
1 Einführung

Durch die Zerlegung des BS2000/OSD in funktionelle Einheiten (Subsysteme) wird die Komplexität des BS2000-Ablaufs reduziert. SSCM und DSSM sind die beiden Softwareprodukte, mit denen alle Subsysteme deklariert und verwaltet werden.

Mit SSCM (Static Subsystem Catalog Management) wird der statische Subsystemkatalog (SSMCAT) verwaltet. Im Subsystemkatalog werden die erforderlichen Deklarationen aller zu einer Konfiguration gehörenden Subsysteme abgelegt. Der SSMCAT bildet die Datengrundlage für DSSM.

DSSM (Dynamic Subsystem Management) ist die zentrale Instanz in BS2000/OSD zur dynamischen Verwaltung von Subsystemen.

Grundlage dieser Beschreibung sind die Versionen DSSM V4.0 und SSCM V2.3.

1.1 Zielgruppe und Konzept des Handbuchs

Diese Handbuch wendet sich an die Systembetreuung des BS2000/OSD.

Das vorliegende Kapitel „**Einführung**“ enthält eine allgemeine Übersicht über das Handbuch, eine Zusammenstellung der Änderungen gegenüber DSSM V3.6 bzw. SSCM V2.1 und die für die DSSM-Kommandos und SSCM-Anweisungen gültige SDF-Syntaxdarstellung.

Das Handbuch ist in folgende Hauptkapitel gegliedert:

Im Kapitel „**Das Subsystem-Konzept in BS2000/OSD**“ werden Begriffe erklärt und Zusammenhänge dargestellt, die für das Subsystem-Konzept eine wichtige Rolle spielen. Nach der Darstellung der Versionsabhängigkeiten zwischen BS2000/OSD, DSSM und SSCM folgt eine Übersicht über wichtige DSSM-fähige Produkte im BS2000-Grundausbau sowie ausgewählte entbündelte DSSM-fähige Produkte.

Im Kapitel „**DSSM**“ wird die Dynamische Subsystemverwaltung (DSSM) beschrieben: ihre Aufgaben und Funktionen, Verwaltungsstrategien und Fehlerbehandlung sowie die Inbetriebnahme des DSSM durch den Parameterservice zum Startup-Zeitpunkt und die Abrechnungssätze von DSSM.

Die Kommandos von DSSM werden ausführlich beschrieben.

Zwei größere Beispiele schließen das Kapitel ab.

Das Kapitel „**SSCM**“ beschreibt die Subsystemkatalog-Verwaltung (SSCM). Nach einer kurzen Einführung werden die Anweisungen von SSCM ausführlich beschrieben. Es folgen Hinweise zur Installation von SSCM und Beispiele.

Am Ende des Handbuchs finden Sie ein Literatur- und ein Stichwortverzeichnis.

Literaturhinweise werden im Text in Kurztiteln mit einer Verweisnummer in eckigen Klammern angegeben. Der vollständige Titel jeder Druckschrift, auf die verwiesen wird, ist im Literaturverzeichnis ab [Seite 295](#) aufgeführt.

1.2 Änderungen gegenüber der Ausgabe Dezember 1996 (DSSM V3.6 und SSCM V2.1)

Gegenüber dem Vorgängerhandbuch ergeben sich folgende, wesentliche Änderungen:

Neue Funktionen in DSSM V4.0

- Das neue Kommando SHOW-SUBSYSTEM-ATTRIBUTES informiert den Benutzer über die Attribute von globalen und lokalen Subsystemen. Die Informationen können nach SYSLST, SYSOUT oder in S-Variablen ausgegeben werden, siehe [Seite 122](#).
- Das Kommando SHOW-SUBSYSTEM-STATUS zeigt den internen Status eines Subsystems genauer an als bisher und zwar für Subsysteme mit Status IN-CREATE, IN-RESUME, IN-DELETE oder IN-HOLD, siehe [Seite 142](#).
- Mit dem neuen Operanden MONJV im Kommando START-SUBSYSTEM können Subsysteme mit einer Monitor-Jobvariablen überwacht werden. Die MONJV zeigt an, ob das Subsystem aktiv, angehalten, unterbrochen oder gesperrt ist, siehe [Seite 153](#).
- Bei folgenden Kommandos wird mit dem neuen Wert *HIGHEST im Operanden VERSION die höchste Version des Subsystems ausgewählt, die im statischen Subsystemkatalog eingetragen ist:
 - HOLD-SUBSYSTEM
 - RESUME-SUBSYSTEM
 - START-LOCAL-SUBSYSTEM
 - START-SUBSYSTEM
 - STOP-LOCAL-SUBSYSTEM
 - STOP-SUBSYSTEM
- Durch die dynamische Versorgung der Einsprungstellen aus dem BLS-Namensverzeichnis zum Lade-Zeitpunkt wird die Verbindung zu nicht-privilegierten Subsystemen verbessert.

- Eine verbesserte Fehlerbehandlung beim Systemlauf ermöglicht es dem Operator, z.B. einen neuen, gültigen Namen für eine nicht aufgefundene Datei anzugeben und anschließend den Systemlauf fortzusetzen (siehe [Seite 64](#)).

Neue Funktionen in SSCM V2.3

- Mit der neuen Anweisung GENERATE-CATALOG-SOURCE erstellt SSCM eine Datei mit einer Liste aller SSCM-Anweisungen, die für die (Wieder-)Generierung eines Subsystemkatalogs benötigt werden.
- Mit der Angabe von VERSION=*ALL in der Anweisung REMOVE-CATALOG-ENTRY können die Definitionen aller Versionen des angegebenen Subsystems aus dem Katalog gelöscht werden.
- Mit dem neuen Wert *BY-PROGRAM(...) im Operanden SUBSYSTEM-ENTRIES der Anweisung SET-SUBSYSTEM-ATTRIBUTES werden die Einsprungstellen des angegebenen Subsystems dynamisch zum Ladezeitpunkt aus dem BLS-Namensverzeichnis versorgt. In der Anweisung MODIFY-SUBSYSTEM-ATTRIBUTES kann die getroffene Einstellung *BY-PROGRAM(...) modifiziert werden.

1.3 Readme-Datei - Änderungen zu den aktuellen Versionen

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei. Sie finden die Readme-Dateien auf Ihrem BS2000/OSD-System unter dem Dateinamen

SYSRME .produkt.version.sprache

Die Benutzerkennung, unter der sich die Readme-Dateien befinden, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Mit IMON können Sie die Dateinamen auch mit folgendem Kommando ermitteln:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=produkt, LOGICAL-ID=SYSRME.sprache
```

Die Readme-Dateien können Sie mit /SHOW-FILE oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

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

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:



Für Hinweise auf besonders wichtige Informationen.

- Begriffe im Fließtext, die besonders hervorgehoben werden sollen, sind **halbfett** dargestellt.
- In den Beispielen sind die Benutzereingaben in **halbfetter Schreibmaschinenschrift** und Systemausgaben in *normaler Schreibmaschinenschrift* wiedergegeben.
- Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

1.5 SDF-Syntaxdarstellung

Bild 1 zeigt ein Beispiel für die Syntaxdarstellung eines Kommandos in einem Handbuch. Das Kommandoformat besteht aus einem Feld mit dem Kommandonamen. Anschließend werden alle Operanden mit den zulässigen Operandenwerten aufgelistet. Struktureinleitende Operandenwerte und die von ihnen abhängigen Operanden werden zusätzlich aufgelistet.

HELP-SDF	Kurzname: HPSDF
<p>GUIDANCE-MODE = *NO / *YES</p> <p>,SDF-COMMANDS = *NO / *YES</p> <p>,ABBREVIATION-RULES = *NO / *YES</p> <p>,GUIDED-DIALOG = *YES (...)</p> <p style="padding-left: 2em;">*YES(...)</p> <p style="padding-left: 4em;">SCREEN-STEPS = *NO / *YES</p> <p style="padding-left: 4em;">,SPECIAL-FUNCTIONS = *NO / *YES</p> <p style="padding-left: 4em;">,FUNCTION-KEYS = *NO / *YES</p> <p style="padding-left: 4em;">,NEXT-FIELD = *NO / *YES</p> <p>,UNGUIDED-DIALOG = *YES (...) / *NO</p> <p style="padding-left: 2em;">*YES(...)</p> <p style="padding-left: 4em;">SPECIAL-FUNCTIONS = *NO / *YES</p> <p style="padding-left: 4em;">,FUNCTION-KEYS = *NO / *YES</p>	

Bild 1: Syntaxdarstellung des Benutzer-Kommandos HELP-SDF

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

Zu *Tabelle 1: Metasyntax*

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

Zu *Tabelle 2: Datentypen*

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

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

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

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

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

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

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

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

Metasyntax

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

Tabelle 1: Metasyntax

(Teil 1 von 2)

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

Tabelle 1: Metasyntax

(Teil 2 von 2)

Datentypen

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

Tabelle 2: Datentypen

(Teil 1 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
filename	A...Z 0...9 \$, #, @ Bindestrich Punkt	<p>Eingabeformat:</p> $[:cat:][\$user.] \left\{ \begin{array}{l} \text{datei} \\ \text{datei(nr)} \\ \text{gruppe} \end{array} \right\}$ $\text{gruppe} \left\{ \begin{array}{l} (*\text{abs}) \\ (+\text{rel}) \\ (-\text{rel}) \end{array} \right\}$ <p>:cat: wahlfreie Angabe der Katalogkennung; Zeichenvorrat auf A...Z und 0...9 eingeschränkt; max. 4 Zeichen; ist in Doppelpunkte einzuschließen; voreingestellt ist die Katalogkennung, die der Benutzerkennung laut Eintrag im Benutzerkatalog zugeordnet ist.</p> <p>\$user. wahlfreie Angabe der Benutzerkennung; Zeichenvorrat ist A...Z, 0...9, \$, #, @; max. 8 Zeichen; darf nicht mit einer Ziffer beginnen; \$ und Punkt müssen angegeben werden; voreingestellt ist die eigene Benutzerkennung.</p> <p>\$. (Sonderfall) System-Standardkennung</p> <p>datei Datei- oder Jobvariablenname; kann durch Punkt in mehrere Teilnamen gegliedert sein: name₁[.name₂[...]] name_i enthält keinen Punkt und darf nicht mit Bindestrich beginnen oder enden; datei ist max. 41 Zeichen lang, darf nicht mit \$ beginnen und muss mindestens ein Zeichen aus A...Z enthalten.</p>

Tabelle 2: Datentypen

(Teil 2 von 6)

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

Tabelle 2: Datentypen

(Teil 3 von 6)

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

Tabelle 2: Datentypen

(Teil 4 von 6)

Datentyp	Zeichenvorrat	Besonderheiten
product-version	A...Z 0...9 Punkt Hochkomma	<p>Eingabeformat: $[[C']][V][m]m.naso[']$</p> <div style="text-align: right; margin-right: 50px;"> $\begin{array}{c} \\ \\ \text{Korrekturstand} \\ \text{Freigabestand} \end{array}$ </div> <p>wobei m, n, s und o jeweils eine Ziffer und a ein Buchstabe ist. Ob Freigabe- und/oder Korrekturstand angegeben werden dürfen oder ob sie angegeben werden müssen, bestimmen Zusätze zu dem Datentyp (siehe Tabelle 3, Zusätze without-corr, without-man, mandatory-man und mandatory-corr). product-version kann in Hochkommata eingeschlossen werden, wobei der Buchstabe C vorangestellt werden kann. Die Versionsangabe kann mit dem Buchstaben V beginnen.</p>
structured-name	A...Z 0...9 \$, #, @ Bindestrich	<p>alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @</p>
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
time	0...9 Strukturkennzeichen: Doppelpunkt	<p>Angabe einer Tageszeit</p> <p>Eingabeformat: $\left. \begin{array}{l} hh:mm:ss \\ hh:mm \\ hh \end{array} \right\}$</p> <p>hh: Stunden } mm: Minuten } führende Nullen können ss: Sekunden } weggelassen werden</p>
vsn	<p>a) A...Z 0...9</p> <p>b) A...Z 0...9 \$, #, @</p>	<p>a) Eingabeformat: pvsid.folgenummer max. 6 Zeichen; pvsid: 2-4 Zeichen; Eingabe von PUB nicht erlaubt folgenummer: 1-3 Zeichen</p> <p>b) max. 6 Zeichen; PUB darf vorangestellt werden, dann dürfen jedoch nicht \$, #, @ folgen.</p>

Tabelle 2: Datentypen

(Teil 5 von 6)

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

Tabelle 2: Datentypen

(Teil 6 von 6)

Zusätze zu Datentypen

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

Tabelle 3: Zusätze zu Datentypen

(Teil 1 von 7)

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

Tabelle 3: Zusätze zu Datentypen

(Teil 2 von 7)

Zusatz	Bedeutung
with-wild(n) (Forts.)	<p><s₁,...> Ersetzt alle Zeichenfolgen, auf die eine der mit s angegebenen Zeichenkombinationen zutrifft. s kann auch die leere Zeichenfolge sein. Jede Zeichenfolge s kann auch eine Bereichsangabe "s_x:s_y" sein (siehe oben).</p> <p>-s Ersetzt alle Zeichenfolgen, die der angegebenen Zeichenfolge s nicht entsprechen. Das Minuszeichen darf nur am Beginn der Zeichenfolge stehen. Innerhalb der Datentypen filename bzw. partial-filename kann die negierte Zeichenfolge -s genau einmal verwendet werden, d.h., -s kann einen der drei Namensteile cat, user oder datei ersetzen.</p>
	<p>Platzhalter sind in Generations- und Versionsangaben von Dateinamen nicht erlaubt. In Benutzerkennungen ist die Angabe von Platzhaltern der Systembetreuung vorbehalten. Platzhalter können nicht die Begrenzer der Namensteile cat (Doppelpunkte) und user (\$ und Punkt) ersetzen.</p>
POSIX-Platzhalter	Bedeutung
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.
?	Ersetzt genau ein beliebiges Zeichen. Ist als erstes Zeichen außerhalb von Hochkommata nicht zulässig.
[c _x -c _y]	Ersetzt genau ein Zeichen aus dem Bereich c _x und c _y einschließlich der Bereichsgrenzen. c _x und c _y müssen einfache Zeichen sein.
[s]	Ersetzt genau ein Zeichen aus der Zeichenfolge s. Die Ausdrücke [c _x -c _y] und [s] können kombiniert werden zu [s ₁ c _x -c _y s ₂]
[!c _x -c _y]	Ersetzt genau ein Zeichen, das nicht in dem Bereich c _x und c _y einschließlich der Bereichsgrenzen enthalten ist. c _x und c _y müssen einfache Zeichen sein. Die Ausdrücke [!c _x -c _y] und [!s] können kombiniert werden zu [!s ₁ c _x -c _y s ₂]
[!s]	Ersetzt genau ein Zeichen, das nicht in der Zeichenfolge s enthalten ist. Die Ausdrücke [!s] und [!c _x -c _y] können kombiniert werden zu [!s ₁ c _x -c _y s ₂]

Tabelle 3: Zusätze zu Datentypen

(Teil 3 von 7)

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

Tabelle 3: Zusätze zu Datentypen

(Teil 4 von 7)

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

Tabelle 3: Zusätze zu Datentypen

(Teil 5 von 7)

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

Tabelle 3: Zusätze zu Datentypen

(Teil 6 von 7)

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

Tabelle 3: Zusätze zu Datentypen

(Teil 7 von 7)

2 Das Subsystem-Konzept in BS2000/OSD

In diesem Kapitel werden wichtige Begriffe erläutert und anschließend wird informiert über

- die Verwaltung von Subsystemen mit DSSM und SSCM (siehe [Seite 26](#))
- die lokale Subsystemverwaltung (siehe [Seite 27](#))
- die Versionsabhängigkeiten zwischen BS2000/OSD, DSSM und SSCM (siehe [Seite 28](#))
- die wichtigen DSSM-fähigen Produkte im BS2000-Grundausbau (siehe [Seite 31](#))
- ausgewählte entbündelte, DSSM-fähige Produkte (siehe [Seite 33](#)).

2.1 Erläuterung wichtiger Begriffe

In BS2000/OSD lassen sich neben globalen, systemweit gültigen Subsystemen auch lokale Subsysteme zu einer lokalen Subsystem-Konfiguration zusammenstellen und in einem lokalen Subsystemkatalog tasklokal verwalten. Wird im Folgenden von Subsystemen, Konfiguration und Katalog gesprochen, so sind immer die globalen, systemweit gültigen Subsysteme, deren Konfiguration und Katalog gemeint. Lokale Subsysteme, deren Konfiguration und Katalog werden immer mit dem Wort „lokal“ gekennzeichnet.

Subsystem

Ein Subsystem im Sinne der Dynamischen Subsystemverwaltung (DSSM) ist eine automatisch und unabhängig lad-, start- und beendbare Funktionsausführungseinheit, wobei Abhängigkeiten zu anderen Subsystemen beachtet werden müssen. Es kann aus mehreren Subsystemkomponenten bestehen.

Name und Versionsnummer identifizieren ein Subsystem eindeutig gegenüber DSSM. Die Festlegung der Subsystemkomponenten und der Eigenschaften des Subsystems definieren ein Subsystem für DSSM.

Die Eigenschaften eines Subsystems (Attribute) liefern Informationen über:

- Identifikation (Name und Version)
- Zugriffsrechte
- Lade- und Bearbeitungsmechanismen
- Umgebung und Abhängigkeiten
- benötigte Adressräume
- Aufrufpräferenzen
- existierende Auftragseingänge

Die Beziehungen zwischen verschiedenen Subsystemen werden geprägt durch:

- Adressraum-Trennungen
- Bindebeziehungen
- funktionale Abhängigkeiten
- Holdertask-Sharing

Subsystemkomponenten

Unter Subsystemkomponenten sind das Subsystem-Modul (Programmmodul) sowie die Nebenkomponten zu verstehen.

Das Subsystem-Modul ist eine Komponente, die zur Installation des Subsystems unbedingt benötigt wird. Ein Subsystem kann aus verschiedenen Modulen aufgebaut sein.

Nebenkomponten für Subsysteme, die bei der Installation zu vereinbaren sind und beim Aktivieren benötigt werden, sind die Rep-Datei, die NOREF-Datei, die Meldungsdatei, die Syntaxdatei und die Informationsdatei.

Subsystem-Auftragseingang

Der Subsystem-Auftragseingang (call entry) ist die sichtbare Einsprungstelle, an der das Subsystem für Tasks aufrufbar ist.

Ein Auftragseingang wird bestimmt durch seinen Namen, den Typ, den Geltungsbereich und die Lebensdauer. Ein Subsystem kann mehrere Auftragseingänge mit unterschiedlichen Attributen haben. Sie werden von DSSM verwaltet.

Subsystem-Konfiguration

Eine Subsystem-Konfiguration ist die Menge aller Subsysteme, die in einem Systemlauf verfügbar sein sollen. Da immer nur eine Subsystem-Konfiguration im System bedient werden kann, muss sie alle Subsysteme enthalten, die gleichzeitig im System verfügbar sein sollen.

Definiert wird eine Subsystem-Konfiguration durch die Subsysteme, die zu dieser Konfiguration gehören, sowie deren Komponenten und Eigenschaften.

Ein Subsystem (bzw. eine Subsystemversion) kann innerhalb einer gegebenen Subsystem-Konfiguration und in einer bestimmten Kombination verträglich sein, d.h. das Subsystem kann in diesem System mit bestimmten anderen Subsystemen laufen.

Ein Subsystem kann optional oder zwingend sein, d.h. die Subsystem-Konfiguration ist ohne dieses Subsystem ablauffähig oder nicht.

Im laufenden System unterscheidet man zwei Zustände in einer Subsystem-Konfiguration:

- a) Lade-Zustand:
dieser ist gegeben durch die tatsächlich **aktiven** und damit auch wirklich nutzbaren Subsysteme
- b) Deklarations-Zustand:
dieser ist gegeben durch die deklarierten und damit potenziell **aktivierbaren** Subsysteme bzw. Subsystem-Versionen im Subsystemkatalog

Aus einem Deklarations-Zustand können mehrere verschiedene Lade-Zustände abgeleitet werden.

Subsystemkatalog

Die statische Subsystemkatalog-Verwaltung (SSCM) generiert einen statischen Subsystemkatalog (SSMCAT), in dem die Subsystem-Konfiguration definiert ist.

Dieser Subsystemkatalog wird durch die dynamische Subsystem-Verwaltung (DSSM) zum Zeitpunkt der Systemeinleitung geladen. Mit dem Laden wird aus dem statischen ein dynamischer Subsystemkatalog.

Während des Systemlaufs kann der Subsystemkatalog durch die DSSM-Kommandos ADD-SUBSYSTEM, MODIFY-SUBSYSTEM-PARAMETER und REMOVE-SUBSYSTEM verwaltet werden.

Sowohl SSCM als auch DSSM benutzen die Datenstruktur des statischen Subsystemkatalogs. Der Subsystemkatalog kann bis zu 1000 Subsysteme enthalten mit insgesamt max. 16000 Auftragseingängen, 16000 funktionalen Abhängigkeiten untereinander und 200 Adressraum-Einschränkungen.

2.2 Verwaltung von Subsystemen mit DSSM und SSCM

Die dynamische Verwaltung von Subsystemen (DSSM) verfolgt das Ziel, die Komplexität des BS2000-Ablaufs zu reduzieren. Dabei wird das Betriebssystem in funktionelle Einheiten (Subsysteme) zerlegt.

Als Subsysteme werden von DSSM unterstützt:

- Software-Produkte (soweit ausdrücklich als DSSM-fähig ausgewiesen, siehe auch die Tabellen 4 und 5 ab Seite 31)
- System-Exit-Routinen
- Share-Produkte

Darüber hinaus kann die Systembetreuung eigene TU-Programme und System-Exit-Routinen DSSM-fähig machen und als Subsysteme aktivieren.

DSSM unterscheidet in seinen Funktionen privilegiert ablaufende Subsysteme (TPR-Subsysteme), die Teil des BS2000/OSD sind, und nicht-privilegiert ablaufende Subsysteme (TU-Subsysteme), z.B. Benutzerprogramme.

Ebenso unterscheidet DSSM danach, ob ein Subsystem im Systemadressraum oder im Benutzeradressraum geladen ist. TPR-Subsysteme sind immer im Systemadressraum geladen, TU-Subsysteme können entweder im System- oder im Benutzeradressraum geladen sein.

Das Subsystem SSCM ermöglicht eine flexible und anwenderfreundliche Verwaltung des statischen Subsystemkatalogs (SSMCAT).

Das SSD-Object

Das SSD-Object ist eine ISAM-Datei mit der Schlüssellänge von 11 Byte. Es enthält die Definition(en) eines oder mehrerer Subsysteme (jedoch nicht die Definitionen verschiedener Versionen ein und desselben Subsystems).

Jede dieser Definitionen enthält Eigenschaften, Auftragseingänge, Referenzen und Abhängigkeiten des betreffenden Subsystems sowie Informationen über disjunkte Subsysteme und zur Holdertask.

Das SSD-Object selbst kann nicht geladen werden. Zur Aktivierung muss es mit der SSCM-Anweisung ADD-CATALOG-ENTRY in einen statischen Subsystemkatalog eingebunden werden.

2.3 Lokale Subsystemverwaltung

In BS2000/OSD lässt sich eine lokale Subsystem-Konfiguration tasklokal erstellen und verwalten. Es können ein lokaler Subsystemkatalog geladen und entladen und lokale Subsysteme aktiviert und deaktiviert werden.

Die lokale Subsystemverwaltung ist auf nicht-privilegierte Subsysteme beschränkt.

Eine **lokale Subsystem-Konfiguration** besteht aus einer Gruppe von Subsystemen, die lokal im Benutzeradressraum verfügbar sind. Diese werden als **lokale Subsysteme** bezeichnet. Jedes nicht-privilegierte Subsystem kann als lokales Subsystem genutzt werden.

Bestandteile einer lokalen Subsystem-Konfiguration können sowohl Subsysteme bzw. Subsystemversionen sein, die nicht in der globalen Subsystem-Konfiguration enthalten sind, als auch solche, die dort bereits zur Verfügung stehen. Wenn das betreffende Subsystem bereits in der globalen Subsystem-Konfiguration enthalten ist, muss es jedoch zusätzlich als lokales Subsystem für die aufrufende Task definiert werden.

Es muss ein **lokaler Subsystemkatalog** erstellt werden, der die Subsystemdefinitionen der gewünschten lokalen Subsystem-Konfiguration aufnimmt. In diesen Katalog können nur nicht-privilegierte, also TU-Subsysteme aufgenommen werden. Sobald ein Subsystem als lokales Subsystem definiert wird, erhält es folgende Eigenschaften:

- es wird immer in den Benutzeradressraum geladen (MEMORY-CLASS=*LOCAL-UNPRIVILEGED)
- es wird immer bereits zum Zeitpunkt seiner Anforderung durch das Kommando START-LOCAL-SUBSYSTEM geladen werden (CREATION-TIME=*AT-CREATION-REQUEST)
- es kann jederzeit angehalten und entladen werden (SUBSYSTEM-HOLD=*ALLOWED)
- mehr als eine Version des gleichen Subsystems darf nicht gleichzeitig aktiv sein (VERSION-COEXISTENCE=*FORBIDDEN)
- das Laden des Subsystems im Exchange-Modus ist nicht erlaubt (VERSION-EXCHANGE=*FORBIDDEN)

Diese Eigenschaften sind unabhängig von den Eigenschaften, die das gleiche Subsystem im globalen Subsystemkatalog besitzt.

Bereitstellung von lokalen Subsystemen

Der lokale Subsystemkatalog wird erstellt mit der SSCM-Anweisung START-CATALOG-CREATION. Die Subsystemdefinitionen (SSD) der gewünschten lokalen Subsystem-Konfiguration wird in den Katalog aufgenommen mit der Anweisung ADD-CATALOG-ENTRY.

Mit dem Kommando LOAD-LOCAL-SUBSYSTEM-CATALOG wird der erstellte Subsystemkatalog tasklokal geladen. Damit sind die in ihn aufgenommenen lokalen Subsysteme verfügbar, aber noch nicht aktiviert. Die Aktivierung der lokalen Subsysteme erfolgt mit dem Kommando START-LOCAL-SUBSYSTEM.

Ruft die Task jetzt ein Subsystem auf, wird die Verbindung vorzugsweise zum lokalen Subsystem im lokalen Subsystemkatalog aufgebaut. Ist das Subsystem nicht im lokalen Katalog enthalten - ist es also kein lokales Subsystem - oder ist der lokale Subsystemkatalog nicht geladen, erfolgt der Verbindungsaufbau zum globalen Subsystem.

Mit dem Kommando SELECT-PRODUCT-VERSION besteht die Möglichkeit, dynamisch die verschiedenen Versionen eines Subsystems zu priorisieren.

Mit den Kommandos STOP-LOCAL-SUBSYSTEM und UNLOAD-LOCAL-SUBSYSTEM-CATALOG können lokale Subsysteme deaktiviert und der lokale Subsystemkatalog entladen werden.

Zur Arbeit mit lokalen Subsystemen siehe Beispiel auf [Seite 170](#).

2.4 Versionsabhängigkeiten zwischen BS2000/OSD, DSSM und SSCM

Welche SSCM- und DSSM-Versionen stehen unter den verschiedenen BS2000/OSD-Versionen zur Verfügung?

BS2000/OSD-BC V3.0:	mit DSSM V3.6	SSCM V1.0, V2.0, V2.1
	mit DSSM V3.8	SSCM V1.0, V2.0, V2.1, V2.2
	mit DSSM V3.9	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A
	mit DSSM V4.0	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A, V 2.3B
BS2000/OSD-BC V3.1 (in OSD-SVP V1.0/V2.0):	mit DSSM V3.7	SSCM V1.0, V2.0, V2.1
	mit DSSM V3.8	SSCM V1.0, V2.0, V2.1, V2.2
	mit DSSM V3.9	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A
	mit DSSM V4.0	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A, V 2.3B
BS2000/OSD-BC V4.0:	mit DSSM V3.8	SSCM V1.0, V2.0, V2.1, V2.2
	mit DSSM V3.9	SSCM V1.0, V2.0, V2.1, V2.2, V2.3A
	mit DSSM V4.0	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A, V 2.3B
BS2000/OSD-BC V5.0:	mit DSSM V4.0	SSCM V1.0, V2.0, V2.1, V2.2, V 2.3A, V 2.3B

Zu jeder BS2000/OSD-Version gibt es eine oder zwei bestimmte DSSM-Version(en). Höhere DSSM-Versionen können bis DSSM V3.9 jedoch problemlos auf niedrigeren BS2000/OSD-Versionen laufen. Für DSSM V4.0 ist BS2000/OSD V3.0 oder höher erforderlich.

Welches Katalogformat des Subsystemkatalogs (entspricht der SSCM-Version) wird von den verschiedenen DSSM-Versionen ausgewertet?

SSCM V1.0: ab DSSM V3.0

SSCM V2.0: ab DSSM V3.5

SSCM V2.1: ab DSSM V3.6

SSCM V 2.2: ab DSSM V3.8

SSCM V 2.3: ab DSSM V3.9

Die Erstellung oder Änderung eines Subsystemkatalogs ist unabhängig von der aktuell laufenden DSSM-Version.

Aber jede DSSM-Version kann nur die Katalogformate auswerten, die von „ihrer“ SSCM-Version oder einer niedrigeren SSCM-Version erstellt wurden.

Dabei kommt die volle Funktionalität einer DSSM-Version immer erst dann zum Einsatz, wenn mit der zugehörigen SSCM-Version gearbeitet wird.

Das Katalogformat einer SSCM-Version wird von höheren DSSM- und SSCM-Versionen kompatibel unterstützt, und beim Lesen und Zurückschreiben in das höhere Format konvertiert.

Beispiel

Wird ein mit SSCM V2.1 generierter Katalog von DSSM V4.0 oder SSCM V2.3 gelesen und zurückgeschrieben, erhält er das Format von SSCM V2.3.

Unter welchen Bedingungen können die SSCM-Versionen V1.0 bis V2.3B gleichzeitig geladen werden?

Bis DSSM V3.4 ist muss die entsprechende Prozedur SYSPRC.SSCM.0xx zum Laden der höheren Version verwendet werden.

Beispiel

Um das Laden von SSCM V2.3 zu ermöglichen, wenn SSCM V1.0 als Subsystem im Einsatz ist, muss die Prozedur SYSPRC.SSCM.023 verwendet werden, die SSCM V2.3 als Programm lädt.

Ab DSSM V3.5 kann die Version mit dem Kommando SELECT-PRODUCT-VERSION ausgewählt werden. Das Kommando START-SSCM startet dann die gewählte Version.

Da die volle Funktionalität einer DSSM-Version jedoch erst mit der dafür entwickelten, „eigenen“ SSCM-Version zum Einsatz kommt, bringt das gleichzeitige Laden von niedrigeren SSCM-Versionen keine Vorteile für die aktuell laufende BS2000-Session.

Das parallele Laden von verschiedenen SSCM-Versionen ist aber dann nötig, wenn eine Subsystem-Deklarationsdatei SYSSSC bzw. SYSSSD für eine kleinere SSCM-/DSSM-Version als der gegenwärtig aktuellen generiert werden soll (z.B. zum Versionsrückstieg).

2.5 Übersicht über wichtige DSSM-fähige Produkte im BS2000-Grundausbau (GA)

Sp-KI: Speicherklassen (3-6), in die das Produkt geladen werden kann

TU/TPR: Funktionszustand des Subsystems (TU: nicht-privilegiert / TPR: privilegiert)

Lit: Verweis auf das Literaturverzeichnis ab [Seite 295](#)

Produkt	Funktion	Sp-KI	TU/TPR	Lit
ACS	Ermöglicht den Dateizugriff über Aliasnamen	3,4	TPR	14
ADAM	Bedienen von Geräten, die nicht durch die logischen Zugriffsmethoden des BS2000/OSD unterstützt werden	3,4	TPR	1
AIDSYS	Systemabhängiger Teil der Test- und Diagnosehilfe AID (zu AID siehe Tabelle 5)	3,4	TPR	-
BINDER	Zusammenfügen eines übersetzten Quellprogramms mit anderen Objektmodulen und LLMs zu einer ladefähigen Einheit	4,6	TU	5
CALENDAR	Erstellung von benutzerspezifischen Kalendern	3,4	TPR	6
DIV	Objektorientierte Zugriffsmethode, besonders zur Verarbeitung unstrukturierter Daten	3,4	TPR	13
FASTPAM	Blockorientierte Zugriffsmethode für NK4-Plattendateien	3,4	TU/TPR	13
GET-TIME	Bereitstellen von Datum und standardisierter Weltzeit und lokaler Landeszeit	4	TU	14
GSMAN	Global-Storage-Verwaltung	3,4	TPR	14
IMON	Monitor zur Softwareinstallation			17
IMON	(IMON-BAS) Installation und Registration von Software	4	TPR	-
IMON-GPN	Unterstützung der Zuordnung von logischen Namen und Pfadnamen von Dateien	4	TPR	-
INIT	Initialisieren von Magnetbändern, MBKs und Disketten	4,6	TU/TPR	9
MIP	Verwalten und Ausgeben von Systemmeldungen	4	TPR	9
NDM	Geräteverwaltung			14
NKS	Überwachen der Betriebsmittelreservierung	4	TPR	-
NKV[T] NKV[D]	Überwachen der montierten Datenträger (T=Tape; D=Disk)	4	TPR	-
NKISAM	Satzorientierte Zugriffsmethode für Dateien mit dem Blockformat "DATA" (ohne gesonderten PAM-Schlüssel)	4	TPR	13
PAMCONV	Konvertieren von Dateiformaten	4,6	TU	9

Tabelle 4: DSSM-fähige Produkte im Grundausbau (GA)

(Teil 1 von 2)

Produkt	Funktion	Sp-KI	TU/TPR	Lit
PCA	Zwischenpuffern von Daten außerhalb des Betriebssystems in einer Plattenspeicher-Steuereinheit (Hardware-Caching)	4	TPR	24
POSIX-BC	GA-Komponente von POSIX	3,4	TPR	26 27
POSPRRTS	Runtime-Funktionen für C (in TPR)	4	TPR	
SDF	System Dialog Facility; Sprachoberfläche des BS2000/OSD	4	TPR	35
SDFSYS	Systemabhängiger Teil von SDF (nur bis einschließlich SDF V3.0 vorhanden)	4	TPR	35
SDF-CONV	Konvertieren von Prozedurformaten	6	TU	33
SDF-P-BASYS	GA-Komponente des Produkts SDF-P (zu SDF-P siehe Tabelle 5)	4	TPR	20
SHOW-FILE	Ausgabe von Dateiinhalten auf den Bildschirm	3, 4	TPR	19
SIR	Installieren von Software	4	TPR	40
SPOOL	Ein/Ausgabe-Steuerung auf bestimmte Gerätefamilien			
PRMPRES	Erstellen und Verwalten von Druckressourcen im BS2000-SPOOL (Komponente von PRM)	4	TU	28
PRMMAN	Erstellen und Verwalten von Druckressourcen im BS2000-SPOOL (Komponente von PRM)	4	TPR	28
SPOOL	Organisation von SPOOLIN/SPOOLOUT und Verwalten von Druckaufträgen	4	TPR	38
SRPMNUC	System Resources and Privilege Management (GA-Komponente von SRPM; zu SECOS siehe Tabelle 5)	4	TPR	36
SSCM	Generieren von Subsystemkatalogen	4,6	TU, ab S.	179
SYSFILE	Unterstützen der Systemdateien SYSLST und SYSOUT	4	TPR	19
TANGRAM	Zuordnung affiner Taskgruppen zu Prozessoren unter Berücksichtigung ihres Leistungsbedarfs			
TANGRAM	Regelungsfunktion	3,4	TPR	14
TANGBAS	Verwaltung der Taskgruppen	3,4	TPR	-
TSDRIVER	Synchronisation der Systemzeit mit dem Time Server	4	TPR	-
VOLIN	Formatieren und Initialisieren von Festplatten	4,5	TPR	9
WARTOPT	Überwachen der unter der Kennung HARDWARE-MAINTENANCE laufenden Wartungstask	3,4	TPR	14

Tabelle 4: DSSM-fähige Produkte im Grundausbau (GA)

(Teil 2 von 2)

2.6 Übersicht über ausgewählte entbündelte, DSSM-fähige Produkte

Sp-KI: Speicherklassen (3-6), in die das Produkt geladen werden kann

TU/TPR: Funktionszustand des Subsystems (TU: nichtprivilegiert / TPR: privilegiert)

Lit: Verweis auf das Literaturverzeichnis ab [Seite 295](#)

Produkt	Funktion	Sp-KI	TU/TPR	Lit
AID	Test- und Diagnosehilfe			
AID	Symbolisches und nicht-symbolisches Testen	4	TPR	2
LLMAID	Information über Bindelademodule	4	TPR	-
ARCHIVE	Sichern, Rekonstruieren und Transfer von Daten in Dateien und Jobvariablen	4	TPR	3
CRTE	Runtime-Funktionen für C und COBOL	4,6	TU	7
DAB	Caching in BS2000/OSD zur Vermeidung von Performance-Engpässen (Software-Caching)	3,4	TPR	8
Dprint	(Distributed Print Services) Drucken in Computernetzen			
DPRINTCL	Client-Part für Dprint: Erzeugen von Druckaufträgen	4	TPR	10
DPRINTCM	Basismechanismen: Realisierung allgemeiner Dienste	4	TPR	10
DPRINTSV	Server-Part für Dprint: Verwalten von Druckaufträgen	4	TPR	10
DRV	Aufzeichnungsverfahren, mit dem Daten auf zwei Platten doppelt geführt werden können	4	TPR	11
DUAL-COPY	Erhöhung der Ausfallsicherheit von Systemen und der Datensicherheit durch redundante Aufzeichnung	4	TPR	14
EDT	Editor für SAM-, ISAM- und POSIX-Dateien sowie für Elemente aus Programmbibliotheken	4,6	TU	12
FDDRL	Physikalische Datensicherung von Platten und Pubsets	4,5	TPR	15
HIPLEX MSCF	Einrichten eines Rechnerverbundes auf der Basis des Datenkommunikationsnetzes BCAM	3,4	TPR	23
HSMS	Sichern, Archivieren und Rekonstruieren von Daten sowie Unterstützen der Datenverwaltung auf externen Speichern	4	TPR	16
HSMS-SV	Bedient den HSMS-CL (HSMS-Client im SINIX)	4	TPR	16
IMON-XT	SHOW-Funktionen für IMON (zu IMON siehe Tabelle 4)	4,6	TU	17
JV	Steuern von Aufträgen und Programmen mit Jobvariablen	4	TPR	18

Tabelle 5: DSSM-fähige entbündelte Produkte (Auswahl)

(Teil 1 von 2)

Produkt	Funktion	Sp-KI	TU/TPR	Lit
LMS	Verwalten von Bibliotheken	4,6	TU	21
MAREN	Verwalten von Datenträgerbeständen im BS2000-RZ	3,4	TPR	22
PCS	Unterstützen des Systembetreuers bei der optimalen Einstellung und dem optimalen Betrieb der BS2000-Anlage	3,4	TPR	25
PROP-XT	RZ-Automatisierung und benutzerspezifische Problemlösung im RZ	4	TPR	29
RSO	Steuern der Ausgabe von Fern-SPOOL-Aufträgen auf dezentrale (RSO-) Drucker (nur mit SPOOL ablauffähig)	4	TPR	31
SDF-A	Verwalten und Anpassen der SDF-Benutzeroberfläche	6	TU	32
SDF-P	Prozedur- und Variablenkonzept, Kommandos zur Prozedursteuerung, blockorientierte Fehlerbehandlung	4	TPR	34
SDF-P-BIF	Built-in-Funktionen von SDF-P	4	TPR	34
SECOS	Gewährleisten der Zugriffssicherheit für das BS2000/OSD			
GUARDS	Verwalten von Objekten (Guards) und Auswerten der Zugriffsbedingungen (Komponente von GUARDS)	4	TPR	36
GUARDEF	Verwalten des Standardschutzes sowie der Attribut- und Objektpfade	4	TPR	36
GUARDCOO	Verwalten des Miteigentümerschutzes	4	TPR	36
SATCP	Überwachen von Ereignissen und Alarmen (Komponente von SAT)	4	TPR	36
SRPMOPT	(Komponente von SRPM)	4	TPR	36
SM2	Messsystem zum Erfassen und Auswerten von Daten zur Leistung des BS2000/OSD und zur Auslastung der Betriebsmittel	3,4	TPR	37
SPS	BS2000-SPOOL-Zusatzkomponente und Druckertreiber für APA-Drucker	4	TPR	39

Tabelle 5: DSSM-fähige entbündelte Produkte (Auswahl)

(Teil 2 von 2)

3 DSSM

Die dynamische Verwaltung von Subsystemen (DSSM) verfolgt das Ziel, die Komplexität des BS2000-Ablaufs zu reduzieren. Dabei wird das Betriebssystem in funktionelle Einheiten (Subsysteme) zerlegt.

Subsysteme besitzen folgende Eigenschaften:

- Jedes Subsystem bildet eine in sich geschlossene Einheit.
- Sie können während des BS2000-Systemlaufs aktiviert, angehalten, fortgesetzt und deaktiviert werden.
- Ist für ein Subsystem eine neue Version erstellt worden, kann die alte gegen die neue Version ausgetauscht werden oder beide Versionen können parallel betrieben werden.

DSSM verwaltet als zentrale Instanz die Subsystemkonfiguration in BS2000/OSD.

Es steuert das Laden, Initialisieren, Anhalten, Fortsetzen und Beenden von Subsystemen im Systemlauf. DSSM kann das aktuelle System dynamisch (online) verändern, indem es Subsysteme in den Subsystemkatalog einbinden oder aus ihm herausnehmen kann, ohne dass das gesamte System angehalten und wieder neu gestartet werden muss.

DSSM ist für BS2000/OSD zwingend, da wichtige Bestandteile des Grundaubaus (siehe die Produktübersicht auf [Seite 31](#)) nur mit DSSM betrieben werden können.

DSSM V4.0 steht für BS2000-Versionen ab BS2000/OSD-BC V3.0 zur Verfügung. Für DSSM V4.0 können Subsystemkataloge ab SSCM V1.0 generiert werden.

Zu Fragen der Kompatibilität und Portabilität von DSSM in Bezug zu SSCM und BS2000/OSD siehe [Seite 28](#).

Die DSSM-Privilegierung für Kommandos

Im Rahmen des BS2000-Privilegienkonzeptes existiert ein systemglobales Privileg für die Ausführung der DSSM-Kommandos. Das Privileg zur Subsystem-Verwaltung wird in Kommandos und Meldungen mit SUBSYSTEM-MANAGEMENT bezeichnet und ist nach First-Startup der Benutzerkennung TSOS zugeteilt (Standard).

Das Privileg kann bei Einsatz von SECOS an jede beliebige Benutzerkennung vergeben werden, die dann die alleinige Berechtigung besitzt, diese Kommandos ausführen zu lassen (siehe auch Handbuch „SECOS“ [36]).

Die nachfolgende Tabelle enthält eine Übersicht mit allen DSSM-Kommandos und den zur Kommandoausführung erforderlichen Privilegien.

Kommando	Bedeutung	nötiges Privileg				Seite
		SM	O	STD	HM	
ADD-SUBSYSTEM	Dynamischen Subsystemkatalog erweitern	X				75
HOLD-SUBSYSTEM	Subsystem in den Wartezustand versetzen	X	X			80
LOAD-LOCAL-SUBSYSTEM-CATALOG	Lokalen Subsystemkatalog laden			X		83
MODIFY-SUBSYSTEM-PARAMETER	Subsystem-Parameter verändern	X				85
RELEASE-SUBSYSTEM-SPACE	Reservierten Adressraum von Subsystemen freigeben	X		X	X	110
REMOVE-SUBSYSTEM	Inaktives Subsystem aus dynamischem Katalog entfernen	X				111
RESUME-SUBSYSTEM	Wartezustand für ein Subsystem aufheben	X	X			113
SAVE-SUBSYSTEM-CATALOG	Änderungen des dynamischen Subsystemkatalogs sichern	X				116
SET-DSSM-OPTIONS	Protokollierungsfunktion von DSSM aus- oder einschalten	X	X			120
SHOW-SUBSYSTEM-ATTRIBUTES	Informationen über Subsystem-Attribute anzeigen	X		X		122
SHOW-SUBSYSTEM-INFO	Informationen über aktuelle Subsystem-Konfiguration anfordern	X				138

Tabelle 6: DSSM-Kommandos und ihre Privilegierung

(Teil 1 von 2)

Kommando	Bedeutung	nötiges Privileg				Seite
		SM	O	STD	HM	
SHOW-SUBSYSTEM-STATUS subsys=*NON-PRIV-CLASS-5 subsys=*ALL / <subsys-name>	Informationen über den Zustand von Subsystemen anfordern	X		X		142
START-LOCAL-SUBSYSTEM	Lokales Subsystem im Benutzeradressraum aktivieren		X	X		150
START-SUBSYSTEM	Subsystem aktivieren	X	X			153
STOP-LOCAL-SUBSYSTEM	Lokales Subsystem im Benutzeradressraum deaktivieren			X		159
STOP-SUBSYSTEM	Subsystem deaktivieren	X	X			162
UNLOAD-LOCAL-SUBSYSTEM-CATALOG	Lokalen Subsystemkatalog entladen			X		166
UNLOCK-SUBSYSTEM	Subsystem im Zustand LOCKED in den Zustand NOT-CREATED überführen	X				168

Tabelle 6: DSSM-Kommandos und ihre Privilegierung

(Teil 2 von 2)

SM: Privileg SUBSYSTEM-MANAGEMENT

Verwaltung von Subsystemen

O: Privileg OPERATING

Eingabe von Bedienplätzen

STD: Privileg STD-PROCESSING

Ausführen von Benutzerkommandos

HM: Privileg HARDWARE-MAINTENANCE

Aufruf von Hardware-Testprogrammen

3.1 Aufgaben und Funktionen von DSSM

Das Softwareprodukt DSSM führt folgende Aktionen durch:

- DSSM baut die gesamte Subsystem-Konfiguration bis zur Funktionsfähigkeit auf.
- DSSM aktiviert und deaktiviert Subsysteme, um die Subsystem-Konfiguration zu erweitern bzw. an die aktuellen Anforderungen anzupassen.
- DSSM überprüft die Verträglichkeit der einzelnen Subsystemversionen beim Aktivieren und Deaktivieren von Subsystemen.
- DSSM stellt die Bezüge zwischen Subsystemen her und überwacht diese beim Aktivieren bzw. hebt vorhandene Beziehungen eines Subsystems zu anderen Subsystemen beim Deaktivieren auf.
- DSSM unterstützt den Anschluss der Tasks und Programme an Subsysteme.
- DSSM unterstützt die Nutzung gemeinsamen Adressraums in Memory Pools.

Alle in den folgenden Punkten als synchron bezeichnete Aktivitäten laufen in der Task des Auftraggebers ab, alle asynchronen Aktivitäten in der Holdertask.

3.1.1 Subsystem-Deklaration (SSC)

Zur Verwaltung der Subsysteme benötigt DSSM Informationen in Form einer so genannten Deklaration. Die Deklaration bestimmt, welche Haupt- und NebenkompONENTEN zu einem Subsystem gehören, wie es aktiviert werden kann und wie die Anschlüsse an das Subsystem hergestellt werden müssen. Die Erstellung der Deklaration erfordert also die Kenntnis des Subsystems. Deshalb werden für alle DSSM-fähigen Subsysteme die dafür notwendigen Deklarationen in einer Deklarationsdatei mit ausgeliefert.

Abhängig von der Deklaration aktiviert DSSM das Subsystem und stellt Anschlüsse und Bezüge zu anderen Subsystemen und zu DSSM her. Anschlüsse an ein Subsystem können von DSSM überwacht werden. Dadurch ist ein Deaktivieren des Subsystems unter Steuerung von DSSM möglich. Werden die Anschlüsse nicht überwacht, erfolgt das Deaktivieren in eigener Verantwortung des Subsystems.

Die Deklarationen werden nicht jedem DSSM-Aufruf als Operanden mitgegeben, da dies zu umständlich und umfangreich wäre. Die Deklarationen werden mit SSCM festgelegt und im Subsystemkatalog abgespeichert, weil:

- diese Deklarationen stabil sind und sich selten ändern
- diese Deklarationen umfangreich und komplex sind
- auf diese Deklarationen häufig von DSSM zugegriffen wird
- nicht nur subsystem-interne, sondern auch subsystem-übergreifende Informationen existieren

Die Deklarationen aller zu einer Konfiguration gehörenden Subsysteme („Deklarations-Zustand“) bilden den Subsystemkatalog SSMCAT und werden in einer PAM-Datei abgelegt. Diese Datei wird beim Startup eingelesen und bleibt bis zur Systembeendigung im Speicher. Der Katalog kann im laufenden System mit dem Kommando ADD-SUBSYSTEM erweitert werden.

Mit dem Kommando LOAD-LOCAL-SUBSYSTEM-CATALOG kann ein lokaler Katalog gelesen und dynamisch in den Benutzeradressraum der aufrufenden Task geladen werden.

3.1.2 Aktivieren bzw. Wiederanlauf

Die **Aktivierung** eines Subsystems wird gesteuert durch die Attribute, die im Subsystemkatalog mit der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES definiert wurden.

Die Aktivierung erfolgt durch eine der folgenden Möglichkeiten:

- a) automatisch während des ersten DSSM-Aufrufs der Startup-Routine, wenn CREATION-TIME=*BEFORE-DSSM-LOAD oder *AT-DSSM-LOAD definiert wurde; die Aktivierung muss dabei erfolgreich abgeschlossen werden (*BEFORE-DSSM-LOAD bedeutet dabei, dass das Subsystem bereits geladen ist, bevor DSSM erstmals durch die Startup-Routine aufgerufen wurde. Nach dem Aufruf von DSSM wird dieses Subsystem wie jedes andere verwaltet. Versionsaustausch oder Versionskoexistenz mit einer anderen Version, die mit CREATION-TIME=*AT-DSSM-LOAD definiert wurde, ist erlaubt. Es liegt in der Verantwortung der Systembetreuung dafür zu sorgen, dass zu jeder Zeit eine Version des Subsystems verfügbar ist.)
- b) automatisch während des zweiten DSSM-Aufrufs der Startup-Routine, wenn CREATION-TIME=*MANDATORY-AT-STARTUP definiert wurde; die Aktivierung muss dabei erfolgreich abgeschlossen werden
- c) automatisch während des zweiten DSSM-Aufrufs der Startup-Routine, wenn CREATION-TIME=*BEFORE-SYSTEM-READY definiert wurde (synchron)
- d) automatisch nach dem zweiten Rücksprung zur Startup-Routine, wenn CREATION-TIME=*AFTER-SYSTEM-READY definiert wurde (asynchron)
- e) explizit durch das Kommando START-SUBSYSTEM
- f) explizit als lokales Subsystem durch das Kommando START-LOCAL-SUBSYSTEM
- g) explizit beim Aufruf der \$ESMCRE-Schnittstelle
- h) implizit nach dem ersten SVC-Aufruf für ein Subsystem, das mit CREATION-TIME=*AT-SUBSYSTEM-CALL(ON-ACTION=*STD oder *ANY) definiert wurde bzw. implizit nach dem ersten ISL-Aufruf für ein Subsystem, das mit CREATION-TIME=*AT-SUBSYSTEM-CALL(ON-ACTION=*ISL-CALL oder *ANY) definiert wurde

Zum Startzeitpunkt für Subsysteme siehe auch die SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES ab [Seite 243](#).

Die Programm- oder Bindemodulbibliothek (OML), die REP- und die „NOREF“-Datei von DSSM müssen ebenso wie die anderen für die DSSM-Initialisierung benötigten Dateien (z.B. der SSMCAT) und die Dateien, die für die Aktivierung von Startup-gebundenen Subsystemen benötigt werden, unter der Kennung TSOS auf dem Home-Pubset eingerichtet und während des Startup verfügbar sein.

Startup-gebunden sind solche Subsysteme, deren Aktivierungszeitpunkt mit den folgenden Werten angegeben wurden:

- *BEFORE-SYSTEM-READY
- *AFTER-SYSTEM-READY
- *BEFORE-DSSM-LOAD
- *AT-DSSM-LOAD
- *MANDATORY-AT-STARTUP

Voraussetzung für das Aktivieren ist in jedem Fall, dass das Subsystem bekannt, d.h. im globalen oder lokalen Subsystemkatalog deklariert ist.

Die Aktivierung läuft in folgenden Stufen ab:

1. Prüfen des Auftrags, insbesondere Abhängigkeiten zu anderen Subsystemen (synchron)

Hat ein Subsystem Komponenten, die mit dem Attribut *INSTALLED deklariert und mit IMON installiert wurden, ruft DSSM während der Prüfphase IMON-GPN auf, um die Pfadnamen der Dateien zu erhalten.

Abhängig von der Verfügbarkeit von IMON-GPN und dem Status der installierten Liefergruppe (siehe Operand INSTALLATION-UNIT) verhält sich DSSM wie folgt:

- a) Die INSTALLATION-UNIT existiert und ein Dateiname kann mit dem angegebenen logischen Namen assoziiert werden:
DSSM wird diesen Dateinamen solange benutzen, wie das Subsystem aktiv ist (bis STOP-SUBSYSTEM) oder bis durch das Kommando MODIFY-SUBSYSTEM-PARAMETER ein anderer Dateiname festgelegt wurde.
- b) In folgenden Situationen greift DSSM auf den im Katalog definierten DEFAULT-NAME zu, wenn vorhanden:
 - IMON-GPN ist nicht verfügbar
 - die INSTALLATION-UNIT existiert nicht in den IMON-GPN-Daten
 - die INSTALLATION-UNIT existiert in den IMON-GPN-Daten, jedoch ohne Verbindung zu einem logischen Namen (LOGICAL-ID)

Die Meldung ESM0665 informiert über die Vorgehensweise:

