

# SDF V4.7

Dialogschnittstelle SDF

Stand der Beschreibung:

SDF-I V4.1

SDF-U V4.1

SDF-PAR V1.1

SDF-CONV V3.0B

DISPLAY V1.1

BS2000/OSD V8.0

JV V15.0

SDF-P V2.5A

## **Kritik... Anregungen... Korrekturen...**

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

Sie können uns Ihre Kommentare per E-Mail an [manuals@ts.fujitsu.com](mailto:manuals@ts.fujitsu.com) senden.

## **Zertifizierte Dokumentation nach DIN EN ISO 9001:2008**

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

cognitas. Gesellschaft für Technik-Dokumentation mbH  
[www.cognitas.de](http://www.cognitas.de)

## **Copyright und Handelsmarken**

Copyright © Fujitsu Technology Solutions GmbH 2010.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

---

# Inhalt

<b>1</b>	<b>Einleitung</b> . . . . .	<b>11</b>
<b>1.1</b>	<b>Kurzbeschreibung des Produkts</b> . . . . .	<b>11</b>
<b>1.2</b>	<b>Zielgruppe des Handbuchs</b> . . . . .	<b>13</b>
<b>1.3</b>	<b>Konzept des Handbuchs</b> . . . . .	<b>13</b>
<b>1.4</b>	<b>Änderungen gegenüber der vorherigen Ausgabe</b> . . . . .	<b>14</b>
<b>1.5</b>	<b>Readme-Datei</b> . . . . .	<b>15</b>
<b>2</b>	<b>Schnelleinstieg in die SDF-Anwendung</b> . . . . .	<b>17</b>
<b>3</b>	<b>Kommando- und Anweisungseingabe</b> . . . . .	<b>33</b>
<b>3.1</b>	<b>Eingabe im ungeführten Dialog</b> . . . . .	<b>33</b>
3.1.1	EXPERT-Form . . . . .	34
3.1.2	NO-Form . . . . .	34
3.1.3	Spezielle Eingaben . . . . .	35
3.1.4	Funktionstasten . . . . .	36
3.1.5	Geblockte Eingaben . . . . .	37
<b>3.2</b>	<b>Eingabe im geführten Dialog</b> . . . . .	<b>39</b>
3.2.1	Stufen der Dialogführung . . . . .	39
	Maximale Benutzerführung . . . . .	39
	Mittlere Benutzerführung . . . . .	40
	Minimale Benutzerführung . . . . .	40
3.2.2	Aufbau der Bildschirmmasken . . . . .	41
3.2.3	Spezielle Eingaben . . . . .	43
3.2.4	Eingaben in die NEXT-Zeile . . . . .	45
3.2.5	Funktionstasten . . . . .	47
3.2.6	Wirkung der Rückzugsfunktionen . . . . .	50
3.2.7	Anwendungsbereichs-Menü . . . . .	52

3.2.8	Kommando- und Anweisungs-Menü . . . . .	53
3.2.9	Operandenfragebogen . . . . .	54
3.2.10	Eingaben in den Operandenfragebogen . . . . .	56
3.2.11	Positionieren im Operandenfragebogen . . . . .	57
<b>3.3</b>	<b>Eingabe im temporär geführten Dialog . . . . .</b>	<b>58</b>
3.3.1	Wirkung der Rückzugsfunktionen . . . . .	60
<b>3.4</b>	<b>Weitere Eingabemöglichkeiten . . . . .</b>	<b>62</b>
3.4.1	Eingabe aus Prozeduren . . . . .	62
3.4.2	Eingabe im Stapelbetrieb . . . . .	69
<b>3.5</b>	<b>Ausdrücke in der Eingabe ersetzen . . . . .</b>	<b>70</b>
<b>3.6</b>	<b>Verkürzte Eingabe . . . . .</b>	<b>77</b>
3.6.1	Namen abkürzen . . . . .	77
3.6.2	Default-Wert übernehmen . . . . .	79
3.6.3	Stellungsoperanden angeben . . . . .	80
3.6.4	Strukturen abkürzen . . . . .	81
<b>3.7</b>	<b>Protokollierung der Eingabe . . . . .</b>	<b>83</b>
3.7.1	LOGGING-Einstellung . . . . .	83
3.7.2	MENU-LOGGING-Einstellung . . . . .	84
3.7.3	Einschränkungen . . . . .	84
<b>3.8</b>	<b>Auswahlmenü für Kommandos und Anweisungen . . . . .</b>	<b>85</b>
3.8.1	Musterzeichen in Kommando- oder Anweisungsnamen . . . . .	85
3.8.2	Mehrdeutige Abkürzung von Kommando- oder Anweisungsnamen . . . . .	86
<b>3.9</b>	<b>Vorangegangene Eingaben wieder verwenden . . . . .</b>	<b>87</b>
<b>3.10</b>	<b>Testmodus . . . . .</b>	<b>89</b>
<b>3.11</b>	<b>Task-spezifische Default-Werte . . . . .</b>	<b>93</b>
<b>4</b>	<b>Beispiele für die Kommandoeingabe im Dialogbetrieb . . . . .</b>	<b>99</b>
<b>4.1</b>	<b>HELP-SDF . . . . .</b>	<b>100</b>
<b>4.2</b>	<b>Ungeführter Dialog . . . . .</b>	<b>104</b>
4.2.1	EXPERT-Form . . . . .	104
4.2.2	NO-Form . . . . .	106
<b>4.3</b>	<b>Temporär geführter Dialog . . . . .</b>	<b>108</b>

<b>4.4</b>	<b>Geführter Dialog</b> . . . . .	<b>120</b>
4.4.1	Minimale Benutzerführung . . . . .	120
4.4.2	Mittlere Benutzerführung . . . . .	125
4.4.3	Maximale Benutzerführung . . . . .	129
<b>5</b>	<b>SDF-Syntaxdateien</b> . . . . .	<b>139</b>
<hr/>		
<b>5.1</b>	<b>Systemsyntaxdatei</b> . . . . .	<b>140</b>
<b>5.2</b>	<b>Gruppensyntaxdatei</b> . . . . .	<b>141</b>
<b>5.3</b>	<b>Benutzersyntaxdatei</b> . . . . .	<b>142</b>
<b>5.4</b>	<b>Privilegien in Syntaxdateien</b> . . . . .	<b>143</b>
<b>5.5</b>	<b>Dateihierarchie</b> . . . . .	<b>143</b>
<b>5.6</b>	<b>Bearbeiten von Syntaxdateien</b> . . . . .	<b>145</b>
<b>6</b>	<b>Benutzerschnittstelle SDF</b> . . . . .	<b>147</b>
<hr/>		
<b>6.1</b>	<b>Kommandos aus dem Anwendungsbereich SDF</b> . . . . .	<b>147</b>
<b>6.2</b>	<b>Standardanweisungen</b> . . . . .	<b>148</b>
	END . . . . .	148
	EXECUTE-SYSTEM-CMD . . . . .	149
	HELP-MSG-INFORMATION . . . . .	150
	HOLD-PROGRAM . . . . .	150
	MODIFY-SDF-OPTIONS . . . . .	151
	REMARK . . . . .	151
	RESET-INPUT-DEFAULTS . . . . .	152
	RESTORE-SDF-INPUT . . . . .	152
	SHOW-INPUT-DEFAULTS . . . . .	153
	SHOW-INPUT-HISTORY . . . . .	153
	SHOW-SDF-OPTIONS . . . . .	154
	SHOW-STMT . . . . .	154
	STEP . . . . .	155
	WRITE-TEXT . . . . .	155
<b>6.3</b>	<b>Makroaufrufe für SDF</b> . . . . .	<b>156</b>
<b>6.4</b>	<b>SDF-Schnittstelle zu höheren Programmiersprachen</b> . . . . .	<b>157</b>

<b>7</b>	<b>Hinweise für OMNIS und CHECKPOINT/RESTART</b>	<b>159</b>
<b>8</b>	<b>Installation von SDF</b>	<b>161</b>
8.1	SDF-Installationsdateien	161
8.2	Subsystemkatalog generieren	163
8.3	SDF bereitstellen	163
8.3.1	Dateien für SDF bereitstellen	163
8.3.2	SDF starten	163
<b>9</b>	<b>Verhalten im Fehlerfall</b>	<b>165</b>
<b>10</b>	<b>SDF-I</b>	<b>167</b>
10.1	Ein-/Ausgaben von SDF-I	167
10.2	Aufruf und Ablauf von SDF-I	168
10.3	Behandlung von Arbeitsdateien	170
10.4	SDF-Standardanweisungen und Anwendungsbereiche	171
10.5	Hinweise	171
10.6	SDF-I-Anweisungen	173
10.6.1	Funktionelle Übersicht	173
10.6.2	Beschreibung der Anweisungen	174
	CONVERT-SYNTAX-FILE	
	Syntaxdatei mit neuem Format versehen	174
	END	
	SDF-I beenden	175
	MERGE	
	Syntaxdateien mischen	176
	OPEN	
	Eingabe- und Ausgabesyntaxdatei zuweisen	180
	REMOVE	
	Syntaxdateien entmischen	183
	SHOW-SYNTAX-FILE	
	Informationen über eine Syntaxdatei anzeigen	185

---

<b>11</b>	<b>SDF-U</b> . . . . .	<b>189</b>
<b>11.1</b>	<b>SDF-U-Anweisungen</b> . . . . .	<b>191</b>
11.1.1	Funktionelle Übersicht . . . . .	191
11.1.2	Beschreibung der Anweisungen . . . . .	193
	COPY	
	Inhalte einer Syntaxdatei kopieren . . . . .	193
	EDIT	
	Auf ein Kommando der Syntaxdatei positionieren . . . . .	196
	END	
	Programmablauf beenden . . . . .	198
	MODIFY-CMD	
	Kommandodefinition ändern . . . . .	199
	MODIFY-VALUE	
	Operandenwertdefinition ändern . . . . .	200
	OPEN-SYNTAX-FILE	
	Syntaxdatei öffnen . . . . .	201
	REMOVE	
	Objekte der Syntaxdatei löschen . . . . .	202
	SET-GLOBALS	
	Globalinformation ändern . . . . .	204
	SHOW	
	Objekte der Syntaxdatei anzeigen . . . . .	210
	SHOW-CORRECTION-INFORMATION	
	Korrekturinformation der Syntaxdatei ausgeben . . . . .	215
	SHOW-STATUS	
	Status der geöffneten Syntaxdatei anzeigen . . . . .	216
	STEP	
	Definieren eines Wiederaufsetzpunktes . . . . .	217
<b>12</b>	<b>SDF-PAR</b> . . . . .	<b>219</b>
<b>12.1</b>	<b>SDF-PAR-Anweisungen</b> . . . . .	<b>220</b>
12.1.1	Funktionelle Übersicht . . . . .	220
12.1.2	Beschreibung der Anweisungen . . . . .	222
	MODIFY-SYNTAX-FILE	
	Einträge für Syntaxdateien ändern . . . . .	222
	MODIFY-SYSTEM-LOGON-PROCEDURE	
	Eintrag für systemweite LOGON-Call-Prozedur ändern . . . . .	224
	MODIFY-SYSTEM-LOGON-INCLUDE	
	Eintrag für systemweite LOGON-Include-Prozedur ändern . . . . .	225

---

MODIFY-VERSION	
Format der Parameterdatei ändern . . . . .	226
OPEN-PARAMETER-FILE	
Parameterdatei erzeugen oder öffnen . . . . .	227
SHOW-PARAMETER-FILE	
Parameterdatei anzeigen . . . . .	230
<b>13</b>	<b>SDF-CONV . . . . . 231</b>
<b>13.1</b>	<b>Starten und Beenden des Programms . . . . . 232</b>
<b>13.2</b>	<b>Anweisungen . . . . . 233</b>
13.2.1	Übersicht über alle Anweisungen . . . . . 233
13.2.2	CONVERT - Anweisung . . . . . 234
13.2.3	Steuerfunktionen . . . . . 247
<b>13.3</b>	<b>Ausgaben . . . . . 248</b>
13.3.1	Ausgabeprozedur . . . . . 248
13.3.2	Ausgaben nach SYSOUT . . . . . 250
13.3.3	Übersetzungsprotokoll nach SYSLST . . . . . 250
13.3.4	Programmüberwachung durch Monitor-Jobvariable . . . . . 251
<b>13.4</b>	<b>Einschränkungen . . . . . 252</b>
13.4.1	Allgemeine Voraussetzungen und Einschränkungen . . . . . 252
13.4.2	Einschränkungen bei der Umwandlung zu S-Prozeduren . . . . . 254
13.4.3	Nicht übersetzbare Kommandos . . . . . 256
13.4.4	Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen . . . . . 258
13.4.5	Einschränkungen bei Prozedurparametern als Operanden . . . . . 262
<b>13.5</b>	<b>Bestandteile und Installation von SDF-CONV . . . . . 265</b>
<b>13.6</b>	<b>Beispiele . . . . . 267</b>
	Beispiel 1: . . . . . 268
	Beispiel 2 . . . . . 274
	Beispiel 3 . . . . . 282
<b>13.7</b>	<b>Zuordnung von ISP- zu SDF-Kommandos . . . . . 286</b>
13.7.1	Benutzerkommandos . . . . . 286
13.7.2	Kommandos für die Systemverwaltung . . . . . 291
<b>13.8</b>	<b>Übersetzung von ISP- und alten SDF-Kommandos durch SDF-CONV . . . . . 295</b>
13.8.1	Benutzerkommandos . . . . . 295
13.8.2	Kommandos für die Systemverwaltung . . . . . 300
<b>13.9</b>	<b>Meldungen von SDF-CONV . . . . . 302</b>



---

<b>14</b>	<b>DISPLAY</b> . . . . .	<b>305</b>
<b>14.1</b>	<b>Produkteinsatz</b> . . . . .	<b>305</b>
14.1.1	Einschränkungen . . . . .	307
<b>14.2</b>	<b>Kommandos</b> . . . . .	<b>308</b>
14.2.1	Übersicht über alle Kommandos . . . . .	308
14.2.2	CREATE-DISPLAY-CMD	
	DISPLAY-Kommando erzeugen . . . . .	309
	DELETE-DISPLAY-CMD	
	DISPLAY-Kommando löschen . . . . .	311
<b>14.3</b>	<b>Installation</b> . . . . .	<b>312</b>
	<b>Literatur</b> . . . . .	<b>313</b>
	<b>Stichwörter</b> . . . . .	<b>315</b>

---



---

# 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 [Kapitel „SDF-CONV“ auf Seite 231](#)).

## 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.

Die Kapitel 1 bis 7 und Teile der Kapitel 10 und 13 sind für alle Benutzer relevant. Die übrigen Kapitel richten sich an die Systemverwaltung.

Weitergehende SDF-Leistungen für Zwecke der Systembetreuung sowie selbstdefinierte Kommandos oder die Programm-Schnittstelle sind 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 gibt eine Zusammenstellung der Regeln und Hinweise, die bei der Kommando- und Anweisungseingabe zu beachten sind.
- Kapitel 4 zeigt anhand einer Beispielsitzung, welche Eingabemöglichkeiten der Benutzer im geführten und ungeführten Dialog besitzt.
- Kapitel 5 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 6 enthält die SDF-spezifischen Kommandos, Standardanweisungen und eine Übersicht der Programmschnittstellen, die dem Benutzer zur Verfügung stehen.
- Kapitel 7 gibt Hinweise für OMNIS und CHECKPOINT/RESTART bei Einsatz von SDF.
- Kapitel 8 zeigt, wie SDF installiert wird.
- Kapitel 9 gibt Hinweise zum Verhalten in Fehlerfällen.
- Kapitel 10 beschreibt das Dienstprogramm SDF-I zum Mischen von Syntaxdateien.
- Kapitel 11 beschreibt das Dienstprogramm SDF-U zur Modifikation von Syntaxdateien.
- Kapitel 12 beschreibt das Dienstprogramm SDF-PAR für die Verwaltung von SDF-Parameterdateien.

- Kapitel 13 beschreibt das Dienstprogramm SDF-CONV zum Umwandeln von Kommando-Prozeduren in das aktuelle Prozedur- und Kommandoformat.
- Kapitel 14 beschreibt das Sybsystem DISPLAY. Dieses ermöglicht es, nicht privilegierten Benutzern den vollen Ausgabeumfang der meisten SHOW-Kommandos zur Verfügung zu stellen.

Die BS2000-Kommandos einschließlich der privilegierten Kommandos sind in den Handbüchern „Kommandos“ [1] beschrieben.

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

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 V4.7 kann ab BS2000/OSD-BC V6.0 eingesetzt werden. Die Beschreibung des vorliegenden Handbuchs bezieht sich auf den Einsatz von SDF V4.7 in einem System mit BS2000/OSD-BC V8.0. Vor allem wurden auch die Ablaufbeispiele und Beispielsitzungen in einem System mit BS2000/OSD-BC V8.0 erstellt.

### *Inhalt*

In diesem Handbuch sind die bisherigen Handbücher „Einführung in die Dialogschnittstelle SDF“, „SDF-Verwaltung“ und „SDF-CONV“ zusammengefasst.

Zusätzlich wird das Subsystem DISPLAY beschrieben. Dieses ermöglicht es, nicht privilegierten Benutzern den vollen Ausgabeumfang der meisten SHOW-Kommandos zur Verfügung zu stellen.

### *Kommandosprache*

Der Aufbau der Kommandosprache SDF zusammen mit der SDF-Syntaxdarstellung und die Kommando-Returncodes sind ausführlich im Handbuch „Kommandos“ [1] beschrieben und daher in diesem Handbuch nicht mehr enthalten.

## 1.5 Readme-Datei

Ergänzungen gegenüber den Handbüchern sind gegebenenfalls in den Readme-Dateien zu den jeweiligen Produktversionen aufgeführt. Solche Readme-Dateien finden Sie unter <http://manuals.ts.fujitsu.com> bei dem jeweiligen Produkt.

### *Readme-Datei unter BS2000/OSD*

Auf Ihrem BS2000-System finden Sie Readme-Dateien für die installierten Produkte unter dem Dateinamen:

```
SYSRME.<produkt>.<version>.D
```

Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Den vollständigen Pfadnamen erhalten Sie auch mit folgendem Kommando:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<produkt>, LOGICAL-ID=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 (z.B. für SDF V4.7):

```
/PRINT-DOCUMENT FROM-FILE=SYSRME.SDF.047.D, LINE-SPACING=*BY-EBCDIC-CONTROL
```

### *Ergänzende Produkt-Informationen*

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





---

## 2 Schnelleinstieg in die SDF-Anwendung

Die Kommandosprache SDF (System Dialog Facility) ersetzt ab BS2000/OSD V1.0 die Kommandosprache ISP. Kommandos der alten Kommandosprache ISP werden weiter kompatibel unterstützt. Die Nutzung der vollen SDF-Funktionalität ist jedoch nur für Kommandos und Anweisungen mit SDF-Syntax möglich (ISP-Kommandos werden nur auf ihren Namen hin geprüft).

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 33](#)) vornehmen. Sie können aber auch in einen menügesteuerten Eingabemodus (geführter Dialog, [Seite 39](#)) 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 87](#)).

### *Auskunftssystem 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 Auswahlménü 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 58](#)) genannt.

## Anwendungsbereiche ausgeben

Mit der Eingabe eines Fragezeichens und **[DUE]** liefert Ihnen SDF eine Auswahl von Anwendungsbereichen (siehe Anwendungsbereichs-Menü, [Seite 52](#)). Für das nächste Beispiel wählen Sie den Anwendungsbereich USER-ADMINISTRATION. Dazu geben Sie in das Eingabefeld NEXT (siehe NEXT-Zeile, [Seite 45](#)) 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 Auswahlménü mit allen Kommandos dieses Anwendungsbereichs (siehe Kommando-Ménü, [Seite 53](#)). 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 Auswahlnenü 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 63](#)). 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 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 85](#)). 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-FOR1-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  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:$SDFUSER1.ABK.ABKLIST
%      9 :20SG:$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 CODE '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 dd-mm-yy FOR TSN '1PYT'
% EXC0421 CPU TIME USED: 0.8174
% JMS0150 INSTALLATION ' S150-40', BS2000 VERSION 'V170', 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 Auswahlménü 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 dd-mm-yy AT 17:14, TSN = 1PY8
*** MINDESTENS 20 TAGE KEIN DATEIZUGRIFF AUF:
%      6 :20SG:$SDFUSER1.ABK.ABKLIST
%      9 :20SG:$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=sdfest
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 99](#) 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 33 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 139](#) 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 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 70](#)).

Die meisten der in diesem Kapitel gegebenen Informationen kann sich der Benutzer mit dem Kommando HELP-SDF ausgeben lassen (siehe Handbuch „Kommandos“ [1]).

### 3.1 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 39](#) und [58](#)).

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.

### 3.1.1 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 58](#)).

### 3.1.2 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 58](#)). Mit dem Kommando `RESTORE-SDF-INPUT` kann die letzte Eingabe angezeigt werden (siehe Handbuch „Kommandos“ [1]).

### 3.1.3 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 58](#)).
- ?? 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 37](#)).

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 58](#)).

### 3.1.4 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 1: 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 2: 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 2: 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.

### 3.1.5 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*q
(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 'nn' FROM LIBRARY ':2OSH:$TSOS.SYSLNK.SORT.
nnn' IN PROCESS
% BLS0524 LLM 'SRT80', VERSION 'nnn' OF 'yyyy-mm-dd 15:20:53' LOADED
% BLS0551 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.ALL RIGHTS RESERVED
% SRT1001 18:49:13/000000.00 SORT/MERGE STARTED, VERSION nn.nA00/BS2000Vnn.n
% SRT1130 PLEASE ENTER SORT STATEMENTS
(BL) sort-record
(BL) end
% SRT1016 SORT/MERGE INPUT RECORDS:.....688 (FROM 01)
% SRT1016 SORT/MERGE INPUT RECORDS:.....2.003 (FROM 02)
% SRT1017 RECORDS TO BE SORTED/MERGED:.....2.691
% SRT1030 SORT/MERGE OUTPUT RECORDS:.....2.691
% SRT1002 18:49:14/000000.19 SORT/MERGE COMPLETED
(BL) show-file-attr out.sort1-2
% 66 :2OSG:$USER1.OUT.SORT1-2
%:2OSG: PUBLIC: 1 FILE RES= 66 FRE= 2 REL= 0 PAGES
(BL) show-file-attr datei.
%PLEASE ACKNOWLEDGE
% 21 :2OSG:$USER1.DATEI.1
% 48 :2OSG:$USER1.DATEI.2
%:2OSG: PUBLIC: 2 FILES RES= 69 FRE= 4 REL= 0 PAGES
(BL) show-file-link link=sort*
%-- LINK-NAME ----- FILE-NAME -----
% SORTOUT :2OSG:$USER1.OUT.SORT1-2
% SORT01 :2OSG:$USER1.DATEI.1
% SORT02 :2OSG:$USER1.DATEI.2
/

```

## 3.2 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.

### 3.2.1 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.



### 3.2.2 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.

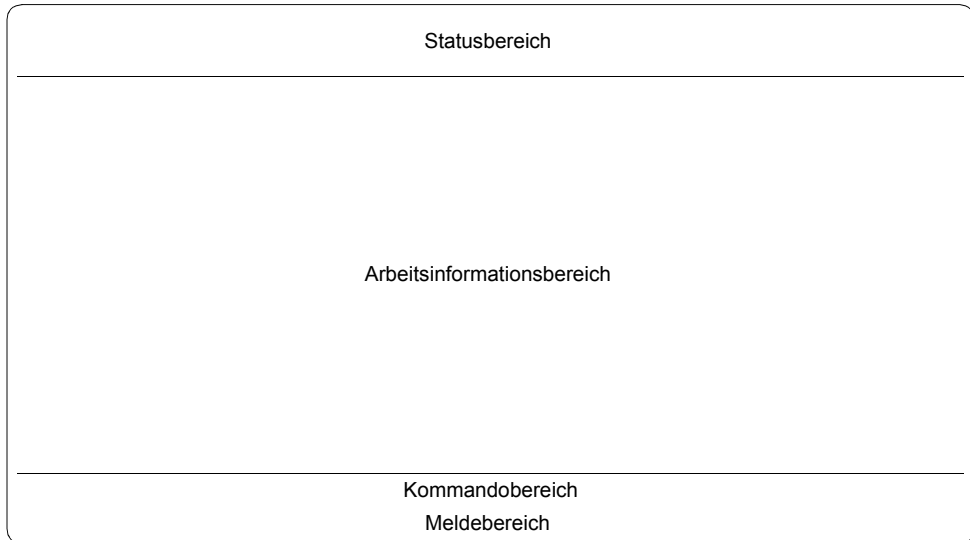


Bild 1: Bildschirmaufbau

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 (FUNCTION-KEYS=\*OLD-MODUS) werden alle zulässigen Steueranweisungen angezeigt. Funktionstasten, die einer zulässigen Steueranweisung entsprechen werden ebenfalls angezeigt.
  - Im Style-Guide-Modus (FUNCTION-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.

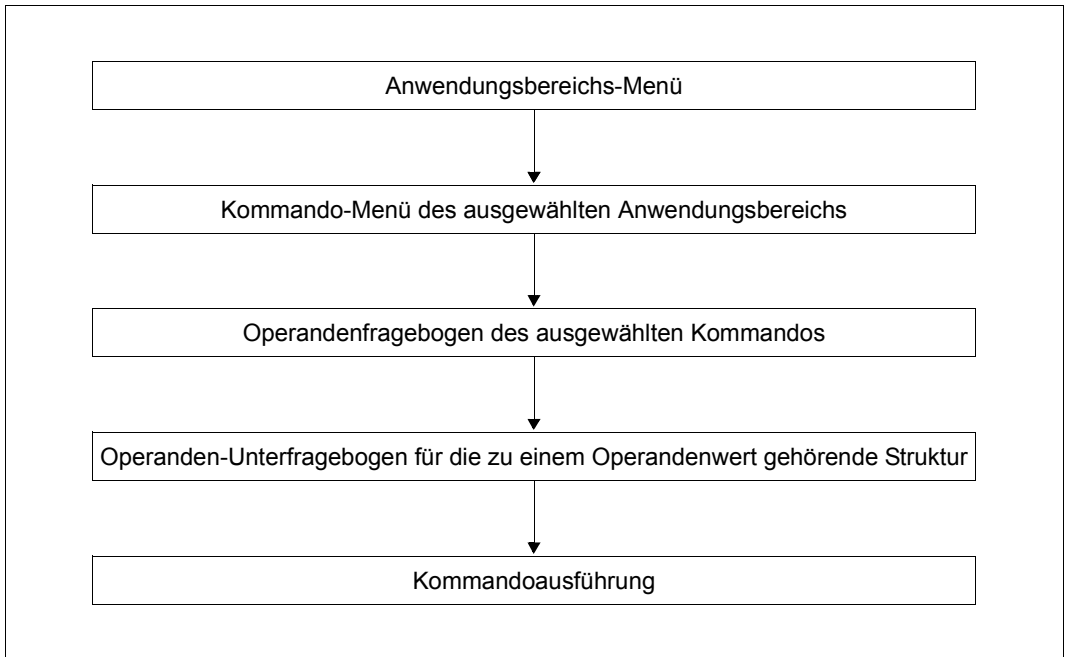


Bild 2: Weg vom Anwendungsbereichs-Menü zur Kommandoausführung

### 3.2.3 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 58](#)).
- ?? 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.

### 3.2.4 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 [F7]-Taste und „+“ der [F8]-Taste im Style-Guide-Modus.
*EXECUTE	führt die aktuelle Operation (Kommando bzw. Anweisung) aus. Entspricht der [F3]-Taste im Old-Modus bzw. der [F11]-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 [F2]-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 [F12]-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 [K1]-Taste im Old-Modus bzw. der [F3]-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

	<p>*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.</p>
*REFRESH	<p>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.</p>
*DOM-MENU	<p>wechselt in das Anwendungsbereichs-Menü. Wird *DOM-MENU in einem Operandenfragebogen angegeben, so wird vor dem Wechsel die aktuelle Operation ausgeführt.</p>
<number>	<p>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.</p>
(<domain>)	<p>gibt das Kommando-Menü des Anwendungsbereichs &lt;domain&gt; aus. Wird (&lt;domain&gt;) in einem Operandenfragebogen angegeben, so wird vor dem Wechsel die aktuelle Operation ausgeführt.</p>
<command>?	<p>führt die aktuelle Operation aus und gibt anschließend den Operandenfragebogen des Kommandos &lt;command&gt; aus. Bereits angegebene Operandenwerte werden in den Fragebogen übernommen.</p>
<command>	<p>führt die aktuelle Operation aus und anschließend auch das Kommando &lt;command&gt;. &lt;command&gt; schließt die Angabe der zugehörigen Operanden mit ein.</p>
!<command>	<p>führt die aktuelle Operation aus und definiert anschließend für das Kommando &lt;command&gt; task-spezifische Default-Werte. &lt;command&gt; schließt die Angabe der zugehörigen Operanden mit ein.</p>
<statement>?	<p>führt die aktuelle Operation aus und gibt anschließend den Operandenfragebogen der Anweisung &lt;statement&gt; aus. Bereits angegebene Operandenwerte werden in den Fragebogen übernommen.</p>
<statement>	<p>führt die aktuelle Operation aus und anschließend auch die Anweisung &lt;statement&gt;. &lt;statement&gt; schließt die Angabe der zugehörigen Operanden mit ein.</p>
!<statement>	<p>führt die aktuelle Operation aus und definiert anschließend für die Anweisung &lt;statement&gt; task-spezifische Default-Werte. &lt;statement&gt; schließt die Angabe der zugehörigen Operanden mit ein.</p>

- ? 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.

### 3.2.5 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 3: 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 3: 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 4: 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 4: 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.

### 3.2.6 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.

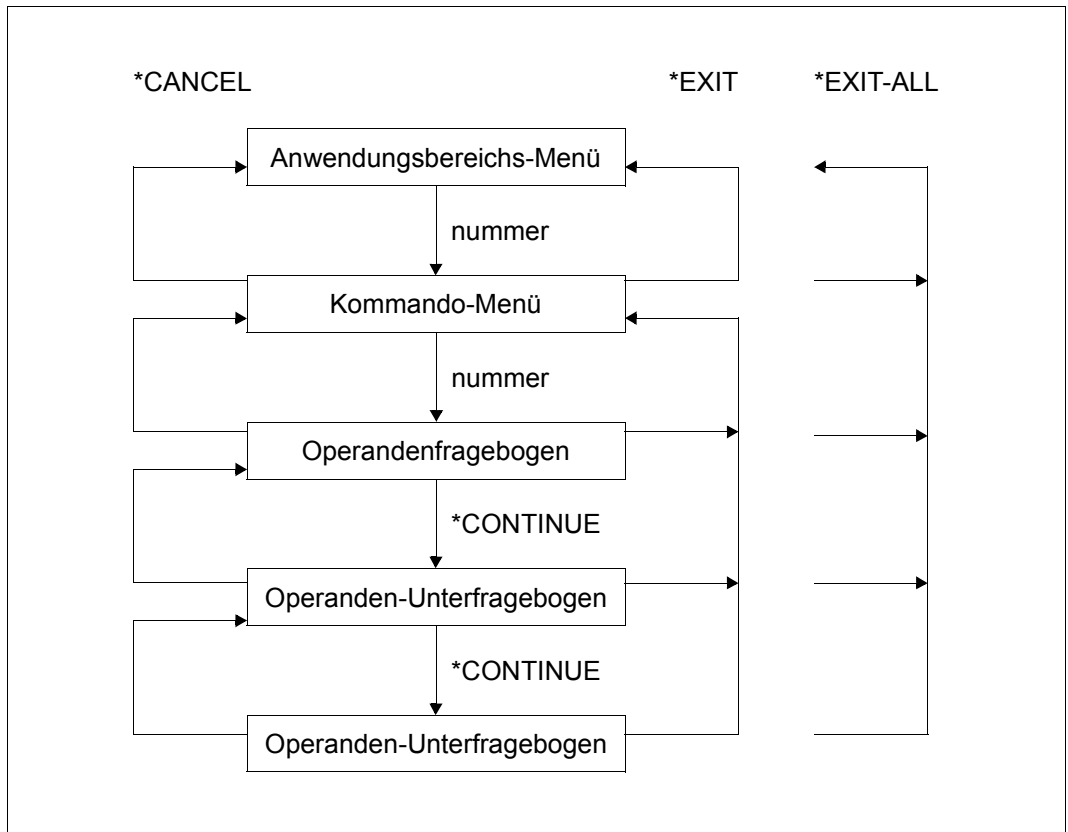


Bild 3: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Systemmodus (geführter Dialog)

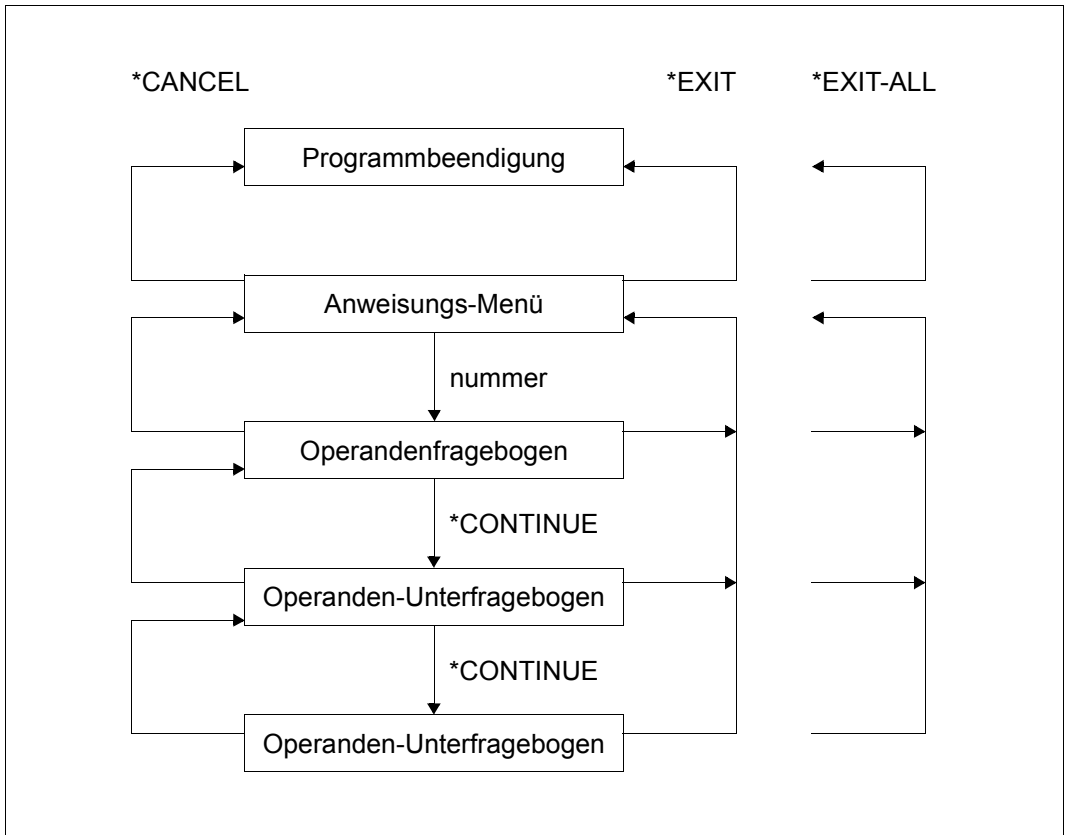


Bild 4: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Programmodus (geführter Dialog)

### 3.2.7 Anwendungsbereichs-Menü

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

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

Die hier aufgelisteten Anwendungsbereiche entsprechen dem Stand von BS2000/OSD-BC V8.0A. Anwendungsbereiche ohne Inhalt (Kommandos gesperrt oder nicht implementiert) werden nicht angezeigt.

*Anmerkungen:*

- 1) Der Anwendungsbereich enthält alle Kommandos, deren Eingabe im Dialog erlaubt ist.
- 2) Der Anwendungsbereich ersetzt langfristig den Anwendungsbereich MULTI-CATALOG.
- 3) Der Anwendungsbereich ersetzt langfristig den Anwendungsbereich RSO-SPOOL-ADMINISTRATION.
- 4) Der Anwendungsbereich ersetzt langfristig den Anwendungsbereich SPOOL.
- 5) Der Anwendungsbereich 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.

### 3.2.8 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

### 3.2.9 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 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.

### 3.2.10 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. Der Zeilenrest muss nicht gelöscht werden.
??	als Operandenwert liefert Informationen zu Datentypen, die als Operandenwert zugelassen sind. 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). 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.



### 3.2.11 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.

### 3.3 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 40](#)).

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 vorbesetzt. 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 vorbesetzt 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.

### 3.3.1 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.

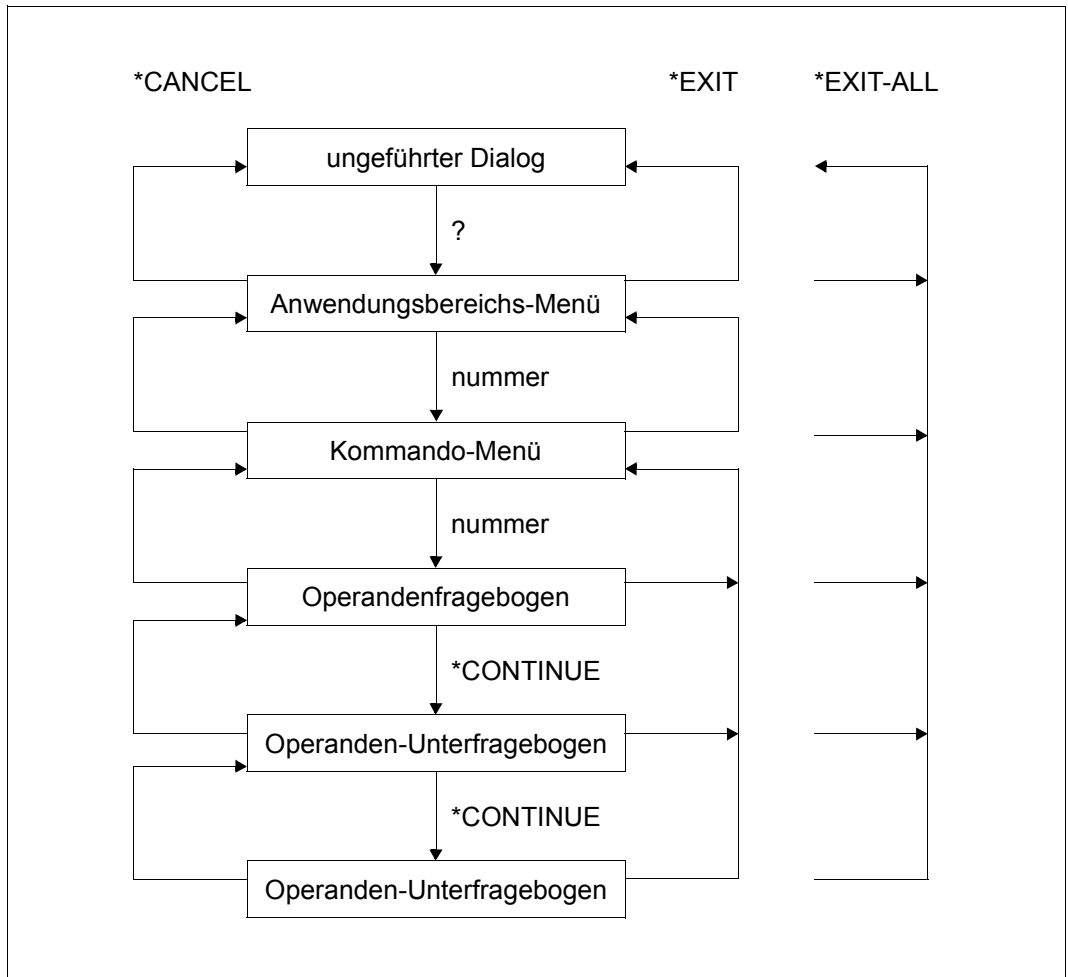


Bild 5: Die Rückzugsfunktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Systemmodus (temporär geführter Dialog)

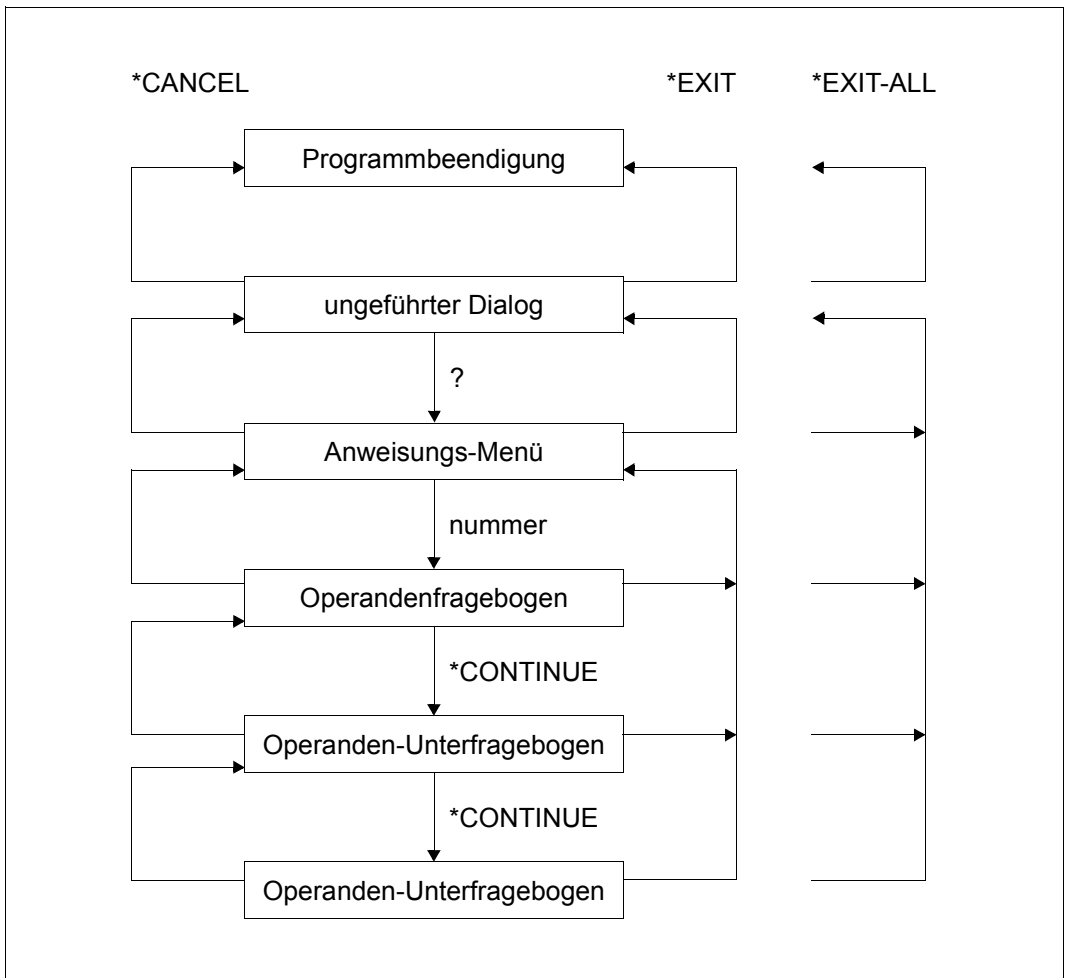


Bild 6: Wirkung der Funktionen \*CANCEL, \*EXIT und \*EXIT-ALL im Programmodus (temporär geführter Dialog)

## 3.4 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.

### 3.4.1 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 93](#) beschrieben.

#### Implementierte Prozeduren

Der Benutzer kann eigene Kommandos in einer Benutzersyntaxdatei definieren (siehe [Seite 139](#) und [145](#)). Er erstellt dabei für das Kommando eine Syntaxbeschreibung, die den Kommandonamen, die Operandennamen und die jeweils möglichen Operandenwerte ent-

hält. 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“ [5]).

LOGON-Prozeduren dürfen kein DO-Kommando enthalten und dürfen nicht mit dem Kommando ENDP-RESUME beendet werden. Außerdem sollten sie nicht unterbrechbar sein (siehe „[Unterbrechungsschutz von Prozeduren](#)“ auf Seite 65).

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 „[Verwaltung systemweiter Prozedurdateien](#)“ auf Seite 68).

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

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

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).

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

### 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“ [5]).

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 „[Verwaltung systemweiter Prozedurdateien](#)“ auf Seite 68). System-LOGOFF-Prozeduren werden erst nach Ablauf der vereinbarten Benutzer-LOGOFF-Prozeduren gestartet. Benutzer- und Gruppensyntaxdateien werden vor ihrem Start deaktiviert.

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



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

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).

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

### 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
	BKPT-Makro	dia	prog	dia	cmd
Programm (Forts.)	CMD-Makro	any	prog	same	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 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 programm aufrufende Prozedur nichtunterbrechbar ist

## Verwaltung systemweiter Prozedurdateien

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

- \$TSOS.SYS.SDF.LOGON.SYSPROC (Call-Prozedur für LOGON)
- \$TSOS.SYS.SDF.LOGON.SYSINCL (Include-Prozedur für LOGON)
- \$TSOS.SYS.SDF.LOGOFF.USERPROC (Call-Prozedur für LOGOFF)
- \$TSOS.SYS.SDF.LOGOFF.USERINCL (Include-Prozedur für LOGOFF)

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

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

### 3.4.2 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.

## 3.5 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 Batchbetrieb 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 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, da eine Änderung dieses Verhaltens nicht auszuschließen ist.

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 Jobvariablenamen in der Form `&(jv-name)`.
- indirekt über ihren Jobvariablenkettungsname 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`.

### *Hinweis*

Vor der Jobvariablenersetzung ersetzt ACS ggf. den Aliasnamen einer JV durch ihren realen Pfadnamen (gemäß Aliaskatologeintrag).

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 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 75](#)).
- 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 Builtin-Funktion `JV( )` im Handbuch „Kommandos“ [1] bzw. im Handbuch „SDF-P“ [5]. 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 einer S-Variablen, einer Built-in-Funktion oder ein gültiger S-Variablenausdruck sein kann. Für eine S-Variable, deren Name keinen Punkt enthält, kann der Ausdruck auch in der Form „&s-variable“ 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)
/&(amp;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)
%      3 :20SG:$USER1.PROC.JV _____ (6)
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES

```

- (1) Es wird der Jobvariablenname „EGON“ vereinbart.
- (2) Die Jobvariable EGON 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 EGON wird geändert. Er enthält jetzt nur noch einen Teil des Kommandonamens('-FILE-ATTRIBUTES') und den teilqualifizierten Dateinamen 'PROC.'.
- (5) Der Jobvariablen EGON 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
/(&cmd) proc.jv ----- (2)
%          3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      1 FILE RES=          3 FRE=          2 REL=          0 PAGES
/cmd='PRINT-DOCUMENT' ----- (3)
/sh-var cmd
CMD = PRINT-DOCUMENT
/(&cmd) proc.jv ----- (4)
% SCPO810 SPOOLOUT FOR FILE ':20SG:$USER1.PROC.JV' ACCEPTED. TSN: '1FAL', SPOOL
OUT-NAME: 'SDFTEST', MONJV: '*NONE'
% SCP1025 PRINT JOB ACCEPTED BY SERVER 'GH5090Y0' WITH TSN '5BXC'
/(&(:20sg:cmd) proc.jv ----- (5)
%          3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      1 FILE RES=          3 FRE=          2 REL=          0 PAGES
/(&(jv(jv-name='CMD')) proc.jv ----- (6)
%          3 :20SG:$USER1.PROC.JV
%:20SG: 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. Name der zu ersetzenden Jobvariablen wird als Prozedurparameter einer Nicht-S-Prozedur übergeben:

```

/BEGIN-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 (4).
- (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:

```

/show-jv jv=kmd ----- (1)
%SHOW-FILE-ATTR
/call-proc do.jvtest,log=*yes
%          1 1 /SET-PROC-OPT  JV-REPLACE=*AFTER-BUILTIN
%          2 1 /DECL-PAR      JV-1(INIT=*PROMPT)
%JV-1: kmd ----- (2)
%          3 1 /SHOW-FILE-ATTR  FILE-NAME=LST.JOB ----- (3)
%          3 :N:$USER0001.LST.JOB
%:N:      PUBLIC:      1 FILE RES=          3 FRE=          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 KMD wird SHOW-FILE-ATTR angezeigt.
- (2) Nach Aufruf der Prozedur DO.JVTEST wird die Prozedurparameter JV-1 gepromptet und der Wert KMD 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 KMD ersetzt. Voraussetzung ist, dass keine gleichnamige S-Variable oder Builtin-Funktion existiert und im SET-PROCEDURE-OPTIONS die Jobvariablen-Ersetzung explizit eingestellt ist.

## 3.6 Verkürzte Eingabe

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

### 3.6.1 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 Schlüsselwortwert von einem weiteren möglichen Operandenwert, dessen Wertebereich die Zeichenfolge des Schlüsselwortwertes beinhaltet. Der Stern allein stellt selbst bei gegebener Eindeutigkeit keine gültige Abkürzung dar.
- Schlüsselwortwerte werden 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 führendem Stern dargestellt wurden, auch ohne Stern akzeptiert.
- Der Name bzw. ein Teilname eines Schlüsselwortwertes kann auch das Zeichen Punkt enthalten (z.B. \*V4.7 oder \*OSD-V8.0). 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 Namens 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 `/CR-F`. Sollen Prozeduren gegen Umbenennungen invariant bleiben so, müssen alle Namen in der Langform angegeben werden.

### 3.6.2 Default-Wert übernehmen

Die Angabe der meisten Operanden ist wahlfrei. Wahlfreie Operanden sind bereits mit einem Operandenwert vorbesetzt, dem so genannten Default-Wert. 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 93](#)).

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`).

### 3.6.3 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.



### 3.6.4 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.

*Beispiel*

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

ist die verkürzte Schreibweise von

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

## 3.7 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.

### 3.7.1 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.

### 3.7.2 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.

### 3.7.3 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.

## 3.8 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 Anweisungsmenü vornehmen. Eine weitere Auswahlmöglichkeit besteht bei der Verwendung von Musterzeichen oder mehrdeutigen Abkürzungen in Kommando- bzw. Anweisungsamen.

### 3.8.1 Musterzeichen in Kommando- oder Anweisungsamen

Bei Eingabe eines Kommando- bzw. Anweisungsamens 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. Anweisungsamen darf maximal 30 Zeichen lang sein.
- Das Musterzeichen „/“ zu Beginn eines Kommandonamens wird als Kommando-Prompt interpretiert.
- Die Zeichenfolge „//“ zu Beginn eines Anweisungsamens 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. Anweisungsmenü. Ein temporärer Dialog wird beendet.

### 3.8.2 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 Anwendungsreichs- bzw. Anweisungsmenü. Ein temporärer Dialog wird beendet.

## 3.9 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.



## 3.10 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 geprüft, 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“ [5]). 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:*

```

/c!p 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.

## 3.11 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 `SYS-OUT` 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. Darüber hinaus 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).

---

## 4 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 unter BS2000/OSD-BC V8.0A durchgeführt. Im Beispiel meldet sich der Benutzer unter der Benutzerkennung **USER1** im System an. Für die Benutzerkennung USER1 ist im Katalogeintrag der Sprachschlüssel D eingetragen. Der SDF-Dialog und die Meldungs Ausgabe erfolgen deshalb in deutscher Sprache (bei Bedarf kann die Spracheinstellung mit dem Kommando MODIFY-MSG-ATTRIBUTES angepasst werden). Zur Demonstration der Eingabemöglichkeiten wurde das häufig verwendete Benutzer-Kommando SHOW-FILE-ATTRIBUTES verwendet. Eine vollständige Beschreibung des Kommandos enthält das Handbuch „Kommandos“ [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 eigenen 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 entsprechende Funktionstaste, muss die entsprechende Steueranweisung bzw. das entsprechende Kommando in der NEXT-Zeile eingegeben werden (siehe [Abschnitt „Funktionstasten“ auf Seite 47](#)).

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.

## 4.1 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 'Vnnn', 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 dd-mm-yy 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 Auswahlnueue 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(SPECIAL-FUNCTIONS=*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.  
%
```

## 4.2 Ungeführter Dialog

### 4.2.1 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.

## 4.2.2 NO-Form

Der Dialog wird folgendermaßen fortgesetzt:

```
%KDO:sh-f-a test.,file-struct=i _____ (1)
% CMD0187 DIE ABKUEZUNG '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 Stellungoperand) und weist ihn als fehlerhaft zurück. Das hinter dem Wert \*PARAMETERS stehende Klammersymbol 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.

### 4.3 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.

```
%KD0: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-STRUCTU
              RE=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.

## 4.4 Geführter Dialog

### 4.4.1 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 FEHLERHAFTE(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

```

## 4.4.2 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-
                                   namen. 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 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
                             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                  : 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-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 Klammerspaar 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 *SYSLST() 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 Klammerspaar 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 ge- wechselt. 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
```



### 4.4.3 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
                                en 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                      : Ermoeoglicht 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 Dateikettungsamen 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	Art der auszugebenden Information
SELECT	= <b>*by(file-struct=(i,s))</b>	
	*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 Kommando-prozeduren
 17  PROGRAM                : Steuern von Programm-laeufen

-----
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      : Ermoeeglicht 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 SYSLSST 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.



---

## 5 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 können für eine Task eine Gruppen- und eine mehrere Benutzersyntaxdateien aktiviert sein. Für diesen Fall gibt es eine feste Dateihierarchie, die festlegt, wie SDF mit mehreren Syntaxdateien arbeitet.

## 5.1 Systemsyntaxdatei

Die Benutzung von SDF setzt voraus, dass eine Basis-Systemsyntaxdatei vorhanden ist, die automatisch beim Systemladen aktiviert wird. Daneben können 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.047` 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 Systemeinleitung wird sie automatisch aktiviert. Ist der Pubset, auf dem sie abgelegt ist, bei der Systemeinleitung 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.

## 5.2 Gruppensyntaxdatei

Eine Gruppensyntaxdatei kann, muss aber nicht aktiviert sein. Die Systembetreuung ordnet sie der Benutzerkennung 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 Benutzerkennung 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 Technology Solutions angebotenen Funktionsumfang in der Systemsyntaxdatei systemweit eingeschränkt, so kann sie über eine Gruppensyntaxdatei diese Einschränkung gezielt für eine Benutzerkennung aufheben. Andererseits kann sie systemweit angebotene Funktionen über eine Gruppensyntaxdatei gezielt für eine Benutzerkennung 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.

### 5.3 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.

## 5.4 Privilegien in Syntaxdateien

In BS2000/OSD-BC 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 Benutzererkennung 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.

## 5.5 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.

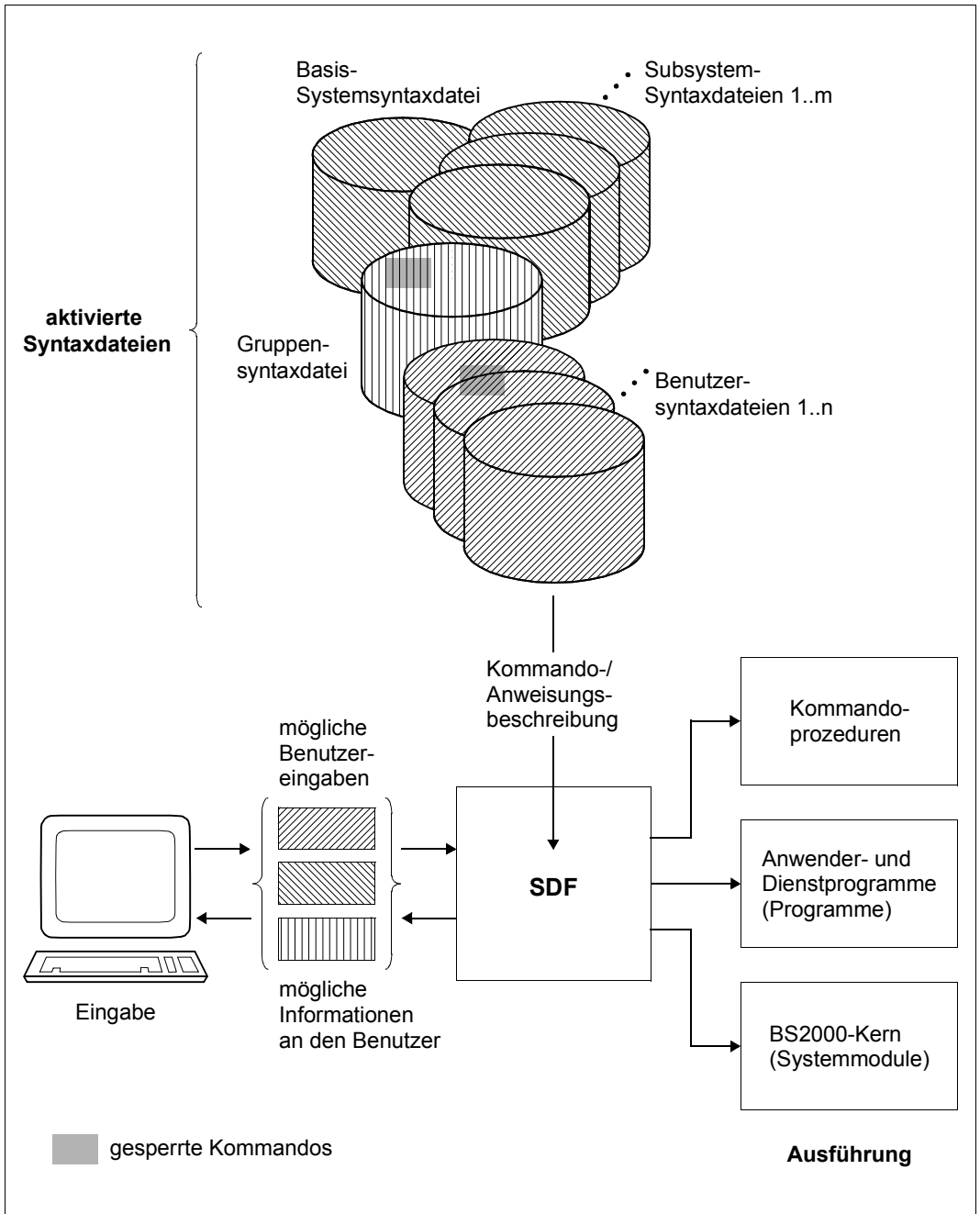


Bild 50: Arbeitsweise von SDF bei Aktivierung verschiedener Syntaxdateitypen



## 5.6 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 Anweisungsanordnung 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“ auf Seite 173](#) beschrieben.

- Konvertieren einer Syntaxdatei im alten Format. Mit SDF V2.0A änderte sich der interne logische Aufbau von Syntaxdateien. SDF unterstützt auch das alte Format, jedoch mit verringerter Performance. Wenn noch eine Syntaxdatei im alten Format existiert, kann sie mit der SDF-I-Anweisung CONVERT-SYNTAX-FILE in das neue Format konvertiert werden (siehe [„Kompatibilität von Syntaxdatei-Formaten“ auf Seite 171](#)).
- 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.

---

## 6 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.

### 6.1 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 und im Handbuch „Kommandos“ [1] beschrieben:

Kommando	Bedeutung
HELP-SDF	Informationen über SDF ausgeben
MODIFY-SDF-OPTIONS	Benutzer-Syntaxdatei aktivieren und SDF-Optionen ändern
MODIFY-SDF-PARAMETERS	SDF-Parameter ändern
REMARK	Bemerkungen in Kommandodatei einfügen
RESET-INPUT-DEFAULTS	Task-spezifische Default-Werte löschen
RESTORE-SDF-INPUT	Bereits erfolgte Eingabe wiederherstellen
SHOW-CMD	Syntaxbeschreibung eines Kommandos ausgeben
SHOW-INPUT-DEFAULTS	Task-spezifische Default-Werte ausgeben
SHOW-INPUT-HISTORY	Gespeicherte Eingaben nach SYSOUT ausgeben
SHOW-SDF-OPTIONS	Aktive Syntaxdateien und SDF-Optionen anzeigen
SHOW-SDF-PARAMETERS	Einträge der SDF-Parameterdatei ausgeben
SHOW-SYNTAX-VERSIONS	Versionsstände von Syntaxdateien anzeigen
WRITE-TEXT	Text auf SYSOUT bzw. SYSLST ausgeben

Tabelle 5: Kommandos aus dem Anwendungsbereich SDF

## 6.2 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 140](#)) 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 155](#)).

Die Anweisung wird abgewiesen, wenn die Programmunterbrechung innerhalb einer Prozedur oder durch das Programm selbst momentan nicht zugelassen wird (z.B. aus Sicherheitsgründen).

<b>EXECUTE-SYSTEM-CMD</b>
<b>CMD = &lt;text 0..1800 with-low&gt;</b>

**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</b> = * <b>LAST</b> / <alphanum-name 4..7> <b>,INFORMATION-LEVEL</b> = * <b>MAXIMUM</b> / * <b>MEDIUM</b> / * <b>MINIMUM</b> <b>,LANGUAGE</b> = * <b>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 Programmodus.

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).

<b>HOLD-PROGRAM</b>

## MODIFY-SDF-OPTIONS

Mit der Anweisung MODIFY-SDF-OPTIONS kann der Benutzer während des Programm- laufs eine Benutzersyntaxdatei aktivieren bzw. deaktivieren und SDF-Einstellungen verän- dern.

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> *ADD(...)   <b>ADD-NAME = *<u>STD</u> / list-poss(2000): *<u>STD</u> / &lt;filename 1..54&gt;</b> *REMOVE(...)   <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> *ON(...)   <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 Kommen- tar und ist ausschließlich für die Dokumentation des Auftrags- bzw. des Prozedurablaufs von Bedeutung. Ein Semikolon außerhalb von Klammern wird als Anweisungstrenner inter- pretiert, 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 = *<u>STMT</u>(...)/ *<u>CMD</u> (...)/ *<u>ALL</u> / &lt;integer 1..9999&gt;</b> <b>*<u>STMT</u>(...)</b> <b>STMT = *<u>ALL</u> / &lt;structured-name 1..30 with-wild(50)&gt;</b> <b>,PROGRAM = *<u>CURRENT</u> / *<u>ALL</u> / &lt;structured-name 1..30&gt;</b> <b>*<u>CMD</u>(...)</b> <b>CMD = *<u>ALL</u> / &lt;structured-name 1..30 with-wild(50)&gt;</b>	

## 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 = *<u>LAST-STMT</u> / &lt;integer -100..-1&gt; / &lt;integer 1..9999&gt;</b>	

### **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
<p><b>OBJECT = <u>*STMT</u>(...) / *CMD(...) / *ALL</b></p> <p><b><u>*STMT</u>(...)</b></p> <ul style="list-style-type: none"> <li><b>STMT = <u>*ALL</u> / &lt;structured-name 1..30 with-wild(50)&gt;</b></li> <li><b>,PROGRAM = <u>*CURRENT</u> / *ALL / &lt;structured-name 1..30&gt;</b></li> </ul> <p><b>*CMD(...)</b></p> <ul style="list-style-type: none"> <li><b>CMD = <u>*ALL</u> / &lt;structured-name 1..30 with-wild(50)&gt;</b></li> </ul> <p><b>,OUTPUT = <u>*SYSOUT</u> / *SYSLST(...)</b></p> <p><b>*SYSLST(...)</b></p> <ul style="list-style-type: none"> <li><b>SYSLST-NUMBER = <u>*STD</u> / &lt;integer 1..99&gt;</b></li> </ul> <p><b>,INPUT-SERIAL-NUMBER = <u>*NO</u> / *YES</b></p>	

## 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
<p><b>ENTRIES = <u>g</u> / &lt;integer 1..100&gt; / *ALL</b></p> <p><b>,SELECT = <u>*STMT</u> / *CMD / *ALL</b></p> <p><b>,PATTERN = <u>*NONE</u> / &lt;structured-name 1..30&gt;</b></p> <p><b>,INPUT-SERIAL-NUMBER = <u>*NO</u> / *YES</b></p>	

## 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.

**SHOW-SDF-OPTIONS**

Kurzname: **SHSDFO**

**INFORMATION = \*ALL / \*USER**

## 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).

**SHOW-STMT**

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

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

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

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

**,OUTPUT = \*SYSOUT / \*SYSLST(...)**

**\*SYSLST(...)**

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

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

## 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“ [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>
<p><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;</p>	

## 6.3 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 Auswahlménü für definierte Objekte und liefert das Auswahlergebnis 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 83</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.

## 6.4 SDF-Schnittstelle zu höheren Programmiersprachen

SDF bietet eine Schnittstelle für Programme in höheren Programmiersprachen an (**High Level Languages**). 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!



---

## 7 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.





---

## 8 Installation von SDF

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

### 8.1 SDF-Installationsdateien

SDF V4.7 ist unter BS2000/OSD-BC V8.0 bzw. OSD/XC V4.0 ablauffähig.

Die folgenden Dateien sind Bestandteil der Standardlieferung und werden mit SDF V4.7 ausgeliefert:

Datei	Bemerkung
SIPLIB.SDF.047	TPR-Makrobibliothek
SPMLNK.SDF.047	Ladebibliothek für SDF (für SQ-Server)
SKMLNK.SDF.047	Ladebibliothek für SDF (für SX-Server)
SYSFGM.SDF.047.D	Freigabemitteilung für SDF (deutsch)
SYSFGM.SDF.047.E	Freigabemitteilung für SDF (englisch)
SYSLIB.SDF.047	PLAM-Bibliothek, die die SDF-Makros, die Definitionen der Schnittstellen-Funktionen für COBOL, FORTRAN und C, das Bin-demodul CMDCSTM und die Schnittstellen-Module für die höheren Programmiersprachen enthält.
SYSLNK.SDF.047	Ladebibliothek für SDF (für S-Server)
SYSMES.SDF.047	SDF-Meldungsdatei
SYSRMS.SDF.047	RMS-Liefermenge für SDF
SYSSDF.SDF.047	Syntaxdatei für die Kommandos aus dem Anwendungsbereich SDF
SYSSII.SDF.047 <sup>1</sup>	Installations-Informationsdatei für die Installation mit IMON
SYSSSC.SDF.047	SSCM-Katalog-Deklarationen für das Subsystem SDF

Datei	Bemerkung
SYSMES.SDF-I.041	Meldungsdatei für SDF-I
SYSPRG.SDF-I.041	Programm zum Mischen von Softwareeinheitssyntaxdateien (siehe <a href="#">Kapitel „SDF-I“ auf Seite 167</a> )
SYSSDF.SDF-I.041	Syntaxdatei, die das Kommando START-SDF-I enthält
SYSSII.SDF-I.041 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON
SYSPRG.SDF-U.041	Programm, mit dem die Systembetreuung Syntaxdateien modifizieren kann (siehe <a href="#">Kapitel „SDF-U“ auf Seite 189</a> )
SYSSDF.SDF-U.041	Syntaxdatei, die die SDF-U-Anweisungen enthält
SYSLNK.SDF-U.041	Programmmodule von SDF-U
SYSMES.SDF-U.041	Meldungsdatei für SDF-U
SYSRMS.SDF-U.041	RMS-Liefermenge für SDF-U V4.1
SYSPRG.SDF-PAR.011	Programm zu Erzeugen, Ändern und Anzeigen der SDF-Parameterdatei (siehe <a href="#">Kapitel „SDF-PAR“ auf Seite 219</a> )
SYSMES.SDF-PAR.011	Meldungsdatei für SDF-PAR
SYSPRC. SDF-PAR.011.112	Startprozedur für SDF-PAR
SYSSDF.SDF-PAR.010.USER	Benutzersyntaxdatei, die die SDF-PAR-Anweisungen enthält
SYSSII.SDF-PAR.011 <sup>1)</sup>	Installations-Informationsdatei für die Installation mit IMON

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

## 8.2 Subsystemkatalog generieren

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

Die Deklarationen dazu werden in der Datei **SYSSSC.SDF.047** ausgeliefert.

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

Im Normalfall erfolgt die Generierung des Subsystemkatalogs durch IMON.

## 8.3 SDF bereitstellen

### 8.3.1 Dateien für SDF bereitstellen

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



IMON aktiviert die Syntaxdatei von SDF (d.h. die Datei SYSSDF.SDF.047) als Basis-Systemsyntaxdatei. Alle anderen Systemsyntaxdateien einschließlich SYSSDF.BS2CP.<bs2vers> werden als Subsystem-Syntaxdateien aktiviert.

### 8.3.2 SDF starten

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



---

## 9 Verhalten im Fehlerfall

### **SDF-Parameterdatei kann beim Laden nicht geöffnet werden**

Kann die Parameterdatei beim Laden von SDF nicht geöffnet werden, fordert SDF über eine Konsolmeldung eine neue Parameterdatei an. Bei der Antwort „\*STD“ aktiviert SDF die Datei \$TSOS.SYSSDF.SDF.047 als Basis-Systemsyntaxdatei und die Datei \$TSOS.SYSSDF.BS2CP.<bs2vers> als Subsystem-Syntaxdatei. Nach dem Laden von SDF kann die Parameterdatei gelöscht und eine neue Parameterdatei mit MODIFY-SDF-PARAMETERS oder dem Dienstprogramm SDF-PAR erstellt werden.

### **Basis-Systemsyntaxdatei kann beim Laden nicht aktiviert werden**

Wenn die Basis-Systemsyntaxdatei beim Laden nicht aktiviert werden kann, wird eine Fehlermeldung ausgegeben und eine neue Systemsyntaxdatei wird angefordert. Gibt der Operator dann eine gültige Basis-Systemsyntaxdatei an, so wird diese aktiviert, ihr Name jedoch nicht in die Parameterdatei eingetragen. Wird keine gültige Systemsyntaxdatei angegeben, wird der Ladevorgang abgebrochen. In diesem Fall muss das System vom Backup-Pubset geladen werden. Steht ein solches nicht zur Verfügung, so ist ein Restaurieren der Platten mit dem Dienstprogramm FDDRL (siehe Handbuch „FDDRL“ [8]) oder eventuell eine Systemneugenerierung erforderlich.

### **Systemsyntaxdatei nicht mit USER-ACCESS=\*SPECIAL katalogisiert**

Wenn eine Systemsyntaxdatei zwar mehrbenutzbar, aber nicht mit USER-ACCESS=\*SPECIAL katalogisiert ist, wird eine Fehlermeldung ausgegeben und das Attribut USER-ACCESS auf \*SPECIAL gesetzt. Wenn die Systemsyntaxdatei nicht mehrbenutzbar ist, muss der Operator diese Attribut-Änderung bestätigen. Tut er das nicht, dann kann die Systemsyntaxdatei nicht aktiviert werden.

### **Basis-Systemsyntaxdatei ohne Kommando MODIFY-SDF-PARAMETERS**

Wenn die Basis-System-Syntaxdatei nicht das Kommando MODIFY-SDF-PARAMETERS enthält, wird eine Fehlermeldung ausgegeben. Der Operator kann in diesem Fall den Ladevorgang fortsetzen, abbrechen oder den Namen einer anderen Basis-Systemsyntaxdatei eingeben.



---

## 10 SDF-I

**Versionsstand:** SDF-I V4.1

**Privilegierung:** **STD-PROCESSING** (für nicht privilegierte Anweisungen)  
**TSOS** (für privilegierte Anweisungen)

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



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

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

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

Folgende Dateien können Eingabedateien sein:

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

Folgende Dateien können in die Eingabesyntaxdatei eingemischt werden:

- SESDs.
- jede andere, vom Benutzer mit SDF-A erstellte Syntaxdatei (siehe Handbuch „SDF-A“ [4]).

Wenn noch Syntaxdateien (Benutzer, Gruppe, System) im alten Format existieren, können diese mit der Anweisung CONVERT-SYNTAX-FILE in das neue Format konvertiert werden (siehe auch „Kompatibilität von Syntaxdatei-Formaten“ auf Seite 171).



**Achtung!**

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

Weitere Informationen dazu finden Sie im Abschnitt „Hinweise“ ab Seite 171.

## 10.2 Aufruf und Ablauf von SDF-I

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

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

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

**VERSION =**

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

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

**VERSION = \*STD**

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

**VERSION = <product-version>**

Angabe der SDF-I-Version im Format mm.n[a[so]].



**MONJV =**

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

**MONJV = \*NONE**

Es wird keine Monitor-Jobvariable verwendet.

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

Name der zu verwendenden Jobvariablen.

**CPU-LIMIT =**

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

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

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

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

Es soll nur die angegebene Zeit verwendet werden.

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

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

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

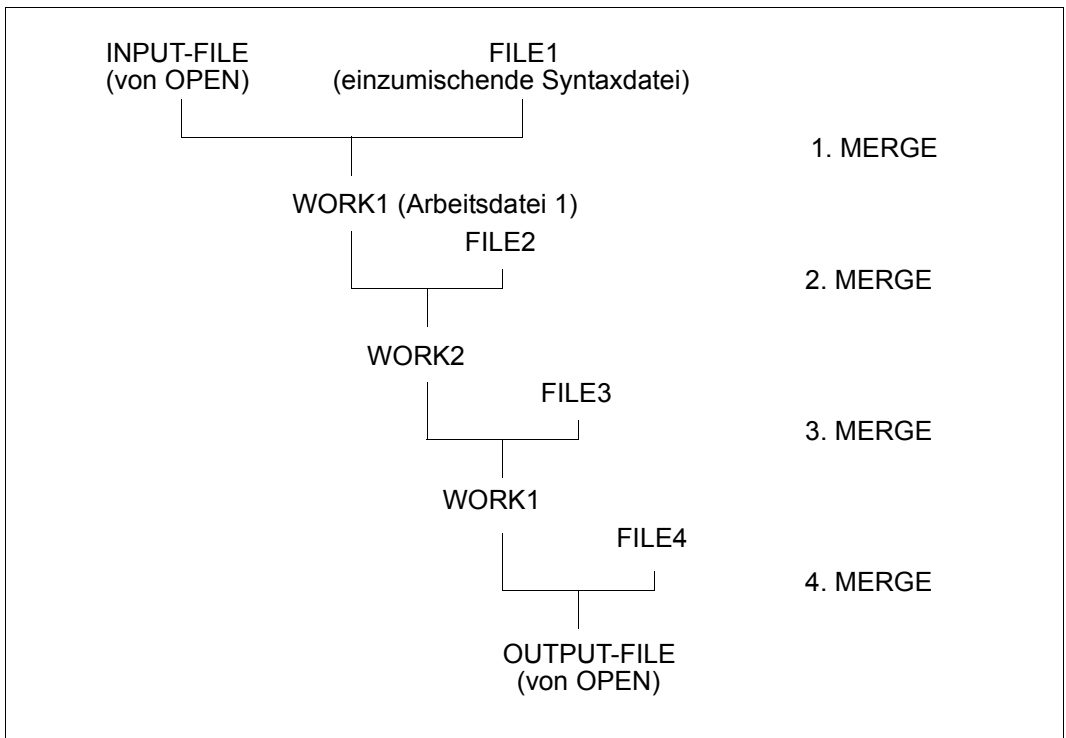
Nach jeder MERGE-Anweisung wird eine Arbeitsdatei erzeugt. Diese Arbeitsdatei enthält alle bis dahin eingemischten Syntaxdateien einschließlich der Eingabesyntaxdatei. Mit der END-Anweisung oder der OPEN-Anweisung für die nächste Syntaxdatei wird die Arbeitsdatei in die Ausgabedatei umgewandelt. Mit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

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

## 10.3 Behandlung von Arbeitsdateien

SDF-I benutzt maximal zwei Arbeitsdateien, die von SDF-I angelegt und wieder gelöscht werden. Sie werden unter dem Namen SYSSDF.SDF-I.version.tsn.n angelegt. Dabei bezeichnet *version* die SDF-I-Version (z.B. 041 bei SDF-I V4.1), *tsn* die TSN der aufrufenden Task und *n* mit den Werten 1 oder 2 die Nummer der Arbeitsdatei.

Der Benutzer sollte darauf achten, dass keine eigenen Dateien mit diesen Namen existieren. Beim Einmischen von mehr als einer Syntaxdatei in die Eingabesyntaxdatei (Operand INPUT-FILE in der OPEN-Anweisung) werden zwei Arbeitsdateien im Wechsel verwendet. Sie sehen im Bild, wie die beiden Arbeitsdateien im Mischvorgang verwendet werden.



