

FUJITSU Software BS2000 SDF-P

\*06  
\*06

Version 2.5G  
Juli 2017

Readme-Datei

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

\*06

© 2017 Fujitsu Technology Solutions GmbH

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern. BS2000 ist eine Marke von Fujitsu Technology Solutions GmbH in Europa und in anderen Ländern.

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Programmschnittstellen für den Anwender (Handbuch: Kapitel 13.2 Update)</b>	<b>4</b>
2.1	DELVAR	5
<b>3</b>	<b>Vorgegebene Funktionen</b>	<b>7</b>
3.1	Erweiterung der vorgegebenen Funktionen LOWER-CASE() und UPPER-CASE()	7
3.2	IS-EMPTY-FILE ( )	8
3.3	PROC-NAME ( )	8
<b>4</b>	<b>Kommandos</b>	<b>8</b>
4.1	EXECUTE-CMD	8
4.2	ASSIGN-STREAM (Handbuch: Kapitel 7.3.1 Update)	8
4.3	READ-VARIABLE (siehe Seite 710 im Handbuch)	9
4.4	SHOW-VARIABLE (siehe Seite 767 im Handbuch)	9
4.5	SET-PROCEDURE-OPTIONS (siehe Seite 742 im Handbuch)	10
4.6	SELECT-VARIABLE-ELEMENTS (siehe Seite 734 im Handbuch)	11
4.7	CLOSE-VARIABLE-CONTAINER (siehe Seite 586 im Handbuch)	12
4.8	DECLARE-PARAMETER (siehe Seite 610 im Handbuch)	13
4.9	READ-VARIABLE (siehe Seite 724 im Handbuch)	13
<b>5</b>	<b>Meldungen (Handbuch: Kapitel 17 Update)</b>	<b>13</b>

# 1 Einleitung

\*05 Diese Readme-Datei beschreibt die für SDF-P V2.5A, V2.5B, V2.5C, V2.5D, V2.5E, V2.5F und V2.5G geltenden Änderungen / Erweiterungen.

\*01 Änderungen in Version SDF-P V2.5B sind mit \*01 am linken Rand hervorgehoben  
 \*02 Änderungen in Version SDF-P V2.5C sind mit \*02 am linken Rand hervorgehoben  
 \*03 Änderungen in Version SDF-P V2.5D sind mit \*03 am linken Rand hervorgehoben  
 \*04 Änderungen in Version SDF-P V2.5E sind mit \*04 am linken Rand hervorgehoben  
 \*05 Änderungen in Version SDF-P V2.5F sind mit \*05 am linken Rand hervorgehoben  
 \*06 Änderungen in Version SDF-P V2.5G sind mit \*06 am linken Rand hervorgehoben

Die Änderungen / Erweiterungen betreffen folgendes Handbuch:

\*03 SDF-P V2.4A  
 \*03 Ausgabe Mai 2007  
 \*03 Bestellnummer: U6442-J-Z125-6

## Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5A

Kommandos:

ASSIGN-STREAM	Neue Einschränkungen: FHS Stream-Server Zuordnung.
READ-VARIABLE	Neuer Operandart für Bibliothekselementversion.
SHOW-VARIABLE	Neuer Operandart für Bibliothekselementversion.

## \*01 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5B

\*01 Vorgegebene Funktionen:

*01	UPPER-CASE	neuer Parameter CODED-CHARACTER-SET
*01	LOWER-CASE	neuer Parameter CODED-CHARACTER-SET

\*01 Kommandos:

*01	SET-PROCEDURE-OPTIONS	
*01		neuer Operand TRANSLATION-CCS zur Vorgabe der
*01		CCS für UPPER-CASE und LOWER-CASE vorgegebenen
*01		Funktionen

## \*02 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5C

\*02 Kommandos:

*02	SELECT-VARIABLE-ELEMENTS	
*02		HEADER-LINE bis to 240 Zeichen
*02		neuer Operand LENGTH in der Struktur
*02		SELECTION-CODE=*YES, um die Breite der Auswahl
*02		Spalte anzugeben.
*02	SET-PROCEDURE-OPTIONS	
*02		Hinweis zum Wert *SPIN-OFF-COMPATIBLE des Operanden
*02		ERROR-MECHANISM.