```
ESM0665 'DEFAULT-NAME' FUER DATEIEN DES SUBSYSTEMS '(&00)' VERWENDET
```

- c) Die INSTALLATION-UNIT existiert in den IMON-GPN-Daten mit einigen zugehörigen Pfadnamen, aber der angefragte logische Name existiert nicht. DSSM nimmt nun an, dass die Datei nicht existiert. Zwei Fälle sind zu unterscheiden:
1. Subsysteme die mit den Attributen *AT-DSSM-LOAD oder *MANDATORY-AT-STARTUP definiert wurden:
Während dem Startup wird eine Frage an der Bedienstation ausgegeben, die beantwortet werden muss. Der Operator kann nun entweder für die erforderliche Datei (Informationsdatei, Modulbibliothek oder Rep-Datei mit dem Attribut REP-FILE-MANDATORY=*YES) einen neuen, gültigen Namen angeben oder die Aktivierung des Subsystems stoppen.
 2. Subsysteme die mit den Attributen *AT-CREATION-REQUEST, *AFTER-SYSTEM-READY oder *BEFORE-SYSTEM-READY definiert wurden:
Fehlt eine der Dateien (Modulbibliothek, Informationsdatei oder Rep-Datei mit dem Attribut REP-FILE-MANDATORY=*YES), wird das Subsystem nicht aktiviert, was durch eine Meldung angezeigt wird. Fehlt die Meldungsdatei oder die Rep-Datei (mit Attribut REP-FILE-MANDATORY=*NO), wird das Subsystem aktiviert; es wird trotzdem eine Warnung ausgegeben.

2. Laden in eine Holdertask (asynchron)

Das Laden des Subsystemcodes wird durch die Holdertask veranlasst. Zusätzlich stellt sie ihren lokalen Adressraum (Benutzeradressraum) für die Subsysteme zur Verfügung, die mit MEMORY-CLASS=*LOCAL-PRIVILEGED/*LOCAL-UNPRIVILEGED definiert wurden. Das Laden von Subsystemen mit MEMORY-CLASS=*SYSTEM-GLOBAL wird zwar ebenfalls von der Holdertask veranlasst, sie werden aber nicht in den Benutzeradressraum der Holdertask, sondern in den gemeinsam benutzbaren Systemadressraum geladen.

Bei der Aktivierung von TU-Subsystemen mit MEMORY-CLASS=*BY-SLICE wird der mehrbenutzbare Programmbereich in den gemeinsam benutzbaren Systemadressraum geladen, der nicht-mehrbenutzbare Programm- und/oder Datenbereich in den Benutzeradressraum der Holdertask.

Stellt eine Task den Anschluss zu einem solchen Subsystem her, wird nur noch der bereits im Benutzeradressraum der Holdertask geladene Datenbereich in die privaten Benutzeradressräume der angeschlossenen Tasks an dieselbe Adresse kopiert. Dadurch sind adressbezogene Referenzen zwischen Programm- und Datenbereich immer möglich. Die Performance wird durch diese Art der Adressraumaufteilung wesentlich erhöht, weil beim Anschluss einer Task an das Subsystem kein Zugriff zur Programm- oder Bindemodulbibliothek erfolgt und Externverweise nicht aufgelöst werden müssen.

Wenn ein Subsystem mit MEMORY-CLASS=*BY-SLICE definiert wurde und erstmals gestartet wird, informiert DSSM das Subsystem BLSSERV darüber, dass mit dem Makro VSVI1 auf die Kopie des Datenbereichs im privaten Benutzeradressraum zugegriffen werden kann.

Das Makro VSVI1 informiert den Anwender über Einträge in den Tabellen des DBL. Einzelheiten zum Makro siehe Handbuch „BLSSERV“ [4].

Die Holdertask kann auch als Arbeitstask dienen.

3. nur für TPR-Subsysteme: Starten der Subsystemaktivierung in der Holdertask, falls eine INIT-Routine definiert ist (asynchron)
4. Nach abgeschlossener Aktivierung Öffnen des Subsystems für Anschlüsse von berechtigten Benutzern (asynchron)

Der **Wiederanlauf** eines Subsystems (RESUME-SUBSYSTEM) nach einem vorhergehenden HOLD-SUBSYSTEM besteht aus den Stufen 1, 3 und 4.

Bei Stufe 1 entfällt das Aufrufen von IMON-GPN.

Wenn die Aktivierung bzw. der Wiederanlauf eines Subsystems explizit durch das Kommando START- bzw. RESUME-SUBSYSTEM erfolgte, kann an Stelle des asynchronen der synchrone Verarbeitungsmodus gewählt werden.

3.1.3 Anschluss herstellen und wieder aufheben

Der **Anschluss** an den Auftragseingang eines Subsystems erfolgt durch eine der folgenden Möglichkeiten:

- a) implizit und global durch Bindebeziehung; dies ist nur zu bereits geladenen Subsystemen oder zwischen gleichzeitig geladenen Subsystemen möglich
- b) implizit und taskspezifisch
 - durch eine subsystemspezifische SVC-, ISL- bISL- bzw. System-Exit-Routine
 - bei nicht-privilegierten Subsystemen durch DBL-Schnittstellen (BIND-Makro, START-PROGRAM, LOAD-PROGRAM und START-<produktname>, Autolink)
- c) explizit und taskspezifisch durch systeminterne Makros (\$ESMCON und \$ESMCCS)

Der Anschluss an ein Subsystem umfasst:

1. Erzeugen einer Subtask, wenn entsprechender systeminterner Aufruf (\$ESMCCS)
2. Anschließen an den Memory Pool, falls erforderlich

3. Wenn ein Subsystem mit MEMORY-CLASS=*BY-SLICE definiert wurde, wird der private Bereich in den lokalen Adressraum (Benutzeradressraum) kopiert. Wird das Subsystem erstmals gestartet, informiert DSSM das Subsystem BLSSERV darüber, dass mit dem Makro VSVI1 auf die Kopie im privaten Benutzeradressraum zugegriffen werden kann.
4. Übergeben der Adresse an die Task oder Aufruf des Subsystemcodes
5. Setzen des subsystemspezifischen Anschlussmerkers
6. Erhöhen des taskspezifischen Anschlusszählers (Ausnahme: bei Definition des Subsystems mit dem Attribut CONNECTION-SCOPE=*FREE)

Es kann auch ein Anschluss für Auftragseingänge hergestellt werden, die das Subsystem selbst verwaltet.

Das **Aufheben** eines Anschlusses erfolgt durch eine der folgenden Möglichkeiten:

- a) implizit und taskspezifisch
 - durch Programm- bzw. Task-Beendigung
 - nach Rücksprung aus dem Subsystem-Auftragseingang, wenn dieser mit dem Attribut MODE=*SVC/*ISL und CONNECTION-SCOPE=*CALL definiert wurde
 - bei der Deaktivierung von Subsystemen mit CONNECTION-SCOPE=*OPTIMAL
- b) explizit und taskspezifisch durch einen systeminternen Makroaufruf bei Beendigung einer SVC-Routine (\$ESMDCN)
- c) implizit und subsystemspezifisch bei Deaktivierung eines Subsystems, das mit dem Attribut CONNECTION-SCOPE=*FREE definiert wurde

Das Aufheben einer Beziehung enthält:

1. Weghängen vom Memory Pool
2. Wenn das Subsystem mit MEMORY-CLASS=*BY-SLICE definiert wurde, wird der Anteil aus dem Benutzeradressraum der angeschlossenen Task entladen. DSSM informiert das Subsystem BLSSERV beim Aufheben der letzten Verbindung darüber, dass nicht mehr auf diesen privaten Anteil zugegriffen werden kann.
3. Zurücksetzen des taskspezifischen Anschlussmerkers
4. Erniedrigen des subsystemspezifischen Anschlusszählers (Ausnahme: bei Definition des Subsystems mit dem Attribut CONNECTION-SCOPE=*FREE)
5. „Function detach“ bei explizitem Aufruf

Die beim Anschluss an die Task gelieferte Adresse kann nicht mehr verwendet werden.

3.1.4 Subsystem deaktivieren oder anhalten

Voraussetzung für diese Funktionen ist generell, dass das Subsystem aktiv ist.

Das **Deaktivieren** eines Subsystems erfolgt durch eine der folgenden Möglichkeiten:

- a) explizit bei Eingabe des Kommandos STOP-SUBSYSTEM (bei HOLD-SUBSYSTEM wird das Subsystem angehalten)
- b) explizit bei Aufruf privilegierter Makros an der Programmschnittstelle (\$ESMDEL und \$ESMHLD)
- c) implizit bei Eingabe des Kommandos START-SUBSYSTEM mit dem Operanden VERSION-PARALLELISM=*EXCHANGE-MODE
- d) automatisch bei Shutdown für alle Subsysteme, bei deren Definition dies vereinbart wurde (STOP-AT-SHUTDOWN=*YES)

Das Deaktivieren eines Subsystems (STOP-SUBSYSTEM) läuft in folgenden Stufen ab:

1. Prüfen des Auftrags, insbesondere der Abhängigkeiten zu anderen Subsystemen (synchron)
2. Starten der CLOSE-CTRL-Routine, sofern eine solche definiert ist (asynchron)
3. Schließen des Subsystems für neue Benutzer (asynchron); so können keine Tasks mehr an das Subsystem angeschlossen werden;
Ausnahme: Einsprungstellen mit CONNECTION-SCOPE=*FREE oder SVC/ISL-Aufrufe für Subsysteme mit Einsprungstellen mit CREATION-TIME=*AT-SUBSYSTEM-CALL. Auf Code, dessen Einsprungstellen mit CONNECTION-SCOPE=*OPTIMAL definiert wurde, kann von diesem Moment an nicht mehr zugegriffen werden.
4. Starten der Auftragsbeendigungs-Routine (STOPCOM-Routine), falls definiert (asynchron)
5. Warten, bis das Subsystem auftragslos ist, d.h. der subsystemspezifische Anschlusszähler gleich null ist und keine Task auf Code zugreift, dessen Einsprungstellen mit CONNECTION-SCOPE=*OPTIMAL definiert wurde (asynchron);
Ausnahme: Wenn im DSSM-Kommando der Operand FORCED=*YES angegeben ist, wird die Deinitialisierung sofort gestartet.
6. Starten der Deinitialisierung, falls definiert (asynchron)
7. Entladen aus der Holdertask (asynchron)

Das **Anhalten** eines Subsystems (HOLD-SUBSYSTEM) besteht aus den Stufen 1 bis 6. Wenn die Deaktivierung bzw. das Anhalten eines Subsystems explizit durch ein Kommando (STOP-/HOLD-SUBSYSTEM) erfolgte, kann an Stelle des asynchronen der synchrone Verarbeitungsmodus gewählt werden.

3.1.5 Austausch von Subsystemversionen

Der Austausch von Versionen eines Subsystems ist auf folgende drei Arten möglich:

- a) Die alte Version wird deaktiviert (STOP-SUBSYSTEM) und die neue Version des Subsystems aktiviert (START-SUBSYSTEM). Es kann u.U. zu sehr langen Ausfallzeiten des Subsystems kommen, weil die neue Version erst dann aktiviert wird, wenn die alte vollständig deaktiviert ist, d.h. alle Tasks die Verbindung abgebaut haben.
- b) Die neue Version des Subsystems wird mit START-SUBSYSTEM ...,VERSION-PARALLELISM=*EXCHANGE-MODE aktiviert. Ab diesem Zeitpunkt werden keine neuen Verbindungen zur alten Version des Subsystems mehr hergestellt. Während die letzten Tasks die Verbindung zur alten Version des Subsystems abbauen, wird die neue Version initialisiert. Dieses Verfahren verkürzt den Zeitraum der Nichtverfügbarkeit des Subsystems erheblich.

Es werden nacheinander folgende Routinen aufgerufen, sofern sie definiert sind:

- STOPCOM-Routine der alten Version
- INIT-Routine der neuen Version
- DEINIT-Routine der alten Version

Für manche Subsysteme ist es problematisch, dass die DEINIT-Routine der alten Version läuft, während die neue Version bereits aktiviert ist und arbeitet. Ein Subsystem, das mit VERSION-EXCHANGE=*FORBIDDEN definiert wurde, kann nicht als neue Version eingetauscht werden. Es kann jedoch (als alte Version) im Austausch gegen eine neue Version, die mit VERSION-EXCHANGE=*ALLOWED definiert ist, deaktiviert werden.

Die alte Version ist so lange im Zustand IN-DELETE, bis keine Task mehr Anschluss hat. Ist die neue Version im Zustand CREATED, ist die Aktivierung des alten Subsystems mit RESET=*YES nur dann möglich, wenn für beide Versionen bei der Definition Koexistenz zugelassen wurde.

Die Aktivierung der alten Version mit RESET=*YES ist dann erlaubt, wenn die neue Version im Zustand IN-DELETE ist, und die alte Version nicht mit VERSION-EXCHANGE=*FORBIDDEN definiert wurde.

- c) Bei Verwendung der CLOSE-CTRL-Routine ist ein Versionsaustausch auch ohne Unterbrechung der Subsystemverfügbarkeit möglich.

Es werden nacheinander folgende Routinen aufgerufen, sofern sie definiert sind:

- CLOSE-CTRL-Routine der alten Version
- INIT-Routine der neuen Version
- STOPCOM-Routine der alten Version
- DEINIT-Routine der alten Version

Läuft die Initialisierungsroutine der neuen Version nicht fehlerfrei, wird die alte Version wieder automatisch aktiviert und ist im Zustand CREATED (das Ergebnis der CLOSE-CTRL-Routine ist reversibel). So wird eine Unterbrechung der Subsystemverfügbarkeit vermieden.

Der Versionsaustausch ist erlaubt, wenn eine Version des Subsystems im Zustand CREATED ist und alle anderen im Katalog deklarierten Versionen des Subsystems im Zustand NOT-CREATED oder LOCKED sind.

Ein Versionsaustausch wird nicht ausgeführt, wenn alle deklarierten Versionen des Subsystems im Zustand NOT-CREATED oder LOCKED sind. In diesem Fall wird die Version aktiviert, die im START-SUBSYSTEM-Kommando angegeben wurde.

Beispiel

Die Subsystemversionen SPOOL V04.2.A und V04.3.A sind mit VERSION-EXCHANGE=*ALLOWED definiert.

```
/SHOW-SUBSYSTEM-STATUS SPOOL,*ALL
SUBSYSTEM SPOOL    /V04.2.A IS NOT CREATED
SUBSYSTEM SPOOL    /V04.3.A IS NOT CREATED

/START-SUBSYSTEM SPOOL,04.2.A,VERSION-PARAL=*EXCHANGE-MODE,SYNCH=*YES
ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'SPOOL /V04.3.A' COMPLETELY
        PROCESSED
ESM0400 'CREATE' OR 'RESUME' SUBSYSTEM 'SPOOL /V04.3.A' WITH
        'SYNCHRONOUS=YES' AND 'RESET=NO'
ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'SPOOL /V04.3.A' COMPLETELY
        PROCESSED
```

Gibt es eine oder mehrere Versionen des Subsystems, die nicht im Zustand NOT-CREATED oder LOCKED sind (außer der auszutauschenden Version im Zustand CREATED), wird der Austausch abgewiesen, auch wenn alle Versionen Koexistenz erlauben.

Beispiel

Die Subsystemversionen UTM V04.0, V05.0 und V05.1 sind mit VERSION-COEXISTENCE=*ALLOWED und VERSION-EXCHANGE=*ALLOWED definiert.

```
/SHOW-SUBSYSTEM-STATUS UTM,*ALL
SUBSYSTEM UTM      /V04.0    IS NOT RESUMED
SUBSYSTEM UTM      /V05.0    IS NOT CREATED
SUBSYSTEM UTM      /V05.1    IS CREATED