Verwendung von zwei Arbeitsdateien beim Mischen

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

## 10.4 SDF-Standardanweisungen und Anwendungsbereiche

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

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

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

## 10.5 Hinweise

### Eingaben

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

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

### Kompatibilität von Syntaxdatei-Formaten

Mit der Weiterentwicklung von SDF waren in der Vergangenheit manchmal auch Änderungen am Syntaxdateiformat verbunden. Dabei entstanden die Formate V1, V2, V3, V4 und zuletzt das jetzt aktuell verwendete Format V4.1. Die Formate V1 bis V4 sind historisch und können nur noch bei älteren Syntaxdateien auftreten. Das Format einer Syntaxdatei ist in den Ausgabeinformationen der Anweisung SHOW-SYNTAX-FILE enthalten.

Mit dem Übergang von Format V1 auf V2 änderte sich auch der interne logische Aufbau einer Syntaxdatei. Deshalb werden Syntaxdateien ab dem Format V2 auch als Syntaxdateien im „neuen“ Format bezeichnet (V1 war das „alte“ Format).

Bei der Verarbeitung mit SDF-I ist Folgendes zu beachten:

- SDF-I verarbeitet nur Eingabesyntaxdateien im neuen Format (ab Format V2). Falls noch Syntaxdateien im alten Format existieren, können Sie diese mit der Anweisung CONVERT-SYNTAX-FILE in das neue Format konvertieren.
- Syntaxdateien im aktuellen Syntaxdateiformat (V4.1-Format) können nicht mit SDF-I-Versionen < 4.1 verarbeitet werden.
- Beim Mischen hängt das Format der Ausgabesyntaxdatei vom Format der Eingabesyntaxdateien ab. SDF-I erzeugt die Ausgabesyntaxdatei mit dem höchsten Format, das unter den Eingabesyntaxdateien vorkommt. Wenn z.B. nur Syntaxdateien mit V2- und V3-Format eingemischt werden, erhält die Ausgabesyntaxdatei das Format V3.
- SDF-I trägt als Name des Produktes SDF-A „SDF-A“ in die Ausgabesyntaxdatei ein. Bei Syntaxdateien im alten Format (OLD bzw. V1-Format) ist der Name von SDF-A noch mit „SDFA“ festgelegt. Wenn Sie eine solche Syntaxdatei mit SDF-I bearbeiten, sollten Sie sich mit der Anweisung SHOW-SYNTAX-FILE den tatsächlich eingetragenen Namen anzeigen lassen und nur diesen Namen in den SDF-I-Anweisungen verwenden.
- Während einer OPEN/MERGE-Sequenz werden Produkte, die schon in der Syntaxdatei enthalten sind, beim Mischen durch die neuen Versionen ersetzt. Eine Ausnahme bildet das Produkt SDF bei Versionen < V4.0.

## 10.6 SDF-I-Anweisungen

### 10.6.1 Funktionelle Übersicht

#### Anzeigefunktion

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

#### Syntaxdateien konvertieren

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

#### Syntaxdateien mischen und entmischen *(Privilegierte Funktionen)*

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

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

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

#### SDF-I beenden

END                      beendet den SDF-I-Lauf

## 10.6.2 Beschreibung der Anweisungen

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

**Privilegierung:**            **STD-PROCESSING, TSOS**

Die Anweisung CONVERT-SYNTAX-FILE konvertiert eine beliebige Syntaxdatei (Gruppe, System, Benutzer) mit altem Format (Format V1) in eine Syntaxdatei mit neuem Format (Format V4.1).

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

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

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

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

Bestimmt den Namen der Ausgabesyntaxdatei mit der neuen Struktur.

#### Hinweise

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

**END**  
**SDF-I beenden****Privilegierung:       STD-PROCESSING, TSOS**

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

END

Diese Anweisung hat keine Operanden.

## MERGE

### Syntaxdateien mischen

**Privilegierung:** TSOS

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

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

**Format 1:** Einmischen von Softwareeinheitssyntaxdateien

#### MERGE

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

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

Gibt die Syntaxdatei an, die mit der unter OPEN angegebenen Eingabedatei gemischt werden soll. Wurden nach dem OPEN schon MERGE-Anweisungen ausgeführt, so wird die Syntaxdatei in die gerade verwendete Arbeitsdatei eingemischt, in der die vorher mit MERGE eingemischten Syntaxdateien schon vorhanden sind (siehe [Seite 170](#)).

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

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

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

Eine Prüfung auf Namenskonsistenz findet nicht statt.

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

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



**CHECK-VERSION =**

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

**CHECK-VERSION = \*NO**

Eine Prüfung auf Versionskonsistenz findet nicht statt.

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

Versionsnummer einer Softwareeinheit. Die Versionsnummern der von Fujitsu Technology Solutions gelieferten SEDSs entsprechen dem SDF-Datentyp <product-version> (siehe Handbuch „Kommandos“ [1]).

Format: 'nn.naxx'.

nn.na = Versionsnummer der Softwareeinheit

xx = Änderungsstand der Syntaxdatei

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

**REPLACE-PRODUCT =**

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

**REPLACE-PRODUCT = YES**

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

**REPLACE-PRODUCT = NO**

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

**UPDATE-SDF-GLOBALS =**

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

**UPDATE-SDF-GLOBALS = YES**

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

**UPDATE-SDF-GLOBALS = NO**

Die Globalinformation der Eingabesyntaxdatei wird nicht überschrieben.

**Format 2:** Einmischen von selbsterstellten Syntaxdateien

**MERGE**

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

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

Gibt die Syntaxdatei an, die mit der unter OPEN angegebenen Eingabedatei gemischt werden soll. Wurden nach dem OPEN schon MERGE-Anweisungen ausgeführt, so wird die Syntaxdatei in die gerade verwendete Arbeitsdatei eingemischt, in der die vorher mit MERGE eingemischten Syntaxdateien schon vorhanden sind (siehe [Seite 170](#)).

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

Kurzname der Syntaxdatei (notwendig für späteres Löschen).

**REPLACE-PRODUCT =**

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

**REPLACE-PRODUCT = YES**

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

**REPLACE-PRODUCT = NO**

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

**Hinweise**

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

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

## OPEN

### Eingabe- und Ausgabesyntaxdatei zuweisen

**Privilegierung:** TSOS

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

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

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

#### OPEN

```

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

```

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

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

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

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

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

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

**FORMAT =**

Bestimmt, welches Format die Ausgabesyntaxdatei erhalten soll. Die Anweisung SHOW-SYNTAX-FILE (siehe [Seite 185](#)) gibt als Format einer Syntaxdatei die Werte V1, V2, V3, V4 oder V4.1 aus. Das aktuelle Format ist V4.1. Die anderen Formate sind historisch und können nur noch bei älteren Syntaxdateien auftreten.

**FORMAT = \*STD**

Die Ausgabesyntaxdatei wird mit der neuen Syntaxdateistruktur erzeugt, wenn mindestens eine Eingabesyntaxdatei mit der neuen Syntaxdateistruktur verarbeitet wird.

Nur wenn ausschließlich Eingabesyntaxdateien mit alter Struktur verarbeitet werden, wirkt \*STD wie FORMAT=\*OLD.

**FORMAT = \*NEW**

Die Ausgabesyntaxdatei wird explizit mit der neuen Syntaxdateistruktur erzeugt.

Die Eingabesyntaxdateien können beliebige Struktur aufweisen.

FORMAT=\*NEW wirkt nur, wenn anschließend mindestens eine MERGE-Anweisung ausgeführt wird. Eine nachfolgende REMOVE-Anweisung konvertiert nicht die gesamte Syntaxdatei, sondern nur die von REMOVE betroffenen Dateiinhalte (z.B. Globalinformationen, Kommandolisten, Programmlisten, ...).

**FORMAT = \*OLD**

*Der Operandenwert wird nur aus Kompatibilitätsgründen unterstützt.*

Syntaxdateien mit der alten Syntaxdateistruktur (entspricht Format V1) sollten nicht mehr erstellt werden. Außerdem verarbeitet SDF-I in diesem Fall auch nur Eingabesyntaxdateien mit der alten Syntaxdateistruktur.

**BLKCTRL =**

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

**BLKCTRL = \*STD**

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

**BLKCTRL = \*DATA**

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

**BLKCTRL = \*PAMKEY**

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

**WRITE-MODE =**

Legt fest, ob die Ausgabesyntaxdatei schon existiert oder nicht.

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

**WRITE-MODE = \*NEW**

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

**WRITE-MODE = \*REPLACE**

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

**Hinweis**

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

## REMOVE Syntaxdateien entmischen

**Privilegierung:** TSOS

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

**Format 1:** Entmischen von Softwareeinheitssyntaxdateien

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

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

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

**Format 2:** Entmischen selbsterstellter Syntaxdateien

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

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

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

## Beispiel

### *Softwareeinheits-Syntaxdatei entmischen*

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

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

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

- (1) Die Eingabesyntaxdatei *SF.base* und der Name der Ausgabesyntaxdatei *SF.out* werden festgelegt.
- (2) Das Produkt L2 wird aus der Eingabesyntaxdatei *SF.base* entfernt, da im nächsten Schritt die Syntaxdatei *SF.in* eingemischt wird, die eine neue Version des Produkts L2 enthält. So wird sichergestellt, dass in der Syntaxdatei keine Reste der Version 1 verbleiben (wenn in der Version 2 z.B. ein Kommando entfallen ist, würde dieses durch bloßes Einmischen der Version 2 nicht entfernt werden).
- (3) Die Syntaxdatei *SF.in* wird mit der Eingabesyntaxdatei *SF.base* in die interne SDF-I-Arbeitsdatei (siehe [Seite 170](#)) gemischt.
- (4) SDF-I wird beendet, die Arbeitsdatei erhält den mit OPEN festgelegten Namen *SF.out* und enthält folgende SESDs: L1 Version 1, L2 Version 2, L3 Version 1, L4 Version 1. Die SESD L2 ist nun in der neuen Version vorhanden.

## Hinweise

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



## SHOW-SYNTAX-FILE

### Informationen über eine Syntaxdatei anzeigen

**Privilegierung:**        **STD-PROCESSING, TSOS**

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

Die Anweisung gibt folgende Informationen aus:

- Name, Typ, Version und Format der Syntaxdatei
  - Das aktuelle Format ist V4.1. Die Formate V1 bis V4 sind historisch und können nur noch bei älteren Syntaxdateien auftreten.
- Versionsinformationen zu Produkten, die in der Syntaxdatei enthalten sind
- Globalinformationen aus der Syntaxdatei (ausschnittsweise)
- Privilegierung der Kommandos (nur bei INFORMATION=CMD-INTERFACE)

#### SHOW-SYNTAX-FILE

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

#### FILE =

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

#### FILE = \*CURRENT

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

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

#### FILE = \*INPUT-FILE

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

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

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

**INFORMATION =**

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

**INFORMATION = ALL-ATTRIBUTES**

Folgende Informationen werden ausgegeben:

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

**INFORMATION = VERSIONS**

Folgende Informationen werden ausgegeben:

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

**INFORMATION = GLOBALS**

Folgende Informationen werden ausgegeben:

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

**INFORMATION = CMD-INTERFACE**

Ausgegeben wird Name, Typ, Version und Format der Syntaxdatei und eine Liste der Kommandos, die in der Syntaxdatei definiert sind. Es wird auch die Privilegierung der Kommandos ausgegeben (siehe Beispiel, [Seite 187](#)).

**PRODUCT-NAME =**

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

**PRODUCT-NAME = \*ALL**

Alle Kommandos der angegebenen Syntaxdatei werden ausgegeben.

Bei der Ausgabe aus einer von Fujitsu Technology Solutions ausgelieferten SEDS (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 Technology Solutions ausgelieferte SEDS (Softwareeinheitssyntaxdatei) oder eine INSD (Installationssyntaxdatei) eingemischt wurden, angezeigt werden, muss die beim Einmischen verwendete REMOVE-ID als Produktname angegeben werden.



- (1) SDF-I wird gestartet.
- (2) Für die Systemsyntaxdatei des BS2000-Grundaubaus werden Informationen zu allen enthaltenen Kommandos und deren Privilegierung angefordert. Im Beispiel wird die Ausgabe nur ausschnittsweise wiedergegeben.
- (3) Die Syntaxdatei hat das aktuelle Format V4.1. Nur bei älteren Syntaxdatei wird ggf. ein anderes Format (V1, V2, V3, V4) angezeigt werden.
- (4) Unter anderem wird für das Kommando SHOW-FILE die Privilegierung für das Kommando (ALL-Zeile) und für mögliche Eingabemodi ausgegeben. Die Eingabemodi entsprechen den Modi, die mit der Anweisung ADD-CMD (siehe Handbuch „SDF-A“ [4]) bei der Kommando-Definition festgelegt wurden:

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

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

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

---

# 11 SDF-U

**Versionsstand:** SDF-U V4.1

**Privilegierung:** TSOS

SDF-U ist ein Programm, mit dem die Systemverwaltung Gruppen- und Systemsyntaxdateien gemäß den Belangen des Rechenzentrums modifizieren kann. Es ist vorrangig für die Installation neuer Syntaxdateiversionen gedacht und bedient sich dabei einer Untermenge an Funktionen und Anweisungen des Softwareprodukts SDF-A (siehe Handbuch „SDF-A“ [4]).

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

## Starten von SDF-U

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

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

### VERSION =

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

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

### VERSION = \*STD

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

### VERSION = <product-version>

Angabe der SDF-U-Version im Format mm.n[a[so]].

**MONJV =**

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

**MONJV = \*NONE**

Es wird keine Monitor-Jobvariable verwendet.

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

Name der zu verwendenden Jobvariablen.

**CPU-LIMIT =**

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

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

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

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

Es soll nur die angegebene Zeit verwendet werden.

**SDF-Standardanweisungen**

Die SDF-Standardanweisungen (siehe [Seite 191](#)) sind nicht in der SDF-U-Syntaxdatei, sondern in der Syntaxdatei von SDF definiert. Sie werden systemweit für Benutzerprogramme angeboten, die ihre Anweisungen über SDF einlesen.

**Unterbrechen und Fortsetzen von SDF-U**

Wird SDF-U während der Bearbeitung einer Anweisung mit K2 unterbrochen, so kann es auf zwei Arten fortgesetzt werden.

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

## 11.1 SDF-U-Anweisungen

### 11.1.1 Funktionelle Übersicht

#### SDF-Standardanweisungen

Die ausführliche Beschreibung der SDF-Standardanweisungen finden Sie im [Abschnitt „Standardanweisungen“ auf Seite 148](#).

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

**Syntaxdateien bearbeiten und erstellen**

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

**Anzeigefunktion**

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



## 11.1.2 Beschreibung der Anweisungen

### COPY

#### Inhalte einer Syntaxdatei kopieren

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

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

**OBJECT =**

Art des Objekts, dessen Definition zu kopieren ist.

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

Bestimmt, dass die Definitionen von Anwendungsbereichen kopiert werden.

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

Die Definitionen von allen Anwendungsbereichen werden kopiert.

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

Die Definitionen der hier angegebenen Anwendungsbereiche werden nicht kopiert.

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

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

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

Bestimmt, dass die Definitionen von Kommandos kopiert werden.

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

Die Definitionen von allen Kommandos werden kopiert.

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

Die Definitionen der hier angegebenen Kommandos werden nicht kopiert.

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

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

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

Bestimmt, dass die Definitionen von Programmen kopiert werden.

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

Die Definitionen von allen Programmen werden kopiert.

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

Die Definitionen der hier angegebenen Programme werden nicht kopiert.

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

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

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

Syntaxdatei, die die zu kopierenden Definitionen enthält.

**ATTACHED-INFO =**

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

**ATTACHED-INFO = \*YES**

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

**ATTACHED-INFO = \*NO**

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

**ATTACHED-INFO = \*ONLY**

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

**OVERWRITE-POSSIBLE =**

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

**OVERWRITE-POSSIBLE = \*NO**

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

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

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

**OVERWRITE-POSSIBLE = \*YES**

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

## EDIT

### Auf ein Kommando der Syntaxdatei positionieren

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

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

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

Ein Kommando wird aktuelles Objekt.

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

Name des Kommandos.

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

Ein Operandenwert wird aktuelles Objekt.

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

Bestimmt den Operanden, zu dem der zum aktuellen Objekt werdende Operandenwert gehört. \*ABOVE-CURRENT bedeutet, dass ein zu OPERAND-L1 gehörender Wert aktuelles Objekt ist. <structured-name 1..20> muss ein kommandoglobal eindeutiger Operandenname sein.

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

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

**ORIGIN =**

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

**ORIGIN = \*CURRENT**

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

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

Der Operandenwert gehört zu dem bei NAME angegebenen Kommando.

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

Name des Kommandos.

## **END**

### **Programmlauf beenden**

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

<b>END</b>

Diese Anweisung hat keine Operanden.

## MODIFY-CMD

### Kommandodefinition ändern

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

#### MODIFY-CMD

```
IMPLEMENTOR = *UNCHANGED / *PROCEDURE(...)  
  *PROCEDURE(...)  
    | NAME = *UNCHANGED / <c-string 1..54>
```

#### IMPLEMENTOR =

Art der Kommandoimplementierung.

#### IMPLEMENTOR = \*UNCHANGED

Keine Änderung bezüglich der Kommandoimplementierung.

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

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

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

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

## MODIFY-VALUE

### Operandenwertdefinition ändern

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

#### MODIFY-VALUE

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

#### VALUE =

Bestimmt, welche Werte als Eingabe zulässig sind.

#### VALUE = \*UNCHANGED

Keine Änderung bezüglich der zulässigen Eingabewerte.

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

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

#### OUTPUT =

Bestimmt, welcher Wert an die Implementierung übergeben wird.

#### OUTPUT = \*UNCHANGED

Keine Änderung bezüglich des zu übergebenden Werts.

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

Der hier angegebene Wert wird übergeben.



## OPEN-SYNTAX-FILE

### Syntaxdatei öffnen

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

#### OPEN-SYNTAX-FILE

```

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

```

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

Name der zu öffnenden Syntaxdatei.

#### **TYPE =**

Art der zu öffnenden Syntaxdatei.

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

Eine Gruppensyntaxdatei ist zu öffnen.

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

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

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

Eine Systemsyntaxdatei ist zu öffnen.

#### **MODE =**

Legt fest, wie die geöffnete Syntaxdatei bearbeitet werden darf.

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

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

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

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

## REMOVE

### Objekte der Syntaxdatei löschen

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

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

#### OBJECT =

Art des Objekts, das zu löschen ist.

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

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

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

Alle Anwendungsbereiche werden gelöscht.

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

Die hier angegebenen Anwendungsbereiche werden nicht gelöscht.

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

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

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

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

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

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

Alle Kommandos werden gelöscht.

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

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

Die hier angegebenen Kommandos werden nicht gelöscht.

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

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

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

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

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

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

Alle Programme werden gelöscht.

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

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

Die hier angegebenen Programme werden nicht gelöscht.

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

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

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

## SET-GLOBALS

### Globalinformation ändern

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

#### SET-GLOBALS

```

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

```

#### **VERSION =**

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

#### **VERSION = \*UNCHANGED**

Die Versionsnummer bleibt unverändert.

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

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

#### **GUIDANCE =**

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

#### **GUIDANCE = \*UNCHANGED**

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

#### **GUIDANCE = \*STD**

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

**GUIDANCE = \*MAXIMUM**

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

**GUIDANCE = \*MEDIUM**

Zur Auswahl von Anwendungsbereichen, Kommandos und Anweisungen werden Menüs mit Erläuterungstexten ausgegeben. Zur Operandenbesetzung werden Fragebögen ausgegeben. Für Strukturen, deren einleitender Wert mit SIZE=\*LARGE (siehe Anweisung ADD-VALUE im Handbuch „SDF-A“ [4]) definiert ist, gibt es eigene Fragebögen. Die Fragebögen enthalten die Default-Werte und alle erlaubten Operandenwerte.

**GUIDANCE = \*MINIMUM**

Zur Auswahl von Anwendungsbereichen, Kommandos und Anweisungen werden Menüs ausgegeben. Zur Operandenbesetzung werden Fragebögen ausgegeben. Für Strukturen, deren einleitender Wert mit SIZE=\*LARGE (siehe Anweisung ADD-VALUE im Handbuch „SDF-A“ [4]) definiert ist, gibt es eigene Fragebögen. Die Fragebögen enthalten nur die Default-Werte.

**GUIDANCE = \*NO**

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

**GUIDANCE = \*EXPERT**

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

**LOGGING =**

Bestimmt, wie die Eingaben protokolliert werden.

**LOGGING = \*UNCHANGED**

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

**LOGGING = \*STD**

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

**LOGGING = \*INPUT-FORM**

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

**LOGGING = \*ACCEPTED-FORM**

Protokolliert werden:

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

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

**LOGGING = \*INVARIANT-FORM**

Protokolliert werden:

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

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

**PROCEDURE-DIALOGUE =**

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

**PROCEDURE-DIALOGUE = \*UNCHANGED**

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

**PROCEDURE-DIALOGUE = \*STD**

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

**PROCEDURE-DIALOGUE = \*YES**

Der Benutzer wird zur interaktiven Korrektur aufgefordert.

**PROCEDURE-DIALOGUE = \*NO**

Der Benutzer wird zur interaktiven Korrektur nicht aufgefordert.

**UTILITY-INTERFACE =**

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

**UTILITY-INTERFACE = \*UNCHANGED**

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

**UTILITY-INTERFACE = \*STD**

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

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

Programme sollen ihre Anweisungen mit RDATA lesen.

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

Programme sollen ihre Anweisungen über SDF mit RDSTMT lesen.

**CONTINUATION =**

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

**CONTINUATION = \*UNCHANGED**

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

**CONTINUATION = \*STD**

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

**CONTINUATION = \*OLD-MODE**

Das Fortsetzungszeichen muss in Spalte 72 angegeben werden.

**CONTINUATION = \*NEW-MODE**

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

**FUNCTION-KEYS =**

Bestimmt die Funktionstastenbelegung. Die verschiedenen Modi sind ausführlich im [Abschnitt „Funktionstasten“ auf Seite 36](#) beschrieben. Nicht unterstützte Funktionstasten wirken nicht wie  , sondern bewirken die leere Leistung.

**FUNCTION-KEYS = \*UNCHANGED**

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

**FUNCTION-KEYS = \*STD**

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

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

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

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

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

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

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

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

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



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



**INPUT-HISTORY =**

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

**INPUT-HISTORY = \*UNCHANGED**

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

**INPUT-HISTORY = \*STD**

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

**INPUT-HISTORY = \*ON**

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

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

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



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

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

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

**INPUT-HISTORY = \*OFF**

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

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

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

## SHOW Objekte der Syntaxdatei anzeigen

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

(Teil 1 von 2)

SHOW
<p><b>OBJECT = *GLOBAL-INFORMATION / *DOMAIN(...) / *COMMAND(...) / *PROGRAM(...)</b></p> <p><b>*DOMAIN(...)</b></p> <ul style="list-style-type: none"> <li><b>NAME = *ALL(...)</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> <li><b>*ALL(...)</b> <ul style="list-style-type: none"> <li><b>EXCEPT = *NONE</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> </ul> </li> </ul> <p><b>*COMMAND(...)</b></p> <ul style="list-style-type: none"> <li><b>NAME = *ALL(...)</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> <li><b>*ALL(...)</b> <ul style="list-style-type: none"> <li><b>EXCEPT = *NONE</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> </ul> </li> </ul> <p><b>*PROGRAM(...)</b></p> <ul style="list-style-type: none"> <li><b>NAME = *ALL(...)</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> <li><b>*ALL(...)</b> <ul style="list-style-type: none"> <li><b>EXCEPT = *NONE</b> / &lt;structured-name 1..30 with-wild&gt; / list-poss(2000): &lt;structured-name 1..30&gt;</li> </ul> </li> </ul> <p><b>,ATTACHED-INFORMATION = *YES / *NO / *IMMEDIATE</b></p> <p><b>,SIZE = *MINIMUM / *MAXIMUM / *MEDIUM</b></p> <p><b>,IMPLEMENTATION-INFO = *NO (...)</b></p> <p><b>*NO(...)</b></p> <ul style="list-style-type: none"> <li><b>FORM = *UNGUIDED / *GUIDED</b></li> <li><b>,LANGUAGE = E</b> / &lt;name 1..1&gt;</li> </ul>

Fortsetzung ➔

(Teil 2 von 2)

```

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

```

**OBJECT =**

Art des Objekts, dessen Definition auszugeben ist.

**OBJECT = \*GLOBAL-INFORMATION**

Bestimmt, dass die Globalinformation der Syntaxdatei ausgegeben wird.

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

Bestimmt, dass die Definitionen von Anwendungsbereichen ausgegeben werden.

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

Die Definitionen von allen Anwendungsbereichen werden ausgegeben.

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

Die Definitionen der hier angegebenen Anwendungsbereiche werden nicht ausgegeben.

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

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

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

Bestimmt, dass die Definitionen von Kommandos ausgegeben werden.

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

Die Definitionen von allen Kommandos werden ausgegeben.

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

Die Definitionen der hier angegebenen Kommandos werden nicht ausgegeben.

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

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

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

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

Bestimmt, dass die Definitionen von Programmen ausgegeben werden.

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

Die Definitionen von allen Programmen werden ausgegeben.

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

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

Die Definitionen der hier angegebenen Programme werden nicht ausgegeben.

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

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

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

**ATTACHED-INFORMATION =**

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

**ATTACHED-INFORMATION = \*YES**

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

**ATTACHED-INFORMATION = \*NO**

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

**ATTACHED-INFORMATION = \*IMMEDIATE**

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

**SIZE =**

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

**SIZE = \*MINIMUM**

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

**SIZE = \*MAXIMUM**

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

**SIZE = \*MEDIUM**

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

**IMPLEMENTATION-INFO =**

Bestimmt, wie die Ausgabe aufbereitet wird.

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

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

**FORM =**

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

**FORM = \*UNGUIDED**

Die Definitionen werden ausgegeben.

**FORM = \*GUIDED**

Die Definitionen werden nicht ausgegeben.

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

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

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

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

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

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

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

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

**OUTPUT-FORM =**

Legt fest, welche Zeichen am Zeilenanfang ausgegeben werden.

**OUTPUT-FORM = \*STD**

Das erste Zeichen jeder Zeile ist ein Leerzeichen.

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

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

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

**OUTPUT =**

Bestimmt das Ausgabemedium für die gewünschten Informationen.

**OUTPUT = \*SYSOUT**

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

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

Die Ausgabe erfolgt in die logische Systemdatei SYSLST.

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

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

**LINE-LENGTH = \*STD / <integer 72..132>**

Bestimmt die Zeilenlänge für die Ausgabe.

**LINE-LENGTH = \*STD**

Der Default-Wert beträgt 74 Zeichen bei Ausgabe auf den Bildschirm und 72 Zeichen bei Ausgabe in eine Datei.

**PRIVILEGE = \*ANY / list-poss(64): <structured-name 1..30>**

Es werden nur die Objekte ausgegeben, denen mindestens eines der in der Liste genannten Privilegien zugeordnet ist. Bei Angabe von \*ANY werden die Objekte ohne Beachtung ihrer Privilegien ausgegeben.

## SHOW-CORRECTION-INFORMATION

### Korrekturinformation der Syntaxdatei ausgeben

Die Anweisung SHOW-CORRECTION-INFORMATION gibt Informationen über in der Syntaxdatei enthaltene Korrekturen aus.

Die Anweisung ist nur für Diagnosezwecke vorgesehen.

SHOW-CORRECTION-INFORMATION
<p><b>CORRECTION-ID</b> = <b>*SOURCE</b> (...) / <b>*OBJECT</b>(...)</p> <p><b>*SOURCE</b>(...)</p> <ul style="list-style-type: none"> <li>  <b>PM-NUMBER</b> = <b>*ALL</b> / list-poss(100): &lt;alphanum-name 8..8&gt;</li> </ul> <p><b>*OBJECT</b>(...)</p> <ul style="list-style-type: none"> <li>  <b>PM-NUMBER</b> = <b>*ALL</b> / &lt;alphanum-name 8..8&gt;</li> <li>  <b>JULIAN-DATE</b> = <b>*ANY</b> / &lt;integer 1..366&gt;</li> </ul> <p><b>,PRODUCT-NAME</b> = <b>*ANY</b> / &lt;structured-name 1..15&gt;(…)</p> <p>&lt;structured-name 1..15&gt;(…)</p> <ul style="list-style-type: none"> <li>  <b>VERSION</b> = <b>*ANY</b> / &lt;product-version&gt;</li> </ul>

Die Operandenbeschreibung entfällt an dieser Stelle, da die Anweisung nur für Diagnosezwecke vorgesehen ist.

## SHOW-STATUS

### Status der geöffneten Syntaxdatei anzeigen

Mit der Anweisung SHOW-STATUS veranlassen Sie, dass SDF-U den Namen der gerade geöffneten Syntaxdatei und Informationen über eine oder mehrere zugeordnete Referenzsyntaxdateien ausgibt.

SHOW-STATUS

Diese Anweisung hat keine Operanden.



## STEP

### Definieren eines Wiederaufsetzpunktes

Mit der Anweisung STEP definieren Sie in einer Folge von SDF-U-Anweisungen einen Wiederaufsetzpunkt für die Fehlerbehandlung. Die Eingabe der Anweisung STEP ist nur in Prozeduren und Stapelaufträgen zulässig.

STEP

Diese Anweisung hat keine Operanden.

Trifft SDF-U auf einen syntaktischen oder schwerwiegenden logischen Fehler, dann leitet es folgende Schritte ein:

- es wird eine Fehlermeldung ausgegeben,
- die laufende SDF-U-Anweisung wird abgebrochen,
- die nachfolgenden Anweisungen werden bis zum Erreichen von STEP oder END überlesen.

Erreicht SDF-U als Nächstes ein END, so erzeugt es ein abnormales Programmende (TERM UNIT=STEP,MODE=ABNORMAL) und aktiviert den Spin-Off-Mechanismus. Erreicht SDF-U als Nächstes ein STEP, so fährt es mit der auf STEP folgenden Anweisung fort. Falls ein Fehler keinen Einfluss auf den ordnungsgemäßen Ablauf des laufenden Jobs hat, so muss der Benutzer von SDF-U den abnormalen Programmabbruch durch Einfügen eines STEP abfangen.

Geringfügige logische Fehler korrigiert SDF-U selbstständig unter Ausgabe einer Warnmeldung. Es wird kein Spin-Off-Mechanismus aktiviert.



---

## 12 SDF-PAR

**Versionsstand:** SDF-PAR V1.1

**Privilegierung:** TSOS

SDF-PAR ist ein Dienstprogramm, mit dem Sie eine Parameterdatei für das Produkt SDF erzeugen, modifizieren und anzeigen können. Es bietet somit eine Alternative zu den privilegierten Kommandos MODIFY-SDF-PARAMETERS und SHOW-SDF-PARAMETERS, die im Handbuch „Kommandos“ [1] beschrieben sind. Ein Vorteil von SDF-PAR besteht darin, dass Sie die Parameterdatei off-line bearbeiten können.

Das Programm SDF-PAR V1.1 kann ab SDF V1.4 eingesetzt werden. Die Anweisungen an SDF-PAR sind in einer Benutzersyntaxdatei definiert, die mit SDF-PAR ausgeliefert wird.

### Starten von SDF-PAR

Für den Start von SDF-PAR in BS2000/OSD ist folgende, im Lieferumfang enthaltene Prozedur aufzurufen:

```
/CALL-PROCEDURE $.SYSPRC.SDF-PAR.011.112
```

Die Prozedur enthält neben dem eigentlichen Aufruf von SDF-PAR auch die notwendigen Vorbereitungen für den Programmaufruf (Meldungs- und Syntaxdatei von SDF-PAR werden aktiviert).



Vor dem Starten von SDF-PAR muss die SDF-PAR-Meldungsdatei zugewiesen und die SDF-PAR-Benutzersyntaxdatei aktiviert werden. Alle notwendigen Vorbereitungen für den Programmlauf werden auch von der mitgelieferten SYSPRC-Prozedur ausgeführt.

Da für SDF-PAR kein START-Kommando existiert, muss die SYSPRG-Datei mit dem Kommando START-EXECUTABLE-PROGRAM (bzw. START-PROGRAM) gestartet werden.

Weitere Hinweise zur Installation von SDF-PAR finden Sie in der Freigabemittelung zum Softwareprodukt SDF.

## 12.1 SDF-PAR-Anweisungen

### 12.1.1 Funktionelle Übersicht

#### SDF-Parameterdatei erzeugen, bearbeiten und anzeigen

END	SDF-PAR beenden
OPEN-PARAMETER-FILE	SDF-Parameterdatei erzeugen oder zur Bearbeitung öffnen
MODIFY-SYNTAX-FILE	Einträge für Syntaxdateien ändern
MODIFY-SYSTEM-LOGON-INCLUDE	Eintrag für die systemweite LOGON-Include-Prozedur ändern
MODIFY-SYSTEM-LOGON-PROCEDURE	Eintrag für die systemweite LOGON-Call-Prozedur ändern
MODIFY-VERSION	Format der Parameterdatei ändern
SHOW-PARAMETER-FILE	Inhalt der geöffneten SDF-Parameterdatei anzeigen

#### SDF-Standardanweisungen

Die Beschreibung dieser Anweisungen finden Sie im [Abschnitt „Standardanweisungen“ auf Seite 148](#).

END	SDF-PAR beenden
EXECUTE-SYSTEM-CMD	Systemkommando während des SDF-PAR-Laufs ausführen
HELP-MSG-INFORMATION	Text einer Systemmeldung auf SYSOUT ausgeben
HOLD-PROGRAM	SDF-PAR-Lauf anhalten
MODIFY-SDF-OPTIONS	Benutzersyntaxdatei aktivieren oder deaktivieren und SDF-Einstellungen ändern
REMARK	Kommentare in die Anweisungsfolge schreiben und die Ausführung eines Programms dokumentieren
RESET-INPUT-DEFAULTS	Taskspezifische Default-Werte zurücksetzen
RESTORE-SDF-INPUT	Vorher eingegebene Anweisungen oder Kommandos am Bildschirm wiederanzeigen
SHOW-INPUT-DEFAULTS	Taskspezifische Default-Werte anzeigen
SHOW-INPUT-HISTORY	Inhalt des Eingabepuffers anzeigen

---

SHOW-SDF-OPTIONS	Taskspezifische Information über aktivierte Syntaxdateien und die Festlegungen für die Anweisungseingabe und -verarbeitung anzeigen
SHOW-STMT	Syntax einer Anweisung anzeigen
STEP	Wiederaufsetzpunkt in einer Folge von SDF-PAR-Anweisungen definieren
WRITE-TEXT	Einen bestimmten Text nach SYSOUT oder SYSLST ausgeben

## 12.1.2 Beschreibung der Anweisungen

### MODIFY-SYNTAX-FILE

#### Einträge für Syntaxdateien ändern

Mit der Anweisung MODIFY-SYNTAX-FILE können Sie Syntaxdatei-Namen in die Parameterdatei eintragen oder bestehende Syntaxdatei-Einträge ändern oder entfernen. Folgende Einträge sind möglich:

- **Systemsyntaxdatei:**  
Ein Eintrag zur Basis-Systemsyntaxdatei muss in jeder Parameterdatei vorhanden sein. Der Eintrag kann geändert, aber nicht entfernt werden. Die Basis-Systemsyntaxdatei enthält die Syntaxbeschreibung aller Systemkommandos und gilt für alle aktuell angemeldeten Benutzeraufträge.
- **Subsystem-Syntaxdateien:**  
Einträge zu Subsystem-Syntaxdateien sind optional. Subsystem-Syntaxdateien sind Systemsyntaxdateien, die zusätzlich zur Basis-Systemsyntaxdatei aktiviert werden. Eine solche Systemsyntaxdatei muss nicht unbedingt zu einem von DSSM aktivierten Subsystem gehören.
- **Gruppensyntaxdateien:**  
Einträge zu Gruppensyntaxdateien sind optional. Eine Gruppensyntaxdatei enthält die Syntaxbeschreibung aller Kommandos, Programme und Anweisungen, die nur für eine von der Systemverwaltung definierte Gruppe von Benutzern verfügbar sind. Die Gruppensyntaxdateien werden über eine PROFILE-ID zugewiesen.

#### MODIFY-SYNTAX-FILE

```

TYPE = *SYSTEM / *SUBSYSTEM(...) / *GROUP(...)
  *SUBSYSTEM(...)
    | SUBSYSTEM-NAME = <structured-name 1..8>
  *GROUP(...)
    | PROFILE-ID = <structured-name 1..30>
    | ,HIERARCHY = *YES / *NO
,NAME = *UNCHANGED / *NONE / <filename 1..54 without-gen-vers>

```

**TYPE =**

Legt fest, für welchen Syntaxdatei-Typ der Eintrag geändert wird.

**TYPE = \*SYSTEM**

Der Eintrag für die Basis-Systemsyntaxdatei wird geändert.

**TYPE = \*SUBSYSTEM(...)**

Der Eintrag für eine Subsystem-Syntaxdatei wird geändert.

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

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

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

Der Eintrag für eine Gruppensyntaxdatei wird geändert.

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

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

**HIERARCHY =**

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

**HIERARCHY = \*YES**

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

**HIERARCHY = \*NO**

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

**NAME =**

Bestimmt den Syntaxdatei-Namen. Die angegebene Syntaxdatei muss von dem bei TYPE festgelegten Typ sein.

**NAME = \*UNCHANGED**

Der Eintrag wird nicht verändert.

**NAME = \*NONE**

Der Syntaxdatei-Eintrag wird aus der Parameterdatei gelöscht. Der Eintrag für die Systemsyntaxdatei kann nicht gelöscht werden.

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

Bestimmt den Syntaxdatei-Namen.

## MODIFY-SYSTEM-LOGON-PROCEDURE

### Eintrag für systemweite LOGON-Call-Prozedur ändern

Mit der Anweisung MODIFY-SYSTEM-LOGON-PROCEDURE können Sie den Namen einer systemweiten LOGON-Call-Prozedur in die geöffnete Parameterdatei eintragen. Diese Prozedur wird bei jedem LOGON-Vorgang mit dem Kommando CALL-PROCEDURE aufgerufen und läuft automatisch ab.