\*03 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5D

\*03

\*03 Vorgegebene Funktion

\*03

\*03 IS-EMPTY-FILE Unterstützung von ISAM Dateien

\*04 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5E

\*04

\*04 Kommando:

\*04

\*04 CLOSE-VARIABLE-CONTAINER

\*04

\*04 Löschen der im Variablenbehälter declarierten Variab-

\*04

\*04 len.

\*05 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5F

\*05

\*05 Makro:

\*05

\*05 DELVAR

\*05

\*05 Entfernung von Variablen aus Programmen, die in Nicht-

\*05

\*05 S-Prozeduren ablaufen.

\*05

\*05 Meldungen:

\*05

\*05 SDP0201, SDP0203, SDP0204, SDP0207, SDP0215

\*05

\*05 Angabe der Kontrollstruktur-beginnenden Kommando-

\*05

\*05 Zeile.

\*06 Zusammenfassung der Änderungen / Erweiterungen für SDF-P V2.5G

\*06

\*06 Vorgegebene Funktion

\*06

\*06 PROC-NAME() gibt den Namen der aktuellen Prozedur zurück

In den folgenden Abschnitten werden nur die neuen, geänderten oder erweiterten Operanden und Operandenwert beschrieben. Die Position der Operanden in der Kommandosyntax ist jedoch ersichtlich.

## 2 Programmschnittstellen für den Anwender (Handbuch: Kapitel 13.2 Update)

\*05 SDF-P bietet dem Assembler-Programmierer folgende Schnittstellen:

\*05

Macro	Function
DELVAR	Entfernen von einfachen und zusammengesetzten Variablen

\*05

\*05 Über die Makros PUTVAR, GETVAR, DELVAR, SHOWSSA, TRANSVV und VARINF

\*05

\*05 können S-Variablen angesprochen werden, das heißt die Variablen, die auch über die

\*05

\*05 SDF-P-Kommandoschnittstelle angesprochen werden können..

\*05 **2.1 DELVAR**

\*05 Der Makroaufruf DELVAR entfernt eine Variable. DELVAR kann auf einfache und zu-  
 \*05 sammengesetzte Variablen angewendet werden. DELVAR kann nicht auf Elemente  
 \*05 zusammengesetzter Variablen angewendet werden.

Operation	Operands
DELVAR	MF = E ,PARAM = <name 1..8> / (<integer 1..15> )
	MF = D ,PREFIX = D / prefix
	MF = C ,PREFIX = D / prefix ,MACID = ELV / macid
	MF = L ,NAMLEN = <integer 1..255> ,NAMADR = <name 1..8> ,SCOPE = *VISIBLE / *TASKONLY ,CALLER = USER / SYSTEM

\*05 **Operandenbeschreibung**

\*05 **MF = E**

\*05 Execute-Form des Makroaufrufs: erzeugt einen SVC.

\*05 **PARAM**

\*05 Designates Bezeichnet die Adresse der Operandenliste, die für den Makroaufruf  
 \*05 ausgewertet wird (Adresse des Makroaufrufs mit MF=L).

\*05 **= <name 1..8>**

\*05 Bezeichnet die symbolische Adresse der Operandenliste.

\*05 **= (<integer 1..15>)**

\*05 Bezeichnet das Register, das die Adresse der Operandenliste enthält.

\*05 **MF = D**

\*05 DSECT-Form des Makroaufrufs; erzeugt eine DSECT für die Operandenliste. Die gene-  
 \*05 rierten Namen beginnen mit dem Buchstaben D; sie können mit PREFIX verändert wer-  
 \*05 den.

\*05 **PREFIX = D / prefix**

\*05 Definiert das erste Zeichen der generierten Namen.

\*05 Voreinstellung: Die generierten Namen beginnen mit dem Buchstaben D.

\*05 **MF = C**

\*05 C-Form des Makroaufrufs; erzeugt eine Operandenliste, deren symbolische Namen mit  
 \*05 der Zeichenfolge DELV beginnen. Sie können durch PREFIX und MACID verändert wer-  
 \*05 den.

\*05 **PREFIX = D / prefix**

\*05 Definiert das erste Zeichen der generierten Namen.