/START-SUBSYSTEM UTM,05.0,VERSION-PARALLELISM=*EXCHANGE-MODE
ESM0206 SOME ACTIONS IN PROGRESS FOR SUBSYSTEM 'UTM/V04.0'.
        NO FURTHER ACTION ON ANOTHER VERSION POSSIBLE
ESM0224 REQUESTED FUNCTION 'CREATE' FOR SUBSYSTEM 'UTM/V05.0'
        REJECTED
```

Während des Austauschs der alten durch die neue Version des Subsystems wird auch die Syntaxdatei der neuen Version geladen. Das bedeutet, dass die Syntax der neuen Version auch Kommandos und Anweisungen der alten Version erkennen und ausführen muss. Es muss also darauf geachtet werden, dass die Syntax der neuen Version die der alten Version kompatibel unterstützt.

Es wird empfohlen, den Austausch von Versionen unter Verwendung der CLOSE-CTRL-Routine nur dann auszuführen, wenn die neu zu aktivierende Version auch die höhere Version von beiden ist.

3.1.6 Koexistenz von Versionen

Es gibt die Möglichkeit, temporär oder permanent zwei Versionen eines Subsystems aktiv zu halten, um allen Tasks, die an die alte Version angeschlossen sind, die Fortsetzung ihrer Arbeit zu gewährleisten. Der Koexistenz-Modus muss bei der Definition des Subsystems (SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES ...,VERSION-COEXISTENCE=*ALLOWED) verankert und beim Kommando START-SUBSYSTEM explizit angefordert werden.

Wird bei **temporärer Koexistenz** die Version B eines Subsystems geladen, wenn bereits Version A des Subsystems aktiv ist, werden alle neuen Aufrufer an die Version B angeschlossen. Die Aufträge, die an die Version A angeschlossen sind, werden noch bearbeitet. Nach Bearbeitung aller Aufträge an Version A wird diese automatisch beendet. Bei der Definition ist darauf zu achten, dass die zu ersetzende, „alte“ Version nicht abhängig sein darf von der ersetzenden, „neuen“ Version.

Ist eine **permanente Koexistenz** verschiedener Versionen eines Subsystems gewünscht, muss für jedes dieser Subsysteme das entsprechende Attribut bei der Definition verankert sein und beim Kommando START-SUBSYSTEM dieser Modus explizit angefordert werden.

3.1.7 Beziehungen zwischen Subsystemen

Als zentrale Instanz verwaltet DSSM die Beziehungen der Subsysteme untereinander, wobei verschiedene Beziehungen explizit überwacht werden können.

Auftragsbeziehungen auf ein Subsystem über expliziten oder impliziten Anschluss werden von DSSM eingerichtet und überwacht. Eine Auftragsbeziehung kann immer zwischen einer Task und einem Auftragseingang geknüpft werden.

Adressbeziehungen über Binder und Lader müssen explizit im Operanden REFERENCED-SUBSYSTEMS der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES angegeben werden. Der Operand sollte jedoch nur auf Subsysteme der gleichen „Familie“ (Privilegierungsstufe, Speicherklasse usw.) angewandt werden. Bei entkoppelten Subsystemen sollte die Sys-

tembetreuung nur dynamische Adressmechanismen wählen (SVC, ISL oder System-Exit-Mechanismen für privilegierte Subsysteme, die BLS-Schnittstelle BIND für nicht-privilegierte Subsysteme).

Adressbeziehungen schränken die Entladbarkeit eines Subsystems ein, da in umgekehrter Reihenfolge als beim Laden entladen werden muss. Adressbeziehungen werden beim Laden und Entladen beachtet, aber nicht überwacht. Sie verhindern das Entladen eines Subsystems unabhängig davon, ob Aufträge bestehen oder nicht.

Abhängigkeitsbeziehungen setzen die Verfügbarkeit eines anderen Subsystems voraus. Diese Beziehungen können mit dem Operanden RELATED-SUBSYSTEMS der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES deklariert werden und werden bei der Lade- und Entladereihenfolge beachtet.

3.1.8 Beziehungen zwischen NebenkompONENTEN und Subsystemen

Während der Installation eines Subsystems wird der Bezug zu seinen NebenkompONENTEN hergestellt.

Die **Modulbibliothek** ist eine verpflichtende NebenkompONENTE, da kein Subsystem ohne sie geladen werden kann.

Ist eine **Informationsdatei** oder eine **Syntaxdatei** bei der Definition angegeben, erhalten auch diese den Rang einer Pflicht-KompONENTE, deren Fehlen das Laden des Subsystems verhindert.

Ist eine **Rep-Datei** bei der Definition angegeben, ist diese beim Laden nur dann verpflichtend, wenn dies bei der Definition explizit vereinbart wird (REP-FILE-MANDATORY=*YES).

Eine **Meldungsdatei** kann bei der Definition angegeben werden, ist aber keine Pflicht-KompONENTE, da das Laden des Subsystems trotz fehlender Meldungsdatei durchgeführt werden kann.

In folgenden Situationen sendet DSSM eine Frage an die Bedienstation, die vom Operator beantwortet werden muss:

- die Modulbibliothek eines verpflichtenden Subsystems fehlt (Attribute *AT-DSSM-LOAD oder *MANDATORY-AT-STARTUP)
- die angegebene Informationsdatei ist zwar deklariert aber nicht vorhanden
- die Rep-Datei mit dem Attribut REP-FILE-MANDATORY=*YES fehlt

Der Operator hat trotz einer fehlenden Datei die Möglichkeit, den Startvorgang fortzusetzen (z.B. das Subsystem ohne Informationsdatei zu starten). Die Verantwortung für eine mögliche abnormale Subsystem- oder Systembeendigung liegt ausschließlich beim Operator.

3.1.9 Kommunikation zwischen Subsystem und DSSM

Der Austausch von Informationen und Meldungen ist erforderlich bei subsystemspezifischen Routinen zur Initialisierung, Deinitialisierung, Auftragsbeendigung und Kontrolle der Auftragsbeendigung (INIT-, DEINIT-, STOPCOM- und CLOSE-CTRL-Routinen).

Der Kommunikationsbereich besteht aus einem Informationsbereich für die gestartete Routine (DSSM → Subsystem) und einem Rückmeldungsbereich (Subsystem → DSSM).

Das Starten der Routine erfolgt über

- a) Prozeduraufruf
bei Initialisierung (ohne Bedingung) oder bei Deinitialisierung, Auftragsbeendigung und Kontrolle der Auftragsbeendigung, wenn die Holdertask nicht als Arbeitstask genutzt wird
- b) die bei der Initialisierung übermittelte Schnittstelle (Börse oder FITC)
wenn die Holdertask als Arbeitstask genutzt wird

Die Rückmeldung an DSSM erfolgt über Prozedur-Rücksprung mit Rückmeldungsbereich bei Deinitialisierung, Auftragsbeendigung und Kontrolle der Auftragsbeendigung, wenn die Holdertasks nicht oder nicht mehr als Arbeitstask verwendet wird. Wird die Holdertask als Arbeitstask genutzt, erfolgt die DSSM-Benachrichtigung mittels NOTIFY-Aufruf mit dem Rückmeldungsbereich als Input-Parameter.

3.1.10 Informieren über Subsysteme

Der Benutzer kann über die Kommandos SHOW-SUBSYSTEM-STATUS und SHOW-SUBSYSTEM-INFO den Zustand eines Subsystems abfragen. Darüberhinaus kann er sich einen Überblick über die gesamte Subsystem-Konfiguration verschaffen.

3.1.11 Zustände eines Subsystems

NOT-CREATED

Das Subsystem ist im laufenden System deklariert, wurde aber bisher noch nicht durch ein START-SUBSYSTEM-Kommando aktiviert, oder es ist nach vorhergehender Aktivierung wieder deaktiviert.

Eine Task kann auf dieses Subsystem nicht zugreifen, bevor es aktiviert ist.

IN-CREATE

Das Subsystem wird aktiviert.

Der Lade- und Initialisierungsprozess ist noch nicht abgeschlossen.

Eine Task kann auf dieses Subsystem noch nicht zugreifen.

CREATED

Das Subsystem ist geladen und initialisiert.
Tasks können auf das Subsystem zugreifen.

IN-DELETE

Das Subsystem ist durch ein STOP-SUBSYSTEM-Kommando deaktiviert.
Der Entlade- und Deinitialisierungsprozess ist noch nicht abgeschlossen.
Eine weitere Task kann auf dieses Subsystem nicht mehr zugreifen.
An das Subsystem angeschlossene Tasks werden noch abgearbeitet.

IN-HOLD

Das Subsystem ist durch ein HOLD-SUBSYSTEM-Kommando angehalten.
Der Deinitialisierungsprozess ist noch nicht abgeschlossen.
Weitere Tasks können auf dieses Subsystem nicht mehr zugreifen.
An das Subsystem angeschlossene Tasks werden noch abgearbeitet.

IN-RESUME

Das Subsystem wird durch ein RESUME-SUBSYSTEM-Kommando fortgesetzt.
Der Reinitialisierungsprozess ist noch nicht abgeschlossen.
Eine Task kann auf dieses Subsystem noch nicht zugreifen.

NOT-RESUMED

Das Subsystem ist durch ein HOLD-SUBSYSTEM-Kommando angehalten.
Der Deinitialisierungsprozess ist abgeschlossen.
Keine Task kann auf dieses Subsystem zugreifen, bis ein RESUME-SUBSYSTEM-Kommando angegeben und dessen Ausführung abgeschlossen ist.

LOCKED

Ein nicht behebbarer Fehler ist aufgetreten, während das Subsystem aktiv war bzw. aktiviert, deaktiviert, fortgesetzt oder angehalten wurde.
Ein weiterer Versuch, die entsprechenden Kommandos abzusetzen, wird abgewiesen.

Folgende Situationen können ein Subsystem in den Zustand LOCKED überführen:

- Der Übergang in den Zustand LOCKED ist in der INIT-, DEINIT, STOPCOM-, CLOSE-CTRL-Routine verankert (gilt nur für privilegierte Subsysteme).
- Die Holdertask des Subsystems ist abnormal beendet worden und ein Restart für diese Task ist entweder nicht vorgesehen oder nicht durchführbar (siehe [Seite 72](#)).
- Während des nicht-unterbrechungsfreien Austauschs von Versionen tritt ein Problem beim Deaktivieren der alten Version auf. Unabhängig vom Wert des Operanden RESTART ist das Subsystem im Zustand LOCKED; die Aktivierung der neuen Version wird fortgesetzt.

Beim Aktivieren, Deaktivieren, Anhalten und Fortsetzen eines Subsystems ändert sich der Zustand des Subsystems und erreicht (ausgehend von einem Anfangszustand), einen Zielzustand; z.B. ist der Zielzustand beim Aktivieren des Subsystems der Zustand CREATED. Es gibt verschiedene Anfangszustände für eine Anforderung; z.B. ist das Kommando START-SUBSYSTEM für ein Subsystem akzeptabel, dessen Zustand NOT-CREATED oder IN-DELETE ist, wenn der Operand RESET=*YES gesetzt ist.

Alle Zustände, die ein Subsystem annehmen kann, sind in der folgenden Tabelle zusammengefasst. Ein Zustandswechsel, der durch ein DSSM-Kommando erreicht wird, ist in einer Zeile aufgezeigt. Der Anfangszustand des Subsystems ist mit 1 bezeichnet. Die höchste Ziffer in der Zeile markiert den Zielzustand, der durch das entsprechende DSSM-Kommando erreicht werden kann. Mögliche Zwischenzustände sind ebenfalls aufgezeigt.

DSSM-Kommandos	Zustände eines Subsystems							Operand des DSSM-Kommandos ¹
	NOT-CREATED	IN-CREATE	CREATED	IN-DELETE	IN-HOLD	IN-RESUME	NOT-RESUMED	
START-SUBSYSTEM	1	2	3					---
			3				2	1
	1	2	3	1				RESET=*YES
		2	3		1	2		RESET=*YES
		3			2		1	RESET=*YES
		3			2			RESET=*YES
STOP-SUBSYSTEM	3		1	2				---
	3			2			1	---
	2				1			FORCED=*YES
	2			1				FORCED=*YES
HOLD-SUBSYSTEM			1		2		3	---
					1		2	FORCED=*YES
RESUME-SUBSYSTEM			3			2	1	---
			3			2	1	RESET=*YES
			3		1	2		RESET=*YES

¹ Operand des DSSM-Kommandos, der für den Zustandswechsel gesetzt werden muss

Tabelle 7: Zustände eines Subsystems

3.1.12 Subsystem-Überwachung mittels Monitor-Jobvariablen

Subsysteme können mittels Monitor-Jobvariablen (MONJV) überwacht werden. Die MONJV muss im Kommando START-SUBSYSTEM angegeben werden, siehe [Seite 153](#). DSSM verwaltet und setzt die MONJV während der gesamten Laufzeit des Subsystems bis zu seiner Beendigung bei:

- expliziten DSSM-Aufrufen (HOLD- /RESUME- /STOP- /UNLOCK-SUBSYSTEM)
- impliziten Operationen (Subsystem, automatische Wiederherstellung ...)
- SHUTDOWN

Die MONJV zeigt an, ob das Subsystem aktiv, angehalten, unterbrochen oder gesperrt ist. Die MONJV kann folgende Inhalte haben:

Byte	Länge	Inhalt	Werte
1	3	Status	\$R (running) / \$A (abnormal end) / \$L (loaded) / \$T (terminate)
4	1	reserviert	0
5	4	TSN	???? (4 Fragezeichen)
9	4	Home Catid	
13	4	reserviert	
17	1	Typ	J / P / S
18	53	reserviert	
71	3	Session-Nummer	
74	8	Name des Subsystems	
82	7	Version des Subsystems	
89	15	Zustand des Subsystems	für \$R: created für \$A: abnormal end / locked für \$L: in create für \$T: not created / not resumed / in delete / in resume / in hold
104	24	unbenutzt	
128	127	reserviert für Anwender des Subsystems	

Tabelle 8: Inhalte der Monitor-Jobvariablen

3.1.13 Funktionsübersicht

Die Tabelle gibt einen Überblick über die von DSSM/SSCM angebotenen Funktionen und welche Operanden für die einzelnen Subsystem-Klassen angegeben werden können.

Funktion (Operanden)	TPR	TU		Sys-	Share	PK
	SAR	SAR	BAR	Exits	Prod.	
Subsystem-Deklaration mit der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES						
Identifikation SUBSYSTEM-NAME, VERSION,- DYNAMIC-CHECK-ENTRY	X	X	X	X	X	
Binden und Laden LIBRARY, REP-FILE, LINK-ENTRY, AUTOLINK,- UNRESOLVED-EXTERNALS	X	X	X	X	X	
CREATION-TIME = *BEFORE-DSSM-LOAD/*AT-DSSM-LOAD/ *BEFORE-SYSTEM-READY	X			X		
= *AFTER-SYSTEM-READY/ *AT-CREATION-REQUEST	X	X	X	X	X	
= *AT-SUBSYSTEM-CALL	1					
REFERENCED-SUBSYSTEMS	2	2	2	2	2	
Adressraum bestimmen MEMORY-CLASS = *SYSTEM-GLOBAL = *LOCAL-PRIVILEGED = *LOCAL-UNPRIVILEGED = *BY-SLICE	X	X		X	10	
			X		10	
			X		10	
		X	X		10	
SUBSYSTEM-ACCESS = *SYSTEM = *HIGH	X		X	X		
SIZE			X		X	
START-ADDRESS			3		3	
Nebenkomponenten MESSAGE-FILE, SYNTAX-FILE	X	X	X	8	8	
SUBSYSTEM-INFO-FILE	X			8		
Starten und Beenden INIT-, STOPCOM-, DEINIT-, CLOSE-CTRL-Routine,- INTERFACE-VERSION	X			X		

Tabelle 9: DSSM-Funktionsübersicht

(Teil 1 von 2)

Funktion (Operanden)	TPR	TU		Sys-	Share	PK
	SAR	SAR	BAR	Exits	Prod.	
Holdertask zum Ablauf RESTART-REQUIRED	X			X		
Ablauf (Operand SUBSYSTEM-ENTRIES=(...)) MODE = *LINK = *SVC/*SYS-EXIT/*ISL CONNECTION-ACCESS = *SYSTEM = *ALL CONNECTION-SCOPE = *PROGRAM/*TASK/*FREE = *CALL/*OPTIMAL	X X X 9	X X X X	X X X X	X X X X	X X X X	
Subsystem-Konfiguration mit den SSCM-Anweisungen						
ASSIGN-HOLDER-TASK	X	4	X		8	
SET-SUBSYSTEM-ATTRIBUTES (Operand RELATED-SUBSYSTEMS)	X	X	X	X	X	
SET-SUBSYSTEM-ATTRIBUTES (Operand REFERENCED-SUBSYSTEMS)	11	11	11	11	11	
SEPARATE-ADDRESS-SPACE			5		5	
Subsystem-Konfiguration mit den DSSM-Kommandos						
START-SUBSYSTEM, STOP-SUBSYSTEM,- HOLD-SUBSYSTEM, RESUME-SUBSYSTEM	X	X	X	X	X	
ADD-/REMOVE-/UNLOCK-SUBSYSTEM,- SAVE-SUBSYSTEM-CATALOG						X
MODIFY-SUBSYSTEM-PARAMETER	X	X	X	X	X	X
Subsystem-Information mit den DSSM-Kommandos						
SHOW-SUBSYSTEM-ATTRIBUTES						X
SHOW-SUBSYSTEM-INFO						X
SHOW-SUBSYSTEM-STATUS	X	X	X	X	X	
globale Subsystem-Verwaltung mit den DSSM-Kommandos						
RELEASE-SUBSYSTEM-SPACE			6			
SET-DSSM-OPTIONS	7	7	7	7	7	7

Tabelle 9: DSSM-Funktionsübersicht

Erläuterung

TPR SAR	privilegierte Subsysteme (nur Systemadressraum)
TU SAR	nicht-privilegierte Subsysteme mit Systemadressraum
TU BAR	nicht-privilegierte Subsysteme mit Benutzeradressraum
Sys-Exits	relevant für System-Exits
Share-Prod.	relevant für Share-Produkte
PK	Privilegierte Benutzererkennung für die Systembetreuung (Kennung mit dem Privileg SUBSYSTEM-MANAGEMENT)
X	Funktion verfügbar
1	nur für Subsysteme mit SVC- und/oder ISL-Anschluss
2	Beziehungen nur von BAR nach SAR
3	nur für Subsysteme im Klasse-6-Speicher
4	im Systemadressraum ist die Holdertask nur für Abläufe erforderlich
5	nur für Subsysteme im Adressraum-Streifen nötig
6	Reservierung für den nicht-privilegierten bzw. beide Adressraum-Streifen kann aufgehoben werden
7	nur für Diagnose und Test
8	nur soweit sinnvoll
9	nur in Verbindung mit MODE=*SVC bzw. MODE=*ISL
10	je nach Ablage im Adressraum
11	sollte für Subsysteme der gleichen „Familie“ reserviert werden (Privilegierung, Speicherklasse usw.)

3.2 Speicher- und Taskkonzept

Speicherkonzept

Subsysteme werden je nach Deklaration folgendermaßen geladen:

- in den Systemadressraum (Klasse-3- oder Klasse-4-Speicher) bei MEMORY-CLASS=*SYSTEM-GLOBAL
- in den Benutzeradressraum (Memory Pool: Klasse-5- oder Klasse-6-Speicher) bei MEMORY-CLASS=*LOCAL-PRIVILEGED/*LOCAL-UNPRIVILEGED
- in beide Adressräume bei MEMORY-CLASS=*BY-SLICE

Anschlüsse an Subsysteme im Benutzeradressraum sind nur möglich, wenn der zugewiesene Adressraum auch in der eigenen Task frei ist. Der Code muss nach dem Anschluss immer an der gleichen Adresse liegen. Für allgemein verfügbare Subsysteme im Benutzeradressraum, die jederzeit und von allen Tasks angesprochen werden können, wird ein fester Bereich im Klasse-5-Speicher reserviert, der so genannte Adressraum-Streifen. Diesen Adressraum-Streifen teilen sich alle allgemein verfügbaren Subsysteme, wobei für nicht-privilegierte Subsysteme je ein eigener Streifen zur Verfügung steht.

Für Benutzertasks können bei nicht ausreichendem Klasse-5-Speicher Probleme auftreten. Der für Subsysteme reservierte Adressraum-Streifen kann durch das DSSM-Kommando RELEASE-SUBSYSTEM-SPACE freigegeben werden, um mehr Klasse-5-Speicher zur Verfügung zu haben.

Subsysteme, die nicht gleichzeitig gebraucht werden (die sich also nicht gegenseitig aufrufen), können im Adressraum parallel liegen. Je mehr Subsysteme parallel liegen, umso kleiner ist der benötigte Adressraum-Streifen. Dabei ist auf die Ausgewogenheit der parallelen Subsysteme zu achten, da hohe Parallelität einerseits eine Einsparung des Adressraums, andererseits eine mögliche Verschlechterung der Performance zur Folge haben kann (je mehr Parallelität, umso mehr gegenseitige Verdrängung). Die Verteilung der Subsysteme im Adressraum-Streifen kann mit der SSCM-Anweisung SEPARATE-ADDRESS-SPACE geregelt werden.

Adressraum-Haushalt

Der Adressraum-Haushalt von DSSM gibt die Möglichkeit, den Systemadressraum (Klasse-3- und Klasse-4-Speicher) zu entlasten, indem Subsysteme in den Benutzeradressraum der Holdertask gelegt werden. Das ist jedoch nur für nicht-privilegierte Subsysteme relevant, denn alle privilegierten Subsysteme werden immer in den Systemadressraum geladen (MEMORY-CLASS=*SYSTEM-GLOBAL).

Die Verlagerung ist dann problemlos, wenn die Subsysteme sich nicht gegenseitig aufrufen und nicht über ein drittes Programm voneinander abhängen oder im Klasse-6-Speicher als Hauptprogramm ablaufen können.

Zu beachten ist, dass nur die Verlagerung nicht-privilegierter Subsysteme angestrebt werden kann, die unterhalb der 16 MByte-Grenze angesiedelt sind.

Für alle anderen nicht-privilegierten Subsysteme wird das Laden oberhalb der 16 MByte-Grenze empfohlen (SUBSYSTEM-ACCESS=*HIGH).

Auch Subsysteme, die reentrant-fähig sind und als Hauptprogramme laufen, können aus dem Systemadressraum ausgelagert werden. Sie müssen durch MEMORY-CLASS=*LOCAL-UNPRIVILEGED im Memory Pool im Klasse-6-Speicher verfügbar sein.

Für Subsysteme, die aus einem mehrbenutzbaren Code (Programmbereich) und einem nicht-mehrbenutzbaren Code (Datenbereich) bestehen, gibt es das Konzept zur Minimierung der Systemadressraum-Belegung:

Der Programmbereich wird in den gemeinsam benutzbaren Adressraum geladen (das entspricht MEMORY-CLASS=*SYSTEM-GLOBAL). Der Datenbereich wird in den Benutzeradressraum der Holdertask geladen und zum Zeitpunkt des Anschlusses einer Task an das Subsystem in deren privaten Benutzeradressraum an dieselbe Adresse kopiert.

Dieses Konzept wird mit der Definition eines Subsystems mit MEMORY-CLASS=*BY-SLICE realisiert.

Die Vorteile dieses Konzepts sind:

- Adressbezogene Referenzen zwischen Programm- und Datenbereich sind immer möglich, weil die Kopie des Datenbereichs an derselben Adresse beginnt wie das Original.
- Die Performance wird durch diese Art der Adressraumaufteilung wesentlich erhöht, weil beim Anschluss einer Task an das Subsystem kein Zugriff zur Programm- oder Bindebibliothek erfolgt und Externverweise nicht aufgelöst werden müssen.

Die Nachteile dieses Konzepts sind:

- Der zum Startup-Zeitpunkt bestimmte und vorreservierte Adressraum für den Datenbereich kann nur sehr begrenzt erweitert werden. Eine Speicherplatzoptimierung ist wegen der strengen Adressraumaufteilung nicht möglich.
- Ist der zur Aufnahme des Datenbereichs vorgesehene Adressraum der sich anschließenden Task bereits belegt, wird der Subsystemcode (Daten- und Programmbereich!) vollständig in den Benutzeradressraum dieser Task geladen.

Die Verlagerung vom Systemadressraum in den Benutzeradressraum lohnt sich nur in folgenden Fällen:

- a) Subsysteme, die voneinander unabhängig sind und von Benutzerprogrammen nicht gleichzeitig benutzt werden, können parallel in den Benutzeradressraum gelegt werden. Ist dies nicht der Fall, ist eine Verlagerung nicht sinnvoll, da der Verbrauch im Benutzeradressraum ungünstiger ausfällt als im Systemadressraum.
- b) Die Subsysteme sind ausreichend groß, um den auftretenden Verschnitt durch die Verwendung von Memory Pool auszugleichen (min. Größe eines Memory Pools: 1 MByte).
- c) Das Subsystem muss unterhalb der 16 MByte-Grenze angesiedelt sein. (Für Subsysteme, die auch oberhalb der Grenze geladen werden dürfen, kann der Operand `SUBSYSTEM-ACCESS=*HIGH` verwendet werden, um einer Überlastung der unteren 16 MByte zu begegnen.)

Zusammenfassung:

Unabhängige Subsysteme parallel in Memory Pools zu konfigurieren ist nur dann sinnvoll, wenn die Summe der Größe aller Subsysteme mehr als 1 MByte beträgt und kein Subsystem die 1 MByte-Grenze in seiner Größe überschreitet.

Taskkonzept - Holdertask

Die Aktivierung eines Subsystems erfolgt unter einer eigenen Task, der Holdertask. Abhängig vom Typ des Subsystems kann diese Task als Subsystem-Arbeitstask oder nur als reine Holdertask verwendet werden. Der Benutzeradressraum dieser Task kann für die Auslagerung aus dem Systemadressraum genutzt werden.

Die Anzahl der benötigten Holdertasks sollte möglichst gering gehalten werden. Eine hohe Taskanzahl beeinflusst zwar die Parallelität günstig, da umso mehr Subsysteme installiert werden können, je mehr Tasks gleichzeitig angelegt werden. Andererseits benötigen mehr Tasks auch mehr taskspezifische Tabellen.

DSSM selbst minimiert die Anzahl der Holdertasks; die Verteilung der Subsysteme kann aber durch die SSCM-Anweisung `ASSIGN-HOLDER-TASK` (siehe [Seite 193](#)) bei der Generierung des Subsystemkatalogs mit SSCM gesteuert werden.

Standardmäßig werden alle Subsysteme, die mit `MEMORY-CLASS=*BY-SLICE` definiert wurden, an dieselbe Holdertask angeschlossen.

Im Fehlerfall wird automatisch der Wiederanlauf der Holdertask initiiert, um nicht alle Subsysteme, die an die Holdertask angeschlossen sind, in Mitleidenschaft zu ziehen. Gesteuert durch den Operanden `RESTART-REQUIRED` (Anweisung `SET-SUBSYSTEM-ATTRIBUTES`) ist auch der Wiederanlauf eines Subsystems vorgesehen. Der Operand ermöglicht, die Subsystem-Initialisierung dann noch einmal aufzurufen, wenn die Holdertask während des Ablaufs von Subsystem-Routinen beendet wurde (siehe [Seite 72](#)).

3.3 Verwaltung von Share-Programmen

DSSM unterstützt die Verwaltung von Share-Programmen. Dies ermöglicht im Systemlauf:

- das Entladen von Share-Programmen auch aus dem Klasse-4-Speicher
- das Auslagern von Share-Programmen vom Klasse-4-Speicher in den Klasse-5- bzw. Klasse-6-Speicher

Share-Programme können als Subsysteme deklariert und als solche von DSSM verwaltet werden. Sie können also wie andere Subsysteme dynamisch aktiviert, deaktiviert, angehalten und fortgesetzt werden.

Voraussetzung ist, dass die Share-Programme bei der Generierung des Subsystemkataloges deklariert worden sind. Mit dieser Definition kann nach erfolgter Inbetriebnahme des DSSM ein Share-Programm ebenso aktiviert und deaktiviert werden wie ein Subsystem.

Die Benutzer können auf das Programm über den BIND-Makro bzw. mit den Kommandos START-PROGRAM und LOAD-PROGRAM zugreifen.

Der Einsprungpunkt muss immer mit der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES, ...SUBSYSTEM-ENTRIES= angegeben werden, auch wenn er mit dem LINK-ENTRY (der für das Laden des Subsystems verwendete Bezugsadresse) identisch ist.

3.4 Verwaltung des dynamischen Subsystemkatalogs

Erweitern des dynamischen Subsystemkatalogs

Während des Systemlaufs kann die aktuelle dynamische Subsystem-Konfiguration von der Systembetreuung erweitert werden (Kommando ADD-SUBSYSTEM). Der Katalog für die neue Subsystem-Konfiguration muss mit SSCM generiert werden und es müssen folgende Punkte beachtet werden:

- Der neue Katalog kann wahlweise größer als der alte sein (d.h. auch alle Subsysteme des alten Kataloges enthalten) oder als „Delta“-Katalog angelegt sein und lediglich die neuen Definitionen von Subsystemen enthalten.
- Der „alte“ Subsystemkatalog, der während Startup benutzt wurde, wird nicht automatisch aktualisiert. Beim nächsten Startup ergeben sich folgende Möglichkeiten:
 - der „neue“ Katalog wird benutzt
 - ein ganz neu erstellter Katalog wird benutzt, in dem z.B. die nicht mehr benutzten Versionen von Subsystemen nicht mehr enthalten und Änderungen von Eigenschaften (wie z.B. CREATION-TIME) bereits vorgenommen sind
- Anweisung ASSIGN-HOLDER-TASK

Die Anweisung darf nicht für alte und neue Subsysteme gegeben werden.

Beispiel

Subsysteme im alten Katalog: A, B, C

Subsysteme im neuen Katalog: A, B, C, D und E

dann ist erlaubt:

```
//ASSIGN-HOLDER-TASK TYPE=SHARED-HOLDER(BY-SUBSYSTEMS=(A,B))
```

```
//ASSIGN-HOLDER-TASK TYPE=SHARED-HOLDER(BY-SUBSYSTEMS=(D,E))
```

dann ist nicht erlaubt:

```
//ASSIGN-HOLDER-TASK TYPE=SHARED-HOLDER(BY-SUBSYSTEMS=(A,B,E))
```

- Anweisung SET-SUBSYSTEM-ATTRIBUTES

Der Operand CREATION-TIME für ein neues Subsystem muss verträglich sein mit den bereits existierenden Versionen des Subsystems in der alten Subsystem-Konfiguration.

Beispiel

Subsystem X in der Version 1 ist im alten Katalog deklariert mit CREATION-TIME=*AT-SUBSYSTEM-CALL. Dann kann Subsystem X in der Version 2 im neuen Katalog nur CREATION-TIME=*AT-CREATION-REQUEST haben.

Falls für ein neues Subsystem CREATION-TIME=*BEFORE-SYSTEM-READY oder *AFTER-SYSTEM-READY ist, dann wird das Subsystem nicht kreiert, da der Zeitpunkt „SYSTEM READY“ (Systemeinleitung) nicht mehr gegeben ist. Es erfolgt eine entsprechende Meldung.

Neu hinzuzufügende Subsysteme mit den Attributen MEMORY-CLASS=*LOCAL-PRIVILEGED (Klasse-5-Speicher) dürfen weder die Breite des System- bzw. Benutzer-Adressraum-Streifens noch die Lage der alten Subsysteme in diesem Streifen ändern.

Es dürfen keine Abhängigkeiten von alten zu neuen Subsystemen deklariert werden (Operanden REFERENCED-SUBSYSTEM und RELATED-SUBSYSTEM).

Beispiel

Subsysteme im alten Katalog: A, B, C
 Subsysteme im neuen Katalog: A, B, C, D und E

Wenn $X \rightarrow Y$ bedeutet, dass X zur Auflösung von Extern-Aufrufen Y benötigt,

dann ist $A \rightarrow D$ nicht erlaubt,
 $D \rightarrow E$ erlaubt,
 $D \rightarrow B,C$ erlaubt.

- Anweisung SEPARATE-ADDRESS-SPACE

Neue und alte Subsysteme dürfen disjunkt sein, sich also adressmäßig überschneiden.

Beispiel

Subsysteme im alten Katalog: A, B, C
 Subsysteme im neuen Katalog: A, B, C, D und E

Wenn $X \rightarrow Y$ bedeutet, dass X zu Y disjunkt ist,

dann ist $A \rightarrow D$ erlaubt,
 $D \rightarrow E$ erlaubt,
 $D \rightarrow B,C$ erlaubt.

Dynamische Änderungen von Subsystem-Attributen

Mit dem Kommando `MODIFY-SUBSYSTEM-PARAMETER` ist es möglich, Eigenschaften von Subsystemen dynamisch zu ändern. Es ist weder ein Shutdown noch das Hinzufügen einer neuen Version des Subsystems nötig. Mit dem Kommando `SAVE-SUBSYSTEM-CATALOG` werden die Konfigurationsänderungen für den nächsten Startup abgespeichert.

Diese Möglichkeiten sind vor allem in folgenden Situationen hilfreich:

- Um ein Subsystem stoppen zu können, müssen alle angeschlossenen Tasks ihre Verbindung abgebaut haben. Wurde das Subsystem nicht mit dem Attribut `FORCED-STATE-CHANGE=*ALLOWED` definiert, besteht keine Möglichkeit, das Deaktivieren des Subsystems zu beschleunigen. Ist dies aber aus bestimmten Gründen nötig, kann mit dem Kommando `MODIFY-SUBSYSTEM-PARAMETER` das Attribut dynamisch geändert werden.
- Kommt es zu einer Memory-Pool-Kollision, weil z.B. zwei Subsysteme an derselben Adresse liegen und die Task nur eins von beiden nutzen kann, ist es mit dem Kommando `MODIFY-SUBSYSTEM-PARAMETER` möglich, die Adressen der Subsysteme zu ändern und so die Memory-Pool-Kollision zu vermeiden.

3.5 Inbetriebnahme der dynamischen Subsystemverwaltung

Die dynamische Subsystemverwaltung wird während der BS2000-Systemeinleitung gestartet.

Über den Parameterservice werden alle zur Initialisierung von DSSM notwendigen Informationen mitgegeben. Dazu zählen der Name des statischen Subsystemkatalogs und die DSSM-Versionsnummer. Des Weiteren kann bereits - wenn unbedingt erforderlich - die Protokollierung von DSSM-spezifischen Daten für die Fehlerdiagnose eingeschaltet werden.

Alle Sätze, die über den Parameterservice verarbeitet wurden, werden in Form von Meldungen in der Protokolldatei CONSLOG erfasst.

Das Schlüsselwort für die Inbetriebnahme der Subsystemverwaltung ist **DSSM**.

Der Ablauf der Inbetriebnahme von DSSM über den Parameterservice wird auch im Handbuch „Einführung in die Systembetreuung“ [14] beschrieben.

Für den späteren Betrieb von DSSM V4.0 werden folgende Dateien unter der Benutzerkennung TSOS auf dem Home-Pubset vorausgesetzt:

SYSLNK.DSSM.040 Bibliothek mit Lademodulen für DSSM

SYSNRF.DSSM.040 Verweisdatei für DSSM-Rep-Datei-Verarbeitung (NOREF-Datei)

SYSREP.DSSM.040 Rep-Korrekturdatei für DSSM

SYSSDF.DSSM.040 SDF-Syntaxdatei

SYSTEMS.DSSM.040 Meldungsdatei

Der zu erzeugende Subsystemkatalog muss ebenfalls auf das Home-Pubset gebracht und unter der Benutzerkennung TSOS abgelegt werden. Der Name des Katalogs ist frei wählbar und kann über den Parameterservice dem System bekannt gemacht werden.

Für die Generierung eines Subsystemkatalogs steht SSCM zur Verfügung, das ab Seite [179](#) ausführlich beschrieben wird.

Problembehandlung beim Systemlauf

Wenn DSSM nicht initialisiert werden kann, wird in einer Meldung der Grund dafür angezeigt (z.B. fehlender statischer Subsystemkatalog) und die Meldung ESM0401 ausgegeben. Der Operator kann während des Systemlaufs an der Bedienstation einen neuen statischen Subsystemkatalog angeben, wenn er in der Parameterdatei nicht festgelegt wurde oder der Standard-Katalog (SYS.SSD.CAT.X) nicht vorhanden ist.

Der Systemlauf wird im Allgemeinen nicht fortgesetzt, wenn verpflichtende Subsysteme nicht in Betrieb genommen werden können. Der Grund für den Fehler bei der Initialisierung von DSSM muss erst behoben werden. Anschließend ist der Systemlauf zu wiederholen.

Wenn Subsysteme mit dem Attribut *BEFORE-SYSTEM-READY nicht in Betrieb genommen werden können, setzt DSSM den Systemlauf fort.

Wenn eine der folgenden Dateien eines Subsystems mit den Attributen *AT-DSSM-LOAD oder *MANDATORY-AT-STARTUP fehlt, kann der Operator während des Systemlaufs einen neuen, gültigen Dateinamen angeben:

- Informationsdatei
- Rep-Korrekturdatei
- Modulbibliothek

Wird vom Operator kein neuer Dateiname für die fehlende Rep-Korrekturdatei oder die Modulbibliothek eingegeben, wird der Systemlauf gestoppt.

Wenn die Informationsdatei fehlt kann der Operator

- einen neuen, gültigen Dateinamen eingeben
- die Meldung ignorieren und den Systemlauf fortsetzen
- die Meldung ignorieren und den Systemlauf stoppen.

Format des Parametersatzes zur Inbetriebnahme von DSSM

Format	Bedeutung
SSMCAT=name	Name des statischen Subsystemkatalogs
VERSION=versnr	Versionsnummer von DSSM
LOGGING=ON / OFF	Steuert die DSSM-spezifische Protokollierung zur Fehlerdiagnose
REPROFILE=<repfile name>	Name der REP-Korrekturdatei für das Laden von DSSM

Für Subsysteme, die nicht während der Systemeinleitung automatisch aufgebaut werden, müssen im BS2000-Systemlauf START-SUBSYSTEM-Kommandos gegeben werden.

Ausschnitt aus der Parameterdatei