<b>MODIFY-SYSTEM-LOGON-PROCEDURE</b>
<b>NAME = <u>*UNCHANGED</u> / *NONE / *STD / &lt;filename 1..54 without-gen-vers&gt;</b>

**NAME = \*UNCHANGED / \*NONE / \*STD / <filename 1..54 without-gen-vers>**

Gibt an, ob eine systemweite LOGON-Call-Prozedur in die Parameterdatei eingetragen wird. Dieser Eintrag ist optional.

**NAME = \*NONE**

Der Eintrag für eine systemweite LOGON-Call-Prozedur wird aus der Parameterdatei gelöscht.

**NAME = \*STD**

Der Standardname \$.SYS.SDF.LOGON.SYSPROC wird eingetragen.

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

Name der neuen systemweiten LOGON-Call-Prozedur.



## MODIFY-SYSTEM-LOGON-INCLUDE

### Eintrag für systemweite LOGON-Include-Prozedur ändern

Mit der Anweisung MODIFY-SYSTEM-LOGON-INCLUDE können Sie den Namen einer systemweiten LOGON-Include-Prozedur in die geöffnete Parameterdatei eintragen. Diese Prozedur wird bei jedem LOGON-Vorgang mit dem Kommando INCLUDE-PROCEDURE aufgerufen und läuft automatisch ab.

<b>MODIFY-SYSTEM-LOGON-INCLUDE</b>
<b>NAME = <u>*UNCHANGED</u> / *NONE / *STD / &lt;filename 1..54 without-gen-vers&gt;</b>

**NAME = \*UNCHANGED / \*NONE / \*STD / <filename 1..54 without-gen-vers>**

Gibt an, ob eine systemweite LOGON-Include-Prozedur in die Parameterdatei eingetragen wird. Dieser Eintrag ist optional.

**NAME = \*NONE**

Der Eintrag für eine systemweite LOGON-Include-Prozedur wird aus der Parameterdatei gelöscht.

**NAME = \*STD**

Der Standardname \$.SYS.SDF.LOGON.SYSINCL wird eingetragen.

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

Name der neuen systemweiten LOGON-Include-Prozedur.

## MODIFY-VERSION

### Format der Parameterdatei ändern

Mit der Anweisung MODIFY-VERSION können Sie das Format der geöffneten Parameterdatei ändern. Das Format einer Parameterdatei wird beim Erzeugen mit der Anweisung OPEN-PARAMETER-FILE festgelegt.



Das Format \*V2 ist das aktuell unterstützte Format für Parameterdateien. Parameterdateien im Format \*V1 sind in den aktuellen SDF-Versionen nicht einsetzbar. Deshalb ist nur eine Konvertierung in das Format \*V2 sinnvoll.

<b>MODIFY-VERSION</b>
<b>VERSION = *<u>UNCHANGED</u> / *V2 / *V1</b>

#### **VERSION = \*UNCHANGED / \*V2 / \*V1**

Legt das Format der Parameterdatei fest.

#### **VERSION = \*V2**

Die Parameterdatei wird in das aktuell unterstützte \*V2-Format konvertiert.

Die Satzlänge der Parameterdatei wird bei der Konvertierung von \*V1-Format nach \*V2-Format auf 67 Byte verlängert. Die zusätzlichen Bytes werden mit X'00' aufgefüllt. Die Sätze für die Gruppensyntaxdatei-Einträge werden mit 'YES' und 10mal X'00' erweitert. Jedes Objekt der Parameterdatei wird rekonstruiert und wieder in die geöffnete Parameterdatei eingefügt. Für jedes Objekt wird eine Meldung ausgegeben.

#### **VERSION = \*V1**

*Der Operandenwert wird nur aus Kompatibilitätsgründen unterstützt.*

Parameterdateien im \*V1-Format sind in den aktuellen SDF-Versionen nicht einsetzbar (Konvertierung verkürzt die Sätze auf 54 Byte).

## OPEN-PARAMETER-FILE

### Parameterdatei erzeugen oder öffnen

Mit der Anweisung OPEN-PARAMETER-FILE öffnen Sie eine bereits vorhandene oder erzeugen eine neue Parameterdatei. Beim Erzeugen der Parameterdatei legen Sie auch das Format fest.



Das Format \*V2 ist das aktuell unterstützte Format für Parameterdateien. Parameterdateien im Format \*V1 sind in den aktuellen SDF-Versionen nicht einsetzbar. Deshalb ist beim Erzeugen einer Datei nur das Format \*V2 sinnvoll.

Beim Erzeugen der Parameterdatei können auch schon Syntaxdateinamen eingetragen werden. Standardmäßig wird eine Parameterdatei im \*V2-Format erzeugt, in der als Name für die Basis-Systemsyntaxdatei \$.SYS.SDF.SYSTEM.SYNTAX eingetragen ist.

#### OPEN-PARAMETER-FILE

```

NAME = <filename 1..54 without-gen-vers>
,MODE = *READ / *UPDATE / *CREATE(...)
  *CREATE(...)
    |
    |   VERSION = *V2 / *V1
    |   ,SYSTEM-SYNTAX-FILE = *STD / <filename 1..54 without-gen-vers>
    |   ,GROUP-SYNTAX-FILE = *NONE / *STD(...) / <filename 1..54 without-gen-vers>(…)
    |   *STD(...)
    |     |
    |     |   PROFILE-ID = SYS-TSOS / <structured-name 1..30>
    |     |   ,HIERARCHY = *YES / *NO
    |     |
    |     |   <filename 1..54 without-gen-vers>(…)
    |     |   PROFILE-ID = SYS-TSOS / <structured-name 1..30>
    |     |   ,HIERARCHY = *YES / *NO
  
```

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

Name der zu öffnenden oder zu erzeugenden Parameterdatei.

**MODE =**

Legt fest, ob die Parameterdatei nur zum Lesen oder zum Ändern geöffnet wird oder ob sie erzeugt werden soll.

**MODE = \*READ**

Die bereits vorhandene Parameterdatei wird nur zum Lesen geöffnet.

**MODE = \*UPDATE**

Die bereits vorhandene Parameterdatei wird im Lese-/Schreibmodus geöffnet.

**MODE = \*CREATE(...)**

Die Parameterdatei wird erzeugt und im Lese-/Schreibmodus geöffnet. Existiert eine Parameterdatei mit dem angegebenen Namen bereits, so wird die Anweisung abgewiesen.

**VERSION =**

Legt das Format der Parameterdatei fest.

**VERSION = \*V2**

Die Parameterdatei wird im aktuell unterstützten \*V2-Format erzeugt.

**VERSION = \*V1**

*Der Operandenwert wird nur aus Kompatibilitätgründen unterstützt.*

Parameterdateien im \*V1-Format sind in den aktuellen SDF-Versionen nicht einsetzbar.

**SYSTEM-SYNTAX-FILE =**

Legt den Eintrag für die Basis-Systemsyntaxdatei fest.

**SYSTEM-SYNTAX-FILE = \*STD**

Als Name für die Basis-Systemsyntaxdatei wird \$.SYS.SDF.SYSTEM.SYNTAX in die Parameterdatei eingetragen.

**SYSTEM-SYNTAX-FILE = <filename 1..54 without-gen-vers>**

Name der Basis-Systemsyntaxdatei, der in die Parameterdatei eingetragen wird.

**GROUP-SYNTAX-FILE =**

Der Gruppensyntaxdatei-Eintrag für die Default-User-ID wird festgelegt.

**GROUP-SYNTAX-FILE = \*NONE**

Für die Default-User-ID wird keine Gruppensyntaxdatei eingetragen.

**GROUP-SYNTAX-FILE = \*STD**

Für die Default-User-ID wird die Gruppensyntaxdatei \$.SYS.SDF.GROUP.SYNTAX eingetragen.

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

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet wird. Default-Wert ist SYS-TSOS, da SDF für die User-ID TSOS auch diesen Wert verwendet.

**HIERARCHY =**

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

**HIERARCHY = \*YES**

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

**HIERARCHY = \*NO**

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

**GROUP-SYNTAX-FILE = <filename 1..54 without-gen-vers>**

Für die Default-User-ID wird die Gruppensyntaxdatei mit dem angegebenen Namen eingetragen.

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

Bestimmt die PROFILE-ID, zu der die Gruppensyntaxdatei zugeordnet wird. Default-Wert ist SYS-TSOS, da SDF für die User-ID TSOS auch diesen Wert verwendet.

**HIERARCHY =**

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

**HIERARCHY = \*YES**

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

**HIERARCHY = \*NO**

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

## SHOW-PARAMETER-FILE

### Parameterdatei anzeigen

Die Anweisung SHOW-PARAMETER-FILE zeigt den Inhalt der aktuell geöffneten Parameterdatei an. Folgende Einträge werden ausgegeben, sofern sie vorhanden sind:

- Versionsformat der Parameterdatei
- Name der eingetragenen systemweiten LOGON-Prozedur
- Name der eingetragenen Basis-Systemsyntaxdatei
- zu jedem eingetragenen Subsystem wird der Name der Subsystem-Syntaxdatei ausgegeben
- zu jeder eingetragenen PROFILE-ID wird der Name der zugeordneten Gruppensyntaxdatei ausgegeben. Für Parameterdateien im \*V2-Format wird auch die Hierarchie-Einstellung ausgegeben.

<b>SHOW-PARAMETER-FILE</b>

Diese Anweisung hat keine Operanden.

---

## 13 SDF-CONV

**Versionsstand:** SDF-CONV V3.0B

**Privilegierung:** STD-PROCESSING, TSOS

Mit dem Dienstprogramm SDF-CONV ist es möglich, zwei Arten von Prozedurumwandlungen durchzuführen:

- die Umwandlung der Kommandosprache (von ISP- in SDF-Kommandosprache)
- die Umwandlung des Prozedurformats (von Nicht-S- in S-Prozedurformat)

SDF-CONV übersetzt in Dateien oder Bibliothekselementen stehende ISP-Kommandos und deren zugehörige Operanden in SDF-Kommandos mit den entsprechenden Operanden. Kommandos in SDF-Kommandosätzen werden in ihren vollen Kommandonamen expandiert.

Einige ISP-Kommandos können nicht in SDF-Kommandos übersetzt werden. Sie werden unverändert in die Ausgabeprozedur übernommen. Beispielsweise erfolgt keine Übersetzung von ISP-Kommandos, wenn die Auswahl des entsprechenden SDF-Kommandos vom Laufzeitverhalten oder von Prozedurparametern abhängig ist oder das ISP-Kommando nicht mehr vom BS2000 unterstützt wird.

Im [Abschnitt „Einschränkungen“ auf Seite 252](#) werden alle Kommandos und Operandenkonfigurationen genannt, die durch SDF-CONV nicht übersetzt werden können.

SDF-CONV kann Nicht-S-Prozeduren (ISP- und SDF-Format) in S-Prozeduren umwandeln.

Eine S-Prozedur ("strukturierte Prozedur") ist entsprechend den Regeln von SDF-P aufgebaut. SDF-P ist eine blockorientierte Programmiersprache, die die bisherige Kommandosprache des BS2000 dahingehend erweitert, dass (analog zu höheren Programmiersprachen) strukturiertes Programmieren möglich ist (siehe Handbuch "SDF-P" [5]).

Das Software-Produkt SDF-P besteht aus zwei Teilen:

- dem kostenfreien Subsystem SDF-P-BASYS mit den Grundbestandteilen von SDF-P
- dem kostenpflichtigen Subsystem SDF-P mit der vollständigen Funktionalität

Wandelt SDF-CONV eine Nicht-S-Prozedur in eine S-Prozedur um, so werden nur die kostenfreien Funktionalitäten von SDF-P-BASYS berücksichtigt (siehe Handbuch "Kommandos" [1]).

Die erstellte S-Prozedur ist kompatibel, aber nicht optimiert in Bezug auf die Kontrollstrukturmechanismen einer mit SDF-P erstellten S-Prozedur.

Folgende Umwandlungen der Kommandosprache und des Kommandoformats sind mit SDF-CONV möglich:

Nicht-S-Prozedur mit ISP-Kommandos	→ Nicht-S-Prozedur mit SDF-Kommandos
Nicht-S-Prozedur mit ISP-Kommandos	→ S-Prozedur mit ISP-Kommandos
Nicht-S-Prozedur mit ISP-Kommandos	→ S-Prozedur mit SDF-Kommandos
Nicht-S-Prozedur mit SDF-Kommandos	→ S-Prozedur mit SDF-Kommandos
S-Prozedur mit ISP-Kommandos	→ S-Prozedur mit SDF-Kommandos

SDF-CONV schreibt das Ergebnis der Umwandlung in eine Ausgabeprozedur, die als SAM-Datei oder Element einer PLAM-Bibliothek abgespeichert wird. Ein ausführliches Protokoll wird nach SYSLST ausgegeben.

## 13.1 Starten und Beenden des Programms

Gestartet wird das Dienstprogramm SDF-CONV durch das Kommando

<b>START-SDF-CONV</b>	Kurzname: <b>SDF-CONV</b>
<pre> <b>VERSION</b> = <b>*STD</b> / &lt;product-version&gt; <b>,MONJV</b> = <b>*NONE</b> / &lt;filename 1..54 without-gen-vers&gt; <b>,CPU-LIMIT</b> = <b>*JOB-REST</b> / &lt;integer 1..32767&gt; <b>,PROGRAM-MODE</b> = <b>*ANY</b> / 24 </pre>	

Mit diesem Kommando wird SDF-CONV als Subsystem aufgerufen.

Nach dem Programmstart erscheint die Meldung CVR0060.

Nach jedem Konvertierungslauf und der kurzen Fehlerübersicht (siehe [Abschnitt „Ausgaben nach SYSOUT“ auf Seite 250](#)) zeigt das Programm mit der Meldung CVR0060 seine erneute Bereitschaft an und wartet auf weitere Anweisungen. Dadurch ist es möglich, nacheinander mehrere Konvertierungsläufe zu veranlassen, ohne SDF-CONV jedes Mal neu zu starten.

Beendet wird SDF-CONV mit der Anweisung

```
//END
```



## 13.2 Anweisungen

### 13.2.1 Übersicht über alle Anweisungen

Das Dienstprogramm SDF-CONV akzeptiert die in folgender Tabelle aufgelisteten Anweisungen.

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

CONVERT ist die einzige programmspezifische Anweisung. Die anderen Anweisungen sind so genannte SDF-Standardanweisungen (siehe [Abschnitt „Standardanweisungen“ auf Seite 148](#)).

CONVERT ist die voreingestellte Anweisung im Experten-Modus, d.h., bei Eingabe von "?" erscheint statt des Anweisungsmenüs der Operandenfragebogen der CONVERT-Anweisung .

### 13.2.2 CONVERT - Anweisung

Die Anweisung CONVERT übersetzt in Dateien oder Bibliothekselementen stehende ISP-Kommandos in SDF-Kommandos.

```

CONVERT

FROM-FILE = <filename 1..54> / *LIBRARY-ELEMENT(...)

  *LIBRARY-ELEMENT(...)
    LIBRARY = <filename 1..54 without-gen-vers>
  ,ELEMENT = *ALL(...) / <composed-name 1..64>(…) /
    <filename 1..64 without-cat-gen-vers with-wild>(…)
    *ALL(...)
      | VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
      | <filename 1..24 without-cat-gen-vers with-wild>
    <composed-name>(…)
      | VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
      | <filename 1..24 without-cat-gen-vers with-wild>
    <filename>(…)
      | VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
      | <filename 1..24 without-cat-gen-vers with-wild>
    ,TYPE = J / <alphanum-name 1..8>

,EXPECT-CONTINUATION = *STD / *NEW-MODE / *OLD-MODE
,PARAMETER-LINES = *CONVERT / *COPY-ONLY
,TO-FILE = *STD / <filename 1..54> / *LIBRARY-ELEMENT(...)

  *LIBRARY-ELEMENT(...)
    LIBRARY = <filename 1..54 without-gen-vers>
  ,ELEMENT = *SAME(...) / <composed-name 1..64>(…)
    *SAME(...)
      | VERSION = *HIGHEST-EXISTING / *SAME / <composed-name 1..24>
    <composed-name>(…)
      | VERSION = *HIGHEST-EXISTING / *SAME / <composed-name 1..24>
    ,TYPE = *SAME / <alphanum-name 1..8>

```

Fortsetzung ➡

```

,PRODUCE-CONTINUATION = *STD / *NEW-MODE / *OLD-MODE
,REPLACE-OLD-FILE = *YES / *NO
,SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD / <filename 1..54>
,GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD / <filename 1..54> / *NO
,UNCHANGED-CMD = *NONE / list-poss(30): <structured-name 1..30>
,DOCUMENTATION = *NO / *MINIMUM / *MAXIMUM
,TARGET-VERSION = *CURRENT / *V10 / *V11 / *OSD-V1 / *OSD-V2
,OUTPUT-FORM = *STD / *ACCEPTED-FORM / *INVARIANT-FORM(...)
  *INVARIANT-FORM(...)
    | CONTEXT = *ALL / list-poss(3): *BATCH / *BATCH-PROC / *DIALOG-PROC
,PROCEDURE-FORMAT = *SAME (...) / *S-PROCEDURE(...)
  *SAME(...)
    | BLOCK-INPUT = *NOT-ALLOWED / *ALLOWED
  *S-PROCEDURE(...)
    | CMD-FORMAT = *SDF / *SAME
    ,PROGRAM-INPUT = *DATA(...) / *STMT(...)
      *DATA(...)
        | EXCEPT-AFTER-CMD = *NONE / list-poss(30): <text 0..30>
        ,WARNING = *NO / *YES
      *STMT(...)
        | EXCEPT-AFTER-CMD = *NONE / list-poss(30): <text 0..30>
    ,BLOCK-INPUT = *NOT-ALLOWED / *ALLOWED

```

**FROM-FILE =**

Name(n) der zu übersetzenden SAM- oder ISAM-Datei bzw. des Elements oder der Elemente aus einer PLAM- oder OSM-Bibliothek (Eingabeprozedur).

Es werden auch Dateien und Bibliothekselemente auf NK4-Platten unterstützt.

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

Name der zu übersetzenden SAM- oder ISAM-Datei.

**FROM-FILE = \*LIBRARY-ELEMENT(...)**

Namen der Bibliothek und deren Element(e).

Ist die Bibliothek eine OSM-Bibliothek, so müssen ihre Elemente vom Typ S sein.

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

Name der Bibliothek, die das zu übersetzende Element bzw. die zu übersetzenden Elemente enthält.

**ELEMENT =**

Gibt den/die Namen des Bibliothekselements bzw. der Bibliothekselemente an.

**ELEMENT = \*ALL(...)**

Alle Elemente der angegebenen Bibliothek sollen übersetzt werden. Die Angabe dieses Operandenwertes ist nur für Elemente einer PLAM-Bibliothek möglich.

**VERSION =**

Angabe der Elementversion.

**VERSION = \*HIGHEST-EXISTING**

Standardmäßig wird die höchste Version verwendet.

**VERSION = \*ALL**

Alle Versionen aller Elemente werden übersetzt. Die Angabe dieses Operandenwertes ist nur für Elemente einer PLAM-Bibliothek möglich.

**VERSION = <composed-name 1..24>**

Versionsnummer des Elements.

**VERSION = <filename 1..24 without-cat-gen-vers with-wild>**

Alle Versionen, die der angegebenen Musterzeichenfolge entsprechen, sollen übersetzt werden. Die Angabe dieses Operandenwertes ist nur mit einer SDF-Version  $\geq 3.0$  und nur für Elemente einer PLAM-Bibliothek möglich.

**ELEMENT = <composed-name 1..64>(...)**

Name des Bibliothekselements.

Ist die Bibliothek eine OSM-Bibliothek, sind die Namen der enthaltenen Elemente maximal 8 Zeichen lang. Werden für den Elementnamen mehr als 8 Zeichen eingegeben, so werden nur die ersten 8 Zeichen für die Auswahl des Elements herangezogen.

**VERSION =**

Für mögliche Angaben siehe Operand ELEMENT = \*ALL(...).

**ELEMENT = <filename 1..64 without-cat-gen-vers with-wild>(...)**

Alle Elemente der Bibliothek, die der angegebenen Musterzeichenfolge entsprechen, sollen übersetzt werden. Die Angabe dieses Operandenwertes ist nur mit einer SDF-Version  $\geq 3.0$  und nur für Elemente einer PLAM-Bibliothek möglich.

**VERSION =**

Für mögliche Angaben siehe Operand ELEMENT = \*ALL(...).

**TYPE = J / <alphanum-name 1..8>**

Bestimmt den Typ des Elements bzw. der Elemente der Eingabebibliothek.

Ist die Eingabebibliothek eine OSM-Bibliothek, muss TYPE=S angegeben werden.

**EXPECT-CONTINUATION =**

Stellung des Fortsetzungszeichens in der Eingabeprozedur.

**EXPECT-CONTINUATION = \*STD**

Die Stellung des Fortsetzungszeichens entspricht der aktuellen SDF-Einstellung.

Die aktuelle SDF-Einstellung kann mit der Anweisung SHOW-SDF-OPTIONS ausgegeben und mit der Anweisung MODIFY-SDF-OPTIONS CONTINUATION=... geändert werden.

**EXPECT-CONTINUATION = \*NEW-MODE**

Das Fortsetzungszeichen wird hinter dem letzten Zeichen der Zeile erwartet.

In S-Prozeduren können Kommandos und Anweisungen in Zeilen beliebiger Länge stehen.

Die Position des Fortsetzungszeichens variiert dementsprechend.

In Nicht-S-Prozeduren muss das Fortsetzungszeichen vor Spalte 73 stehen.

**EXPECT-CONTINUATION = \*OLD-MODE**

Das Fortsetzungszeichen wird genau in Spalte 72 erwartet.

**PARAMETER-LINES =**

Bestimmt, ob Kommandosätze, die Prozedurparameter enthalten oder in denen Jobvariablen ersetzt werden sollen, übersetzt werden.

**PARAMETER-LINES = \*CONVERT**

Kommandosätze, die Prozedurparameter enthalten oder in denen Jobvariablen ersetzt werden sollen, werden übersetzt. Treten während der Übersetzung Fehler auf, werden die betroffenen Kommandosätze unverändert in die Ausgabeprozedur übernommen.

**PARAMETER-LINES = \*COPY-ONLY**

Kommandosätze, die Prozedurparameter enthalten oder in denen Jobvariablen ersetzt werden sollen, werden unverändert in die Ausgabeprozedur übernommen. Diese Kommandosätze werden ungeachtet der eingestellten Dokumentationsart (Operand DOCUMENTATION) nicht kommentiert.

**TO-FILE =**

Name(n) der SAM-Datei bzw. des Elements oder der Elemente einer PLAM-Bibliothek, in die das Übersetzungsergebnis (Ausgabeprozedur) geschrieben werden soll.

**TO-FILE = \*STD**

Wird kein Dateiname bzw. Bibliothekselement angegeben, erzeugt SDF-CONV standardmäßig eine SAM-Ausgabedatei mit dem Namen SYSCVR.tsn.yyyy.mm.dd.hh.mm.ss .

Der Name besteht aus dem Präfix SYSCVR, der Auftragsnummer tsn und der aktuellen Datum-/Zeitangabe. Diese Ausgabedatei ist somit einmalig und wird durch später generierte Ausgabedateien (unabhängig vom Operanden REPLACE-OLD-FILE) nicht überschrieben. Im geführten Dialog wird der generierte Standarddateiname angezeigt.

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

Name der Ausgabedatei, in die das Übersetzungsergebnis (Ausgabeprozedur) geschrieben werden soll. Ist keine Datei mit diesem Namen katalogisiert, wird eine SAM-Datei mit diesem Namen erzeugt. Eine bereits katalogisierte gleichnamige Datei muss durch den Operanden REPLACE-OLD-FILE = \*YES zum Überschreiben freigegeben sein.

**TO-FILE = \*LIBRARY-ELEMENT(...)**

Namen der PLAM-Bibliothek und deren Element(e).

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

Name der PLAM-Bibliothek, die das übersetzte Element bzw. die übersetzten Elemente enthalten soll.

**ELEMENT =**

Gibt den/die Namen des Bibliothekselements bzw. der Bibliothekselemente an.

**ELEMENT = \*SAME(...)**

Der Name des Bibliothekselements entspricht jeweils dem beim Operanden FROM-FILE angegebenen Datei- oder Elementnamen.

**VERSION =**

Angabe der Elementversion.

**VERSION = \*HIGHEST-EXISTING**

Standardmäßig wird die höchste Version verwendet.

**VERSION = \*SAME**

Die Version des Ausgabe-Bibliothekselements entspricht der des Eingabe-Bibliothekselements bzw. der höchstmöglichen Version, wenn eine SAM- oder ISAM-Datei übersetzt werden soll.

**VERSION = <composed-name 1..24>**

Versionsnummer des Elements.

**ELEMENT = <composed-name 1..64>(...)**

Name des Bibliothekselements.

**VERSION =**

Für mögliche Angaben siehe Operand ELEMENT = \*SAME(...)

**TYPE = \*SAME / <alphanum-name 1..8>/**

Bestimmt den Typ des Elements bzw. der Elemente der Ausgabebibliothek.

Wurde beim Operanden FROM-FILE der Name einer SAM- oder ISAM-Datei angegeben, so ist das Element der Ausgabebibliothek automatisch vom Typ J.

**PRODUCE-CONTINUATION =**

Stellung des Fortsetzungszeichens in der Ausgabeprozedur.

Soll die Ausgabeprozedur eine S-Prozedur sein, wird sie in einem variablen Format angelegt. Fortsetzungszeichen werden - wie bei PRODUCE-CONTINUATION=\*NEW-MODE beschrieben - gesetzt.

**PRODUCE-CONTINUATION = \*STD**

Die Stellung des Fortsetzungszeichens entspricht der aktuellen SDF-Einstellung.

Die aktuelle SDF-Einstellung kann mit der Anweisung SHOW-SDF-OPTIONS ausgegeben und mit der Anweisung MODIFY-SDF-OPTIONS CONTINUATION=... geändert werden.

**PRODUCE-CONTINUATION = \*NEW-MODE**

Das Fortsetzungszeichen wird hinter das letzte Zeichen der Zeile geschrieben.

**PRODUCE-CONTINUATION = \*OLD-MODE**

Das Fortsetzungszeichen wird genau in Spalte 72 geschrieben.

**REPLACE-OLD-FILE = \*YES / \*NO**

Gibt an, ob eine bereits bestehende Ausgabedatei bzw. ein Bibliothekselement überschrieben werden darf (\*YES) oder nicht (\*NO).

**SYSTEM-SYNTAX-FILE =**

Name der (System-)Syntaxdatei zur Übersetzung der Benutzerkommandos.

**SYSTEM-SYNTAX-FILE = \$.SYSSDF-CONV.030.USER.U-CMD**

Voreingestellte (System-) Syntaxdatei zur Übersetzung der Benutzerkommandos.

**SYSTEM-SYNTAX-FILE = <filename 1..54>**

Name der (System-)Syntaxdatei, die zur Übersetzung von Benutzerkommandos genutzt werden soll.

**GROUP-SYNTAX-FILE =**

Name der (Gruppen-)Syntaxdatei zur Übersetzung der privilegierten Kommandos.

**GROUP-SYNTAX-FILE = \$.SYSSDF-CONV.030.USER.S-CMD**

Voreingestellte (Gruppen-)Syntaxdatei zur Übersetzung der privilegierten Kommandos.

**GROUP-SYNTAX-FILE = <filename 1..54>**

Name der (Gruppen-)Syntaxdatei, die zur Übersetzung von privilegierten Kommandos genutzt werden soll.

**GROUP-SYNTAX-FILE = \*NO**

Eine Übersetzung der privilegierten Kommandos soll nicht stattfinden. Steht dem Benutzer keine (Gruppen-)Syntaxdatei zur Verfügung, muss \*NO angegeben werden, um einen Abbruch der Übersetzung zu verhindern.

**UNCHANGED-CMD =**

Die Übersetzung eines oder mehrerer Kommandos kann unterdrückt werden. Trifft SDF-CONV auf ein solches Kommando, wird der Kommandosatz unverändert in die Ausgabe-prozedur kopiert. In das Übersetzungsprotokoll wird zusätzlich eine Fehlermeldung mit dem Gewicht `NOTE` geschrieben.

Es ist zu beachten, dass nur solche Kommandos als nicht zu übersetzende Kommandos erkannt werden, deren Schreibweise mit der beim Operanden UNCHANGED-CMD angegebenen Schreibweise übereinstimmt.

*Beispiel*

Um das Kommando ERASE von der Übersetzung auszuschließen, müssen alle seine (in der Eingabeprozedur) benutzten Schreibweisen angegeben werden. In diesem Fall z.B.: ERASE, ERAS, ER.

**UNCHANGED-CMD = \*NONE**

Es sollen keine Kommandos von der Übersetzung ausgeschlossen werden.

**UNCHANGED-CMD = list-poss(30): <structured-name 1..30>**

Die angegebenen Kommandos werden von vornherein von der Übersetzung ausgeschlossen. Die Angabe von bis zu 30 Kommandos bzw. Kommandoschreibweisen ist möglich. Soll die Ausgabe-prozedur eine S-Prozedur sein, wird das Kommando PROCEDURE auch dann gemäß einer S-Prozedur umgewandelt, wenn es bei UNCHANGED-CMD angegeben wurde.

**DOCUMENTATION =**

Inline-Kommentare in Kommandos der Eingabeprozedur gehen bei einer erfolgreichen Umwandlung der Kommandosprache verloren. Eine Ausnahme bilden solche Inline-Kommentare, die vor dem Kommandonamen stehen.

Die mit TO-FILE benannte Ausgabe-prozedur kann nun zusätzliche Informationen in Form von Kommentarzeilen enthalten. Dabei ist die Menge dieser Informationen nicht allein vom Wert des Operanden DOCUMENTATION, sondern auch von der Einstellung des Operanden UNCHANGED-CMD abhängig (siehe Tabelle [7 auf Seite 249](#): für Kommandos, die beim Operanden UNCHANGED-CMD angegeben wurden, werden keine zusätzlichen Kommentarzeilen generiert).

Prozedurparameter, die in den zusätzlich generierten Kommentarzeilen enthalten sind, werden nicht ersetzt.

**DOCUMENTATION = \*NO**

Inline-Kommentare in Kommandos der Eingabeprozedur werden nicht in die Ausgabe-prozedur übernommen.

**DOCUMENTATION = \*MINIMUM**

Kommandosätze der Eingabeprozedur, die Inline-Kommentare enthalten oder aus mehr als einer Eingabezeile bestehen, werden zur Dokumentation in die Ausgabe-prozedur übernommen. Sie stehen (als Kommentarzeile) vor den Zeilen mit der erzeugten Übersetzung.



Konnte der Kommandosatz nicht übersetzt werden, weil die Übersetzung kontext- oder laufzeitabhängig ist, wird in die Kommentarzeile ein Übersetzungsvorschlag geschrieben. Die Kommentarzeilen haben folgenden Aufbau:

```
/"*****" REMARK <kommandosatz der eingabeprozedur bzw. übersetzungsvorschlag>
```

Kommandosätze, die die Kommandos BEGIN-PROCEDURE, PROCEDURE, REMARK (aber nur in Nicht-S-Prozeduren), SET-PROCEDURE-OPTIONS, BEGIN-PARAMETER-DECLARATION, DECLARE-PARAMETER, END-PARAMETER-DECLARATION oder ihre garantierten Abkürzungen enthalten, sind von dieser Regelung ausgeschlossen.

#### **DOCUMENTATION = \*MAXIMUM**

Alle Kommandosätze der Eingabeprozedur werden zusätzlich (als Kommentarzeile) vor die Zeile mit der erzeugten Übersetzung geschrieben. Konnte der Kommandosatz nicht übersetzt werden, weil die Übersetzung kontext- oder laufzeitabhängig ist, wird in die Kommentarzeile ein Übersetzungsvorschlag geschrieben.

Die Kommentarzeilen haben folgenden Aufbau

```
/"*****" REMARK <kommandosatz der eingabeprozedur bzw. übersetzungsvorschlag>
```

Kommandosätze, die die Kommandos BEGIN-PROCEDURE, PROCEDURE, REMARK (aber nur in Nicht-S-Prozeduren), SET-PROCEDURE-OPTIONS, BEGIN-PARAMETER-DECLARATION, DECLARE-PARAMETER, END-PARAMETER-DECLARATION oder ihre garantierten Abkürzungen enthalten, sind von dieser Regelung ausgeschlossen.

#### **TARGET-VERSION =**

Es wird festgelegt, in welcher BS2000-Version die Prozedur ablauffähig sein soll.

#### **TARGET-VERSION = \*CURRENT**

Die übersetzte Prozedur soll unter der BS2000/OSD-Version ablauffähig sein, die zum Zeitpunkt der Übersetzung geladen ist.

#### **TARGET-VERSION = \*V10 / \*V11 / \*OSD-V1**

Die übersetzte Prozedur soll in der entsprechenden BS2000- bzw. BS2000/OSD Version ablauffähig sein. In diesen Fällen werden bei der Übersetzung einiger Kommandos größere Einschränkungen gemacht, als bei der Übersetzung für BS2000/OSD V2 und höher.

Diese Angaben werden nur noch aus Kompatibilitätsgründen unterstützt.

#### **TARGET-VERSION = \*OSD-V2**

Die übersetzte Prozedur soll ab BS2000/OSD-BC V2.0 ablauffähig sein. Diese Angabe ist äquivalent zu TARGET-VERSION = \*CURRENT.

#### **OUTPUT-FORM =**

Bestimmt die Behandlung von optionalen Operanden (also Operanden mit voreingestellten Operandenwerten), die im Kommandosatz nicht explizit angegeben sind.

Unabhängig von OUTPUT-FORM werden alle angegebenen Operanden und Operanden-

werte in die entsprechenden (SDF-)Operanden und Operandenwerte übersetzt, soweit dies nicht eingeschränkt ist (z.B. durch die im [Abschnitt „Einschränkungen“ auf Seite 252](#) aufgelisteten Einschränkungen oder mit dem Operanden UNCHANGED-CMD).

### **OUTPUT-FORM = \*STD**

Nicht angegebene ISP-Operanden und Operandenwerte werden bei der Übersetzung ignoriert, d.h., SDF-CONV übersetzt nur explizit angegebene ISP-Operanden und Operandenwerte in die entsprechenden SDF-Formen.

Zur Zeit der Ausführung der übersetzten Prozedur werden alle nicht angegebenen Operanden mit ihren Standardwerten besetzt.

#### *Hinweis*

Es ist möglich, Standardwerte in den Syntaxdateien zu ändern. Die Ergebnisse der Kommandoausführung können dann voneinander verschieden sein.

### **OUTPUT-FORM = \*ACCEPTED-FORM**

Alle angegebenen SDF-Operanden und Operandenwerte werden in ihren vollen Namen expandiert.

Die Angabe dieses Operandenwertes ist nur mit einer SDF-Version  $\geq 4.0$  möglich.

### **OUTPUT-FORM = \*INVARIANT-FORM(...)**

Auch nicht angegebene ISP-Operanden werden bei der Übersetzung durch ihre entsprechenden SDF-Formen ersetzt und mit ihren Standardwerten belegt.

Dabei ist zu beachten, dass nicht alle SDF-Kommandos mit ihren Operanden und deren Standardwerten unabhängig vom Tasktyp und von der Ablaufart (Aufruf direkt oder innerhalb einer (weiteren) Prozedur) ausführbar sind. In diesen Fällen ist es wichtig, die Umgebung (Tasktyp/Ablaufart) anzugeben, unter der die übersetzte Prozedur laufen soll, da es im Fehlerfall zum Programmabbruch kommt (Kontextabhängigkeit).

Die Angabe dieses Operandenwertes ist nur mit einer SDF-Version  $\geq 3.0$  möglich.

### **CONTEXT =**

Dieser Operand erlaubt die Angabe der Umgebung, in der die übersetzte Prozedur laufen soll. Es werden dann diejenigen SDF-Operanden und Standardwerte generiert, die garantiert in der angegebenen Umgebung ablauffähig sind.

Es besteht weiter die Möglichkeit, bis zu drei verschiedene Umgebungen anzugeben, deren übereinstimmende Operanden und Standardwerte ersetzt werden. Nicht von allen angegebenen Umgebungen unterstützte Operanden oder Standardwerte werden nicht vorbelegt.

### **CONTEXT = \*ALL**

Die Kommandos werden um die Operanden und Standardwerte erweitert, die garantiert in allen möglichen Umgebungen ablauffähig sind. Mögliche Umgebungen sind in diesem Fall:

im Dialogbetrieb:

- geführter Dialog
- ungeführter Dialog

- von der Datensichtstation aufgerufene Prozedur (DIALOG-PROC)
- im Stapelbetrieb:
- in der ENTER-Datei (BATCH)
  - in einer von der ENTER-Datei aufgerufenen Prozedur (BATCH-PROC)

**CONTEXT = \*BATCH**

Die Ausgabebeziehung soll im (einfachen) Stapelbetrieb ablauffähig sein.

**CONTEXT = \*BATCH-PROC**

Die Ausgabebeziehung soll in einer im Stapelbetrieb aufgerufenen Prozedur ablauffähig sein.

**CONTEXT = \*DIALOG-PROC**

Die Ausgabebeziehung soll in einer im Dialog von der Datensichtstation aufgerufenen Prozedur ablauffähig sein.

**PROCEDURE-FORMAT =**

Bestimmt die Art der Übersetzung bzw. Umwandlung der Eingabebeziehung(en).

**PROCEDURE-FORMAT = \*SAME(...)**

Das Prozedurformat der Eingabebeziehung(en) bleibt unverändert.

Die Kommandos der Eingabebeziehung(en) werden in SDF-Kommandos umgewandelt.

**BLOCK-INPUT =**

Bestimmt, wie die von SDF-CONV generierten SDF-Kommandos in die Ausgabebeziehung geschrieben werden.

In S-Prozeduren können mehrere Kommandos in eine Kommandozeile (als "Block" ) geschrieben werden. Dabei werden sie durch je ein Semikolon voneinander getrennt. In Nicht-S-Prozeduren existieren keine Kommandoseparatoren, d.h., jedes Kommando muss in eine eigene Zeile geschrieben werden.