\*05 Voreinstellung: Die generierten Namen beginnen mit dem Buchstaben D.

\*05 **MACID = ELV / macid**

\*05 Bis zu drei Zeichen langer String, der die Zeichen 2 bis 4 der generierten Namen  
 \*05 ersetzt. Voreinstellung: ELV

\*05 **MF = L**  
 \*05 LIST-Form des Makroaufrufs: erzeugt die Operandenliste für den Makroaufruf mit MF=E  
 \*05 (Execute-Form); der Makroaufruf muss über eine symbolische Adresse adressierbar  
 \*05 sein.

\*05 **NAMLEN = <integer 1..255>**  
 \*05 Bezeichnet die Länge des Variablennamens.

\*05 **NAMADR = <name 1..8>**  
 \*05 Bezeichnet den symbolischen Namen der Adresse des Variablennamens.

\*05 **SCOPE**  
 \*05 Bezeichnet den Geltungsbereich der Variablen.

\*05 **= \*VISIBLE**  
 \*05 Die Variable ist eine prozedurlokale Variable.

\*05 **= \*TASKONLY**  
 \*05 Die Variable ist eine taskglobale Variable.

\*05 **Return codes**  
 \*05 Die folgende Tabelle listet die Returncodes in hexadezimaler Schreibweise auf.

Subcode2	Subcode1	Maincode	Meaning
00	00	0000	Makroaufruf war erfolgreich; kein Fehler
01	00	0000	Variable nicht deklariert
00	01	0001	Parameter-Fehler
00	01	0002	Syntaxfehler im Variablennamen
00	40	0019	Entfernen nicht möglich
00	40	001A	Entfernen nicht erfolgreich
00	01	FFFF	Unbekannte Unit- oder Funktions-Nummer
00	02	FFFF	Funktion nicht verfügbar
00	03	FFFF	Falsche Version der Operandenliste

### 3 Vorgegebene Funktionen

#### \*01 3.1 Erweiterung der vorgegebenen Funktionen LOWER-CASE() \*01 und UPPER-CASE() \*01

\*01 Anwendungsgebiet: **Konvertierungsfunktionen**

\*01 Die vorgegebenen Funktionen LOWER-CASE() and UPPER-CASE() übersetzen die  
\*01 angegebene Zeichenkette auf Klein- bzw. Grossschreibung für die Zeichen, die im an-  
\*01 gegebenen Coded-Character-Set (CCS) definiert sind. Der Umfang der zu übersetzen-  
\*01 den codes kann um den für die Task aktiven CCS bzw. einen angegebenen CCS erwei-  
\*01 tert werden.

#### \*01 **Format**

LOWER-CASE ( ) UPPER-CASE ( ) .... ,CODED-CHARACTER-SET = <u>*STD</u> / *EDF03IRV / *CURRENT / string_expression
---

#### \*01 **Eingabeparameter**

\*01 **CODED-CHARACTER-SET =**  
\*01 Gibt an, welcher CCS zur Übersetzung nach Klein- bzw. Grossschreibung im Einsatz  
\*01 kommt.

\*01 Sollte der CCS weder unterstützt noch bekannt sein, oder nicht der basic EDF03IRV  
\*01 sein und das Subsystem XHCS nicht geladen sein, wird die Ausführung zurückgewie-  
\*01 sen. Der CCS muss ein 8-bit Code sein. 16-bit codes (UTFE, UNICODE) werden nicht  
\*01 unterstützt.

\*01 **CODED-CHARACTER-SET = \*STD**  
\*01 Standardeinstellung. Die auf Prozedurebene Vorgabe kommt im Einsatz (Operand  
\*01 TRANSLATION-CCS im Kommando /SET-PROCEDURE-OPTIONS). Ausser Proze-  
\*01 durumgebung ist \*STD der kompatibel \*EDF03IRV Wert solange keine externe Einstel-  
\*01 lung das auf \*CURRENT ändert. Die externe Einstellung ist ein optionaler REP für SDF-  
\*01 P-BASYS. Dieser REP kommt dann für alle Tasks im Einsatz.

