuelles Programm oder eine Prozedur, oder
%     bricht die Ausgabe eines Kommandos ab.
% K3  Refresh-Funktion
%     Wiederholt das zuletzt ausgegebene Menue.
%     Entspricht *REFRESH in der NEXT-Zeile.
%     Entspricht F5 im *STYLE-GUIDE-MODE (siehe dort).
% F1  EXIT-ALL-Funktion
%     Entspricht *EXIT-ALL in der NEXT-Zeile.
%     Entspricht F6 im *STYLE-GUIDE-MODE (siehe dort).
% F2  Test-Funktion
%     Prueft Eingaben auf Syntaxfehler.
%     Entspricht *TEST in der NEXT-Zeile.
% F3  Execute-Funktion
%     Fuehrt die aktuelle Operation aus.
%     Entspricht *EXECUTE in der NEXT-Zeile.
%     Entspricht F11 im *STYLE-GUIDE-MODE (siehe dort).
%
% *STYLE-GUIDE-MODE
%
% K2  Unterbrechungs-Funktion

```

```
%      Unterbricht ein aktuelles Programm oder eine Prozedur,  
%      oder bricht die Ausgabe eines Kommandos ab.  
% F1   Hilfe-Funktion  
%      Wechselt fuer die aktuelle Eingabe in die naechsthoehere  
%      Fuehrungsstufe.  
%      Entspricht "?" in der NEXT-Zeile.  
% F3   EXIT-Funktion  
%      Bricht das aktuelle Menue oder Fragebogen ab und wechselt  
%      in das uebergeordnete Menue.  
%      Entspricht *EXIT in der NEXT-Zeile.  
%      Entspricht K1 im *OLD-MODE.  
% F5   Refresh-Funktion  
%      Wiederholt das zuletzt ausgegebene Menue.  
%      Entspricht *REFRESH in der NEXT-Zeile.  
%      Entspricht K3 im *OLD-MODE.  
% F6   EXIT-ALL-Funktion  
%      Bricht das aktuelle Menue ab und wechselt im gefuehrten  
%      hoechsten Menuestufe oder beendet den temporaer  
%      gefuehrten Dialog.  
%      Entspricht *EXIT-ALL in der NEXT-Zeile.  
%      Entspricht F1 im *OLD-MODE.  
% F7   Rueckwaerts Blaettern  
%      Blaettert im Menue oder Fragebogen rueckwaerts.  
%      Entspricht "-" in der NEXT-Zeile.  
% F8   Vorwaerts Blaettern  
%      Blaettert im Menue oder Fragebogen vorwaerts.  
%      Entspricht "+" in der NEXT-Zeile.  
% F9   RESTORE-SDF-INPUT  
%      Gibt das zuletzt eingegebene Kommando bzw. Anweisung  
%      erneut aus.  
%      Entspricht RESTORE-SDF-INPUT in der NEXT-Zeile.  
% F11  Execute-Funktion  
%      Fuehrt die aktuelle Operation aus.  
%      Entspricht *EXECUTE in der NEXT-Zeile.  
%      Entspricht F3 im *OLD-MODE.  
% F12  Cancel-Funktion  
%      Bricht das aktuelle Menue bzw. Fragebogen ab und wechselt  
%      in das uebergeordnete Menue bzw. Fragebogen.  
%      Entspricht *EXIT in der NEXT-Zeile.  
%
```

*HELP-SDF mit GUIDED-DIALOG=\*YES(NEXT-FIELD=\*YES)*

```

%
% Gefuehrter Dialog: NEXT-Zeile
%
% +,++,--,-- : Blaettert im Menue vor und zurueck.
%              "+" bzw. "-" entsprechen F8 bzw. F9 im
%              *STYLE-GUIDE-MODE.
% *EXECUTE   : Fuehrt das aktuelle Kommando aus.
%              Entspricht F3 im *OLD-MODE bzw. F11 im
%              *STYLE-GUIDE-MODE.
% *CONTINUE  : Entspricht *EXECUTE oder "+" gemaess dem Wert im
%              aktuellen Fragebogen.
% *TEST      : Prueft Eingaben auf Syntaxfehler.
%              Entspricht F2 im *OLD-MODE.
% *EXIT      : Bricht das aktuelle Menue oder Fragebogen ab und
%              wechselt in das uebergeordnete Menue.
%              in das uebergeordnete Menue.
%              Entspricht K1 im *OLD-MODE bzw. F3 im
%              *STYLE-GUIDE-MODE.
% *EXIT-ALL  : Bricht das aktuelle Menue oder Fragebogen ab und
%              wechselt im gefuehrten Dialog in die hoechste
%              Menue oder beendet den temporaer gefuehrten Dialog.
%              Entspricht F1 im *OLD-MODE bzw. F6 im
%              *STYLE-GUIDE-MODE.
% *REFRESH   : Wiederholt das zuletzt ausgegebene Menue.
%              Entspricht K3 im *OLD-MODE bzw. F5 im
%              *STYLE-GUIDE-MODE.
% *DOM-MENU  : Wechselt in das Anwendungsbereichs-Menue.
% (<domain>) : Gibt das Kommando-Menue des Anwendungsbereichs
%              <domain> aus.
% <command>? : Fuehrt die aktuelle Operation aus und gibt
%              anschliessend den Operandenfragebogen des Kommandos
%              <command> aus. Bereits angegebene.
%              Operandenwerte werden in den Fragebogen
%              uebernommen.
% <command>  : Fuehrt die aktuelle Operation aus und daran
%              anschliessend das angegebene Kommando <command>.
% !<command> : Fuehrt die aktuelle Operation aus und definiert die
%              angegebenen Operandenwerte als task-spezifische
%              Defaultwerte des Kommandos. In Kombination mit
%              <command>? kvnnen die Defaultwerte in den
%              Operandenfragebogen eingegeben werden.
% *DOWN(<operand>) :
%              Gibt den Unterfragebogen fuer den angegebenen
%              Operandenwert <operand> aus.
% *UP        : Wechselt vom Unterfragebogen zurueck in den
%              Operandenfragebogen.

```



```

% ?      : Erhoeht die Fuehrungsstufe fuer die aktuelle
%        Eingabe.
% ??     : Stellt die maximale Fuehrungsstufe fuer die
%        aktuelle Eingabe ein und liefert Informationen
%        ueber Datentypen.
%
%

```

**HELP-SDF mit UNGUIDED-DIALOG=\*YES(SPECIAL-FUNCTIONS=\*YES)**

```

% Ungefuehrter Dialog: Spezialfunktionen
%
% ? als Operandenwert:
%     Liefert Hilfetext und Informationen ueber moegliche
%     Operandenwerte (Wertebereiche, usw.).
% ? als Kommando:
%     Gibt das Anwendungsbereichs-Menue aus
%     (temporaer gefuehrter Dialog)
% ?? als Operandenwert:
%     Liefert Hilfetexte, eine Uebersicht der moeglichen
%     Operandenwerte und einen Hilfetext mit Informationen
%     ueber Datentypen, die dem Operanden zugeordnet sind.
% <command>? als Kommando:
%     Gibt den Operandenfragebogen des Kommandos <command> aus
%     (temporaer gefuehrter Dialog). Bereits angegebene
%     Operandenwerte werden eingeblendet.
%     Existieren keine Operanden, wird der Hilfetext des
%     Kommandos ausgegeben.
% !<command> als Kommando:
%     Definiert die angegebenen Operandenwerte als task-spezifische
%     Defaultwerte des Kommandos. In Kombination mit <command>?
%     kvnnen die Defaultwerte im temporaer gefuehrten Dialog
%     eingegeben werden.
% ^ bzw. *SECRET als Operandenwert (nur bei geheimen Operanden):
%     Gibt ein dunkelgesteuertes Eingabefeld fuer den
%     Operandenwert (z.B. Kennwort) aus.
% LZF-Taste:
%     Loescht ab der Schreibmarke alle Zeichen der Eingabezeile
% LZE-Taste:
%     Ermoeglicht geblockte Eingabe. Es koennen mehrere durch
%     LZE getrennte Kommandos mit DUE abgeschickt und
%     gleichzeitig ausgefuehrt werden.

```

*HELP-SDF mit UNGUIDED-DIALOG=\*YES(FUNCTION-KEY=\*YES)*

```
%  
% Ungefuehrter Dialog: Funktionstasten  
%  
% Die Wirkung der Funktionstasten haengt ab von Einstellung der  
% SDF-Option FUNCTION-KEYS (Kommando/Standardanweisung  
% MODIFY-SDF-OPTIONS).  
% Es sind zwei Einstellungen moeglich: der *OLD-MODE und  
% der *STYLE-GUIDE-MODE der mehr Funktionalitaet bietet.  
%  
%  
% *OLD-MODE  
%  
% K1  EXIT-Funktion  
%      Beendet das aktuell laufende Programm. In einer Meldung  
%      wird der Benutzer gefragt, ob das Programm beendet werden  
%      soll.  
%      Entspricht F3 im *STYLE-GUIDE-MODE.  
% K2  Unterbrechungs-Funktion  
%      Unterbricht ein aktuelles Programm oder eine Prozedur,  
%      oder bricht die Ausgabe eines Kommandos ab.  
% F1  EXIT-ALL-Funktion  
%      Beendet das aktuell laufende Programm. In einer Meldung  
%      wird der Benutzer gefragt, ob das Programm beendet werden  
%      soll.  
%      Entspricht F6 im *STYLE-GUIDE-MODE.  
%  
%  
% *STYLE-GUIDE-MODE  
%  
% K2  Unterbrechungs-Funktion  
%      Unterbricht ein aktuelles Programm oder eine Prozedur,  
%      oder bricht die Ausgabe eines Kommandos ab.  
% F1  Hilfe-Funktion  
%      Wechselt den temporaer gefuehrten Dialog.  
%      Entspricht der Eingabe von "?".  
% F3  EXIT-Funktion  
%      Beendet das aktuell laufende Programm. In einer Meldung  
%      wird der Benutzer gefragt, ob das Programm beendet werden  
%      soll.  
%      Entspricht K1 im *OLD-MODE.  
% F6  EXIT-ALL-Funktion  
%      Beendet das aktuell laufende Programm. In einer Meldung  
%      wird der Benutzer gefragt, ob das Programm beendet werden  
%      soll.  
%      Entspricht F1 im *OLD-MODE.
```

% F9 RESTORE-SDF-INPUT  
% Gibt das zuletzt eingegebene Kommando bzw. Anweisung erneut  
% aus.  
% Entspricht RESTORE-SDF-INPUT.  
% F12 Cancel-Funktion  
% Beendet das aktuell laufende Programm. In einer Meldung  
% wird der Benutzer gefragt, ob das Programm beendet werden  
% soll.  
%

## MODIFY-SDF-OPTIONS

### Benutzersyntaxdatei 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

### Funktionsbeschreibung

Mit dem Kommando MODIFY-SDF-OPTIONS kann der Benutzer

- Benutzersyntaxdateien aktivieren oder deaktivieren. Benutzersyntaxdateien werden mit dem Dienstprogramm SDF-A erstellt, siehe dazu: Handbuch „SDF-A“ [4].

#### *Hinweis*

SDF verwendet ab der Version V2.0A ein *geüderes* Syntaxdatei-Format! Aus Performance-Gründen sollten nur Syntaxdateien im neuen Format verwendet werden. Benutzersyntaxdateien werden entweder bei Bearbeitung mit SDF-A (Version  $\geq$  V2.0A) oder dem Dienstprogramm SDF-I (Anweisung CONVERT-SYNTAX-FILE) in das neue Format konvertiert. Eine Syntaxdatei im neuen Format kann nicht in das alte Format konvertiert werden!

- die Einstellungen der SDF-Optionen task-spezifisch ändern:
  - 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. Aktiviert der Benutzer eine Benutzersyntaxdatei, werden die Einstellungen entsprechend den darin enthaltenen Global-Informationen geändert. Unter dem Dienstprogramm OMNIS wird jedoch bei Beginn der Task und bei einem Wechsel der Benutzersyntaxdatei GUIDANCE=\*EXPERT eingestellt.

Der Operandenfragebogen des Kommandos MODIFY-SDF-OPTIONS enthält im (temporär) geführten Dialog als voreingestellte Werte die aktuellen Einstellungen, die der Benutzer auch mit dem Kommando SHOW-SDF-OPTIONS abfragen kann. Der Operandenwert \*UNCHANGED vereinbart, dass der aktuell eingestellte Wert nicht verändert wird.

In Programmen mit SDF-Schnittstelle ist MODIFY-SDF-OPTIONS mit gleicher Syntax und Funktionalität, jedoch ohne die Operanden MODE, DEFAULT-PROGRAM-NAME und COMMAND-STATISTICS, als Standardanweisung verfügbar.

### *Privilegierte Funktionen*

Die Systembetreuung (Privileg TSOS) kann über den Operanden CMD-STATISTICS die statistische Erfassung aller im System abgesetzten Kommandos ein- bzw. ausschalten. Berücksichtigt werden alle Kommandos, die in der System- oder den Subsystem-Syntaxdateien enthalten sind.

Die statistische Erfassung erfolgt in die Datei SYS.SDF.CMD.STATISTICS. Ist die Statistik-Routine aktiv (CMD-STATISTICS=\*YES), zählt SDF jedes Kommando, das abgesetzt wird. Dazu wird für jedes Kommando und jeden Aliasnamen ein Zähler geführt.

Wird die Statistik-Routine ausgeschaltet (CMD-STATISTICS=\*NO), wertet SDF den Zählerstand aus und schreibt die aufbereitete Statistik in die oben genannte Datei. Eine bereits existierende Statistikdatei wird dabei überschrieben. Ferner setzt SDF die Zähler der Kommandos auf null zurück.

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.

## Format

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

## Operandenbeschreibung

### **SYNTAX-FILE** = \*UNCHANGED / \*ADD(...) / \*REMOVE(...)

Gibt an, ob eine Benutzersyntaxdatei aktiviert oder deaktiviert werden soll. Es können mehrere Benutzersyntaxdateien 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 Operanden USER-ACCESS, BASIC-ACL, GUARDS im Kommando CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES).

Ist die Datei mit einem Kennwort gegen Ausführung geschützt, so muss das Kennwort in der Kennwort-Tabelle des Auftrags eingetragen sein (Kommando ADD-PASSWORD).

**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 Standarddateinamen **SDF.USER.SYNTAX** wird aktiviert, sofern sie in der Benutzerkennung existiert.

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

Die angegebene Benutzersyntaxdatei wird aktiviert.

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

Eine bzw. mehrere Benutzersyntaxdateien sollen deaktiviert werden.

**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**

Es werden alle aktivierten Benutzersyntaxdateien deaktiviert.

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

In einer Auswahlmaske werden alle aktivierten Benutzersyntaxdateien angezeigt. In einer Markierungsspalte kann der Benutzer die Syntaxdateien markieren, die deaktiviert werden sollen.

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

Die angegebene Benutzersyntaxdatei wird deaktiviert.

**REMOVE-NAME = \*STD**

Die Benutzersyntaxdatei mit dem Standarddateinamen **SDF.USER.SYNTAX** wird deaktiviert.

**SYNTAX-FILE = \*NONE**

Für den Auftrag ist keine Benutzersyntaxdatei aktiviert.

Momentan aktive 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 Eingaben protokolliert werden. Angaben zu Operanden, die als „geheim“ definiert sind, werden in Protokollen immer ausgeblendet.

**LOGGING = \*UNCHANGED**

Es gilt die bisherige Vereinbarung.

**LOGGING = \*INPUT-FORM**

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



**LOGGING = \*ACCEPTED-FORM**

Protokolliert werden:

- alle Namen in ihrer Langform,
- jeder angegebene Operand mit seinem Namen und dem angegebenen Wert,
- die ggf. durch einen Korrekturdialog erzeugte Enddarstellung.

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 Defaultwert,
- 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**

*Die Angabe dieses Operanden ist nur möglich für die Systembetreuung (Privileg TSOS):*

Bestimmt, ob eine Statistik der abgesetzten Kommandos aus der Systemsyntaxdatei angelegt wird.

**MODE = \*UNCHANGED / \*EXECUTION / \*TEST(...)**

Bestimmt, ob der Testmodus ein- oder ausgeschaltet wird.

**MODE = \*EXECUTION**

Der Testmodus wird ausgeschaltet.

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

Der Testmodus wird eingeschaltet. Kommandos, die nach dem MODIFY-SDF-OPTIONS-Kommando abgesetzt werden, werden nur einer Syntaxprüfung unterzogen, aber nicht ausgeführt. Die Kommandos MODIFY-SDF-OPTIONS und SHOW-SDF-OPTIONS werden *immer* ausgeführt.

Der untergeordnete Operand CHECK-PRIVILEGES bestimmt, ob der Benutzer die für die Eingabe erforderlichen Privilegien besitzt.

Die Behandlung von Anweisungen im Testmodus kann im Operanden DEFAULT-PROGRAM-NAME festgelegt werden.

In *S-Prozeduren* werden auch SDF-P-Kontrollflusskommandos ausgeführt. Dies kann zu Fehlern führen, da die Kommandos, die S-Variablen deklarieren bzw. setzen, im Testmodus nicht ausgeführt werden. Zum Testen von S-Prozeduren stellt das kostenpflichtige Subsystem SDF-P eine eigene Testhilfe zur Verfügung (siehe „Testhilfe“, Handbuch „SDF-P“ [6]).

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

Bestimmt, ob zusätzlich zur Syntaxprüfung die Privilegien des Benutzers geprüft werden.

**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 Syntaxprüfung wird geprüft, ob der Benutzers die für die Eingabe erforderlichen Privilegien besitzt.

**DEFAULT-PROGRAM-NAME = \*UNCHANGED / \*NONE / <structured-name 1..30>**

Bestimmt, ob nach Einschalten des Testmodus (siehe Operand MODE) auch SDF-Programmweisungen (beginnen mit //) einer Syntaxprüfung unterzogen werden. In S-Prozeduren können Anweisungen nicht überprüft werden.

Die Überprüfung wird anhand der Syntax der Anweisungen vorgenommen, die für den angegebenen Programmnamen in der Syntaxdatei definiert wurden. Die Überprüfung ist unabhängig von dem Programmaufruf, da die Kommandos LOAD-/START-EXECUTABLE-PROGRAM (bzw. LOAD-/START-PROGRAM) im Testmodus nicht ausgeführt werden.

**FUNCTION-KEYS = \*UNCHANGED / \*STYLE-GUIDE-MODE / \*BY-TERMINAL-TYPE / \*OLD-MODE**

Bestimmt die Belegung der Funktionstasten.

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

Die Belegung der Funktionstasten entspricht dem Fujitsu-Siemens-Style-Guide.

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

*Hinweis*

Bei der Einstellung wird ausgewertet, mit welchem Typ das Terminal im System generiert wurde. Weicht der generierte Terminaltyp von dem 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 dem Wert einer Umgebungsvariablen abhängen. Genaueres ist der jeweiligen Beschreibung des Emulationprogramms zu entnehmen.

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

Die Belegung der Funktionstasten erfolgt im bisherigen Modus, der von allen Terminaltypen unterstützt wird.

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

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

Bestimmt, ob der Eingabepuffer ein-, ausgeschaltet oder zurückgesetzt wird.

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

Der Eingabepuffer wird eingeschaltet. SDF speichert alle syntaktisch richtigen Eingaben (Kommandos und Anweisungen) im Eingabepuffer. Nicht gespeichert werden die Kommandos bzw. Standardanweisungen RESTORE-SDF-INPUT und SHOW-INPUT-HISTORY. Ob ISP-Kommandos gespeichert werden, ist abhängig von der Angabe im Operanden PASSWORD-PROTECTION.

Mit dem Kommando (bzw. der Standardanweisung) SHOW-INPUT-HISTORY kann sich der Benutzer die gespeicherten Eingaben ausgeben lassen. Mit dem Kommando RESTORE-SDF-INPUT kann der Benutzer eine bestimmte Eingabe ausgeben lassen um sie dann unverändert oder auch modifiziert erneut einzugeben.

*Hinweis*

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 immer 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>**

Maximale Anzahl der zu speichernden Eingaben.

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

*Der Operand ist im Prozedurmodus nicht erlaubt.*

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

**PASSWORD-PROTECTION = \*YES**

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

**PASSWORD-PROTECTION = \*NO**

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

*Hinweis*

Bei dieser Einstellung werden Kennwörter mit SHOW-INPUT-HISTORY bzw. RESTORE-SDF-INPUT am Bildschirm im Klartext ausgegeben und können dadurch ggf. unbefugten Personen bekannt werden. Vor allem sollte der Benutzer vor jedem Verlassen des Bildschirmarbeitsplatzes dafür sorgen, dass sich keine unbefugte Person den Inhalt des Eingabepuffers ausgeben lassen kann. Besitzt der Bildschirmarbeitsplatz keine entsprechenden Sicherungsfunktionen (z.B. Chipkarten-Terminal), sollte vorher mindestens der Eingabepuffer gelöscht werden.

**INPUT-HISTORY = \*OFF**

Der Eingabepuffer wird ausgeschaltet. Die nachfolgenden Eingaben werden nicht gespeichert. Aber alle bisher gespeicherten Eingaben sind 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-Returncode**

(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, CMD0679

## Beispiele