```

/BS2000 PARAMS
:
/BEGIN DSSM
SSMCAT=name _____ (1)
VERSION=versnr _____ (2)
LOGGING=ON / OFF _____ (3)
REPROFILE=repfile name _____ (4)
/EOF
:
/END-PARAMS

```

- (1) Die Steuer- und Parametersätze müssen in der Parameterdatei nur vorhanden sein, wenn die Systembetreuung von folgenden Standardwerten abweichen will:
- | | |
|-------------------------|---|
| für BS2000/OSD-BC V5.0: | SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=040 |
| für BS2000/OSD-BC V4.0: | SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=040
oder
SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=039
oder
SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=038 |
| für BS2000/OSD-BC V3.0: | SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=040
oder
SSMCAT=\$TSOS.SYS.SSD.CAT.X und VERSION=036 |
- (2) Die Versionsnummer bezieht sich auf alle DSSM-spezifischen Dateinamen (z.B. SYSLNK.DSSM.040, SYSREP.DSSM.040 bei BS2000/OSD-BC V5.0).
- (3) Die Anweisung LOGGING=OFF schaltet die Protokollierung aus. Mit LOGGING=ON wird bereits während der Inbetriebnahme von DSSM ein Protokoll der Diagnosedaten erzeugt.
Wenn nur Standardwerte angegeben werden, ist das LOGGING nicht möglich.
- (4) Name der gewünschten REP-Korrekturdatei für das Laden von DSSM.
Ist der Parameter nicht angegeben, wird DSSM mit dem Standardnamen der Rep-Korrekturdatei geladen (SYSREP.DSSM.*version*).

3.6 Abrechnungssätze des DSSM (Accounting)

Allgemeines zum Abrechnungssystem des BS2000/OSD

Das gesamte Abrechnungssystem des BS2000/OSD wird von der Systembetreuung gesteuert. Diese bestimmt den Zeitpunkt, an dem das Abrechnungssystem gestartet werden soll, vereinbart den Namen der Abrechnungsdatei und legt Namen und Anzahl der Abrechnungssätze und Satzerweiterungen fest, die in der Abrechnungsdatei erfasst werden sollen.

Die Systembetreuung legt ferner den Zyklus und den Umfang der periodischen Erfassung fest, die bestimmte Abrechnungssätze und Jobklassen umfasst.

Mit dem Abrechnungssystem kann die Systembetreuung dynamisch Abrechnungssätze ganz oder teilweise ein- und ausschalten und Einfluss auf den Umfang der einzelnen Abrechnungssätze nehmen.

Mit dem Kommando MODIFY-ACCOUNTING-PARAMETERS können nicht benötigte Abrechnungssätze und Satzerweiterungen ausgeschaltet werden.

Die bei der Darstellung pro Datenfeld verwendeten Attribute sind:

Feld	laufende Nummer des Datenfeldes innerhalb des beschriebenen Satzteils
Distanz	relativer Abstand des Datenfeldes zum Anfang des beschriebenen Satzteils
Länge	Länge des Datenfeldes in Byte
Format	A = alphanumerisch
	B = Binärzahl
	B2 = binäre Darstellung der CPU-Zeit
	C = abdruckbare Zeichen, inklusive Sonderzeichen
	F = Dateiname
	X = nicht abdruckbare Zeichen
	Z = ungepackte Dezimalzahl (*)
	* = wird bei den einzelnen Satzarten oder Erweiterungselementen festgelegt
	- = reserviert für künftige Erweiterungen, enthält entweder Leerzeichen oder binär null
(*)	die Uhrzeit wird in der Form hhmmss, das Datum in der Form jjmmtt dargestellt

Ein Abrechnungssatz besteht aus folgenden vier Teilen:

(A) Satzbeschreibung	:	Satzkennung, Zeitstempel, ...
(B) Kennzeichnungsteil	:	Benutzerkennung, Abrechnungsnummer, überwachte Betriebsmittel, ...
(C) Grundinformation	:	Standarddaten
(D) Variable Information	:	Satzerweiterungen

Für nähere Informationen zum Abrechnungssystem des BS2000/OSD siehe Handbuch „Einführung in die Systembetreuung“ [14] oder „RZ-Taschenbuch, Teil 1“ [41].

Subsystem-Satzbeschreibung

Die Satzbeschreibung (A) enthält eine Satznummer, die die einzelnen Abrechnungssätze voneinander abgrenzt, einen Zeitstempel sowie Angaben zur Länge des Kennzeichnungsteils und der Grundinformation.

Adresse der Satzbeschreibung = Satzanfang

Aufbau und Inhalt:

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	0	4	A	Satznummer ¹
2	04	4	8	B	Zeitstempel der Tageszeituhr ²
3	0C	12	2	B	Länge des Kennzeichnungsteils
4	0E	14	2	B	Länge der Grundinformation
5	10	16	4	-	- reserviert -

¹ Die Satznummer dient zur Unterscheidung der einzelnen Satzarten.

² Der Zeitstempel wird vom System als letzte Information in den Satz eingetragen, nachdem der Satz fertig erstellt bzw. dem Abrechnungssystem übergeben wurde.

Länge der Satzbeschreibung: 20 Byte

Subsystem-Kennzeichnung

Die Subsystem-Kennzeichnung im Kennzeichnungsteil (B) beschreibt das Subsystem, auf das sich die Subsystem-Abrechnungsdaten beziehen.

Aufbau und Inhalt:

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	0	8	A	Name des Subsystems
2	08	8	7	A	Version des Subsystems
3	0F	15	8	Z	Datum des Aufrufs ¹
4	17	23	6	Z	Uhrzeit des Aufrufs ²

¹ Datum in der Form `yyyymmdd`

² Uhrzeit in der Form `hhmms`

Länge der DSSM-Kennzeichnung: 29 Byte

ESMC - Subsystem-Initialisierungs-Abrechnungssatz

Der Abrechnungssatz wird bei jedem Ablauf der Initialisierungsphase eines Subsystems geschrieben.

Diese Aktivierungsroutine durchläuft das Subsystem, unter Steuerung von DSSM, bei den Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM.

Maximale Länge des Subsystem-Initialisierungs-Abrechnungssatzes: 54 Byte

(A) Satzbeschreibung: Satzkennung: „ESMC“

(B) Kennzeichnungsteil: Subsystem-Kennzeichnung

(C) Grundinformation

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	0	1	B	Zustandsanzeige ¹
2	01	1	1	A	Kennzeichen Jahreszeit (aktuell) ²
3	02	2	1	-	- reserviert -

¹ Die Zustandsanzeige kann zwei mögliche Werte annehmen:

- 1 Subsystem wird nach Wartezustand wieder gestartet (Kommando RESUME-SUBSYSTEM)
- 0 Subsystem ist gestartet (Kommando START-SUBSYSTEM)

² „S“ für Sommerzeit; „W“ für Winterzeit

Länge der Grundinformation: 3 Byte

(D) Variable Information

Die variable Information des Subsystem-Initialisierungs-Abrechnungssatzes enthält **keine** Satzerweiterung.

ESMD - Subsystem-Beendigungs-Abrechnungssatz

Der Abrechnungssatz wird bei jedem Ablauf der Deinitialisierungsphase eines Subsystems geschrieben.

Diese Routine zur Beendigung durchläuft das Subsystem, unter Steuerung von DSSM, bei den Kommandos STOP-SUBSYSTEM und HOLD-SUBSYSTEM.

Maximale Länge des Subsystem-Beendigungs-Abrechnungssatzes: 54 Byte

(A) Satzbeschreibung: Satzkennung: „ESMD“

(B) Kennzeichnungsteil: Subsystem-Kennzeichnung

(C) Grundinformation

Feld-Nr.	Distanz		Länge (Byte)	Format	Bedeutung
	hex	dez			
1	00	0	1	B	Zustandsanzeige ¹
2	01	1	1	A	Kennzeichen Jahreszeit (aktuell) ²
3	02	2	1	-	- reserviert -

¹ Die Zustandsanzeige kann zwei mögliche Werte annehmen:

- 1 Subsystem wird in den Wartezustand versetzt (Kommando HOLD-SUBSYSTEM)
- 0 Subsystem ist beendet (Kommando STOP-SUBSYSTEM)

² „S“ für Sommerzeit; „W“ für Winterzeit

Länge der Grundinformation: 3 Byte

(D) Variable Information

Die variable Information des Subsystem-Beendigungs-Abrechnungssatzes enthält **keine** Satzerweiterung.

3.7 Fehlerbehandlung in DSSM

DSSM-Task-Error

1. DSSM beendet sich abnormal während des ersten Schrittes von Startup, z.B. wenn Subsysteme aktiviert werden, die mit dem Attribut *BEFORE-DSSM-LOAD oder *AT-DSSM-LOAD definiert wurden. Es wird ein Fehlercode zur Startup-Task geschickt. Normalerweise wird der Startup abgebrochen; u.U. kann er jedoch ohne DSSM-Initialisierung fortgeführt werden (abhängig von der Startup-Realisierung).

Zu den Startup-Schritten siehe [Seite 211](#) oder Handbuch „Einführung in die Systembetreuung“ [14].

2. DSSM beendet sich abnormal während des zweiten Schrittes von Startup, z.B. wenn Subsysteme aktiviert werden, die mit dem Attribut *MANDATORY-AT-STARTUP oder *BEFORE-SYSTEM-READY definiert wurden oder während die Datenstrukturen eines Subsystems mit dem Attribut *AFTER-SYSTEM-READY aktualisiert werden. Hat DSSM versucht, eines dieser Subsysteme zu aktivieren, wird das Subsystem in den Zustand LOCKED überführt.

Wenn ein Subsystem mit *MANDATORY-AT-STARTUP in den Zustand LOCKED überführt wurde, wird ein Fehlercode zur Startup-Task geschickt. Ob der Startup abgebrochen wird, ist abhängig von der Startup-Realisierung.

3. DSSM beendet sich abnormal während Shutdown, wenn das zu deaktivierende Subsystem im Zustand LOCKED ist.
Die Behandlung anderer Subsysteme wird von Shutdown normal fortgesetzt.
4. andere Ursachen, die zur abnormalen Beendigung von DSSM führen:
DSSM analysiert die Situation, bereinigt seine internen Tabellen und trifft folgende Entscheidungen abhängig davon, wo der Fehler auftrat:

Der Fehler tritt auf während ...	Reaktion
CREATE/DELETE/RESUME/HOLD	Subsystem LOCKED (mit Fehlermeldung)
Austausch von Subsystemversionen – mit Unterbrechung der Verfügbarkeit – ohne Unterbrechung der Verfügbarkeit	die fehlerhafte Routine wird nochmals aufgerufen; führt dies wiederum zum Fehler, – wird die betreffende Version – werden beide Subsystemversionen in den Zustand LOCKED gesetzt
ADD-/MODIFY-SUBSYSTEM-PARAMETER, REMOVE-SUBSYSTEM	die Anweisung wird abgebrochen, wobei Teile des Katalogs bereits geändert sein können
Wiederherstellung der Holdertask	die Wiederherstellung wird abgebrochen
SHOW-SUBSYSTEM-INFO, SAVE-CATALOG	Anforderung wird abnormal beendet

Nach dem Bereinigen der Tabellen setzt DSSM seine Arbeit mit den Anforderungen fort, die in der DSSM-Börse warten.

Holdertask-Error

Treten in der Holdertask Probleme auf oder wird sie abnormal beendet, analysiert DSSM die Situation, initiiert automatisch den Wiederanlauf der Holdertask und trifft folgende Entscheidungen:

Das Problem tritt auf ...	Reaktion auf den Holdertask-Error bei	
	RESTART-REQUIRED=*YES	RESTART-REQUIRED=*NO
- während normaler Anforderungen an das Subsystem		
Aktivierung	Subsystem LOCKED	Subsystem LOCKED
INIT-Routine	Aufruf der INIT-Routine	Subsystem LOCKED
CLOSE-CTRL-Routine	Subsystem CREATED	Subsystem LOCKED
STOPCOM-Routine	Aufruf der INIT-Routine	Subsystem LOCKED
DEINIT-Routine	Fortsetzung der Deinitialisierung	Subsystem LOCKED
Deaktivierung	Subsystem LOCKED	Subsystem LOCKED
Subsystem-Session (Arbeitstask)	Aufruf der INIT-Routine	Subsystem LOCKED
- beim Austausch von Subsystemversionen mit Unterbrechung der Verfügbarkeit		
Aktivierung V2	Subsystem V2 LOCKED und Abbruch des Austauschs	Subsystem V2 LOCKED und Abbruch des Austauschs
STOPCOM-Routine V1	Subsystem V1 LOCKED und Aufruf der INIT-Routine für V2	Subsystem V1 LOCKED und Aufruf der INIT-Routine für V2
INIT-Routine V2	Aufruf INIT-Routine V2 und Fortsetzung der DEINIT-Routine von V1	Subsystem V2 LOCKED und Fortsetzung der DEINIT-Routine von V1
Deaktivierung V1	Subsystem V1 LOCKED	Subsystem V1 LOCKED
DEINIT-Routine V1	Subsystem V1 LOCKED	Subsystem V1 LOCKED
- beim Austausch von Subsystemversionen ohne Unterbrechung der Verfügbarkeit		
Aktivierung V2	Subsystem V2 LOCKED und Abbruch des Austauschs	Subsystem V2 LOCKED und Abbruch des Austauschs
CLOSE-CTRL-Routine V1	Abbruch des Austauschs (Subsystem V1 CREATED)	Subsystem V1 LOCKED und Entladen von V1
INIT-Routine V2	Aufruf INIT-Routine V2 und Fortsetzung der DEINIT-Routine von V1	Subsystem V2 LOCKED und Abbruch des Austauschs
STOPCOM-Routine V1	Subsystem V1 LOCKED	Subsystem V1 LOCKED
DEINIT-Routine V1	Subsystem V1 LOCKED	Subsystem V1 LOCKED
Deaktivierung V1	Subsystem V1 LOCKED	Subsystem V1 LOCKED

Fehlerdatei SERSLOG

DSSM schreibt einen Eintrag in die Error-Logging-Datei SERSLOG, wenn:

- ein Systemaufruf fehlerhaft ist
- inkonsistente Daten im internen Subsystemkatalog erkannt werden
- die DSSM-Task abnormal beendet wird (in diesem Fall enthält der SERSLOG-Eintrag eine Mitteilung, die die aktuelle Situation beschreibt)

Der Eintrag kann aus bis zu drei Teilen bestehen:

1. der Parameterliste des fehlerhaften Systemaufrufs oder den inkonsistenten Daten (in Einheiten zu 2 KByte)
2. dem Returncode (wenn er nicht schon Bestandteil der Parameterliste ist)
3. der Adresse der fehlerhaften Routine und die Adresse der Routine, die die fehlerhafte Routine aufgerufen hat. Dieser Eintrag ist für die Diagnose nützlich, denn DSSM benutzt einen eigenen zentralen SERSLOG-Aufruf.

3.8 Die Kommandos von DSSM

Kommando	Bedeutung	Seite
ADD-SUBSYSTEM	Dynamischen Subsystemkatalog erweitern	75
HOLD-SUBSYSTEM	Subsystem in den Wartezustand versetzen	80
LOAD-LOCAL-SUBSYSTEM-CATALOG	Lokalen Subsystemkatalog laden	83
MODIFY-SUBSYSTEM-PARAMETER	Subsystem-Parameter verändern	85
RELEASE-SUBSYSTEM-SPACE	Reservierten Adressraum von Subsystemen freigeben	110
REMOVE-SUBSYSTEM	Inaktives Subsystem aus dynamischem Katalog entfernen	111
RESUME-SUBSYSTEM	Wartezustand für ein Subsystem aufheben	113
SAVE-SUBSYSTEM-CATALOG	Änderungen des dynamischen Subsystemkatalogs sichern	116
SET-DSSM-OPTIONS	Protokollierungsfunktion von DSSM aus- oder einschalten	120
SHOW-SUBSYSTEM-ATTRIBUTES	Informationen über Subsystem-Attribute anzeigen	122
SHOW-SUBSYSTEM-INFO	Informationen über aktuelle Subsystem-Konfiguration anfordern	138
SHOW-SUBSYSTEM-STATUS	Informationen über Zustand von Subsystemen anfordern	142
START-LOCAL-SUBSYSTEM	Lokales Subsystem im Benutzeradressraum aktivieren	150
START-SUBSYSTEM	Subsystem aktivieren	153
STOP-LOCAL-SUBSYSTEM	Lokales Subsystem im Benutzeradressraum deaktivieren	159
STOP-SUBSYSTEM	Subsystem deaktivieren	162
UNLOAD-LOCAL-SUBSYSTEM-CATALOG	Lokalen Subsystemkatalog entladen	166
UNLOCK-SUBSYSTEM	Subsystem im Zustand LOCKED in den Zustand NOT-CREATED überführen	168

Tabelle 10: Kommandos von DSSM

Die **SDF-Syntaxdarstellung** der Kommandos ist in [Abschnitt „SDF-Syntaxdarstellung“](#) auf [Seite 5](#) erläutert.

ADD-SUBSYSTEM

Dynamischen Subsystemkatalog erweitern

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Die Systembetreuung kann mit diesem Kommando im laufenden System die aktuelle Subsystemkonfiguration erweitern. Es kann entweder ein neuer Katalog angegeben werden, der auch alle Einträge seines Vorgängers enthält, oder ein Katalog, der lediglich neue Subsysteme aufweist, die dem aktuellen Katalog hinzugefügt werden sollen.

Der anzugebende Subsystemkatalog muss in jedem Fall mit SSCM erstellt worden sein.

Der bei der Systemeinleitung verwendete („alte“) statische Subsystemkatalog wird nicht automatisch aktualisiert. Für den **nächsten** Systemlauf kann die Systembetreuung

- den mit ADD-SUBSYSTEM erstellten Katalog während der Systemeinleitung verwenden
- einen komplett neu erstellten und aktualisierten Subsystemkatalog generieren und für die Systemeinleitung verwenden. Dieser neue Katalog muss weder quantitativ auf den evtl. veralteten Vorgängerkatalog aufbauen noch qualitativ dessen Bezüge und Merkmale unterstützen.

Format

ADD-SUBSYSTEM

CATALOG = <filename 1..54 without-gen-vers>

,TYPE = *~~EXTENDED-ACTIVE-CONFIGURATION~~ / *NEW-SUBSYSTEMS

Operandenbeschreibung

CATALOG = <filename 1..54 without-gen-vers>

Name des neuen Subsystemkataloges.

TYPE =

Legt fest, ob der aktuelle Katalog erweitert oder ersetzt werden soll.

TYPE = *EXTENDED-ACTIVE-CONFIGURATION

Es soll ein komplett neuer Katalog aktiviert werden, der nicht nur die Einträge seines Vorgängers, sondern auch die neuen Subsysteme enthält.

TYPE = *NEW-SUBSYSTEMS

Der angegebene Katalog enthält lediglich neue Subsysteme, die zum alten Katalog hinzugefügt werden sollen.

Der Katalog wird von DSSM daraufhin geprüft, ob die darin enthaltenen Subsysteme für die Konfiguration wirklich neu sind. Wird ein auch im zu erweiternden Katalog geführtes Subsystem gefunden, wird das Kommando abgewiesen.

Einschränkungen:

- Die bei Generierung mit SSCM mit RELATED-SUBSYSTEM und REFERENCED-SUBSYSTEM angegebenen Subsysteme müssen zyklensfrei sein, d.h. frei von gegenseitiger Abhängigkeit.
- Unzulässig ist die Definition verschiedener Versionen eines Subsystems mit den Start-Attributen AT-SUBSYSTEM-CALL, BEFORE-SYSTEM-READY, AFTER-SYSTEM-READY, BEFORE-DSSM-LOAD, AT-DSSM-LOAD und MANDATORY-AT-STARTUP (Ausnahme: AT-SUBSYSTEM-CALL ist dann zulässig, wenn für alle beteiligten Versionen Koexistenz vereinbart wird).

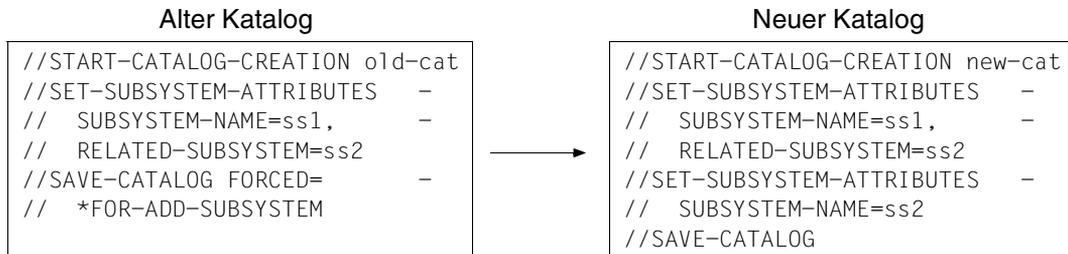
Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	32	ESM0296	Abnormale Beendigung (REQM-Fehler)
	32	ESM0350	Internes DSSM-Problem während der Verarbeitung
	64	ESM0260	Datei nicht gefunden
	64	ESM0261	Speicherplatzgrenze im DSSM-Katalog erreicht
	64	ESM0262	Datei ist kein Katalog
	64	ESM0322	Maximale Anzahl Beziehungen überschritten
	64	ESM0325	Inkonsistenzen im Katalog
	64	ESM0332	Katalog-Version nicht kompatibel
	64	ESM0340	Nicht genügend reservierter Adressraum vorhanden für Subsysteme mit MEMORY-CLASS=*BY-SLICE

Hinweise

Es kann Beziehungen zwischen Subsystemen geben, die nicht im aktuellen Katalog definiert sind (siehe auch SSCM-Anweisung SAVE-CATALOG mit Operand FORCED=*FOR-ADD-SUBSYSTEM auf [Seite 239](#)). Dieses Problem kann dadurch umgangen werden, dass diese Subsysteme in einem neuen Katalog definiert und zum alten Katalog mit /ADD-SUBSYSTEM hinzugefügt werden.

Beispiel für TYPE=*EXTENDED-ACTIVE-CONFIGURATION:



Einschränkungen für den Operanden TYPE=*EXTENDED-ACTIVE-CONFIGURATION

- Subsysteme mit dem Attribut MEMORY-CLASS=*LOCAL-PRIVILEGED, die im neu erstellten Katalog hinzukommen, dürfen weder die Größe des Adressraum-Streifens im System- oder Benutzeradressraum überschreiten, noch darf sich ihre Lage im Adressraum mit der des alten Kataloges überschneiden.
- Der Operand CREATION-TIME für neue Subsysteme muss für bereits im alten Katalog definierte Versionen dieses Subsystems kompatibel gewählt werden. Die Werte BEFORE-SYSTEM-READY, AFTER-SYSTEM-READY, BEFORE-DSSM-LOAD, AT-DSSM-LOAD und MANDATORY-AT-STARTUP sind dabei zwar zulässig, aber stets ohne Wirkung, da der Startzeitpunkt im laufenden System bereits überschritten ist; d.h. die Systembetreuung erhält eine entsprechende Warnung, das Subsystem wird aber nicht geladen.
- Bei der Aufteilung von Subsystemen auf Holdertasks (Anweisung ASSIGN-HOLDER-TASK) muss das „Stand-alone-Prinzip“ gewahrt bleiben, d.h. Subsysteme von verschiedenen Katalogen dürfen nicht einer Holdertask zugeteilt werden.

Beispiel

Subsysteme im alten Katalog: A, B, C

Subsysteme im neuen Katalog: A, B, C, D, E

Es gilt:

//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(A,B)) ist zulässig

//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(D,E)) ist zulässig

aber:

//ASSIGN-HOLDER-TASK *SHARE-HOLDER(BY-SUB=(A,D,C)) ist nicht zulässig

- Der neue Katalog muss größer sein als der Vorgänger, da er nicht nur die alten Subsysteme mit deren Attributen enthält (Beziehungen, Abhängigkeiten und Ladevorgaben), sondern auch die neuen Subsysteme führen muss.
- Binde- und Abhängigkeitsbeziehungen (REFERENCED-/RELATED-SUBSYSTEM) dürfen die Kataloggrenze nicht überschreiten: es dürfen im Katalog A keine Beziehungen eines dort definierten Subsystems zu einem Subsystem hinterlegt sein, das im Katalog B definiert ist.
- Sobald mit REMOVE-SUBSYSTEM ein Subsystem aus dem Katalog gelöscht wurde, kann TYPE=*EXTENDED-ACTIVE-CONFIGURATION nicht mehr angegeben werden.

Nicht zulässig ist die Definition verschiedener

- Subsysteme mit der identischen Attribut-Kombination: SVC-NUMBER / FUNCTION-NUMBER / FUNCTION-VERSION.
- Subsysteme mit der identischen Attribut-Kombination: FUNCTION-NUMBER / FUNCTION-VERSION (falls die Werte für VERSION-COEEXISTENCE bzw. VERSION-EXCHANGE auf *ALLOWED gesetzt sind) bei Subsystemen mit indirekter Verbindung über System Procedure Linkage (ISL).
- Versionen eines Subsystems mit der identischen Attribut-Kombination: SVC-NUMBER / FUNCTION-NUMBER / FUNCTION-VERSION, falls die Werte für VERSION-COEEXISTENCE bzw. VERSION-EXCHANGE auf *ALLOWED gesetzt sind.
- Versionen eines Subsystems mit der identischen Attribut-Kombination: FUNCTION-NUMBER / FUNCTION-VERSION / VERSION-COEEXISTENCE bzw. VERSION-EXCHANGE bei Subsystemen mit indirekter Verbindung über System Procedure Linkage (ISL).

Eine Überlappung von Subsystemen muss vermieden werden. DSSM vergleicht dahingehend die Werte für die Operanden SIZE und START-ADDRESS in der Anweisung SET-SUBSYSTEM-ATTRIBUTES.

Die Zuweisung einer Holdertask (durch SSCM-Anweisung) für ein altes und ein neues Subsystem hat keine Auswirkung auf die Aufteilung der Holdertasks.

Tabelle von Unverträglichkeiten bei Beziehungen zwischen Subsystemen im alten und neuen Katalog:

- x: diese Kombination ist nicht möglich:
sowohl Bindebeziehungen (REFERENCED-SUBSYSTEM) als auch andere Abhängigkeitsbeziehungen (RELATED-SUBSYSTEM) sind nicht erlaubt
- r: Bindebeziehungen (REFERENCED-SUBSYSTEM) sind nicht erlaubt

Subsystem im neuen Katalog	Subsystem im alten Katalog												
	MANDATORY-AT-STARTUP	BEFORE-SYSTEM-READY	AFTER-SYSTEM-READY	AT-CREATION-REQUEST	AT-SUBSYSTEM-CALL	BEFORE-DSSM-LOAD	AT-DSSM-LOAD	STOP-AT-SHUTDOWN=*YES	SUBSYSTEM-ACCESS=*LOW	SUBSYSTEM-ACCESS=*HIGH	MEMORY-CLASS=*LOCAL-PRIVILEGED	MEMORY-CLASS=*LOCAL-UNPRIVILEGED	MEMORY-CLASS=*BY-SLICE
MANDATORY-AT-STARTUP		x	x	x	x								
BEFORE-SYSTEM-READY			x	x	x								
AFTER-SYSTEM-READY				x	x								
BEFORE-DSSM-LOAD	x	x	x	x	x		x						
AT-DSSM-LOAD	x	x	x	x	x								
STOP-AT-SHUTDOWN=*NO								r					
SUBSYSTEM-ACCESS=*SYSTEM									x	x			x
MEMORY-CLASS=*SYSTEM-GLOBAL											x	x	
MEMORY-CLASS=*BY-SLICE											r	r	r

Tabelle 11: Unverträglichkeiten bei Beziehungen zwischen Subsystemen im alten und neuen Katalog

HOLD-SUBSYSTEM

Subsystem in den Wartezustand versetzen

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	OPERATING SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Mit dem Kommando HOLD-SUBSYSTEM kann ein beliebiges Subsystem in den Wartezustand versetzt werden.

Zu dem angegebenen Subsystem wird keine neue Verbindung mehr zugelassen; die notwendigen Betriebsmittel (Holdertask, Adressraum) bleiben verfügbar. Über die Option FORCED ist es zudem möglich, die Beendigung aller belegenden Tasks abzuwarten oder deren unverzüglichen Abbruch zu bewirken. Nachdem die Deinitialisierungsphase durchlaufen ist, befindet sich das Subsystem im Wartezustand; dieser kann mit dem Kommando RESUME-SUBSYSTEM wieder aufgehoben werden.

Das Kommando HOLD-SUBSYSTEM wird abgewiesen, wenn ein Subsystem mit SUBSYSTEM-HOLD=*FORBIDDEN definiert wurde.



Um ein hohes Maß an Parallelität und Datenintegrität zu gewährleisten, werden zeitaufwändige Verwaltungsaufgaben nicht unter der Kontrolle der aufrufenden Task ausgeübt, sondern einer DSSM-Task übertragen.

In der Regel wird nur die Prüfung der geforderten Funktion **synchron** (d.h. verbunden mit einem Wartezustand für die aufrufende Task) realisiert. Die eigentliche Verarbeitung jedoch führt DSSM **asynchron** und unabhängig von der aufrufenden Task durch.

Format

HOLD-SUBSYSTEM

SUBSYSTEM-NAME = <structured-name 1..8>
,VERSION = ***STD** / <product-version mandatory-man-corr> / <product-version without-man-corr> / ***HIGHEST**
,SUBSYSTEM-PARAMETER = ***NONE** / <c-string 1..254>
,FORCED = ***NO** / ***YES**
,SYNCHRONOUS = ***NO** / ***YES**

Operandenbeschreibung

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

Name des Subsystems, das in den Wartezustand versetzt wird.

VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturzustand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = *STD

Existiert nur **eine** Subsystemversion, die geladen ist, wird diese Version ausgewählt. Existieren **mehrere** passende Versionen, muss die Version spezifiziert werden.

VERSION = *HIGHEST

Es wird die höchste Version des Subsystems ausgewählt.

SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>

Vereinbart, ob spezielle Parameter verarbeitet werden, die nur das angegebene Subsystem auswerten kann.

FORCED =

Bestimmt Verhalten und Dringlichkeit der Kommandobearbeitung.

FORCED = *NO

Die Verarbeitung wird abgewartet und damit die normale Beendigung aller Tasks, die auf dieses Subsystem zugreifen.

FORCED = *YES

Der unverzügliche Abbruch aller belegenden Prozesse wird veranlasst. Dies kann im Falle eines privilegierten Subsystems zu einem Systemdump führen; das Programm von Tasks, die an ein nicht-privilegiertes Subsystem angeschlossen sind, wird abgebrochen und einer Fehlerbehandlung der Ereignisklasse ABEND unterzogen.

SYNCHRONOUS =

Erlaubt die Wahl zwischen synchroner und asynchroner Verarbeitung.

SYNCHRONOUS = *NO

Das Kommando soll asynchron verarbeitet werden, d.h ohne mit einer weiteren Eingabe auf dessen Ausführung warten zu müssen. Nach der Syntaxprüfung des Kommandos erhält die aufrufende Task die Meldung ESM0216. Fehlermeldungen über den Ablauf des Kommandos werden nicht ausgegeben.

SYNCHRONOUS = *YES

Die Ausführung des Kommandos muss abgewartet werden. Begleitende Fehlermeldungen werden ausgegeben. Im Fall eines Versionsaustauschs ist diese Angabe nur für die neu zu aktivierende Version relevant. Die Deaktivierung der anderen, „alten“ Version geschieht immer asynchron.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	CMD0001	Keine Aktion notwendig; Subsystem ist bereits im Wartezustand
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0224	Kommando wird nicht verarbeitet
	32	ESM0228	Kommando abnormal beendet

LOAD-LOCAL-SUBSYSTEM-CATALOG

Lokalen Subsystemkatalog laden

Anwendungsbereich: SYSTEM-MANAGEMENT

Privilegierung: STD-PROCESSING

Funktionsbeschreibung

Mit dem Kommando LOAD-LOCAL-SUBSYSTEM-CATALOG kann der Aufrufer einen lokalen Subsystemkatalog in seinen taskeigenen Benutzeradressraum (Klasse-5-Speicher) laden.

Ein lokaler Subsystemkatalog darf keine privilegierten Subsysteme enthalten.

Es kann immer nur ein lokaler Subsystemkatalog geladen sein. Der Versuch, einen weiteren lokalen Subsystemkatalog zu laden - ohne den ersten vorher mit dem Kommando UNLOAD-LOCAL-SUBSYSTEM-CATALOG zu entladen-, wird mit einer Fehlermeldung abgewiesen.

Das Kommando LOAD-LOCAL-SUBSYSTEM-CATALOG wird nur ausgeführt, wenn der angegebene Katalog als lokaler Katalog definiert wurde, also keine privilegierten Subsysteme enthält und mit SSCM V2.0 oder höher generiert wurde.

Ein lokaler Subsystemkatalog kann nicht geladen werden, wenn bereits ein Programm geladen wurde.

Während ein lokaler Subsystemkatalog geladen ist, wird das Kommando RESTART-PROGRAM abgewiesen. Bevor das Programm erneut gestartet werden kann, muss der lokale Subsystemkatalog explizit entladen werden.

Format

LOAD-LOCAL-SUBSYSTEM-CATALOG
CATALOG-NAME = <filename 1..54 without-gen-vers>

Operandenbeschreibung

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

Name des lokalen Subsystemkatalogs, der in den taskeigenen Benutzeradressraum (Klasse-5-Speicher) geladen werden soll.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando erfolgreich ausgeführt
	64	ESM0255	Kommando nicht ausgeführt; es ist bereits ein lokaler Katalog oder ein Programm geladen
	64	ESM0326	Kommandobearbeitung abgebrochen; (Systemfehler, nicht-kompatible Katalogversionen, DVS-Fehler, Katalog enthält privilegierte Subsysteme oder die angegebene Datei ist kein Katalog)

Hinweise

- Die Größe des Speicherplatzes, die der lokale Subsystemkatalog im Benutzeradressraum einnimmt, ist nur von der Größe der Subsystemdefinitionen im Katalog abhängig. Es besteht daher keine Notwendigkeit für eine nachträgliche Speicherplatzvergrößerung.
- Enthält der lokale Subsystemkatalog zu viele oder zu große Subsysteme, kann es zur Sättigung des Benutzeradressraums kommen. Diesem Problem wird vorgebeugt, wenn der lokale Subsystemkatalog oberhalb der 16 MByte-Grenze geladen wird.
- Abhängigkeitsbeziehungen und andere Verbindungen zwischen Subsystemen werden in der lokalen Subsystemverwaltung nicht berücksichtigt.
- Wird in einem Programm das Makro WRCPT aufgerufen, während ein lokaler Subsystemkatalog geladen ist, sichert er die lokale Subsystem-Konfiguration in die Checkpoint-Datei. Wird das Programm mit dem Kommando RESTART-PROGRAM fortgesetzt, kann die lokale Subsystem-Konfiguration von dort wieder eingelesen werden.

Beispiel

```

/load-local-subsystem-catalog catalog-name=local.dssmcat.1 _____ (1)
%ESM0254 COMMAND '/LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/load-local-subsystem-catalog catalog-name=local.dssmcat.2 _____ (2)
%ESM0344 A LOCAL CATALOG IS ALREADY LOADED
%ESM0255 COMMAND '/LOAD-LOCAL-SUBSYSTEM-CATALOG' NOT PROCESSED

```

- (1) Zuerst wird der lokale Subsystemkatalog LOCAL.DSSMCAT.1 geladen. Die Ausführung des Kommandos ist erfolgreich.
- (2) Das Laden eines weiteren lokalen Subsystemkatalogs wird mit der Meldung ESM0344 abgewiesen: Es kann immer nur eine Datei als lokaler Subsystemkatalog geladen werden.

MODIFY-SUBSYSTEM-PARAMETER

Subsystem-Parameter verändern

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Der mit dem Privileg SUBSYSTEM-MANAGEMENT ausgestattete privilegierte Benutzer kann im laufenden System mit diesem Kommando die Parameter eines Subsystems verändern; nur explizit angegebene Parameter werden geändert.

Das Kommando ändert lediglich den dynamischen, nicht jedoch den statischen Subsystemkatalog. Vorgenommene Änderungen sind demnach nur während der laufenden Sitzung wirksam, nicht jedoch beim nächsten Startup.

Damit die Änderungen dauerhaft erhalten bleiben, können sie mithilfe des Kommandos SAVE-SUBSYSTEM-CATALOG in einem statischen Katalog abgespeichert werden. Dabei ist zu beachten, dass manche Änderungen beim nächsten Startup möglicherweise sinnlos oder sogar nachteilig sein können (z.B. wenn eine Meldungsdatei einem Subsystem mit dem Start-Attribut BEFORE-DSSM-LOAD¹ zugeordnet wird).



Gute Kenntnisse des zu verändernden Subsystems sind Voraussetzung für die Anwendung des Kommandos, da sich mit /MODIFY-SUBSYSTEM-PARAMETER tief greifende Veränderungen der Subsystem-Attribute vornehmen lassen.

Das Kommando verfügt über drei unterschiedliche Operandenarten:

- Operanden, deren Wert im dynamischen Subsystemkatalog gespeichert sind und die sofort wirksam werden (z.B. VERSION-COEXISTENCE).
- Operanden, deren Wert im dynamischen Subsystemkatalog gespeichert sind, die aber erst nach dem nächsten /START-SUBSYSTEM wirksam werden (z.B. SUBSYSTEM-LIBRARY).
- Operanden, die nur akzeptiert werden, wenn das Subsystem nicht gestartet ist (z.B. MESSAGE-FILE).

¹ Das Attribut BEFORE-DSSM-LOAD kann nur über SSCM verändert werden, siehe [Kapitel „SSCM“ auf Seite 179](#).

Format

(Teil 1 von 2)

MODIFY-SUBSYSTEM-PARAMETER

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

,**VERSION** = <product-version mandatory-man-corr> / <product-version without-man-corr>

,**INSTALLATION-UNIT** = *UNCHANGED / *NONE / *STD / <text 1..30>

,**INSTALLATION-USERID** = *UNCHANGED / *NONE / <name 1..8> / *DEFAULT-USERID

,**COPYRIGHT** = *UNCHANGED / *NONE / <c-string 1..54>(…)

 <c-string 1..54>(…)

 | **YEAR** = *YEAR-1990 / <c-string 4..4>

,**SUBSYSTEM-LIBRARY** = *UNCHANGED / *STD / *INSTALLED(…) / <filename 1..54 without-gen-vers>

 *INSTALLED(…)

 | **LOGICAL-ID** = *UNCHANGED / *REFRESH / <filename 1..30 without-catid-userid-gen-vers>

 | **DEFAULT-NAME** = *UNCHANGED / <filename 1..54>

,**SUBSYSTEM-LOAD-MODE** = *UNCHANGED / *STD / *ADVANCED

,**REP-FILE** = *UNCHANGED / *STD / *NO / *INSTALLED(…) / <filename 1..54 without-gen-vers>

 *INSTALLED(…)

 | **LOGICAL-ID** = *UNCHANGED / *REFRESH / <filename 1..30 without-catid-userid-gen-vers>

 | **DEFAULT-NAME** = *UNCHANGED / *NONE / <filename 1..54>

,**REP-FILE-MANDATORY** = *UNCHANGED / *NO / *YES

,**MESSAGE-FILE** = *UNCHANGED / *NO / *INSTALLED(…) / <filename 1..54 without-gen-vers>

 *INSTALLED(…)

 | **LOGICAL-ID** = *UNCHANGED / *REFRESH / <filename 1..30 without-catid-userid-gen-vers>

 | **DEFAULT-NAME** = *UNCHANGED / *NONE / <filename 1..54>

,**SUBSYSTEM-INFO-FILE** = *UNCHANGED / *NO / *INSTALLED(…) / <filename 1..54 without-gen-vers>

 *INSTALLED(…)

 | **LOGICAL-ID** = *UNCHANGED / *REFRESH / <filename 1..30 without-catid-userid-gen-vers>

 | **DEFAULT-NAME** = *UNCHANGED / *NONE / <filename 1..54>

,**SYNTAX-FILE** = *UNCHANGED / *NO / *INSTALLED(…) / <filename 1..54 without-gen-vers>

 *INSTALLED(…)

 | **LOGICAL-ID** = *UNCHANGED / *REFRESH / <filename 1..30 without-catid-userid-gen-vers>

 | **DEFAULT-NAME** = *UNCHANGED / *NONE / <filename 1..54>

,**DYNAMIC-CHECK-ENTRY** = *UNCHANGED / *STD / *NO / <text 1..8 without-sep>

Fortsetzung ➔