**BLOCK-INPUT = \*NOT-ALLOWED**

SDF-CONV schreibt jedes Kommando der Ausgabebeziehung in eine eigene Zeile.

**BLOCK-INPUT = \*ALLOWED**

Generiert SDF-CONV zu einem ISP-Kommandos mehrere SDF-Kommandos, ist es möglich, diese fortlaufend in eine Kommandozeile zu schreiben. Die Angabe dieses Operandenwertes wird bei der Übersetzung von Kommandos aus Nicht-S-Prozeduren ignoriert.

*Beispiel*

aus /ISP1	könnte werden /SDF11;SDF12
aus /ISP2;ISP3	könnte werden /SDF21;SDF22;SDF23;SDF3

**PROCEDURE-FORMAT = \*S-PROCEDURE(...)**

Die Eingabeprozedur soll in eine S-Prozedur umgewandelt werden. Dabei können die Kommandos der Eingabeprozedur wahlweise in SDF-Kommandos übersetzt werden (Operand CMD-FORMAT), Datenzeilen zu Anweisungszeilen umgewandelt werden (Operand PROGRAM-INPUT) und die Form der Ausgabeprozedur bestimmt werden (Operand BLOCK-INPUT).

**CMD-FORMAT =**

Gibt an, ob die Kommandos der Eingabeprozedur in SDF-Kommandos übersetzt werden sollen oder nicht.

**CMD-FORMAT = \*SDF**

Die Kommandos der Eingabeprozedur sollen in SDF-Kommandos entsprechend den Einstellungen beim Operanden OUTPUT-FORMAT übersetzt werden.

Ist die Eingabeprozedur eine Nicht-S-Prozedur, so wird der Operand UNCHANGED-CMD für das Kommando PROCEDURE ignoriert.

**CMD-FORMAT = \*SAME**

Die Kommandos der Eingabeprozedur bleiben unverändert.

Die Darstellungen der Prozedurparameter werden jedoch an die Vorgaben für eine S-Prozedur angepasst.

*Beispiel*

```
aus /FS &FILENAME                wird /FS &(FILENAME)
aus /REMARK Ersetzen von &LIB     wird /REMARK 'Ersetzen von &&(LIB)'
```

*Hinweise*

- Jobvariablen-Ausdrücke werden folgendermaßen angepasst:

```
aus &(jobvariable) wird &(JV('jobvariable'))
```

Die Jobvariablen müssen in Kommando- und Anweisungszeilen mit dem Zeichen '&' beginnen. In Datenzeilen müssen sie ebenfalls mit '&' beginnen, durch die Angabe PROGRAM-INPUT=\*STMT werden sie aber nicht als Daten, sondern als Anweisungen interpretiert.

- Prozedurparameter werden folgendermaßen angepasst:

```
aus &(procpa) wird &(procpa)
```

Die Prozedurparameter müssen folgenden Bildungsregeln genügen:

- Sie müssen im Prozedurkopf ([BEGIN]-PROC-Zeile) deklariert werden.
- In Kommando- und Anweisungszeilen müssen sie mit dem Zeichen '&' beginnen, in Datenzeilen mit dem bei SUBDTA= angegebenen Zeichen
- Der Parametername darf maximal 20 Zeichen lang sein (Ausnahme: Stellungsparameter, sie werden von SDF-CONV automatisch gekürzt).

**PROGRAM-INPUT =**

Legt fest, ob bei der Umwandlung einer Nicht-S- in eine S-Prozedur Datenzeilen in SDF-Anweisungszeilen umgewandelt werden sollen oder nicht.

Während in S-Prozeduren alle Anweisungszeilen mit zwei Schrägstrichen beginnen

müssen, können in Nicht-S-Prozeduren Anweisungszeilen wie Datenzeilen angegeben werden, also ohne führende Schrägstriche.

In jedem Fall werden aber die Darstellungen der Prozedurparameter an die Vorgaben für eine S-Prozedur angepasst (siehe Beispiel zu CMD-FORMAT=\*SAME).

#### **PROGRAM-INPUT = \*DATA(...)**

Zeilen der Eingabeprozedur ohne führende Schrägstriche werden unverändert, also als Datenzeilen, in die Ausgabeprozedur geschrieben.

##### **EXCEPT-AFTER-CMD =**

Bestimmt, ob bei der Behandlung von Datenzeilen Ausnahmen gemacht werden sollen.

##### **EXCEPT-AFTER-CMD = \*NONE**

Alle Datenzeilen werden unverändert in die Ausgabeprozeduren übernommen.

##### **EXCEPT-AFTER-CMD = list-poss(30): <text 1..30>**

Erlaubt, dass einzelne Datenzeilen ausnahmsweise zu SDF-Anweisungszeilen umgewandelt werden.

Eine Datenzeile wird dann in eine SDF-Anweisungszeile umgewandelt, wenn mindestens ein Element der angegebenen Liste in der letzten Kommandozeile vor der Datenzeile gefunden wird. Die Ausnahmebehandlung endet mit der nächsten Kommandozeile, die keines der angegebenen Elemente enthält.

##### **WARNING = \*NO / \*YES**

Bestimmt, ob eine Meldung mit dem Gewicht `WARNING` ausgegeben werden soll (\*YES) oder nicht (\*NO), wenn SDF-CONV während der Umsetzung auf eine Datenzeile trifft.

#### **PROGRAM-INPUT = \*STMT(...)**

Zeilen der Eingabeprozedur ohne führende Schrägstriche werden zu SDF-Anweisungszeilen umgewandelt. Dazu generiert SDF-CONV vor jede dieser Zeilen zwei Schrägstriche.

##### **EXCEPT-AFTER-CMD =**

Bestimmt, ob bei der Behandlung von Datenzeilen Ausnahmen gemacht werden sollen.

##### **EXCEPT-AFTER-CMD = \*NONE**

Alle Datenzeilen werden zu SDF-Anweisungszeilen umgewandelt.

##### **EXCEPT-AFTER-CMD = list-poss(30): <text 1..30>**

Erlaubt, dass einzelne Datenzeilen nicht zu SDF-Anweisungszeilen umgewandelt, sondern unverändert in die Ausgabeprozedur geschrieben werden.

Eine Datenzeile wird dann unverändert übernommen, wenn mindestens ein Element der angegebenen Liste in der letzten Kommandozeile vor der Datenzeile gefunden wird. Die Ausnahmebehandlung endet mit der nächsten Kommandozeile, die keines der angegebenen Elemente enthält.

**BLOCK-INPUT =**

Bestimmt, wie die von SDF-CONV generierten SDF-Kommandos in die Ausgabeprozedur geschrieben werden.

In S-Prozeduren können mehrere Kommandos in eine Kommandozeile (als "Block") geschrieben werden. Dabei werden sie durch je ein Semikolon voneinander getrennt. In Nicht-S-Prozeduren existieren keine Kommandoseparatoren, d.h., jedes Kommando muss in eine eigene Zeile geschrieben werden.

**BLOCK-INPUT = \*NOT-ALLOWED**

SDF-CONV schreibt jedes Kommando der Ausgabeprozedur in eine eigene Zeile.

**BLOCK-INPUT = \*ALLOWED**

Generiert SDF-CONV zu einem ISP-Kommandos mehrere SDF-Kommandos, ist es möglich, diese fortlaufend in eine Kommandozeile zu schreiben (siehe Beispiel bei PROCEDURE-FORMAT=\*SAME (BLOCK-INPUT=\*ALLOWED)).

### 13.2.3 Steuerfunktionen

Folgende Eingaben dienen zur Steuerung der Dialogführung. Sie werden im Anweisungsteil von SDF-CONV akzeptiert und sofort ausgeführt.

<b>Funktion</b>	<b>Kurzbeschreibung</b>
<anweisung> ?	SDF-CONV wechselt in den (temporär) geführten Dialog
<anweisung> .. <operand> = ?	SDF-CONV informiert über die Funktion des Operanden und die möglichen Operandenwerte
K1 - Taste	kehrt aus dem (temporär) geführten Dialog in den ungeführten Dialog zurück
K2 - Taste	verzweigt in die Kommandoebene; Rückkehr zu SDF-CONV mit dem Kommando RESUME-PROGRAM

Tabelle 6: Steuerfunktionen im Anweisungsteil von SDF-CONV

Eine vollständige Übersicht und die Beschreibung aller möglichen Eingaben im (temporär) geführten und ungeführten Dialog finden Sie im [Kapitel „Kommando- und Anweisungseingabe“](#) auf Seite 33.

## 13.3 Ausgaben

### 13.3.1 Ausgabeprozedur

In der CONVERT-Anweisung wird mit dem Operanden TO-FILE der Name der Ausgabeprozedur bestimmt, bzw. die Namen der Ausgabeprozeduren, wenn mehrere, in Bibliothekselementen abgelegte Eingabeprozeduren übersetzt werden sollen.

Dabei ist zu beachten, dass sich die Angaben bei FROM-FILE und TO-FILE immer voneinander unterscheiden müssen. Unabhängig vom Format der Eingabeprozedur wird für die Ausgabeprozedur eine katalogisierte SAM-Datei oder ein Element in einer PLAM-Bibliothek erzeugt. Mit dem Operanden REPLACE-OLD-FILE = \*YES kann angegeben werden, dass eine bereits existierende Datei bzw. ein Bibliothekselement gleichen Namens überschrieben werden darf. Es werden keine leeren Dateien bzw. Bibliothekselemente erzeugt. Wird also die Konvertierung abgebrochen, bevor ein Kommandosatz in die bei TO-FILE angegebene Ausgabeprozedur geschrieben werden konnte, wird diese Ausgabeprozedur nicht erzeugt.

Die Ausgabeprozedur enthält die übersetzten Kommandos bzw. die aus der Eingabeprozedur unverändert übernommenen ISP-Kommandosätze.

Sind SDF-Kommandosätze Bestandteile der Eingabeprozedur, werden die angegebenen Kommandos in ihren vollen Kommandonamen expandiert.

Abhängig vom Wert der Operanden DOCUMENTATION und UNCHANGED-CMD können auch zusätzlich eingefügte Kommentarzeilen in der Ausgabeprozedur enthalten sein (siehe [Tabelle 7 auf Seite 249](#)). Diese zusätzlich eingefügten Kommentarzeilen können dem Benutzer in folgenden Fällen eine Hilfe sein:

1. für die manuelle Nachbearbeitung der übersetzten Prozedur (Ausgabeprozedur):  
Einige ISP-Kommandos (z.B. JOIN) können nicht automatisch übersetzt werden, da sie (kontextabhängig) verschiedene SDF-Äquivalente haben. In der Kommentarzeile wird von SDF-CONV ein Vorschlag gemacht, wie die Übersetzung lauten könnte. Diese Kommentarzeile wird vor die Kommandozeile mit dem nichtübersetzten Kommando geschrieben. In der Ausgabeprozedur stehen folgende Zeilen:

```
/"*****" REMARK <sdf-kdo = übersetzungsvorschlag>
/<isp-kdo>
```

2. für die Erhaltung von Inline-Kommentaren in einem ISP-Kommandosatz:  
Trifft SDF-CONV auf einen Kommandosatz mit Inline-Kommentaren, werden Kommando, Operanden und Operandenwerte übersetzt, der Kommentar jedoch wird nicht übertragen.

Die Einfügung des ISP-Kommandosatzes gewährleistet, dass die im Kommentar enthaltene Information nicht verloren geht und bei Bedarf in den übersetzten Kommandosatz manuell eingearbeitet werden kann. In der Ausgabeprozedur stehen folgende Zeilen:



```
/"*****" REMARK <isp-kdo " mit inline-kommentar">
/sdf-kdo
```

Die folgende Tabelle zeigt den Umfang der Ausgabe-prozedur in Abhängigkeit von den Einstellungen der Operanden UNCHANGED-CMD und DOCUMENTATION. Das soll am Beispiel der Übersetzung zweier ISP-Kommandos verdeutlicht werden:

```
/SYSFILE SYSLST=( )           → /ASSIGN-SYSLST TO-FILE=*PRIMARY
/ERASE OTTO "LOESCHEN"       → /DELETE-FILE FILE-NAME=OTTO
```

DOC =	UNCHANGED-CMD = *NO <sup>1</sup>	UNCHANGED-CMD = isp-kdo <sup>2</sup>
<b>*NO</b>	<pre>/sdf-kdo ... Das Kommando wird übersetzt.  /ASSIGN-SYSLST TO-FILE=*PRIMARY  /DELETE-FILE FILE-NAME=OTTO</pre>	<pre>/isp-kdo ... Das Kommando wird nicht übersetzt.  /SYSFILE SYSLST=( )  /ERASE OTTO "LOESCHEN"</pre>
<b>*MIN</b>	<pre>/REMARK isp-kdo ... (nicht immer) /sdf-kdo ... Das Kommando wird übersetzt. Ist das Kommando länger als eine Zeile oder enthält es einen Inline-Kommentar, schreibt SDF-CONV diesen Kommando- satz zusätzlich in eine eingefügte Kom- mentarzeile.  /ASSIGN-SYSLST TO-FILE=*PRIMARY  /REMARK ERASE OTTO "LOESCHEN" /DELETE-FILE FILE-NAME=OTTO</pre>	<pre>/REMARK sdf-kdo ... /isp-kdo ... Das Kommando wird nicht übersetzt, aber SDF-CONV schreibt einen Übersetzungsvor- schlag in eine eingefügte Kommentarzeile.  /REMARK ASSIGN-SYSLST TO-FILE=*PRIMARY /SYSFILE SYSLST=( )  /REMARK DELETE-FILE FILE-NAME=OTTO /ERASE OTTO "LOESCHEN"</pre>
<b>*MAX</b>	<pre>/REMARK isp-kdo ... /sdf-kdo ... Das Kommando wird übersetzt. Zusätzlich schreibt SDF-CONV den Kom- mandosatz in eine eingefügte Kommen- tarzeile.  /REMARK SYSFILE SYSLST=( ) /ASSIGN-SYSLST TO-FILE=*PRIMARY  /REMARK ERASE OTTO "LOESCHEN" /DELETE-FILE FILE-NAME=OTTO</pre>	<pre>/REMARK sdf-kdo ... /isp-kdo ... Das Kommando wird nicht übersetzt, aber SDF-CONV schreibt einen Übersetzungsvor- schlag in eine eingefügte Kommentarzeile.  /REMARK ASSIGN-SYSLST TO-FILE=*PRIMARY /SYSFILE SYSLST=( )  /REMARK DELETE-FILE FILE-NAME=OTTO /ERASE OTTO "LOESCHEN"</pre>

Tabelle 7: Dokumentationsgrad in der Ausgabe-prozedur, abhängig von den Werten der Operanden UNCHANGED-CMD und DOCUMENTATION

<sup>1</sup> Vorausgesetzt, dass die Übersetzung des ISP-Kommandos keiner Einschränkung unterliegt.

<sup>2</sup> bzw. das ISP-Kommando kann nicht übersetzt werden, weil seine Bedeutung kontext- oder laufzeitabhängig ist.

### 13.3.2 Ausgaben nach SYSOUT

Nach jedem Konvertierungslauf gibt SDF-CONV eine Übersicht über die Anzahl der bei der Übersetzung aufgetretenen Fehler, Warnungen und Anmerkungen aus. Diese Übersicht ist wie folgt aufgebaut (i = Anzahl):

```
% CVR1000 CONVERSION TERMINATED WITH:
% CVR1001          FAILURES:      i
% CVR1002          ERRORS   :      i
% CVR1003          WARNINGS:      i
% CVR1004          NOTES    :      i
```

FAILURES und ERRORS kennzeichnen schwerwiegende (DMS-) Fehler. Wenn sie auftreten, wird die Fehlermeldung nicht nur nach SYSLST (siehe unten), sondern auch nach SYSOUT ausgegeben. Die Übersetzung durch SDF-CONV wird unter Umständen nach einer solchen Fehlermeldung abgebrochen. SDF-CONV gibt die Übersicht aus und wartet nach der Meldung CVR0060 auf weitere Anweisungen.

Zu Aufbau der Meldungen siehe [Abschnitt „Meldungen von SDF-CONV“ auf Seite 302](#).

### 13.3.3 Übersetzungsprotokoll nach SYSLST

Zusätzlich zur Ausgabeprozedur wird ein Übersetzungsprotokoll erstellt und automatisch nach SYSLST geschrieben. Es ist in drei Teile gegliedert:

1. Teil: Ausgabe der aktuell gültigen Operandenwerte für den durchgeführten SDF-CONV-Lauf.
2. Teil: Ausgabe aller Sätze der Eingabeprozedur und aller Meldungen, die Fehler beim Übersetzen anzeigen.
3. Teil: Übersicht über die Anzahl der Fehler und Anmerkungen (entspricht der Übersicht, die nach SYSOUT ausgegeben wird).

Im zweiten Teil des Protokolls werden in folgenden Fällen Meldungen ausgegeben:

- Es gibt kein SDF-Kommando, das äquivalent dem ISP-Kommando ist.
- Das ISP-Kommando ist nicht eindeutig einem SDF-Kommando zuzuordnen.
- In Verbindung mit bestimmten Operanden haben einige ISP-Kommandos kein SDF-Äquivalent
- Es wurde ein Syntaxfehler gefunden.

Kommandosätze mit solchen Fehlern werden nicht übersetzt, sondern unverändert in die Ausgabeprozedur übernommen. Wird noch vor Beginn der Übersetzung ein ERROR oder FAILURE gemeldet, entfällt der zweite Teil des Protokolls. Zur Bedeutung der Fehlermeldungen siehe Abschnitt "Meldungen von SDF-CONV" im Anhang.

Die Systemdatei SYSLST kann mit dem Kommando ASSIGN-SYSLST einer katalogisierten Datei zugeordnet werden oder mit dem Kommando COPY-SYSTEM-FILE kopiert werden.

### 13.3.4 Programmüberwachung durch Monitor-Jobvariable

Eine programmüberwachende Jobvariable kann von SDF-CONV mit folgenden Werten versorgt werden:

'\$T 0000'	Programm normal beendet. Übersetzung ohne Anmerkungen.
'\$T 0001'	Programm normal beendet. Übersetzung mit Anmerkungen.
'\$T 1002'	Programm normal beendet. Übersetzung mit Warnungen.
'\$T 1003'	Programm normal beendet. Übersetzung mit Fehlern.
'\$A 3004'	Das Programm wurde durch einen schwerwiegenden Fehler vorzeitig beendet oder das CPU-Zeitlimit wurde überschritten.

Die programmüberwachende Jobvariable muss im Kommando START-SDF-CONV (bzw. im Kommando START-EXECUTABLE-PROGRAM SDF-CONV für den kompatibel unterstützten Aufruf) vereinbart werden. Siehe dazu die Handbücher „Kommandos“ [1] oder "Jobvariablen" [9].

## 13.4 Einschränkungen

SDF-CONV kann nicht alle ISP- und alten SDF-Kommandos und deren Operanden in SDF-Kommandos mit den entsprechenden Operanden übersetzen. Es gibt Kommandos, Operanden und Operandenkonfigurationen, die von vornherein von der Übersetzung ausgeschlossen werden. Des weiteren gibt es sog. bedingte Übersetzungen, die versionsabhängig sind.

Einschränkungen kann es auch bei der Übersetzung von Prozedurparametern als Operanden und bei der Umwandlung von Nicht-S- in S-Prozeduren geben.

Die betroffenen Kommandosätze werden unverändert in die Ausgabeprozedur übernommen (SDF-CONV gibt nach SYSLST eine Fehlermeldung aus) bzw. übersetzt, wenn die versionsabhängige Bedingung erfüllt wurde.

### 13.4.1 Allgemeine Voraussetzungen und Einschränkungen

- Fehlerfrei laufende Prozeduren im ISP-Format sind Grundvoraussetzung für fehlerfrei laufende Prozeduren im SDF-Format.
- SDF-CONV übersetzt nur ISP-Kommandos, die im  $BS2000 \leq V10$  definiert wurden.
- Die Operanden FROM-FILE und TO-FILE müssen immer voneinander verschiedene Werte zugewiesen bekommen.  
Das gilt auch, wenn der Operand REPLACE-OLD-FILE=\*YES angegeben ist.
- Kommandozeilen einer Nicht-S-Eingabeprozedur werden nur bis zum 72. Zeichen ausgewertet. Alle Zeichen ab Spalte 73 werden ignoriert.
- Jedes Kommando der Eingabeprozedur, das nicht in einer der Syntaxdateien beschrieben ist, wird unverändert in die Ausgabeprozedur übernommen. Es wird eine Fehlermeldung nach SYSLST ausgegeben.
- Alle Kommandos, die sich auf Kartenleser beziehen, werden unverändert in die Ausgabeprozedur übernommen, da solche Geräte ab V10 nicht mehr vom BS2000 unterstützt werden.
- Zu den SDF-Kommandos können Operanden generiert werden, denen explizit keine Werte zugewiesen sind. Diese Operanden erhalten implizit den in der Operandenbeschreibung des entsprechenden Kommandos angegebenen Standardwert (siehe Handbuch „Kommandos“ [1]).

- Prozedurparameter dürfen nicht Teil eines Kommandos oder eines Operanden und nicht selber Kommando oder Operand sein.

*Beispiel*

```
/PROC N,(&PARAM1='SHOW',&PARAM2='FILE')
/&PARAM1-SDF-OPTIONS
/CREATE-FILE &PARAM2=OTTO.DAT
/...
/ENDP
```

- Es können nur solche Kommandos mit Prozedurparametern in das SDF-Format übertragen werden, deren Übersetzung unabhängig von den Parametern ist. In allen übrigen Fällen kann keine eindeutige Zuordnung erfolgen, die Kommandos werden nicht übersetzt (siehe [Seite 262ff](#)).

*Beispiel*

```
/PROC N,(&USE1)
/...
/RFD USE=&USE1
/...
/ENDP
```

```
&USE1 = NO    wird in SDF: /STOP-DISKETTE-INPUT
&USE1 = INPUT wird in SDF: /START-DISKETTE-INPUT
```

- Enthält die Eingabeprozedur das SDF-Kommando MODIFY-SDF-OPTIONS, darf der Wert des Operanden CONTINUATION kein Prozedurparameter sein.
- Enthält die Eingabeprozedur das SDF-Kommando ENDP-RESUME und soll die Ausgabeprozedur eine S-Prozedur sein, so wird das Kommando in EXIT-PROCEDURE RESUME=YES umgewandelt.  
Das Kommando wird nicht umgewandelt, wenn die Ausgabeprozedur eine Nicht-S-Prozedur sein soll, da sich in diesem Fall das Spin-Off-Verhalten der Prozedur ändert.
- Alle Namen, die mit einem Stern beginnen und keinen weiteren Platzhalter enthalten, werden als Schlüsselwörter interpretiert.
- Das alte Markenformat (.marke) wird nicht in das neue Markenformat (marke:) umgewandelt.
- Soll das Kommandoformat einer S-Prozedur von ISP- auf SDF-Kommandoformat umgestellt werden, können geschachtelte Prozedurparameter oder geschachtelte zu ersetzende Jobvariablen nicht erkannt werden. Die betroffenen Kommandosätze werden von SDF-CONV nicht übersetzt.
- Inline-Kommentare gehen bei einer erfolgreichen Umwandlung der Kommandosprache durch SDF-CONV verloren. Durch Angabe von DOCUMENTATION=\*MIN/\*MAX in der CONVERT-Anweisung können Inline-Kommentare in die Ausgabeprozedur übernommen werden.

### 13.4.2 Einschränkungen bei der Umwandlung zu S-Prozeduren

- ENTER-Dateien können mit SDF-CONV nicht in S-Prozeduren umgewandelt werden.
- SDF-CONV gibt eine Fehlermeldung aus, wenn der Name eines Prozedurparameters länger als 20 Zeichen ist. Namen von Prozedurparametern dürfen in Nicht-S-Prozeduren max. 255 Zeichen lang sein, während in S-Prozeduren nur max. 20 Zeichen zugelassen sind.  
Diese Einschränkung gilt nicht für Stellungsoperanden, da diese bei Bedarf von SDF-CONV automatisch gekürzt werden.
- Bei der Umwandlung kann sich die Steuerung der Protokollierung ändern. Bei Nicht-S-Prozeduren wird die Protokollierung nur von der aufgerufenen Prozedur bestimmt, bei S-Prozeduren zusätzlich von der Einstellung, die die aufrufende Umgebung hat.
- Es findet keine Umwandlung statt, wenn der Wert eines Prozedurparameters mehrere zu ersetzende Jobvariablen enthält.
- Ist in der Nicht-S-Eingabeprozedur das erste signifikante Zeichen nach einem Kommandonamen das Gleichheitszeichen, gibt SDF-CONV eine `WARNING` aus. Eine Ausnahme bilden Kommandosätze mit dem `REMARK`-Kommando.
- Die zweistufige Ersetzung von Prozedurparametern wird von SDF-P nicht unterstützt.
- Prozedurparameter und Jobvariablen in S-Prozeduren können innerhalb der einschließenden Klammer weitere zu ersetzende Ausdrücke enthalten, z. B. `&(JOBVAR1&(JOBVAR2))`. Diese Eigenschaft wird von SDF-CONV nicht unterstützt.

#### *Beispiel*

##### 1. Nicht-S-Prozedur:

```
/BEGIN-PROC A,PAR=YES(PROC-PAR=(&TEST=))
/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
/SHO-JV JV(JV-NAME=JOBVAR)
/REMARK &TEST
/WR-TEXT '&(JOBVAR) '
/EXIT-PROC
```

##### Ablaufprotokoll der Nicht-S-Prozedur:

```
/CALL-PROC BSP-NICHT-S
%/BEGIN-PROC A,PAR=YES(PROC-PAR=(&TEST=))
%/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&TEST'
%/SHO-JV JV(JV-NAME=JOBVAR)
%&TEST
%/REMARK &TEST
%&TEST=hallo
%/REMARK HALLO
```

```
%/WR-TEXT 'HALLO'
HALLO
%/EXIT-PROC
```

**Die Jobvariable JOBVAR wurde zweistufig ersetzt.**

**Der Inhalt der Jobvariablen JOBVAR ist der Wert des Prozedurparameters TEST, also HALLO.**

## 2. S-Prozedur:

```
/SET-PROC-OPTIONS LOG=YES
/BEGIN-PAR-DEC
/DECLARE-PAR NAME=TEST (INIT=*PROMPT)
/END-PAR-DEC
/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
/SHO-JV JV(JV-NAME=JOBVAR)
/REMARK &TEST
/WR-TEXT '&(JV('JOBVAR'))'
```

### Ablaufprotokoll der S-Prozedur:

```
/CALL-PROC BSP-S,LOG=YES
% 1 1 /SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=YES,INTERRUPT-
% ALLOWED=YES,DATA-ESCAPE-CHAR=NONE,SYSTEM-FILE-CONTEXT=OWN,-
% DATA-ERROR-HANDLING=NO
% 1 1 /BEGIN-PARAMETER-DECLARATION
% 1 1 /DECLARE-PARAMETER NAME=TEST,TYPE=ANY,INIT=*PROMPT
% 1 1 /END-PARAMETER-DECLARATION
% 6 1 /MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
% 7 1 /SHOW-JV JV(JV-NAME=JOBVAR)
%&TEST
%TEST: hallo
% 8 1 /REMARK 'HALLO'
% 9 1 /WRITE-TEXT '&TEST'
&TEST
% 11 1 /IF-BLOCK-ERROR
% 11 1 /END-IF
% 11 1 /EXIT-PROCEDURE
```

**Die Jobvariable JOBVAR wurde nur einmal ersetzt.**

**Der Inhalt der Jobvariablen JOBVAR ist &TEST.**

### 13.4.3 Nicht übersetzbare Kommandos

- für nichtprivilegierte Benutzer

ISP-Kommando	Erläuterung
ACCOUNT	Das Kommando ist nicht mehr verfügbar.
AUDIT	Es ist keine eindeutige Zuordnung möglich (Übersetzung ist laufzeitabhängig).
DATA	Das Kommando hat kein SDF-Äquivalent.
DIAG	DIAG ist ein IDA-Kommando, kein ISP-Kommando.
DO	Das Kommando kann mit globalen Parametern kollidieren (Übersetzung ist kontextabhängig). Ein Übersetzungsvorschlag steht in einer Kommentarzeile der Ausgabeprozedur, wenn der Operand DOCUMENTATION = *MIN/*MAX eingestellt ist.
END	Das Kommando hat kein SDF-Äquivalent.
PARAMETER	Das Kommando hat kein SDF-Äquivalent.

Tabelle 8: Nicht übersetzbare Kommandos für nichtprivilegierte Benutzer

- für die Systemverwaltung

ISP-Kommando	Erläuterung
JOIN	Es ist keine eindeutige Zuordnung möglich (Übersetzung ist kontextabhängig). Ein Übersetzungsvorschlag steht in einer Kommentarzeile der Ausgabeprozedur, wenn der Operand DOCUMENTATION = *MIN/*MAX eingestellt ist.
LOADAID	Das Kommando hat kein SDF-Äquivalent.
LOADAIDSYS	Das Kommando hat kein SDF-Äquivalent.
RCARD	Kartenleser werden nicht mehr unterstützt.
SDVC	Es ist keine eindeutige Zuordnung möglich (Übersetzung ist kontextabhängig).

Tabelle 9: Nicht übersetzbare Kommandos für die Systemverwaltung



- für die Operator an der Konsole

ISP-Kommando	Erläuterung
AGOGO ASR ASTOP BCACT BCAPPL BCASP BCCONN BCCONP BCDAC BCDISCON BCDISP BCEND BCGEN BCIN BCLOSE BCMAP BCMOD BCMOFF BCMON BCOPTION BCOUT BCSET BCSHOW BCSWP BCTIMES BCXAF BROADCAST CONSOLE DADM DCDIAG DCSTART MESSAGE MRSEND MRSMOD MRSSTART SHUTDOWN TURN	Diese Kommandos für die Operator können nur an der Konsole und nicht an einer Datensichtstation eingegeben werden.

Tabelle 10: Nicht übersetzbare Kommandos für die Operator an der Konsole

### 13.4.4 Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen

Folgende Kommandos werden nicht oder nur bedingt übersetzt, wenn die genannten Operanden angegeben sind.

- Kommandos für nichtprivilegierte Benutzer

Zu übersetzen- des Kommando	nicht übersetzbare bzw. die Übersetzung ein- schränkende Operanden	Erläuterung
FILE	LINK  *DUMMY, LINK, TSET zu- sammen	Das Kommando wird nicht übersetzt, wenn bei Angabe des Operanden LINK noch andere Operanden als der Dateiname angegeben wurden oder diese drei Operanden zusammen in einem FILE-Kommando angegeben wurden.
MODIFY-JV- CONDITIONALLY	SET-VALUE	Wegen der Änderung der JV-Syntax in JV V11.2 (ab BS2000/OSD-BC V2.0) kann es zu Fehlern bei der Interpretation des Linknamens kommen.
PRINT	(datei1 , datei2 , ...)  CONTROL=12LPI  PNAME=	Format3 (= mehrere Dateien) wird nur übersetzt, wenn die Dateien mit vollem oder teilqualifizierten Namen bzw. mit ihren <eam-datei-nummern> angegeben werden. Die Angabe mehrerer *SYSLSTnn- Dateien wird nicht unterstützt. Der Operand wird nicht mehr unterstützt.  Ein im Namen enthaltener Bindestrich wird bei der Übersetzung zu PRINT-FILE SPOOL-OUT-NAME=... ersatzlos gestrichen.

Tabelle 11: Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen (nicht-privilegierte Benutzer)

Zu übersetzen- des Kommando	nicht übersetzbare bzw. die Übersetzung ein- schränkende Operanden	Erläuterung
PRINT-FILE	LAYOUT-CONTROL CHARACTER-SETS	Dieser Operand führt zu Problemen bei gleichzeitiger Angabe von PRINTER-RESSOURCES=APA( TABLE-REFERENCE-CHAR).
	CHARS-MODIFICATION	Prozedurparameter werden als Operandenwert nicht akzeptiert.
	CONTROL-CHARACTERS	Prozedurparameter werden als Operandenwert nicht akzeptiert.
	=PHYSICAL( LINE-SPACING=...)	Prozedurparameter werden als Operandenwert nicht akzeptiert. Wegen der Unterscheidung in SPOOL und RSO im Kommando PRINT-DOCUMENT wird der Standardwert ignoriert.
	=PHYSICAL(... , PAGE-CONTROL=..)	Prozedurparameter werden als Operandenwert nicht akzeptiert.
	=TRANSPARENT()	Wegen der Unterscheidung in SPOOL und RSO im Kommando PRINT-DOCUMENT wird der Standardwert ignoriert.
	PRINTER-RESSOURCES =APA(..., TABLE-REFERENCE- CHAR=...)	Prozedurparameter werden als Operandenwert nicht akzeptiert.
	TRAY-NUMBER	Dieser Operand führt zu Problemen bei gleichzeitiger Angabe von PRINTER-RESSOURCES=LP65( INPUT-TRAY-NUMBER=...).
	ROTATION	Die Angabe einer Liste wird nicht unterstützt.
	DELETE-FILE	Prozedurparameter werden bei der Übersetzung akzeptiert, dabei wird jedoch ein bei ERROR-PROCESSING=*PAR( TRUNCATION=...) angegebener Wert ignoriert.
DEVICE-NAME =...(DESTINATION=...)	Prozedurparameter werden als Operandenwert nicht akzeptiert.	
=...(DEVICE-TYPE=...)	Dieser Operand führt zu Problemen bei gleichzeitiger Angabe von PRINTER-RESSOURCES=LP65/APA.	

Tabelle 11: Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen  
(nicht-privilegierte Benutzer)

Zu übersetzen- des Kommando	nicht übersetzbare bzw. die Übersetzung ein- schränkende Operanden	Erläuterung
PRINT-FILE (Fortsetzung)	ERROR-PROCESSING  =*PAR(TRUNCATION=...)  FILE-NAME =*SYSLST-NUMBER()  =*LIB-ELEM(..., CREATION-DATE=...)  ----  FILE-PART  =SECTION( SECTION-ID=... ,FIRST-RECORD=... ,LAST-RECORD=...)  RECORD-PART =PAR(FIRST-CHAR=... ,LAST-CHAR=...)  REPEAT  SPOOL-OUT-NAME  START-SPOOL  =NO	Prozedurparameter werden als Operandenwert nicht akzeptiert.  Prozedurparameter werden als Operandenwert nicht akzeptiert.  Die Angabe von *EAM() und *SYSLST-NUMBER() in Form von Listen wird nicht unterstützt.  Diesen Unteroperanden gibt es nicht im Kommando PRINT-DOCUMENT.  Prozedurparameter werden als Operandenwerte akzeptiert; ist der Prozedurparameter jedoch ein Integerwert (z.B. eine SYSLST-Dateinummer) ist das Konvertierungsergebnis falsch.  Prozedurparameter werden als Operandenwert nicht akzeptiert.  Prozedurparameter werden als Operandenwerte nicht akzeptiert.  Ist der Operandenwert Null, wird er zu Eins konvertiert.  Prozedurparameter werden als Operandenwert nicht akzeptiert.  Prozedurparameter werden als Operandenwert nicht akzeptiert.  Prozedurparameter werden als Operandenwert nicht akzeptiert.  Dieser Operandenwert wird nicht mehr unterstützt.
PUNCH	(datei1 , datei2 , ...)  <eam-datei-nummer>  TAPE=	Format3 (= mehrere Dateien) hat kein SDF-Äquivalent.  Dateinamen, die nur aus Ziffern bestehen, sind nicht erlaubt.  Der Operand wird nicht mehr unterstützt (Ausnahme: TAPE=NO).

Tabelle 11: Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen  
(nicht-privilegierte Benutzer)

Zu übersetzen- des Kommando	nicht übersetzbare bzw. die Übersetzung ein- schränkende Operanden	Erläuterung
RESUME	mit Operanden	RESUME mit Angabe von Operanden ist ein IDA-Kommando.
RESUME- PROCEDURE	MODE=C	Der Operand wird nicht übersetzt da C sowohl für CURRENT als auch für COMMAND bzw. CMD stehen kann.
SECURE	WORK=...	Der Operand hat kein SDF- Äquivalent.
SYSFILE	FILE=...	Der Operand ist zu komplex, keine eindeutige Zuordnung möglich. Eine Übersetzung erfolgt nur dann, wenn der Dateiname wie folgt angegeben wird: FILE=<filename without-gen-vers>
TCHNG	CORR=C'a'	Der Operand wird nicht mehr unterstützt.

Tabelle 11: Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen (nicht-privilegierte Benutzer)

- Kommandos für die Systemverwaltung

ISP-Kommando	nicht übersetzbare bzw. die Übersetzung ein- schränkende Operanden	Erläuterung
EXCAT	QUIET	Der Operand wird nicht mehr unterstützt.
IMCAT	FORM=CAT  RESET= BUFNUM=	Der Operand wird nicht mehr unterstützt. Ein Übersetzungsvorschlag steht in der Kommentarzeile der Ausgabeprozedur, wenn der Operand DOCUMENTATION =*MIN/*MAX eingestellt ist.  Die Operanden werden nicht mehr unterstützt.
RFD	USERID=userid OWNERID='ownerid'	Die Schlüsselwörter müssen angegeben werden.

Tabelle 12: Nicht übersetzbare Operandenkonfigurationen und bedingte Übersetzungen (Systemverwaltung)

### 13.4.5 Einschränkungen bei Prozedurparametern als Operanden

Folgende Beispiele zeigen die Schwierigkeiten, die entstehen können, wenn ISP-Kommandos mit Prozedurparametern als Operanden in das SDF-Format übersetzt werden sollen.