```

/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:$TSOS.SYSSDF.AIDSYSA.140
%              VERSION : SESD14.0A600
%
%
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.SPACEOPT.020
%              VERSION : SESD02.0A300
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.SDF-A.041
%              VERSION : SESD04.1E10
% SUBSYSTEM   : :20SH:$TSOS.SYSSDF.TASKDATE.140
%              VERSION : SESD14.0A100
% GROUP       : *NONE
% USER       : *NONE
%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

```

```

/mod-sdf-opt syntax=*add(syssdf.user.special.01) _____ (2)

```

**/show-sdf-opt inf=\*user** \_\_\_\_\_ (3)

```
% USER      : :20SG:$USER1.SYSSDF.USER.SPECIAL.01
%           : VERSION : UNDEFINED
%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
```

**/mod-sdf-opt guidance=\*max** \_\_\_\_\_ (4)

```
.
.
.      (geführter Dialog; beendet mit MOD-SDF-OPT GUIDANCE=*EXPERT)
.
.
```

**/mod-sdf-opt syntax=\*none** \_\_\_\_\_ (5)

**/show-sdf-opt inf=\*user** \_\_\_\_\_ (6)

```
% USER      : *NONE
%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) Das Kommando SHOW-SDF-OPTIONS gibt Informationen über alle aktivierten Syntaxdateien und die eingestellten SDF-Optionen. Die Ausgabe der Subsystem-Syntaxdateien ist je nach Anzahl der aktivierten Subsysteme sehr umfangreich. Sie wird in diesem Beispiel nur auszugweise wiedergegeben.
- (2) Aktivieren der Benutzer-Syntaxdatei *SYSSDF.USER.SPECIAL01* mit dem Kommando MODIFY-SDF-OPTIONS.
- (3) Das Kommando SHOW-SDF-OPTIONS mit INFORMATION=\*USER informiert über aktivierte Benutzer-Syntaxdatei(en) und die eingestellten SDF-Optionen.
- (4) Einschalten des geführten Dialoges mit maximaler Hilfestufe.
- (5) Die Benutzer-Syntaxdatei wird wieder deaktiviert.
- (6) Erneute Ausgabe der Informationen über Benutzer-Syntaxdateien und eingestellte SDF-Optionen.



## REMARK

### Bemerkungen in Kommando-datei einfügen

<b>Anwendungsbereich:</b>	JOB PROCEDURE
<b>Privilegierung:</b>	STD-PROCESSING OPERATING HARDWARE-MAINTENANCE SAT-FILE-MANAGEMENT SAT-FILE-EVALUATION SECURITY-ADMINISTRATION
<b>Berechtigungsschlüssel:</b>	@

### Funktionsbeschreibung

Mit dem REMARK-Kommando lassen sich Bemerkungen in Kommando-dateien (ENTER- oder Prozedurdateien) einfügen, um den Auftrags- bzw. Prozedurablauf zu dokumentieren. Das Kommando wird unverändert protokolliert. Im Dialog ist das Kommando nur sinnvoll, wenn der Auftragsablauf protokolliert wird.

Das Kommando kann von allen Bedienstationen und berechtigten Benutzerprogrammen abgesetzt werden. Es ist auch in Operator-Kommando-dateien zulässig.

In Programmen mit SDF-Schnittstelle ist REMARK mit gleicher Syntax und Funktionalität als Standardanweisung verfügbar.

### Format

<b>REMARK</b>
<b>TEXT</b> = <cmd-rest 0..1800>

## Operandenbeschreibung

**TEXT = <command-rest 0..1800>**

Bemerkungstext. Die maximale Länge des Textes beträgt 1800 Zeichen.

Das Gleichheitszeichen darf nicht erstes signifikantes Zeichen sein, da das Kommando sonst als Wertzuweisung (SET-VARIABLE ohne Kommandoname) interpretiert wird. Ein Semikolon außerhalb von Klammern wird als Kommandotrenner interpretiert, d.h. nachfolgende Zeichen werden als ein folgendes Kommando interpretiert.

Bei der Eingabe am Bedienungsplatz ist die Länge des Textes auf eine Bildschirmzeile begrenzt.

## Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

## Beispiel

Eine Prozedurdatei beginnt mit folgenden Kommandos:

```
/BEG-PROC LOG=*CMD
/REMARK PROZEDUR ZUM BINDEN
...
...
```

Im BEGIN-PROCEDURE-Kommando wird mit der Angabe LOGGING=\*CMD die Ausgabe der Bemerkung nach SYSOUT ermöglicht. Beim Prozedurablauf wird dann die Bemerkung folgendermaßen protokolliert:

```
% /REMARK PROZEDUR ZUM BINDEN
```

# RESET-INPUT-DEFAULTS

## Task-spezifische Default-Werte löschen

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

### Funktionsbeschreibung

Mit dem Kommando RESET-INPUT-DEFAULTS kann der Benutzer task-spezifische Default-Werte löschen. Innerhalb der Task können Default-Werte für Kommandos und Anweisungen definiert werden. Der Benutzer kann alle Default-Werte löschen oder das Löschen auf Default-Werte von Kommandos oder Anweisungen beschränken. Sollen nur Default-Werte von Kommandos bzw. Anweisungen gelöscht werden, kann der Löschvorgang auf bestimmte Kommandos bzw. Anweisungen beschränkt werden.

Das Kommando RESET-INPUT-DEFAULTS kann sich auch auf einen task-spezifischen Default-Wert mit einer bestimmten Eingabeseriennummer beziehen. Dazu kann die Ausgabe des Kommandos SHOW-INPUT-DEFAULTS mit Eingabeseriennummern angefordert werden (Operand INPUT-SERIAL-NUMBER=\*YES).

In Programmen mit SDF-Schnittstelle ist RESET-INPUT-DEFAULTS mit gleicher Funktionalität als Standardanweisung verfügbar.

### Format

RESET-INPUT-DEFAULTS	Kurzname: <b>RSID</b>
<b>OBJECT</b> = <u>*CMD</u> (...) / <u>*STMT</u> (...) / <u>*ALL</u> / <integer 1..9999>     <b>CMD</b> = <u>*ALL</u> / <structured-name 1..30 with-wild(50)>     <b>*STMT</b> (...)     <b>STMT</b> = <u>*ALL</u> / <structured-name 1..30 with-wild(50)>     <b>,PROGRAM</b> = <u>*CURRENT</u> / <u>*ALL</u> / <structured-name 1..30>	

## Operandenbeschreibung

**OBJECT = \*CMD(...) / \*ALL / \*STMT / <integer 1..9999>**

Gibt an, für welche Art von Eingaben die task-spezifischen Default-Werte gelöscht werden sollen.

**OBJECT = \*CMD(...)**

Es werden nur die task-spezifischen Default-Werte von Kommandos gelöscht. Es können die Default-Werte aller oder auch nur bestimmter Kommandos gelöscht werden.

**CMD = \*ALL / <structured-name 1..30 with-wild(50)>**

Gibt an, ob die task-spezifischen Default-Werte aller Kommandos oder nur bestimmter Kommandos gelöscht werden sollen.

**CMD = \*ALL**

Es werden alle task-spezifischen Default-Werte von Kommandos gelöscht.

**CMD = <structured-name 1..30 with-wild(50)>**

Name des Kommandos, dessen task-spezifische Default-Werte gelöscht werden sollen. Bei Verwendung von Musterzeichen bezieht sich der Löschvorgang auf die Default-Werte aller Kommandos, die der eingegebenen Musterzeichenfolge entsprechen.

**OBJECT = \*STMT(...)**

Es werden nur die task-spezifischen Default-Werte von Anweisungen gelöscht. Es können die Default-Werte aller oder auch nur bestimmter Anweisungen eines Programms gelöscht werden.

**STMT = \*ALL / <structured-name 1..30 with-wild(50)>**

Gibt an, ob die task-spezifischen Default-Werte aller Anweisungen oder nur bestimmter Anweisungen gelöscht werden sollen. Im Operanden PROGRAM kann der Benutzer angeben, ob sich der Löschvorgang auf die Default-Werte von Anweisungen eines bestimmten oder aller Programme beziehen soll.

**STMT = \*ALL**

Es werden alle task-spezifischen Default-Werte von Anweisungen gelöscht.

**STMT = <structured-name 1..30 with-wild(50)>**

Name der Anweisung, deren task-spezifische Default-Werte gelöscht werden sollen. Bei Verwendung von Musterzeichen bezieht sich der Löschvorgang auf die Default-Werte aller Anweisungen, die der eingegebenen Musterzeichenfolge entsprechen.

**PROGRAM = \*CURRENT / \*ALL / <structured-name 1..30>**

Bestimmt das Programm, für dessen im Operanden STMT angegebene Anweisungen die task-spezifischen Default-Werte gelöscht werden sollen.

**PROGRAM = \*CURRENT**

Es werden nur Default-Werte von Anweisungen des aktuell in den SDF-Optionen definierten Programms gelöscht. Der Programmname kann mit dem Kommando MODIFY-SDF-OPTIONS (Operand DEFAULT-PROGRAM-NAME) eingestellt werden.

**PROGRAM = \*ALL**

Es werden die Default-Werte aller Anweisungen unabhängig vom Programmnamen gelöscht.

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

Programmname, der in einer aktuell zugewiesenen Syntaxdatei definiert ist.

Es werden nur Default-Werte von Anweisungen des angegebenen Programms gelöscht.

**OBJECT = \*ALL**

Es werden alle task-spezifischen Default-Werte, d.h. von Kommandos und Anweisungen, gelöscht.

**OBJECT = <integer 1..9999>**

Eingabeseriennummer des zu löschenden Default-Wertes.

Die Eingabeseriennummer eines Default-Wertes kann der Ausgabe des Kommandos SHOW-INPUT-DEFAULTS (Operand INPUT-SERIAL-NUMBER=\*YES) entnommen werden.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei ausgeführt
1	0	CMD0001	Es existiert kein task-spezifischer Default-Wert, der den Angaben entspricht. garantierte Meldung: CMD0561
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
1	64	CMD0561	Kommandoausführung nicht erfolgreich garantierte Meldung: CMD0561

## Beispiele

```

/!print-doc line-spacing=*by-ebcdic _____ (1)
/!cre-file sup=*priv(vol=work01,dev-type=d3490-30) _____ (2)
/!cre-file sup=*public _____ (3)
/!cre-file basic-acl=*all-rx _____ (4)

/show-input-defaults input-serial-number=*yes _____ (5)
/" 1 : " !PRINT-DOC LINE-SPACING=*BY-EBCDIC-CONTROL
/" 2 : " !CRE-FILE SUPPORT=*PRIVATE-DISK(VOLUME=WORK01,DEVICE-TYPE=D3490-30)
/" 3 : " !CRE-FILE SUPPORT=*PUBLIC-DISK
/" 4 : " !CRE-FILE BASIC-ACL=*ALL-RX
/reset-input-defaults 2 _____ (6)
/reset-input-defaults *cmd(cmd=create-file) _____ (7)

/show-input-defaults _____ (8)
/!PRINT-DOC LINE-SPACING=*BY-EBCDIC-CONTROL

```

- (1) Für das Kommando PRINT-DOCUMENT wird Auswertung von EBCDIC-Drucksteuerzeichen als Default-Wert eingestellt. Der Operand LINE-SPACING wird ohne Struktureinleiter angegeben, da in der Syntaxdatei \*TEXT-FORMAT der Default-Wert des Operanden DOCUMENT-FORMAT ist.
- (2) Für das Kommando CREATE-FILE wird als Datenträger die Privatplatte *WORK01* vom Gerätetyp *D3490-30* als Default-Wert eingestellt. Der Struktureinleiter muss in diesem Fall angegeben werden, da die VSN und der Gerätetyp eine Privatplatte bezeichnen. Zum Anlegen einer Datei auf gemeinschaftlicher Platte muss jetzt `SUPPORT=*PUBLIC` angegeben werden.
- (3) Für das Kommando CREATE-FILE werden wieder gemeinschaftliche Platten als Default-Wert eingestellt. Zum Anlegen einer Datei auf der Privatplatte *WORK01* genügt jedoch die Angabe `SUPPORT=*PRIVAT`.
- (4) Für das Kommando CREATE-FILE wird eine BASIC-ACL als Default-Wert eingestellt, die nur dem Eigentümer alle Zugriffsrechte einräumt, den anderen Benutzern aber den Lesezugriff und das Ausführen der Datei erlaubt. Das Schutzattribut wird entsprechend gesetzt, wenn mit `PROTECTION=*PARAMETERS` die übergordnete Struktur aktiviert wird.
- (5) Ausgabe aller task-spezifischen Default-Werte mit ihrer Eingabeseriennummer.
- (6) Die Definition mit der Eingabeseriennummer 2 wird gelöscht. Das ist in diesem Fall die Definition mit CREATE-FILE für die Privatplatte.
- (7) Es werden alle Definitionen für das Kommando CREATE-FILE gelöscht.
- (8) Die Ausgabe der task-spezifischen Default-Werte zeigt nur noch die Definition für das Kommando PRINT-DOCUMENT.

## RESTORE-SDF-INPUT

### Bereits erfolgte Eingabe wiederherstellen

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

### Funktionsbeschreibung

Das Kommando RESTORE-SDF-INPUT gibt eine bereits erfolgte Eingabe, die im Eingabepuffer gespeichert wurde, wieder am Bildschirm aus. Der Benutzer kann das angezeigte Kommando bzw. die Anweisung unverändert oder auch modifiziert erneut zur Eingabe verwenden ohne den gesamten Eingabestring erneut eingeben zu müssen. Für die erneute Eingabe muss aber im Eingabestring mindestens ein Zeichen geändert werden (z.B. ein Zeichen mit demselben Zeichen überschreiben).

Im Operanden INPUT wählt der Benutzer die auszugebende Eingabe aus. Voreingestellt ist mit \*LAST-CMD das zuletzt gespeicherte Kommando. Weiter zurückliegende Eingaben kann der Benutzer über die relative bzw. absolute Eingabeseriennummer auswählen. Mit dem Kommando (bzw. der Standardanweisung) SHOW-INPUT-HISTORY kann er sich dabei über alle gespeicherten und damit verfügbaren Eingaben informieren.

Die Ausgabe von RESTORE-SDF-INPUT ist abhängig von dem aktuellen Führungsmodus (siehe Kommando SHOW-SDF-OPTIONS, Ausgabefeld *GUIDANCE*):

- Im geführten Dialog wird für das auszugebende Kommando bzw. die Anweisung ein temporär geführter Dialog angestoßen. Im Operandenfragebogen sind alle Eingaben des Benutzers enthalten.  
Für Kommandos bzw. Anweisungen ohne Operanden ist ein geführter Dialog nicht möglich. Deshalb wird nur der Hilfetext und die Fehlermeldung CMD0070 ausgegeben. Bei Auswahl eines AID-Kommandos wird die Warnung CMD0559 ausgegeben, da AID-Kommandos im geführten Dialog nicht unterstützt werden.
- Im ungeführten Dialog wird der gespeicherte Eingabestring ausgegeben. Will sich der Benutzer bei Modifikationen der Dialogführung bedienen, kann er durch Eingabe eines Fragezeichens direkt nach dem Kommando-/Anweisungsnamen einen temporär geführten Dialog anstoßen.

Die Steuerung des Eingabepuffers (Ein-/Ausschalten und Löschen) erfolgt mit dem Kommando (bzw. der Standardanweisung) MODIFY-SDF-OPTIONS. Eingaben im geführten Dialog werden in der ACCEPTED-Form, Eingaben im ungeführten Dialog in der INPUT-Form gespeichert.

Das Kommando bzw. die Anweisung RESTORE-SDF-INPUT wird nicht gespeichert.

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

Im ungeführten Dialog hat der Benutzer bei der Wiederanzeige mit RESTORE-SDF-INPUT folgende Möglichkeiten:

- Er kann das Kommando bzw. die Anweisung unverändert abschicken. In diesem Fall gibt SDF für jeden geheimen Operanden ein dunkelgesteuertes Eingabefeld aus, in das der Benutzer den gewünschten Wert eingeben kann.
- Er kann das „^“ löschen, dafür den gewünschten Wert direkt einfügen und danach das Kommando bzw. die Anweisung abschicken.

In Programmen mit SDF-Schnittstelle ist RESTORE-SDF-INPUT mit annähernd gleicher Syntax und Funktionalität als Standardanweisung verfügbar.

## Format

RESTORE-SDF-INPUT	Kurzname: <b>RRSDFI</b>
<b>INPUT</b> = <u>*LAST-CMD</u> / <integer -100..-1> / <integer 1..9999>	

## Operandenbeschreibung

**INPUT = \*LAST-CMD / <integer -100..-1> / <integer 1..9999>**

Bestimmt, welche Eingabe aus dem Eingabepuffer erneut ausgegeben werden soll.

**INPUT = \*LAST-CMD**

Das zuletzt gespeicherte Kommando wird ausgegeben.

**INPUT = <integer -100..-1>**

Bezeichnet die gewünschte Eingabe relativ zur aktuellen Eingabe.

**INPUT = <integer 1..9999>**

Bezeichnet die gewünschte Eingabe absolut über ihre Eingabeseriennummer, die SDF bei der Speicherung automatisch vergibt. Der Benutzer kann sich den Eingabepuffer mit Eingabeseriennummern anzeigen lassen (siehe Kommando SHOW-INPUT-HISTORY, Operand INPUT-SERIAL-NUMBER=\*YES).



## Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
0	0	CMD0001	Kommando fehlerfrei ausgeführt
1	0	CMD0001	Ausgabe ist nicht möglich, da der Eingabepuffer leer ist. garantierte Meldung: CMD0558
2	0	CMD0559	AID-Kommando nicht unterstützt im geführten Dialog garantierte Meldung: CMD0559
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
	64	CMD0558	Kommandoausführung nicht erfolgreich

## Beispiele

```

/mod-f-attr sf.dummy,prot=(basic-acl=(owner=(y,y,y),group=(y,n,n),
                                others=(y,n,n))) _____ (1)
/show-job-sta job-id=tsn(1e9q),inf=envir _____ (2)
% EXC0755 INFORMATION ON TASK WITH (&00) '1E9Q' CANNOT BE GIVEN
/restore-sdf _____ (3)
/show-job-sta job-id=tsn(1e9w),inf=envir _____ (4)
%NAME      TSN      STATION  PROCESSOR  HOLD MRSCAT
%          1E9W    $$$06002  FIREBALL

/cre-file test _____ (5)
/show-file-attr test,inf=min
%N NNN NW                3 :20SG:$USER1.TEST

/show-input i-s-n=y _____ (6)
" 1 : " mod-f-attr sf.dummy,prot=(basic-acl=(owner=(y,y,y),group=(y,n,n),
others=(y,n,n)))
/" 2 : " show-job-sta job-id=*tsn(1e9q),inf=envir
/" 3 : " show-job-sta job-id=*tsn(1e9w),inf=envir
/" 4 : " cre-file test
/" 5 : " show-file-attr test,inf=*min

/restore-sdf 1 _____ (7)
/mod-f-attr test ,prot=(basic-acl=(owner=(y,y,y),group=(y,n,n),
                                others=(y,n,n))) _____ (8)

/restore-sdf
/mod-f-attr?test ,prot=(basic-acl=(owner=(y,y,y),group=(y,n,n),
                                others=(y,n,n))) _____ (9)

```

```

COMMAND : MODIFY-FILE-ATTRIBUTES
OPERANDS : ...-NAME=TEST,PROTECTION=*PARAMETERS(BASIC-ACL=*PARAMETERS(OWNER=*PAR
AMETERS(READ=*YES,WRITE=*YES,EXEC=*YES),GROUP=*PARAMETERS(READ=*YE
S,WRITE=*NO,EXEC=*NO),OTHERS=*PARAMETERS(READ=*YES,WRITE=*NO,EXEC=...
-----
FILE-NAME           = TEST
NEW-NAME            = *SAME
SUPPORT             = *UNCHANGED
PROTECTION          = (PROTECTION-ATTR=*UNCHANGED,ACCESS=*BY-PROTECTION-ATTR,US
ER-ACCESS=*BY-PROTECTION-ATTR,BASIC-ACL=(OWNER=(READ=Y,WR
ITE=Y,EXEC=Y),GROUP=(READ=Y,WRITE=N,EXEC=N),OTHERS=(READ=
Y,WRITE=N,EXEC=N)),GUARDS=*BY-PROTECTION-ATTR,WRITE-PASSW
ORD=*BY-PROT-ATTR-OR-UNCH,READ-PASSWORD=*BY-PROT-ATTR-OR-
UNCH,EXEC-PASSWORD=*BY-PROT-ATTR-OR-UNCH,DESTROY-BY-DELET
E=*BY-PROTECTION-ATTR,AUDIT=*UNCHANGED,SPACE-RELEASE-LOCK
=*BY-PROTECTION-ATTR,EXPIRATION-DATE=*BY-PROTECTION-ATTR,
FREE-FOR-DELETION=*BY-PROT-ATTR-OR-UNCH)
-----
NEXT = *down(basic-acl) ←----- (10)
KEYS : F1=?      F3=*EXIT   F5=*REFRESH  F6=*EXIT-ALL  F8==  F9=REST-SDF-IN
      F11=*EXECUTE F12=*CANCEL

```

```

COMMAND : MODIFY-FILE-ATTRIBUTES
STRUCTURE: BASIC-ACL=
OPERANDS : ...-NAME=TEST,PROTECTION=*PARAMETERS(BASIC-ACL=*PARAMETERS(OWNER=*PAR
AMETERS(READ=*YES,WRITE=*YES,EXEC=*YES),GROUP=*PARAMETERS(READ=*YE
S,WRITE=*NO,EXEC=*NO),OTHERS=*PARAMETERS(READ=*YES,WRITE=*NO,EXEC=...
-----
OWNER               = (READ=Y,WRITE=Y,EXEC=Y)
GROUP               = (READ=Y,WRITE=N,EXEC=y) ←----- (11)
OTHERS              = (READ=Y,WRITE=N,EXEC=y)
-----
NEXT = *CONTINUE
KEYS : F1=?      F3=*EXIT   F5=*REFRESH  F6=*EXIT-ALL  F8==  F9=REST-SDF-IN
      F11=*EXECUTE F12=*CANCEL

```

- (1) Für die Datei *SFDUMMY* wird im Katalogeintrag BASIC-ACL-Schutz eingetragen (Eigentümer alle Rechte, Gruppe und Andere nur Leserecht).
- (2) Das Kommando SHOW-JOB-STATUS soll Informationen über die Task mit der TSN *IE9Q* ausgeben. Eine Task mit dieser TSN existiert jedoch nicht.
- (3) Das Kommando RESTORE-SDF-INPUT soll die letzte Eingabe erneut ausgeben.

- (4) Das Kommando SHOW-JOB-STATUS (siehe Punkt 2) wird ausgegeben. In der ausgegebenen Zeichenfolge wird die TSN korrigiert zu *IE9W* und mit `[DUE]` abgeschickt. Für diese TSN werden Informationen ausgegeben.
- (5) Die Datei *TEST* wird katalogisiert. Anschließend werden mit SHOW-FILE-ATTRIBUTES die Schutzrechte in Kurzform ausgegeben.
- (6) Mit SHOW-INPUT-HISTORY wird der Inhalt des Eingabepuffers mit Eingabeseriennummer (INPUT-SERIAL-NUMBER=\*YES) ausgegeben.
- (7) RESTORE-SDF-INPUT gibt das Kommando mit der Seriennummer 1 erneut aus.
- (8) In dem angezeigten MODIFY-FILE-ATTRIBUTES-Kommando wird der Dateiname geändert in *TEST* und mit DUE abgeschickt. Die Datei *TEST* erhält somit den gleichen BASIC-ACL-Schutz wie zuvor die Datei *SFDUMMY*.
- (9) Der Benutzer will die Schutzattribute der Datei *TEST* erneut ändern. RESTORE-SDF-INPUT gibt das zuletzt eingegebene Kommando (siehe Punkt 8) erneut aus. Nach dem Kommandonamen wird ein Fragezeichen eingefügt und die Zeile mit `[DUE]` abgeschickt.
- (10) SDF wechselt in den temporär geführten Dialog und gibt den Operandenfragebogen des Kommandos MODIFY-FILE-ATTRIBUTES mit den explizit gesetzten Angaben zu BASIC-ACL aus. Für eine übersichtlichere Eingabe fordert der Benutzer den Unterfragebogen zu BASIC-ACL (Struktur \*PARAMETERS) mit „\*down(basic-acl)“ in der NEXT-Zeile an.
- (11) Im Unterfragebogen trägt er die gewünschten Werte ein (Gruppe und Andere erhalten Ausführungsrecht mit EXEC=\*YES) und schickt ihn mit `[DUE]` weg.

#### *Anmerkung*

Die Änderung der BASIC-ACL-Zugriffsrechte in der kürzest möglichen Form erfolgt durch Editieren der von RESTORE-SDF-INPUT ausgegebenen Kommandozeile (bzw. bereits durch Editieren der entsprechenden Kommandozeile in der Ausgabe des Kommandos SHOW-INPUT-HISTORY).

## SHOW-CMD

### Syntaxbeschreibung eines Kommandos ausgeben

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

### Funktionsbeschreibung

Das Kommando SHOW-CMD gibt die Syntaxbeschreibung eines Kommandos aus. Zusätzlich werden der Name und Typ der verwendeten Syntaxdatei angezeigt. Mit CMD-NAME=\*ALL erhält der Benutzer eine Liste aller Kommandonamen. Durch Verwendung von Musterzeichen im Kommandonamen erhält der Benutzer eine Liste Kommandonamen, die der Musterzeichenfolge entsprechen.

Die Ausgabe erfolgt wahlweise auf SYSOUT oder SYSLST.

Der Operand INFORMATION steuert den Umfang der Syntaxbeschreibung. Mit der Angabe INFORMATION=\*MINIMUM erhält der Benutzer den Kommandonamen, den Namen der Syntaxdatei, die die Syntaxbeschreibung enthält, die Operandennamen und eventuelle Default-Werte. Alternative Operandenwerte werden bei INFORMATION=\*MEDIUM zusätzlich angezeigt. Bei INFORMATION=\*MAXIMUM enthält die Ausgabe auch Hilfetexte. Bei der Ausgabe wird nur die Syntaxbeschreibung der zugewiesenen Syntaxdateien wiedergegeben. Insbesondere werden task-spezifische Default-Werte nicht angezeigt.

Der Operand FORM steuert die Ausgabe von Syntaxobjekten, deren Eingabe im geführten Dialog nicht zulässig ist. Mit FORM=\*UNGUIDED erhält der Benutzer die Kommando-syntax, die auch im ungeführten Dialog eingegeben werden kann, bzw. eine Kommando-liste, die auch Aliasnamen der Kommandonamen enthält. Einschränkungen, die für Kommandos bzw. Operanden bezüglich der Eingabemodi (Dialog, Stapelbetrieb, Prozedur) bestehen, werden bei der Ausgabe mit einem Stern (\*) gekennzeichnet. Der Benutzer kann sich somit z.B. auch über einen Operanden informieren, dessen Eingabe nur im Stapelbetrieb zulässig ist.

Der Operand CHECK-PRIVILEGES bestimmt, ob bei der Ausgabe die Privilegien des Benutzers berücksichtigt werden. Mit CHECK-PRIVILEGES=\*YES erhält er nur Informationen über Operanden und Operandenwerte von Kommandos, die er entsprechend seinen Privilegien auch verwenden darf. Bei der Ausgabe einer Liste von Kommandonamen sind Kommandos, die der Benutzer nicht verwenden darf, mit einem Stern (\*) gekennzeichnet.

## Format

SHOW-CMD
<pre> <b>CMD-NAME</b> = *ALL / &lt;structured-name 1..30 with-wild&gt; , <b>INFORMATION</b> = *MINIMUM / *MEDIUM / *MAXIMUM , <b>FORM</b> = *GUIDED / *UNGUIDED , <b>CHECK-PRIVILEGES</b> = *YES / *NO , <b>OUTPUT</b> = *SYSOUT / *SYSLST(...) *SYSLST(...)         <b>SYSLST-NUMBER</b> = *STD / &lt;integer 1..99&gt;         <b>LINES-PER-PAGE</b> = *STD / *UNLIMITED / &lt;integer 1..200&gt; </pre>

## Operandenbeschreibung

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

Name des gewünschten Kommandos.

**CMD-NAME = \*ALL**

Listet alle Kommandos in alphabetischer Reihenfolge auf. Mit FORM=\*UNGUIDED werden auch definierte Aliasnamen (in eigenen Ausgabezeilen) ausgegeben.

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

Name des Kommandos, dessen Syntax ausgegeben werden soll. Bei Angabe eines Aliasnamens wird der reale Kommandoname ausgegeben, gefolgt vom Aliasnamen in Klammern.

Bei Verwendung von Musterzeichen werden alle Kommandos in alphabetischer Reihenfolge aufgelistet, die der eingegebenen Musterzeichenfolge entsprechen. Dabei werden mit FORM=\*UNGUIDED auch der Musterzeichenfolge entsprechende definierte Aliasnamen (in eigenen Ausgabezeilen) ausgegeben.

**INFORMATION = \*MINIMUM / \*MEDIUM / \*MAXIMUM**

Bestimmt den Umfang der Ausgabe. Bei Ausgabe von Kommandolisten wird der Operand INFORMATION ignoriert.

**INFORMATION = \*MINIMUM**

Die Ausgabe enthält den Kommandonamen, Operanden und voreingestellte Operandenwerte.

**INFORMATION = \*MEDIUM**

Die Ausgabe enthält den Kommandonamen, Operanden, voreingestellte und alternative Operandenwerte.

**INFORMATION = \*MAXIMUM**

Die Ausgabe enthält den Kommandonamen, Operanden, voreingestellte und alternative Operandenwerte. Zusätzlich werden alle Hilfetexte angezeigt.

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

Bestimmt, ob die Syntax für den geführten oder den ungeführten Dialog ausgegeben werden soll.

**FORM = \*GUIDED**

Operanden und Operandenwerte, die im geführten Dialog nicht erlaubt sind, werden nicht ausgegeben.

**FORM = \*UNGUIDED**

Die Ausgabe enthält auch Operanden und Operandenwerte, die im geführten Dialog nicht erlaubt sind. Bei Ausgabe von Kommandolisten werden Aliasnamen in eigenen Ausgabzeilen angezeigt.

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

Bestimmt, ob bei der Ausgabe die Privilegien des Benutzers berücksichtigt werden.

**CHECK-PRIVILEGES = \*YES**

Der Benutzer erhält nur Informationen über Kommandos, Operanden und Operandenwerte, die er entsprechend seinen Privilegien auch verwenden darf.

**CHECK-PRIVILEGES = \*NO**

Der Benutzer erhält unabhängig von seinen Privilegien Informationen über alle Kommandos, Operanden und Operandenwerte.

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

Gibt an, wohin die Information auszugeben ist.

**OUTPUT = \*SYSOUT**

Die Ausgabe erfolgt in die Systemdatei SYSOUT.

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

Die Ausgabe erfolgt druckaufbereitet in die Systemdatei SYSLST.

Das erste Byte jedes Ausgabesatzes ist X'40'. Der Operand LINES-PER-PAGES bestimmt nach wie vielen Ausgabesätzen eine Überschriftszeile mit Seitenvorschub erzeugt wird.

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

Bestimmt, ob die Ausgabe in die Systemdatei SYSLST oder in eine SYSLST-Datei aus der Menge SYSLST01 bis SYSLST99 erfolgen soll.

Default-Wert ist \*STD, d.h. die Ausgabe erfolgt in die Systemdatei SYSLST.

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

Gibt an, nach wie vielen Ausgabesätzen eine neue Druckseite beginnen soll. Jede Druckseite beginnt mit einer Überschriftszeile, die im ersten Byte ein Seitenvorschub-Steuerzeichen enthält. In der Überschrift steht der Name des angezeigten Kommandos und die Seitennummer.

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

Nach 55 Ausgabesätzen beginnt eine neue Druckseite.

**LINES-PER-PAGE = \*UNLIMITED**

Die Ausgabe wird nicht in Druckseiten aufgeteilt. Es werden keine Überschriftenzeilen ausgegeben.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei ausgeführt
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft
1	64	CMD0812	Kommandoausführung nicht erfolgreich
			garantierte Meldungen: CMD0500, CMD0812

**Beispiele**

```

/show-cmd cmd=add-pass _____ (1)
%ADD-PASSWORD (ADD-PASS,ADPW)
% FROM :20SH:$TSOS.SYSSDF.BS2CP.140 (SYSTEM)
% PASSWORD =

/show-cmd cmd=add-pass,information=max _____ (2)
%ADD-PASSWORD (ADD-PASS,ADPW)
% FROM :20SH:$TSOS.SYSSDF.BS2CP.140 (SYSTEM)
%Adds passwords for files or job variables to the password table of the
%task
% PASSWORD =
% -list-possible (63)-: x-string_1..8 or c-string_1..4 or
% integer_-2147483648..2147483647
% Specifies passwords to be added to the password table

/show-cmd cmd=write* _____ (3)
% WRITE-ACCOUNTING-RECORD
% WRITE-DISKETTE
% WRITE-SPOOL-TAPE
% WRITE-TEXT

```

- (1) Ausgabe der Syntax des Kommandos ADD-PASSWORD.
- (2) Ausgabe der Syntax des Kommandos ADD-PASSWORD mit allen Operanden, Operandenwerten und Hilfetexten (INFORMATION=\*MAXIMUM).
- (3) Ausgabe aller Kommandos, die mit der Zeichenfolge „WRITE“ beginnen.

## SHOW-INPUT-DEFAULTS

### Task-spezifische Default-Werte ausgeben

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

### Funktionsbeschreibung

Mit dem Kommando SHOW-INPUT-DEFAULTS kann sich der Benutzer über alle aktuell definierten task-spezifischen Default-Werte informieren. Innerhalb der Task können Default-Werte für Kommandos und Anweisungen definiert werden. Der Benutzer kann sich alle Default-Werte ausgeben lassen oder die Ausgabe auf Default-Werte von Kommandos oder Anweisungen beschränken. Werden nur Default-Werte von Kommandos bzw. Anweisungen ausgegeben, kann die Ausgabe auf bestimmte Kommandos bzw. Anweisungen beschränkt werden.

Die Ausgabe erfolgt wahlweise auf SYSOUT oder SYSLST.

Task-spezifische Default-Werte können mit dem Kommando RESET-INPUT-DEFAULTS wieder gelöscht werden. Zur Identifizierung eines bestimmten Default-Wertes im Kommando RESET-INPUT-DEFAULTS kann die Ausgabe des Kommandos SHOW-INPUT-DEFAULTS mit Eingabeseriennummern angefordert werden (Operand INPUT-SERIAL-NUMBER=\*YES).

In Programmen mit SDF-Schnittstelle ist SHOW-INPUT-DEFAULTS mit gleicher Funktionalität als Standardanweisung verfügbar.



## Format

SHOW-INPUT-DEFAULTS	Kurzname: SHID
<p><b>OBJECT = *CMD (...)</b> / *STMT(...) / *ALL</p> <p><b>*CMD(...)</b></p> <p>      <b>CMD = *ALL</b> / &lt;structured-name 1..30 with-wild(50)&gt;</p> <p><b>*STMT(...)</b></p> <p>      <b>STMT = *ALL</b> / &lt;structured-name 1..30 with-wild(50)&gt;</p> <p>      <b>,PROGRAM = *CURRENT</b> / *ALL / &lt;structured-name 1..30&gt;</p> <p><b>,OUTPUT = *SYSOUT</b> / *SYSLST(...)</p> <p><b>*SYSLST(...)</b></p> <p>      <b>SYSLST-NUMBER = *STD</b> / &lt;integer 1..99&gt;</p> <p><b>,INPUT-SERIAL-NUMBER = *NO</b> / *YES</p>	

## Operandenbeschreibung

### OBJECT = \*CMD(...) / \*ALL / \*STMT

Gibt an, für welche Art von Eingaben die task-spezifischen Default-Werte ausgegeben werden sollen.

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

Es werden nur die task-spezifischen Default-Werte von Kommandos ausgegeben. Die Ausgabe kann für alle oder auch nur für bestimmte Kommandos angefordert werden.

### CMD = \*ALL / <structured-name 1..30 with-wild(50)>

Gibt an, ob die task-spezifischen Default-Werte aller Kommandos oder nur bestimmter Kommandos ausgegeben werden sollen.

### CMD = \*ALL

Es werden alle task-spezifischen Default-Werte von Kommandos ausgegeben.

### CMD = <structured-name 1..30 with-wild(50)>

Name des Kommandos, dessen task-spezifische Default-Werte ausgegeben werden sollen. Bei Verwendung von Musterzeichen erfolgt die Ausgabe für alle Kommandos, die der eingegebenen Musterzeichenfolge entsprechen.

**OBJECT = \*STMT(...)**

Es werden nur die task-spezifischen Default-Werte von Anweisungen ausgegeben. Die Ausgabe kann für alle oder auch nur für bestimmte Anweisungen eines Programms angefordert werden.

**STMT = \*ALL / <structured-name 1..30 with-wild(50)>**

Gibt an, ob die task-spezifischen Default-Werte aller Anweisungen oder nur bestimmter Anweisungen ausgegeben werden sollen. Im Operanden PROGRAM kann der Benutzer angeben, ob die Ausgabe für Anweisungen eines bestimmten oder aller Programme erfolgen soll.

**STMT= \*ALL**

Es werden alle task-spezifischen Default-Werte von Anweisungen ausgegeben.

**STMT = <structured-name 1..30 with-wild(50)>**

Name der Anweisung, deren task-spezifische Default-Werte ausgegeben werden sollen. Bei Verwendung von Musterzeichen erfolgt die Ausgabe für alle Anweisungen, die der eingegebenen Musterzeichenfolge entsprechen.

**PROGRAM = \*CURRENT / \*ALL / <structured-name 1..30>**

Bestimmt das Programm, für dessen im Operanden STMT angegebene Anweisungen die task-spezifischen Default-Werte ausgegeben werden sollen.

**PROGRAM = \*CURRENT**

Es werden nur Default-Werte von Anweisungen des aktuell in den SDF-Optionen definierten Programms ausgegeben. Der Programmname kann mit dem Kommando MODIFY-SDF-OPTIONS (Operand DEFAULT-PROGRAM-NAME) eingestellt werden.

**PROGRAM = \*ALL**

Es werden die Default-Werte aller Anweisungen unabhängig vom Programmnamen ausgegeben.

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

Programmname, der in einer aktuell zugewiesenen Syntaxdatei definiert ist.

Es werden nur Default-Werte von Anweisungen des angegebenen Programms ausgegeben.

**OBJECT = \*ALL**

Es werden alle task-spezifischen Default-Werte, d.h. von Kommandos und Anweisungen ausgegeben.

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

Gibt an, wohin die Information auszugeben ist.