\*01 **CODED-CHARACTER-SET = \*EDF03IRV**  
\*01 Nur die Zeichen, die im basic coded character set definiert werden, werden übersetzt  
\*01 (kompatibles Verhalten)

\*01 **CODED-CHARACTER-SET = \*CURRENT**  
\*01 Die Zeichen werden entsprechend des für die Task im Einsatz Coded-Character-Sets  
\*01 übersetzt.

\*01 **CODED-CHARACTER-SET = string\_expression**  
\*01 Die Zeichen werden entsprechend des angegebenen Coded-Character-Sets übersetzt.

- \*01 **Bemerkung** Wenn die vorgegebene Funktionen literale Zeichenketten aus einer Prozedurzeile bearbeiten, werden diese Angaben schon vom System nach einem EBCDIC-kompatiblen Format übersetzt. Die Angabe eines CCS (expliziter Name oder implizit über Task-Einstellung) ist nur sinnvoll, wenn erweiterte EBCDIC Coded-Character-Set (zB. EDF041) angegeben werden oder die Variablen-Inhalte im gegebenen CCS codiert wurden (zB. mit der vorgegebenen Funktion FROM-X-LITERAL).
- \*01
- \*01
- \*01
- \*01
- \*01
- \*01
- \*01
- \*01 Zusätzliche Meldungen für CODED-CHARACTER-SET Angabe:
- \*01
- \*01 SDP0539 CODED-CHARACTER-SET '(&00)' NICHT UNTERSTUETZT
- \*01 SDP0540 CODED-CHARACTER-SET '(&00)' UNGUELTIG
- \*01 SDP0541 CODED-CHARACTER-SET '(&00)' UNBEKANNT
- \*01 SDP0542 CODED-CHARACTER-SET FUNKTION NICHT VON DER LAUFENDEN SUBSYSTEM 'XHCS' VERSION UNTERSTUETZT
- \*01
- \*01 SDP0543 SUBSYSTEM 'XHCS' NICHT VERFUEGBAR

\*03 **3.2 IS-EMPTY-FILE ( )**

- \*03
- \*03 Anwendungsgebiet: **Umgebungsinformationen**
- \*03
- \*03 Die Funktion IS-EMPTY-FILE( ) prüft, ob eine SAM oder ISAM Datei leer ist d.h. ob die keine Daten enthält.
- \*03

\*06 **3.3 PROC-NAME ( )**

- \*06
- \*06 Anwendungsgebiet: **Umgebungsinformationen**
- \*06
- \*06 Die Funktion PROC-NAME( ) gibt den Namen der Prozedur zurück von der die Anweisung abgesetzt wurde.
- \*06

## 4 Kommandos

### 4.1 EXECUTE-CMD

**Format**

EXECUTE-CMD
CMD = <c-string 0..1800 with-low> / <text 0..1800 with-low>
...

### 4.2 ASSIGN-STREAM (Handbuch: Kapitel 7.3.1 Update)

Bevor eine dieser Optionen verwendet werden kann, muss der vom Programm verwendete Variablenstrom mit dem Operanden SERVER des ASSIGN-STREAM-Kommandos FHS zugewiesen werden. Der Standard-Variablenstrom (d.h. SYSINF, SYSMMSG und SYSVAR) konnte dem FHS-Server nicht mehr zugewiesen werden. Diese Einschränkung hat ihren Grund darin, dass FHS ggf. ebenfalls strukturierte Meldungen ausgeben und damit eine Deadlock-Situation verursachen könnte.



### 4.3 READ-VARIABLE (siehe Seite 710 im Handbuch)

#### Format

READ-VARIABLE
<pre> ... , INPUT = *LIBRARY-ELEMENT (...)     ...     *LIBRARY-ELEMENT (...)           ELEMENT = &lt;composed-name_1..64_with-underscore&gt; (...)             &lt;composed-name_1..64_with-underscore&gt; (...)                   VERSION = ... / &lt;composed-name_1..24_with-underscore&gt;</pre>