```

,CREATION-TIME = *UNCHANGED / *AT-CREATION-REQUEST / *AT-SUBSYSTEM-CALL(...)/
                *AT-DSSM-LOAD / *MANDATORY-AT-STARTUP / *BEFORE-SYSTEM-READY /
                *AFTER-SYSTEM-READY
                *AT-SUBSYSTEM-CALL(...)
                |   ON-ACTION = *STD / *ISL-CALL / *ALL
,INIT-ROUTINE = *UNCHANGED / *NO / <text 1..8 without-sep>
,CLOSE-CTRL-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,STOPCOM-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,DEINIT-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,STOP-AT-SHUTDOWN = *UNCHANGED / *NO / *YES
,INTERFACE-VERSION = *UNCHANGED / *NO / <text 1..8 without-sep>
,SUBSYSTEM-HOLD = *UNCHANGED / *ALLOWED / *FORBIDDEN
,STATE-CHANGE-CMDS = *UNCHANGED / *ALLOWED / *FORBIDDEN / *BY-ADMINISTRATOR-ONLY
,FORCED-STATE-CHANGE = *UNCHANGED / *ALLOWED / *FORBIDDEN
,RESET = *UNCHANGED / *ALLOWED / *FORBIDDEN
,RESTART-REQUIRED = *UNCHANGED / *NO / *YES
,VERSION-COEXISTENCE = *UNCHANGED / *FORBIDDEN / *ALLOWED
,VERSION-EXCHANGE = *UNCHANGED / *FORBIDDEN / *ALLOWED
,MEMORY-CLASS = *UNCHANGED / *SYSTEM-GLOBAL(...)/ *LOCAL-UNPRIVILEGED(...)/ *BY-SLICE(...)
                *SYSTEM-GLOBAL(...)
                |   SUBSYSTEM-ACCESS = *LOW / *HIGH
                *LOCAL-UNPRIVILEGED(...)
                |   SIZE = *UNCHANGED / <integer 1..32767 4Kbyte>
                |   ,SUBSYSTEM-ACCESS = *UNCHANGED / *LOW / *HIGH
                |   ,START-ADDRESS = *UNCHANGED / *ANY / <x-string 7..8>
                *BY-SLICE(...)
                |   SIZE = <integer 1..32767 4Kbyte>
,LINK-ENTRY = *UNCHANGED (...) / <text 1..8 without-sep>(...)
                *UNCHANGED(...)
                |   AUTOLINK = *UNCHANGED / *ALLOWED / *FORBIDDEN
                <text 1..8 without-sep>(...)
                |   AUTOLINK = *ALLOWED / *FORBIDDEN
,UNRESOLVED-EXTERNALS = *UNCHANGED / *ALLOWED / *FORBIDDEN
,CHECK-REFERENCES = *UNCHANGED / *YES / *NO
,CHANGE-STATE = *UNCHANGED / *YES / *NO

```

Operandenbeschreibung

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

Legt den Namen des Subsystems fest, dessen Parameter geändert werden soll(en).

VERSION = <product-version mandatory-man-corr> / <product-version without-man-corr>

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

INSTALLATION-UNIT =

Legt den Namen der installierten Liefergruppe fest (eine Liefergruppe wird auch Release Unit genannt). Für alle mit IMON installierten Subsysteme muss ein Wert ungleich *NONE angegeben werden, wenn bei einem der Operanden SUBSYSTEM-LIBRARY, REP-FILE, SUBSYSTEM-INFO-FILE, MESSAGE-FILE und SYNTAX-FILE der Wert *INSTALLED(LOGICAL-ID=...) definiert wurde.

INSTALLATION-UNIT = *UNCHANGED

Der Name der installierten Liefergruppe bleibt unverändert.

INSTALLATION-UNIT = *NONE

Es wird kein Name vergeben. Für alle mit IMON installierten Subsysteme ist diese Angabe nicht erlaubt.

INSTALLATION-UNIT = *STD

Der beim Operanden SUBSYSTEM-NAME angegebene Name wird als Name der installierten Liefergruppe genutzt.

INSTALLATION-UNIT = <text 1..30>

Neuer Name der installierten Liefergruppe.

INSTALLATION-USERID = *UNCHANGED / *NONE / <name 1..8> / *DEFAULT-USERID

Verändert die Standard-Benutzerkennung der mit dem Subsystem verknüpften Dateien (Operanden REP-FILE, SUBSYSTEM-LIBRARY, SYNTAX-FILE, MESSAGE-FILE, SUBSYSTEM-INFO-FILE). Ohne Benutzerkennung angegebene Dateinamen werden auf der hier festgelegten neuen Installations-Benutzerkennung gesucht.

Die Änderung der Installations-Benutzerkennung wird abgewiesen, wenn das betroffene Subsystem aktiv ist und ihm eine Meldungsdatei (Operand MESSAGE-FILE) oder Syntaxdatei (Operand SYNTAX-FILE) ohne angegebene Benutzerkennung zugeordnet ist.

Die Änderung wird sofort wirksam.

INSTALLATION-USERID = *UNCHANGED

Die Installations-Benutzerkennung wird nicht verändert.

INSTALLATION-USERID = *NONE

Entfernt die Installations-Benutzerkennung.

War vor Absetzen des Kommandos eine Installations-Benutzerkennung festgelegt, so wird diese allen Dateien entzogen, denen sie zugeordnet war.

INSTALLATION-USERID = <name 1..8>

Die hier angegebene Benutzerkennung wird Installations-Benutzerkennung. Der Name der Benutzerkennung muss ohne „\$“ angegeben werden.

INSTALLATION-USERID = *DEFAULT-USERID

Legt die Standard-Benutzerkennung des Systems als Installations-Benutzerkennung fest (d.h. Dateien beginnen mit „\$.“).

COPYRIGHT = *UNCHANGED / *NONE / <c-string 1..54>(…)

Ändert die beim Laden des Subsystems ausgegebene Copyright-Meldung.

Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

COPYRIGHT = *UNCHANGED

Die Copyright-Angabe wird nicht verändert.

COPYRIGHT = *NONE

Es wird keine Copyright-Meldung mehr ausgegeben.

COPYRIGHT = <c-string 1..54>(…)

Verändert die Copyright-Meldung, die beim Laden des Subsystems ausgegeben wird.

YEAR = *YEAR-1990 / <c-string 4..4>

Legt die in der Copyright-Meldung enthaltene Jahresangabe für das erste Produktionsjahr fest. Standardmäßig wird das Jahr 1990 in die Copyright-Meldung eingesetzt. Eine davon abweichende Jahresangabe muss explizit angegeben werden. Dabei ist zu beachten, dass für den Operandenwert keine semantische Gültigkeitsprüfung erfolgt.

SUBSYSTEM-LIBRARY = *UNCHANGED / *STD / *INSTALLED(…) / <filename 1..54 without-gen-vers>

Ändert für die angegebene Subsystemversion die Zuweisung einer Modulbibliothek. Aus der Modulbibliothek wird der Code für ein noch nicht gestartetes Subsystem geladen. Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

SUBSYSTEM-LIBRARY = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SUBSYSTEM-LIBRARY = *STD

Für die Bibliotheksangabe wird folgender Standardname vergeben:

SYSLNK.<subsysname>.<subsysvers#>

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

SUBSYSTEM-LIBRARY = *INSTALLED(...)

Der Bibliotheksname wird durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt.

LOGICAL-ID =

Legt den logischen Namen der Programm- oder Bindemodulbibliothek fest, unter dem die Bibliothek bei IMON bekannt ist.

LOGICAL-ID = *UNCHANGED

Der logische Name der Programm- oder Bindemodulbibliothek bleibt unverändert.

LOGICAL-ID = *REFRESH

Der zum logischen Namen gehörende Pfadname wurde geändert und soll nun im Katalog aktualisiert werden. Der logische Name selber ist unverändert.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Neuer logischer Name der Programm- oder Bindemodulbibliothek.

DEFAULT-NAME =

Name der Bibliothek bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

DEFAULT-NAME = *UNCHANGED

Der Bibliotheksname bleibt unverändert.

DEFAULT-NAME = <filename 1..54>

Neuer Bibliotheksname.

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

Der hier angegebene vollqualifizierte Dateiname wird als neuer Bibliotheksname festgelegt (siehe auch „[Hinweise](#)“ auf Seite 109).

SUBSYSTEM-LOAD-MODE = *UNCHANGED / *STD / *ADVANCED

Legt fest, auf welche Art das Subsystem gestartet wird.

SUBSYSTEM-LOAD-MODE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten. Die Art, das Subsystem zu starten, wird nicht verändert.

SUBSYSTEM-LOAD-MODE = *STD

BLS wird im STD-Run-Modus (über die BLS-DSSM-Schnittstelle \$PBBND1) aufgerufen, den Subsystem-Code als Objekt-Modul zu laden.

SUBSYSTEM-LOAD-MODE = *ADVANCED

BLS wird im ADVANCED-Run-Modus (über die BLS-DSSM-Schnittstelle \$PBBND1) aufgerufen, den Subsystem-Code als Link-and-Load-Modul zu laden.

REP-FILE = *UNCHANGED / *STD / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>

Ändert für die angegebene Subsystemversion die Zuweisung der Rep-Korrekturdatei. Rep-Korrekturdateien sind dafür vorgesehen, Modulkorrekturen in ein nicht gestartetes Subsystem einzubringen. Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

REP-FILE = *UNCHANGED

Die Parameter für die Rep-Korrekturdatei werden nicht geändert.

REP-FILE = *STD

Für die Rep-Korrekturdatei wird folgender Standardname vergeben:

SYSREP.<subsysname>.<subsysvers>

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

REP-FILE = *NO

Für das Subsystem existiert keine Rep-Korrekturdatei.

REP-FILE = *INSTALLED(...)

Der Name der Rep-Datei wird durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt.

LOGICAL-ID =

Legt den logischen Namen der Rep-Datei fest, unter dem die Datei bei IMON bekannt ist.

LOGICAL-ID = *UNCHANGED

Der logische Name der Rep-Datei bleibt unverändert.

LOGICAL-ID = *REFRESH

Der zum logischen Namen gehörende Pfadname wurde geändert und soll nun im Katalog aktualisiert werden. Der logische Name selber ist unverändert.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Neuer logischer Name der Rep-Datei.

DEFAULT-NAME =

Name der Rep-Datei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

DEFAULT-NAME = *UNCHANGED

Der Name der Rep-Datei bleibt unverändert.

DEFAULT-NAME = *NONE

Es wird kein Standardname für die Rep-Datei vergeben.

DEFAULT-NAME = <filename 1..54>

Neuer Name der Rep-Datei.

REP-FILE = <filename 1..54 without-gen-vers>

Der hier angegebene vollqualifizierte Dateiname wird als neuer Rep-Korrekturdateiname festgelegt (siehe „[Hinweise](#)“ auf Seite 109).

REP-FILE-MANDATORY = *UNCHANGED / *NO / *YES

Legt fest, ob das Subsystem gestartet wird, wenn bei der Bearbeitung der Rep-Korrekturdatei Fehler auftreten.

Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

REP-FILE-MANDATORY = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

REP-FILE-MANDATORY = *NO

Fehler beim Bearbeiten der Rep-Korrekturdatei sind ohne Einfluss auf das Laden des Subsystems.

REP-FILE-MANDATORY = *YES

Die dynamische Subsystemverwaltung (DSSM) weist in den folgenden Fällen das Laden des Subsystems zurück:

- DVS-Fehler beim Verarbeiten der Rep-Korrekturdatei (z.B. Rep-Korrekturdatei nicht katalogisiert)
- Fehler beim Überprüfen der Rep-Korrekturdatei
- Name der Rep-Korrekturdatei fehlerhaft
- DVS-Fehler beim Verarbeiten der NOREF-Datei

**MESSAGE-FILE = *UNCHANGED / *NO / *INSTALLED(...)
<filename 1..54 without-gen-vers>**

Ändert die für die angegebene Subsystemversion gültige Meldungsdatei-Festlegung. Die Subsystemversion darf nicht gestartet sein.

Vom DVS an den Dateinamen gestellte Anforderungen werden nicht überprüft.

MESSAGE-FILE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

MESSAGE-FILE = *NO

Es ist keine subsystemspezifische Meldungsdatei verfügbar.

Diese Einstellung ist erforderlich, wenn für das Subsystem als Aktivierungszeitpunkt BEFORE-DSSM-LOAD definiert ist, siehe [Kapitel „SSCM“ auf Seite 179](#).

MESSAGE-FILE = *INSTALLED(...)

Der Name der Meldungsdatei wird durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt.

LOGICAL-ID =

Legt den logischen Namen der Meldungsdatei fest, unter dem die Datei bei IMON bekannt ist.

LOGICAL-ID = *UNCHANGED

Der logische Name der Meldungsdatei bleibt unverändert.

LOGICAL-ID = *REFRESH

Der zum logischen Namen gehörende Pfadname wurde geändert und soll nun im Katalog aktualisiert werden. Der logische Name selber ist unverändert.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Neuer logischer Name der Meldungsdatei.

DEFAULT-NAME =

Name der Meldungsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

DEFAULT-NAME = *UNCHANGED

Der Name der Meldungsdatei bleibt unverändert.

DEFAULT-NAME = *NONE

Es wird kein Standardname für die Meldungsdatei vergeben.

DEFAULT-NAME = <filename 1..54>

Neuer Name der Meldungsdatei.

MESSAGE-FILE = <filename 1..54 without-gen-vers>

Der hier angegebene vollqualifizierte Dateiname wird als neuer Meldungsdateiname festgelegt (siehe „Hinweise“ auf Seite 109).

SUBSYSTEM-INFO-FILE = *UNCHANGED / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>

Legt fest, welche Informationsdatei für die angegebene Subsystemversion verwendet wird.

SUBSYSTEM-INFO-FILE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SUBSYSTEM-INFO-FILE = *NO

Es ist keine Informationsdatei verfügbar.

SUBSYSTEM-INFO-FILE = *INSTALLED(...)

Der Informationsdateiname wird durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt.

LOGICAL-ID =

Legt den logischen Namen der Informationsdatei fest, unter dem die Datei bei IMON bekannt ist.

LOGICAL-ID = *UNCHANGED

Der logische Name der Informationsdatei bleibt unverändert.

LOGICAL-ID = *REFRESH

Der zum logischen Namen gehörende Pfadname wurde geändert und soll nun im Katalog aktualisiert werden. Der logische Name selber ist unverändert.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Neuer logischer Name der Informationsdatei.

DEFAULT-NAME =

Name der Informationsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

DEFAULT-NAME = *UNCHANGED

Der Name der Informationsdatei bleibt unverändert.

DEFAULT-NAME = *NONE

Es wird kein Standardname für die Informationsdatei vergeben.

DEFAULT-NAME = <filename 1..54>

Neuer Name der Informationsdatei.

SUBSYSTEM-INFO-FILE = <filename 1..54 without-gen-vers>

Der hier angegebene vollqualifizierte Dateiname wird als neuer Informationsdateiname festgelegt (siehe „Hinweise“ auf Seite 109).

SYNTAX-FILE = *UNCHANGED / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>

Verändert die für die angegebene Subsystemversion gültige Syntaxdatei-Festlegung. Die Syntaxdatei enthält die für die Subsystemversion gültigen Kommando- und Operandenwerte.

Bei Ausführung des Kommandos darf die Subsystemversion nicht geladen sein. Vom DVS an den Dateinamen gestellte Anforderungen werden nicht überprüft.

SYNTAX-FILE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SYNTAX-FILE = *NO

Es ist keine Syntaxdatei verfügbar.

Diese Einstellung ist erforderlich, wenn für das Subsystem als Aktivierungszeitpunkt BEFORE-DSSM-LOAD oder AT-DSSM-LOAD definiert ist.

SYNTAX-FILE = *INSTALLED(...)

Der Name der Syntaxdatei wird durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt.

LOGICAL-ID =

Legt den logischen Namen der Syntaxdatei fest, unter dem die Datei bei IMON bekannt ist.

LOGICAL-ID = *UNCHANGED

Der logische Name der Syntaxdatei bleibt unverändert.

LOGICAL-ID = *REFRESH

Der zum logischen Namen gehörende Pfadname wurde geändert und soll nun im Katalog aktualisiert werden. Der logische Name selber ist unverändert.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Neuer logischer Name der Syntaxdatei.

DEFAULT-NAME =

Name der Syntaxdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

DEFAULT-NAME = *UNCHANGED

Der Name der Syntaxdatei bleibt unverändert.

DEFAULT-NAME = *NONE

Es wird kein Standardname für die Syntaxdatei vergeben.

DEFAULT-NAME = <filename 1..54>

Neuer Name der Syntaxdatei.

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

Der hier angegebene vollqualifizierte Dateiname wird als neuer Syntaxdateiname festgelegt (siehe „Hinweise“ auf Seite 109).

DYNAMIC-CHECK-ENTRY = *UNCHANGED / *STD / *NO / <text 1..8 without-sep>

Ändert die Bezugsadresse, die dazu dient, die Richtigkeit der zum Subsystem gehörenden geladenen Verschlüsselung festzustellen.

DYNAMIC-CHECK-ENTRY = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

DYNAMIC-CHECK-ENTRY = *STD

Die im Operanden LINK-ENTRY angegebene Bezugsadresse wird als Prüf-Bezugsadresse verwendet.

DYNAMIC-CHECK-ENTRY = *NO

Es wird keine Überprüfung durchgeführt. Diese Einstellung ist unzulässig, wenn für das Subsystem als Aktivierungszeitpunkt BEFORE-DSSM-LOAD definiert ist.

DYNAMIC-CHECK-ENTRY = <text 1..8 without-sep>

Die hier angegebene Adresse wird für die dynamische Prüfung als neue Bezugsadresse festgelegt.

CREATION-TIME = *UNCHANGED / *AT-CREATION-REQUEST / *AT-SUBSYSTEM-CALL(...) / *AT-DSSM-LOAD / *MANDATORY-AT-STARTUP / *BEFORE-SYSTEM-READY / *AFTER-SYSTEM-READY

Ändert den Startzeitpunkt eines Subsystems.

CREATION-TIME = *UNCHANGED

Der bisher gültige Wert wird beibehalten.

CREATION-TIME = *AT-CREATION-REQUEST

Der Startzeitpunkt für das Subsystem wird auf den (Generierungs-) Standardwert „Start nach Aufruf des Kommandos START-SUBSYSTEM“ zurückgesetzt.

CREATION-TIME = *AT-SUBSYSTEM-CALL(...)

Der Startzeitpunkt für das Subsystem wird auf den Wert AT-SUBSYSTEM-CALL geändert. Das Subsystem wird damit beim ersten SVC- oder ISL-Aufruf automatisch gestartet. Dieser Operandenwert ist über SVC oder ISL aufgerufenen Subsystemen vorbehalten.

Der Operandenwert kann für die angegebene Subsystemversion abgesetzt werden, nachdem dieses Attribut, falls erforderlich, einer anderen Subsystemversion entzogen worden ist. Besitzt keine andere Version dieses Attribut, so wird es direkt zu der angegebenen Subsystemversion transferiert. Besitzt eine andere Subsystemversion dieses Attribut, so wird dem Subsystem das Attribut entweder sofort (Subsystem ist noch nicht gestartet) oder nach dem Entladen (bei aktivem Subsystem) entzogen.

Eine Ausnahme stellen Subsysteme mit dem Attribut VERSION-COEXISTENCE=*ALLOWED dar. Hier können unterschiedliche Versionen desselben Subsystems das Attribut CREATION-TIME=*AT-SUBSYSTEM-CALL besitzen.

Ebenso wie für SSCM kann dieses Attribut nur für ein Subsystem mit CALL-Eingang (Entry) MODE=*SVC / *ISL gegeben werden.

ON-ACTION =

Bestimmt, wodurch das automatische Laden des Subsystems veranlasst wird.

ON-ACTION = *STD

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC-Einsprungstelle.

ON-ACTION = *ISL-CALL

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden ISL-Einsprungstelle.

ON-ACTION = *ANY

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC- oder ISL-Einsprungstelle.

CREATION-TIME = *AT-DSSM-LOAD

Das Subsystem soll während der Systemeinleitung unter der Kontrolle der DSSM-Task geladen werden. Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die ebenfalls mit diesem Start-Attribut definiert sind oder das Start-Attribut BEFORE-DSSM-LOAD besitzen.

Der Dateiname für dieses Subsystem muss unter der Benutzerkennung TSOS auf dem Home-Pubset angelegt sein, da zum Startzeitpunkt weder der Benutzerkatalog zugreifbar, noch die IMPORT-PUBSET-Verarbeitung abgeschlossen ist.

Für diese Subsysteme ist die Einbindung einer Syntaxdatei nicht zulässig.

CREATION-TIME = *MANDATORY-AT-STARTUP

Das Subsystem muss während der Systemeinleitung (Phase 2: nach dem Laden von DSSM; siehe [Seite 212](#)) geladen werden. Die Aktivierung wird - wie bei BEFORE-SYSTEM-READY - synchron angestoßen. Im Unterschied zum oben Genannten muss das Laden des Subsystems allerdings **erfolgreich abgeschlossen** werden. Andernfalls geht eine Meldung an die Startup-Routine, dass ein verpflichtendes Subsystem nicht geladen werden konnte. Die Startup-Routine entscheidet in diesem Fall, ob die Verarbeitung fortgesetzt oder abgebrochen wird.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Start-Attributen BEFORE-DSSM-LOAD oder AT-DSSM-LOAD definiert wurden. Der Dateiname für dieses Subsystem muss auf dem Home-Pubset katalogisiert sein.

CREATION-TIME = *BEFORE-SYSTEM-READY

Das Subsystem soll während der Systemeinleitung (Phase 2; siehe [Seite 212](#)) geladen werden. Die Aktivierung wird synchron angestoßen; die Steuerung geht erst nach dem vollständigen Laden (oder nach Lade-Fehler) an die Startup-Routine zurück, die dann „SYSTEM READY“ melden kann.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Start-Attributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD oder MANDATORY-AT-STARTUP definiert wurden.

Der Dateiname für dieses Subsystem muss auf dem Home-Pubset katalogisiert sein.

CREATION-TIME = *AFTER-SYSTEM-READY

Das Laden des Subsystems wird während der Systemeinleitung (Phase 2; siehe [Seite 212](#)) angestoßen. Die Durchführung dieses Auftrags wird nicht mit der Startup-Routine synchronisiert, die vor dem Abschluss des Ladens „SYSTEM READY“ melden kann.

Das Subsystem darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Start-Attributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD, MANDATORY-AT-STARTUP oder BEFORE-SYSTEM-READY definiert wurden.

Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

INIT-ROUTINE = *UNCHANGED / *NO / <text 1..8 without-sep>

Ändert die Initialisierungs-Routine des Subsystems (siehe dazu auch [Seite 214](#)), sofern dadurch nicht der Arbeitsprozess des Subsystems verändert wird.

Die Änderung wird sofort wirksam, um eine evtl. notwendige Rekonstruktion des Subsystems zu ermöglichen.

INIT-ROUTINE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

INIT-ROUTINE = *NO

Es wird keine Initialisierung durchgeführt.

INIT-ROUTINE = <text 1..8 without-sep>

Der hier angegebene Name wird als neuer Bezugsadressen-Name für das Subsystem festgelegt (siehe „[Hinweise](#)“ auf [Seite 109](#)).

CLOSE-CTRL-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>

Ändert die CLOSE-CTRL-Routine des Subsystems (siehe dazu auch [Seite 215](#) oder [Seite 257](#)), sofern dadurch nicht der Arbeitsprozess des Subsystems verändert wird. Die Änderung wird sofort wirksam.

CLOSE-CTRL-ROUTINE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

CLOSE-CTRL-ROUTINE = *NO

DSSM bearbeitet die Kommandos STOP-SUBSYSTEM bzw. HOLD-SUBSYSTEM, ohne dabei eine CLOSE-CTRL-Routine aufzurufen.

CLOSE-CTRL-ROUTINE = *DYNAMIC

Die CLOSE-CTRL-Routine wird vom Subsystem dynamisch am Ende der Initialisierungs-(INIT-)Routine aufgerufen. Für die Routine muss ein Bezugsadressen-Name festgelegt sein.

Beim Aufruf der CLOSE-CTRL-Routine muss die Holdertask des Subsystems Arbeitstask sein (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

CLOSE-CTRL-ROUTINE = <text 1..8 without-sep>

Der hier angegebene Name wird als neuer Bezugsadressen-Name für die CLOSE-CTRL-Routine festgelegt (siehe „Hinweise“ auf Seite 109).

STOPCOM-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>

Ändert die STOPCOM-Routine des Subsystems (siehe dazu auch Seite 215 oder Seite 258), sofern dadurch nicht der Arbeitsprozess des Subsystems verändert wird. Die Änderung wird sofort wirksam.

STOPCOM-ROUTINE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

STOPCOM-ROUTINE = *NO

DSSM bearbeitet die Kommandos STOP-SUBSYSTEM bzw. HOLD-SUBSYSTEM, ohne dabei eine STOPCOM-Routine aufzurufen.

STOPCOM-ROUTINE = *DYNAMIC

Die STOPCOM-Routine wird vom Subsystem dynamisch am Ende der CLOSE-CTRL-Routine oder sofern keine solche vereinbart ist, am Ende der Initialisierungs- (INIT-)Routine aufgerufen.

Beim Aufruf der STOPCOM-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, Seite 193).

STOPCOM-ROUTINE = <text 1..8 without-sep>

Der hier angegebene Name wird als neue Bezugsadressen-Name für die STOPCOM-Routine festgelegt (siehe „Hinweise“ auf Seite 109).

DEINIT-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>

Verändert die DEINIT-Routine des Subsystems (siehe dazu auch Seite 216 oder Seite 259), sofern dadurch nicht der Arbeitsprozess des Subsystems verändert wird. Die Änderung wird sofort wirksam.

DEINIT-ROUTINE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

DEINIT-ROUTINE = *NO

DSSM bearbeitet die Kommandos STOP-SUBSYSTEM bzw. HOLD-SUBSYSTEM, ohne dabei eine DEINIT-Routine aufzurufen.

DEINIT-ROUTINE = *DYNAMIC

Die DEINIT-Routine wird vom Subsystem dynamisch am Ende der STOPCOM-Routine oder sofern keine solche vereinbart ist, am Ende der CLOSE-CTRL-Routine aufgerufen. Ist auch keine solche vereinbart, so wird die DEINIT-Routine am Ende der Initialisierungs- (INIT-)Routine aufgerufen.

Beim Aufruf der DEINIT-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, Seite 193).

DEINIT-ROUTINE = <text 1..8 without-sep>

Der hier angegebene Name wird als neuer Bezugsadressen-Name für die DEINIT-Routine festgelegt (siehe „Hinweise“ auf Seite 109).

STOP-AT-SHUTDOWN = *UNCHANGED / *NO / *YES

Veranlasst DSSM, das Subsystem automatisch zu beenden, sobald das Kommando SHUTDOWN (Systemlauf beenden) abgesetzt wird. Die Änderung wird sofort wirksam.

STOP-AT-SHUTDOWN = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

STOP-AT-SHUTDOWN = *NO

Bei Absetzen des Kommandos SHUTDOWN wird das Subsystem von DSSM ignoriert.

STOP-AT-SHUTDOWN = *YES

DSSM beendet das Subsystem, sobald das Kommando SHUTDOWN abgesetzt wird (analog /STOP-SUBSYSTEM).

INTERFACE-VERSION = *UNCHANGED / *NO / <text 1..8 without-sep>

Bezeichnet die Einsprungstelle (Entry), über die DSSM auf diejenige Schnittstellenversion zugreifen kann, die für freilaufende Aufrufe (decoupled calling) der Routinen INIT, CLOSE-CTRL, STOPCOM und DEINIT verwendet wird. Die Änderung wird sofort wirksam.

INTERFACE-VERSION = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

INTERFACE-VERSION = *NO

Keiner der folgenden Eingangs- (Entry-) Namen ist verfügbar:
INIT-ROUTINE, DEINIT-ROUTINE, STOPCOM-ROUTINE, CLOSE-CTRL-ROUTINE.

INTERFACE-VERSION = <text 1..8 without-sep>

Der hier angegebene Name wird als neue Einsprungstelle (Entry) festgelegt.

SUBSYSTEM-HOLD = *UNCHANGED / *ALLOWED / *FORBIDDEN

Legt fest, ob ein Subsystem über Kommando oder Makro angehalten oder entladen werden kann. Die Änderung wird sofort wirksam.

SUBSYSTEM-HOLD = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SUBSYSTEM-HOLD = *ALLOWED

Das Subsystem kann über Kommando oder Makro angehalten oder entladen werden.

SUBSYSTEM-HOLD = *FORBIDDEN

Analog dem Operandenwert STOP-AT-SHUTDOWN kann das Subsystem erst dann entladen werden, wenn das BS2000-System über das Kommando SHUTDOWN beendet wird.

STATE-CHANGE-CMDS = *UNCHANGED / *ALLOWED / *FORBIDDEN / *BY-ADMINISTRATOR-ONLY

Legt fest, ob für das Subsystem die DSSM-Kommandos START-SUBSYSTEM, RESUME-SUBSYSTEM, STOP-SUBSYSTEM und HOLD-SUBSYSTEM abgesetzt werden dürfen.

Die Änderung wird sofort wirksam.

Im Fall eines Versionsaustauschs ist diese Angabe nur für die neu zu aktivierende Version relevant. Die Deaktivierung der anderen, „alten“ Version geschieht in jedem Fall.

STATE-CHANGE-CMDS = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

STATE-CHANGE-CMDS = *ALLOWED

Die genannten Kommandos können von der Bedienstation bzw. unter einer mit dem Privileg SUBSYSTEM-MANAGEMENT ausgestatteten Benutzerkennung abgesetzt werden.

STATE-CHANGE-CMDS = *FORBIDDEN

Die genannten Kommandos können weder von der Bedienstation noch unter einer mit dem Privileg SUBSYSTEM-MANAGEMENT ausgestatteten Benutzerkennung abgesetzt werden.

STATE-CHANGE-CMDS = *BY-ADMINISTRATOR-ONLY

Die genannten Kommandos können lediglich unter einer mit dem Privileg SUBSYSTEM-MANAGEMENT ausgestatteten Benutzerkennung abgesetzt werden, nicht aber von der Bedienstation.

FORCED-STATE-CHANGE = *UNCHANGED / *ALLOWED / *FORBIDDEN

Legt fest, ob für das Subsystem die DSSM-Kommandos STOP-SUBSYSTEM und HOLD-SUBSYSTEM mit dem Operanden FORCED abgesetzt werden dürfen.

Die Änderung wird sofort wirksam.

FORCED-STATE-CHANGE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

FORCED-STATE-CHANGE = *ALLOWED

Für die genannten Kommandos darf der Operand FORCED verwendet werden.

FORCED-STATE-CHANGE = *FORBIDDEN

Die Verwendung des Operanden FORCED ist untersagt.

RESET = *UNCHANGED / *ALLOWED / *FORBIDDEN

Legt fest, ob für das Subsystem die DSSM-Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM mit dem Operanden RESET=*YES abgesetzt werden dürfen.

Die Änderung wird sofort wirksam.

RESET = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

RESET = *ALLOWED

Die DSSM-Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM werden akzeptiert, wenn sie mit dem Operanden RESET=*YES abgesetzt werden.

RESET = *FORBIDDEN

Die DSSM-Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM werden abgelehnt, wenn sie mit dem Operanden RESET=*YES abgesetzt werden.

RESTART-REQUIRED = *UNCHANGED / *NO / *YES

Legt fest, ob für das Subsystem bei abnormal beendeter Holdertask die Initialisierungs-(INIT-)Routine aufgerufen wird, um einen Subsystem-Restart durchzuführen.

Die Initialisierungs-Routine wird während der Rekonstruktion der Holdertask aufgerufen. Die Änderung wird sofort wirksam.

RESTART-REQUIRED = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

RESTART-REQUIRED = *NO

Im Falle einer abnormal beendeten Holdertask wird das Subsystem während der Rekonstruktion der Holdertask gesperrt.

RESTART-REQUIRED = *YES

Der Restart für das Subsystem wird durch Aufrufen der Initialisierungs-Routine durchgeführt.

VERSION-COEXISTENCE = *UNCHANGED / *FORBIDDEN / *ALLOWED

Legt fest, ob verschiedene Versionen des Subsystems gleichzeitig aktiv sein können. Die Änderung wird sofort wirksam.

VERSION-COEXISTENCE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

VERSION-COEXISTENCE = *FORBIDDEN

Es kann nur eine Version des Subsystems aktiv sein.

VERSION-COEXISTENCE = *ALLOWED

Verschiedene Versionen des Subsystems können gleichzeitig aktiv sein.

VERSION-EXCHANGE = *UNCHANGED / *FORBIDDEN / *ALLOWED

Legt fest, ob eine neue Subsystemversion aktiviert werden kann, ohne dass die alte Version gelöscht sein muss. Die Änderung wird sofort wirksam.

VERSION-EXCHANGE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

VERSION-EXCHANGE = *FORBIDDEN

Eine neue Subsystemversion kann nur aktiviert werden, wenn die alte Subsystemversion vollständig gelöscht ist.

VERSION-EXCHANGE = *ALLOWED

Eine Subsystemversion kann aktiviert werden, ohne dass die andere Subsystemversion gelöscht sein muss.

MEMORY-CLASS = *UNCHANGED / *SYSTEM-GLOBAL(...) / *LOCAL-UNPRIVILEGED(...) / *BY-SLICE(...)

Ändert die Speicherklasse des Subsystems bzw. legt dessen Hauptspeicherstelle fest (über oder unter 16 MByte). Dabei ist Folgendes zu beachten:

- Ein privilegiertes Subsystem kann nicht in ein nicht-privilegiertes Subsystem geändert werden.
- Wird die Speicherklasse geändert, so müssen alle Unteroperanden angegeben werden.
- Ein Subsystem kann nicht LOCAL-UNPRIVILEGED werden, wenn eine Adressenüberlappung zwischen zwei LOCAL-UNPRIVILEGED-Subsystemen besteht, die sich dieselbe Holdertask teilen.

Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

MEMORY-CLASS = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

MEMORY-CLASS = *SYSTEM-GLOBAL(...)

Die Speicherklasse des Subsystems wird auf Klasse-3- bzw. Klasse-4-Speicher geändert.

SUBSYSTEM-ACCESS = *LOW / *HIGH

Legt die Zugriffsrechte für den angeforderten Speicherplatz sowie die Stelle im Adressraum fest, wo sich der angeforderte Speicherplatz befindet.

SUBSYSTEM-ACCESS = *LOW

Nicht-privilegierter Adressraum wird angefordert. Die Hauptspeicherstelle liegt unterhalb 16 MByte.

SUBSYSTEM-ACCESS = *HIGH

Es wird nicht-privilegierter Adressraum bis zu 2 GByte angefordert.

MEMORY-CLASS = *LOCAL-UNPRIVILEGED(...)

Der Speicher-Pool wird als Klasse-6-Speicher angelegt (nur für Subsysteme, die auf dieselbe Art wie Programme ausgeführt werden sollen).

SIZE = *UNCHANGED / <integer 1..32767 4Kbyte>

Legt die Größe des für den Memory-Pool benötigten Adressraums in 4K-Seiten fest. Der festgelegte Wert muss so groß sein, dass das Subsystem und sämtliche dynamisch vom Subsystem zu ladenden Units und Lade-Units vollständig untergebracht werden können.

SIZE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SIZE = <integer 1..32767 4Kbyte>

Der hier angegebene Adressraum wird für den Memory-Pool festgelegt.

SUBSYSTEM-ACCESS = *UNCHANGED / *LOW / *HIGH

Legt die Zugriffsrechte für den angeforderten Speicherplatz sowie die Stelle im Adressraum fest, wo sich der angeforderte Speicherplatz befindet.

SUBSYSTEM-ACCESS = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

SUBSYSTEM-ACCESS = *SYSTEM

Privilegierter Adressraum wird angefordert. Die Ladeadresse liegt oberhalb 16 MByte.

SUBSYSTEM-ACCESS = *LOW

Nicht-privilegierter Adressraum wird angefordert. Die Ladeadresse liegt unterhalb 16 MByte.

SUBSYSTEM-ACCESS = *HIGH

Nicht-privilegierter Adressraum bis zu 2 GByte wird angefordert.

START-ADDRESS = *UNCHANGED / *ANY / <x-string 7..8>

Legt die Ladeadresse des Subsystems fest. Sie muss ein Vielfaches von X'100000' sein. Es liegt im Verantwortungsbereich des Benutzers, eine Adresse anzugeben, die auf einen Klasse-6-Speicher zeigt.

START-ADDRESS = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

START-ADDRESS = *ANY

Die Lage des Subsystems im Klasse-6-Speicher wird von DSSM festgelegt.

START-ADDRESS = <x-string 7..8>

Startadresse im Segmentraster, an der die Anfangsadresse des Subsystems liegen soll. Die Adresse muss ein Vielfaches von X'100000' sein.

MEMORY-CLASS = *BY-SLICE(...)

Das angegebene Subsystem ist ein nicht-privilegiertes Subsystem und besteht aus einem LLM, das aus einem mehrbenutzbaren Code (Programmbereich) und einem nicht-mehrbenutzbaren Code (Datenbereich) besteht.

Der Programmbereich wird in den gemeinsam benutzbaren Adressraum geladen (das ent-

spricht MEMORY-CLASS=*SYSTEM-GLOBAL). Der Datenbereich wird in den Benutzer-adressraum der Holdertask geladen und in die privaten Benutzeradressräume der angeschlossenen Tasks an dieselbe Adresse kopiert.

Wenn das Subsystem mit *BY-SLICE definiert wird, ist Folgendes zu beachten:

- Existiert bereits ein reservierter Adressbereich zur Aufnahme des Datenbereichs, wird das Kommando nur dann ausgeführt, wenn innerhalb dieses reservierten Adressbereichs genügend freier Platz zur Aufnahme des geänderten Subsystems zur Verfügung steht.
- Existiert kein reservierter Adressbereich zur Aufnahme des Datenbereichs, wird ein solcher vom System eingerichtet. Die Tasks, die zum Zeitpunkt der Einrichtung eines solchen reservierten Adressraums an das Subsystem angeschlossen sind, können den Datenbereich nicht nutzen.

Bei der ersten Verbindung einer Task zu einem Subsystem, das mit *BY-SLICE definiert wurde, informiert DSSM das Subsystem BLSSERV darüber, dass mit dem Makro VSVI1 auf die Kopie des Datenbereichs im privaten Benutzeradressraum zugegriffen werden kann.

Das Makro VSVI1 informiert den Anwender über Einträge in den Tabellen des DBL. Einzelheiten zum Makro siehe Handbuch „BLSSERV“ [4].

Beim Aufheben der letzten Verbindung informiert DSSM das Subsystem BLSSERV darüber, dass nicht mehr auf diesen privaten Anteil zugegriffen werden kann.

DSSM akzeptiert eine Änderung des Adressraums in das neue Attribut *BY-SLICE nur, wenn für den Typ des vereinbarten Auftragseingangs für das Subsystem MODE=*LINK angegeben wurde und für alle Subsystemeinträge CONNECTION-SCOPE=*TASK / *PROGRAM angegeben wurde.

SIZE = <integer 1..32767 4Kbyte>

Gibt die Größe des angeforderten Speicherplatzes für den Datenbereich in 4K-Seiten an.

LINK-ENTRY = *UNCHANGED(...) / <text 1..8 without-sep>(…)

Verändert die für das Laden des Subsystems verwendete Bezugsadresse. Darüber hinaus kann festgelegt werden, ob das automatische Einfügen von Modulen in Phasen (AUTOLINK) zugelassen wird. Die Änderung wird wirksam, sobald das Subsystem neu gestartet wird (Kommando START-SUBSYSTEM).

LINK-ENTRY = *UNCHANGED(...)

Die bisher gültige Einstellung wird beibehalten.

AUTOLINK = *UNCHANGED / *ALLOWED / *FORBIDDEN

Legt fest, ob das automatische Einfügen von Modulen in Phasen (AUTOLINK) zugelassen wird.

AUTOLINK = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

AUTOLINK = *ALLOWED

AUTOLINK wird zugelassen.

AUTOLINK = *FORBIDDEN

AUTOLINK wird nicht zugelassen.

LINK-ENTRY = <text 1..8 without-sep>(...)

Die hier angegebene Adresse wird die neue Bezugsadresse für das Laden des Subsystems.

AUTOLINK = *ALLOWED / *FORBIDDEN

Legt fest, ob das automatische Einfügen von Modulen in Phasen (AUTOLINK) zugelassen wird.

AUTOLINK = *ALLOWED

AUTOLINK wird zugelassen.

AUTOLINK = *FORBIDDEN

AUTOLINK wird nicht zugelassen.

UNRESOLVED-EXTERNALS = *UNCHANGED / *ALLOWED / *FORBIDDEN

Legt fest, ob unbefriedigte externe Programmadressen das Starten des Subsystems verhindern.

UNRESOLVED-EXTERNALS = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

UNRESOLVED-EXTERNALS = *ALLOWED

Unbefriedigte externe Programmadressen verhindern nicht den Start des Subsystems. Diese Einstellung ist nur für Testzwecke vorgesehen.

UNRESOLVED-EXTERNALS = *FORBIDDEN

Unbefriedigte externe Programmadressen verhindern den Start des Subsystems.

CHECK-REFERENCES = *UNCHANGED / *YES / *NO

Legt fest, ob DSSM den Status von Subsystemen überprüfen soll, zu denen eine Abhängigkeitsbeziehung unterhalten wird. Der Status dieser Subsysteme kann dafür ausschlaggebend sein, ob das Subsystem ge- oder entladen werden darf.

Die Änderung wird sofort wirksam.

CHECK-REFERENCES = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

CHECK-REFERENCES = *YES

DSSM überprüft den Status der Subsysteme, zu denen eine Abhängigkeitsbeziehung unterhalten wird. Je nach Status dieser Subsysteme entscheidet DSSM, ob das in diesem Kommando angesprochene Subsystem ge- oder entladen werden kann.

CHECK-REFERENCES = *NO

Steht das in diesem Kommando angesprochene Subsystem in einer Abhängigkeitsbeziehung zu einem anderen Subsystem, so wird überprüft, ob das andere Subsystem bereits geladen ist. Ist dies der Fall, so kann das eigene Subsystem geladen werden, auch dann, wenn das andere Subsystem noch nicht ablauffähig ist (der Verweis gilt als aufgelöst).

Die Kommandos RESUME-SUBSYSTEM, STOP-SUBSYSTEM und HOLD-SUBSYSTEM werden ausgeführt, ohne bestehende Abhängigkeitsbeziehungen zu beachten.

CHANGE-STATE = *UNCHANGED / *YES / *NO

Beschränkt die Steuerung des Subsystems oder hebt eine bestehende Beschränkung wieder auf.

Besondere Bedeutung kommt diesem Operanden bei einem fehlerhaftem Subsystem zu, da sich damit das Laden, Aktivieren und Deaktivieren des Subsystems während der Zeit verhindern lässt, die für erforderliche Korrekturen benötigt wird. Die Korrekturen können damit risikolos durchgeführt werden.

Die Änderung wird sofort wirksam. Sie behält nur während der aktuellen Sitzung Gültigkeit (die Änderung wird nicht in den über Kommando SAVE-SUBSYSTEM-CATALOG angesprochenen Katalog gesichert). Beim nächsten Subsystemstart erhält der Operand den Wert *NO.

CHANGE-STATE = *UNCHANGED

Die bisher gültige Einstellung wird beibehalten.

CHANGE-STATE = *YES

Die vorher mit dem Kommando MODIFY-SUBSYSTEM-PARAMETER CHANGE-STATE=*NO veranlasste Sperre der dort aufgeführten Kommandos wird wieder aufgehoben. Die uneingeschränkte Steuerung des Subsystems ist damit wieder erlaubt.

CHANGE-STATE = *NO

Verhindert das Laden, Aktivieren, Deaktivieren, Anhalten, Wiederanlaufen und Entsperren des Subsystems. Folgende Kommandos werden bis zur nächsten Eingabe des Kommandos MODIFY-SUBSYSTEM-PARAMETER CHANGE-STATE=*YES gesperrt:

- START-SUBSYSTEM
- RESUME-SUBSYSTEM
- STOP-SUBSYSTEM
- HOLD-SUBSYSTEM
- REMOVE-SUBSYSTEM
- UNLOCK-SUBSYSTEM

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando ohne Fehler ausgeführt
	0	CMD0001	garantierte Meldung: ESM0254
	0	CMD0001	Kommando mit Warnungen ausgeführt
	0	CMD0001	garantierte Meldung: ESM0647
1	0	CMD0001	Kommando mit Warnungen ausgeführt
			(keine LOGICAL-ID gefunden oder Meldungsdatei (de-)aktiviert)
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	1	ESM0415	Syntaxfehler: es wurde ein ungültiger INSTALLATION-UNIT-Name angegeben
	1	ESM0653	Syntaxfehler: es wurde ein ungültiger Auftragseingangs-Name angegeben
	32	ESM0646	Internes DSSM-Problem während der Verarbeitung
	32	ESM0646	interner Fehler
	64	ESM0201	Subsystem nicht gefunden
	64	ESM0269	Subsystem ohne SVC- oder ISL-Eingang
	64	ESM0280	Kommando nicht ausgeführt um Inkonsistenzen im Subsystemkatalog zu vermeiden
	64	ESM0340	Nicht genügend reservierter Adressraum vorhanden für Subsysteme mit MEMORY-CLASS=*BY-SLICE
	64	ESM0613	Änderung der Meldungs- bzw. Syntaxdatei, oder der Installations-Userid nicht erlaubt. Das Subsystem muss vorher mit STOP-SUBSYSTEM gelöscht werden.
	64	ESM0617	MEMORY-CLASS geändert; es müssen alle Parameter angegeben werden

Hinweise

- Wenn für das Subsystem Attribute eingegeben werden, die es auch vorher schon hatte, wird die Meldung `ESM0647` ausgegeben. Diese Meldung weist auf erfolgte Änderungen hin. Die Ausgabe dieser Änderungs-Meldung lässt sich vermeiden durch die Verwendung der Standardwerte anstelle bereits bestehender Werte (in der Regel Standardwert `*UNCHANGED`).
- Der Operand `CREATION-TIME` ist dafür vorgesehen, die Parameter von Subsystemen zu ändern, die über das Kommando `ADD-SUBSYSTEM` hinzugefügt wurden.
- Werden Dateinamen bei den Operanden `LIBRARY`, `MESSAGE-FILE`, `SYNTAX-FILE`, `REP-FILE` bzw. `SUBSYSTEM-INFO-FILE` ohne Benutzerkennung angegeben, so werden die Dateien auf der Installations-Benutzerkennung des Subsystems gesucht.
- Änderungen der Operanden `INIT-ROUTINE`, `CLOSE-CTRL-ROUTINE`, `STOPCOM-ROUTINE` bzw. `DEINIT-ROUTINE` werden nur akzeptiert, wenn die Arbeitsfunktion des Subsystems nicht beeinträchtigt wird.
- Kann eine Änderung nicht durchgeführt werden, so ergeht eine entsprechende Meldung nach `SYSOUT`. Meldungen, die akzeptierte Änderungen zum Inhalt haben, werden in die `CONSLOG`-Datei eingetragen.

Beispiel

Das Subsystem `ARCHIVE V2.8` soll automatisch gestartet werden, sobald der erste `SVC`-Aufruf erfolgt:

```
/MODIFY-SUBSYSTEM-PARAMETER SUBSYSTEM-NAME=ARCHIVE,VERSION='02.8', -  
      CREATION-TIME=*AT-SUBSYSTEM-CALL
```

RELEASE-SUBSYSTEM-SPACE

Reservierten Adressraum von Subsystemen freigeben

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	STD-PROCESSING HARDWARE-MAINTENANCE SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Mit dem Kommando RELEASE-SUBSYSTEM-SPACE verzichtet ein Benutzer während einer Taskdauer auf eine Subsystemgruppe, für die mit SCOPE=*GLOBAL ein Teil des Klasse-5-Speichers reserviert wurde. Dadurch wird die Reservierung des entsprechenden Adressraumes aufgehoben, und der Adressbereich kann für andere Zwecke verwendet werden.

Format

RELEASE-SUBSYSTEM-SPACE

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	32	ESM0423	Problem der Speicherplatzverwaltung
	32	ESM0424	Interner DSSM-Fehler

REMOVE-SUBSYSTEM

Inaktives Subsystem aus dynamischem Katalog entfernen

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Die Systembetreuung kann im laufenden System mit diesem Kommando ein inaktives Subsystem aus dem aktuellen dynamischen Subsystemkatalog entfernen. Da dies nur logisch geschieht, verändert sich die Anzahl der Subsysteme und CALL-Entries nicht, die im Anschluss an dieses Kommando über Kommando ADD-SUBSYSTEM dem aktuellen Subsystemkatalog hinzugefügt werden können (Maximalanzahl: 1000 Subsysteme und 16000 CALL-Entries).

Das Kommando wird abgewiesen, wenn:

- das zu entfernende Subsystem aktiv ist
- vonseiten eines anderen Subsystems Verweise oder Abhängigkeitsbeziehungen bestehen

Format

REMOVE-SUBSYSTEM
SUBSYSTEM-NAME = <structured-name 1..8> ,VERSION = <product-version mandatory-man-corr> / <product-version without-man-corr>

Operandenbeschreibung

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

Name des zu entfernenden Subsystems.

VERSION = <product-version mandatory-man-corr> /

<product-version without-man-corr>

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrektur-

stand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando ohne Fehler ausgeführt
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0646	Kommandobearbeitung abnormal beendet; während des Vorrouten-Updates oder der Berechnung des Standard-Version ist ein Fehler aufgetreten.
	64	ESM0642	Kommando nicht ausgeführt

Hinweise

- Sobald das Subsystem entfernt ist, existieren dessen Verweise und Abhängigkeitsbeziehungen zu einem anderen Subsystem nicht mehr.
- Wird die neueste Version eines nicht-privilegierten Subsystems entfernt, so sind alle das Subsystem betreffenden Verweise und Abhängigkeitsbeziehungen hinfällig.
- Sobald ein Subsystem mit REMOVE-SUBSYSTEM entfernt wurde, ist eine dynamische Erweiterung des aktuellen Subsystemkatalogs mit dem Kommando ADD-SUBSYSTEM unter Angabe von TYPE=*EXTENDED-ACTIVE-CONFIGURATION nicht möglich. Die dynamische Erweiterung ist jedoch mit Angabe von TYPE=*NEW-SUBSYSTEMS im Kommando ADD-SUBSYSTEM möglich.

Beispiel

Das Subsystem DAB, Version 6.0, soll entfernt werden:

```
/REMOVE-SUBSYSTEM SUBSYSTEM-NAME=DAB,VERSION='06.0'
```

RESUME-SUBSYSTEM

Wartezustand für ein Subsystem aufheben

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	OPERATING SUBSYSTEM-MANAGEMENT
Berechtigungsschlüssel:	R

Funktionsbeschreibung

Mit dem Kommando RESUME-SUBSYSTEM hebt die Systembetreuung den Wartezustand für ein beliebiges Subsystem auf.

Nach erfolgreicher Ausführung des Kommandos lassen sich zu dem angegebenen Subsystem wieder Verbindungen aufbauen. Voraussetzung hierfür ist, dass das Subsystem zuvor durch ein HOLD-SUBSYSTEM-Kommando in einen definierten Wartezustand versetzt wurde. Gewährleistet ist somit, dass alle notwendigen Ressourcen (Holdertask, Adressraum) verfügbar geblieben sind und die Initialisierungsroutine ablaufen kann.

Format

RESUME-SUBSYSTEM
<p>SUBSYSTEM-NAME = <structured-name 1..8></p> <p>,VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST</p> <p>,SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254></p> <p>,RESET = *NO / *YES</p> <p>,SYNCHRONOUS = *NO / *YES</p>

Operandenbeschreibung

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

Name des Subsystems, dessen Wartezustand aufgehoben wird.

VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturzustand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = *STD

Existiert nur **eine** Version des Subsystems, die sich im Wartezustand befindet, gilt der Standardwert für diese Version.

Existieren **mehrere** passende Versionen, muss die Version spezifiziert werden.

VERSION = *HIGHEST

Es wird die höchste Version des Subsystems ausgewählt.

SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>

Vereinbart, ob spezielle Parameter verarbeitet werden, die nur das angegebene Subsystem auswerten kann.

RESET =

Beeinflusst Verhalten und Dringlichkeit der Kommandobearbeitung.

RESET = *NO

Befindet sich das betreffende Subsystem noch nicht in einem definierten Wartezustand, wird das Kommando solange abgewiesen, bis es diesen erreicht hat.

RESET = *YES

Das Kommando wird ohne Rücksicht auf einen evtl. noch ausstehenden Abbau-Prozess akzeptiert und das Subsystem oder einige Komponenten sofort initialisiert (siehe auch folgende „Hinweise“).

SYNCHRONOUS =

Erlaubt die Wahl zwischen synchroner und asynchroner Verarbeitung.

SYNCHRONOUS = *NO

Das Kommando soll asynchron verarbeitet werden, d.h die Ausführung des Kommandos muss nicht abgewartet werden. Fehlermeldungen über den Ablauf des Kommandos werden nicht ausgegeben.

SYNCHRONOUS = *YES

Die Ausführung des Kommandos muss abgewartet werden.
Entsprechende Fehlermeldungen über den Ablauf werden ausgegeben.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	CMD0001	Keine Aktion notwendig; Subsystem ist bereits im Wartezustand
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0224	Kommando wird nicht verarbeitet
	32	ESM0228	Kommando abnormal beendet

Hinweise

- Um ein hohes Maß an Parallelität und Datenintegrität zu gewährleisten, werden zeitaufwändige Verwaltungsaufgaben nicht unter der Kontrolle der aufrufenden Task ausgeübt, sondern einer DSSM-Task übertragen.
In der Regel wird nur die Prüfung der geforderten Funktion **synchron** (d.h. verbunden mit einem Wartezustand für die aufrufende Task) realisiert. Die eigentliche Verarbeitung jedoch führt DSSM **asynchron** und unabhängig von der aufrufenden Task durch.
- Nach dem Kommando HOLD-SUBSYSTEM wird RESUME-SUBSYSTEM abgewiesen, wenn DSSM die Aktion „Subsystem anhalten“ noch nicht vollständig durchführen konnte. Mit dem Operanden RESET=*YES kann die Systembetreuung jedoch das unbedingte Aufheben des Wartezustands für das Subsystem erreichen; die vollständige Abarbeitung eines HOLD-SUBSYSTEM-Kommandos muss nicht abgewartet werden.
In diesem Fall wird die Initialisierungsroutine eingeleitet. Das betreffende Subsystem, das über den RESET informiert wird, kann den Umfang der Initialisierungsroutine (vollständige Initialisierung, Teil-Initialisierung, keine Initialisierung) selbst festlegen.
Sollen zwei Versionen eines Subsystems ausgetauscht werden, ist bei Verwendung des Parameters RESET=*YES auf Folgendes zu achten:
 - Befindet sich Version A im Zustand IN-DELETE und Version B im Zustand CREATED, ist RESET=*YES für A nur zulässig, wenn für beide Versionen bei der Definition mit SSCM Koexistenz zugelassen wurde (siehe [Seite 47](#)).
 - Befinden sich beide Versionen im Zustand IN-DELETE, ist RESET=*YES für eine dieser Versionen zulässig, wenn diese definiert wurde mit RESET=*ALLOWED, VERSION-EXCHANGE=*ALLOWED.

SAVE-SUBSYSTEM-CATALOG

Änderungen des dynamischen Subsystemkatalogs sichern

Anwendungsbereich: SYSTEM-MANAGEMENT
Privilegierung: SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Der mit dem Privileg SUBSYSTEM-MANAGEMENT ausgestattete privilegierte Benutzer kann mit diesem Kommando in den dynamischen Subsystemkatalog eingebrachte Änderungen in einen statischen Subsystemkatalog sichern.

Über die Kommandos ADD-SUBSYSTEM, REMOVE-SUBSYSTEM oder MODIFY-SUBSYSTEM-PARAMETER erfolgte Änderungen betreffen grundsätzlich nur den dynamischen, nicht jedoch den statischen Subsystemkatalog.

Mit dem Kommando SAVE-SUBSYSTEM-CATALOG können die Änderungen in einen statischen Katalog gesichert werden. Sie sind damit auch beim nächsten Startup wirksam. Dabei ist zu beachten, dass manche Änderungen beim nächsten Startup möglicherweise sinnlos oder sogar nachteilig sein können (z.B. wenn eine Meldungsdatei einem Subsystem mit dem Start-Attribut BEFORE-DSSM-LOAD¹ zugeordnet wird).

Format

SAVE-SUBSYSTEM-CATALOG

CATALOG-NAME = ***STD** / ***STARTUP-CATALOG** / <filename 1..54 without-gen-vers>
,**FORCED** = ***NO** / ***YES**

¹ Das Attribut BEFORE-DSSM-LOAD kann nur über SSCM verändert werden, siehe [Kapitel „SSCM“ auf Seite 179](#).

Operandenbeschreibung

CATALOG-NAME = *STD / *STARTUP-CATALOG /

<filename 1..54 without-gen-vers>

Legt den Namen der Datei fest, in die der dynamische Katalog gesichert werden soll.

CATALOG-NAME = *STD

Der dynamische Katalog wird unter dem Standard-Dateinamen \$.SYS.SSD.CAT.X abgespeichert.

CATALOG-NAME = *STARTUP-CATALOG

Der dynamische Katalog wird unter dem Namen des zum Startup-Zeitpunkt verwendeten Katalogs abgespeichert.

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

Der hier angegebene Dateiname wird für den statischen Katalog festgelegt.

FORCED = *NO / *YES

Legt fest, ob der dynamische Katalog trotz Fehler in den statischen Katalog gesichert wird.

FORCED = *NO

Der fehlerhafte dynamische Katalog wird nicht in den statischen Katalog gesichert.

FORCED = *YES

Obwohl Fehler erkannt wurden, wird der dynamische Katalog in den statischen Katalog gesichert.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando erfolgreich ausgeführt
	32	ESM0288	DSSM-Börse nicht verfügbar
	32	ESM0296	Speicherplatzanforderung (REQM) nicht ausgeführt
	32	ESM0350	Interner DSSM-Fehler; DSSM-Task wird neu gestartet
	32	ESM0360	Fehler wegen Speichermangels
	32	ESM0409	DSSM nicht initialisiert
	32	ESM0643	Interner Fehler während des Sicherungsprozesses
	64	ESM0648	Kommando nicht ausgeführt

Hinweise

- Der zu sichernde dynamische Katalog kann aus mehreren Gründen inkonsistent sein. Zum einen kann der von DSSM benötigte Katalog unter Verwendung des Operanden `FORCED=*YES` gesichert sein, wodurch sich Inkonsistenzen zwischen den Subsystemen ergeben, zum anderen können durch `/MODIFY-SUBSYSTEM-PARAMETER` veranlasste Änderungen beim BS2000-Startup inakzeptabel sein, während sie in der laufenden BS2000-Sitzung akzeptiert werden. Aus diesen Gründen muss der Katalog verschiedenen Prüfungen standhalten, bevor er gesichert wird. Werden bei diesen Überprüfungen Fehler festgestellt, so werden diese gemeldet. Darüberhinaus ergeht eine entsprechende Meldung nach `SYSOUT`.
- Wurde der dynamische Katalog ohne aufgetretene Inkonsistenzen gesichert, so lässt sich daraus nicht ableiten, dass der nächste mit diesem Katalog durchgeführte Startup erfolgreich verlaufen wird. Ist z.B. über ein vorhergehendes `/MODIFY-SUBSYSTEM-PARAMETER` der Startzeitpunkt (`CREATION-TIME`) eines Subsystems so verändert worden, dass das Subsystem beim BS2000-Systemstart nicht mehr automatisch gestartet wird, so können daraus für andere Subsysteme evtl. ernsthafte Probleme erwachsen.
- Wenn beim Sichern des Katalogs ein DVS-Fehler bzgl. der Katalogdatei auftritt, wird die Meldung `ESM1806` ausgegeben. Das Ergebnis der Sicherung ist dementsprechend zu prüfen. Wird die gleiche Meldung bzgl. eines beteiligten Subsystems ausgegeben, ist sie lediglich als Warnung zu interpretieren und hat keinen Einfluss auf das Ergebnis von `/SAVE-SUBSYSTEM-CATALOG`.
- Stimmt der angegebene Katalogname mit dem Namen einer bereits existierenden Datei überein, so wird eine Meldung mit der Frage ausgegeben, ob der Benutzer dem Überschreiben dieser Datei zustimmt.
- Können Funktionen nicht korrekt bearbeitet werden, so ergehen entsprechende Meldungen nach `SYSOUT`.

Beispiel

Der dynamische Katalog soll als statischer Katalog unter dem Dateinamen NEW.STATIC.CAT abgespeichert werden, sofern er keine Fehler enthält:

```

/SAVE-SUBSYSTEM-CATALOG CATALOG-NAME=COPY.DSSMCAT, FORCED=*NO
CHECK REPORT:
**** NO ERROR ****
CHECK OF LINK REFERENCES:
VERSION RANGE CHECK:
**** NO ERROR ****
LINK RELATION CHECK:
**** NO ERROR ****
CHECK OF FUNCTIONAL DEPENDENCE:
VERSION RANGE CHECK:
**** NO ERROR ****
DEPENDENCE RELATION CHECK:
**** NO ERROR ****
CYCLE CHECK:
**** NO ERROR ****
CHECK OF RELATED FILES:
*****
*   2 * SUBSYSTEM NAME:      ACS      VERSION:   14.0      *
*****
**** NO ERROR ****
*****
*   3 * SUBSYSTEM NAME:      ADAM     VERSION:  14.0A00   *
*****
**** NO ERROR ****
.
.
.
*****
*  176 * SUBSYSTEM NAME:    CRTEBASR  VERSION:   01.3    *
*****
**** NO ERROR ****
% ESM1200 CATALOG ' :CAM1:$TSOS.COPY.DSSMCAT' GENERATED
% ESM0254 COMMAND 'SAVE-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED

```

SET-DSSM-OPTIONS

Protokollierungsfunktion von DSSM aus- oder einschalten

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	OPERATING SUBSYSTEM-MANAGEMENT
Berechtigungsschlüssel:	R

Funktionsbeschreibung

Mit dem Kommando kann die Protokollierungsfunktion von DSSM gesteuert werden. Die Protokollierung in die Datei DSSMLOG geht zulasten der Performance. Deshalb sollte diese Funktion nur im Fall, dass tatsächlich Fehler auftreten, eingeschaltet werden.

Das Kommando kann unabhängig vom Zustand der Subsystem-Betreuung gegeben werden. Zum Zeitpunkt des Systemstarts ist die Protokollierung ausgeschaltet (Standardwert LOGGING=*OFF). Sie kann über den Startup-Parameter LOGGING=*ON aktiviert werden (siehe auch [Seite 64](#)).

Format

SET-DSSM-OPTIONS
<pre>LOGGING = *OFF / *ON , TITLE = *NONE / <c-string 1..100></pre>

Operandenbeschreibung

LOGGING =

Steuert, ob eine DSSM-spezifische Protokollierung für die Fehlerdiagnose durchgeführt wird.

LOGGING = *OFF

Es findet keine DSSM-spezifische Protokollierung statt.

LOGGING = *ON

Alle für die Fehlerdiagnose relevanten DSSM-spezifischen Daten werden in der Datei DSSMLOG.<date>.<time> protokolliert.

TITLE =

Vereinbarung einer Kopfzeile, die in die Protokolldatei aufgenommen wird.

TITLE = *NONE

Es soll kein Text zusätzlich zur Protokollierung in der Logging-Datei erfasst werden.

TITLE = <c-string 1..100>

Der angegebene Text wird als erster Datensatz in die Logging-Datei geschrieben.

Ist diese bereits eröffnet, wird keine neue Datei erzeugt, sondern an der aktuellen Position fortgeschrieben.

Bei Ausschaltung der Protokollierungsfunktion wird dieser Operand ignoriert.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	32	ESM0432	Kommando nicht ausgeführt

SHOW-SUBSYSTEM-ATTRIBUTES

Informationen über Subsystem-Attribute anzeigen

Anwendungsbereich: SYSTEM-MANAGEMENT

Privilegierung: STD-PROCESSING
SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Das Kommando informiert den Benutzer über die Attribute von globalen und lokalen Subsystemen.

Die nachfolgende Tabelle zeigt, welche Informationen über welchen Subsystem-Typ (global oder lokal) je nach Privilegierung des Benutzers ausgegeben werden.

Privilegierung	Informationen	Subsystem-Typ
STD-PROCESSING	Attribute nicht-privilegierter Subsysteme	lokal und global im Klasse-5-Speicher
SUBSYSTEM-MANAGEMENT	Attribute aller Subsysteme	global

Das Kommando unterstützt die strukturierte Ausgabe in S-Variablen (siehe Handbuch „Kommandos Band 6“ [20] und [Seite 128](#)).

Format

SHOW-SUBSYSTEM-ATTRIBUTES
<p>SUBSYSTEM-NAME = *ALL / <structured-name 1..8></p> <p>,VERSION = *ALL / <product-version mandatory-man-corr> / <product-version without-man-corr></p> <p>,INFORMATION = *MINIMUM / *ALL-ATTRIBUTES / *PARAMETERS(...)</p> <p> *PARAMETERS(...)</p> <p> GENERAL-ATTRIBUTES = *NO / *YES</p> <p> ,INTERNAL-ENTRIES = *NO / *YES</p> <p> ,MEMORY-ATTRIBUTES = *NO / *YES</p> <p> ,RELATED-FILES = *NO / *YES</p> <p> ,LINK-ATTRIBUTES = *NO / *YES</p> <p> ,REFERENCE-RELATION = *NO / *YES</p> <p> ,DEPENDENCE-RELATION = *NO / *YES</p> <p> ,HOLDER-TASK-INFO = *NO / *YES</p> <p> ,SUBSYSTEM-ENTRIES = *NO / *YES</p> <p>,OUTPUT = *SYSOUT / *SYSLST(...) / *NONE</p> <p> *SYSLST(...)</p> <p> SYSLST-NUMBER = *STD / <integer 1..99></p> <p> ,LINES-PER-PAGE = <integer 1..99></p>

Operandenbeschreibung

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

Vereinbart, über welche Subsysteme Informationen angefordert werden.

SUBSYSTEM-NAME = *ALL

Es sollen Informationen über alle Subsysteme angefordert werden, die im Katalog verzeichnet sind (abhängig von der Privilegierung).

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

Name des Subsystems, über das SSCM die Informationen aus dem Katalog bereitstellt.

VERSION = *ALL / <product-version mandatory-man-corr> / <product-version without-man-corr>

Vereinbart die Version des ausgewählten Subsystems.

VERSION = *ALL

Die Attribute aller im Katalog gespeicherten Versionen des Subsystems sollen in die Informationsausgabe eingeschlossen werden. Die Ausgabemenge ist abhängig von der Privilegierung des Benutzers.

**VERSION = <product-version mandatory-man-corr> /
 <product-version without-man-corr>**

Das hier angegebene Format muss mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturzustand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

INFORMATION = *MINIMUM / *ALL-ATTRIBUTES / *PARAMETERS(...)

Vereinbart den Umfang der Informationsausgabe.

INFORMATION = *MINIMUM

Es werden nur Name, Version und Status des Subsystems ausgegeben.

INFORMATION = *ALL-ATTRIBUTES

Es werden alle Informationen über das Subsystem ausgegeben.

INFORMATION = *PARAMETERS(...)

Gibt an, welche Informationen gewünscht sind.

GENERAL-ATTRIBUTES = *NO / *YES

Vereinbart, ob folgende generelle Eigenschaften der genannten Subsysteme aus dem Katalog gelesen werden sollen (*YES) oder nicht (*NO):

- Wann soll das Subsystem nach Systemeinführung gestartet werden?
(CREATION-TIME)
- In welchem Lademodus soll das Subsystem geladen werden?
(SUBSYSTEM-LOAD-MODE)
- Soll das Subsystem bei Shutdown automatisch entladen werden?
(STOP-AT-SHUTDOWN)
- Darf das geladene Subsystem angehalten oder entladen werden?
(SUBSYSTEM-HOLD)
- Dürfen die Kommandos zur Steuerung des Subsystems verwendet werden?
(STATE-CHANGE-CMDS)
- Ist die FORCE-Option zugelassen? (FORCED-STATE-CHANGE)
- Ist die RESET-Option zugelassen? (RESET)
- Muss bei abnormaler Beendigung der Holdertask die Initialisierungsroutine durchlaufen werden? (RESTART-REQUIRED)
- Darf mehr als eine Version des Subsystems gleichzeitig aktiv sein?
(VERSION-COEXISTENCE)
- Dürfen zwei Versionen eines Subsystems dynamisch ausgetauscht werden?
(VERSION-EXCHANGE)

- Wie lautet der Name der INSTALLATION-UNIT des Subsystems? (INSTALLATION-UNIT)
- Wie lautet das Copyright (Text und Datum) des Subsystems? (COPYRIGHT)

INTERNAL-ENTRIES = *NO / *YES

Vereinbart, ob folgende Informationen über die Einsprungstellen der angegebenen Subsysteme von SSCM bereitgestellt werden sollen (*YES) oder nicht (*NO):

- die Namen der Einsprungstellen für die Subsystemroutinen INIT, STOPCOM, DEINIT und CLOSE-CRTL
- der Name der Einsprungstelle, die für die dynamische Identitätsprüfung herangezogen wird (DYNAMIC-CHECK-ENTRY)
- der Name der Schnittstellenversion für den Aufruf der INIT-, STOPCOM-, DEINIT- und CLOSE-CTRL-Routinen (INTERFACE-VERSION)

MEMORY-ATTRIBUTES = *NO / *YES

Vereinbart, ob folgende Speicher-relevanten Informationen, die im Katalog über die Subsysteme gespeichert sind, ausgegeben werden sollen (*YES) oder nicht (*NO): Die Speicher-relevanten Informationen sind:

- Speicherklasse (MEMORY-CLASS)
- Größe des benötigten Adressraums (SIZE)
- Anfangsadresse des Subsystemcodes (START-ADDRESS)
- Privilegierung und Zugriffsberechtigung bezüglich Adressraum (SUBSYSTEM-ACCESS)

RELATED-FILES = *NO / *YES

Vereinbart, ob Informationen über die Nebenkompenten des Subsystems geliefert werden sollen (*YES) oder nicht (*NO). In die Ausgabe eingeschlossen ist die Information, ob die Verwendung einer Rep-Datei für dieses Subsystem verpflichtend ist (REP-FILE-MANDATORY) und unter welcher Benutzerkennung die Nebenkompenten katalogisiert sind (INSTALLATION-USERID).

Unter dem Begriff Nebenkompenten werden zusammengefasst:

- die Objektmoduldatei des Subsystems (LIBRARY)
- die Meldungsdatei (MESSAGE-FILE)
- die Syntaxdatei (SYNTAX-FILE)
- die Informationsdatei des Subsystems (SUBSYSTEM-INFO-FILE)
- die Rep-Datei (REP-FILE)

LINK-ATTRIBUTES = *NO / *YES

Vereinbart, ob die Informationen aus dem Katalog zu lesen sind (*YES) oder nicht zu lesen sind (*NO), die in Bezug auf das Binden und Laden des Subsystems gespeichert wurden:

- der Name des zum Laden benötigten Bindemoduls/ENTRY/CSECT (LINK-ENTRY)
- die Einbindung der Autolink-Funktion (AUTOLINK)
- die Informationen über das Verhalten bei nicht auflösbaren Externverweisen (UNRESOLVED)
- die Einbindung des Prüflaufs für Referenz-Subsysteme (CHECK-REFERENCE)

REFERENCE-RELATION = *NO / *YES

Vereinbart, ob die Liste der Subsysteme, zu denen Adressbeziehungen bestehen, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

DEPENDENCE-RELATION = *NO / *YES

Vereinbart, ob die Liste der Subsysteme, zu denen Abhängigkeitsbeziehungen bestehen, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

HOLDER-TASK-INFO = *NO / *YES

Vereinbart, ob die Identifikation der Holdertask und die Liste der Subsysteme, die in einer gemeinsamen Holdertask anzulegen sind, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

SUBSYSTEM-ENTRIES = *NO / *YES

Vereinbart, ob die Liste der bei der Definition des Subsystems vereinbarten Auftrags-eingänge und deren folgende Eigenschaften aus dem Katalog gelesen werden soll (*YES) oder nicht (*NO):

- Typ des vereinbarten Auftragseingangs (MODE)
- Nummer der Routine (bei *SVC oder *SYSTEM-EXIT) (NUMBER)
- die Funktionsnummer der Einsprungstelle (FUNCTION-NUMBER)
- die Version der Funktionsnummer (FUNCTION-VERSION)
- die Information über Aufruf durch System-Exit-Routinen (CALL-BY-SYSTEM-EXIT)
- die Privilegierung und Zugriffsberechtigung bezüglich Einsprungstellen (CONNECTION-ACCESS und CONNECTION-SCOPE)

OUTPUT = *SYSOUT / *SYSLST(...) / *NONE

Gibt an, in welche Systemdatei die Informationen ausgegeben werden sollen.

OUTPUT = *SYSOUT

Die Informationen werden nach SYSOUT ausgegeben.

OUTPUT = *SYSLST(...)

Die Informationen werden nach SYSLST ausgegeben.

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

Gibt die Nummer der SYSLST-Datei an, in die die Informationen ausgegeben werden sollen.

LINES-PER-PAGE = <integer 1..99>

Gibt die Anzahl Zeilen einer SYSLST-Seite an.

OUTPUT = *NONE

Es erfolgt keine Ausgabe nach SYSOUT oder SYSLST. Es werden nur S-Variablen erzeugt (siehe unten). Zu S-Variablen siehe auch Handbuch „Kommandos Band 6“ [20].

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	CMD2009	Systemfehler: beim Schreiben des S-Variablen-Bereichs
	32	ESM0360	Systemfehler: XVT oder TCB nicht erreichbar
	32	ESM0602	Systemfehler: Probleme mit Speicherverwaltung
	32	ESM0611	Systemfehler: Anschluss-Tabellen für diese Task existieren nicht
	32	ESM0671	Systemfehler: beim Schreiben nach SYSOUT
	64	ESM0600	Operandenfehler: keine Version mit *ALL
	64	ESM0601	Angegebenes System nicht gefunden
	64	ESM0608	Subsystemversion nicht gefunden
	64	OPS0002	Kommando unterbrochen
	130	OPS0001	Kommando nicht ausgeführt wegen fehlendem Speicherplatz: Kommando später wiederholen

S-Variablen

Die in die S-Variablen ausgegebenen Informationen stimmen mit der Ausgabe nach SYSOUT bzw. SYSLST überein (siehe Operandenbeschreibung oben). In der folgenden Tabelle sind die S-Variablen in Gruppen zusammengefasst, entsprechend ihrer Zuordnung zu den Unteroperanden von INFORMATION=*PARAMETERS(...).

Schreibweise im Kommando	gekürzte Schreibweise in Tabelle
INFORMATION=*PARAMETERS(GENERAL-ATTRIBUTES=*NO/*YES)	GEN-ATT
INFORMATION=*PARAMETERS(INTERNAL-ENTRIES=*NO/*YES)	INT-ENT
INFORMATION=*PARAMETERS(MEMORY-ATTRIBUTES=*NO/*YES)	MEM-ATT
INFORMATION=*PARAMETERS(RELATED-FILES=*NO/*YES)	REL-FIL
INFORMATION=*PARAMETERS(LINK-ATTRIBUTES=*NO/*YES)	LINK-ATT
INFORMATION=*PARAMETERS(REFERENCE-RELATION=*NO/*YES)	REF-REL
INFORMATION=*PARAMETERS(DEPENDENCE-RELATION=*NO/*YES)	DEP-REL
INFORMATION=*PARAMETERS(HOLDER-TASK-INFO=*NO/*YES)	HOL-TASK
INFORMATION=*PARAMETERS(SUBSYSTEM-ENTRIES=*NO/*YES)	SUB-ENT

(Teil 1 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Einbindung der Autolink-Funktion	var(*LIST).AUTOLINK	S	*ALLOW *FORBID	LINK-ENT
Einbindung des Prüflaufs für Referenz-Subsysteme	var(*LIST).CHECK-REF	S	*YES *NO	LINK-ENT
Adresse des Namens der Einsprungstelle für die Subsystemroutine CLOSE-CTRL-ROUTINE (wenn CRE)	var(*LIST).CLOSE-CTRL-ROUT.ADDR	S	<text 1..8>	INT-ENT
Name der Einsprungstelle für die Subsystemroutine CLOSE-CTRL-ROUTINE	var(*LIST).CLOSE-CTRL-ROUT.NAME	S	<text 1..8> *NO *DYN	INT-ENT
Copyright (Text und Datum) des Subsystems	var(*LIST).COPYRIGHT	S	<string 1..54> *NONE	GEN-ATT

(Teil 2 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Startzeitpunkt des Subsystems nach Systemeinführung	var(*LIST).CRE-TIME	S	*BEFORE-DSSM-LOAD *AT-DSSM-LOAD *MANDATORY-AT-STARTUP *BEFORE-SYS-READY *AFTER-SYS-READY *AT-CRE-REQ *AT-SUBSYS-CALL	GEN-ATT
Name des Subsystems	var(*LIST).DATA(*LIST).SUBSYS-NAME	S	<structured-name 1..8>	
Version des Subsystems	var(*LIST).DATA(*LIST).SUBSYS-VERSION	S	<product-version>	
Adresse des Namens der Einsprungstelle für die Subsystemroutine DEINIT (wenn CRE)	var(*LIST).DEINIT-ROUT.ADDR	S	<text 1..8>	INT-ENT
Name der Einsprungstelle für die Subsystemroutine DEINIT	var(*LIST).DEINIT-ROUT.NAME	S	<text 1..8> *NO *DYN	INT-ENT
Adresse des Namens der Einsprungstelle, die für die dynamische Identitätsprüfung herangezogen wird (wenn CRE)	var(*LIST).DYN-CHECK-ENTRY-NAME.ADDR	S	<text 1..8>	INT-ENT
Name der Einsprungstelle, die für die dynamische Identitätsprüfung herangezogen wird	var(*LIST).DYN-CHECK-ENTRY-NAME.NAME	S	<text 1..8> *NO	INT-ENT
Zulassen der FORCE-Option	var(*LIST).FORCED-STATE-CHA	S	*ALLOW *FORBID	GEN-ATT
Subsystemname in der gemeinsamen Holdertask	var(*LIST).HOLDER-TASK.SHARED-WITH-SUBSYS(*LIST).SUBSYS-NAME	S	<structured-name 1..8> *WORK-TASK	HOL-TASK
Produktversion des Subsystems in der gemeinsamen Holder-Task	var(*LIST).HOLDER-TASK.SHARED-WITH-SUBSYS(*LIST).SUBSYS-VERSION	S	<product-version>	HOL-TASK
TID der Holder-Task (wenn CREATED)	var(*LIST).HOLDER-TASK.TID	S	<text 8>	HOL-TASK
TSN der Holder-Task	var(*LIST).HOLDER-TASK.TSN	S	<text 4>	HOL-TASK
Adresse des Namens der Einsprungstelle für die Subsystemroutine INIT (wenn CRE)	var(*LIST).INIT-ROUT.ADDR	S	<text 1..8>	INT-ENT
Name der Einsprungstelle für die Subsystemroutine INIT	var(*LIST).INIT-ROUT.NAME	S	<text 1..8> *NO	INT-ENT

(Teil 3 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Name der INSTALLATION-UNIT des Subsystems	var(*LIST).INSTALL-UNIT	S	<text 1..30> *NONE	GEN-ATT
Benutzerkennung, unter der die Nebenkompenten katalogisiert sind	var(*LIST).INSTALL-USERID	S	<name 1..8> *NONE *DEF	REL-FIL
Adresse des Namens der Schnittstellenversion für den Aufruf der INIT-/STOPCOM-/DEINIT-/CLOSE-CTRL-Routinen (wenn CRE)	var(*LIST).INTERF-VERSION.ADDR	S	<text 1..8>	INT-ENT
Name der Schnittstellenversion für den Aufruf der INIT-/STOPCOM-/DEINIT-/CLOSE-CTRL-Routinen	var(*LIST).INTERF-VERSION.NAME	S	<text 1..8> *NO	INT-ENT
Adresse des Namens des zum Laden benötigten Bindemoduls/ENTRY/CSECT (wenn CRE)	var(*LIST).LINK-ENTRY.ADDR	S	<text 1..8>	LINK-ENT
Name des zum Laden benötigten Bindemoduls/ENTRY/CSECT	var(*LIST).LINK-ENTRY.NAME	S	<text 1..8>	LINK-ENT
Speicherklasse	var(*LIST).MEM-CLASS	S	*SYS-GBL *LOC-PRIVIL *LOC-UNPRIVIL *BY-SLICE	MEM-ATT
Monitor-Jobvariable	var(*LIST).MONJV	S	*YES *NO	GEN-ATT
Defaultname der Meldungsdatei (für *INSTALL)	var(*LIST).MSG-F.DEF-NAME	S	<filename 1..54> *NONE	REL-FIL
Logic-Id der Meldungsdatei (für *INSTALL)	var(*LIST).MSG-F.LOGIC-ID	S	<filename 1..30>	REL-FIL
Name der Meldungsdatei	var(*LIST).MSG-F.NAME	S	<filename 1..54> *INSTALL *NO	REL-FIL
Kriterium für das automatische Laden, wenn als Startzeitpunkt CREATION-TIME=*AT-SUBSYS-CALL eingestellt ist: beim ersten Aufruf einer zugehörigen SVC-, ISL- oder einer beliebigen Schnittstelle	var(*LIST).ON-ACTION	S	*STD *ISL-CALL *ANY	GEN-ATT
Produktversion des Subsystems, zu dem Adressbeziehungen bestehen (*HIGH=höchste Produktversion)	var(*LIST).REF-SUBSYS(*LIST).HIGH-VERSION	S	<product-version> *HIGH	REF-REL

(Teil 4 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Produktversion des Subsystems, zu dem Adressbeziehungen bestehen (*LOW=niedrigste Produktversion)	var(*LIST).REF-SUBSYS(*LIST).LOW-VERSION	S	<product-version> *LOW	REF-REL
Name des Subsystems, zu dem Adressbeziehungen bestehen	var(*LIST).REF-SUBSYS(*LIST).SUBSYS-NAME	S	<structured-name 1..8>	REF-REL
Produktversion des Subsystems, zu dem Abhängigkeitsbeziehungen bestehen (*HIGH=höchste Produktversion)	var(*LIST).RELATED-SUBSYS(*LIST).HIGH-VERSION	S	<product-version> *HIGH	DEP-REL
Produktversion des Subsystems, zu dem Abhängigkeitsbeziehungen bestehen (*LOW=niedrigste Produktversion)	var(*LIST).RELATED-SUBSYS(*LIST).LOW-VERSION	S	<product-version> *LOW	DEP-REL
Name des Subsystems, zu dem Abhängigkeitsbeziehungen bestehen	var(*LIST).RELATED-SUBSYS(*LIST).SUBSYS-NAME	S	<structured-name 1..8>	DEP-REL
Defaultname der Rep-Datei (für *INSTALL)	var(*LIST).REP-F.DEF-NAME	S	<filename 1..54> *NONE	REL-FIL
Logic-Id der Rep-Datei (für *INSTALL)	var(*LIST).REP-F.LOGIC-ID	S	<filename 1..30>	REL-FIL
Verwendung einer Rep-Datei ist für dieses System verpflichtend	var(*LIST).REP-F.MANDATORY	S	*YES *NO	REL-FIL
Name der Rep-Datei	var(*LIST).REP-F.NAME	S	<filename 1..54> *INSTALL *NO	REL-FIL
RESET-Option zulassen	var(*LIST).RESET	S	*ALLOW *FORBID	GEN-ATT
Bei abnormaler Beendigung der Holdertask Durchlaufen der Initialisierungsroutine	var(*LIST).RESTART-REQ	S	*YES *NO	GEN-ATT
Größe des benötigten Adressraums (für *LOC-PRIVIL und *BY-SLICE)	var(*LIST).SIZE	I	<integer 1..32767>	MEM-ATT
Anfangsadresse des Subsystem-codes (für *LOC-UNPRIVIL)	var(*LIST).START-ADDR	S	<text 1..8>	MEM-ATT
Verwenden der Kommandos zur Steuerung des Subsystems	var(*LIST).STATE-CHA-CMDS	S	*ALLOW *FORBID *BY-ADM-ONLY	GEN-ATT
Soll das Subsystem bei Shutdown automatisch entladen werden	var(*LIST).STOP-AT-SHUTDOWN	S	*YES *NO	GEN-ATT

(Teil 5 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Adresse des Namens der Einsprungsstelle für die Subsystemroutine STOPCOM (wenn CRE)	var(*LIST).STOPCOM-ROUT.ADDR	S	<text 1..8>	INT-ENT
Name der Einsprungsstelle für die Subsystemroutine STOPCOM	var(*LIST).STOPCOM-ROUT.NAME	S	<text 1..8> *NO *DYN	INT-ENT
Privilegierung und Zugriffsberechtigung bezüglich Adressraum (für *SYS-GBL und *LOC-UNPRIVIL)	var(*LIST).SUBSYS-ACCESS	S	*LOW *SYS *HIGH	MEM-ATT
Adressen der Einsprungsstellen (wenn CREATED)	var(*LIST).SUBSYS-ENTRIES(*LIST).ADDR	S	<text 1..8>	SUB-ENT
Privilegierung bezüglich Einsprungsstellen	var(*LIST).SUBSYS-ENTRIES(*LIST).CONN-ACCESS	S	*ALL *SYS *SIH	SUB-ENT
Zugriffsberechtigung bezüglich Einsprungsstellen	var(*LIST).SUBSYS-ENTRIES(*LIST).CONN-SCOPE	S	*TASK *PROG *FREE *CALL *OPTIM	SUB-ENT
Zulassen von Einsprungsstellen beim ersten Verbindungsaufbau	var(*LIST).SUBSYS-ENTRIES(*LIST).FIRST-CONN	S	*ALLOW *FORBID	SUB-ENT
Funktionsnummer der Einsprungsstelle (für ISL oder SVC)	var(*LIST).SUBSYS-ENTRIES(*LIST).FUNC-NUM	I	<integer 0..255>	SUB-ENT
Version der Funktionsnummer (für ISL oder SVC)	var(*LIST).SUBSYS-ENTRIES(*LIST).FUNC-VERSION	I	<integer 1..255>	SUB-ENT
Typ des vereinbarten Auftragseingangs	var(*LIST).SUBSYS-ENTRIES(*LIST).MODE	S	*LINK *ISL *SVC *SYS-EXIT	SUB-ENT
Name der Einsprungsstelle	var(*LIST).SUBSYS-ENTRIES(*LIST).NAME	S	<text 1..8>	SUB-ENT
Das geladene Subsystem anhalten oder entladen	var(*LIST).SUBSYS-HOLD	S	*ALLOW *FORBID	GEN-ATT
Defaultname der Informationsdatei des Subsystems (für *INSTALL)	var(*LIST).SUBSYS-INFO-F.DEF-NAME	S	<filename 1..54> *NONE	REL-FIL
Logic-Id der Informationsdatei des Subsystems (für *INSTALL)	var(*LIST).SUBSYS-INFO-F.LOGIC-ID	S	<filename 1..30>	REL-FIL
Name der Informationsdatei des Subsystems	var(*LIST).SUBSYS-INFO-F.NAME	S	<filename 1..54> *INSTALL *NO	REL-FIL

(Teil 6 von 6)

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Interner Status	var(*LIST).SUBSYS-INT-STA	S	*INSTALLED *INITIALIZED *CONNECTABLE *WAIT-CLS-CTRL *WAIT-DISCON *WAIT-DEINIT *WAIT-STOP-COM	GEN-ATT
Defaultname Objektmoduldatei des Subsystems (für *INSTALL)	var(*LIST).SUBSYS-LIB.DEF-NAME	S	<filename 1..54>	REL-FIL
Logic-Id der Objektmoduldatei des Subsystems (für *INSTALL)	var(*LIST).SUBSYS-LIB.LOGIC-ID	S	<filename 1..30>	REL-FIL
Name der Objektmoduldatei des Subsystems	var(*LIST).SUBSYS-LIB.NAME	S	<filename 1..54> *INSTALL *CPLINK	REL-FIL
Lademodus, in dem das Subsystem geladen wird	var(*LIST).SUBSYS-LOAD-MODE	S	*STD *ADV *ANY	GEN-ATT
Status des Subsystems	var(*LIST).SUBSYS-STA	S	*NOT-CRE *IN-CRE *IN-HOLD *IN-DEL *IN-RESUME *NOT-RESUMED *CRE *LOCK	GEN-ATT
Defaultname der Syntaxdatei (für *INSTALL)	var(*LIST).SYNTAX-F.DEF-NAME	S	<filename 1..54> *NONE	REL-FIL
Logic-Id der Syntaxdatei (für *INSTALL)	var(*LIST).SYNTAX-F.LOGIC-ID	S	<filename 1..30>	REL-FIL
Name der Syntaxdatei	var(*LIST).SYNTAX-F.NAME	S	<filename 1..54> *INSTALL *NO	REL-FIL
Informationen über das Verhalten bei nicht auflösbaren Externverweisen	var(*LIST).UNRESOLVED-EXTERNAL	S	*ALLOW *FORBID	LINK-ENT
Mehr als eine Version des Subsystems gleichzeitig aktiv	var(*LIST).VERSION-COEXIST	S	*ALLOW *FORBID	GEN-ATT
Zwei Versionen eines Subsystems dynamisch austauschen	var(*LIST).VERSION-EXCHA	S	*ALLOW *FORBID	GEN-ATT
Jahreszahlangebe (wenn COPYRIGHT)	var(*LIST).YEAR	S	<string 4>	GEN-ATT

Hinweise

- /SHOW-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=*ALL zeigt dieselben Informationen wie /SHOW-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=*ALL, VERSION=*ALL.
- /SHOW-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=*ALL, VERSION=<version> wird nicht unterstützt.
- Wurde ein Subsystem im lokalen Subsystemkatalog gefunden, wird in der Ausgabe-Information das Wort „Subsystem“ durch „LOCAL SUBSYSTEM“ ersetzt.
- Hat ein Benutzer nur das Privileg *STD-PROCESSING, erhält er die Attribute des lokalen Subsystems (falls vorhanden) und die Attribute der nicht-privilegierten globalen Subsysteme (d.h. Subsysteme, für die SUBSYSTEM-ACCESS nicht *SYSTEM ist).
- Tritt beim Schreiben der Informationen nach SYSOUT oder SYSLST ein Fehler auf (ESM0671), wird das Schreiben der S-Variablen normal fortgesetzt.
- Tritt beim Schreiben der S-Variablen ein Fehler auf (OPS0001 oder CMD2009), wird das Schreiben der Informationen auf SYSOUT oder SYSLST normal fortgesetzt.
- Wird bei einer Eingabeanforderung (PLEASE ACKNOWLEDGE) die Unterbrechungstaste **[K2]** betätigt, wird sowohl die Ausgabe nach SYSOUT als auch die Ausgabe in S-Variablen sofort unterbrochen. Es wird der Returncode OPS0002 gesetzt.
- Die volle Übereinstimmung der Ausgaben nach SYSOUT/SYSLST kann nur dann garantiert werden, wenn die das Kommando absetzende Prozedur mit dem Operanden INTERRUPTION-ALLOWED=*NO eine Unterbrechung der Ausgabe verboten hat.

Beispiele

Beispiel 1

Ausgabe der Attribute nicht-privilegierter Subsysteme
(Benutzer mit Privileg *STD-PROCESSING)