**OUTPUT = \*SYSOUT**

Die Ausgabe erfolgt in die Systemdatei SYSOUT.

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

Die Ausgabe erfolgt in die Systemdatei SYSLST.

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

Bestimmt, ob die Ausgabe in die Systemdatei SYSLST oder in eine SYSLST-Datei aus der Menge SYSLST01 bis SYSLST99 erfolgen soll.

Default-Wert ist \*STD, d.h. die Ausgabe erfolgt in die Systemdatei SYSLST.

**INPUT-SERIAL-NUMBER = \*NO / \*YES**

Gibt an, ob die Eingaben mit ihrer Eingabeseriennummer ausgegeben werden sollen. Voreingestellt ist mit \*NO die Ausgabe ohne Eingabeseriennummer. Die Eingabeseriennummer kann im Kommando RESET-INPUT-DEFAULTS zum Löschen eines bestimmten Default-Wertes verwendet werden.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei ausgeführt
1	0	CMD0001	Es existiert kein task-spezifischer Default-Wert, der den Angaben entspricht. garantierte Meldung: CMD0561
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
1	64	CMD0561	Kommandoausführung nicht erfolgreich garantierte Meldung: CMD0561

**Beispiel**

Siehe Kommando RESET-INPUT-DEFAULTS.

## SHOW-INPUT-HISTORY

### Gespeicherte Eingaben nach SYSOUT ausgeben

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

### Funktionsbeschreibung

Das Kommando SHOW-INPUT-HISTORY gibt den Inhalt des Eingabepuffers nach SYSOUT aus. Der Benutzer kann sich somit über zurückliegende Eingaben informieren. Eine bestimmte Eingabe kann er dann mit dem Kommando RESTORE-SDF-INPUT anfordern und erneut unverändert oder modifiziert wieder eingeben. Für die erneute Eingabe muss aber im Eingabestring mindestens ein Zeichen geändert werden (z.B. ein Zeichen mit demselben Zeichen überschreiben).

Zur Identifizierung im Kommando RESTORE-SDF-INPUT kann die Information mit Eingabeseriennummern angefordert werden (Operand INPUT-SERIAL-NUMBER=\*YES).

Den Umfang der Ausgabe des Eingabepuffers kann der Benutzer einschränken:

- Angabe der maximal auszugegeben Anzahl von gespeicherten Eingaben (Operand ENTRIES); voreingestellt sind acht Eingaben.
- Angabe des gewünschten Eingabetyps (Operand SELECT): möglich sind Kommandos oder Anweisungen. Voreingestellt ist die Ausgabe von Kommandos.
- Angabe einer Musterzeichenfolge (Operand PATTERN): Es werden nur Kommandos bzw. Anweisungen ausgewählt, die der Musterzeichenfolge entsprechen. Voreingestellt wird keine Musterzeichenfolge verwendet.

Die Steuerung des Eingabepuffers (Ein-/Ausschalten und Löschen) erfolgt mit dem Kommando MODIFY-SDF-OPTIONS. Eingaben im geführten Dialog werden in der ACCEPTED-Form, Eingaben im ungeführten Dialog in der INPUT-Form gespeichert. Das Kommando bzw. die Anweisung SHOW-INPUT-HISTORY selbst wird nicht gespeichert.

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

Im ungeführten Dialog hat der Benutzer bei der Wiederanzeige mit RESTORE-SDF-INPUT folgende Möglichkeiten:

- Er kann das Kommando bzw. die Anweisung unverändert abschicken. In diesem Fall gibt SDF für jeden geheimen Operanden ein dunkelgesteuertes Eingabefeld aus, in das der Benutzer den gewünschten Wert eingeben kann.
- Er kann das „^“ löschen, dafür den gewünschten Wert direkt einfügen und danach das Kommando bzw. die Anweisung abschicken.

In Programmen mit SDF-Schnittstelle ist SHOW-INPUT-HISTORY mit gleicher Funktionalität als Standardanweisung verfügbar.

## Format

<b>SHOW-INPUT-HISTORY</b>	Kurzname: <b>SHIH</b>
<p><b>ENTRIES</b> = <u>8</u> / &lt;integer 1..100&gt; / *ALL</p> <p>,<b>SELECT</b> = *<u>CMD</u> / *ALL / *STMT</p> <p>,<b>PATTERN</b> = *<u>NONE</u> / &lt;structured-name 1..30 with-wild&gt;</p> <p>,<b>INPUT-SERIAL-NUMBER</b> = *<u>NO</u> / *YES</p>	

## Operandenbeschreibung

**ENTRIES = 8 / <integer 1..100> / \*ALL**

Bestimmt die maximale Anzahl von Eingaben, die ausgegeben werden sollen. Bei Angabe von \*ALL wird der gesamte Inhalt des Eingabepuffers ausgegeben.

**SELECT = \*CMD / \*ALL / \*STMT**

Gibt an, welche Art von Eingaben ausgegeben werden sollen.

**SELECT = \*CMD**

Es werden nur Kommandos ausgegeben.

**SELECT = \*ALL**

Es werden Kommandos und Anweisungen ausgegeben.

**SELECT = \*STMT**

Es werden nur Anweisungen ausgegeben.

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

Gibt an, ob bei der Auswahl der auszugebenden Eingaben eine Musterzeichenfolge gelten soll. Voreingestellt \*NONE, d.h. keine Auswahl nach Musterzeichen. Bei Angabe einer Musterzeichenfolge werden nur Eingaben berücksichtigt, die der angegebenen Musterzeichenfolge entsprechen.

**INPUT-SERIAL-NUMBER = \*NO / \*YES**

Gibt an, ob die Eingaben mit ihrer Eingabeseriennummer ausgegeben werden sollen. Voreingestellt ist mit \*NONE die Ausgabe ohne Eingabeseriennummer. Die Eingabeseriennummer kann bei RESTORE-SDF-INPUT zur Auswahl der gewünschten Eingabe verwendet werden.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando fehlerfrei ausgeführt
1	0	CMD0001	Ausgabe ist nicht möglich, da der Eingabepuffer leer ist. garantierte Meldung: CMD0560
1	32	CMD0500	Syntaxbeschreibung in der aktuellen Syntaxdatei fehlerhaft garantierte Meldung: CMD0500
1	64	CMD0560	Kommandoausführung nicht erfolgreich garantierte Meldung: CMD0560

**Beispiel**

Siehe Kommando RESTORE-SDF-INPUT.

# SHOW-RETURNCODE

## Kommando-Returncode des letzten Kommandos ausgeben

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

### Funktionsbeschreibung

Das Kommando SHOW-RETURNCODE gibt den Kommando-Returncode des zuletzt eingegebenen Kommandos nach SYSOUT aus.

### Format

<b>SHOW-RETURNCODE</b>	Kurzname: <b>SHRTC</b>

Das Kommando hat keine Operanden und wird sofort ausgeführt.

### Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

### Beispiel

```
/cre-file test
% CMD0051 INVALID OPERAND 'FILE-NAME'
% DMS0683 FILE ':20SG:$USER1.TEST' ALREADY EXISTS
/show-returncode
%Returncode:
%Subcode2 = 0
%Subcode1 = 64
%Maincode = DMS0683
```

## SHOW-SDF-OPTIONS

### Aktive Syntaxdateien und SDF-Optionen anzeigen

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

### Funktionsbeschreibung

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 erhält der Benutzer nur Informationen über Einstellungen, die er selbst mit MODIFY-SDF-OPTIONS tasklokal ändern kann: Er erhält Namen und Versionen aller aktivierten Benutzersyntaxdateien und alle Einstellungen der SDF-Optionen.

Der Benutzer kann mit dem Kommando MODIFY-SDF-OPTIONS Benutzersyntaxdateien aktivieren bzw. deaktivieren und die Einstellung der SDF-Optionen task-spezifisch verändern.

Bei Einsatz des Subsystems SDF-P (ab Version V2.0A) kann die Ausgabe in strukturierte S-Variablen gelenkt werden.

### Format

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



## Operandenbeschreibung

### INFORMATION = \*ALL / \*USER

Bestimmt den Umfang der auszugebenden Informationen.

### INFORMATION = \*ALL

Gibt Informationen über die aktivierten Syntaxdateien und die aktuellen Einstellungen der SDF-Optionen aus.

### INFORMATION = \*USER

Gibt Informationen über die aktivierten Benutzersyntaxdateien und die aktuellen Einstellungen der SDF-Optionen aus.

## Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

## Ausgabeformat

Die Ausgabefelder und Werte unter der Überschrift „CURRENT SDF OPTIONS“ entsprechen den Operanden und Operandenwerten des Kommandos MODIFY-SDF-OPTIONS.

Ausgabefeld	Bedeutung und mögliche Werte
SYNTAX FILES CURRENTLY ACTIVATED :	Überschriftszeile für Informationen über die aktivierten Syntaxdateien
SYSTEM	Name der aktiven Systemsyntaxdatei bzw. *NONE, wenn ausschließlich eine Gruppensyntaxdatei zugewiesen ist
VERSION	Version der Systemsyntaxdatei
SUBSYSTEM	Name einer aktiven Subsystem-Syntaxdatei
VERSION	Version der Subsystem-Syntaxdatei
GROUP	Name einer aktiven Gruppensyntaxdatei
VERSION	Version der Gruppensyntaxdatei; entfällt, wenn keine Gruppensyntaxdatei aktiviert ist (GROUP=*NONE)
USER	Name einer aktiven Benutzersyntaxdatei
VERSION	Version der Benutzersyntaxdatei; entfällt, wenn keine Benutzersyntaxdatei aktiviert ist (USER=*NONE)

Tabelle 8: Ausgabeformat des Kommandos SHOW-SDF-OPTIONS (Teil 1 von 2)

Ausgabefeld	Bedeutung und mögliche Werte
CURRENT SDF OPTIONS :	Überschriftszeile für Informationen über die aktuellen Einstellungen der SDF-Optionen
GUIDANCE	Führungsstufe: *EXPERT / *NO / *MINIMUM / *MEDIUM / *MAXIMUM
LOGGING	Umfang der Protokollierung von Kommandos und Anweisungen: *INPUT-FORM / *ACCEPTED-FORM / *INVARIANT-FORM
CONTINUATION	Kommandofortsetzung: *OLD-MODE / *NEW-MODE
UTILITY-INTERFACE	Eingabeschnittstelle für Utilities: *OLD-MODE / *NEW-MODE
PROCEDURE-DIALOGUE	Syntaxfehlerdialog bzw. Hilfedialog in Prozeduren: *YES / *NO
MENU-LOGGING	Menüprotokollierung: *YES / *NO
CMD-STATISTICS	<i>Ausgabe nur für privilegierte Benutzer (TSOS)</i> Statistische Erfassung von Kommandos: *YES / *NO
MODE	Syntaxprüfung von Kommandos: *EXECUTION / *TEST
CHECK-PRIVILEGES	Privilegienprüfung im Testmodus: *YES / *NO
DEFAULT-PROGRAM-NAME	Syntaxprüfung von Anweisungen eines Programms: *NONE / <structured-name 1..20>
FUNCTION-KEYS	Funktionstastenbelegung: *OLD-MODE / *NEW-MODE
INPUT-HISTORY	Eingabepuffer: *ON / *OFF
NUMBER-OF-INPUTS	Größe des Eingabepuffers (max. Anzahl der zu speichernden Eingaben): <integer 1..100>
PASSWORD-PROTECTION	Schutz von geheimen Operandenwerten im Eingabepuffer: *YES / *NO

Tabelle 8: Ausgabeformat des Kommandos SHOW-SDF-OPTIONS (Teil 2 von 2)

## Ausgabe in S-Variablen

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Berücksichtigung der Privilegien bei Syntaxprüfung	var(*LIST).CHECK-PRIV	S	*NO *YES
statistische Erfassung von Kommandos	var(*LIST).CMD-STATIS	S	leere Zeichenkette *NO *YES
Kommandofortsetzungsart	var(*LIST).CONTI	S	*NEW-MODE *OLD-MODE
Funktionstastenbelegung	var(*LIST).FUNC-KEY	S	*OLD-MODE *STYLE-GUIDE-MODE
Dialogführung	var(*LIST).GUIDE	S	*EXPERT *MAX *MED *MIN *NO
Aufzeichnung im Eingabepuffer	var(*LIST).INPUT-HIST	S	*OFF *ON
Kommandoprotokollierung	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>
Aufzeichnung von Kennwörtern und ISP-Kommandos im Eingabepuffer	var(*LIST).PASSWORD-PROT	S	*NO *YES
Syntaxfehlerdialog	var(*LIST).PROC-DIALOG	S	*NO *YES
Name der Syntaxdatei	var(*LIST).SF(*LIST).F-NAME	S	<filename 1..54>
Art der Syntaxdatei	var(*LIST).SF(*LIST).TYPE	S	*GROUP *SUBSYS *SYS *USER
Version der Syntaxdatei	var.SF(*LIST).VERSION	S	<text 1..12>

Tabelle 9: Ausgabevariablen des Kommandos SHOW-SDF-OPTIONS (Teil 1 von 2)

Ausgabe-Information	Name der S-Variablen	T	Inhalt
Programmname zur Syntaxprüfung von Anweisungen	var(*LIST).TEST-PROG-NAME	S	*NONE <structured-name 1..30>
Eingabeschnittstelle für Utilities	var(*LIST).UTILITY-INTERF	S	*NEW-MODE *OLD-MODE

Tabelle 9: Ausgabevariablen des Kommandos SHOW-SDF-OPTIONS (Teil 2 von 2)

*Hinweise zum Inhalt der Variablen:*

- **CMD-STATIS:**  
Für nicht-privilegierte Benutzer enthält die Variable nur die leere Zeichenkette.
- **F-NAME, TYPE und VERSION:**  
Die Elemente werden für jede aktivierte Syntaxdatei angelegt. Ist keine Benutzersyntaxdatei aktiviert, werden die Elemente trotzdem einmal für TYPE=\*USER angelegt, wobei F-NAME und VERSION nur die leere Zeichenkette enthalten.  
Bei Angabe von INFORMATION=\*USER entfallen die Informationen über System-, Subsystem- und Gruppensyntaxdateien.  
VERSION enthält die in der Globalinformation der Syntaxdatei festgelegte Version.  
Wurde dort keine Version festgelegt, enthält sie den Wert UNDEFINED.

**Beispiele**

```

/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:$TSOS.SYSSDF.SPACEOPT.020
%             VERSION : SESD02.0A300
% SUBSYSTEM   : :20SH:$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 : UNDEFINED
%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

/mod-sdf-opt syntax=*add(syssdf.user.special.01) _____ (2)
/show-sdf-opt inf=*user _____ (3)

% USER           : :20SG:$USER1.SDF.USER.SYNTAX
%                 VERSION : UNDEFINED
% USER           : :20SG:$USER1.SYSSDF.USER.SPECIAL.01
%                 VERSION : UNDEFINED
%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) Das Kommando SHOW-SDF-OPTIONS gibt Informationen über alle aktivierten Syntaxdateien und die eingestellten SDF-Optionen. Die Ausgabe der Subsystem-Syntaxdateien ist je nach Anzahl der aktivierten Subsysteme sehr umfangreich. Sie wird in diesem Beispiel nur auszugsweise wiedergegeben. Die Benutzersyntaxdatei SYSSDF.SDF.USER.SYNTAX ist aktiv.
- (2) Aktivieren der zusätzlichen Benutzersyntaxdatei SYSSDF.USER.SPECIAL01 mit dem Kommando MODIFY-SDF-OPTIONS.
- (3) Das Kommando SHOW-SDF-OPTIONS mit INFORMATION=\*USER informiert über die Benutzersyntaxdatei und die eingestellten SDF-Optionen.

# SHOW-SYNTAX-VERSIONS

## Versionsstände von Syntaxdateien anzeigen

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

### Funktionsbeschreibung

Das Kommando SHOW-SYNTAX-VERSIONS gibt Informationen der momentan für die Task aktivierten System-, Subsystem- und Gruppensyntaxdateien nach SYSOUT aus. Die Ausgabe enthält die Namen und Versionsstände aller momentan gültigen Syntaxbeschreibungen der eingesetzten Software-Einheiten und Komponenten.

Der Benutzer kann die Ausgabe auch für eine bestimmte bzw. eine Liste von Software-Einheiten und Komponenten anfordern.

Bei Einsatz des Subsystems SDF-P (ab Version V2.0A) kann die Ausgabe in strukturierte S-Variablen gelenkt werden.

### Format

```
SHOW-SYNTAX-VERSIONS
```

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

### Operandenbeschreibung

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

Gibt die Namen der Software-Einheiten und Komponenten an, über die Informationen ausgegeben werden sollen.

## Kommando-Returncode

(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

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

Tabelle 10: Ausgabevariablen des Kommandos SHOW-SYNTAX-VERSIONS

*Hinweis zum Inhalt der Variablen:*

\*SYS                      Bezeichnet eine System- oder Subsystem-Syntaxdatei.

## Beispiele

```
/show-syn-vers (sdf-p,lms) _____ (1)
%SYNTAX VERSION OF SDF-P           :
%FOR SYSTEM SYNTAX FILE :
%:20SH:$TSOS.SYSSDF.SDF-P.021
%-----
% SDF-P           = 02.1C100
%                 CLI           = 550
%
%SYNTAX VERSION OF LMS           :
%FOR SYSTEM SYNTAX FILE :
%:20SH:$TSOS.SYSSDF.LMS.033
%-----
% LMS             = 03.3A200
%                 LMS           = 033
%
```

```
/show-syn-vers sdf _____ (2)
%SYNTAX VERSION OF SDF           :
%FOR SYSTEM SYNTAX FILE :
%:20SH:$TSOS.SYSSDF.SDF.045
%-----
% SDF             = 04.5A300
%                 CMD           = 045
%
```

- (1) Ausgabe der Informationen über die Software-Einheiten SDF-P und LMS.
- (2) Ausgabe der Informationen über die Software-Einheit SDF.



## WRITE-TEXT

### Text auf SYSOUT bzw. SYSLST ausgeben

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

### Funktionsbeschreibung

Das Kommando WRITE-TEXT gibt den angegebenen Text auf SYSOUT bzw. auf SYSLST aus.

In Programmen mit SDF-Schnittstelle ist WRITE-TEXT mit gleicher Syntax und Funktionalität als Standardanweisung verfügbar.

### Format

<b>WRITE-TEXT</b>	Kurzname: <b>WRTX</b>
<pre> <b>TEXT</b> = '...' / &lt;c-string 1..1024 with-low&gt; , <b>OUTPUT</b> = *<b>SYSOUT</b> / *<b>SYSLST</b>(...)       *<b>SYSLST</b>(...)           <b>SYSLST-NUMBER</b> = *<b>STD</b> / &lt;integer 1..99&gt; </pre>	

### Operandenbeschreibung

**TEXT** = '...' / <c-string 1..1024 with-low>

Text, der auf SYSOUT bzw. SYSLST auszugeben ist. Voreingestellt ist eine Zeichenfolge, die ein Leerzeichen enthält.

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

Gibt an, wohin die Information auszugeben ist.

**OUTPUT = \*SYSOUT**

Die Ausgabe erfolgt in die Systemdatei SYSOUT.

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

Die Ausgabe erfolgt in die Systemdatei SYSLST.

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

Bestimmt, ob die Ausgabe in die Systemdatei SYSLST oder in eine SYSLST-Datei aus der Menge SYSLST01 bis SYSLST99 erfolgen soll.

Default-Wert ist \*STD, d.h. die Ausgabe erfolgt in die Systemdatei SYSLST.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando fehlerfrei ausgeführt

**Beispiel**

Eine Prozedurdatei (PROC. 1) beginnt mit folgenden Kommandos:

```
/BEG-PROC  PAR=*YES(PROC-PAR=(&FILE=))
/WRITE-TEXT 'Prozedur zum Binden'
...
...
...
/WRITE-TEXT 'Prozedur ohne Fehler beendet'
/END-PROC
```

Beim Prozedurablauf wird dann der im Kommando WRITE-TEXT angegebene Text nach SYSOUT ausgegeben:

```
/call-proc name=proc.1
Prozedur zum Binden
.
.
.
Prozedur ohne Fehler beendet
```

## 7.2 Standardanweisungen

Programme, die ihre Anweisungen über SDF einlesen, besitzen so genannte Standardanweisungen. Diese Anweisungen sind in ihrer Funktionsweise unabhängig von den anderen Anweisungen, die programmspezifisch angewendet werden.

Die Standardanweisungen stehen über die Basis-Systemsyntaxdatei (siehe [Seite 158](#)) allen Programmen mit SDF-Schnittstelle automatisch zur Verfügung.

Die Standardanweisungen dienen der Standardisierung der Benutzeroberfläche. Sie bieten Funktionen, die generell in Programmen für sinnvoll erachtet werden. Diese Funktionen sind deshalb auch immer mit gleichem Anweisungsnamen ansprechbar.

### END

Die Anweisung END beendet die Eingabe an das aufgerufene Programm. Die Anweisung hat keine Operanden und wird sofort ausgeführt.

END

## EXECUTE-SYSTEM-CMD

Mit der Anweisung EXECUTE-SYSTEM-CMD kann der Benutzer während des Programmlaufs ein Kommando ausführen lassen. Die Anweisung EXECUTE-SYSTEM-CMD unterbricht das aufgerufene Programm, setzt das angegebene Kommando ab und kehrt nach der Kommandoausführung in den Programmodus zurück. Fehler bei der Kommandoausführung lösen Spin-Off auf Programmebene aus (siehe Standardanweisung STEP, [Seite 234](#)).

Die Anweisung wird abgewiesen, wenn die Programmunterbrechung innerhalb einer Prozedur oder durch das Programm selbst momentan nicht zugelassen wird (z.B. aus Sicherheitsgründen).

### *Hinweis*

Die Anweisung wird ab BS2000/OSD-BC V2.0 unterstützt.

<b>EXECUTE-SYSTEM-CMD</b>
<b>CMD</b> = <text 0..1800 with-low>

**CMD = <text 0..1800 with-low>**

Auszuführendes Kommando ohne führenden Schrägstrich, eingeschlossen in runde Klammern.

## HELP-MSG-INFORMATION

Die Anweisung HELP-MSG-INFORMATION gibt den Text einer Systemmeldung auf SYSOUT aus. Zusätzlich kann der Benutzer Erläuterungen zu Meldungen anfordern und festlegen, in welcher Sprache der Text ausgegeben wird.

Die Funktionsweise entspricht der des gleichnamigen Kommandos.

### *Hinweis*

Die Operanden sind wie nachfolgend dargestellt anzugeben. Für die Operanden gelten die SDF-Abkürzungsregeln. SDF-Funktionen, wie Auskünfte über mögliche Operandenwerte oder Korrekturdialog, sind auf Operandenebene nicht verfügbar.

<b>HELP-MSG-INFORMATION</b>	Kurzname: <b>HP / HPMSGI</b>
<b>MSG-IDENTIFICATION = *LAST</b> / <alphanum-name 4..7> <b>,INFORMATION-LEVEL = *MAXIMUM / *MEDIUM / *MINIMUM</b> <b>,LANGUAGE = *STD</b> / <name 1..1>	

## HOLD-PROGRAM

Die Anweisung HOLD-PROGRAM unterbricht ein auf Eingaben wartendes Programm und ermöglicht die Eingabe von Kommandos. Das Kommando RESUME-PROGRAM beendet die Kommandoeingabe und bewirkt die Rückkehr in den Programmmodus.

Die Funktionsweise entspricht der des gleichnamigen Kommandos.

Die Anweisung wird abgewiesen, wenn die Programmunterbrechung innerhalb einer Prozedur oder durch das Programm selbst momentan nicht zugelassen wird (z.B. aus Sicherheitsgründen).

### *Hinweis*

Die Anweisung wird ab BS2000/OSD-BC V2.0 unterstützt.

<b>HOLD-PROGRAM</b>

## MODIFY-SDF-OPTIONS

Mit der Anweisung MODIFY-SDF-OPTIONS kann der Benutzer während des Programmlaufs eine Benutzersyntaxdatei aktivieren bzw. deaktivieren und SDF-Einstellungen verändern.

Die Funktionsweise entspricht der des gleichnamigen Kommandos. Der Testmodus und die Kommandostatistik können nur auf Kommandoebene eingestellt werden.

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

## REMARK

Die Anweisung REMARK kennzeichnet den nachfolgend angegebenen Text als Kommentar und ist ausschließlich für die Dokumentation des Auftrags- bzw. des Prozedurablaufs von Bedeutung. Ein Semikolon außerhalb von Klammern wird als Anweisungstrenner interpretiert, d.h. nachfolgende Zeichen werden als eine folgende Anweisung interpretiert.

<b>REMARK</b>
<b>TEXT = &lt;cmd-rest 0..1800&gt;</b>

## RESET-INPUT-DEFAULTS

Mit der Anweisung RESET-INPUT-DEFAULTS kann der Benutzer task-spezifische Default-Werte für Anweisungen oder Kommandos löschen.

Die Funktionsweise entspricht der des gleichnamigen Kommandos. Voreingestellt ist jedoch das Löschen der task-spezifischen Default-Werte von Anweisungen.

<b>RESET-INPUT-DEFAULTS</b>	Kurzname: <b>RSID</b>
<b>OBJECT = *STMT(...)</b> / <b>*CMD (...)</b> / <b>*ALL</b> / <integer 1..9999> <b>*STMT(...)</b>   <b>STMT = *ALL</b> / <structured-name 1..30 with-wild(50)>   <b>,PROGRAM = *CURRENT</b> / <b>*ALL</b> / <structured-name 1..30> <b>*CMD(...)</b>   <b>CMD = *ALL</b> / <structured-name 1..30 with-wild(50)>	

## RESTORE-SDF-INPUT

Die Anweisung RESTORE-SDF-INPUT gibt eine bereits erfolgte Eingabe, die im Eingabepuffer gespeichert wurde, wieder am Bildschirm aus.

Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos. Voreingestellt ist jedoch die Ausgabe der zuletzt gespeicherten Anweisung.

<b>RESTORE-SDF-INPUT</b>	Kurzname: <b>RRSDFI</b>
<b>INPUT = *LAST-STMT</b> / <integer -100..-1> / <integer 1..9999>	

### **INPUT = \*LAST-STMT**

Die zuletzt gespeicherte Anweisung wird ausgegeben.

Ein Anwendungsbeispiel finden Sie bei dem gleichnamigen Kommando.

## SHOW-INPUT-DEFAULTS

Mit der Anweisung SHOW-INPUT-DEFAULTS kann sich der Benutzer über alle aktuell definierten task-spezifischen Default-Werte informieren.

Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos. Voreingestellt ist jedoch die Ausgabe der task-spezifischen Default-Werte von Anweisungen.

SHOW-INPUT-DEFAULTS	Kurzname: SHID
<b>OBJECT = *STMT(...) / *CMD(...) / *ALL</b> <b>*STMT(...)</b>   <b>STMT = *ALL</b> / <structured-name 1..30 with-wild(50)>   <b>,PROGRAM = *CURRENT</b> / *ALL / <structured-name 1..30> <b>*CMD(...)</b>   <b>CMD = *ALL</b> / <structured-name 1..30 with-wild(50)> <b>,OUTPUT = *SYSOUT</b> / *SYSLST(...) <b>*SYSLST(...)</b>   <b>SYSLST-NUMBER = *STD</b> / <integer 1..99> <b>,INPUT-SERIAL-NUMBER = *NO</b> / *YES	

## SHOW-INPUT-HISTORY

Mit der Anweisung SHOW-INPUT-HISTORY kann sich der Benutzer den Inhalt des Eingabepuffers nach SYSOUT ausgeben lassen. Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos, mit der Ausnahme, dass die Ausgabe von Anweisungen voreingestellt ist.

SHOW-INPUT-HISTORY	Kurzname: SHIH
<b>ENTRIES = 8</b> / <integer 1..100> / *ALL <b>,SELECT = *STMT</b> / *CMD / *ALL <b>,PATTERN = *NONE</b> / <structured-name 1..30> <b>,INPUT-SERIAL-NUMBER = *NO</b> / *YES	



## SHOW-SDF-OPTIONS

Mit der Anweisung SHOW-SDF-OPTIONS kann der Benutzer aktuelle Informationen über alle aktivierten Syntaxdateien und SDF-Einstellungen für seine Task abfragen. Hat das Programm eine eigene Syntaxdateihierarchie eröffnet, werden die Namen dieser Syntaxdateien angezeigt.

Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos.

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

## SHOW-STMT

Mit der Anweisung SHOW-STMT kann sich der Benutzer die Syntaxbeschreibung einer Anweisung des aktuell geladenen Programmes anzeigen lassen. Mit STMT-NAME=\*ALL erhält der Benutzer eine Liste aller Anweisungen. Bei Verwendung von Musterzeichen wird eine Liste der entsprechenden Anweisungen ausgegeben.

Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos, mit dem Unterschied, dass die Syntaxbeschreibung von Anweisungen statt von Kommandos ausgegeben werden (der Operand STMT-NAME entspricht dabei dem Operanden CMD-NAME).

<b>SHOW-STMT</b>
<b>STMT-NAME = *ALL / &lt;structured-name 1..30 with-wild&gt;</b> <b>,INFORMATION = *MINIMUM / *MEDIUM / *MAXIMUM</b> <b>,FORM = *GUIDED / *UNGUIDED</b> <b>,CHECK-PRIVILEGES = *YES / *NO</b> <b>,OUTPUT = *SYSOUT / *SYSLST(...)</b> <b>*SYSLST(...)</b>   <b>SYSLST-NUMBER = *STD / &lt;integer 1..99&gt;</b>   <b>,LINES-PER-PAGE = *STD / *UNLIMITED / &lt;integer 1..200&gt;</b>

## STEP

Die Anweisung STEP kennzeichnet einen Abschnitt von Programmanweisungen innerhalb einer Kommandodatei. Bei einer fehlerhaften Anweisung wird der Spin-Off-Mechanismus ausgelöst. Das bedeutet, dass alle nachfolgenden Anweisungen bis zu einem solchen Abschnitt ignoriert werden. Wird vor der END-Anweisung keine STEP-Anweisung gefunden, so erhält das Programm einen Returncode, auf den es mit einer abnormalen Beendigung reagieren kann. Der Spin-Off setzt sich bei abnormaler Beendigung auf Kommandoebene fort (siehe Kommando SET-JOB-STEP im Handbuch „Kommandos, Band 4“ [1]).

<b>STEP</b>

## WRITE-TEXT

Die Anweisung WRITE-TEXT gibt einen angegebenen Text auf SYSOUT aus. Die Funktion der Anweisung entspricht der des gleichnamigen Kommandos.

<b>WRITE-TEXT</b>	Kurzname: <b>WRTX</b>
<b>TEXT</b> = ' <u>  </u> ' / <c-string 1..1024 with-low> <b>,OUTPUT</b> = <b>*SYSOUT</b> / <b>*SYSLST</b> (...) <b>*SYSLST</b> (...)   <b>SYSLST-NUMBER</b> = <b>*STD</b> / <integer 1..99>	

## 7.3 SDF-I-Anweisungen

Das Dienstprogramm SDF-I ist umfassend in dem Handbuch „SDF-Verwaltung“ [5] beschrieben. Das Programm wird mit dem Kommando START-SDF-I gestartet. Nachfolgend werden nur die zwei Anweisungen CONVERT-SYNTAX-FILE und SHOW-SYNTAX-FILE beschrieben, da sie auch Funktionen für den nicht privilegierten Benutzer zur Bearbeitung seiner Benutzersyntaxdateien enthalten. Benutzersyntaxdateien, die mit einer SDF-A-Version kleiner V2.0 erstellt wurden, sollten aus Performance-Gründen in das von SDF V2.0 unterstützte neue Format konvertiert werden.

### CONVERT-SYNTAX-FILE

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>**

Name der Syntaxdatei alten Formats.

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

Name der Syntaxdatei, die im neuen Format erstellt wird.

## SHOW-SYNTAX-FILE

Gibt Informationen über eine Syntaxdatei aus. Die Anweisung gibt weitgehend dieselben Informationen aus wie das SDF-Kommando SHOW-SYNTAX-VERSIONS. Zusätzlich wird der Syntaxdateityp (SYSTEM oder GROUP) und das Format der Syntaxdatei (V1, V2, V3, V4 oder V4.1) ausgegeben.

<b>SHOW-SYNTAX-FILE</b>
<b>FILE</b> = * <u>CURRENT</u> / * <u>INPUT-FILE</u> / <filename 1..54>
<b>,INFORMATION</b> = <u>ALL-ATTRIBUTES</u> / <u>VERSIONS</u> / <u>GLOBALS</u> / <u>CMD-INTERFACE</u>
<b>,PRODUCT-NAME</b> = * <u>ALL</u> / <structured-name 1..16>

### FILE =

Name der Syntaxdatei, über die Informationen auszugeben sind.

### FILE = \*CURRENT

Der Operandenwert darf ausschließlich nach (mindestens) einer vorausgegangenen OPEN-Anweisung angegeben werden. Die Information bezieht sich unmittelbar nach einer OPEN-Anweisung auf die unter INPUT-FILE angegebene Syntaxdatei bzw. nach einer MERGE-Anweisung auf die aktuelle temporäre SDF-I-Arbeitsdatei.

### FILE = \*INPUT-FILE

Der Operandenwert darf ausschließlich nach (mindestens) einer vorausgegangenen OPEN-Anweisung angegeben werden. Die Information bezieht sich auf die Eingabe-Syntaxdatei, die bei der Anweisung OPEN unter INPUT-FILE angegeben wurde.

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

Die Information bezieht sich auf die explizit angegebene Syntaxdatei.

### INFORMATION =

Bestimmt die Art der Information, die ausgegeben wird. Alle Ausgaben enthalten den Typ (GROUP oder SYSTEM) und das Format (V1, V2, V3, V4 oder V4.1) der Syntaxdatei.

### INFORMATION = ALL-ATTRIBUTES

Die Ausgabe enthält zusätzlich die Versionen der enthaltenen Liefereinheiten (nur bei Gruppen- und System-Syntaxdateien) und die Globalinformation. Bei Syntaxdateien, die von Kunden erstellt wurden, wird die Version mit \*CUSTOM\* angezeigt.

### INFORMATION = VERSIONS

Die Ausgabe enthält zusätzlich die Versionen der enthaltenen Liefereinheiten (nur bei Gruppen- und System-Syntaxdateien). Bei Syntaxdateien, die von Kunden erstellt wurden, wird die Version mit \*CUSTOM\* angezeigt.

**INFORMATION = \*GLOBALS**

Die Ausgabe enthält zusätzlich die Globalinformation.

**INFORMATION = CMD-INTERFACE**

Die Ausgabe enthält zusätzlich eine Liste der Kommandos.

**PRODUCT-NAME =**

Dieser Operand wirkt nur bei Angabe von INFORMATION=CMD-INTERFACE und bestimmt, ob die Ausgabe der Kommandos auf ein bestimmtes Produkt beschränkt werden soll.

**PRODUCT-NAME = \*ALL**

Alle Kommandos der angegebenen Syntaxdatei werden ausgegeben.

Bei der Ausgabe aus einer von Fujitsu Siemens Computers ausgelieferten SESD (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 SESD (Softwareeinheitssyntaxdatei) oder eine INSD (Installationssyntaxdatei) eingemischt wurden, angezeigt werden, muss die beim Einmischen verwendete REMOVE-ID als Produktname angegeben werden.

## 7.4 Makroaufrufe für SDF

Für Benutzerprogramme ist die Verwendung der SDF-Bedienoberfläche ebenfalls möglich. Hierzu bietet SDF folgende Makroaufrufe, deren Anwendung in dem Handbuch „SDF-A“ [4] ausführlich beschrieben ist:

CLSCALL	schließt eine im Programm eröffnete Syntaxdateihierarchie
CMDALLW	generiert eine Liste der zulässigen Operationen
CMDANALY	generiert EQUATES für Returncodes
CMDCST	stößt einen Semantikfehlerdialog an
CMDMEM	generiert einen Übergabebereich für Statusinformationen
CMDRC	setzt einen Kommando-Returncode
CMDRETC	generiert eine Dsect für den Kommando-Returncode
CMDRST	liest und analysiert eine Anweisung und liefert die Resultate zurück
CMDSEL	erzeugt ein Auswahlmenü für definierte Objekte und liefert das Auswahlresultat zurück (möglich sind Einfach- und Mehrfachauswahl)
CMDSTA	informiert über aktivierte Syntaxdateien
CMDSTRUC	generiert für eine analysierte Anweisung einen normierten Übergabebereich im alten Format, der nur von den noch kompatibel unterstützten Makroaufrufen RDSTMT, TRSTMT und CORSTMT ausgewertet wird (neues Format siehe CMDTA).
CMDTA	generiert für eine analysierte Anweisung einen normierten Übergabebereich im neuen Format
CMDTST	analysiert eine Anweisung
CMDVAL	überprüft, ob ein Wert einem Datentyp entspricht
CMDWCC	prüft Wildcard-Syntax und führt einen Mustervergleich durch
CMDWCO	erzeugt aus einer Selektionszeichenkette, einem Namen und einer Konstruktionszeichenkette einen Namen
CORSTMT	stößt einen Semantikfehlerdialog an Der Makroaufruf verwendet den normierten Übergabebereich im alten Format und wird nur noch kompatibel unterstützt. Für Anwendungen, die den normierten Übergabebereich im neuen Format benutzen, muss der Makroaufruf CMDCST verwendet werden.
OPNCALL	eröffnet eine eigene Syntaxdateihierarchie auf Programmebene

RDSTMT	liest und analysiert eine Anweisung und liefert die Resultate zurück Der Makroaufruf verwendet den normierten Übergabebereich im alten Format und wird nur noch kompatibel unterstützt. Für Anwendungen, die den normierten Übergabebereich im neuen Format benutzen, muss der Makroaufruf CMDRST verwendet werden.
TRCMD	analysiert eine Kommandoeingabe und generiert daraus das gewünschte LOGGING-Format (vgl. <a href="#">Abschnitt „Protokollierung der Eingabe“ auf Seite 100</a> )
TRSTMT	analysiert eine Anweisung Der Makroaufruf verwendet den normierten Übergabebereich im alten Format und wird nur noch kompatibel unterstützt. Für Anwendungen, die den normierten Übergabebereich im neuen Format benutzen, muss der Makroaufruf CMDTST verwendet werden.

## 7.5 SDF-Schnittstelle zu höheren Programmiersprachen

Ab Version V3.0 bietet SDF eine Schnittstelle für Programme in höheren Programmiersprachen an (**H**igh **L**evel **L**anguages). Programme, die in den Programmiersprachen COBOL, FORTRAN oder C geschrieben werden, können die Funktionen der SDF-Makroaufrufe direkt nutzen, d.h. es müssen keine Assembler-Unterprogramme verwendet werden, um das Programm mit einer SDF-Bedienoberfläche auszustatten. SDF stellt geeignete Funktionsaufrufe zur Verfügung, die die Funktionen der SDF-Makroaufrufe RDSTMT, CORSTMT, TRSTMT, CMDRC, CMDSTA und CMD unterstützen. Zusätzlich bietet SDF Funktionen, die die Analyse des normierten Übergabebereiches erleichtern.

Die Schnittstelle zu höheren Programmiersprachen ist ausführlich in dem Handbuch „SDF-A“ [\[4\]](#) beschrieben.



Die Schnittstelle zu höheren Programmiersprachen unterstützt nur den normierten Übergabebereich im alten Format. Die Nutzung des neuen Formats ist nur über die Assembler-Schnittstelle möglich!





---

## 8 Strukturierte Ausgabe in S-Variablen

Die Informationen von SHOW-Kommandos werden im Dialog am Terminal ausgegeben, wenn die Systemdatei SYSOUT keine andere Zuordnung besitzt. Zusätzlich wird bei einigen SHOW-Kommandos die Ausgabe in die Systemdatei SYSLST oder in eine katalogisierte Datei angeboten. Die Gesamtinformation ist jedoch nur für die Bildschirmausgabe aufbereitet.

Auf einzelne Teile dieser Gesamtinformation kann der Benutzer in Prozeduren nicht direkt zugreifen. Er muss die Gesamtinformation durch Umlenkung von SYSOUT in eine Datei (bzw. S-Variablen-Liste) abspeichern (siehe Handbuch „Kommandos, Band 1“ [1], Kommando ASSIGN-SYSOUT). Jede Bildschirmzeile entspricht dabei einem unstrukturierten Datensatz (bzw. Listenelement), der sowohl Einzelinformationen als auch konstante Bezeichner enthalten kann. Aktuelle Einzelinformationen können nur bei genauer Kenntnis des Ausgabelayouts durch Stringverarbeitung (z.B. mit EDT-Prozedur oder Builtin-Funktionen) selektiert werden.

Da sich SYSOUT-Ausgaben in Folgeversionen ändern können (z.B. zusätzliche Informationen oder neuer Aufbau), müssen diese Prozeduren ständig angepasst werden.

Ab SDF-P V2.0 können SHOW-Kommandos ihre Informationen in zusammengesetzte S-Variable vom Typ Struktur ausgeben. Der Benutzer kann direkt auf Einzelinformationen zugreifen. Jedes SHOW-Kommando mit dieser Funktionalität gibt den Aufbau der Struktur vor:

- Für ein im SHOW-Kommando angegebenes Objekt (z.B. Datei, Gerät) wird eine Struktur definiert. Werden mehrere Objekte angegeben (z.B. als Musterzeichenfolge), wird eine Liste von Strukturen angelegt.
- Für jede Einzelinformation zu diesem Objekt wird eine S-Variable als Element dieser Struktur definiert und die Einzelinformation als Inhalt zugewiesen.
- Lassen sich Informationen zu einem Objekt hierarchisch weiter untergliedern, wird für jede Hierarchie eine zusammengesetzte S-Variable als Element der übergeordneten Struktur definiert. Eine hierarchisch untergeordnete S-Variable kann dabei eine einfache S-Variable, eine Struktur oder Liste einfacher S-Variablen und/oder Strukturen sein.
- Die Namen der Elemente sind für das jeweilige SHOW-Kommando vorgegeben. Sie entsprechen, soweit möglich, korrespondierenden Operandennamen bzw. einer eindeutigen Abkürzung. Innerhalb aller SHOW-Kommandos werden Elemente, die gleiche Informationen enthalten, konsistent benannt.

- Die Inhalte der S-Variablen (die Einzelinformationen) entsprechen, soweit möglich, korrespondierenden Operandenwerten bzw. eindeutigen Abkürzungen.
- Die S-Variablen besitzen einen festgelegten Typ: String, Integer oder Boolean.

Ab SDF V4.0 bieten die Kommandos SHOW-SDF-OPTIONS, SHOW-SYNTAX-VERSIONS und das privilegierte Kommando SHOW-SDF-PARAMETERS diese Funktionalität. Die Variablenstruktur kann der jeweiligen Kommandobeschreibung entnommen werden. Weitere Subsysteme werden diese Funktionalität in zukünftigen Versionen ebenfalls anbieten.

Der Benutzer kann strukturierte Systemausgaben erzeugen und auswerten, wenn das kostenpflichtige Subsystem SDF-P (ab Version V2.0) geladen ist. Dabei ist wie folgt vorzugehen:

### 1. S-Variable deklarieren

Der Benutzer deklariert eine zusammengesetzte S-Variable vom Typ Struktur, wobei die Struktur als Element einer Liste definiert wird. Den Namen kann der Benutzer im Rahmen der SDF-P-Namenskonventionen frei wählen. Die Struktur sollte dynamisch erweiterbar sein (Voreinstellung).