1. Der ISP-Operandenwert bestimmt das SDF-Kommando:

```
/CATJV ... ,STATE=UPDATE    → /MODIFY-JV-ATTRIBUTES ...
/CATJV ... ,STATE=NEW        → /CREATE-JV ...
/CATJV ... ,STATE=&OP1       → / ?
```

2. Der ISP-Operandenwert bestimmt den SDF-Operandenwert:

```
/EXECUTE ... ,SYMTEST=NO     → /START-PROGRAM ... ,TEST-OPTIONS=NONE
/EXECUTE ... ,SYMTEST=ALL    → /START-PROGRAM ... ,TEST-OPTIONS=AID
/EXECUTE ... ,SYMTEST=&OP2   → /START-PROGRAM ... ,TEST-OPTIONS= ?

/GETJV <name>                → /SHOW-JV JV-ID=JV-NAME(JV-NAME=<name>)
/GETJV *<name>               → /SHOW-JV JV-ID=JV-LINK(JV-LINK=<name>)
/GETJV &OP3                  → /SHOW-JV JV-ID= ?
```

#### *Hinweis*

Kommandosätze, die Prozedurparameter enthalten oder in denen Jobvariablen ersetzt werden sollen, können von der Übersetzung ausgeschlossen werden. Dazu wird in der CONVERT-Anweisung der Operand `PARAMETER-LINES = *COPY-ONLY` angegeben. SDF-CONV übersetzt die betroffenen Kommandosätze nicht, sondern schreibt sie unverändert in die Ausgabeprozedur. In diesen Fällen werden keine Meldungen ausgegeben.

Die folgenden Tabellen zeigen eine Übersicht über die Kommandos, die nicht übersetzt werden, wenn die aufgelisteten Stellungs- und/oder Schlüsselwortoperanden als Prozedurparameter angegeben sind. Es bedeuten:

<i>                    der i-te Stellungsoperand darf kein Parameter sein  
 OPERAND=            der Wert des Schlüsselwortoperanden darf kein Parameter sein

- Kommandos für nicht-privilegierte Benutzer

ISP-Kommando	kein Parameter als Stellungs-operand ...	kein Parameter als Wert des Schlüsselwort-operanden ...
CANCEL	<2> ,<3>	
CATJV		SHARE=, STATE=
COPY		DIALOG=, WRITE=
DELON	<1>	
ENTER		LOG=, TIME=
ERAJV		alle Operanden
ERASE	<1> ,<2>	CR=, EX=, LA=, LIST=, POS=
EXECUTE	<1>	CLASSII=, IDA=, SYMTEST=
FSTATUS	<2>	CRDATE=, EXDATE=, EXTENDS=, FREESIZE=, FROM=, LADATE=, LIST=, PASS=, SHARE=, SIZE=, SORT=, SUPPORT=, STATE=, TYPE=, VTOC=
GETJV	<1>	
HELP		INF=
IMPORT		LIST=
LOAD	<1>	CLASSII=, IDA=, SYMTEST=
LOGOFF	<1> ,<2>	
OPTION		MSG=, TESTPRIV=
PASSWORD		REL=
PRINT	<1> ,<2>	BINARY=, CHARS-POOL=,CONTROL=, COPIES=, DEFER=, DELETE-FILE=, DEVICE=, DEVIN=, FROM=, HEAD=,LOOP=, ROTATION=, SPACE=, TAPE=, TO=, TRANSLATION-TABLE=, VOLUME=
PRIORITY	<2>	
PUNCH	<1> ,<2>	ACCESS=, BY-PASS=, DEVICE=, DEVIN=, FDTYPE=, SKEL=, TAPE=, VOLUME=, BLKTYPE=, WRITEPR=
RDTFT	<1>	LINK=
RELEASE	<2>	
RESTART	<2> ,<3>	CHECK=
SECURE		FILE=, VOLUME=
SETJV	<1> ,<2>	
SETSW		INVERT=, OFF=, ON=

Tabelle 13: Einschränkungen bei Prozedurparametern als Operanden (nicht-privilegierte Benutzer)

ISP-Kommando	kein Parameter als Stellungs-operand ...	kein Parameter als Wert des Schlüsselwort-operanden ...
SET-SS-OPTIONS		SS-NAME=
SKIP	<1>	
SPARAM		COMPRESS=
SQUC		REL=, SUSP=, TYPE=
STAJV	<1> ,<2>	
STATUS	<1> ,<2>	DISP=, IDENT=, INTYPE=, TIMEREQ=, TYPE=
SYSFILE		SYSDTA=, SYSIPT=, SYSLST=, ..., TASKLIB=
SYSTATUS	<1>	
TCHNG		OFLOW=, READ=, TCHAR=
VERIFY		REPAIR=, SUPPORT=

Tabelle 13: Einschränkungen bei Prozedurparametern als Operanden (nicht-privilegierte Benutzer)

- Kommandos für die Systemverwaltung

ISP-Kommando	kein Parameter als Stellungs-operand ...	kein Parameter als Wert des Schlüsselwort-operanden ...
CATM		BUFCLS=, STATE=, WAIT=
EXCAT	<2>	
IMCAT		ACNTNUM=, BUFCLS=, FORM=
MSGCONTROL		FILE=
RFD		USE=
SEVER	<2>	
SPMGT		SECONDARY=
SQUC		REL=, SUSP=, TYPE=
STAM	<1>	HOST=, REF=

Tabelle 14: Einschränkungen bei Prozedurparametern als Operanden (Systemverwaltung)



## 13.5 Bestandteile und Installation von SDF-CONV

Das Dienstprogramm SDF-CONV ist unter BS2000/OSD ablauffähig. SDF-CONV wird als Subsystem ausgeliefert, kann aber aus Kompatibilitätsgründen auch als Programm gestartet werden. Es werden ISP-Kommandos übersetzt, die in einer BS2000-Version  $\leq V10$  definiert wurden.

Das Dienstprogramm SDF-CONV besteht aus folgenden Teilen:

- der Modulbibliothek SYSLNK.SDF-CONV.030 für den Aufruf des Dienstprogramms als Subsystem mit dem Kommando START-SDF-CONV
- dem Übersetzungsprogramm SYSPRG.SDF-CONV.030, das aus Kompatibilitätsgründen für den Aufruf mit dem Kommando START-EXECUTABLE-PROGRAM zur Verfügung steht
- den Syntaxdateien
  - SYSSDF.SDF-CONV.030 (System-)Syntaxdatei mit der Syntax der Programmanweisungen für SDF-CONV
  - SYSSDF.SDF-CONV.030.USER.U-CMD (System-)Syntaxdatei und
  - SYSSDF.SDF-CONV.030.USER.S-CMD (Gruppen-)Syntaxdatei mit der Kommandosyntax der zu übersetzenden Kommandos, getrennt nach Funktionsumfang für die Systemverwaltung (Gruppen-Syntaxdatei) und die nichtprivilegierten Benutzer (System-Syntaxdatei)
- der Deklarationsdatei
  - SYSSSC.SDF-CONV.030.112
- der Meldungsdatei
  - SYSMES.SDF-CONV.030
- der Repdatei SYSREP.SDF-CONV.030 für Objektkorrekturen.

### Installation des Subsystems:

SDF-CONV ist ein Subsystem des BS2000 und wird von DSSM verwaltet. Um von DSSM geladen zu werden, muss es im Subsystemkatalog deklariert sein. Die dazu benötigten Subsystem-Deklarationen sind in der Datei SYSSSC.SDF-CONV.030.112 enthalten.

Die Systemverwaltung muss das Subsystem wie folgt installieren:

Die Subsystem-Deklaration SYSSSC.SDF-CONV.030.112 muss mit dem Programm SSCM in den Subsystemkatalog eingebracht werden. Mit dem Kommando START-SUBSYSTEM wird das Subsystem SDF-CONV aktiviert. Dazu gehören auch die Meldungsdatei, die Syntaxdatei SYSSDF.SDF-CONV.030 und die Repdatei; sie werden automatisch aktiviert.

Die beiden anderen Syntaxdateien (SYSSDF.SDF-CONV.030.USER.U-CMD und SYSSDF.SDF-CONV.030.USER.S-CMD) werden unter der System-Standardkennung "\$." abgelegt. Um sie auf einer anderen Kennung zu installieren, müssen bei den Operanden SYSTEM-SYNTAX-FILE und GROUP-SYNTAX-FILE der CONVERT-Anweisung die voreingestellten Pfadnamen entsprechend geändert werden.

Diese Syntaxdateien dürfen nicht in die aktuellen System- oder Gruppen-Syntaxdateien des BS2000 gemischt werden!

Damit SDF-CONV mit dem Kommando START-EXECUTABLE-PROGRAM aufgerufen werden kann, muss folgendes Kommando eingegeben werden:

```
/MODIFY-FILE-ATTRIBUTES SYSPRG.SDF-CONV.030, NEW-NAME=SDF-CONV,-  
/ USER-ACCESS=*ALL
```

Die Erstellung des Subsystemkatalogs und das Programm SSCM sind im Handbuch "Einführung in die Systemverwaltung" [2], die Verwaltung von System- und Gruppen-Syntaxdateien im [Kapitel „SDF-Syntaxdateien“ auf Seite 139](#), das Dienstprogramm SDF-I im [Kapitel „SDF-I“ auf Seite 167](#) und das Programm UGEN im Handbuch "Systeminstallation" [3] beschrieben.

#### *Hinweis*

Werden im System optionale Reps eingesetzt, können voreingestellte Operandenwerte (z. B. beim Kommando CATALOG) geändert werden. SDF-CONV berücksichtigt diese Änderungen nur dann, wenn sie der Systemverwalter mit dem Dienstprogramm SDF-A in die Syntaxdateien aufnimmt.

## 13.6 Beispiele

Dieses Kapitel enthält Beispiele für die Umwandlung von Nicht-S-Prozeduren zu S-Prozeduren sowie für die Umwandlung von Kommandos in ISP-Kommandosprache zu Kommandos in SDF-Kommandosprache.

Für jedes der Beispiele werden die Ein- und Ausgabeprozeduren sowie die von SDF-CONV erzeugten Protokolle gezeigt und kommentiert.

Beispiel 1: Die Prozedur DO.REORG reorganisiert Speicherplatz von ISAM-Dateien. Die Übersetzung in SDF-Kommandos sowie die Umwandlung in eine S-Prozedur erfolgt ohne Einschränkungen. SDF-CONV wurde ohne Fehlermeldung beendet.

Eingabeprozedur: DO.REORG  
Protokolldatei: LST.REORG  
Ausgabeprozedur: SDF.SDFP.DO.REORG

Beispiel 2: Die Prozedur DO.LIB.INHALT veranlasst die Ausgabe der wichtigsten Elementdaten aller Elemente einer Bibliothek. Die Übersetzung erfolgt mit Einschränkungen beim PRINT- und DO-Kommando. SDF-CONV meldet je eine NOTE.

Das Prozedurformat (Nicht-S-Prozedur) bleibt unverändert.

Eingabeprozedur: DO.LIB.INHALT  
Protokolldatei: LST.LIB.INHALT  
Ausgabeprozedur: SDF.DO.LIB.INHALT

Beispiel 3: Die Prozedur PROC.ASSXT ist eine kleine Hilfsprozedur, die bei der häufigen Nutzung des ASSEMBH-Assemblers ein schnelleres Abarbeiten (Zuweisung von Systemdateien, Aufruf des Assemblers) ermöglichen soll. Sie ist in SDF-Kommandosprache geschrieben. Die Umwandlung dieser Nicht-S-Prozedur in eine S-Prozedur erfolgt ohne Einschränkungen. SDF-CONV wurde ohne Fehlermeldung beendet.

Eingabeprozedur PROC.ASSXT  
Protokolldatei: LST.ASSXT  
Ausgabeprozedur: SDFP.PROC.ASSXT

**Beispiel 1:**

Die Prozedur DO.REORG reorganisiert den Speicherplatz einer ISAM-Datei. Die Übersetzung in SDF-Kommandos sowie die Umwandlung in eine S-Prozedur erfolgt ohne Einschränkungen. SDF-CONV wurde ohne Fehlermeldung beendet.

**Eingabeprozedur DO.REORG**

```

/PROC N,(&DATEI),SUBDTA=&
/DCLJV #NEUBILD
/SETJV #NEUBILD,X'OC'
/SETSW ON=(4,5), OFF=(15)
/SYSFILE SYSDTA=(SYSCMD)
/WR-TEXT ' '
/WR-TEXT '*****'
/WR-TEXT '***      Reorganisieren von ISAM-Dateien      ***'
/WR-TEXT '*****'
/WR-TEXT ' '
/WR-TEXT 'Name der zu reorganisierenden Datei: '
/REMARK &DATEI
/EXEC $EDT
@GET '&DATEI' N
@CON
@IF N:@IF N D:@SY'COPY &DATEI,&DATEI..VORHER,SAME'
@CON
@IF N : @IF N D : @SAVE O
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP .FEHL, ON=(15)
/TCHNG OFLOW=NO
/WR-TEXT '&(#NEUBILD)'
/TCHNG OFLOW=ACK
/WR-TEXT ' '
/WR-TEXT 'Die Datei &DATEI wurde reorganisiert.'
/WR-TEXT ' '
/WR-TEXT 'Speicherplatzbedarf vorher:'
/FS &DATEI..VORHER
/WR-TEXT ' '
/ERASE &DATEI..VORHER
/WR-TEXT 'Speicherplatzbedarf jetzt:'
/FS &DATEI
/SKIP .EOP, OFF=(15)
/.FEHL REMARK
/WR-TEXT ' '
/WR-TEXT '***** Fehler bei der Dateibearbeitung *****'

```

```

/WR-TEXT '** REORGANISATION WURDE NICHT DURCHGEFUEHRT ! **'
/WR-TEXT ' '
/.EOP STEP
/SETSW OFF=(4,5,15)
/SYSFILE SYSDTA=(PRIMARY)
/ENDP

```

### SDF-CONV-Ablauf

```

//start-sdf-conv _____ (1)
% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0001 SDF-CONV VERSION 'V03.0B80' STARTED
% CVR0060 SDF-CONV VERSION 'V03.0B80' READY
//convert from-file=do.reorg,to-file=sdf.sdfp.do.reorg,doc=*min,- _____ (2)
//procedure-format=*s-proc(prog-inp=*data,block-inp=*allowed)
% CVR1000 CONVERSION TERMINATED WITH:
% CVR1001          FAILURES:    0
% CVR1002          ERRORS   :    0
% CVR1003          WARNINGS:    0
% CVR1004          NOTES    :    0
//end _____ (3)

```

- (1) Start des Dienstprogramms SDF-CONV.
- (2) In der CONVERT-Anweisung werden die Ein- und Ausgabeprozedur zugewiesen sowie die Dokumentationsmenge bestimmt. Die Prozedur soll in eine S-Prozedur umgewandelt werden. Die geblockte Schreibweise von generierten SDF-Kommandos wird erlaubt. Alle anderen Operanden der CONVERT-Anweisung behalten den Standardwert. Nach erfolgter Abarbeitung meldet SDF-CONV, dass die Umwandlung ohne Warnungen oder Fehler durchgeführt wurde. Diese Übersicht wird nach SYSOUT ausgegeben.
- (3) Mit der Anweisung END wird SDF-CONV beendet.

**Ablaufprotokoll (SYSLST)**

```

***** BS2000 COMMAND CONVERTER : SDF-CONV V3 ***** page 1

% CVR1010 OPTIONS IN EFFECT: _____ (1)
% CVR1011 FROM-FILE = DO.REORG
% CVR1012 EXPECT-CONTINUATION = *NEW-MODE
% CVR1013 PARAMETER-LINES = *CONVERT
% CVR1014 TO-FILE = SDF.SDFP.DO.REORG
% CVR1015 PRODUCE-CONTINUATION = *NEW-MODE
% CVR1016 REPLACE-OLD-FILE = *YES
% CVR1017 SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018 GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019 UNCHANGED-CMD = *NONE
% CVR1020 DOCUMENTATION = *MINIMUM
% CVR1021 TARGET-VERSION = *OSD-V2
% CVR1022 OUTPUT-FORM = *STD
% CVR1023 PROCEDURE-FORMAT = *S-PROCEDURE(CMD-
FORMAT=*SDF,PROGRAM-INPUT=*DATA(EXCEPT-AFTER-CMD=*NONE,WARNING=*NO),BLOCK-
INPUT=*ALLOWED)

```

```

***** BS2000 COMMAND CONVERTER : SDF-CONV V3 ***** page 2

/PROC N,(&DATEI),SUBDTA=& _____ (2)
/DCLJV #NEUBILD
/SETJV #NEUBILD,X'OC'
/SETSW ON=(4,5),OFF=(15)
/SYSFILE SYSDTA=(SYSCMD)
/WR-TEXT ' '
/WR-TEXT '*****'
/WR-TEXT '*** REORGANISIEREN VON ISAM-DATEIEN ***'
/WR-TEXT '*****'
/WR-TEXT ' '
/WR-TEXT 'NAME DER ZU REORGANISIERENDEN DATEI: '
/REMARK &DATEI
/EXEC $EDT
@GET '&DATEI' N
@CON
@IF N:@IF N D:@SY'COPY &DATEI,&DATEI..VORHER,SAME'
@CON
@IF N : @IF N D : @SAVE 0
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP .FEHL, ON=(15)
/TCHNG OFLOW=NO

```

```

/WR-TEXT '&(#NEUBILD) '
/TCHNG OFLOW=ACK
/WR-TEXT ' '
/WR-TEXT 'DIE DATEI &DATEI WURDE REORGANISIERT.'
/WR-TEXT ' '
/WR-TEXT 'SPEICHERPLATZBEDARF VORHER:'
/FS &DATEI..VORHER
/WR-TEXT ' '
/ERASE &DATEI..VORHER
/WR-TEXT 'SPEICHERPLATZBEDARF JETZT:'
/FS &DATEI
/SKIP .EOP, OFF=(15)
/.FEHL REMARK
/WR-TEXT ' '
/WR-TEXT '***** FEHLER BEI DER DATEIBEARBEITUNG *****'
/WR-TEXT '** REORGANISATION WURDE NICHT DURCHGEFUEHRT ! **'
/WR-TEXT ' '
/.EOP STEP
/SETSW OFF=(4,5,15)
/SYSFILE SYSDTA=(PRIMARY)
/ENDP

```

```

% CVR1000 CONVERSION TERMINATED WITH: _____ (3)
% CVR1001          FAILURES:    0
% CVR1002          ERRORS   :    0
% CVR1003          WARNINGS:    0
% CVR1004          NOTES    :    0

```

- (1) Es werden die für den durchgeführten SDF-CONV-Lauf eingestellten Operandenwerte ausgegeben.
- (2) Ausgabe aller Kommandosätze der Eingabeprozedur. Wenn bei der Umwandlung ein Fehler auftrat, wird die entsprechende Meldung unter den betreffenden Kommandosatz geschrieben. Im Beispiel ist das nicht der Fall, d.h., SDF-CONV konnte jedem ISP-Kommando ein SDF-Kommando zuordnen sowie die Umwandlung in eine S-Prozedur ohne Probleme ausführen.
- (3) Der letzte Teil des Protokolls ist identisch mit der Ausgabe nach SYSOUT.

## Ausgabeprozedur SDF.SDFP.DO.REORG

Die umgewandelte Prozedur ist eine S-Prozedur.

```

/SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=NO, INTERRUPT-ALLOWED=YES,- _____ (1)
/   DATA-ESCAPE-CHAR=STD, SYSTEM-FILE-CONTEXT=OWN, DATA-ERROR-HANDLING=-
/   NO; BEGIN-PARAMETER-DECLARATION; DECLARE-PARAMETER NAME=DATEI, TYPE=-
/   ANY, INIT=*PROMPT; END-PARAMETER-DECLARATION
/SET-JV-LINK   JV-NAME=#NEUBILD
/MODIFY-JV   JV-CONTENTS=(JV-NAME=#NEUBILD), SET-VALUE=X'OC'
/MODIFY-JOB-SWITCHES  ON=(4,5), OFF=15
/ASSIGN-SYSDTA  TO=*SYSCMD
/WRITE-TEXT ' ' _____ (2)
/WRITE-TEXT '*****'
/WRITE-TEXT '***      REORGANISIEREN VON ISAM-DATEIEN      ***'
/WRITE-TEXT '*****'
/WRITE-TEXT ' '
/WRITE-TEXT 'NAME DER ZU REORGANISIERENDEN DATEI: '
/REMARK '&(DATEI)'
/START-PROGRAM FROM-FILE=$EDT
@GET '&(DATEI)' N
@CON
@IF N:@IF N D:@SY'COPY &(DATEI),&(DATEI).VORHER,SAME'
@CON
@IF N : @IF N D : @SAVE 0
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP-COMMANDS TO-LABEL=FEHL,IF=JOB-SWITCHES(ON=15)
/MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=NO-CONTROL
/WRITE-TEXT '&(JV('#NEUBILD'))'
/MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=USER-ACKNOWLEDGE
/WRITE-TEXT ' '
/WRITE-TEXT 'DIE DATEI &(DATEI) WURDE REORGANISIERT.'
/WRITE-TEXT ' '
/WRITE-TEXT 'SPEICHERPLATZBEDARF VORHER:'
/SHOW-FILE-ATTRIBUTES FILE-NAME=&(DATEI).VORHER,SELECT=BY-ATTRIBUTES
/WRITE-TEXT ' '
/DELETE-FILE FILE-NAME=&(DATEI).VORHER,SELECT=BY-ATTRIBUTES
/WRITE-TEXT 'SPEICHERPLATZBEDARF JETZT:'
/SHOW-FILE-ATTRIBUTES FILE-NAME=&(DATEI),SELECT=BY-ATTRIBUTES
/SKIP-COMMANDS TO-LABEL=EOP,IF=JOB-SWITCHES(OFF=15)
/.FEHL REMARK
/WRITE-TEXT ' '
/WRITE-TEXT '***** FEHLER BEI DER DATEIBEARBEITUNG *****'
/WRITE-TEXT '** REORGANISATION WURDE NICHT DURCHGEFUEHRT ! **'
/WRITE-TEXT ' '

```



```
/.EOP SET-JOB-STEP  
/MODIFY-JOB-SWITCHES OFF=(4,5,15)  
/ASSIGN-SYSDTA TO=*PRIMARY  
/IF-BLOCK-ERROR;EXIT-PROCEDURE ERROR=YES;END-IF;EXIT-PROCEDURE
```

- (1) Die aus dem Kommando PROC N,... generierten SDF-Kommandos für eine S-Prozedur werden in geblockter Schreibweise dargestellt, d.h., die Kommandos werden hintereinander geschrieben und mit je einem Semikolon voneinander getrennt.
- (2) Das Kommando WRITE-TEXT ist ein SDF-Kommando. Es braucht demzufolge nicht übersetzt zu werden. SDF-CONV erweitert das Eingabekommando auf sein volles Eingabeformat.

## Beispiel 2

Die Prozedur DO.LIB.INHALT listet alle Elemente einer Bibliothek auf. Die jeweils wichtigsten Elementdaten werden in eine Liste geschrieben. Die Übersetzung der Kommandos dieser Prozedur erfolgt mit Einschränkungen beim PRINT- und DO-Kommando. SDF-CONV meldet je eine NOTE. Das PRINT-Kommando wird nicht übersetzt, weil es als ersten Operanden einen Prozedurparameter enthält. Das DO-Kommando kann nicht in eine garantierte SDF-Form übersetzt werden, weil es kontextabhängig verschiedene Möglichkeiten der Übersetzung gibt. Beim DO-Kommando macht SDF-CONV Übersetzungsvorschläge in der Ausgabe prozedur.

### Eingabeprozedur DO.LIB.INHALT

```

/PROC N,(&LIBNAME=,&LISTOUT=,&PRINT=,&EINGABE),SUBDTA=&
/DCLJV #NEUBILD
/SETJV #NEUBILD,X'OC'
/WR-T '&(#NEUBILD)'
/WR-T ' '
/WR-T '*****'
/WR-T '****  Auflisten von Elementen einer Bibliothek  ****'
/WR-T '*****'
/WR-T ' '
/WR-T 'Name der Bibliothek                : LIBNAME'
/REMARK Eingabe &LIBNAME
/SYSFILE SYSOUT=*DUMMY
/FS &LIBNAME
/SYSFILE SYSOUT=( )
/SKIP .LINK
/STEP "Fehler beim FSTATUS"
/SYSFILE SYSOUT=( )
/WR-T ' '
/WR-T 'Fehlerhafte Eingabe !'
/WR-T 'Bibliothek &LIBNAME existiert nicht/kann nicht geoeffnet werden'
/SKIP .E
/.LINK REMARK Aufbereiten der Bibliothek
/WR-T 'Name der Ausgabe-Liste                : LISTOUT'
/WR-T 'Bei einer bereits bestehenden Ausgabe-Liste '
/WR-T 'werden die Daten ab Dateiende angefüegt. '
/WR-T 'Soll die Liste ausgedruckt werden (J/N) ? : PRINT'
/REMARK Eingabe &LISTOUT und &PRINT
/SETSW ON=(1)
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSLST=(&LISTOUT,EXTEND)
/SYSFILE SYSOUT=*DUMMY
/FILE &LIBNAME,LINK=LIB001
/EXEC $LMS

```

```
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END
/STEP
/SYSFILE SYSOUT=()
/SYSFILE SYSLST=()
/SYSFILE SYSDTA=()
/SETSW OFF=(1)
/REL LIB001
/WR-T ' '
/WR-T '**** Aufbereitung beendet.'
/WR-T '**** Das Ergebnis steht in der Liste &LISTOUT.'
/SKIP "Verzweigung abhaengig von der PRINT-Variablen" .&PRINT
/WR-T ' '
/.J REMARK Ausdrucken der Liste
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048
/WR-T 'Die Liste &LISTOUT wird ausgedruckt.'
/.N REMARK
/WR-T 'Eine weitere Bibliothek aufbereiten oder die Prozedur beenden ?'
/SKIP "Eingabe: A (aufbereiten) oder E (beenden)" .&EINGABE
/.A REMARK Erneuter Aufruf der Prozedur
/DO DO.LIB.INHALT
/STEP
/.E REMARK Ende
/WR-T ' '
/WR-T '**** Prozedur beendet'
/ENDP
```

**SDF-CONV-Ablauf**

```
/start-sdf-conv _____ (1)
% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0001 SDF-CONV VERSION 'V03.0B80' STARTED
% CVR0060 SDF-CONV VERSION 'V03.0B80' READY
//convert from-file=do.lib.inhalt,to-file=sdf.do.lib.inhalt,doc=*min — (2)
% CVR1000 CONVERSION TERMINATED WITH:
% CVR1001          FAILURES:    0
% CVR1002          ERRORS   :    0
% CVR1003          WARNINGS:    0
% CVR1004          NOTES    :    2
//end _____ (3)
```

- (1) Start des Dienstprogramms SDF-CONV.
- (2) In der CONVERT-Anweisung wird die Ein- und Ausgabeprozedur zugewiesen sowie die Dokumentationsmenge bestimmt.  
Alle anderen Operanden der CONVERT-Anweisung behalten den Standardwert.  
Nach erfolgter Abarbeitung meldet SDF-CONV, dass die Übersetzung mit zwei Anmerkungen (NOTE) beendet wurde. Diese Übersicht wird nach SYSOUT ausgegeben.
- (3) Mit der Anweisung END wird SDF-CONV beendet.

**Ablaufprotokoll (SYSLST)**

```

***** BS2000 COMMAND CONVERTER : SDF-CONV V3 ***** page 1

% CVR1010 OPTIONS IN EFFECT: _____ (1)
% CVR1011 FROM-FILE = DO.LIB.INHALT
% CVR1012 EXPECT-CONTINUATION = *NEW-MODE
% CVR1013 PARAMETER-LINES = *CONVERT
% CVR1014 TO-FILE = SDF.DO.LIB.INHALT
% CVR1015 PRODUCE-CONTINUATION = *NEW-MODE
% CVR1016 REPLACE-OLD-FILE = *YES
% CVR1017 SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018 GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019 UNCHANGED-CMD = *NONE
% CVR1020 DOCUMENTATION = *MINIMUM
% CVR1021 TARGET-VERSION = *V10
% CVR1022 OUTPUT-FORM = *STD
% CVR1023 PROCEDURE-FORMAT = *SAME(BLOCK-INPUT=*NOT-ALLOWED)

```

```

***** BS2000 COMMAND CONVERTER : SDF-CONV V3 ***** page 2

/PROC N,(&LIBNAME=,&LISTOUT=,&PRINT=,&EINGABE),SUBDTA=& _____ (2)
/DCLJV #NEUBILD
/SETJV #NEUBILD,X'0C'
/WR-T '&(#NEUBILD)'
/WR-T ' '
/WR-T '*****'
/WR-T '**** Auflisten von Elementen einer Bibliothek ****'
/WR-T '*****'
/WR-T ' '
/WR-T 'Name der Bibliothek : LIBNAME'
/REMARK Eingabe &LIBNAME
/SYSFILE SYSOUT=*DUMMY
/FS &LIBNAME
/SYSFILE SYSOUT=()
/SKIP .LINK
/STEP "Fehler beim FSTATUS"
/SYSFILE SYSOUT=()
/WR-T ' '
/WR-T 'Fehlerhafte Eingabe !'
/WR-T 'Bibliothek &LIBNAME existiert nicht/kann nicht geoeffnet werden'
/SKIP .E
/.LINK REMARK Aufbereiten der Bibliothek
/WR-T 'Name der Ausgabe-Liste : LISTOUT'
/WR-T 'Bei einer bereits bestehenden Ausgabe-Liste '
/WR-T 'werden die Daten ab Dateiende angefuegt. '
/WR-T 'Soll die Liste ausgedruckt werden (J/N) ? : PRINT'

```

```

/REMARK Eingabe &LISTOUT und &PRINT
/SETSW ON=(1)
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSLST=(&LISTOUT,EXTEND)
/SYSFILE SYSOUT=*DUMMY
/FILE &LIBNAME,LINK=LIB001
/EXEC $LMS
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END
/STEP
/SYSFILE SYSOUT=()
/SYSFILE SYSLST=()
/SYSFILE SYSDTA=()
/SETSW OFF=(1)
/REL LIB001
/WR-T ' '
/WR-T '**** Aufbereitung beendet.'
/WR-T '**** Das Ergebnis steht in der Liste &LISTOUT.'
/SKIP "Verzweigung abhaengig von der PRINT-Variablen" .&PRINT
/WR-T ' '
/.J REMARK Ausdrucken der Liste
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048
% CVRF040 INPUT COMMAND NOT CONVERTED _____ (3)
/WR-T 'Die Liste &LISTOUT wird ausgedruckt.'
/.N REMARK
/WR-T 'Eine weitere Bibliothek aufbereiten oder die Prozedur beenden ?'
/SKIP "Eingabe: A (aufbereiten) oder E (beenden)" .&EINGABE
/.A REMARK Erneuter Aufruf der Prozedur
/DO DO.LIB.INHALT
% CVRF061 COMMAND CAN NEVER BE CONVERTED TO A SECURED SDF FORM; ORIGINAL
INPUT REMAINS VALID _____ (4)
/STEP
/.E REMARK Ende
/WR-T ' '
/WR-T '**** Prozedur beendet'
/ENDP

% CVR1000 CONVERSION TERMINATED WITH: _____ (5)
% CVR1001 FAILURES: 0
% CVR1002 ERRORS : 0
% CVR1003 WARNINGS: 0
% CVR1004 NOTES : 3

```

- (1) Es werden die für den durchgeführten SDF-CONV-Lauf eingestellten Operandenwerte ausgegeben.
- (2) Ausgabe aller Kommandosätze der Eingabeprozedur.
- (3) SDF-CONV meldet eine NOTE, da das PRINT-Kommando nicht übersetzt werden konnte. (Siehe [Abschnitt „Einschränkungen“ auf Seite 252](#): beim PRINT-Kommando darf der erste Stellungsoperand kein Prozedurparameter sein.)
- (4) SDF-CONV meldet eine NOTE, da das DO-Kommando nicht übersetzt werden konnte. (Siehe [Abschnitt „Einschränkungen“ auf Seite 252](#): die Übersetzung des DO-Kommandos ist kontextabhängig.)
- (5) Der letzte Teil des Protokolls ist identisch mit der Ausgabe nach SYSOUT.

### Ausgabeprozedur SDF.DO.LIB.INHALT

```

/BEGIN-PROCEDURE LOGGING=NO,PARAMETERS=YES(PROCEDURE-PARAMETERS=(
/&LIBNAME=,-
/&LISTOUT=,-
/&PRINT=,-
/&EINGABE),ESCAPE-CHARACTER='&')
/SET-JV-LINK JV-NAME=#NEUBILD
/MODIFY-JV JV-IDENTIFICATION=JV-NAME(JV-NAME=#NEUBILD),VALUE=X'OC'
/WRITE-TEXT '&(#NEUBILD)' _____ (1)
/WRITE-TEXT ' '
/WRITE-TEXT '*****'
/WRITE-TEXT '**** Auflisten von Elementen einer Bibliothek ****'
/WRITE-TEXT '*****'
/WRITE-TEXT ' '
/WRITE-TEXT 'Name der Bibliothek : LIBNAME'
/REMARK Eingabe &LIBNAME
/ASSIGN-SYSOUT TO=*DUMMY
/SHOW-FILE-ATTRIBUTES FILE-NAME=&LIBNAME,SELECT=BY-ATTRIBUTES
/ASSIGN-SYSOUT TO=*PRIMARY
/SKIP-COMMANDS TO-LABEL=LINK
/"*****" REMARK '/STEP "Fehler beim FSTATUS"' _____ (2)
/SET-JOB-STEP
/ASSIGN-SYSOUT TO=*PRIMARY
/WRITE-TEXT ' '
/WRITE-TEXT 'Fehlerhafte Eingabe !'
/WRITE-TEXT 'Bibliothek &LIBNAME existiert nicht/kann nicht geoeffnet w-
/erden'
/SKIP-COMMANDS TO-LABEL=E
/.LINK REMARK Aufbereiten der Bibliothek
/WRITE-TEXT 'Name der Ausgabe-Liste : LISTOUT'
/WRITE-TEXT 'Bei einer bereits bestehenden Ausgabe-Liste '
/WRITE-TEXT 'werden die Daten ab Dateieinde angefuegt. -

```

```

/ '
/WRITE-TEXT 'Soll die Liste ausgedruckt werden (J/N) ?      : PRINT'
/REMARK Eingabe &LISTOUT und &PRINT
/MODIFY-JOB-SWITCHES ON=1
/ASSIGN-SYSDTA TO=*SYSCMD
/ASSIGN-SYSLST TO=&LISTOUT,OPEN-MODE=EXTEND
/ASSIGN-SYSOUT TO=*DUMMY
/"*****" REMARK '/CREATE-FILE FILE-NAME=&LIBNAME,SUPPRESS-ERRORS=-
/*FILE-EXISTING'
/"*****" REMARK 'SET-FILE-LINK LINK-NAME=LIB001,FILE-NAME=&LIBNAME'
/ADD-FILE-LINK LINK-NAME=LIB001,FILE-NAME=&LIBNAME,SUPPORT=*DISK
/START-PROGRAM FROM-FILE=$LMS
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END
/SET-JOB-STEP
/ASSIGN-SYSOUT TO=*PRIMARY
/ASSIGN-SYSLST TO=*PRIMARY
/ASSIGN-SYSDTA TO=*PRIMARY
/MODIFY-JOB-SWITCHES OFF=1
/REMOVE-FILE-LINK LINK-NAME=LIB001
/WRITE-TEXT ' '
/WRITE-TEXT '**** Aufbereitung beendet.'
/WRITE-TEXT '**** Das Ergebnis steht in der Liste &LISTOUT.'
/"*****" REMARK '/SKIP "Verzweigung abhaengig von der PRINT-Variablen" - (3)
/.&PRINT'
/SKIP-COMMANDS TO-LABEL=&PRINT
/WRITE-TEXT ' '
/.J REMARK Ausdrucken der Liste
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048 _____ (4)
/WRITE-TEXT 'Die Liste &LISTOUT wird ausgedruckt.'
/.N REMARK
/WRITE-TEXT 'Eine weitere Bibliothek aufbereiten oder die Prozedur beenden ?'
/"*****" REMARK '/SKIP "Eingabe: A (aufbereiten) oder E (beenden)" .&EI- (5)
/NGABE'
/SKIP-COMMANDS TO-LABEL=&EINGABE
/.A REMARK Erneuter Aufruf der Prozedur
/"*****" REMARK '/CALL-PROCEDURE NAME=DO.LIB.INHALT' _____ (6)
/DO DO.LIB.INHALT
/SET-JOB-STEP
/.E REMARK Ende
/WRITE-TEXT ' '
/WRITE-TEXT '**** Prozedur beendet'
/EXIT-PROCEDURE

```



- (1) Das Kommando WRITE-TEXT ist ein SDF-Kommando. Es braucht demzufolge nicht übersetzt zu werden. SDF-CONV erweitert das Eingabekommando auf sein volles Eingabeformat.
- (2) Der Kommandosatz mit dem Inline-Kommentar wird in eine Kommentarzeile geschrieben, die vor der Kommandozeile mit der Übersetzung steht. Dadurch ist gewährleistet, dass die Information des Inline-Kommentars bei der Übersetzung nicht verloren geht (Voraussetzung: DOCUMENTATION=\*MIN/\*MAX).
- (3) siehe Punkt (2).
- (4) Der Kommandosatz mit dem nicht übersetzten PRINT-Kommando wird aus der Eingabeprozedur übernommen.
- (5) siehe Punkt (2).
- (6) Der Kommandosatz mit dem nicht übersetzten DO-Kommando wird aus der Eingabeprozedur übernommen. SDF-CONV macht jedoch zusätzlich einen Vorschlag, wie die Übersetzung dieses DO-Kommandos aussehen könnte. Der Vorschlag wird als Kommentarzeile vor den nicht übersetzten Kommandosatz geschrieben (Voraussetzung: DOCUMENTATION=\*MIN/\*MAX).

### Beispiel 3

Die Prozedur PROC.ASSXT ist eine kleine Hilfsprozedur, die bei der häufigen Nutzung des ASSEMBH-Assemblers ein schnelleres Abarbeiten (Zuweisung von Systemdateien, Aufruf des Assemblers) ermöglichen soll. Sie ist in SDF-Kommandosprache geschrieben. Die Umwandlung dieser Nicht-S-Prozedur in eine S-Prozedur erfolgt ohne Einschränkungen.