#### Operandenbeschreibung

**INPUT = \*LIBRARY-ELEMENT (...)**

**ELEMENT = <composed-name\_1..64\_with-underscore> (...)**  
Name des Bibliothekselements.

**VERSION = ... / <composed-name 1..24\_with-underscore>**  
Version des Bibliothekselements.

### 4.4 SHOW-VARIABLE (siehe Seite 767 im Handbuch)

#### Format

SHOW-VARIABLE
<pre> ... , OUTPUT = ... / *LIBRARY-ELEMENT (...)     *LIBRARY-ELEMENT (...)           ...           , ELEMENT = &lt;composed-name_1..64_with-underscore&gt; (...)             composed-name_1..64_with-underscore(...)                   VERSION = ... / &lt;composed-name_1..24_with-underscore&gt;</pre>

#### Operandenbeschreibung

**OUTPUT = \*LIBRARY-ELEMENT (...)**

**ELEMENT = <composed-name 1..64\_with-underscore> (...)**  
Name des Bibliothekselements.

**VERSION = <composed-name 1..24\_with-underscore>**  
Version des Bibliothekselements.

\*01 **4.5 SET-PROCEDURE-OPTIONS (siehe Seite 742 im Handbuch)**

\*01 **Format**

<pre> SET-PROCEDURE-OPTIONS ... , ERROR-MECHANISM = <u>*SPIN-OFF-COMPATIBLE</u> / *BY-RETURNCODE ... , TRANSLATION-CCS = <u>*STD</u> / *EDF03IRV / *CURRENT                 </pre>
--

\*01 **Operandenbeschreibung**

\*01 **TRANSLATION-CCS =**

\*01 Bestimmt, welcher Coded-Character-Set (CCS) die SDF-P vorgegebene Funktionen  
 \*01 UPPER-CASE und LOWER-CASE für die Übersetzung nach Klein- bzw. Grossschrei-  
 \*01 bung verwenden müssen.

\*01 Falls der im Einsatz CCS nicht unterstützt oder nicht bekannt ist, oder der im Einsatz  
 \*01 CCS nicht EDF03IRV ist und das XHCS Subsystem nicht geladen ist, wird die Ausführ-  
 \*01 ung zurückgewiesen. Der CCS muss ein 8-bit Code sein. 16-bit Codes werden nicht  
 \*01 unterstützt.

\*01 **TRANSLATION-CCS = \*STD**

\*01 Der basic EDF03IRV wird verwendet, falls keine externe Einstellung diese Vorgabe nach  
 \*01 \*CURRENT umsetzt. Diese externe Einstellung ist ein optionaler REP für SDF-P-  
 \*01 BASYS. Dieser REP ist System global.

\*01 **TRANSLATION-CCS = \*EDF03IRV**

\*01 Erzwingt kompatiblen Wert (unter Verwendung des basic EDF03IRV).

\*01 **TRANSLATION-CCS = \*CURRENT**

\*01 Der dem Eingabestrom aktuell zugewiesene CCS wird verwendet.

\*01 **Bemerkung** Wenn die vorgegebene Funktion literale Zeichenketten aus einer Pro-  
 \*01 zedurzeile bearbeitet, wurden diese Angaben bereits vom System in ein  
 \*01 EBCDIC-kompatibles Format übersetzt. Die Angabe eines CCS (explizi-  
 \*01 ter Name oder implizit über Task-Einstellung) ist nur sinnvoll, wenn er-  
 \*01 weiterte EBCDIC Coded-Character-Set (zB. EDF041) angegeben wer-  
 \*01 den oder die Variablen-Inhalte im gegebenen CCS codiert wurden (zB.  
 \*01 mit der vorgegebenen Funktion FROM-X-LITERAL).

\*01 **Zusätzliche Meldungen**

- \*01 SDP0539 CODED-CHARACTER-SET '(&00)' NICHT UNTERSTUETZT
- \*01 SDP0541 CODED-CHARACTER-SET '(&00)' UNBEKANNT
- \*01 SDP0542 CODED-CHARACTER-SET FUNKTION NICHT VON DER LAUFENDEN
- \*01 SUBSYSTEM 'XHCS' VERSION UNTERSTUETZT
- \*01 SDP0543 SUBSYSTEM 'XHCS' NICHT VERFUEGBAR

\*02 **ERROR-MECHANISM = \*SPIN-OFF-COMPATIBLE**  
 \*02 Die Beschreibung des Werts \*SPIN-OFF-COMPATIBLE für den Operanden ERROR-MECHANISM wird um folgenden Hinweis ergänzt:

\*02 **Hinweis**

\*02 Die Einstellung ERROR-MECHANISM=\*SPIN-OFF-COMPATIBLE wirkt grundsätzlich S-Prozedur-lokal. Die Propagierung eines Fehlers in die rufende S-Prozedur geschieht ausschliesslich über den Subcode1.  
 \*02 D. h. insbesondere, dass das implizite Verlassen einer \*SPIN-OFF-COMPATIBLE-S-Prozedur bei aktivem Spin-Off und Subcode1 == 0 in der rufenden \*SPIN-OFF-COMPATIBLE-S-Prozedur bzw. Nicht-S-Prozedur als Nicht-Fehler gewertet und somit der Spin-Off dort deaktiviert wird.

\*02 **4.6 SELECT-VARIABLE-ELEMENTS (siehe Seite 734 im Handbuch)**

\*02 **Format**

```

SELECT-VARIABLE-ELEMENTS
...
, TO-VARIABLE = <composed-name 1..255> (...)
    <composed-name 1..255> (...)
        | SELECTION-CODE = *NO / *YES (...)
            *YES (...)
                | LENGTH = 1 / <integer 1..75>
...
, HEADER-LINE = *NONE / <c-string 1..240>
, MESSAGE = *NONE / <text 1..240 with-low>
...
    
```

\*02 **Operandenbeschreibung**

\*02 **TO-VARIABLE = <composed-name 1..255> (...)**

\*02 **SELECTION-CODE = \*YES (...)**

\*02 ...  
 \*02 Wenn SELECTION-CODE=\*YES angegeben wird, wird der Wert der Auswahl Spalte im SELECTION-CODE Element von der Ausgabe Variablen kopiert. Zurückgegebene Werte haben variabel Länge.

\*02 **LENGTH = 1 / <integer 1..75>**

\*02 Gibt die Länge des SELECTION-CODE Werts an, also die Breite der Auswahl Spalte. Dieser Wert führt zu einer Abkürzung der Länge der Auswahl Zeilen d.h. eine Spalte weniger per zusätzliche Auswahl Spalte. Diese Angabe hat nur einen Sinn, wenn der SELECTION-CODE als Ausgabewert erwartet wird.  
 \*02 NUL-Angaben (0x00) zwischen die in der Auswahl Spalte eingegebenen Daten werden ausgefügt d.h. die durch NUL getrennte Zeichen werden zusammengefügt. Die Länge des zurückgegebenen SELECTION-CODE ist die Länge der nicht-NUL eingegebenen Zeichen.

\*02 **HEADER-LINE = \*NONE / <c-string 1..240>**  
 \*02 Gibt die Titel-Zeile der Ausgabe-Maske an. Bis 3 Zeilen können angezeigt werden. Die  
 \*02 Anzahl der Zeilen in der Auswahl-Liste hängt von der Anzahl der Titel und Meldung Zei-  
 \*02 len ab d.h. eine Zeile weniger per zusätzliche Titel- und Meldungs-Zeile. Der Text wird  
 \*02 so dargestellt, dass kurze Wörter nicht auf 2 Zeilen geteilt werden, so dass der ange-  
 \*02 zeigte Text länger als der angegebene Wert sein kann. Falls die Ausgabe mehr als 3  
 \*02 Zeilen darstellt, wird die mit '...' abgekürzt.  
 \*02  
 \*02 **MESSAGE = \*NONE / <text 1..240 with-low>**  
 \*02 ...  
 \*02 Bis 3 Zeilen können angezeigt werden. Die Anzahl der Zeilen in der Auswahl-Liste hängt  
 \*02 von der Anzahl der Titel und Meldung Zeilen ab d.h. eine Zeile weniger per zusätzliche  
 \*02 Titel- und Meldungs-Zeile.  
 \*02 Der Text wird so dargestellt, dass kurze Wörter nicht auf 2 Zeilen geteilt werden, so dass  
 \*02 der angezeigte Text länger als der angegebene Wert sein kann. Falls die Ausgabe mehr  
 \*02 als 3 Zeilen darstellt, wird die mit '...' abgekürzt.  
 \*02  
 \*02 **Zusätzliche Meldung**  
 \*02  
 \*02 SDP0260 OPERAND ODER WERT IN DIESER VERSION NICHT UNTERSTUETZT.  
 \*02 VERARBEITUNG WIRD FORTGESETZT.  
 \*02  
 \*02 Erklärung: Ein der folgenden Operanden oder Werte benoetigt eine neuere SDF  
 \*02 Version:  
 \*02 - MESSAGE  
 \*02 - TITLE groesser als 80 Buchstaben  
 \*02 - SELECTION-CODE LENGTH groesser als 1  
 \*02 Die Verarbeitung wird mit einem reduzierten Bildschirm ohne Meldung  
 \*02 bzw. mit einem auf 80 Buchstaben gekuerzten Titel bzw. mit einer Aus-  
 \*02 wahlmaske in einer Spalte fortgesetzt.  
 \*02  
 \*02 Massnahme: Bitte nehmen Sie Kontakt mit dem Systemverwalter auf, um das SDF  
 \*02 Subsystem auf eine hoehere Version, die diese Operanden oder Werte  
 \*02 unterstuetzt, hochzuruesten.