```
/DECLARE-VARIABLE NAME=USERVAR(TYPE=*STRUCTURE),MULTIPLE-ELEMENTS=*LIST
```

Wird die Struktur dagegen nur statisch angelegt, kann der Benutzer nur Informationsteile erhalten, für die er auch Strukturelemente explizit mit dem jeweils festgelegten Namen deklariert hat (siehe Handbuch „SDF-P“ [6]).

### 2. Strukturierte Ausgabe erzeugen

Für strukturierte Ausgaben einzelner Kommandos ruft der Benutzer mit EXECUTE-CMD ein SHOW-Kommando auf und gibt an, dass die strukturierte Systemausgabe in die bereits deklarierte Struktur gelenkt werden soll.

```
/EXECUTE-CMD CMD =( SHOW-SYNTAX-VERSIONS SOFTWARE-UNIT-NAME=(JV, LMS) ), -  
STRUCTURED-OUTPUT = USERVAR ,TEXT-OUTPUT = *NONE
```

Die Ausgabe nach SYSOUT wird dabei mit TEXT-OUTPUT=\*NONE unterdrückt.

Statt EXECUTE-CMD kann der Benutzer auch den Systemausgabestrom für strukturierte Systemausgaben SYSINF mit dem Kommando ASSIGN-STREAM der deklarierten Struktur zuordnen. Während der Zuordnung wird jedoch für jedes abgesetzte Kommando, das strukturierte Systemausgabe unterstützt, die S-Variable entsprechend erweitert. Mit EXECUTE-CMD fordert der Benutzer dagegen explizit die Ausgabe des angegebenen Kommandos an. Zu SYSINF siehe auch Handbuch „SDF-P“ [6].

## 3. Inhalt der S-Variable ausgeben

```
/SHOW-VARIABLE USERVAR
```

```

/show-var uservar
USERVAR(*LIST).F-NAME = :20SH:$TSOS.SYSSDF.JV.130
USERVAR(*LIST).TYPE = *SYS
USERVAR(*LIST).SW-UNIT(*LIST).NAME = JV
USERVAR(*LIST).SW-UNIT(*LIST).VERSION = 13.0C800
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = JVS
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).VERSION = 320
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = CJC
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).VERSION = 200
USERVAR(*LIST).F-NAME = :20SH:$TSOS.SYSSDF.LMS.033
USERVAR(*LIST).TYPE = *SYS
USERVAR(*LIST).SW-UNIT(*LIST).NAME = LMS
USERVAR(*LIST).SW-UNIT(*LIST).VERSION = 03.3A200
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = LMS
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).VERSION = 033
/

```

Diagramm zur Strukturierung der Ausgabe:

- (1) Umfasst die gesamte Ausgabe.
- (2) Umfasst die Struktur für die Softwareeinheit LMS.
- (3) Umfasst die Struktur für die Softwareeinheit JV.
- (4) Umfasst die Struktur für die Komponente JVS.
- (5) Umfasst die Struktur für die Komponente CJC.

## Erläuterung der Ausgabe:

Die vom Benutzer definierte S-Variable (hier USERVAR) enthält die Gesamtausgabe. Die Zeichenfolge „(\*LIST)“ gibt an, dass die S-Variable mehrere Elemente in einer Liste enthalten kann. Die S-Variable USERVAR enthält 2 Elemente:

Einmal die Struktur für Informationen über die Softwareeinheit JV, in der Ausgabe mit (1) gekennzeichnet, und einmal die Struktur für Informationen über die Softwareeinheit LMS, in der Ausgabe mit (2) gekennzeichnet.

Die Information für eine Softwareeinheit besteht aus den Elementen F-NAME, TYPE und SW-UNIT, wobei SW-UNIT selbst wieder eine Liste enthalten kann. Im Falle der Softwareeinheit JV enthält SW-UNIT ein Listenelement, nämlich die Struktur, die aus den Elementen NAME, VERSION und COMPONENT gebildet wird (in der Ausgabe mit (3) gekennzeichnet).

COMPONENT kann wiederum eine Liste sein: Die Softwareeinheit JV besteht aus 2 Komponenten, d.h. COMPONENT enthält 2 Listenelemente. Jedes Listenelement ist eine Struktur mit den Elementen NAME und VERSION. In der Ausgabe sind die Komponenten CJC und JVS mit (4) bzw. (5) gekennzeichnet.

### 4. Auf Einzelinformationen zugreifen

Auf einzelne Informationen kann der Benutzer über den Namen der S-Variablen zugreifen. Der anzugebende Name setzt sich wie folgt zusammen:

uservar#i.element

Dabei bedeuten:

- uservar Name der Struktur, die vom Benutzer deklariert wurde
- #i i-tes Element der Liste  
Für i=1 kann „i“ entfallen, d.h. es wird nur „#“ angegeben.
- Punkt Begrenzer im Namen von zusammengesetzten S-Variablen.
- element vordefinierter Name des Strukturelements *element* kann wiederum eine zusammengesetzte Variablen sein:  
z.B. uservar#.SW-UNIT#.NAME

Der Benutzer kann sich den Inhalt z.B. mit „SHOW-VARIABLE uservar#.element“ anzeigen lassen oder mittels Variablenersetzung weiterverwenden:

```
/show-var (uservar#.sw-unit#.component#1.name,uservar#.sw-unit#.component#2.name)
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = CJC
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = JVS

/write-text '*** SW-UNIT &(uservar#2.sw-unit#.name) -
/mit der Version &(uservar#2.sw-unit#.version) ***'
*** SW-UNIT LMS mit der Version 03.3A200 ***

/show-var uservar#2
USERVAR(*LIST).F-NAME = :20SH:$TSOS.SYSSDF.LMS.033
USERVAR(*LIST).TYPE = *SYS
USERVAR(*LIST).SW-UNIT(*LIST).NAME = LMS
USERVAR(*LIST).SW-UNIT(*LIST).VERSION = 03.3A200
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).NAME = LMS
USERVAR(*LIST).SW-UNIT(*LIST).COMPONENT(*LIST).VERSION = 033
```

## Beispiele

*Aktuelle SDF-Optionen für späteres Zurücksetzen abspeichern*

Inhalt der Prozedurdatei PROC.RESET-SDF-OPT:

```

    /DECL-PAR  OLD-SDF-OPT ( INIT=*PROMPT )
  /
    "S-Variable mit SDF-Optionen prompten"
  /WRITE-TEXT 'uebernahme der task-lokalen SDF-Optionen aus S-Variable'
  /WRITE-TEXT 'Bitte Name der S-Variable eingeben: '
  /REMARK    &(OLD-SDF-OPT)
  /
    "enthaelt OLD-SDF-OPT Namen einer Benutzer-Syntaxdatei? "
  /
  /IMPORT-VAR  &(OLD-SDF-OPT)
  /SHOW-VAR   &(OLD-SDF-OPT)
  /DECL-VAR   SDF-OPT (TYPE=*STRUCTURE(DEF=*DYN)),MULT-ELEM=*LIST, -
  /           CONTAINER = *VAR( &(OLD-SDF-OPT) )
  /
  /SF-USER: IF    ( IS-DECLARED('SDF-OPT#.SF'))
  /
    "Information ueber Syntaxdateien vorhanden"
  /
    "aktuell aktivierte Benutzersyntaxdatei deaktivieren:"
  /
    MOD-SDF-OPT SYNTAX-FILE = *NONE
  /
    DECL-VAR SF-VAR(TYPE=*STRUC) "Laufvariable definieren"
  /
    FOR1: FOR SF-VAR = *LIST( SDF-OPT#.SF)
  /
      IF ( SF-VAR.TYPE = '*USER' )
  /
          IF ( SF-VAR.F-NAME = '' )
  /
              "es war keine Benutzersyntaxdatei aktiviert"
  /
              USER-SF = '*NONE'
  /
              CYCLE FOR1 "FOR-Schleife beenden"
  /
          ELSE
  /
              "es war mindestens eine Benutzersyntaxdatei aktiviert"
  /
              MOD-SDF-OPT SYNTAX-FILE = *ADD( &(SF-VAR.F-NAME) )
  /
          END-IF
  /
      END-IF
  /
      "System- oder Gruppensyntaxdatei! kein Aktion"
  /
      " die SDF-Optionen wurden mit INF=*ALL gespeichert!"
  /
    END-FOR
  /ELSE
    "keine Information ueber Syntaxdateien"
  /
    "es war also keine Benutzersyntaxdatei aktiviert"
  /
    MOD-SDF-OPT SYNTAX-FILE = *NONE
  /END-IF
  /
  /
    "die gespeicherten SDF-Optionen setzen"
  /MOD-SDF-OPT SYNTAX-FILE = *UNCHANGED -
  /
    ,GUIDANCE          = &(SDF-OPT#.GUIDE) -
  /
    ,LOGGING           = &(SDF-OPT#.LOG) -
  /
    ,UTILITY-INTERFACE= &(SDF-OPT#.UTILITY-INTERF) -
  /
    ,PROCEDURE-DIALOG = &(SDF-OPT#.PROC-DIALOG) -

```

```
/          ,CONTINUATION      = &(SDF-OPT#.CONTI) -
/          ,MENU-LOGGING     = &(SDF-OPT#.MENU-LOG) -
/          ,MODE              = &(SDF-OPT#.MODE) -
/          ,TEST-PROGRAM-NAME= &(SDF-OPT#.TEST-PROG-NAME) -
/          ,FUNCTION-KEYS    = &(SDF-OPT#.FUNC-KEY) -
/          ,INPUT-HISTORY    = &(SDF-OPT#.INPUT-HIST)
/WRITE-TEXT `SDF-Optionen zurueckgesetzt!`
/SHOW-SDF-OPT INF = *USER
```

Der Benutzer speichert die aktuellen SDF-Optionen in der S-Variablen SDF-OPT-1. Vorher deklariert der Benutzer die S-Variable SDF-OPT-1. Um aus einer Prozedur darauf zugreifen zu können muss er bei der Deklaration zusätzlich SCOPE=\*TASK vereinbaren:

```
/decl-var sdf-opt-1( type=*struc ), mult-elem=*list, scope=*task

/exec-cmd (show-sdf-opt inf=*user),struct-out=sdf-opt-1,text-out=*none

/show-var sdf-opt-1
SDF-OPT-1(*LIST).SF(*LIST).F-NAME = :20SG:$USER1.SYSSDF.USER.SPECIAL.01
SDF-OPT-1(*LIST).SF(*LIST).TYPE = *USER
SDF-OPT-1(*LIST).SF(*LIST).VERSION = UNDEFINED
SDF-OPT-1(*LIST).SF(*LIST).F-NAME = :20SG:$USER1.SYSSDF.USER.SPECIAL.02
SDF-OPT-1(*LIST).SF(*LIST).TYPE = *USER
SDF-OPT-1(*LIST).SF(*LIST).VERSION = UNDEFINED
SDF-OPT-1(*LIST).GUIDE = *EXPERT
SDF-OPT-1(*LIST).LOG = *INPUT-FORM
SDF-OPT-1(*LIST).CONTI = *NEW-MODE
SDF-OPT-1(*LIST).UTILITY-INTERF = *NEW-MODE
SDF-OPT-1(*LIST).PROC-DIALOG = *NO
SDF-OPT-1(*LIST).MENU-LOG = *NO
SDF-OPT-1(*LIST).CMD-STATIS =
SDF-OPT-1(*LIST).MODE = *EXEC
SDF-OPT-1(*LIST).CHECK-PRIV = *YES
SDF-OPT-1(*LIST).TEST-PROG-NAME = *NONE
SDF-OPT-1(*LIST).FUNC-KEY = *STYLE-GUIDE-MODE
SDF-OPT-1(*LIST).INPUT-HIST = *ON
SDF-OPT-1(*LIST).NUM-OF-INPUT = 20
SDF-OPT-1(*LIST).PASSWORD-PROT = *YES
/...
```

Während der weiteren Bearbeitung werden einige der SDF-Optionen verändert (z.B. die Benutzer-Syntaxdateien deaktiviert und eine andere aktiviert). Mit Aufruf der Prozedur PROC.RESET-SDF-OPT erreicht der Benutzer die ursprünglichen Einstellungen:

```

/show-sdf-opt inf=*user
% USER      : :20SG:$USER1.SYSSDF.USER.SPECIAL.03
%          VERSION : UNDEFINED
%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 : LMS
% FUNCTION-KEYS : *STYLE-GUIDE-MODE
% INPUT-HISTORY : *ON
% NUMBER-OF-INPUTS : 99
% PASSWORD-PROTECTION: *YES

/call-proc proc.reset-sdf-opt,log=*yes

```

### Ablaufprotokoll:

```

/call-proc proc.reset-sdf-opt,log=*yes
%      1 1 /DECL-PAR OLD-SDF-OPT ( INIT=*PROMPT )
%      4 1 /WRITE-TEXT 'Uebernahme der task-lokalen SDF-Optionen aus S-Variable'
Uebernahme der task-lokalen SDF-Optionen aus S-Variable
%      5 1 /WRITE-TEXT 'Bitte Name der S-Variable eingeben: '
Bitte Name der S-Variable eingeben:
%OLD-SDF-OPT: sdf-opt-1
%OLD-SDF-OPT: sdf-opt-1
%      6 1 /REMARK      SDF-OPT-1
%      12 1 /IMPORT-VAR SDF-OPT-1
%      13 1 /SHOW-VAR  SDF-OPT-1
SDF-OPT-1(*LIST).SF(*LIST).F-NAME = :20SG:$USER1.SYSSDF.USER.SPECIAL.01
SDF-OPT-1(*LIST).SF(*LIST).TYPE = *USER
SDF-OPT-1(*LIST).SF(*LIST).VERSION = UNDEFINED
SDF-OPT-1(*LIST).SF(*LIST).F-NAME = :20SG:$USER1.SYSSDF.USER.SPECIAL.02
SDF-OPT-1(*LIST).SF(*LIST).TYPE = *USER
SDF-OPT-1(*LIST).SF(*LIST).VERSION = UNDEFINED
SDF-OPT-1(*LIST).GUIDE = *EXPERT
SDF-OPT-1(*LIST).LOG = *INPUT-FORM
SDF-OPT-1(*LIST).CONTI = *NEW-MODE
SDF-OPT-1(*LIST).UTILITY-INTERF = *NEW-MODE
SDF-OPT-1(*LIST).PROC-DIALOG = *NO
SDF-OPT-1(*LIST).MENU-LOG = *NO
SDF-OPT-1(*LIST).CMD-STATIS =
SDF-OPT-1(*LIST).MODE = *EXEC
SDF-OPT-1(*LIST).CHECK-PRIV = *YES
SDF-OPT-1(*LIST).TEST-PROG-NAME = *NONE
SDF-OPT-1(*LIST).FUNC-KEY = *STYLE-GUIDE-MODE
SDF-OPT-1(*LIST).INPUT-HIST = *ON
SDF-OPT-1(*LIST).NUM-OF-INPUT = 20

```

```

SDF-OPT-1(*LIST).PASSWORD-PROT = *YES
%      14 1 /DECL-VAR SDF-OPT (TYPE=*STRUCTURE(DEF=*DYN)),MULT-ELEM=*LIST,
CONTAINER = *VAR( SDF-OPT-1 )
%      17 1 /SF-USER:
%      17 1 / IF ( IS-DECLARED('SDF-OPT#.SF'))
%      20 1 /MOD-SDF-OPT SYNTAX-FILE = *NONE
%      21 1 /DECL-VAR SF-VAR(TYPE=*STRUC) "Laufvariable definieren"
%      22 1 /FOR1:
%      22 1 / FOR SF-VAR = *LIST( SDF-OPT#.SF)
%      23 1 /IF ( SF-VAR.TYPE = '*USER' )
%      24 1 /IF ( SF-VAR.F-NAME = '' )
%      28 1 /ELSE
%      30 1 /MOD-SDF-OPT SYNTAX-FILE = *ADD( :20SG:$USER1.SYSSDF.USER.SPECI
AL.01 )
%      31 1 /END-IF
%      32 1 /END-IF
%      35 1 /END-FOR
%      23 1 /IF ( SF-VAR.TYPE = '*USER' )
%      24 1 /IF ( SF-VAR.F-NAME = '' )
%      28 1 /ELSE
%      30 1 /MOD-SDF-OPT SYNTAX-FILE = *ADD( :20SG:$USER1.SYSSDF.USER.SPECI
AL.02 )
%      31 1 /END-IF
%      32 1 /END-IF
%      35 1 /END-FOR
%      39 1 /END-IF
%      42 1 /MOD-SDF-OPT SYNTAX-FILE = *UNCHANGED ,GUIDANCE
= *EXPERT ,LOGGING = *INPUT-FORM ,UTILI
TY-INTERFACE= *NEW-MODE ,PROCEDURE-DIALOG = *NO ,CONTINU
ATION = *NEW-MODE ,MENU-LOGGING = *NO ,MODE
= *EXEC ,TEST-PROGRAM-NAME= *NONE ,FUNCTION-KEYS
= *OLD-MODE ,INPUT-HISTORY = *ON
%      53 1 /WRITE-TEXT 'SDF-Optionen zurueckgesetzt!'
SDF-Optionen zurueckgesetzt!
%      54 1 /SHOW-SDF-OPT INF = *USER
% USER : :20SG:$USER1.SYSSDF.USER.SPECIAL.01
% VERSION : UNDEFINED
% USER : :20SG:$USER1.SYSSDF.USER.SPECIAL.02
% VERSION : UNDEFINED
%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 : 99
% PASSWORD-PROTECTION: *YES
% 1 /EXIT-PROCEDURE ERROR=*NO
/

```



*Beispiele für den Zugriff auf unterschiedliche Listenelemente*