#### Eingabeprozedur PROC.ASSXT

```
/BEGIN-PROC A,P-P=(&FILE),ESC-CHAR='&'
/REMARK **** Aufruf des ASSEMBH ****
/DEL-FI LST.ASSEMBH
/SET-JOB-STEP
/ASS-SYSDTA *SYSCMD
/DEL-SYS-FI "von frueheren ASSEMBH-Aufrufen" *OMF
/START-PROG $ASSEMBH
//COMPILE SOURCE=&FILE,LISTING=PAR(OUTPUT=LST.ASSEMBH),-
//TEST-SUPPORT=YES
//END
/SET-JOB-STEP
/ASS-SYSDTA *P
/END-PROC
```

**SDF-CONV-Ablauf**

```
/start-sdf-conv _____ (1)
% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0001 SDF-CONV VERSION 'V03.0B80' STARTED
% CVR0060 SDF-CONV VERSION 'V03.0B80' READY
//convert from-file=proc.assxt,to-file=sdfp.proc.assxt,doc=*min,- _____ (2)
//procedure-format=*s-proc(prog-inp=*data)
% CVR1000 CONVERSION TERMINATED WITH:
% CVR1001          FAILURES:    0
% CVR1002          ERRORS   :    0
% CVR1003          WARNINGS:    0
% CVR1004          NOTES    :    0
//end _____ (3)
```

- (1) Start des Dienstprogramms SDF-CONV.
- (2) In der CONVERT-Anweisung wird die Ein- und Ausgabeprozedur zugewiesen sowie die Dokumentationsmenge bestimmt. Die Prozedur soll in eine S-Prozedur umgewandelt werden. Alle anderen Operanden der CONVERT-Anweisung behalten den Standardwert. Nach erfolgter Abarbeitung meldet SDF-CONV, dass die Umwandlung ohne Warnungen oder Fehler durchgeführt wurde. Diese Übersicht wird nach SYSOUT ausgegeben.
- (3) Mit der Anweisung END wird SDF-CONV beendet.

**Ablaufprotokoll (SYSLST)**

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 1

```
% CVR1010 OPTIONS IN EFFECT: _____ (1)
% CVR1011 FROM-FILE = PROC.ASSXT
% CVR1012 EXPECT-CONTINUATION = *NEW-MODE
% CVR1013 PARAMETER-LINES = *CONVERT
% CVR1014 TO-FILE = SDFP.PROC.ASSXT
% CVR1015 PRODUCE-CONTINUATION = *NEW-MODE
% CVR1016 REPLACE-OLD-FILE = *YES
% CVR1017 SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018 GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019 UNCHANGED-CMD = *NONE
% CVR1020 DOCUMENTATION = *MINIMUM
% CVR1021 TARGET-VERSION = *OSD-V2
% CVR1022 OUTPUT-FORM = *STD
% CVR1023 PROCEDURE-FORMAT = *S-PROCEDURE(CMD-
FORMAT=*SDF, PROGRAM-INPUT=*DATA(EXCEPT-AFTER-CMD=*NONE, WARNING=*NO), BLOCK-
INPUT=*NOT-ALLOWED)
```

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 2

```
/BEGIN-PROC A,P-P=(&FILE),ESC-CHAR='&' _____ (2)
/REMARK **** Aufruf des ASSEMBH ****
/DEL-FI LST.ASEMBH
/SET-JOB-STEP
/ASS-SYSDTA *SYSCMD
/DEL-SYS-FI "von fruheren ASSEMBH-Aufrufen" *OMF
/START-PROG $ASSEMBH
//COMPILE SOURCE=&FILE, LISTING=PAR(OUTPUT=LST.ASEMBH),-
//TEST-SUPPORT=YES
//END
/SET-JOB-STEP
/ASS-SYSDTA *P
/END-PROC
```

```
% CVR1000 CONVERSION TERMINATED WITH: _____ (3)
% CVR1001 FAILURES: 0
% CVR1002 ERRORS : 0
% CVR1003 WARNINGS: 0
% CVR1004 NOTES : 0
```

- (1) Es werden die für den durchgeführten SDF-CONV-Lauf eingestellten Operandenwerte ausgegeben.
- (2) Ausgabe aller Kommandosätze der Eingabeprozedur.

- (3) Der letzte Teil des Protokolls ist identisch mit der Ausgabe nach SYSOUT.

### Ausgabeprozedur SDFP.PROC.ASSXT

```

/SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=YES, INTERRUPT-ALLOWED=YES, -
/      DATA-ESCAPE-CHAR=STD, SYSTEM-FILE-CONTEXT=OWN, DATA-ERROR-HANDLING=-
/      NO
/BEGIN-PARAMETER-DECLARATION
/DECLARE-PARAMETER NAME=FILE, TYPE=ANY, INIT=*PROMPT
/END-PARAMETER-DECLARATION
/REMARK '**** AUFRUF DES ASSEMBH ****'
/DELETE-FILE LST.ASEMBH
/SET-JOB-STEP
/ASSIGN-SYSDTA TO=*SYSCMD
/"*****" REMARK '/DEL-SYS-FI "von frueheren ASSEMBH-Aufrufen" *OMF' —— (1)
/DELETE-SYSTEM-FILE "von frueheren ASSEMBH-Aufrufen" *OMF
/START-PROGRAM $ASSEMBH
//COMPILE SOURCE=&(FILE), LISTING=PAR(OUTPUT=LST.ASEMBH), TEST-SUPPORT=-
//      YES
//END
/SET-JOB-STEP
/ASSIGN-SYSDTA TO=*P
/IF-BLOCK-ERROR; END-IF
/END-PROCEDURE

```

- (1) Der Kommandosatz mit dem Inline-Kommentar wird in eine Kommentarzeile geschrieben, die vor der Kommandozeile mit der Übersetzung steht. Dadurch ist gewährleistet, dass die Information des Inline-Kommentars bei der Übersetzung nicht verloren geht (Voraussetzung: DOCUMENTATION=\*MIN/\*MAX).

## 13.7 Zuordnung von ISP- zu SDF-Kommandos

### 13.7.1 Benutzerkommandos

ISP-Kommando	SDF-Kommando
ABEND	EXIT-JOB
ABORT	CANCEL-PROCEDURE
ADD-ISAM-POOL-LINK	ADD-ISAM-POOL-LINK
APPLICATION	SET-DCAM-APPLICATION-LINK
AUDIT	RESUME-HARDWARE-AUDIT
AUDIT	SHOW-HARDWARE-AUDIT
AUDIT	START-HARDWARE-AUDIT
AUDIT	STOP-HARDWARE-AUDIT
BCNTRL	MODIFY-MSG-OPTIONS
BREAK	HOLD-PROGRAM
CALL	CALL-PROCEDURE
CANCEL	CANCEL-JOB
CATALOG	CREATE-FILE
CATALOG	CREATE-FILE-GROUP
CATALOG	MODIFY-FILE-ATTRIBUTES
CATALOG	MODIFY-FILE-GROUP-ATTRIBUTES
CATJV	CREATE-JV
CATJV	MODIFY-JV-ATTRIBUTES
CHANGE	CHANGE-FILE-LINK
CONNECTION	REMOVE-DCAM-CONNECTION-LINK
CONNECTION	SET-DCAM-CONNECTION-LINK
COPY	COPY-FILE
CREATE-ISAM-POOL	CREATE-ISAM-POOL
DCLJV	SET-JV-LINK
DELETE-ISAM-POOL	DELETE-ISAM-POOL
DELON	REMOVE-CJC-ACTION
DROP	UNLOCK-FILE-LINK
ENDON	END-CJC-ACTION

Tabelle 15: Zuordnung von ISP- zu SDF-Benutzerkommandos

<b>ISP-Kommando</b>	<b>SDF-Kommando</b>
ENDP	END-PROCEDURE
ENDP	EXIT-PROCEDURE
ENTER	ENTER-JOB
EOF	EOF
ERAJV	DELETE-JV
ERASE	DELETE-FILE
ERASE	DELETE-FILE-GENERATION
ERASE	DELETE-FILE-GROUP
ERASE	DELETE-SYSTEM-FILE
ERASE	EXPORT-FILE
ESCAPE	HOLD-PROCEDURE
EXECUTE	START-PROGRAM
FILE	CREATE-FILE
FILE	CREATE-FILE-GENERATION
FILE	CREATE-TAPE-SET
FILE	DELETE-TAPE-SET
FILE	DELETE-TAPE-SET
FILE	EXTEND-TAPE-SET
FILE	IMPORT-FILE
FILE	MODIFY-FILE-ATTRIBUTES
FILE	MODIFY-FILE-GENERATION-SUPPORT
FILE	SET-FILE-LINK
INTR	SEND-MESSAGE
LOAD	LOAD-PROGRAM
LOGOFF	EXIT-JOB
LOGOFF	LOGOFF
LOGON	SET-LOGON-PARAMETERS
MODIFY-JOB	MODIFY-JOB
MODIFY-JV-CONDITIONALLY	MODIFY-JV-CONDITIONALLY
MODIFY-MSG-ATTRIBUTES	MODIFY-MSG-ATTRIBUTES
MRSSTA	SHOW-MRS-CONNECTIONS
MSGCONTROL	MODIFY-MSG-FILE-ASSIGNMENT

Tabelle 15: Zuordnung von ISP- zu SDF-Benutzerkommandos

ISP-Kommando	SDF-Kommando
ON	ADD-CJC-ACTION
OPTION	MODIFY-JOB-OPTIONS
OPTION	MODIFY-TEST-OPTIONS
PARAMETER	PARAMETER
PASSWORD	ADD-PASSWORD
PASSWORD	REMOVE-PASSWORD
PAUSE	SEND-MESSAGE
PRINT	PRINT-FILE
PRINT	WRITE-SPOOL-TAPE
PRIORITY	CHANGE-TASK-PRIORITY
PRIORITY	MODIFY-JOB
PROCEDURE	BEGIN-PROCEDURE
PSWORD	MODIFY-USER-PROTECTION
PUNCH	WRITE-DISKETTE
PUNCH	WRITE-POOLER-TAPE
RDIR	REDIRECT-REMOTE-OUTPUT
RDTFT	SHOW-FILE-LINK
RELEASE	REMOVE-FILE-LINK
RELEASE-SUBSYSTEM-SPACE	RELEASE-SUBSYSTEM-SPACE
REMARK	REMARK
REMOVE-ISAM-POOL-LINK	REMOVE-ISAM-POOL-LINK
REMOVE-JV-LINK	REMOVE-JV-LINK
REMOVE-SPOOL-CHARACTER-SET	REMOVE-SPOOL-CHARACTER-SET
REMOVE-SPOOL-DEVICE	REMOVE-SPOOL-DEVICE
REMOVE-SPOOL-FORM	REMOVE-SPOOL-FORM
RESTART	RESTART-PROGRAM
RESUME	RESUME-PROGRAM
RFAEND	REMOVE-RFA-CONNECTION
RFASTART	SET-RFA-CONNECTION
RTI	RESUME-PROCEDURE
SDVC	MODIFY-PRINTER-OUTPUT-STATUS
SDVC	START-PRINTER-OUTPUT

Tabelle 15: Zuordnung von ISP- zu SDF-Benutzerkommandos



<b>ISP-Kommando</b>	<b>SDF-Kommando</b>
SDVC	STOP-PRINTER-OUTPUT
SECURE-RESOURCE-ALLOCATION	SECURE-RESOURCE-ALLOCATION
SHOW-DISK-DEFAULTS	SHOW-DISK-DEFAULTS
SHOW-DISK-STATUS	SHOW-DISK-STATUS
SHOW-FILE	SHOW-FILE
SHOW-JOB-CLASS	SHOW-JOB-CLASS
SHOW-JV-LINK	SHOW-JV-LINK
SHOW-MOUNT-PARAMETER	SHOW-MOUNT-PARAMETER
SHOW-MSG-DEFAULTS	SHOW-MSG-FILE-ASSIGNMENT
SHOW-PCS-OPTION	SHOW-PCS-OPTION
SHOW-RESOURCE-ALLOCATION	SHOW-RESOURCE-ALLOCATION
SHOW-SPOOL-CHARACTER-SETS	SHOW-SPOOL-CHARACTER-SETS
SHOW-SPOOL-DEVICES	SHOW-SPOOL-DEVICES
SHOW-SPOOL-FORMS	SHOW-SPOOL-FORMS
SHOW-SPOOL-PARAMETERS	SHOW-SPOOL-PARAMETERS
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS
SHOW-TAPE-STATUS	SHOW-TAPE-STATUS
SHOW-USER-ATTRIBUTES	SHOW-USER-ATTRIBUTES
SKIP	SKIP-COMMANDS
SKIPJV	SKIP-COMMANDS
SKIPUS	SKIP-COMMANDS
SPARAM	MODIFY-SPOOL-OUT-OPTIONS
SQUC	HOLD-SPOOL-OUT
SQUC	RESUME-SPOOL-OUT
STAJV	SHOW-JV-ATTRIBUTES
STAM	SHOW-MASTER-CATALOG-ENTRY
STATUS	SHOW-JOB-STATUS
STATUS	SHOW-SYSTEM-STATUS
STATUS	SHOW-USER-STATUS
STEP	SET-JOB-STEP
SYSFILE	ASSIGN-SYSDTA
SYSFILE	ASSIGN-SYSIPT

Tabelle 15: Zuordnung von ISP- zu SDF-Benutzerkommandos

<b>ISP-Kommando</b>	<b>SDF-Kommando</b>
SYSFILE	ASSIGN-SYSLST
SYSFILE	ASSIGN-SYSOPT
SYSFILE	ASSIGN-SYSOUT
SYSFILE	REMOVE-TASKLIB
SYSFILE	SET-TASKLIB
SYSTATUS	SHOW-SYSTEM-FILE-ASSIGNMENTS
TCHNG	MODIFY-TERMINAL-OPTIONS
TYPE	SEND-MESSAGE
VERIFY	CHECK-FILE-CONSISTENCY
VERIFY	REMOVE-FILE-ALLOCATION-LOCKS
VERIFY	REPAIR-DISK-FILES
WAIT	WAIT-EVENT
WHEN	WAIT-EVENT
WRITE-ACCOUNTING-RECORD	WRITE-ACCOUNTING-RECORD

Tabelle 15: Zuordnung von ISP- zu SDF-Benutzerkommandos

### 13.7.2 Kommandos für die Systemverwaltung

ISP-Kommando	SDF-Kommando
ADD-SPOOL-DEVICE	ADD-SPOOL-DEVICE
ADD-SPOOL-FORM	ADD-SPOOL-FORM
ADD-SUBSYSTEM	ADD-SUBSYSTEM
AUDIT	HOLD-HARDWARE-AUDIT
AUDIT	RESUME-HARDWARE-AUDIT
AUDIT	SHOW-HARDWARE-AUDIT
AUDIT	START-HARDWARE-AUDIT
AUDIT	STOP-HARDWARE-AUDIT
BIAS	MODIFY-SYSTEM-BIAS
CATALOG	CREATE-FILE
CATALOG	CREATE-FILE-GENERATION
CATALOG	CREATE-FILE-GROUP
CATALOG	MODIFY-FILE-ATTRIBUTES
CATALOG	MODIFY-FILE-GENERATION-SUPPORT
CATALOG	MODIFY-FILE-GROUP-ATTRIBUTES
CATEGORY	MODIFY-TASK-CATEGERIES
CATM	ADD-MASTER-CATALOG-ENTRY
CATM	MODIFY-MASTER-CATALOG-ENTRY
CHANGE-ACCOUNTING-FILE	CHANGE-ACCOUNTING-FILE
CHANGE-CONSLOG	CHANGE-CONSLOG-FILE
CHANGE-SERSLOG	CHANGE-SERSLOG-FILE
ERAM	REMOVE-MASTER-CATALOG-ENTRY
ERASE	DELETE-FILE
ERASE	DELETE-FILE-GENERATION
ERASE	DELETE-FILE-GROUP
ERASE	EXPORT-FILE
EXCAT	EXPORT-CATALOG
EXCAT	EXPORT-PUBLIC-VOLUME-SET
EXCAT	FORCE-CATALOG-EXPORT
FILE	CREATE-FILE
FILE	CREATE-FILE-GENERATION

Tabelle 16: Zuordnung von ISP- zu SDF-Kommandos für die Systemverwaltung

ISP-Kommando	SDF-Kommando
FILE	CREATE-TAPE-SET, EXTEND-TAPE-SET
FILE	IMPORT-FILE
FILE	MODIFY-FILE-ATTRIBUTES
FILE	MODIFY-FILE-GENERATION-SUPPORT
FILE	SET-FILE-LINK
HOLD-JOB	HOLD-JOB
HOLD-JOB-CLASS	HOLD-JOB-CLASS
HOLD-PCS	HOLD-PCS
HOLD-SS	HOLD-SUBSYSTEM
IMCAT	IMPORT-PUBLIC-VOLUME-SET
IMPORT	CHECK-IMPORT-DISK-FILE
IMPORT	IMPORT-FILE
JOIN	ADD-USER
JOIN	MODIFY-USER-ATTRIBUTES
LOADAID	LOADAID
MODIFY-ACCOUNTING-PARAMETERS	MODIFY-ACCOUNTING-PARAMETERS
MODIFY-JOB-CLASS	MODIFY-JOB-CLASS
MODIFY-JOB-STREAM	MODIFY-JOB-STREAM
MODIFY-PCS-OPTION	MODIFY-PCS-OPTIONS
MODIFY-SPOOL-FORM	MODIFY-SPOOL-FORM
MODIFY-SPOOL-PARAMETERS	MODIFY-SPOOL-PARAMETERS
MSGCONTROL	MODIFY-MSG-FILE-ASSIGNMENT
MSGCONTROL	RESET-MSG-FILE-ASSIGNMENT
NCHOLD	HOLD-TASK
NCREL	RESUME-TASK
RDIR	REDIRECT-REMOTE-OUTPUT
RELEASE-JOB	RESUME-JOB
RELEASE-JOB-CLASS	RESUME-JOB-CLASS
RELEASE-JOB-STREAM	RESUME-JOB-STREAM
REMOVE-SPOOL-CHARACTER-SET	REMOVE-SPOOL-CHARACTER-SET
REMOVE-SPOOL-DEVICE	REMOVE-SPOOL-DEVICE
REMOVE-SPOOL-FORM	REMOVE-SPOOL-FORM

Tabelle 16: Zuordnung von ISP- zu SDF-Kommandos für die Systemverwaltung

<b>ISP-Kommando</b>	<b>SDF-Kommando</b>
RESUME-PCS	RESUME-PCS
RESUME-SS	RESUME-SUBSYSTEM
RFD	START-DISKETTE-INPUT
RFD	STOP-DISKETTE-INPUT
SDVC	MODIFY-DISKETTE-OUTPUT-STATUS
SDVC	MODIFY-PRINTER-OUTPUT-STATUS
SDVC	MODIFY-TAPE-OUTPUT-STATUS
SDVC	START-DISKETTE-OUTPUT
SDVC	START-PRINTER-OUTPUT
SDVC	START-TAPE-OUTPUT
SDVC	START-TAPE-REPLAY
SDVC	STOP-DISKETTE-OUTPUT
SDVC	STOP-PRINTER-OUTPUT
SDVC	STOP-TAPE-OUTPUT
SDVC	STOP-TAPE-REPLAY
SET-DSSM-OPTIONS	SET-DSSM-OPTIONS
SET-PUBSET-ATTRIBUTES	SET-PUBSET-ATTRIBUTES
SET-REPLUG-READ-MARK	SET-REPLUG-READ-MARK
SEVER	LOCK-USER
SEVER	REMOVE-USER
SEVER	UNLOCK-USER
SHARE	ADD-SHARED-PROGRAM
SHOW-ACCOUNTING-STATUS	SHOW-ACCOUNTING-STATUS
SHOW-CONSLOG	SHOW-CONSLOG-ATTRIBUTES
SHOW-DISK-STATUS	SHOW-DISK-STATUS
SHOW-JOB-STREAM	SHOW-JOB-STREAM
SHOW-PCS-OPTION	SHOW-PCS-OPTIONS
SHOW-PUBSET-ATTRIBUTES	SHOW-PUBSET-ATTRIBUTES
SHOW-RESOURCE-REQUESTS	SHOW-RESOURCE-REQUESTS
SHOW-SERSLOG	SHOW-SERSLOG-STATUS
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS
SHOW-TRACE-STATUS	SHOW-TRACE-STATUS

Tabelle 16: Zuordnung von ISP- zu SDF-Kommandos für die Systemverwaltung

<b>ISP-Kommando</b>	<b>SDF-Kommando</b>
SHOW-USER-ATTRIBUTES	SHOW-USER-ATTRIBUTES
SPMGT	SET-SPACE-SATURATION-LEVEL
SQUC	HOLD-SPOOLOUT
STAM	SHOW-MASTER-CATALOG-ENTRY
START-ACCOUNTING	START-ACCOUNTING
START-JOB-STREAM	START-JOB-STREAM
START-PCS	START-PCS
START-SERSLOG	START-SERSLOG
START-SS	START-SUBSYSTEM
START-TRACE	START-TRACE
STATUS	SHOW-JOB-STATUS
STATUS	SHOW-SYSTEM-STATUS
STATUS	SHOW-USER-STATUS
STOP-ACCOUNTING	STOP-ACCOUNTING
STOP-JOB-STREAM	STOP-JOB-STREAM
STOP-PCS	STOP-PCS
STOP-SERSLOG	STOP-SERSLOG
STOP-SS	STOP-SUBSYSTEM
STOP-TRACE	STOP-TRACE

Tabelle 16: Zuordnung von ISP- zu SDF-Kommandos für die Systemverwaltung

## 13.8 Übersetzung von ISP- und alten SDF-Kommandos durch SDF-CONV

Die zu übersetzenden Kommandos sind in Benutzerkommandos und Kommandos für die Systemverwaltung unterteilt. Die Tabellen listen in alphabetischer Reihenfolge die zu übersetzenden Kommandos und die von SDF-CONV erzeugten SDF-Kommandos auf. In den Fällen, in denen durch SDF-CONV keine Übersetzung erfolgt, wird eine kurze Begründung angegeben. Einige Operandenkonfigurationen können nicht übersetzt werden. Die betroffenen Kommandos erhalten einen Verweis auf die im [Abschnitt „Einschränkungen“ auf Seite 252](#)) beschriebenen Ausnahmen.

### 13.8.1 Benutzerkommandos

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
ABEND	EXIT-JOB	
ABORT	CANCEL-PROCEDURE	
ACCOUNT	-----	keine eindeutige Zuordnung möglich
APPLICATION LINK=... (als einziger Operand) and. Konfigurationen	REMOVE-DCAM-APPLICATION-LINK SET-DCAM-APPLICATION-LINK	
AUDIT	-----	Übersetzung ist laufzeitabhängig
BCNTRL	MODIFY-MSG-OPTIONS	
BREAK	HOLD-PROGRAM	
CALL	CALL-PROCEDURE	
CANCEL	CANCEL-JOB	(s. S. <a href="#">263</a> )
CATALOG	CREATE-FILE CREATE-FILE-GENERATION CREATE-FILE-GROUP IMPORT-FILE MODIFY-FILE-ATTRIBUTES MODIFY-FILE-GROUP-ATTRIBUTES	(s. S. <a href="#">266</a> )
CATJV STATE=NEW STATE=UPDATE	CREATE-JV MODIFY-JV-ATTRIBUTES	(s. S. <a href="#">263</a> )
CHANGE	CHANGE-FILE-LINK	

Tabelle 17: Übersetzung von Benutzerkommandos durch SDF-CONV

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
CONNECTION LINK=... (als einziger Operand) and. Konfigurationen	REMOVE-DCAM-CONNECTION-LINK  SET-DCAM-CONNECTION-LINK	
COPY	COPY-FILE	(s. S. 263)
DATA	-----	kein SDF-Äquivalent
DCLJV	SET-JV-LINK	
DELON	REMOVE-CJC-ACTION	(s. S. 263)
DIAG	-----	kein ISP-Kommando
DO	-----	Übersetzung ist kontextabhängig; keine automatische Übersetzung, aber Übersetzungsvorschlag (CALL- PROCEDURE) in der Ausgabeproze- dur (s. S. 256)
DROP	UNLOCK-FILE-LINK	
END	-----	kein SDF-Äquivalent
ENDON	END-CJC-ACTION	
ENDP	EXIT-PROCEDURE	
ENTER	ENTER-JOB	(s. S. 263)
EOF	EOF	
ERAJV	DELETE-JV	(s. S. 263)
ERASE nur Dateiname *_Datei oder *SYSdatei POS=... VOLUME=...	DELETE-FILE DELETE-SYSTEM-FILE DELETE-FILE-GENERATION EXPORT-FILE	(s. S. 263)
ESCAPE (ohne Operanden)	HOLD-PROCEDURE	
EXECUTE	START-PROGRAM	(s. S. 263)
FILE	CREATE-FILE CREATE-FILE-GENERATION CREATE-TAPE-SET DELETE-TAPE-SET EXTEND-TAPE-SET IMPORT-FILE MODIFY-FILE-ATTRIBUTES MODIFY-FILE-GENERATION- SUPPORT ADD-FILE-LINK	(s. S. 258)

Tabelle 17: Übersetzung von Benutzerkommandos durch SDF-CONV



Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
FSTATUS VTOC=YES	SHOW-FILE-ATTRIBUTES IMPORT-FILE SUPPORT=BY-FILE-NAME	(s. S. 263)
GETJV	SHOW-JV	(s. S. 263)
GETUS	SHOW-USER-SWITCHES	
HELP INF=CD oder INF=C	HELP-MESSAGE-INFORMATION HELP-MESSAGE-INFORMATION INFORMATION=MAXIMUM	(s. S. 263) für diese Operandenwerte gibt es kein SDF-Äquivalent; sie werden trotzdem übersetzt
HOLD	LOCK-FILE-LINK	
IMPORT	IMPORT-FILE SUPPORT=DISK(...)	(s. S. 263)
INTR	SEND-MESSAGE TO=PROGRAM	
JAC	WRITE-ACCOUNTING-RECORD USER-DATA=(...)	
LOAD	LOAD-PROGRAM	(s. S. 263)
LOGOFF	EXIT-JOB	(s. S. 263)
LOGON	SET-LOGON-PARAMETERS	
MARGIN	MODIFY-TERMINAL-OPTIONS	
MRSSTA	SHOW-MRS-CONNECTIONS	
ON	ADD-CJC-ACTION	
OPTION MSG=... MAXLST=... MAXOPT=... DUMP=... TESTPRIV=...	MODIFY-JOB-OPTIONS MODIFY-JOB-OPTIONS MODIFY-JOB-OPTIONS MODIFY-TEST-OPTIONS MODIFY-TEST-OPTIONS	(s. S. 263)
PARAMETER	-----	kein SDF-Äquivalent
PASSWORD nur Kennwort REL=YES	ADD-PASSWORD REMOVE-PASSWORD	(s. S. 263)
PAUSE	SEND-MESSAGE TO=OPERATOR (WAIT-RESPONSE=YES)	
PRINT TAPE=NO TAPE≠NO	PRINT-DOCUMENT WRITE-SPOOL-TAPE	(s. S. 258 und S. 263)
PRINT-FILE	PRINT-DOCUMENT	

Tabelle 17: Übersetzung von Benutzerkommandos durch SDF-CONV

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
PRIORITY <1..9> <30..255> (<30..255>,EXPRESS)	MODIFY-JOB CHANGE-TASK-PRIORITY MODIFY-JOB ...(START=IMMEDIATE) CHANGE-TASK-PRIORITY	(s. S. 263)
PROCEDURE	BEGIN-PROCEDURE	
PSWORD	MODIFY-USER-PROTECTION	
PUNCH DEVICE=CENTRAL, TAPE=NO DEVICE=CENTRAL, TAPE=POOLER DEVICE=DISKETTE  DEVIN=... FORM=... HREC=... TRLNUM=..., HDRNUM=...	PUNCH-FILE  -----  WRITE-DISKETTE  WRITE-DISKETTE WRITE-DISKETTE WRITE-DISKETTE WRITE-DISKETTE TRAILER-EXIT-NUMBER	(s. S. 260 und S. 263)  keine Unterstützung von Poolerbändern ab BS2000 V11  diese Operanden beziehen sich auf nicht mehr unterstützte Lochkartenleser; sie werden trotzdem übersetzt.
RDTFT	SHOW-FILE-LINK	(s. S. 263)
RELEASE	REMOVE-FILE-LINK	(s. S. 263)
REMARK	REMARK	
RESTART	RESTART-PROGRAM	(s. S. 263)
RESUME mit Operanden ohne Operanden	----- RESUME-PROGRAM	kein ISP-Kommando (s. S. 261)
RFAEND	REMOVE-RFA-CONNECTION	
RFASTART	SET-RFA-CONNECTION	
RTI	RESUME-PROCEDURE	
SECURE	SECURE-RESOURCE-ALLOCATION	(s. S. 261 und S. 263)
SETJV	MODIFY-JV	(s. S. 263)
SETSW mit Operanden ohne Operanden	MODIFY-JOB-SWITCHES SHOW-JOB-SWITCHES	(s. S. 263)
SETUS	MODIFY-USER-SWITCHES	
SET-FILE-LINK	ADD-FILE-LINK	
SET-SS-OPTIONS	SET-SUBSYSTEM-OPTIONS	(s. S. 264)
SKIP	SKIP-COMMANDS IF=NO-CONDITION	(s. S. 264)
SKIPJV	SKIP-COMMANDS IF=JOB-SWITCHES(...)	

Tabelle 17: Übersetzung von Benutzerkommandos durch SDF-CONV

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
SKIPUS	SKIP-COMMANDS IF=USER-SWITCHES(...)	
SPARAM	MODIFY-SPOOL-OUT-OPTIONS	(s. S. 264)
SQUC REL=	HOLD-SPOOL-OUT RESUME-SPOOL-OUT	(s. S. 264)
STAJV	SHOW-JV-ATTRIBUTES	(s. S. 264)
STAM	SHOW-MASTER-CATALOG-ENTRY	
STATUS ENVIR / LIST / PROG BIAS / CATEGORY / SUMMARY andere Operanden	SHOW-USER-STATUS SHOW-SYSTEM-STATUS  SHOW-JOB-STATUS	(s. S. 264)
STEP	SET-JOB-STEP	
SYSFILE SYSDTA=... SYSIPT=... SYSLST=... SYSLSTn=...  SYSOPT=... SYSOUT=... TASKLIB=... TASKLIB=(NO)	ASSIGN-SYSDTA TO= ASSIGN-SYSIPT ASSIGN-SYSLST TO= ASSIGN-SYSLST TO=*SYSLST-NUMBER(n) ASSIGN-SYSOPT ASSIGN-SYSOUT TO= SET-TASKLIB REMOVE-TASKLIB	(s. S. 261 und S. 264)
SYSTATUS	SHOW-SYSTEM-FILE-ASSIGNMENTS	(s. S. 264)
TCHNG	MODIFY-TERMINAL-OPTIONS	(s. S. 261 und S. 264)
TYPE	SEND-MESSAGE TO=OPERATOR (WAIT-RESPONSE=NO)	
VERIFY REPAIR=YES  REPAIR=ABS  REPAIR=NO	REPAIR-DISK-FILES FILE-STATUS=OPEN(...) REPAIR-DISK-FILES FILE-STATUS=ANY(...) REMOVE-FILE-ALLOCATION-LOCKS	(s. S. 264)
WAIT	WAIT-EVENT UNTIL=JV(...)	
WHEN	WAIT-EVENT UNTIL=USER-SWITCHES(...)	

Tabelle 17: Übersetzung von Benutzerkommandos durch SDF-CONV

### 13.8.2 Kommandos für die Systemverwaltung

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
BIAS	MODIFY-SYSTEM-BIAS	
CANCEL , KILL	FORCE-JOB-CANCEL	
CATEGORY	MODIFY-TASK-CATEGORIES	
CATM STATE=NEW STATE=UPDATE	ADD-MASTER-CATALOG-ENTRY MODIFY-MASTER-CATALOG-ENTRY	(s. S. 264)
CREATE-SS	START-SUBSYSTEM	
DELETE-SS	STOP-SUBSYSTEM	
ERAM	REMOVE-MASTER-CATALOG-ENTRY	
EXCAT CANCEL FORCE END	CANCEL-CATALOG-EXPORT FORCE-CATALOG-EXPORT EXPORT-PUBLIC-VOLUME-SET	(s. S. 261 und S. 264)
HOLD-SS	HOLD-SUBSYSTEM	
IMCAT	IMPORT-PUBLIC-VOLUME-SET IMPORT-CATALOG	(s. S. 261 und S. 264)
JOIN	-----	Übersetzung ist kontextabhängig; keine automatische Übersetzung, aber Übersetzungsvorschlag (ADD-USER bzw. MODIFY-USER) in der Kommentarzeile (s. S. 256)
LOADAID	-----	kein SDF-Äquivalent
LOADAIDSYS	-----	kein SDF-Äquivalent
MODIFY-MSG-DEFAULTS	MODIFY-MSG-FILE-ASSIGNMENT	
MSGCONTROL FILE=STD FILE≠STD	RESET-MSG-FILE-ASSIGNMENT MODIFY-MSG-FILE-ASSIGNMENT	(s. S. 264)
MSGCONTROL		
NCHOLD	HOLD-TASK	
NCREL	RESUME-TASK	
RCARD	-----	Kartenleser werden nicht mehr unterstützt
RDIR	REDIRECT-REMOTE-OUTPUT	
RELEASE-JOB	RESUME-JOB	
RELEASE-JOB-CLASS	RESUME-JOB-CLASS	
RELEASE-JOB-STREAM	RESUME-JOB-STREAM	

Tabelle 18: Übersetzung von Kommandos für die Systemverwaltung durch SDF-CONV

Zu übersetzendes Kommando	übersetzt in SDF-Kommando	Grund, warum nicht übersetzt wird bzw. Anmerkungen
RESUME-SS	RESUME-SUBSYSTEM	
RFD USE=INPUT USE=NO	START-DISKETTE-INPUT STOP-DISKETTE-INPUT	(s. S. 261 und S. 264)
SDVC	-----	keine eindeutige Zuordnung möglich
SEVER RESET REMOVE SET	UNLOCK-USER REMOVE-USER LOCK-USER	(s. S. 264)
SHARE	ADD-SHARED-PROGRAM	
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS	
SPMGT	SET-SPACE-SATURATION-LEVEL	(s. S. 264)
SQUC	HOLD-SPOOLOUT	(s. S. 264)
STAM	SHOW-MASTER-CATALOG-ENTRY	(s. S. 264)
START-SS	START-SUBSYSTEM	
STATUS BIAS / CATEGORY / SUMMARY ENVIR / LIST / PROG andere Operanden	SHOW-SYSTEM-STATUS  SHOW-USER-STATUS SHOW-JOB-STATUS	
STOP-SS	STOP-SUBSYSTEM	

Tabelle 18: Übersetzung von Kommandos für die Systemverwaltung durch SDF-CONV

## 13.9 Meldungen von SDF-CONV

SDF-CONV unterscheidet zwischen NOTE, WARNING, ERROR und FAILURE.

Wenn ein ERROR oder FAILURE auftritt, wird die Fehlermeldung nicht nur nach SYSLST, sondern auch nach SYSOUT ausgegeben. Die Übersetzung durch SDF-CONV wird danach vorzeitig beendet.

Bei einer NOTE oder WARNING erscheint die Fehlermeldung nur im Protokoll (Ausgabe nach SYSLST). Das Programm SDF-CONV bearbeitet nach der Fehlermeldung die nächste Kommandozeile der Eingabeprozedur.

Die Meldungen von SDF-CONV haben folgenden Aufbau: CVRxxxx <meldungstext>

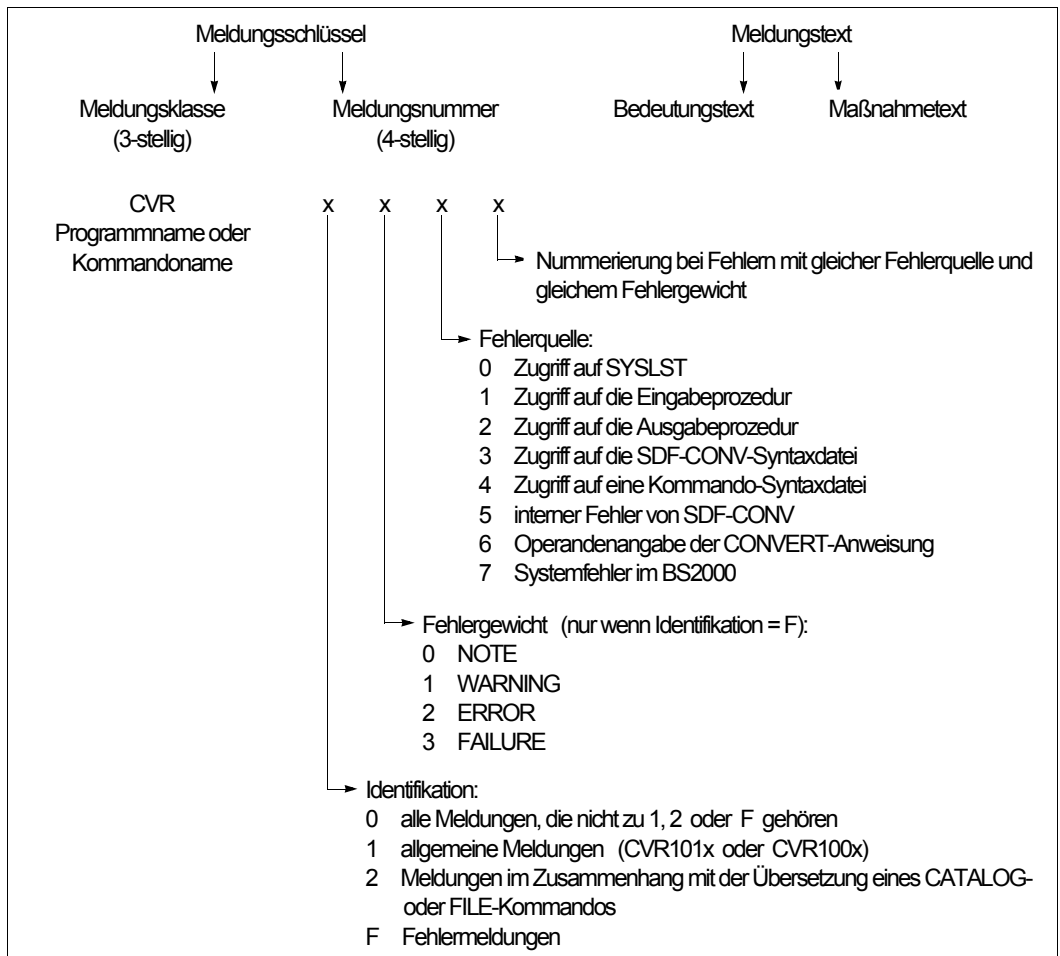


Bild 51: Aufbau der Meldungen von SDF-CONV

In wenigen Fällen folgen einer Meldung eine oder mehrere andere Fehlermeldungen, um die Fehlerursache oder deren Wirkung noch genauer zu beschreiben. Solche erläuternde Meldungen haben z.B. das Format `CMDxxxx <meldungstext>` .