#### \*04 **4.7 CLOSE-VARIABLE-CONTAINER (siehe Seite 586 im Hand- \*04 buch)**

##### \*04 **Kommandobeschreibung**

\*04 ...

##### \*04 *Hinweis*

\*04 Wenn ein Variablenbehälter geschlossen wird, bevor der Geltungsbereich seiner Variab-  
 \*04 len erschöpft ist, werden die Variablen gelöscht.

#### 4.8 DECLARE-PARAMETER (siehe Seite 610 im Handbuch)

```

*05 ...
*05 Beispiel 1: Veranschaulichung des Prompting
*05 ...
*05 /SET-PROCEDURE-OPTIONS
*05 /DECLARE-PARAMETER NAME(INITIAL-VALUE=*PROMPT-
*05 /      (PROMPT-STRING='ENTER THE NAME OF THE COLOR TO BE TRANSLATED',-
*05 /      DEFAULT-VALUE='ROT'))
*05 /COLOR=TRANSLATE(String=NAME , -
*05 /,WHEN1='ROT', THEN1='RED' , -
*05 /,WHEN2='GRUEN', THEN2='GREEN' , -
*05 /,WHEN3='BLAU', THEN3='BLUE' , -
*05 /,WHEN4='GELB', THEN4='YELLOW' , -
*05 /,WHEN5='SCHWARZ',THEN5='BLACK' , -
*05 /,WHEN6='WEISS', THEN6='WHITE' , -
*05 /,ELSE='UNKNOWN')
*05 /SHOW-VAR NAME
*05 /SHOW-VAR COLOR

```

#### 4.9 READ-VARIABLE (siehe Seite 724 im Handbuch)

```

...
*05 Example 5
*05 Die Daten-Datei SDF-P-FILE enthält folgende Datensätze:
*05 ...

```

### 5 Meldungen (Handbuch: Kapitel 17 Update)

```

*05 SDP0201      FALSCHES BLOCK-ABSCHLIESSENDES KOMMANDO FUER KON-
*05              TROLLSTRUKTUR BEGINNEND IN KOMMANDO ZEILE: (&00)
*05
*05 SDP0203      AKTUELLER BLOCK KEIN IF-BLOCK FUER KONTROLLSTRUKTUR
*05              BEGINNEND IN KOMMANDO ZEILE: (&00)
*05
*05 SDP0204      /ELSE-KOMMANDO BEREITS VORHANDEN IN KOMMANDO ZEILE:
*05              (&00)
*05
*05 SDP0207      NICHT ALLE KONTROLLSTRUKTUREN GESCHLOSSEN FUER KON-
*05              TROLLSTRUKTUR BEGINNEND IN KOMMANDO ZEILE: (&00)
*05
*05 SDP0215      LABEL DES BLOCK-ABSCHLIESSENDES KOMMANDOS PASST
*05              NICHT ZUM LABEL DES BLOCK-EINLEITENDEN KOMMANDOS
*05              FUER KONTROLLSTRUKTUR BEGINNEND IN KOMMANDO ZEILE:
*05              (&00)

```