```
/show-subsystem-attributes subsystem-name=*all
```

```
%*****
%* 3 * SUBSYSTEM NAME: ASSEMBH      VERSION: 01.2      *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
%*****
%* 4 * SUBSYSTEM NAME: ASSTRAN      VERSION: 01.7      *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
%*****
%* 5 * SUBSYSTEM NAME: ASSGENH      VERSION: 01.2      *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
%*****
%* 6 * SUBSYSTEM NAME: BUILDER      VERSION: 01.0      *
%*****
% STATUS OF THE SUBSYSTEM : CONNECTABLE
%*****
%* 7 * SUBSYSTEM NAME: CRTEBASR     VERSION: 01.3      *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
%*****
%* 8 * SUBSYSTEM NAME: CRTEBASY     VERSION: 01.3      *
%*****
% STATUS OF THE SUBSYSTEM : CONNECTABLE
%*****
.
.
.
.
%*****
%* 192 * SUBSYSTEM NAME: MONSYS     VERSION: 03.0B20   *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
%*****
%* 193 * SUBSYSTEM NAME: VAS-TU     VERSION: 02.1      *
%*****
% STATUS OF THE SUBSYSTEM : NOT CREATED
% ESM0608 SUBSYSTEM VERSION NOT FOUND. FUNCTION ABORTED
% ESM0255 'SHOW-SUBSYSTEM-ATTRIBUTES' COMMAND NOT PROCESSED
```

Beispiel 2

Ausgabe der Attribute nicht-privilegierter Subsysteme
(Benutzer mit Privileg *STD-PROCESSING)

```
/show-subsystem-attributes subsystem-name=assembh,information=*all-attributes
```

```
%*****
%* 3 * SUBSYSTEM NAME: ASSEMBH      VERSION: 01.2      *
%*****
%GENERAL ATTRIBUTES:
% INSTALLATION-UNIT      : **** NOT SPECIFIED ****
% COPYRIGHT              : FUJITSU SIEMENS COMPUTERS GMBH
% YEAR                  : 2000
% SUBSYSTEM-LOAD-MODE   : ADVANCED
% CREATION-TIME         : AT-CREATION-REQUEST
% STOP-AT-SHUTDOWN     : NO
% SUBSYSTEM-HOLD        : ALLOWED
% STATE-CHANGE-CMDS    : ALLOWED
% FORCED-STATE-CHANGE  : FORBIDDEN
% RESET                 : FORBIDDEN
% RESTART-REQUIRED     : NO
% VERSION-COEXISTENCE  : FORBIDDEN
% VERSION-EXCHANGE     : FORBIDDEN
% STATUS OF THE SUBSYSTEM : NOT CREATED
% MONITORING JOB VARIABLE : NO
%INTERNAL ENTRIES:
% INIT-ROUTINE          : **** NOT SPECIFIED ****
% CLOSE-CTRL-ROUTINE   : **** NOT SPECIFIED ****
% STOPCOM-ROUTINE      : **** NOT SPECIFIED ****
% DEINIT-ROUTINE       : **** NOT SPECIFIED ****
% DYNAMIC-CHECK-ENTRY  : **** NOT SPECIFIED ****
% INTERFACE-VERSION    : **** NOT SPECIFIED ****
%MEMORY ATTRIBUTES:
% MEMORY-CLASS: SYSTEM GLOBAL      SUBSYSTEM-ACCESS: HIGH
%RELATED FILES:
% INSTALLATION-USERID  : *DEFAULT-USERID
% LIBRARY              : $TSOS.SYSLNK.ASSEMBH.012
% MESSAGE-FILE         : **** NOT SPECIFIED ****
% SYNTAX-FILE          : **** NOT SPECIFIED ****
% SUBSYSTEM-INFO-FILE  : **** NOT SPECIFIED ****
% REP-FILE             : **** NOT SPECIFIED ****
% REP-FILE-MANDATORY  : NO
%LINK ATTRIBUTES:
% LINK-ENTRY: IARCOO
% AUTOLINK : ALLOWED      UNRESOLVED: FORBIDDEN      CHECK-REFERENCE: YES
%REFERENCED SUBSYSTEMS:
%      NAME      LOWEST VERSION      HIGHEST VERSION
%      ----      -
%      **** NONE ****
%FUNCTIONAL DEPENDENCE WITH SUBSYSTEMS:
%      NAME      LOWEST VERSION      HIGHEST VERSION
%      ----      -
%      **** NONE ****
```

Fortsetzung ➡

SHOW-SUBSYSTEM-INFO

Informationen über aktuelle Subsystem-Konfiguration anfordern

Anwendungsbereich: SYSTEM-MANAGEMENT
Privilegierung: SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Mit diesem Kommando kann die Systembetreuung Informationen über die aktuelle Subsystem-Konfiguration anfordern. Die Ausgabe, die nur auf SYSLST erfolgt, umfasst die komplette Beschreibung aller bekannten Subsysteme, einschließlich folgender dynamischer Aspekte:

- momentaner Zustand
- Ladeadresse von gestarteten Subsystemen
- Anzahl seit Startup registrierter Verbindungen
- Größe des Subsystems (Größe des beanspruchten Speicherplatzes einschließlich Subsystemcode und Metadaten)

Format

SHOW-SUBSYSTEM-INFO
SUBSYSTEM-NAME = *ALL

Operandenbeschreibung

SUBSYSTEM-NAME = *ALL

Die Ausgabe nach SYSLST umfasst die gesamte dynamische Subsystem-Konfiguration.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	32	ESM0288	Probleme mit Börsen-Kommunikation
	32	ESM0298	Probleme mit Speicher-Verwaltung
	32	ESM0350	Internes DSSM-Problem während der Verarbeitung
	32	ESM0670	Fehler beim Schreiben in die SYSLST-Datei

Bei der Ausgabe werden folgende Abkürzungen verwendet:

für CREATION-TIME

ACR : *AT-CREATION-REQUEST
 ASC : *AT-SUBSYSTEM-CALL
 ADL : *AT-DSSM-LOAD
 BDL : *BEFORE-DSSM-LOAD
 MAS : *MANDATORY-AT-STARTUP
 BSR : *BEFORE-SYSTEM-READY
 ASR : *AFTER-SYSTEM-READY

für MEMORY-CLASS

S : *SYSTEM-GLOBAL
 P : *LOCAL-PRIVILEGED
 U : *LOCAL-UNPRIVILEGED
 B : *BY-SLICE

für SUBSYSTEM-ACCESS

SYS : *SYSTEM
 ALL : *LOW / *HIGH

für INTERNAL-ENTRIES

DYN : *DYNAMIC
 YES : name
 NO : *NO

für CONNECTION-ACCESS

SYS : *SYSTEM

für STATE-CHANGE-CMDS

ADM : *BY-ADMINISTRATOR-ONLY

für REP-FILE

MAN : REP-FILE = *STD / dateiname und REP-FILE-MANDATORY = *YES

für GENERAL-ATTRIBUTES

YES : *ALLOWED
 NO : *FORBIDDEN

SHOW-SUBSYSTEM-STATUS

Informationen über den Zustand von Subsystemen anfordern

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	STD-PROCESSING OPERATING SUBSYSTEM-MANAGEMENT
Berechtigungsschlüssel:	R

Funktionsbeschreibung

Das Kommando SHOW-SUBSYSTEM-STATUS informiert den Benutzer über den Zustand von globalen Subsystemen, die allen Benutzern im Klasse-5-Speicher zur Verfügung gestellt werden (nicht-privilegierte Subsysteme) und über den Zustand seiner lokalen Subsysteme, wenn er einen lokalen Subsystemkatalog geladen hat.

Fehler die bei der Ausführung des Kommandos SHOW-SUBSYSTEM-STATUS auftreten, lösen keinen Spin-Off aus (siehe /SET-JOB-STEP, Handbuch „Kommandos Band 1-5“ [19]).

Das Kommando SHOW-SUBSYSTEM-STATUS gibt seine Informationen nicht nur nach SYSOUT aus. Es unterstützt auch die Ausgabe in S-Variablen, siehe [Seite 145](#). Der Benutzer kann diese dann in seinen eigenen S-Prozeduren auswerten und weiterverarbeiten. Zur Arbeitsweise mit S-Variablen siehe Handbuch „Kommandos Band 6“ [20].

In Abhängigkeit von der Privilegierung des Aufrufers werden verschiedene Informationen ausgegeben (siehe dazu auch die Tabelle auf der nächsten Seite):

Privilegierte Funktionen (Privileg SUBSYSTEM-MANAGEMENT oder OPERATING)

Die Systembetreuung kann mit dem Kommando Informationen über den Zustand von globalen - privilegierten und nicht-privilegierten - Subsystemen anfordern. Es kann zusätzlich zu Typ (global), Name, Version und Status des globalen Subsystems angezeigt werden:

- welche Tasks zum angegebenen Subsystem eine Verbindung haben (TSN und TID)
- die Anzahl der Verbindungen zu einem bezeichneten Subsystem seit Startup

Während der Kommandoausführung können weitere Aufträge eine Verbindung zu dem Subsystem auf- bzw. abbauen; die Liste der angezeigten Aufträge gibt deshalb unter Umständen nicht den aktuellen Stand wieder. Über lokale Subsysteme anderer Tasks können privilegierte Benutzer keine Auskunft erhalten.

Nicht-privilegierte Funktionen (Privileg STD-PROCESSING)

Es wird der Typ (global oder lokal), Name, Version und Status von nicht-privilegierten globalen oder lokalen Subsystemen angezeigt.

Hat ein Benutzer alle drei Privilegien, kann er Informationen über alle Subsysteme erhalten, egal ob lokal oder global, ob privilegiert oder nicht-privilegiert.

Folgende Informationen werden abhängig von den Privilegien des Aufrufers ausgegeben:

STD-PROCESSING	STD-PROCESSING und SUBSYSTEM-MGMT	OPERATING oder SUBSYSTEM-MGMT
die angeforderte Information betrifft nur lokal definierte Subsysteme* :		
a) alle lokalen Subsysteme	a) alle lokalen Subsysteme	keine Ausgabe
b) ist keines von ihnen im Zustand CREATED, dann alle lokalen NOT-CREATED-Subsysteme, aber ohne Versionsangabe	b) ist keines von ihnen im Zustand CREATED, dann alle lokalen NOT-CREATED-Subsysteme, aber ohne Versionsangabe	
die angeforderte Information betrifft lokal definierte und global definierte Subsysteme* :		
a) alle lokalen Subsysteme und alle globalen, nicht-privilegierten Subsysteme	a) alle lokalen Subsysteme und alle globalen Subsysteme mit den angeschlossenen Tasks, solange sie im Zustand CREATED, IN-HOLD oder IN-DELETE sind	a) keine lokalen Subsysteme, aber alle globalen Subsysteme mit den angeschlossenen Tasks, solange sie im Zustand CREATED, IN-HOLD oder IN-DELETE sind
b) sind alle von ihnen im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe	b) sind alle Subsysteme im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe	b) sind alle von ihnen im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe
die angeforderte Information betrifft nur global definierte Subsysteme:		
a) alle globalen, nicht-privilegierten Subsysteme	a) alle globalen Subsysteme mit den angeschlossenen Tasks, solange sie im Zustand CREATED, IN-HOLD oder IN-DELETE sind	a) alle globalen Subsysteme mit den angeschlossenen Tasks, solange sie im Zustand CREATED, IN-HOLD oder IN-DELETE sind
b) sind alle von ihnen im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe	b) sind alle von ihnen im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe	b) sind alle von ihnen im Zustand NOT-CREATED, dann wie a), aber ohne Versionsangabe

* Es werden immer nur die lokalen Subsysteme angezeigt, deren Eigentümer die aufrufende Benutzertask ist.

Format

SHOW-SUBSYSTEM-STATUS

SUBSYSTEM-NAME = *ALL / *NON-PRIVILEGED-CLASS-5 / <structured-name 1..8>
,VERSION = *STD / *ALL / <product-version mandatory-man-corr> / <product-version without-man-corr>

Operandenbeschreibung

SUBSYSTEM-NAME =

Gibt den Namen des Subsystems an, über das Informationen gewünscht sind.

SUBSYSTEM-NAME = *ALL

Gibt Informationen über alle Subsysteme aus.

Die Angabe einer Versionsnummer im Operanden VERSION ist nicht zulässig.

SUBSYSTEM-NAME = *NON-PRIVILEGED-CLASS-5

Gibt Informationen über alle globalen Subsysteme aus, die nicht-privilegierte Klasse-5-Speicherseiten belegen (Subsysteme mit dem Attribut MEMORY-CLASS=*LOCAL-PRIVILEGED).

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

Name des Subsystems, über das Informationen ausgegeben werden sollen.

VERSION =

Vereinbart die Versionsnummer.

VERSION = *STD

Wird die Version nicht angegeben oder explizit *STD vereinbart, gilt für die Zuweisung folgende Reihenfolge:

1. Die Information wird über das Subsystem bereitgestellt, das vom Zustand NOT-CREATED abweicht.
2. Befinden sich mehrere Versionen in einem Zustand ungleich NOT-CREATED, werden im Ausgabertext alle diese Versionen berücksichtigt.

Sind alle Subsystemversionen im Zustand NOT-CREATED, steht im Ausgabertext keine Versionsangabe. Die Informationen werden sowohl für globale Subsystemversionen als auch für lokale Subsystemversionen - wenn ein lokaler Subsystemkatalog geladen ist - ausgegeben.

VERSION = *ALL

Über alle verfügbaren Versionen des betreffenden Subsystems sollen Informationen bereitgestellt werden.

Diese Angabe ist nicht erlaubt mit SUBSYSTEM-NAME = *NON-PRIVILEGED-CLASS-5.

VERSION = <product-version mandatory-man-corr> /**<product-version without-man-corr>**

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstatus müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

In Kombination mit den Angaben SUBSYSTEM-NAME=*ALL/*NON-PRIVILEGED-CLASS-5 ist dieser Operandenwert nicht zulässig.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	CMD2009	Systemfehler: beim Schreiben des S-Variablen-Bereichs
	32	ESM0360	Systemfehler: XVT oder TCB nicht erreichbar
	32	ESM0602	Systemfehler: Probleme mit Speicher-Verwaltung
	32	ESM0611	Systemfehler: Anschluss-Tabellen für diese Task existieren nicht
	32	ESM0671	Systemfehler: beim Schreiben nach SYSOUT
	64	ESM0600	Operandenfehler: Keine Version mit *ALL
	64	ESM0601	Angegebenes Subsystem nicht gefunden
	64	ESM0604	Operandenfehler: Keine Version mit *NON-PRIV-CLASS-5
	64	ESM0608	Subsystemversion nicht gefunden
	64	ESM0610	Kein Subsystem mit *NON-PRIV-CLASS-5
	64	OPS0002	Kommando unterbrochen
	130	OPS0001	Kommando nicht ausgeführt wegen fehlendem Speicherplatz: Kommando später wiederholen!

Ausgabe in S-Variablen

Tritt bei der Ausgabe nach SYSOUT ein Fehler auf (Meldung ESM0671), wird die Ausgabe in S-Variablen normal fortgesetzt, sofern sie angefordert wurde.

Tritt bei der Ausgabe in S-Variablen ein Fehler auf (Meldung ESM0672 mit Returncode CMD2009) wird die Ausgabe nach SYSOUT normal fortgesetzt.

Die hier gegebenen Informationen entsprechen im Wesentlichen denen bei der Ausgabe nach SYSOUT. Wenn eine Version angegeben wurde, beschreibt die S-Variable SUBSYS-TYPE zusätzlich, ob das betreffende Subsystem global oder lokal definiert ist. Sind alle Versionen eines Subsystems im Subsystemkatalog im Zustand NOT-CREATED, und wird die Ausgabe des Subsystems mit VERSION=*STD angefordert, werden nur der Subsystemname und der Subsystemstatus ausgegeben.

Ausgabe-Information	Name der S-Variablen	T	Inhalt	Bedingung
Anzahl der angeschlossenen Tasks seit Start des Subsystems	var(*LIST).CONN-NUM-SINCE-START	I	<integer 0..2 ³¹ -1>	
Information über den Benutzerspeicher für nicht-privilegierte Subsysteme *RESERVED: das Subsystem liegt im Klasse-5-Speicher *UNRESERVED: ein privilegierter Benutzer hat das Kommando RELEASE-SUBSYSTEM-SPACE aufgerufen	var(*LIST).SUBSYS-ADDR-SPACE	S	*RESERVED *UNRESERVED	
Interner Status, siehe Tabelle 12 auf Seite 147	var(*LIST).SUBSYS-INT-STA	S	INSTALLED INITIALIZED CONNECTABLE WAIT-CLS-CTRL WAIT-DISCON WAIT-DEINIT WAIT-STOP-COM NONE	
Name des Subsystems	var(*LIST).SUBSYS-NAME	S	<structured-name 1..8>	
Status des Subsystems	var(*LIST).SUBSYS-STA	S	*CREATED *IN-CREATE *IN-DELETE *IN-HOLD *IN-RESUME *LOCKED *NOT-CREATED *NOT-RESUMED	
Ist das betreffende Subsystem lokal oder global definiert? Wird nur ausgewertet, wenn der Operand VERSION belegt ist	var(*LIST).SUBSYS-TYPE	S	*GLB *LOC	
Version des Subsystems	var(*LIST).SUBSYS-VERSION	S	<product-version>	
TID der Tasks, die zurzeit an das Subsystem angeschlossen sind	var(*LIST).USED-TASK-LIST(*LIST).TID	S	<text 8>	
TSN der Tasks, die zurzeit an das Subsystem angeschlossen sind	var(*LIST).USED-TASK-LIST(*LIST).TSN	S	<text 4>	

Zur Arbeit mit S-Variablen, siehe Handbuch „Kommandos Band 6“ [20] (Konzept, Aufbau, Deklaration, Zuweisung, Zugriff und Weiterverwendung).

Ein Beispiel mit Ausgaben in S-Variablen ist ab [Seite 173](#) zu finden.

Der Inhalt der S-Variablen `var(*LIST).SUBSYS-INT-STA` ist abhängig von dem Status des Subsystems (S-Variable `var(*LIST).SUBSYS-STA`):

Status / <code>var(*LIST).SUBSYS-STA</code>	Sub-Status / <code>var(*LIST).SUBSYS-INT-STA</code>	Bedeutung
IN-CREATE / IN-RESUME	INSTALLED INITIALIZED CONNECTABLE	Das Subsystem ist geladen. Die 'Init'-Routine wurde ausgeführt. Das Subsystem ist bereit.
IN-DELETE / IN-HOLD	WAIT-CLS-CTRL WAIT-DISCON WAIT-STOP-COM WAIT-DEINIT INSTALLED	Die 'Close Control'-Routine wurde aufgerufen. DSSM wartet auf ihre Beendigung. DSSM wartet auf die letzte Trennung (erzwungenes Löschen möglich). Keine Task ist mehr verbunden, und DSSM wartet auf das Ende der 'Stop Commission'-Routine. Keine Task ist mehr verbunden, die 'Deinit'-Routine wurde aufgerufen und DSSM wartet auf ihre Beendigung. Das Subsystem ist geladen aber nicht initialisiert.
NOT-CREATED / NOT-RESUMED / LOCKED / CREATED	NONE	Das Subsystem befindet sich in dem angezeigten Status. Da dies kein Übergangszustand ist, existieren keine weiteren Informationen.

Tabelle 12: Inhalt von `var(*LIST).SUBSYS-INT-STA` in Abhängigkeit von `var(*LIST).SUBSYS-STA`

Hinweise

- Die Kombination der Operanden `SUBSYSTEM-NAME=*ALL` und `VERSION=*ALL` liefert den gleichen Umfang an Informationen wie `SUBSYSTEM-NAME=*ALL`.
- Wird im lokalen Subsystemkatalog ein Subsystem gefunden, wird in der Ausgabe `LOCAL` vor die restlichen Informationen geschrieben.
- Für Subsysteme, die mit den Attributen `CONNECTION-SCOPE=*FREE` und `MEMORY-CLASS=*SYSTEM-GLOBAL` deklariert sind, liefert `SHOW-SUBSYSTEM-STATUS` keine Information.
Subsysteme mit den Attributen `CONNECTION-SCOPE=*FREE` und `MEMORY-CLASS=*LOCAL-PRIVILEGED` bzw. `*LOCAL-UNPRIVILEGED` werden hingegen in der Ausgabe berücksichtigt.

- Sind verschiedene Versionen eines Subsystems geladen (durch erneutes Laden im Coexistence- oder Exchange-Modus), erhält der privilegierte Anwender mit dem Kommando SHOW-SUBSYSTEM-STATUS ohne Versionsangabe (Operand VERSION=*STD / *ALL) Informationen über den Zustand aller Versionen des Subsystems, die sich in einem Zustand ungleich NOT-CREATED befinden.
- Tasks, die an ein globales Subsystem mit CONNECTION-SCOPE=*OPTIMAL angeschlossen sind, werden zwar in der Summe aller angeschlossenen Tasks mitgezählt, aber nicht explizit mit ihrer TID und TSN aufgeführt.
- Tritt bei der Ausgabe nach SYSOUT ein Fehler auf (Meldung ESM0671), wird die Ausgabe in S-Variablen normal fortgesetzt, sofern sie angefordert wurde. Tritt bei der Ausgabe in S-Variablen ein Fehler auf (Meldung ESM0672 oder CMD2009) wird die Ausgabe nach SYSOUT normal fortgesetzt.
- Bei einer Unterbrechung der Kommandoabarbeitung durch das Betätigen der [K2]-Taste sowohl die Ausgabe nach SYSOUT als auch die Ausgabe in S-Variablen unterbrochen. Es wird der Returncode OPS0002 gesetzt.
- Wird bei einer Eingabeaufforderung (PLEASE ACKNOWLEDGE) die Unterbrechungstaste [K2] betätigt, wird die Ausgabe nach SYSOUT sofort abgebrochen. Die Ausgabe in S-Variablen wird normal fortgesetzt, sofern sie angefordert wurde. Es wird keine Meldung ausgegeben.
- Die volle Übereinstimmung der Ausgaben nach SYSOUT und in S-Variablen kann nur dann garantiert werden, wenn die das Kommando absetzende Prozedur mit dem Operanden INTERRUPTION-ALLOWED=*NO eine Unterbrechung der Ausgabe verboten hat.
- Will der privilegierte Benutzer Informationen über ein spezielles Subsystem (SUBSYSTEM-NAME=<structured-name 1...8>) und die Ausgabe in S-Variablen, kann das Feld USED-TASK-LIST leer sein, obwohl es generiert wurde. Mit der Builtin-Funktion SIZE() kann dann die Anzahl der Listenelemente bestimmt werden.
- Befindet sich das Subsystem in einem Übergangszustand, werden weitere Informationen über den aktuellen Verarbeitungszustand (Sub-Status) ausgegeben, siehe [Tabelle 12 auf Seite 147](#).

Beispiele

Beispiel 1

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*NON-PRIVILEGED-CLASS-5 _____ (1)
%SUBSYSTEM VAS-TU /V02.1 IS NOT CREATED IN CL5
% : : : (alle weiteren Subsysteme in CL5)
```

- (1) SYSOUT-Ausgabe von Informationen über alle Subsysteme, die nicht-privilegierten Klasse-5-Speicher belegen. Der nicht-privilegierte Aufrufer erhält dieselben Informationen wie der privilegierte Anwender.