Mit Hilfe des Meldungsschlüssels (`CVRxxxx` oder `CMDxxxx`) lassen sich nähere Informationen über die Fehlerursache und Maßnahmen zur Fehlerbeseitigung einholen (mit der Standardanweisung bzw. Kommando `HELP-MSG-INFORMATION`, siehe Handbuch „Kommandos“ [1]).





---

## 14 DISPLAY

**Versionsstand:**        **DISPLAY V1.1**

**Privilegierung:**       **TSOS**

Viele SHOW-Kommandos im BS2000/OSD geben abhängig von den Privilegien eines Benutzers nur dessen eigene Daten oder die Daten aller Benutzer aus.

Die Vergabe zusätzlicher Privilegien an einen Benutzer kann jedoch unerwünscht sein, da diese Privilegien in der Regel weiterreichende Rechte, z.B. die Ausführbarkeit von Kommandos, einschließen.

Das Subsystem DISPLAY ermöglicht nicht-privilegierten Benutzern Zugang zur vollen Datenmenge ausgewählter SHOW-Kommandos, ohne dass den Benutzern zusätzliche Privilegien gewährt werden müssen. Nur wenige sicherheitsrelevante SHOW-Kommandos sind davon ausgeschlossen.

Um Sicherheit und Kompatibilität zu gewährleisten, enthält die gelieferte Standardkonfiguration nur die erforderlichen Basisverfahren. Es werden keine Funktionskommandos für den Zugang zu privilegierten Daten geliefert.

### 14.1 Produkteinsatz

Es ist Aufgabe der Systemverwaltung zu entscheiden, welche Benutzer entsprechend den geltenden Anforderungen Zugang zu welchen SHOW-Kommandos erhalten. Hierzu dient das Kommando CREATE-DISPLAY-CMD. Mit diesem kann die Systemverwaltung DISPLAY-Kommandos erzeugen. Ein DISPLAY-Kommando ist standardmäßig eine Kopie des SHOW-Kommandos mit dem entsprechenden Namen, z.B. ist das Kommando DISPLAY-FILE-ATTRIBUTES eine Kopie des Kommandos SHOW-FILE-ATTRIBUTES. Für das DISPLAY-Kommando kann die Systemverwaltung jedoch festlegen, welche Privilegien seine Verwendung und somit die Ausgabe der vollen Datenmenge erlauben. Standardmäßig werden DISPLAY-Kommandos in die mit dem Subsystem DISPLAY ausgelieferte Syntaxdatei SYSSDF.DISPLAY.nnn eingetragen, die Systemverwaltung kann jedoch auch eine beliebige andere Syntaxdatei angeben.

Da DISPLAY-Kommandos im Wesentlichen Kopien von SHOW-Kommandos sind, müssen sie nach jeder Änderung des entsprechenden SHOW-Kommandos erneut erzeugt werden, z. B. nach einer Versionsänderung oder einer Änderung des Kommandos mit SDF-A. Daher wird empfohlen, die Kommandos zur Erzeugung von DISPLAY-Kommandos als Prozedur abzuspeichern. Dies vereinfacht die Rekonstruktion nach einer Versionsaktualisierung.

Ein DISPLAY-Kommando (DISPLAY-...) verhält sich zum entsprechenden SHOW-Kommando (SHOW-...) wie folgt:

- Die Textausgabe auf SYSOUT oder SYSLST ist identisch zu der Ausgabe, die das entsprechende SHOW-Kommando liefert, wenn es unter einer Benutzerkennung mit den Privilegien TSOS und PRINT-SERVICE-ADMINISTRATION aufgerufen wird.
- Bei normaler Ausführung sind Returncode und Spin-off-Verhalten wie beim entsprechenden SHOW-Kommando. Falls bei der Ausführung interne Fehler auftreten, wird folgender Kommando-Returncode erzeugt:

(SC2)	SC1	Maincode	Bedeutung
	32	DSP0001	Interner Fehler

- Ein DISPLAY-Kommando erzeugt keine Ausgabe in den S-Variablenstrom SYSINF. Die strukturierte Ausgabe in S-Variable muss explizit mit dem Kommando /EXECUTE-CMD CMD=(DISPLAY-...), STRUCT-OUT=... angefordert werden.

### Modifikation von DISPLAY-Kommandos

Die Systemverwaltung kann DISPLAY-Kommandos nachträglich mit SDF-A bearbeiten. So ist es möglich, die Funktionalität gegenüber dem entsprechenden SHOW-Kommando einzuschränken oder aber auch Operanden oder Werte freizugeben, deren Verwendung beim SHOW-Kommando eingeschränkt ist.

Folgende Aktionen sind erlaubt:

- Löschen optionaler Operanden  
Optionale Operanden eines DISPLAY-Kommandos können vollständig gelöscht werden. Den Operanden wird intern der Standardwert zugewiesen, der für das SHOW-Kommando gilt.
- Ändern der Privilegien von Operanden und Operandenwerten:  
Einzelnen Operanden und Operandenwerten können individuelle Privilegien zugewiesen werden, um ihre Verwendung auf bestimmte Benutzer einzuschränken.

So lange für alle Benutzer eine erlaubte Eingabeoption bestehen bleibt, können die Standardprivilegien eines Operanden oder Operandenwertes nach Bedarf eingeschränkt werden. Sie dürfen jedoch nur erweitert werden, wenn sie schon vorher mindestens eines der Privilegien TSOS und PRINT-SERVICE-ADMINISTRATION enthalten.

- Ändern der Standardwerte  
So lange die syntaktischen Regeln gewahrt bleiben, können die Standardwerte für DISPLAY-Kommandos gelöscht (d.h. die Eingabe des betreffenden Operanden wird zwingend erforderlich), geändert oder (neu) eingeführt werden (d.h. die Eingabe eines bis dahin zwingend erforderlichen Operanden wird optional).
- Einschränken des Wertebereichs:  
Der erlaubte Wertebereich eines Operanden kann durch Löschen oder Ändern der Standardoptionen (z.B. Verbot von Wildcards in Dateinamen) eingeschränkt werden. Insbesondere können allgemeine Datentypen durch eine Liste von tatsächlich erlaubten Werten ersetzt werden (z.B. Datentyp <name 1..8> für eine Benutzerkennung durch die Benutzerkennungen X, Y und Z, für die das Kommando zugelassen werden soll).

### 14.1.1 Einschränkungen

Für den Einsatz des Subsystems DISPLAY gelten folgenden Einschränkungen:

- Das Subsystem DISPLAY darf, entsprechend den F2/Q3-Spezifikationen, nicht in einem Sicherheitssystem verwendet werden.
- Für die Modifikation von DISPLAY-Kommandos ist das Produkt SDF-A erforderlich.
- Das Kommando CREATE-DISPLAY-CMD kann nicht auf Kommandos angewendet werden, die mit CMD-ALLOWED = NO definiert wurden.
- Aus Sicherheitsgründen darf das Kommando CREATE-DISPLAY-CMD nicht verwendet werden bei:
  - dem Kommando SHOW-USER-ATTRIBUTES
  - Kommandos, die mit CMD-ALLOWED=YES (UNLOAD=YES) definiert wurden
- Bei der Freigabe anderer sicherheitsrelevanter Kommandos wie SHOW-FILE (die Datei kann Kennwörter enthalten) oder SHOW-JV-ATTRIBUTES (Ausgabe von Kennwörtern mit INF=\*ALL) wird nachdrücklich empfohlen, den Zugriff durch Modifikationen zu beschränken (siehe „[Modifikation von DISPLAY-Kommandos](#)“ auf Seite 306).

## 14.2 Kommandos

### 14.2.1 Übersicht über alle Kommandos

Das Subsystem DISPLAY akzeptiert die in folgender Tabelle aufgelisteten Kommandos.

<b>Anweisungen</b>	<b>Kurzbeschreibung</b>
CREATE-DISPLAY-CMD	DISPLAY-Kommando erzeugen
DELETE-DISPLAY-CMD	DISPLAY-Kommando löschen

Tabelle 19: Kommandos des Subsystems DISPLAY

## 14.2.2 CREATE-DISPLAY-CMD DISPLAY-Kommando erzeugen

Das Kommando CREATE-DISPLAY-CMD erzeugt ein DISPLAY-Kommando. Damit kann nicht-privilegierten Benutzern die Funktionalität des entsprechenden SHOW-Kommandos zur Verfügung gestellt werden.

CREATE-DISPLAY-CMD	Kurzname: CRDPCMD
<p><b>CMD</b> = &lt;structured-name 1..30&gt;</p> <p><b>,FROM-SYNTAX-FILE</b> = <u>*TASK-HIERARCHY</u> / &lt;filename 1..54&gt;</p> <p><b>,INTERNAL-NAME</b> = <u>*AUTOMATIC</u> / &lt;alphanum-name 1..8&gt;</p> <p><b>,PRIVILEGE</b> = <u>*ALL</u> / <u>*EXCEPT(...)</u> / list-poss (64): &lt;structured-name 1..30&gt;</p> <p style="padding-left: 2em;"><b>*EXCEPT(...)</b></p> <p style="padding-left: 4em;">  <b>EXCEPT-PRIVILEGE</b> = list-poss (64): &lt;structured-name 1..30&gt;</p> <p><b>,TO-SYNTAX-FILE</b> = <u>*SYSSDF</u> / &lt;filename 1..54&gt;</p> <p><b>,WRITE-MODE</b> = <u>*NEW</u> / <u>*EXCHANGE</u></p>	

### **CMD = <structured-name 1..30>**

Vollständiger, nicht abgekürzter Name des SHOW-Kommandos, für das ein entsprechendes DISPLAY-Kommando erzeugt werden soll.

### **FROM-SYNTAX-FILE = \*TASK-HIERARCHY / <filename 1..54>**

Syntaxdatei, in der das SHOW-Kommando definiert ist.

### **INTERNAL-NAME = \*AUTOMATIC / <alphanum-name 1..8>**

Interner Name des DISPLAY-Kommandos. Standardmäßig wird der Name automatisch generiert. Bei Mehrdeutigkeit muss der Name explizit angegeben werden.

### **PRIVILEGE = \*ALL / \*EXCEPT(...)** / list-poss (64): <structured-name 1..30>

Privilegien für das DISPLAY-Kommando. Syntax und Standardwert entsprechen dem Operanden PRIVILEGE bei der SDF-A-Anweisung ADD-CMD.

**PRIVILEGE = \*EXCEPT(...)****EXCEPT-PRIVILEGE = list-poss (64): <structured-name 1..30>**

Ausnahmenliste von Privilegien für das DISPLAY-Kommando. Die Ausführung des DISPLAY-Kommandos ist mit jedem Privileg außer den hier angegebenen erlaubt. Dies gilt auch für Privilegien, die erst in der Zukunft eingeführt werden.

**TO-SYNTAX-FILE = \*SYSSDF / <filename 1..54>**

Syntaxdatei, in die das DISPLAY-Kommando eingetragen werden soll. Diese Syntaxdatei muss vorhanden sein und den Anwendungsbereich DISPLAY enthalten. Standardmäßig wird das Kommando in die für das Subsystem DISPLAY installierte Syntaxdatei eingetragen.

**WRITE-MODE = \*NEW / \*EXCHANGE**

Legt fest, ob ein bereits vorhandener Eintrag in der Syntaxdatei überschrieben wird.

**WRITE-MODE = \*NEW**

Es darf noch kein Eintrag für das zu erzeugende Kommando existieren. Falls doch, wird die Kommandoverarbeitung mit Fehler abgebrochen.

**WRITE-MODE = \*EXCHANGE**

Ein bereits vorhandener Eintrag für das zu erzeugende Kommando wird ersetzt. Falls noch kein Eintrag existiert, wird die Kommandoverarbeitung mit Fehler abgebrochen.

## DELETE-DISPLAY-CMD DISPLAY-Kommando löschen

Das Kommando DELETE-DISPLAY-CMD löscht ein DISPLAY-Kommando.

<b>DELETE-DISPLAY-CMD</b>	Kurzname: <b>DLDP CMD</b>
<b>CMD</b> = <structured-name_1..30> <b>,FROM-SYNTAX-FILE</b> = <u>*SYSSDF</u> / <filename 1..54>	

### **CMD = <structured-name 1..30>**

Vollständiger, nicht abgekürzter Name des zu löschenden DISPLAY-Kommandos.

### **FROM-SYNTAX-FILE = \*SYSSDF / <filename 1..54>**

Syntaxdatei, in der das zu löschende DISPLAY-Kommando definiert ist.

## 14.3 Installation

Detaillierte Informationen zur Installation des Subsystems DISPLAY und zu Hard- und Softwarevoraussetzungen finden Sie in der Freigabemitteilung zum Softwareprodukt DISPLAY.

Die Auslieferung der Dateien zu DISPLAY V1.1A erfolgt mit dem Lieferverfahren SOLIS.

Folgende Dateien werden ausgeliefert:

<b>Datei</b>	<b>Bemerkung</b>
SKMLNK.DISPLAY.011	Subsystemmodulbibliothek für den Betrieb auf SX-Servern
SPMLNK.DISPLAY.011	Subsystemmodulbibliothek für den Betrieb auf SQ-Servern
SYSFGM.DISPLAY.011.D	Freigabemitteilung (deutsch)
SYSFGM.DISPLAY.011.E	Freigabemitteilung (englisch)
SYSLNK.DISPLAY.011	Subsystemmodulbibliothek für den Betrieb auf S-Servern
SYSTEMS.DISPLAY.011	Meldungsdatei des Subsystems
SYSPRC.DISPLAY.011	Prozedur für das Erstellen von Einträgen in der Syntaxdatei
SYSRMS.DISPLAY.011	Laderliefermenge für DISPLAY
SYSSDF.DISPLAY.011	Subsystemsyntaxdatei
SYSSII.DISPLAY.011	Struktur- und Installationsinformationen für IMON
SYSSSC.DISPLAY.011	DSSM-Deklarationen

Im SOLIS2-Lieferanschreiben sind die jeweils gültigen Datei- und Datenträger-Merkmale aufgeführt.



---

# Literatur

Die Handbücher sind online unter <http://manuals.ts.fujitsu.com> zu finden oder in gedruckter Form gegen gesondertes Entgelt unter <http://manualshop.ts.fujitsu.com> zu bestellen.

- [1] **BS2000/OSD-BC  
Kommandos**  
Benutzerhandbuch
- [2] **BS2000/OSD-BC  
Einführung in die Systembetreuung**  
Benutzerhandbuch
- [3] **BS2000/OSD-BC  
Systeminstallation**  
Benutzerhandbuch
- [4] **SDF-A (BS2000/OSD)**  
Benutzerhandbuch
- [5] **SDF-P (BS2000/OSD)**  
**Programmieren in der Kommandosprache**  
Benutzerhandbuch
- [6] **POSIX (BS200/OSD)**  
**POSIX-Grundlagen**  
Benutzerhandbuch
- [7] **POSIX (BS200/OSD)**  
**Kommandos**  
Benutzerhandbuch
- [8] **FDDRL (BS2000/OSD)**  
Benutzerhandbuch

- [9] **JV** (BS2000/OSD)  
Jobvariablen  
Benutzerhandbuch
  
- [10] **SECOS** (BS2000/OSD)  
**Security Control System - Zugangs- und Zugriffskontrolle**  
Benutzerhandbuch
  
- [11] **SECOS** (BS2000/OSD)  
**Security Control System - Beweissicherung**  
Benutzerhandbuch
  
- [12] **XHCS** (BS2000/OSD)  
**8-bit-Code-Verarbeitung im BS2000/OSD**  
Benutzerhandbuch
  
- [13] **BS2000/OSD**  
**Softbooks Deutsch**  
CD-ROM

---

# Stichwörter

## A

- Abhängigkeit von der
  - Betriebssystemversion 241
- Abkürzung
  - garantiert 78
  - mehrdeutig 86
  - von Namen 77
  - von Schlüsselworten 77
- Ablauf von SDF-I 168
- aktuelles Objekt 196
- Anweisung
  - Aufbau 14
  - Menü 53
  - Syntaxanalyse 89
  - Testmodus 89
- Anweisungen
  - CONVERT 234
  - Übersicht 233
- Anwendungsbereich 52
- Anwendungsbereiche
  - ausgeben 19
  - Menü 52
- Aufbau
  - SDF-Anweisung 14
  - SDF-Kommando 14
- Ausgabe
  - der Monitor-Jobvariablen zur
    - Programmüberwachung 251
  - in eine PLAM-Bibliothek
    - (Ausgabeprozedur) 248
  - in eine SAM-Datei (Ausgabeprozedur) 248
  - nach SYSLST (Übersetzungsprotokoll) 250
  - nach SYSOUT (kurze Übersicht) 250
  - von Meldungen bei der Umsetzung von Datenzeilen 245

- Ausgabeprozedur (übersetzte Prozedur)
  - Ausgabeformat 243, 246
  - Bestimmung des Kommandoformates 244
  - Bestimmung des Prozedurformates 243, 244
  - Dokumentationsgrad 240, 248
  - Inhalt 248
  - Tasktyp 242
  - Übersetzungsvorschlag 240, 248
  - Zuweisung zu Datei oder Bibliothekselement 237, 248
- ausgeben
  - Syntaxdateiname 216
- Auskunftssystem 18
- Auswahlmenü 85
  - mehrdeutige Abkürzung 86
  - Musterzeichen 85

## B

- Basis-Systemsyntaxdatei
  - nicht aktivierbar 165
- beenden
  - SDF-I 175
  - SDF-PAR 220
  - SDF-U 198
- Beenden von SDF-CONV 232
- Behandlung von optionalen Operanden 241
- Beispiel
  - Anwendungsbereiche ausgeben 19
  - Benutzerkennwort ändern 20
  - Dialogtask beenden 29
  - Dialogtask starten 29
  - Kommando rekonstruieren 28
  - LOGON-Prozedur 24
  - task-lokale SDF-Optionen abfragen 22
- Beispiele 267

Benutzerkennwort ändern 20  
Benutzerkommandos  
    Einschränkungen bei der Übersetzung 252  
    Übersetzung ISP zu SDF 295  
    Zuordnung ISP zu SDF 286  
benutzerspezifische LOGOFF-Prozeduren 64  
benutzerspezifische LOGON-Prozeduren 63  
Benutzersyntaxdatei 142  
    für SDF-PAR 219  
Bestandteile von SDF-CONV 265  
Bildschirmaufbau 41

### C

Cancel-Funktion 37, 49  
CHECKPOINT/RESTART 159  
CLIGET 67  
CLISSET 67  
CONVERT (Anweisung) 234  
COPY (Anweisung, SDF-U) 193  
CREATE-DISPLAY-CMD (Kommando) 309

### D

Dateihierarchie 223, 228, 229  
Datenzeilen in der Eingabeprozedur 245  
Default-Wert 79  
    task-spezifisch 93  
    übernehmen 79  
Default-Wert (task-spezifisch)  
    definieren 94  
    definieren im Dialog 93  
    definieren in Prozedur 94  
DELETE-DISPLAY-CMD (Kommando) 311  
Dialogbetrieb, Eingabe 11  
Dialogführung  
    abbrechen 59  
    Beispiel 99  
    Bildschirmaufbau 41  
    Eingabe der Operanden 54  
    einstellen 34  
    Einstellung der Sprache 12  
    EXPERT-Modus 33  
    Funktionstasten 47  
    für ISP-Kommando 58  
    maximale Stufe 39

mehrsprachig 12  
Menü 42  
minimale Stufe 40  
mittlere Stufe 40  
NO-Form 34  
Operandenfragebogen 42  
Steueranweisungen 45  
Stufen 39  
    temporär 58  
Dialogführung steuern 247  
Dialogtask  
    beenden 29  
    starten 29  
Dienstprogramm  
    SDF-PAR 219  
Dokumentationsgrad in der  
    Ausgabeprozedur 240, 248  
DSSM 163

### E

EDIT (Anweisung, SDF-U) 196  
Eingabe  
    abkürzen 17  
    Abkürzung 77  
    als Schlüsselwortoperand 80  
    als Stellungsoperand 80  
    aus einer Prozedur 11  
    aus Jobvariablen 70  
    aus Prozedurdatei 62  
    dunkelgesteuert 35  
    explizit 79  
    Fortsetzungszeile 62, 69  
    Funktionstasten 47  
    geblockt 37  
    ignorieren 90  
    im Dialogbetrieb 11  
    im Menümodus 18  
    im Stapelbetrieb 11, 69  
    im Zeilenmodus 18  
    implizit 79  
    in die NEXT-Zeile 42, 45  
    ISP-Kommando 58  
    Korrekturdialog 34  
    Operandenfragebogen 54

- sicherheitsrelevant 88
- speichern 87
- spezielle 35, 43
- Steueranweisung 42
- ungeführter Dialog 33
- verdeckte 11
- von Strukturen 81
- wiederherstellen 11
- wiederverwenden 87
- Eingabeprozedur (zu übersetzende Prozedur)
  - Behandlung von Datenzeilen 245
  - Prozedurformat 243
- Eingabepuffer 18, 87
- Eingabeseriennummer 87
- Einschränkungen bei der Übersetzung 252
  - bedingt übersetzbare Kommandos und Operanden 258
  - nicht übersetzbare Kommandos 256
  - Prozedurparameter als Operanden 262
- Einschränkungen bei der Umwandlung zu S-Prozeduren 254
- END (Anweisung, SDF-I) 175
- END (Anweisung, SDF-PAR) 220
- END (Anweisung, SDF-U) 198
- END (SDF-PAR) 220
- END (Standardanweisung) 148
- Ersetzung
  - von Jobvariablen 71
  - von Prozedurparametern 70
  - von S-Variablen-Ausdrücken 72
- Ersetzung von Jobvariablen 70
- Event 66
- Execute-Funktion 48, 49
- EXECUTE-SYSTEM-CMD (Standardanweisung) 149
- Exit-all-Funktion 36, 47, 48
- Exit-Funktion 36, 47, 48
- EXPERT-Modus 33
- F**
- Fehlerfall (Verhalten von SDF) 165
- Fehlerkorrekturdialog 18
- Fehlermeldungen von SDF-CONV 302
- Folgezeilenbehandlung
  - NEW-Modus 62, 69
  - OLD-Modus 62, 69
- Fortsetzungszeichen
  - in der Ausgabe-prozedur 239
  - in der Eingabe-prozedur 237
- Fortsetzungszeile 62, 69
- Funktionstasten 36
- Funktionstastenbelegung 47
- G**
- garantierte Abkürzung 78
- geblockte Eingabe 37
- geführter Dialog 39, 54
  - Spezialeingaben 56
  - temporär 58
- generieren, Subsystemkatalog 163
- Globalinformation ändern 204
- Gruppensyntaxdatei 141
- H**
- HELP-MSG-INFORMATION (Standardanweisung) 150
- Hierarchie von Syntaxdateien 143
- Hilfe-Funktion 36, 48
- Historie 87
- HOLD-PROGRAM (Standardanweisung) 150
- I**
- Inline-Kommentare 240, 248
- Installation
  - SDF 161
  - SDF-PAR 219
- Installation des Subsystems SDF-CONV 266
- Installationsdateien 161
- ISP (Interactiv Scanner and Processor) 12
  - Dialogführung 58
  - Kommandosprache 12, 17
- ISP-Kommandos
  - Einschränkungen bei der Übersetzung 252
  - Übersetzung in SDF-Kommandos 295
  - Zuordnung zu SDF-Kommandos 286

### J

- Jobvariable zur Programmüberwachung (Monitor-JV) [251](#)
- Jobvariablen ersetzen [71](#)
- Jobvariablen-Ersetzung
  - Einstellung bei der Übersetzung [237](#)

### K

- Kartenleser [252](#)
- Kommando
  - Anwendungsbereich SDF [147](#)
  - Aufbau [14](#)
  - CREATE-DISPLAY-CMD [309](#)
  - Definition ändern mit SDF-U [199](#)
  - DELETE-DISPLAY-CMD [311](#)
  - Eingabebeispiel [99](#)
  - einschränken [12](#)
  - ISP-Format [12](#)
  - Menü [53](#)
  - modifizieren [12](#), [145](#)
  - rekonstruieren [18](#), [28](#)
  - selbst implementieren [62](#)
  - sperrern [12](#)
  - Syntaxanalyse [89](#)
  - temporär geführter Dialog [58](#)
  - Testmodus [89](#)
- Kommando (allgemein)
  - Standardname [78](#)
  - umbenennen [78](#)
- Kommandoprozessor SDF [11](#)
- Kommandos
  - die nicht übersetzt werden sollen [240](#)
  - nicht übersetzbar [256](#)
- Kommandos für die Systemverwaltung
  - Einschränkungen bei der Übersetzung [252](#)
  - Übersetzung ISP zu SDF [300](#)
  - Zuordnung ISP zu SDF [291](#)
- Kommandosprache
  - Aufbau [14](#)
  - ISP [12](#), [17](#)
  - SDF [17](#)
- Kommentarzeilen in der Ausgabeprozedur [240](#)
- Korrektur der Eingabe [34](#)
- Korrekturdialog [11](#)

### L

- laden, SDF [163](#)
- Liefereinheitssyntaxdatei [146](#)
- LOGOFF-Prozedur [64](#)
  - benutzerspezifisch [64](#)
  - Standardnamen [64](#)
  - systemweit [64](#)
- LOGON-Prozedur [63](#)
  - benutzerspezifisch [63](#)
  - erstellen [24](#)
  - Standardnamen [63](#)
  - systemweit [63](#)
- löschen, Objekte der Syntaxdatei [202](#)

### M

- Makroaufrufe [156](#)
- Meldungsdatei
  - Aufbau [302](#)
  - Installation [266](#)
- Menü [42](#)
  - anfordern [53](#)
  - Anwendungsbereich [52](#)
  - der Anweisungen [53](#)
  - der Kommandos [53](#)
  - Protokoll [83](#)
- MERGE (SDF-I) [176](#)
- MODIFY-CMD (Anweisung, SDF-U) [199](#)
- MODIFY-SDF-OPTIONS (Standardanweisung) [151](#)
- MODIFY-SYNTAX-FILE (Anweisung, SDF-PAR) [222](#)
- MODIFY-SYSTEM-LOGON-INCLUDE (Anweisung, SDF-PAR) [225](#)
- MODIFY-SYSTEM-LOGON-PROCEDURE (Anweisung, SDF-PAR) [224](#)
- MODIFY-VALUE (Anweisung, SDF-U) [200](#)
- MODIFY-VERSION (Anweisung, SDF-PAR) [226](#)
- Monitor-Jobvariable (Programmüberwachung) [251](#)
- Musterzeichen
  - im Anweisungsnamen [85](#)
  - im Kommandonamen [85](#)

**N**

NEXT-Zeile [42, 45](#)  
Nicht-S-Prozedur [231](#)

**O**

Objekt, aktuelles [196](#)  
Old-Modus (Funktionstastenbelegung) [36, 47](#)  
OMNIS [159](#)  
OPEN (Anweisung, SDF-I) [180](#)  
OPEN-PARAMETER-FILE (Anweisung, SDF-PAR) [227](#)  
OPEN-SYNTAX-FILE (Anweisung, SDF-U) [201](#)  
Operand  
    ausblenden im Protokoll [83](#)  
    Default-Wert [79](#)  
    wahlfrei [79](#)  
Operandenfragebogen [42, 54](#)  
    anfordern [54](#)  
    Spezialeingaben [56](#)  
Operandenwert  
    explizit Angabe [79](#)  
    implizite Angabe [79](#)  
    vorbesetzt [79](#)  
Operandenwertdefinition ändern [200](#)  
optionale Operanden bei der Übersetzung [241](#)  
optionale Reps [266](#)

**P**

Parameterdatei [140](#)  
Parameterdatei von SDF [165](#)  
    anzeigen [230](#)  
    Einträge ändern [222, 224, 225](#)  
    erzeugen [219, 227](#)  
    Format ändern [226](#)  
    modifizieren [219](#)  
    öffnen [227](#)  
PROFILE-ID [141](#)  
Programmüberwachung mit Monitor-  
    Jobvariable [251](#)  
Programmunterbrechung [66](#)  
Protokoll [83](#)  
    Menü [83](#)  
    Umfang [83](#)

Protokoll der Übersetzung  
    Ausgabe nach SYSLST [250](#)  
    Ausgabe nach SYSOUT [250](#)

Protokollierung  
    einstellen [83](#)

Prozedur  
    bei LOGOFF [64](#)  
    bei LOGON [63](#)  
    Eingabe aus [62](#)  
    implementieren [62](#)  
    Korrekturdialog [62](#)  
    unterbrechen [65](#)

Prozedurdatei  
    als Ausgabeprozedur (übersetzte  
    Prozedur) [237](#)

Prozedurdateien  
    benutzerspezifische [63, 64](#)  
    systemweite [63, 64](#)

Prozedurdialog [65](#)

Prozedurparameter  
    Einschränkungen [253, 254](#)  
    Einstellung bei der Übersetzung [237](#)

Prozedurparameter als Operanden [262](#)

Prozedurparameter ersetzen [70](#)

Prozedurunterbrechung [65](#)

**R**

Readme-Datei [15](#)  
Refresh-Funktion [47, 48](#)  
REMARK  
    (Standardanweisung) [151](#)  
REMOVE (SDF-I) [183](#)  
REMOVE (SDF-U) [202](#)  
Repdatai  
    Installation [266](#)  
    optionale Reps [266](#)  
RESET-INPUT-DEFAULTS  
    (Standardanweisung) [152](#)  
Restore-Funktion [37, 49](#)  
RESTORE-SDF-INPUT  
    Standardanweisung [152](#)  
Rückwärts Blättern [48](#)  
Rückzugsfunktionen [50, 60](#)

### S

- S-Prozedur 231
  - Einschränkungen bei der Umwandlung 254
- S-Variablen-Ausdruck ersetzen 72
- Schlüsselwort
  - Abkürzung 77
- Schlüsselwortoperand 80
- Schlüsselwortwert 77
- Schutz der Eingabe 88
- SDF
  - Aufbau der Kommandosprache 14
  - CHECKPOINT/RESTART 159
  - Dialogführung steuern 247
  - Installation 161
  - Kommando 147
  - Kommandoprozessor 11
  - Kommandosprache 17
  - Korrekturdialog 11
  - laden 163
  - Makroaufrufe 156
  - OMNIS 159
  - Parameterdatei 140
  - PROFILE-ID 141
  - Sprache der Dialogführung 12
  - Standardanweisungen 148
  - starten 163
  - Syntaxbeschreibung 139
  - Syntaxdatei 139
  - verkürzte Eingabe 77
- SDF (System Dialog Facility) 11
- SDF-A (Dienstprogramm)
  - Anweisungsvorrat modifizieren 12
  - Kommandovorrat modifizieren 12
- SDF-CONV
  - Arbeitsweise 231
  - Bestandteile 265
  - Installation 266
  - Starten und Beenden 232
  - Übersicht über alle Anweisungen 233
  - Übersicht über mögliche Umwandlungen 232
- SDF-I
  - Ablauf 168
  - Arbeitsdateien 170
  - Ausgabesyntaxdatei zuweisen 180
  - beenden 175
  - Eingabesyntaxdatei zuweisen 180
  - Fortsetzungszeilen 171
  - Hinweise zu Eingaben 171
  - selbsterstellte Syntaxdateien einmischen 178
  - selbsterstellte Syntaxdateien entmischen 183
  - SESDs einmischen 176
  - SESDs entmischen 183
  - starten 168
  - Syntaxdatei-Kompatibilität 172
  - Syntaxdateien entmischen 183
  - Syntaxdateien mischen 176
- SDF-Kommandos
  - Übersetzung aus ISP-Kommandos 295
  - Zuordnung zu ISP-Kommandos 286
- SDF-PAR 219
  - beenden 220
  - Benutzersyntaxdatei 219
  - installieren 219
  - starten 219
- SDF-Parameterdatei 165, 219
- SDF-Standardanweisungen 233
- SDF-U 189
  - beenden 198
  - fortsetzen 190
  - Kommandodefinition ändern 199
  - Operandenwertdefinition ändern 200
  - starten 189
  - Syntaxdatei öffnen 201
  - Syntaxdatei-Objekte löschen 202
  - unterbrechen 190
- SET-GLOBALS (Anweisung, SDF-U) 204
- SHOW (Anweisung, SDF-U) 210
- SHOW-CORRECTION-INFORMATION (Anweisung, SDF-U) 215
- SHOW-INPUT-DEFAULTS (Standardanweisung) 153
- SHOW-INPUT-HISTORY (Standardanweisung) 153
- SHOW-PARAMETER-FILE (SDF-PAR) 230



- SHOW-SDF-OPTIONS
  - (Standardanweisung) 154
- SHOW-STATUS (Anweisung, SDF-U) 216
- SHOW-STMT (Standardanweisung) 154
- SHOW-SYNTAX-FILE (Anweisung, SDF-I) 185
- Softwareeinheitssyntaxdateien
  - einmischen 176
  - entmischen 183
- Speichern von Eingaben 87
- Spezialeingabe
  - im ungeführten Dialog 35, 43
  - Operandenfragebogen 56
- SSCM 163
- Standardanweisung 148
  - END 148
  - EXECUTE-SYSTEM-CMD 149
  - HELP-MSG-INFORMATION 150
  - HOLD-PROGRAM 150
  - MODIFY-SDF-OPTIONS 151
  - REMARK 151
  - RESET-INPUT-DEFAULTS 152
  - RESTORE-SDF-INPUT 152
  - SHOW-INPUT-DEFAULTS 153
  - SHOW-INPUT-HISTORY 153
  - SHOW-SDF-OPTIONS 154
  - SHOW-STMT 154
  - STEP 155
  - WRITE-TEXT 155
- Standardanweisungen von SDF 171, 191
- Standardname 78
  - Benutzersyntaxdatei 142
  - LOGOFF-Prozedur 64
  - LOGON-Prozedur 63
- Standardwert siehe Default-Wert 79
- Stapelbetrieb, Eingabe 69
- starten
  - SDF-I 168
  - SDF-PAR 219
  - SDF-U 189
- Starten von SDF-CONV 232
- Stellungsoperand 80
- STEP (Anweisung, SDF-U) 217
- STEP (Standardanweisung) 155
- Steueranweisungen für Dialogführung 42, 45
- Steuerfunktionen 247
- Steuerung der Dialogführung 247
- Struktur
  - Attribut NULL-ABBREVIATION 82
  - flache Schreibweise 81
  - STRUCTURE-IMPLICIT-Schreibweise 81
  - verkürzte Eingabe 81
- Stufen der Dialogführung 39
- Style-Guide-Modus
  - (Funktionstastenbelegung) 36, 47
- Subsystem SDF-CONV 265
- Subsystemkatalog generieren 163
- Syntaxanalyse 89, 223, 228, 229
- Syntaxdatei 139
  - Arten 139
  - Benutzer- 142
  - entmischen 146, 167, 183
  - erstellen 145
  - für Benutzergruppe 141
  - Gruppen- 141
  - Hierarchie 143
  - Inhalt 139
  - Inhalte kopieren 193
  - Liefereinheit 146
  - mischen 146, 167, 176
  - modifizieren 145, 189
  - Name ausgeben 216
  - Objekte ausgeben 210
  - Objekte löschen 202
  - öffnen 201
  - positionieren auf ein Kommando 196
  - selbsterstellte einmischen 178
  - selbsterstellte entmischen 183
  - System- 140
- Syntaxdateien
  - Installation 266
  - Zuweisung 239
- SYSLST
  - Ausgaben 250
  - Zuweisung zu einer Datei 250
- SYSOUT-Ausgaben 250

Systemsyntaxdatei 140  
aktivieren 223, 228, 229  
deaktivieren 223, 229  
Informationen ausgeben 185  
nicht aktivierbar 165  
systemweite LOGOFF-Prozeduren 64  
systemweite LOGON-Prozeduren 63

**T**  
Task-lokale SDF-Optionen 22  
Task-spezifischer Default-Wert 93  
Tasktyp der zu übersetzenden Prozedur 242  
temporär geführter Dialog 58  
Test-Funktion 48  
Testmodus  
einschalten 89  
für Anweisungen 89  
für Kommandos 89

**U**  
Übersetzung der Kommandosprache (ISP zu SDF) 231  
Beispiele 267  
Benutzerkommandos 295  
Einschränkungen ff 252  
Kommandos für die Systemverwaltung 300  
Übersetzungsergebnis  
Ausgabeprozedur 248  
im Protokoll nach SYSOUT 250  
Übersetzungsprotokoll (nach SYSLST) 250  
Übersetzungsvorschlag in der  
Ausgabeprozedur 240, 248  
Umwandlung des Prozedurformates (Nicht-S- in S-Prozedur) 231  
Beispiele 267  
ungeführter Dialog 33  
Korrektur 34  
Spezialeingaben 35, 43  
Unterbrechung  
K2-Funktionstaste 65  
Programm 66  
Prozedur 65  
zulassen 65, 67  
zurückweisen 67

Unterbrechungs-Ereignis 66  
Unterbrechungs-Funktion 36, 47, 48

## V

verdeckte Eingabe 35  
Verhalten von SDF im Fehlerfall 165  
Vorwärts Blättern 49

## W

wahlfreier Operand 79  
Wiederaufsetzpunkt definieren 217  
Wiederherstellung der Eingabe 11  
WRITE-TEXT  
(Standardanweisung) 155

## Z

Zuordnung  
Benutzersyntaxdatei 142  
Gruppensyntaxdatei 141  
Systemsyntaxdatei 140  
Zuordnung ISP zu SDF  
Benutzerkommandos 286  
Kommandos für die Systemverwaltung 291  
Zuweisung  
der Ausgabeprozedur 237, 248  
der Eingabeprozedur 235  
von SYSLST (Übersetzungsprotokoll) 250