```
/decl-var syn1(type=*struct),mult-elem=*list
```

```
/exec-cmd (show-syntax-versions (archive,acs)),struct-out=syn1,text-out=*none
```

Anzeigen der erzeugten Gesamtausgabe:

```
/show-var syn1,inf=*par(list-index-number=*yes)
```

```
SYN1#1.F-NAME = :20SH:$TSOS.SYSSDF.ARCHIVE.060
```

```
SYN1#1.TYPE = *SYS
```

```
SYN1#1.SW-UNIT#1.NAME = ARCHIVE
```

```
SYN1#1.SW-UNIT#1.VERSION = 06.0P600
```

```
SYN1#1.SW-UNIT#1.COMPONENT#1.NAME = ARC
```

```
SYN1#1.SW-UNIT#1.COMPONENT#1.VERSION = 560
```

```
SYN1#2.F-NAME = :20SH:$TSOS.SYSSDF.ACS.140
```

```
SYN1#2.TYPE = *SYS
```

```
SYN1#2.SW-UNIT#1.NAME = ACS
```

```
SYN1#2.SW-UNIT#1.VERSION = 14.0B100
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.NAME = ACS
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.VERSION = 016
```

```
SYN1#3.F-NAME = :20SH:$TSOS.SYSSDF.ACS.140
```

```
SYN1#3.TYPE = *SYS
```

```
SYN1#3.SW-UNIT#1.NAME = ACS
```

```
SYN1#3.SW-UNIT#1.VERSION = 14.0B100
```

```
SYN1#3.SW-UNIT#1.COMPONENT#1.NAME = ACS
```

```
SYN1#3.SW-UNIT#1.COMPONENT#1.VERSION = 016
```

Anzeigen des zweiten Listenelements:

```
/show-var syn1#2,inf=*par(list-index-number=*yes)
```

```
SYN1#2.F-NAME = :20SH:$TSOS.SYSSDF.ACS.140
```

```
SYN1#2.TYPE = *SYS
```

```
SYN1#2.SW-UNIT#1.NAME = ACS
```

```
SYN1#2.SW-UNIT#1.VERSION = 14.0B100
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.NAME = ACS
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.VERSION = 016
```

Anzeigen des ersten Komponentennamens mit Version für das zweite Listenelement:

```
/show-var syn1#2.SW-UNIT#.COMPONENT#,inf=*par(list-index-number=*yes)
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.NAME = ACS
```

```
SYN1#2.SW-UNIT#1.COMPONENT#1.VERSION = 016
```

### *Beispiel für eine statische Struktur*

Der Benutzer will nur die aktuelle Einstellung des Führungsmodus in einer Variablen ablegen:

```
/decl-var sdf-opt-2(type=*struct(def=*by-syscmd)),mult-elem=*list
/beg-struct
/decl-elem guide
/end-struct
/exec-cmd (show-sdf-opt inf=*user),struct-out=sdf-opt-2,text-out=*none
/show-var sdf-opt-2
SDF-OPT-2(*LIST).GUIDE = *EXPERT
```

Es wird nur die explizit deklarierte Variable initialisiert. Nachteil dieser Variante ist, dass der Benutzer das gewünschte Strukturelement explizit definieren muss. Der direkte Zugriff auf dieses Element ist auch bei der dynamischen Struktur möglich.

---

## 9 Hinweise für OMNIS und CHECKPOINT/RESTART

### OMNIS

Bei Aufträgen, die über OMNIS gestartet wurden, ist vor dem LOGON grundsätzlich GUIDANCE=\*EXPERT eingestellt, unabhängig von einer evtl. anderen Festlegung in der Globalinformation der aktivierten Syntaxdateien.

Die Einstellung FUNCTION-KEYS=\*STYLE-GUIDE-MODE in den Globalinformationen wird nur aktiviert für ein 9763-Terminal.

Der Wechsel in eine andere Dialogform oder der Wechsel der Funktionstastenbelegung mit dem Kommando oder der Anweisung MODIFY-SDF-OPTIONS ist jedoch möglich.

### CHECKPOINT/RESTART

Nach einem RESTART-PROGRAM wird folgende Syntaxdatei Umgebung wiederhergestellt:

- System- und Gruppensyntaxdatei wie sie aktuell der Task zugewiesen sind
- Benutzer-Syntaxdatei, die zum Zeitpunkt des Fixpunktschreibens zugewiesen war
- programmspezifische Syntaxdatei Umgebung, die zum Zeitpunkt des Fixpunktschreibens geöffnet war, d.h. alle Syntaxdateien die ein Programm mit OPNCALL explizit geöffnet hatte.

CHECKPOINT/RESTART ist in folgenden Fällen nicht anwendbar:

- Die Version des Subsystems SDF ist verschieden von der zum Zeitpunkt des Fixpunktschreibens.
- System- oder Gruppensyntaxdatei wurden seit dem Fixpunktschreiben inkompatibel geändert.



---

# Abbildungen

Bild 1: Bildschirmaufbau . . . . .	59
Bild 2: Weg vom Anwendungsbereichs-Menü zur Kommandoausführung . . . . .	61
Bild 3: Die Rückzugsfunktionen *CANCEL, *EXIT und *EXIT-ALL im Systemmodus (geführter Dialog) . . . . .	68
Bild 4: Die Rückzugsfunktionen *CANCEL, *EXIT und *EXIT-ALL im Programmodus (geführter Dialog) . . . . .	69
Bild 5: Die Rückzugsfunktionen *CANCEL, *EXIT und *EXIT-ALL im Systemmodus (temporär geführter Dialog) . . . . .	78
Bild 6: Wirkung der Funktionen *CANCEL, *EXIT und *EXIT-ALL im Programmodus (temporär geführter Dialog) . . . . .	79
Bild 7: Kommando SET-LOGON-PARAMETERS . . . . .	118
Bild 8: Operandenfragebogen für das Kommando HELP-SDF . . . . .	120
Bild 9: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	126
Bild 10: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	127
Bild 11: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	127
Bild 12: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	128
Bild 13: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	129
Bild 14: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	129
Bild 15: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	130
Bild 16: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	131

Bild 17: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	132
Bild 18: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	133
Bild 19: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	134
Bild 20: Anwendungsbereich-Menü . . . . .	135
Bild 21: Kommando-Menü für den Anwendungsbereich SDF . . . . .	135
Bild 22: Operandenfragebogen für das Kommando MODIFY-SDF-OPTIONS . . . . .	136
Bild 23: Operanden-Unterfragebogen der Struktur *ON des MODIFY-SDF-OPTIONS- Kommandos. . . . .	137
Bild 24: Kommando-Menü für den Anwendungsbereich SDF . . . . .	138
Bild 25: Kommando-Menü für den Anwendungsbereich FILE . . . . .	139
Bild 26: Kommando-Menü für den Anwendungsbereich FILE . . . . .	139
Bild 27: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	140
Bild 28: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	141
Bild 29: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	141
Bild 30: Operanden-Unterfragebogen für die Struktur *BY-ATTRIBUTES von SHOW-FILE-ATTRIBUTES . . . . .	142
Bild 31: Kommando-Menü für den Anwendungsbereich FILE . . . . .	143
Bild 32: Kommando-Menü für den Anwendungsbereich FILE . . . . .	144
Bild 33: Kommando-Menü für den Anwendungsbereich FILE . . . . .	144
Bild 34: Kommando-Menü für den Anwendungsbereich FILE . . . . .	145
Bild 35: Kommando-Menü für den Anwendungsbereich FILE . . . . .	145
Bild 36: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	146
Bild 37: Kommando-Menü für den Anwendungsbereich FILE . . . . .	147
Bild 38: Kommando-Menü für den Anwendungsbereich FILE . . . . .	148
Bild 39: Kommando-Menü für den Anwendungsbereich FILE . . . . .	148
Bild 40: Kommando-Menü für den Anwendungsbereich FILE . . . . .	149
Bild 41: Kommando-Menü für den Anwendungsbereich FILE . . . . .	149

---

Bild 42: Operandenfragebogen für das Kommando SHOW-FILE-ATTRIBUTES . . . . .	150
Bild 43: Anwendungsbereichs-Menü . . . . .	151
Bild 44: Anwendungsbereichs-Menü . . . . .	152
Bild 45: Anwendungsbereichs-Menü . . . . .	152
Bild 46: Kommando-Menü für den Anwendungsbereich SDF . . . . .	153
Bild 47: Kommando-Menü für den Anwendungsbereich SDF . . . . .	153
Bild 48: Operandenfragebogen für das Kommando SHOW-SDF-OPTIONS . . . . .	154
Bild 49: Kommando-Menü für den Anwendungsbereich SDF . . . . .	155
Bild 50: Arbeitsweise von SDF bei Aktivierung verschiedener Syntaxdateitypen . . . . .	162





---

# Tabellen

Tabelle 1: Metasyntax . . . . .	31
Tabelle 2: Datentypen . . . . .	33
Tabelle 3: Zusätze zu Datentypen . . . . .	39
Tabelle 4: Funktionstastenbelegung (Old-Modus) im ungeführten Dialog . . . . .	54
Tabelle 5: Funktionstastenbelegung (Style-Guide-Modus) im ungeführten Dialog . . . . .	54
Tabelle 6: Funktionstastenbelegung (Old-Modus) im geführten Dialog . . . . .	65
Tabelle 7: Funktionstastenbelegung (Style-Guide-Modus) im geführten Dialog . . . . .	66
Tabelle 8: Ausgabeformat des Kommandos SHOW-SDF-OPTIONS . . . . .	217
Tabelle 9: Ausgabevariablen des Kommandos SHOW-SDF-OPTIONS . . . . .	219
Tabelle 10: Ausgabevariablen des Kommandos SHOW-SYNTAX-VERSIONS . . . . .	223



---

# Literatur

Wenden Sie sich zum Bestellen von Handbüchern bitte an Ihre zuständige Geschäftsstelle.

- [1] **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

- [2] **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 V2.2A

*Bestellnummer*

U23165-J-Z125-4

- [3] **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

- [4] **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

- [5] **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.

*Bestellnummer*

U2622-J-Z125-10

- [6] **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-Z125-4

[7] **SDF-CONV V3.0A** (BS2000/OSD)

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an alle BS2000-Anwender.

*Inhalt*

Prozedurformat und Kommandosprache von Prozeduren können wie folgt umgewandelt werden:

- von ISP- in SDF-Kommandosprache
- von Nicht-S- in S-Prozedurformat
- gleichzeitige Umwandlung von Kommandosprache und Prozedurformat. Der komplette Funktionsumfang von SDF-CONV V3.0A ist ab den Versionen BS2000/OSD V1.0A und SDF V4.0A verfügbar.

Zur Version SDF-CONV V3.0B existiert eine Readme-Datei.

*Bestellnummer*

U6540-J-Z125-3

[8] **POSIX** (BS200/OSD)

**POSIX-Grundlagen**

Benutzerhandbuch

*Zielgruppe*

BS2000-Systemverwalter, POSIX-Verwalter, BS2000-Benutzer, Benutzer von UNIX-/SINIX-Workstations.

*Inhalt*

Dieses Handbuch beschreibt Einführung und Arbeiten mit POSIX; BS2000-Softwareprodukte im Umfeld von POSIX; POSIX-Subsystem installieren, steuern und beenden; POSIX-Benutzer durch das BS2000 verwalten.

*Bestellnummer*

U22795-J-Z125-2

[9] **POSIX** (BS200/OSD)

**Kommandos**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an alle Benutzer der POSIX-Shell.

*Inhalt*

Das Handbuch ist ein Nachschlagewerk. Es beschreibt das Arbeiten mit der POSIX-Shell sowie die Kommandos der POSIX-Shell in alphabetischer Reihenfolge.

*Bestellnummer*

U22794-J-Z125-3

- [10] **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

- [11] **XHCS V1.3** (BS2000/OSD)  
**8-bit-Code-Verarbeitung im BS2000/OSD**  
Benutzerhandbuch

*Zielgruppe*

Anwender der Zugriffsmethoden DCAM, TIAM und *open*UTM 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*

- [12] **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

- Abkürzung
  - garantiert 95
  - mehrdeutig 103
  - von Namen 94
  - von Schlüsselworten 94
- Aliasname 32
- alphanum-name (Datentyp) 33
- Anweisung
  - Aufbau 23
  - CONVERT-SYNTAX-FILE (SDF-I) 235
  - Menü 71
  - SHOW-SYNTAX-FILE (SDF-I) 236
  - Syntaxanalyse 106
  - Testmodus 106
- Anwendungsbereich 70
- Anwendungsbereiche
  - ausgeben 9
  - Kommando 24
  - Menü 70
- Aufbau
  - SDF-Anweisung 23
  - SDF-Kommando 23
- Aufzählung von Operandenwerten 24
- Ausdrücke ersetzen 87
- Ausgabeformat
  - SHOW-SDF-OPTIONS 217
- ausgeben mit Eingabeseriennummer 211
- Auskunftssystem 8
- Auswahlmenü 102
  - mehrdeutige Abkürzung 103
  - Musterzeichen 102

## B

- Beispiel
  - Anwendungsbereiche ausgeben 9
  - Benutzerkennwort ändern 10
  - Dialogtask beenden 19
  - Dialogtask starten 19
  - Kommando rekonstruieren 18
  - LOGON-Prozedur 14
    - task-lokale SDF-Optionen abfragen 12
- Benutzerkennwort ändern 10
- Benutzersyntaxdatei 160
  - zuweisen siehe MODIFY-SDF-OPTIONS 180
- Bildschirmaufbau 59
- Builtin-Funktion 5

## C

- Cancel-Funktion 55, 67
- cat (Zusatz zu Datentypen) 44
- cat-id (Datentyp) 33
- CCS(Coded Character Set) 26
- CHECKPOINT/RESTART 251
- CLIGET 85
- CLISSET 85
- Codetabelle 26
- command-rest (Datentyp) 33
- compl (Zusatz zu Datentypen) 39
- composed-name (Datentyp) 33
- CONVERT-SYNTAX-FILE (Anweisung) 235
- corr (Zusatz zu Datentypen) 44, 45
- c-string (Datentyp) 33

## D

- date (Datentyp) 33
- Datentyp 23

- Datentypen SDF 29, 33
  - Zusätze 30
- Default-Wert 25, 96
  - für Kommandos und Anweisungen (task-spezifisch) 208
  - task-spezifisch 110
  - übernehmen 96
- Default-Wert (task-spezifisch) 211
  - ausgeben siehe SHOW-INPUT-DEFAULTS 208
  - definieren 111
  - definieren im Dialog 110
  - definieren in Prozedur 111
  - löschen siehe RESET-INPUT-DEFAULTS 195
- device (Datentyp) 33
- Dialogbetrieb, Eingabe 1
- Dialogblock 5
- Dialogführung
  - abbrechen 77
  - Beispiel 117
  - Bildschirmaufbau 59
  - Eingabe der Operanden 72
  - einstellen 52
  - Einstellung der Sprache 2
  - EXPERT-Modus 51
  - Funktionstasten 65
  - für ISP-Kommando 76
  - maximale Stufe 57
  - mehrsprachig 2
  - Menü 60
  - minimale Stufe 58
  - mittlere Stufe 58
  - NO-Form 52
  - Operandenfragebogen 60
  - Steueranweisungen 63
  - steuern 183
  - Stufen 57
  - temporär 76
- Dialogtask
  - beenden 19
  - starten 19
- E**
- Eingabe
  - abkürzen 7
  - Abkürzung 94
  - als Schlüsselwortoperand 27, 97
  - als Stellungsoperand 27, 97
  - aus einer Prozedur 1
  - aus Jobvariablen 87
  - aus Prozedurdatei 80
  - dunkelgesteuert 53
  - einer Liste 27
  - explizit 96
  - Fortsetzungszeile 80, 86
  - Funktionstasten 65
  - geblockt 55
  - geheime Operanden 188
  - ignorieren 107
  - im Dialogbetrieb 1
  - im Menümodus 7
  - im Stapelbetrieb 1, 86
  - im Zeilenmodus 7
  - implizit 96
  - in die NEXT-Zeile 60, 63
  - ISP-Kommando 76
  - Korrekturdialog 52
  - maximale Länge 28
  - Operandenfragebogen 72
  - sicherheitsrelevant 105
  - speichern 104, 180
  - spezielle 53, 61
  - Steueranweisung 60
  - ungeführter Dialog 51
  - verdeckte 1
  - von Anweisungen 26
  - von Kommandos 26
  - von Kommentaren 27
  - von Leerzeichen 27
  - von Strukturen 98
  - wiederherstellen 1
  - wiederverwenden 104
  - zurückholen siehe RESTORE-SDF-INPUT 199

Eingabepuffer 8, 104, 187, 212  
 ausgeben siehe SHOW-INPUT-HISTORY 212  
 geheime Eingaben schützen 188  
 löschen 189

Eingabeseriennummer 104, 195, 200, 208, 211, 212

END (Standardanweisung) 227

Ersetzung  
 von Ausdrücken 87  
 von Jobvariablen 88  
 von Prozedurparametern 87  
 von S-Variablen-Ausdrücken 89

erweiterter Zeichensatz 26

Event 83

Execute-Funktion 66, 67

EXECUTE-SYSTEM-CMD  
 (Standardanweisung) 228

Exit-all-Funktion 54, 65, 66

Exit-Funktion 54, 65, 66

EXPERT-Modus 51

**F**

Fehlerdialog  
 inhaltlicher Fehler 199  
 Syntaxfehler 185

Fehlerkorrekturdialog 8

filename (Datentyp) 34

fixed (Datentyp) 33

Folgezeilenbehandlung  
 NEW-Modus 80, 86  
 OLD-Modus 80, 86

Fortsetzungszeile 28, 80, 86

full-filename siehe Datentyp filename 34

Funktionstasten 54  
 Old-Modus 186  
 Style-Guide-Modus 186

Funktionstastenbelegung 65, 180

**G**

garantierte Abkürzung 95

garantierte Meldung 48

geblockte Eingabe 55

geführter Dialog 57, 72  
 Spezialeingaben 74  
 temporär 76

gen (Zusatz zu Datentypen) 44

Gruppensyntaxdatei 159

**H**

HELP-MSG-INFORMATION  
 (Standardanweisung) 229

HELP-SDF (Kommando) 166

Hierarchie von Syntaxdateien 161

Hilfe-Funktion 54, 66

Historie 104, 180, 212

HOLD-PROGRAM (Standardanweisung) 229

**I**

Index  
 global 42  
 Konstruktionszeichenfolge 42  
 platzhalter-spezifisch 42  
 -Schreibweise 43

Informationen ausgeben  
 über aktuelle SDF-Einstellungen siehe SHOW-SDF-OPTIONS 216  
 über eingesetzte Produkte siehe SHOW-SYNTAX-VERSIONS 222  
 über SDF siehe HELP-SDF 166

integer (Datentyp) 35

ISP (Interactiv Scanner and Processor) 2  
 Dialogführung 76  
 Kommandosprache 2, 23

**J**

Jobvariablen ersetzen 88

**K**

Kommando  
 Anwendungsbereich 24  
 Anwendungsbereich SDF 165  
 Aufbau 23  
 Eingabebeispiel 117  
 Eingabehinweis 26  
 einschränken 2  
 HELP-SDF 166

- Kommando (Forts.)
  - ISP-Format [2](#)
  - Menü [71](#)
  - modifizieren [2](#), [163](#)
  - MODIFY-SDF-OPTIONS [180](#)
  - Name [23](#)
  - Protokollierung ändern [184](#)
  - rekonstruieren [8](#), [18](#)
  - REMARK [193](#)
  - RESET-INPUT-DEFAULTS [195](#)
  - RESTORE-SDF-INPUT [199](#)
  - selbst implementieren [80](#)
  - SHOW-CMD [204](#)
  - SHOW-INPUT-DEFAULTS [208](#)
  - SHOW-INPUT-HISTORY [212](#)
  - SHOW-RETURNCODE [215](#)
  - SHOW-SDF-OPTIONS [216](#)
  - SHOW-SYNTAX-VERSIONS [222](#)
  - sperrern [2](#)
  - Standardname [95](#)
  - Syntax prüfen [181](#)
  - Syntaxanalyse [106](#)
  - temporär geführter Dialog [76](#)
  - Testmodus [106](#)
  - umbenennen [95](#)
  - WRITE-TEXT [225](#)
- Kommandodatei kommentieren siehe [REMARK 193](#)
- Kommandoprozessor SDF [1](#)
- Kommando-Returncode
  - allgemeine Werte [49](#)
  - Bestandteile [47](#)
  - des letzten Kommandos ausgeben, siehe [SHOW-RETURNCODE 215](#)
- Kommandosprache
  - Aufbau [23](#)
  - ISP [2](#), [23](#)
  - SDF [23](#)
- Kommandosyntax ausgeben siehe [SHOW-CMD 204](#)
- Kommentar [27](#)
  - Zeilenende- [28](#)
- konstanter Operandenwert [23](#)
- Konstruktionsangabe [43](#)
- Konstruktionszeichenfolge [42](#)
- Konvertieren, Syntaxdatei [235](#)
- Korrektur der Eingabe [52](#)
- Korrekturdialog [1](#)
- Kurzname [32](#)
- L**
  - Länge der Eingabe [28](#)
  - Leerzeichen in der Eingabe [27](#)
  - Liefereinheitssyntaxdatei [164](#)
  - Liste
    - von Operandenwerten [24](#)
    - von S-Variablen [241](#)
  - Listenelement
    - Operandenwert [24](#)
    - S-Variable [249](#)
  - LOGOFF-Prozedur [82](#)
    - benutzerspezifisch [82](#)
    - Standardnamen [82](#)
    - systemweit [82](#)
  - LOGON-Prozedur [81](#)
    - benutzerspezifisch [81](#)
    - erstellen [14](#)
    - Standardnamen [81](#)
    - systemweit [81](#)
  - löschen
    - Eingabepuffer [189](#)
    - task-spezifische Default-Werte [195](#)
  - low (Zusatz zu Datentypen) [39](#)
- M**
  - Makroaufrufe [238](#)
  - man (Zusatz zu Datentypen) [44](#), [45](#)
  - mandatory (Zusatz zu Datentypen) [45](#)
  - Marke
    - im Nicht-S-Format [6](#), [27](#)
    - im S-Format [6](#), [27](#)
  - Menü [60](#)
    - anfordern [71](#)
    - Anwendungsbereich [70](#)
    - der Anweisungen [71](#)
    - der Kommandos [71](#)
    - Protokoll [100](#)
  - Metasyntax SDF [29](#), [31](#)

- MODIFY-SDF-OPTIONS  
   (Kommando) 180  
   (Standardanweisung) 230
- Musterzeichen  
   im Anweisungsnamen 102  
   im Kommandonamen 102
- N**
- name (Datentyp) 35  
 NEXT-Zeile 60, 63  
 Nicht-S-Marke 27  
 Nicht-S-Prozedur 5
- O**
- odd (Zusatz zu Datentypen) 44  
 Old-Modus (Funktionstastenbelegung) 54, 65  
 OMNIS 251  
 Operand  
   ausblenden im Protokoll 100  
   Default-Wert 25, 96  
   konstant 23  
   Name 23  
   Schlüsselwort- 27  
   Stellungs- 27  
   untergeordnet 24  
   variabel 23  
   Vorbereitung 25  
   wahlfrei 96  
 Operandenfragebogen 60, 72  
   anfordern 72  
   Spezialeingaben 74  
 Operandenwert  
   Angabe als Liste 24  
   explizit Angabe 96  
   implizite Angabe 96  
   vorbereitet 96
- P**
- Parameterdatei 158  
 partial-filename (Datentyp) 36  
 path-compl (Zusatz zu Datentypen) 39  
 posix-filename (Datentyp) 36  
 posix-pathname (Datentyp) 36  
 POSIX-Platzhalter 40
- Privilegien prüfen 186  
 product-version (Datentyp) 37  
 PROFILE-ID 159  
 Programm, Syntax prüfen 181  
 Programmunterbrechung 83  
 Protokoll 100  
   Menü 100  
   Umfang 100  
 Protokollierung  
   ändern siehe MODIFY-SDF-OPTIONS 184  
   einstellen 100  
 Prozedur  
   bei LOGOFF 82  
   bei LOGON 81  
   Eingabe aus 80  
   implementieren 80  
   Korrekturdialog 80  
   Syntaxfehlerdialog 185  
   unterbrechen 83  
 Prozedurdialog 83  
 Prozedurparameter ersetzen 87  
 Prozedurunterbrechung 83
- Q**
- quotes (Zusatz zu Datentypen) 45
- R**
- Refresh-Funktion 65, 66  
 REMARK  
   (Kommando) 193  
   (Standardanweisung) 230  
 RESET-INPUT-DEFAULTS 195  
   (Kommando) 195  
   (Standardanweisung) 231  
 Restore-Funktion 55, 67  
 RESTORE-SDF-INPUT  
   (Kommando) 199  
   Standardanweisung 231  
 Rückwärts Blättern 66  
 Rückzugsfunktionen 68, 78
- S**
- Schlüsselwort 23  
   abkürzen 94

- Schlüsselwortoperand 27, 97
- Schutz der Eingabe 105
- SDF
  - Aufbau der Kommandosprache 23
  - CHECKPOINT/RESTART 251
  - Dialogführung steuern 183
  - Eingabe wiederholen 199
  - Fehlerdialog (inhaltlicher Fehler) 199
  - Fehlerdialog (Syntaxfehler) 185
  - Information ausgeben über 166
  - Kommando 165
  - Kommandoprozessor 1
  - Kommandosprache 23
  - Korrekturdialog 1
  - Makroaufrufe 238
  - OMNIS 251
  - Optionen ändern 180
  - Optionen ausgeben 216
  - Parameterdatei 158
  - PROFILE-ID 159
  - Sprache der Dialogführung 2
  - Standardanweisungen 227
  - Syntaxbeschreibung 157
  - Syntaxdatei 157
  - Syntaxdatei aktivieren 180
  - Syntaxdatei deaktivieren 180
  - Syntaxdatei-Name ausgeben 216
  - verkürzte Eingabe 94
- SDF (System Dialog Facility) 1
- SDF-A (Dienstprogramm)
  - Anweisungsvorrat modifizieren 2
  - Kommandovorrat modifizieren 2
- SDF-Optionen ändern siehe MODIFY-SDF-OPTIONS 180
- sep (Zusatz zu Datentypen) 44
- SHOW-CMD (Kommando) 204
- SHOW-INPUT-DEFAULTS
  - (Kommando) 208
  - (Standardanweisung) 232
- SHOW-INPUT-HISTORY
  - (Kommando) 212
  - (Standardanweisung) 232
- SHOW-RETURNCODE (Kommando) 215
- SHOW-SDF-OPTIONS
  - (Kommando) 216
  - (Standardanweisung) 233
  - Ausgabe in S-Variablen 219
  - Ausgabeformat 217
- SHOW-STMT (Standardanweisung) 233
- SHOW-SYNTAX-FILE (Anweisung) 236
- SHOW-SYNTAX-VERSIONS
  - (Kommando) 222
  - Ausgabe in S-Variablen 223
- S-Marke 27
- Speichern von Eingaben 104
- Spezialeingabe
  - im ungeführten Dialog 53, 61
  - Operandenfragebogen 74
- Sprache einstellen 46
- Sprachschlüssel 46
- S-Prozedur 5
- Standardanweisung 227
  - END 227
  - EXECUTE-SYSTEM-CMD 228
  - HELP-MSG-INFORMATION 229
  - HOLD-PROGRAM 229
  - MODIFY-SDF-OPTIONS 230
  - REMARK 230
  - RESET-INPUT-DEFAULTS 231
  - RESTORE-SDF-INPUT 231
  - SHOW-INPUT-DEFAULTS 232
  - SHOW-INPUT-HISTORY 232
  - SHOW-SDF-OPTIONS 233
  - SHOW-STMT 233
  - STEP 234
  - WRITE-TEXT 234
- Standardname
  - Benutzersyntaxdatei 160
  - LOGOFF-Prozedur 82
  - LOGON-Prozedur 81
- Standardnamen 95
- Standardwert siehe [Default-Wert](#)
- Stapelbetrieb, Eingabe 86
- Stellungsoperand 27, 97
- STEP (Standardanweisung) 234
- Steueranweisungen für Dialogführung 60, 63
- structured-name (Datentyp) 37

- Struktur
    - Attribut NULL-ABBREVIATION 99
    - dynamisch 242
    - einleiten 24
    - flache Schreibweise 98
    - Schachtelung 24
    - statisch 242, 250
    - STRUCTURE-IMPLICIT-Schreibweise 98
    - S-Variablen-Typ 241
    - verkürzte Eingabe 98
  - Stufen der Dialogführung 57
  - Style-Guide-Modus
    - (Funktionstastenbelegung) 54, 65
    - einstellen siehe MODIFY-SDF-  
OPTIONS 186
  - S-Variable 5
    - strukturierte Ausgabe 241
    - Typ Struktur 241
  - S-Variablen-Ausdruck ersetzen 89
  - S-Variablen-Ausgabe
    - SHOW-SDF-OPTIONS 219
    - SHOW-SYNTAX-VERSIONS 223
  - S-Variablen-Liste 241
  - Syntaxanalyse 106
  - Syntaxdatei 157
    - aktivieren 180
    - Arten 157
    - Benutzer- 160
    - deaktivieren 180
    - entmischen 164
    - erstellen 163
    - Format ausgeben 236
    - Format konvertieren 235
    - für Benutzergruppe 159
    - Gruppen- 159
    - Hierarchie 161
    - Inhalt 157
    - Liefereinheit 164
    - mischen 164
    - modifizieren 163
    - Name ausgeben 216
    - System- 158
  - Syntaxfehlerdialog 180
  - Syntaxprüfung
    - von Anweisungen 180
    - von Kommandos 180
  - Syntaxversionen anzeigen 222
  - SYSINF (Systemausgabestrom) 242
  - SYSLST (Systemdatei), Text ausgeben 225
  - SYSOUT (Systemdatei), Text ausgeben 225
  - Systemausgabe, strukturiert 242
  - Systemausgabestrom SYSINF 242
  - Systemsyntaxdatei 158
- ## T
- Task-lokale SDF-Optionen 12
  - Task-spezifischer Default-Wert 110
  - temp-file (Zusatz zu Datentypen) 44
  - temporär geführter Dialog 76
  - Test-Funktion 66
  - Testmodus
    - einschalten 106
    - für Anweisungen 106
    - für Kommandos 106
    - Kommando 181
    - Programmanweisung 181
  - text (Datentyp) 37
  - Text ausgeben
    - nach SYSLST 225
    - nach SYSOUT 193, 225
  - time (Datentyp) 37
- ## U
- under (Zusatz zu Datentypen) 39
  - ungeführter Dialog 51
    - Korrektur 52
    - Spezialeingaben 53, 61
  - Unterbrechung
    - K2-Funktionstaste 83
    - Programm 83
    - Prozedur 83
    - zulassen 83, 84
    - zurückweisen 85
  - Unterbrechungs-Ereignis 83
  - Unterbrechungs-Funktion 54, 65, 66
  - user (Zusatz zu Datentypen) 45

### V

- variabler Operandenwert [23](#)
- verdeckte Eingabe [53](#)
- vers (Zusatz zu Datentypen) [45](#)
- vordefinierte Funktion siehe [Builtin-Funktion](#)
- Vorwärts Blättern [67](#)
- vsn (Datentyp) [37](#)

### W

- wahlfreier Operand [96](#)
- Wiederherstellung der Eingabe [1](#)
- wild(n) (Zusatz zu Datentypen) [40](#)
- with (Zusatz zu Datentypen) [39](#)
- without (Zusatz zu Datentypen) [44](#)
- WRITE-TEXT
  - (Kommando) [225](#)
  - (Standardanweisung) [234](#)

### X

- XHCS-Unterstützung [26](#)
- x-string (Datentyp) [38](#)
- x-text (Datentyp) [38](#)

### Z

- Zeichensatz erweitert [26](#)
- Zeilenende-Kommentar [28](#)
- Zugriff auf Listenelemente [249](#)
- Zuordnung
  - Benutzersyntaxdatei [160](#)
  - Gruppensyntaxdatei [159](#)
  - Systemsyntaxdatei [158](#)
- Zusätze zu Datentypen [30, 39](#)



---

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Kurzbeschreibung des Produkts	1
1.2	Zielgruppe des Handbuchs	3
1.3	Konzept des Handbuchs	3
1.4	Änderungen gegenüber der vorherigen Ausgabe	4
<b>2</b>	<b>Schnelleinstieg in die SDF-Anwendung</b>	<b>7</b>
<b>3</b>	<b>Die Kommandosprache SDF</b>	<b>23</b>
3.1	Aufbau der Kommandosprache SDF	23
	Konventionen für die Eingabe	26
3.2	SDF-Syntaxdarstellung	29
3.3	Sprache für Ausgaben von SDF	46
3.4	Kommando-Returncode	47
<b>4</b>	<b>Kommando- und Anweisungseingabe</b>	<b>51</b>
4.1	Eingabe im ungeführten Dialog	51
4.1.1	EXPERT-Form	52
4.1.2	NO-Form	52
4.1.3	Spezielle Eingaben	53
4.1.4	Funktionstasten	54
4.1.5	Geblockte Eingaben	55
4.2	Eingabe im geführten Dialog	57
4.2.1	Stufen der Dialogführung	57
	Maximale Benutzerführung	57
	Mittlere Benutzerführung	58
	Minimale Benutzerführung	58
4.2.2	Aufbau der Bildschirmmasken	59
4.2.3	Spezielle Eingaben	61
4.2.4	Eingaben in die NEXT-Zeile	63
4.2.5	Funktionstasten	65
4.2.6	Wirkung der Rückzugsfunktionen	68
4.2.7	Anwendungsbereichs-Menü	70
4.2.8	Kommando- und Anweisungs-Menü	71
4.2.9	Operandenfragebogen	72
4.2.10	Eingaben in den Operandenfragebogen	74

4.2.11	Positionieren im Operandenfragebogen .....	75
4.3	Eingabe im temporär geführten Dialog .....	76
4.3.1	Wirkung der Rückzugsfunktionen .....	78
4.4	Weitere Eingabemöglichkeiten .....	80
4.4.1	Eingabe aus Prozeduren .....	80
4.4.2	Eingabe im Stapelbetrieb .....	86
4.5	Ausdrücke in der Eingabe ersetzen .....	87
4.6	Verkürzte Eingabe .....	94
4.6.1	Namen abkürzen .....	94
4.6.2	Default-Wert übernehmen .....	96
4.6.3	Stellungsoperanden angeben .....	97
4.6.4	Strukturen abkürzen .....	98
4.7	Protokollierung der Eingabe .....	100
4.7.1	LOGGING-Einstellung .....	100
4.7.2	MENU-LOGGING-Einstellung .....	101
4.7.3	Einschränkungen .....	101
4.8	Auswahlmenü für Kommandos und Anweisungen .....	102
4.8.1	Musterzeichen in Kommando- oder Anweisungsnamen .....	102
4.8.2	Mehrdeutige Abkürzung von Kommando- oder Anweisungsnamen .....	103
4.9	Vorangegangene Eingaben wieder verwenden .....	104
4.10	Testmodus .....	106
4.11	Task-spezifische Default-Werte .....	110
<b>5</b>	<b>Beispiele für die Kommandoeingabe im Dialogbetrieb .....</b>	<b>117</b>
5.1	HELP-SDF .....	118
5.2	Ungeführter Dialog .....	122
5.2.1	EXPERT-Form .....	122
5.2.2	NO-Form .....	124
5.3	Temporär geführter Dialog .....	126
5.4	Geführter Dialog .....	138
5.4.1	Minimale Benutzerführung .....	138
5.4.2	Mittlere Benutzerführung .....	143
5.4.3	Maximale Benutzerführung .....	147
<b>6</b>	<b>SDF-Syntaxdateien .....</b>	<b>157</b>
6.1	Systemsyntaxdatei .....	158
6.2	Gruppensyntaxdatei .....	159
6.3	Benutzersyntaxdatei .....	160
6.4	Privilegien in Syntaxdateien .....	161
6.5	Dateihierarchie .....	161
6.6	Bearbeiten von Syntaxdateien .....	163

<b>7</b>	<b>Benutzerschnittstelle SDF</b>	<b>165</b>
7.1	Kommandos aus dem Anwendungsbereich SDF	165
	HELP-SDF	Informationen über SDF ausgeben 166
	MODIFY-SDF-OPTIONS	Benutzersyntaxdatei aktivieren und SDF-Optionen ändern 180
	REMARK	Bemerkungen in Kommandodatei einfügen 193
	RESET-INPUT-DEFAULTS	Task-spezifische Default-Werte löschen 195
	RESTORE-SDF-INPUT	Bereits erfolgte Eingabe wiederherstellen 199
	SHOW-CMD	Syntaxbeschreibung eines Kommandos ausgeben 204
	SHOW-INPUT-DEFAULTS	Task-spezifische Default-Werte ausgeben 208
	SHOW-INPUT-HISTORY	Gespeicherte Eingaben nach SYSOUT ausgeben 212
	SHOW-RETURNCODE	Kommando-Returncode des letzten Kommandos ausgeben 215
	SHOW-SDF-OPTIONS	Aktive Syntaxdateien und SDF-Optionen anzeigen 216
	SHOW-SYNTAX-VERSIONS	Versionsstände von Syntaxdateien anzeigen 222
	WRITE-TEXT	Text auf SYSOUT bzw. SYSLST ausgeben 225
7.2	Standardanweisungen	227
	END	227
	EXECUTE-SYSTEM-CMD	228
	HELP-MSG-INFORMATION	229
	HOLD-PROGRAM	229
	MODIFY-SDF-OPTIONS	230
	REMARK	230
	RESET-INPUT-DEFAULTS	231
	RESTORE-SDF-INPUT	231
	SHOW-INPUT-DEFAULTS	232
	SHOW-INPUT-HISTORY	232
	SHOW-SDF-OPTIONS	233
	SHOW-STMT	233
	STEP	234
	WRITE-TEXT	234
7.3	SDF-I-Anweisungen	235
	CONVERT-SYNTAX-FILE	235
	SHOW-SYNTAX-FILE	236
7.4	Makroaufrufe für SDF	238
7.5	SDF-Schnittstelle zu höheren Programmiersprachen	239
<b>8</b>	<b>Strukturierte Ausgabe in S-Variablen</b>	<b>241</b>
<b>9</b>	<b>Hinweise für OMNIS und CHECKPOINT/RESTART</b>	<b>251</b>

<b>Abbildungen</b> .....	<b>253</b>
<b>Tabellen</b> .....	<b>257</b>
<b>Literatur</b> .....	<b>259</b>
<b>Stichwörter</b> .....	<b>265</b>

---

# SDF V4.5A (BS2000/OSD)

## Einführung in die Dialogschnittstelle SDF

*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.

**Ausgabe: März 2002**

**Datei: sdf\_einf.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  
Einführung in die Dialogschnittstelle SDF



## 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