Beispiel 2

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=EDT _____ (1)
%SUBSYSTEM EDT /V16.6 IS USED BY 8 TASKS _____ (2)
% TASKID 00070011 00020026 0002002A 00010038 0001003A 0001003B
% TSN 01QB 01LX XAAJ XAAD 01LH 01LG
% TASKID 0001003E 0001003F
% TSN 01LF 01LE
% 38 CONNECTIONS SINCE STARTUP
%SUBSYSTEM EDT /V16.5 IS USED BY 2 TASKS _____ (3)
% TASKID 0001004F 00010051
% TSN 01LQ 01LR
% IS IN DELETE / WAIT-DISCON
% 252 CONNECTIONS SINCE STARTUP
```

- (1) SYSOUT-Ausgabe von Informationen über das Subsystem EDT ohne Angabe einer Version. Die dargestellte Ausgabe erhält nur der privilegierte Aufrufer. Der nicht-privilegierte Aufrufer würde nur folgende Ausgabezeile erhalten:

```
%SUBSYSTEM EDT /V16.6 IS CREATED
%SUBSYSTEM EDT /V16.5 IS IN DELETE / WAIT-DISCON
```

- (2) Die Ausgabe zeigt, dass zwei Subsysteme EDT (V16.6 und V16.5) aktiv sind. Das Subsystem EDT V16.6 wurde im Exchange-Modus gestartet und soll das Subsystem EDT V16.5 ersetzen.
- (3) Das Subsystem EDT V16.5 befindet sich im Zustand IN-DELETE, da noch zwei Tasks EDT V16.5 geladen haben.



Ein Beispiel für die Ausgabe in S-Variablen ist im [Abschnitt „Ausgabe in eine S-Variable“ auf Seite 173](#) beschrieben.

START-LOCAL-SUBSYSTEM

Lokales Subsystem im Benutzeradressraum aktivieren

Anwendungsbereich: SYSTEM-MANAGEMENT

Privilegierung: STD-PROCESSING

Funktionsbeschreibung

Mit dem Kommando START-LOCAL-SUBSYSTEM kann ein lokales Subsystem aktiviert werden, indem sein Code in tasklokalen, nicht-privilegierten Adressraum (Benutzeradressraum) geladen wird.

Sind zu diesem Subsystem Syntax- und Meldungsdatei definiert, werden sie ebenfalls aktiviert.

Das Kommando wird abgebrochen und eine Meldung ausgegeben, wenn

- das angegebene Subsystem bereits aktiviert ist
- das angegebene Subsystem nicht im lokalen Subsystemkatalog enthalten ist
- die zugehörige Syntaxdatei oder Meldungsdatei nicht aktiviert werden kann
- der Subsystemcode nicht geladen werden kann

Ein lokales Subsystem kann nicht aktiviert werden, solange ein Programm geladen ist.



Lokale Subsysteme werden immer oberhalb der 16 MByte-Grenze geladen. Da lokale Subsysteme in den Benutzeradressraum geladen werden (nicht-privilegierter Klasse-5-Speicher), kann es beim (nahezu) gleichzeitigen Start mehrerer lokaler Subsysteme zu Speicherbereichsättigung kommen und infolgedessen zum Absturz der Task!

Format

START-LOCAL-SUBSYSTEM

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

,VERSION = ***STD** / <product-version mandatory-man-corr> / <product-version without-man-corr> / ***HIGHEST**

Operandenbeschreibung

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

Name des lokalen Subsystems, das aktiviert wird.

VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstatus müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = *STD

Existieren für das angegebene Subsystem mehrere Versionen und wird keine Version oder explizit *STD angegeben, wird das Subsystem mit der niedrigsten im lokalen Subsystemkatalog vorhandenen Versionsnummer ausgewählt.

VERSION = *HIGHEST

Es wird die höchste Version des Subsystems ausgewählt.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando ausgeführt, Subsystem erfolgreich gestartet oder Kommando ausgeführt, mit Warnung
1	0	CMD0001	Keine Aktion notwendig; das Subsystem ist bereits aktiv
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	64	ESM0255	Kommando nicht ausgeführt (ein Programm ist geladen, das angegebene Subsystem existiert nicht im lokalen Subsystemkatalog oder kein lokaler Subsystemkatalog geladen)
	64	ESM0326	Kommando abnormal beendet

Beispiel

```
/LOAD-LOCAL-SUBSYSTEM-CATALOG CATALOG-NAME=MY.LOCAL _____ (1)
%ESM0254 COMMAND 'LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/START-LOCAL-SUBSYSTEM SUBSYSTEM-NAME=LOC-SUBS,VERSION='01.0'
%ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'LOC-SUBS/01.0' COMPLETELY PROCESSED

/START-LOCAL-SUBSYSTEM SUBSYSTEM-NAME=NO-NAME _____ (2)
%ESM0201 SUBSYSTEM 'NO-NAME' NOT FOUND
%ESM0255 'START-LOCAL-SUBSYSTEM' COMMAND NOT PROCESSED
```

- (1) Zuerst wird ein lokaler Subsystemkatalog geladen, danach ein lokales Subsystem aktiviert. Beide Kommandos wurden erfolgreich ausgeführt.
- (2) Das zweite lokale Subsystem kann nicht aktiviert werden, da es im aktuellen lokalen Subsystemkatalog nicht bekannt ist.

START-SUBSYSTEM

Subsystem aktivieren

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	OPERATING SUBSYSTEM-MANAGEMENT
Berechtigungsschlüssel:	R

Funktionsbeschreibung

Mit diesem Kommando kann die Systembetreuung ein beliebiges Subsystem aktivieren. Für die Aktivierung des Subsystems werden folgende Informationen aus dem dynamischen Subsystemkatalog benutzt:

- Angaben zum Laden und Binden des Subsystems
- Angaben zur Initialisierung/Deinitialisierung und zum Beenden der Auftragsbeziehungen
- Angaben zu Aufrufstellungen, Nebenkomponenten und betrieblichen Abhängigkeiten (siehe entsprechende SSCM-Anweisungen im [Kapitel „SSCM“ auf Seite 179](#))

Das Kommando wird abgewiesen, wenn:

- das Subsystem im dynamischen Subsystemkatalog nicht gefunden wird
- eine andere Version des Subsystems bereits existiert und die Koexistenz nicht zugelassen ist (siehe Operand VERSION-PARALLELISM auf [Seite 156](#))
- Subsysteme nicht geladen sind, von denen das zu aktivierende Subsystem abhängt
- eine benötigte Datei fehlt (z.B. Meldungsdatei, Bibliothek)

Eine entsprechende Meldung informiert die Systembetreuung über die Annahme/Zurückweisung des Kommandos. Über den Operanden RESET=*YES kann auch für solche Subsysteme, die sich im Abbau befinden, die erneute Initialisierung des Subsystems erzwungen werden. Es können beliebig viele START-SUBSYSTEM-Kommandos in verschiedenen Tasks unter der privilegierten Benutzerkennung des Subsystemverwalters abgesetzt werden, es sei denn, die vereinbarten Parameter bei der Subsystemdefinition lassen dies nicht zu.

Abhängig von der Subsystemdefinition (SSCM-Anweisung SET- oder MODIFY-SUBSYSTEM-ATTRIBUTES, Operand SUBSYSTEM-LOAD-MODE) kann das Subsystem auf drei verschiedene Arten aktiviert werden:

- SUBSYSTEM-LOAD-MODE=*STD
Das BLS wird im STD-Run-Mode aufgerufen und lädt das Subsystem als Objektmodul.
- SUBSYSTEM-LOAD-MODE=*ADVANCED
Das BLS wird im ADVANCED-Run-Mode aufgerufen und lädt das Subsystem als Bindelademodul (LLM).
- SUBSYSTEM-LOAD-MODE=*ANY
Das BLS wird im STD-Run-Mode aufgerufen und lädt das Subsystem als Objektmodul. Tritt während des Ladens ein Fehler ein, wird BLS ein zweites Mal aufgerufen. Der Aufruf erfolgt im ADVANCED-Run-Mode und das Subsystem wird als Bindelademodul (LLM) geladen. Ist der erste Aufruf des BLS nicht erfolgreich, wird an der Konsole eine Fehlermeldung des BLS ausgegeben.

Format

START-SUBSYSTEM

```

SUBSYSTEM-NAME = <structured-name 1..8>
, VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST
, SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>
, RESET = *NO / *YES
, SYNCHRONOUS = *NO / *YES
, VERSION-PARALLELISM = *NONE / *EXCHANGE-MODE(...) / *COEXISTENCE-MODE
    *EXCHANGE-MODE(...)
    |   SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>
, MONJV = *NONE / <filename 1..54 without-gen-vers>

```

Operandenbeschreibung

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

Name des Subsystems, das aktiviert wird.

VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = *STD

Existieren für das angegebene Subsystem mehrere Versionen und wird keine Version oder explizit *STD angegeben, wird das Subsystem geladen, das mit dem Start-Attribut CREATION-TIME=*AT-SUBSYSTEM-CALL (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#)) deklariert wurde. Trifft diese Bedingung nicht zu, wird die niedrigste Versionsnummer ausgewählt, die im statischen Subsystemkatalog für dieses Subsystem angelegt wurde.

Ausnahme

Soll eine Version eines Subsystems automatisch beim ersten SVC-Aufruf aktiviert werden, so gilt diese Version als Standardversion.

VERSION = *HIGHEST

Es wird die höchste Version des Subsystems ausgewählt.

SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>

Vereinbart, ob spezielle Parameter verarbeitet werden, die nur das angegebene Subsystem auswerten kann.

RESET =

Beeinflusst Verhalten und Dringlichkeit der Kommandobearbeitung.

RESET = *NO

Befindet sich das betreffende Subsystem im Abbau, wird das Kommando solange abgewiesen, bis dieser blockierende Prozess beendet ist.

RESET = *YES

Das Kommando wird ohne Rücksicht auf einen evtl. noch ausstehenden Abbau-Prozess akzeptiert und das Subsystem oder einige Komponenten initialisiert (siehe auch Hinweise auf [Seite 157](#)). Der Versionsparameter ist für diesen Operanden verpflichtend.

SYNCHRONOUS =

Erlaubt die Wahl zwischen synchroner und asynchroner Verarbeitung.

SYNCHRONOUS = *NO

Das Kommando soll asynchron verarbeitet werden, d.h ohne mit einer weiteren Eingabe auf dessen Ausführung warten zu müssen. Nach der Syntaxprüfung des Kommandos erhält die aufrufende Task die Meldung ESM0216. Fehlermeldungen über den Ablauf des Kommandos werden an der Konsole ausgegeben.

SYNCHRONOUS = *YES

Die Ausführung des Kommandos muss abgewartet werden.

Entsprechende Fehlermeldungen über den Ablauf werden an die Task ausgegeben.

Im Fall eines Versionsaustauschs ist diese Angabe nur für die neu zu aktivierende Version relevant. Die Deaktivierung der anderen, „alten“ Version geschieht immer asynchron.

VERSION-PARALLELISM =

Vereinbart, ob verschiedene Versionen des gleichen Subsystems gleichzeitig aktiv sein dürfen.

VERSION-PARALLELISM = *NONE

Eine Koexistenz verschiedener Versionen eines Subsystems - unabhängig von den Vorgaben bei der Definition - soll nicht zulässig sein. Ist der Status einer Version ungleich „NOT-CREATED“, wird die Aktivierung zurückgewiesen.

VERSION-PARALLELISM = *EXCHANGE-MODE(...)

Eine temporäre Koexistenz zweier Versionen eines Subsystems soll zulässig sein. Die Aktivierung ist nur erlaubt, wenn sich keine oder nur eine Version des Subsystems im Zustand CREATED befindet. Haben bereits zwei Versionen diesen Zustand eingenommen, wird für die zuletzt gestartete Version eine implizite Deaktivierung eingeleitet.

Ist eine Version eines Subsystems gesperrt (Zustand LOCKED), wird diese von DSSM als NOT-CREATED behandelt.

In folgenden Fällen wird das Kommando mit diesem Operanden zurückgewiesen:

- die zu ersetzende Version ist mit HOLD=*NO, aber ohne CLOSE-CTRL-Routine definiert
- das Kommando MODIFY-SUBSYSTEM-PARAMETER CHANGE-STATE=*NO wurde für die zu ersetzende Version verwendet
- es wurde gleichzeitig der Operand RESET=*NO angegeben
- der Status der Version ist ungleich CREATED, NOT-CREATED oder LOCKED

SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254>

Vereinbart, ob spezielle Parameter verarbeitet werden, die nur die angegebene Version des Subsystems auswerten kann.

VERSION-PARALLELISM = *COEXISTENCE-MODE

Eine uneingeschränkte Koexistenz zweier oder mehrerer Versionen des gleichen Subsystems soll zulässig sein. Voraussetzung ist hierbei, dass dies für alle beteiligten Versionen in der SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES zugelassen wurde, siehe [Seite 243](#).

MONJV = *NONE / <filename 1..54 without-gen-vers>

Vereinbart den Namen einer Monitor-Jobvariablen. Die Monitor-Jobvariable zeigt an, ob das Subsystem aktiv, angehalten, unterbrochen oder gesperrt ist. Die Inhalte der Monitor-Jobvariablen sind auf [Seite 52](#) aufgelistet.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	CMD0001	Keine Aktion notwendig; das Subsystem ist bereits aktiv
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0224	Kommando wird nicht verarbeitet
	32	ESM0228	Kommando abnormal beendet

Hinweise

- Subsysteme weisen in der Regel vielfältige Abhängigkeits- und Ladebeziehungen zu anderen Subsystemen auf. Um die Leistungen des einzelnen Subsystems zu gewährleisten, müssen diese Beziehungen berücksichtigt werden. DSSM versucht, mögliche Konflikte zu vermeiden, die sich aus Anforderungen des Anwenders ergeben könnten, und weist daher entsprechende Kommandos zurück. Aktionen, wie die Installation fehlender Subsysteme oder das Entladen abhängiger Subsysteme, werden nicht durchgeführt. Generiert der Anwender allerdings mit der Anweisung CHECK-REFERENCE=*NO auch komplexe Subsysteme (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#)), führt DSSM die geforderten Funktionen **trotz** möglicher Konflikte durch: Das Kommando START-SUBSYSTEM lädt das angegebene Subsystem, auch wenn ein Subsystem, zu dem definierte Beziehungen bestehen, noch nicht vollständig geladen ist.
- Um ein hohes Maß an Parallelität und Datenintegrität zu gewährleisten, werden zeitaufwändige Verwaltungsaufgaben nicht unter der Kontrolle der aufrufenden Task ausgeübt, sondern einer DSSM-Task übertragen. In der Regel wird nur die Prüfung der geforderten Funktion **synchron** (d.h. verbunden mit einem Wartezustand für die aufrufende Task) realisiert. Die eigentliche Verarbeitung jedoch führt DSSM **asynchron** und unabhängig von der aufrufenden Task durch.

- Nach dem Kommando STOP-SUBSYSTEM wird START-SUBSYSTEM abgewiesen, wenn DSSM die Aktion „Subsystem entladen“ noch nicht vollständig durchführen konnte. Mit dem Operanden RESET=*YES kann die Systembetreuung jedoch das unbedingte Laden des Subsystems erreichen; die vollständige Abarbeitung eines STOP-SUBSYSTEM-Kommandos muss nicht abgewartet werden.
In diesem Fall wird die Initialisierungsroutine angestoßen und das betreffende Subsystem, das über den RESET informiert wird, kann den Umfang der INIT-Routine selbst festlegen (vollständige Initialisierung, Teil-Initialisierung oder keine Initialisierung).

Ausnahme

Befindet sich das betreffende Subsystem noch im Zustand IN-DELETE, wurde aber bereits deinitialisiert, wird die Verarbeitung „Subsystem entladen“ trotz RESET=*YES nicht unterbrochen. Das Kommando START-SUBSYSTEM wird zurückgewiesen, bis das Subsystem den Zustand NOT-CREATED erreicht und alle Betriebsmittel freigegeben hat.

- Sollen zwei Versionen eines Subsystems ausgetauscht werden, ist bei Verwendung des Parameters RESET=*YES auf Folgendes zu achten:
 - befindet sich Version A im Zustand IN-DELETE und Version B im Zustand CREATED, ist RESET=*YES für A nur zulässig, wenn bei der SSCM-Definition für beide Versionen Koexistenz zugelassen wurde (siehe [Seite 47](#)).
 - befinden sich beide Versionen im Zustand IN-DELETE, ist RESET=*YES für eine dieser Versionen zulässig, wenn diese definiert wurde mit RESET=*ALLOWED, VERSION-EXCHANGE=*ALLOWED.
- Ein Wiederanlauf (d.h. Aufruf der INIT-Routine für mit RESTART-REQUIRED=*YES definierte Subsysteme) wird unterbunden, da dies ansonsten zu unerlaubten Koexistenzen führen kann.

STOP-LOCAL-SUBSYSTEM

Lokales Subsystem im Benutzeradressraum deaktivieren

Anwendungsbereich: SYSTEM-MANAGEMENT

Privilegierung: STD-PROCESSING

Funktionsbeschreibung

Mit dem Kommando STOP-LOCAL-SUBSYSTEM kann ein lokales Subsystem deaktiviert werden. Sind zu diesem Subsystem Syntax- und Meldungsdatei definiert, werden sie ebenfalls deaktiviert. Der Subsystemcode wird entladen.

Ein lokales Subsystem kann nicht deaktiviert werden, wenn es nicht vorher mit dem Kommando START-LOCAL-SUBSYSTEM aktiviert wurde oder solange ein Programm geladen ist. In diesen Fällen wird eine Meldung ausgegeben.

Die Deaktivierung des angegebenen lokalen Subsystems wird auch dann abgeschlossen, wenn die zugehörige Syntax- und/oder Meldungsdatei nicht fehlerfrei deaktiviert werden konnten.

Format

STOP-LOCAL-SUBSYSTEM

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

,VERSION = ***STD** / <product-version mandatory-man-corr> / <product-version without-man-corr> / ***HIGHEST**

Operandenbeschreibung

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

Name des lokalen Subsystems, das deaktiviert wird.

VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstatus müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = *STD

Existieren für das angegebene Subsystem mehrere Versionen wird das derzeit aktive Subsystem ausgewählt.

VERSION = *HIGHEST

Es wird die höchste Version des Subsystems ausgewählt.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	CMD0001	Keine Aktion notwendig; das Subsystem ist bereits aktiv
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0255	Kommando nicht ausgeführt (ein Programm ist geladen, das angegebene Subsystem existiert nicht im lokalen Subsystemkatalog oder kein lokaler Subsystemkatalog geladen)
	32	ESM0326	Kommando abnormal beendet

Beispiel

```

/load-local-subsystem-catalog catalog-name=local.dssmcat.1 _____ (1)
%ESM0254 COMMAND '/LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/load-local-subsystem-catalog catalog-name=local.dssmcat.11 _____ (1)
%ESM0254 COMMAND 'LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/start-local-subsystem subsystem-name=loc-subs,version='01.0'
%ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'LOC-SUBS/01.0' COMPLETELY PROCESSED
/stop-local-subsystem subsystem-name=loc-subs,version='01.0' _____ (2)
%ESM0220 FUNCTION 'DELETE' FOR SUBSYSTEM 'LOC-SUBS/01.0' COMPLETELY PROCESSED
/unload-local-subsystem-catalog
%ESM0254 COMMAND 'UNLOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/stop-local-subsystem subsystem-name=loc-subs,version='01.0' _____ (3)
%ESM0341 NO LOCAL SUBSYSTEM CATALOG LOADED
%ESM0255 'STOP-LOCAL-SUBSYSTEM' COMMAND NOT PROCESSED

```

- (1) Zuerst wird ein lokaler Subsystemkatalog geladen, danach ein lokales Subsystem aktiviert. Beide Kommandos werden erfolgreich ausgeführt.
- (2) Das lokale Subsystem wird deaktiviert. Danach wird der aktuelle lokale Subsystemkatalog entladen. Beide Kommandos werden erfolgreich ausgeführt
- (3) Das lokale Subsystem kann nicht (zum zweiten Mal) deaktiviert werden, da es keinen aktuellen lokalen Subsystemkatalog gibt.

STOP-SUBSYSTEM

Subsystem deaktivieren

Anwendungsbereich:	SYSTEM-MANAGEMENT
Privilegierung:	OPERATING SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Mit diesem Kommando kann die Systembetreuung ein beliebiges Subsystem deaktivieren. Ablauf und Funktion des Kommandos stellen sich wie folgt dar:

1. Das betreffende Subsystem wird für alle neuen Aufrufer gesperrt.
2. Das Subsystem wird deaktiviert, sobald alle darauf zugreifenden Aufträge normal beendet sind. Wurde im Kommando STOP-SUBSYSTEM der Operand FORCED=*YES angegeben, werden die auf das Subsystem zugreifenden Aufträge unverzüglich abgebrochen (Operand FORCED=*YES wird allerdings nur akzeptiert, wenn vorher das Kommando STOP-SUBSYSTEM mit Operand FORCED=*NO die Beendigung der Aufträge nicht erreichen konnte).
3. Das Subsystem wird entladen.
4. Alle belegten Betriebsmittel werden freigegeben.

Das Kommando STOP-SUBSYSTEM wird abgewiesen, wenn:

- das Subsystem im dynamischen Subsystemkatalog nicht gefunden wird
- bereits aktivierte bzw. im Aufbau befindliche Subsysteme vom zu deaktivierenden Subsystem abhängig sind
- bereits aktivierte bzw. im Aufbau befindliche Subsysteme Binderbeziehungen zu dem zu deaktivierenden Subsystem haben
- Operand FORCED=*YES angegeben wurde, ohne dass vorher versucht wurde, die Beendigung der Aufträge mit Operand FORCED=*NO zu erreichen

Format

STOP-SUBSYSTEM
SUBSYSTEM-NAME = <structured-name 1..8> , VERSION = *STD / <product-version mandatory-man-corr> / <product-version without-man-corr> / *HIGHEST , SUBSYSTEM-PARAMETER = *NONE / <c-string 1..254> , FORCED = *NO / *YES , SYNCHRONOUS = *NO / *YES

Operandenbeschreibung

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

Name des Subsystems, das deaktiviert wird.

VERSION = ***STD** / <product-version mandatory-man-corr> / <product-version without-man-corr> / ***HIGHEST**

Vereinbart die Versionsnummer.

Bei Angabe einer Versionsnummer muss das hier angegebene Format mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturstatus müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

VERSION = ***STD**

Existiert nur **eine** Version des Subsystems, die geladen ist, wird diese Version ausgewählt. Existieren **mehrere** passende Versionen, muss die Version spezifiziert werden.

VERSION = ***HIGHEST**

Es wird die höchste Version des Subsystems ausgewählt.

SUBSYSTEM-PARAMETER = ***NONE** / <c-string 1..254>

Vereinbart, ob spezielle Parameter verarbeitet werden, die nur das angegebene Subsystem auswerten kann.

FORCED =

Bestimmt Verhalten und Dringlichkeit der Kommandobearbeitung.

FORCED = *NO

Die Verarbeitung wird abgewartet und damit die normale Beendigung aller Tasks, die auf dieses Subsystem zugreifen.

FORCED = *YES

Der unverzügliche Abbruch aller belegenden Prozesse wird veranlasst.

Dies kann im Falle eines privilegierten Subsystems zu einem Systemdump führen. Tasks, die an ein nicht-privilegiertes Subsystem angeschlossen sind, ist der Ausgang über die von DSSM angebotene STXIT-Fehlerbehandlung möglich.

SYNCHRONOUS =

Erlaubt die Wahl zwischen synchroner und asynchroner Verarbeitung.

SYNCHRONOUS = *NO

Das Kommando soll asynchron verarbeitet werden, d.h ohne mit einer weiteren Eingabe auf dessen Ausführung warten zu müssen. Nach der Syntaxprüfung des Kommandos erhält die aufrufende Task die Meldung ESM0216. Fehlermeldungen über den Ablauf des Kommandos werden an der Konsole ausgegeben.

SYNCHRONOUS = *YES

Die Ausführung des Kommandos muss abgewartet werden.

Entsprechende Fehlermeldungen über den Ablauf werden an die Task ausgegeben.

Im Fall eines Versionsaustauschs ist diese Angabe nur für die neu zu aktivierende Version relevant. Die Deaktivierung der anderen, „alten“ Version geschieht immer asynchron.

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	CMD0001	Keine Aktion notwendig; das Subsystem ist nicht mehr aktiv
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0224	Kommando wird nicht verarbeitet
	32	ESM0228	Kommando abnormal beendet

Hinweise

- Um ein hohes Maß an Parallelität und Datenintegrität zu gewährleisten, werden zeitaufwändige Verwaltungsaufgaben nicht unter der Kontrolle der aufrufenden Task ausgeübt, sondern einer DSSM-Task übertragen.
In der Regel wird nur die Prüfung der geforderten Funktion **synchron** realisiert, d.h. verbunden mit einem Wartezustand für die aufrufende Task. Die eigentliche Verarbeitung jedoch führt DSSM **asynchron** und unabhängig von der aufrufenden Task durch.
- STOP-SUBSYSTEM mit dem Operanden FORCED=*YES wird nur akzeptiert, wenn zuvor das Kommando mit FORCED=*NO gegeben wurde und das Subsystem nur noch auf den Abbau der belegenden Task wartet.
Die Funktion FORCED kann nicht das „normale“ Verhalten von Tasks garantieren, die an ein privilegiertes Subsystem angeschlossen waren.
Tasks mit Anschluss an ein nicht-privilegiertes Subsystem können eine Fehleroutine ablaufen lassen, die ihnen die Fortsetzung des Programms ermöglicht.
Die Funktion FORCED wird durch Ablauf einer Contingency-Routine für jede an das Subsystem angeschlossene Task realisiert. Der Abbau der Task ist dann abgeschlossen, wenn diese die Contingency-Routine durchläuft.
Da DSSM den Ablauf dieser Routine aber nicht abwartet, können nach einem zwischenzeitlichen START-SUBSYSTEM diese Tasks als noch mit dem Subsystem verbunden registriert werden.

UNLOAD-LOCAL-SUBSYSTEM-CATALOG

Lokalen Subsystemkatalog entladen

Anwendungsbereich: SYSTEM-MANAGEMENT

Privilegierung: STD-PROCESSING

Funktionsbeschreibung

Mit dem Kommando UNLOAD-LOCAL-SUBSYSTEM-CATALOG kann der Aufrufer einen lokalen Subsystemkatalog aus seinem taskeigenen Benutzeradressraum (Klasse-5-Speicher) entladen.

Sind zu diesem Zeitpunkt noch Subsysteme aktiv, werden sie automatisch deaktiviert und die Ausführung des Kommandos UNLOAD-LOCAL-SUBSYSTEM-CATALOG fortgesetzt.

Das Kommando wird mit einer Fehlermeldung abgewiesen, wenn nicht zuvor ein lokaler Subsystemkatalog mit dem Kommando LOAD-LOCAL-SUBSYSTEM-CATALOG in den Benutzeradressraum geladen wurde.

Ein lokaler Subsystemkatalog kann nicht entladen werden, wenn gleichzeitig ein Programm geladen ist.

Format

UNLOAD-LOCAL-SUBSYSTEM-CATALOG

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Kommando erfolgreich ausgeführt
	32	ESM0255	Kommando nicht ausgeführt; es ist kein lokaler Katalog geladen oder es ist ein Programm geladen
	32	ESM0326	Kommandobearbeitung abgebrochen (Systemfehler)

Beispiel

```
/load-local-subsystem-catalog catalog-name=local.dssmcat.1 _____ (1)
%ESM0254 COMMAND 'LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/unload-local-subsystem-catalog _____ (2)
%ESM0254 COMMAND 'UNLOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
/unload-local-subsystem-catalog _____ (3)
%ESM0341 NO LOCAL SUBSYSTEM CATALOG LOADED
%ESM0255 'UNLOAD-LOCAL-SUBSYSTEM-CATALOG' COMMAND NOT PROCESSED
```

- (1) Zuerst wird ein lokaler Subsystemkatalog geladen. Die Ausführung des Kommandos ist erfolgreich.
- (2) Das Entladen des lokalen Subsystemkatalogs ist ebenfalls erfolgreich.
- (3) Der zweite Versuch, den lokalen Subsystemkatalog zu entladen, wird mit der Meldung ESM0341 abgewiesen: Es kann nur ein vorher geladener lokaler Subsystemkatalog entladen werden.

UNLOCK-SUBSYSTEM

Subsystem im Zustand LOCKED in den Zustand NOT-CREATED überführen (Subsystem entsperren)

Anwendungsbereich: SYSTEM-MANAGEMENT
Privilegierung: SUBSYSTEM-MANAGEMENT

Funktionsbeschreibung

Mit diesem Kommando kann die Systembetreuung ein gesperrtes Subsystem (Subsystem im Zustand LOCKED) wieder in einen deklarierten, aber nicht aktivierten Zustand (Zustand NOT-CREATED) überführen. Das Subsystem wird also in der laufenden Session entsperrt. UNLOCK-SUBSYSTEM unterstützt damit den unterbrechungsfreien Betrieb des BS2000/OSD.

Ein Subsystem kann durch seine INIT-/DEINIT-/STOPCOM- oder CLOSE-CTRL-Routinen in den Zustand LOCKED überführt werden. Diese Routinen fordern entweder selber die Sperrung des Subsystems bei DSSM an oder sie veranlassen einen Speicherabzug und den Wiederanlauf der Holdertask, der dann nicht fehlerfrei ausgeführt werden kann (RESTART-REQUIRED=*NO oder - bei *YES - Überschreitung der max. erlaubten Versuche). Tritt das Sperren eines Subsystems während der Aktivierungsphase (INIT-Routine) auf, wäre das Subsystem ohne die Möglichkeit eines UNLOCK-SUBSYSTEM-Kommandos bis zum Shutdown und dem folgenden Restart nicht verfügbar.

Es muss beachtet werden, dass sich nicht alle Subsysteme problemlos entsperren lassen bzw. dass das erneute Starten nicht in allen Fällen realisierbar ist. Siehe dazu die Hinweise auf der nächsten Seite.

Format

UNLOCK-SUBSYSTEM

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

,**VERSION** = <product-version mandatory-man-corr> / <product-version without-man-corr>

Operandenbeschreibung

SUBSYSTEM-NAME = <structured-name 1..8>
 Name des Subsystems, das entsperrt werden soll.

**VERSION = <product-version mandatory-man-corr> /
<product-version without-man-corr>**

Versionsnummer des oben genannten Subsystems.

Das hier angegebene Format muss mit dem bei der Definition des Subsystems benutzten Format übereinstimmen. Freigabe- und Korrekturzustand müssen angegeben werden oder dürfen nicht angegeben werden (siehe SSCM-Anweisung SET-SUBSYSTEM-ATTRIBUTES auf [Seite 243](#) und SDF-Datentyp „product-version“ auf [Seite 13](#)).

Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	ESM0414	Syntaxfehler: es wurde eine ungültige Version angegeben
	32	ESM0228	Kommando abnormal beendet
	64	ESM0224	Kommando wird nicht verarbeitet

Hinweise

Das zu entsperrende Subsystem muss sich im Zustand LOCKED befinden.

Für die Anwendung des Kommandos UNLOCK-SUBSYSTEM gibt es Einschränkungen:

- Einige Subsysteme können nicht regulär in einer Session beendet und erneut gestartet werden. Für das zu entsperrende Subsystem muss die Ausführung der Kommandos STOP-SUBSYSTEM und HOLD-SUBSYSTEM erlaubt sein, d.h. es darf nicht mit der Eigenschaft SUBSYSTEM-HOLD=*FORBIDDEN definiert sein (siehe auch Einschränkungen des Kommandos STOP-SUBSYSTEM auf [Seite 162](#)).
- Auf Grund von Abhängigkeiten zu anderen Subsystemen kann das Kommando UNLOCK-SUBSYSTEM zu Inkonsistenzen im Subsystemkatalog führen. Um eine solche Situation zu vermeiden, müssen zuerst alle zu diesem Subsystem in Beziehung stehenden Subsysteme mit /STOP-SUBSYSTEM deaktiviert werden. Anschließend müssen sie mit /START-SUBSYSTEM zusammen mit dem entsperrten Subsystem erneut gestartet werden.
- Auch wenn das Subsystem aus jedem Zustand erneut gestartet werden kann und wenn alle eventuell bestehenden Abhängigkeiten zu anderen Subsystemen berücksichtigt wurden (siehe oben), ist nach Ausführung des Kommandos UNLOCK-SUBSYSTEM nicht gewährleistet, dass das Subsystem nach dem nächsten START-SUBSYSTEM-Kommando aktiv ist. Diese Einschränkung besteht insbesondere dann, wenn Probleme oder Situationen auftreten, die das betreffende Subsystem selbst verursacht.

3.9 Beispiele

3.9.1 Nutzung von lokalen Subsystemen

```

/START-SSCM _____ (1)
% BLS0517 MODULE 'SSCM' LOADED
% SCM5000 PROGRAM SSCM VERSION V02.3B10 STARTED

//START-CATALOG-CREATION CATALOG-NAME=MEIN.KATALOG }
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSC.SDF-A.040 } _____ (2)
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSC.INIT.130 }
//CHECK-CATALOG _____ (3)
CHECK REPORT:
**** NO ERROR ****
CHECK OF LINK REFERENCES:
VERSION RANGE CHECK:
**** NO ERROR ****
LINK RELATION CHECK:
**** NO ERROR ****
CHECK OF FUNCTIONAL DEPENDENCE:
VERSION RANGE CHECK:
**** NO ERROR ****
DEPENDENCE RELATION CHECK:
**** NO ERROR ****
CYCLE CHECK:
**** NO ERROR ****
//SAVE-CATALOG _____ (4)
% SCM0200 CATALOG 'MEIN.KATALOG' GENERATED

//END
% SCM0405 SSCM NORMALLY TERMINATED

//LOAD-LOCAL-SUBSYSTEM-CATALOG CATALOG-NAME=MEIN.KATALOG _____ (5)
% ESM0254 COMMAND 'LOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED
//SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (6)
% LOCAL SUBSYSTEM INIT /V13.0 IS NOT CREATED
% LOCAL SUBSYSTEM SDF-A /V04.0 IS NOT CREATED
% SUBSYSTEM INIT /V14.0 IS CREATED
% SUBSYSTEM SSCM /V02.3 IS CREATED
% SUBSYSTEM SDF-A /V04.1 IS CREATED
//START-SDF-A _____ (7)
% BLS0517 MODULE 'SDF-A' LOADED
% SDA0001 'SDF-A' VERSION '04.1E10' STARTED
//END

```

```

/START-LOCAL-SUBSYSTEM SUBSYSTEM-NAME=SDF-A,VERSION=04.0 _____ (8)
% ESM0220 FUNCTION 'CREATE' FOR SUBSYSTEM 'SDF-A /04.0' COMPLETELY
    PROCESSED

/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (9)
LOCAL SUBSYSTEM INIT /V13.0 IS NOT CREATED
LOCAL SUBSYSTEM SDF-A /V04.0 IS CREATED
% SUBSYSTEM INIT /V14.0 IS CREATED
% SUBSYSTEM SSCM /V02.3 IS CREATED
% SUBSYSTEM SDF-A /V04.1 IS CREATED

/START-SDF-A _____ (10)
% BLS0517 MODULE 'SDF-A' LOADED
% SDA0001 'SDF-A' VERSION '04.0A00' STARTED
//END

/SELECT-PRODUCT-VERSION PRODUCT-NAME=SDF-A,VERSION=04.1,-
    SCOPE=*PROGRAM _____ (11)
/START-SDF-A _____ (12)
% BLS0517 MODULE 'SDF-A' LOADED
% SCM5000 PROGRAM SDF-A VERSION V04.1E10 STARTED
//END
% SCM0405 SDF-A NORMALLY TERMINATED

/STOP-LOCAL-SUBSYSTEM SUBSYSTEM-NAME=SDF-A,VERSION=04.0 _____ (13)
% ESM0220 FUNCTION 'DELETE' FOR SUBSYSTEM 'SDF-A /04.0' COMPLETELY
    PROCESSED

/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (14)
% LOCAL SUBSYSTEM INIT /V13.0 IS NOT CREATED
% LOCAL SUBSYSTEM SDF-A /V04.0 IS NOT CREATED
% SUBSYSTEM INIT /V14.0 IS CREATED
% SUBSYSTEM SSCM /V02.3 IS CREATED
% SUBSYSTEM SDF-A /V04.1 IS CREATED

/START-SDF-A _____ (15)
% BLS0517 MODULE 'SDF-A' LOADED
% SDA0001 'SDF-A' VERSION '04.1E10' STARTED
//END

/SELECT-PRODUCT-VERSION PRODUCT-NAME=SDF-A,VERSION=04.0, -
    SCOPE=*PROGRAM _____ (16)
/START-SDF-A _____ (17)
% BLS0334 SYMBOL 'SDF-A' CANNOT BE FOUND. LOADING ABORTED
% NRTT101 ABNORMAL JOBSTEP TERMINATION BLS0532

/UNLOAD-LOCAL-SUBSYSTEM-CATALOG _____ (18)
% ESM0254 COMMAND 'UNLOAD-LOCAL-SUBSYSTEM-CATALOG' COMPLETELY PROCESSED

/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (19)
% SUBSYSTEM INIT /V14.0 IS CREATED
% SUBSYSTEM SSCM /V02.3 IS CREATED
% SUBSYSTEM SDF-A /V04.1 IS CREATED

```

- (1) Der Katalogverwalter SSCM wird aufgerufen.
- (2) Es wird ein lokaler Subsystemkatalog mit dem Namen MEIN.KATALOG erstellt. Die Subsysteme SDF-A V04.0 und INIT V13.0 werden als lokale Subsysteme aufgenommen.
- (3) Anschließend wird der lokale Katalog auf Verträglichkeit und Konsistenz der beiden lokalen Subsysteme geprüft. Die Prüfung ergibt keine Fehler.
- (4) Der lokale Katalog MEIN.KATALOG wird abgespeichert und SSCM beendet.
- (5) Der lokale Katalog MEIN.KATALOG wird in den Benutzeradressraum geladen.
- (6) Ab diesem Zeitpunkt können die lokalen Subsysteme (gemeinsam mit den globalen Subsystemen) angezeigt und/oder - bei Anforderung - in eine S-Variable ausgegeben werden.
Die beiden lokalen Subsysteme SDF-A V4.0 und INIT V13.0 sind im lokalen Katalog gespeichert, befinden sich aber im Zustand NOT-CREATED, weil sie noch nicht aktiviert wurden. Im globalen Subsystemkatalog stehen z.Zt. die aufgeführten drei Subsysteme zur Verfügung.
- (7) Ein zum Beispiel jetzt ausgeführtes Kommando START-SDF-A ruft das globale Subsystem SDF-A V4.1 auf, nicht das lokale mit der Version V4.0.
Das Subsystem SDF-A V4.1 wird ohne Ausführung von Anweisungen beendet.
- (8) Das lokale Subsystem SDF-A V4.0 wird aktiviert.
- (9) In der Anzeige ist zu sehen, dass es jetzt im Zustand CREATED ist.
- (10) Bei Aufruf START-SDF-A hat nun die lokale Subsystemversion Vorrang vor der globalen.
- (11) Die Priorität der verschiedenen Subsystemversionen wird zu Gunsten des globalen Subsystems SDF-A V4.1 geändert.
- (12) Das Kommando START-SDF-A ruft wieder das globale Subsystem SDF-A V4.1 auf.
- (13) Das lokale Subsystem SDF-A wird deaktiviert.
- (14) Beide lokalen Subsysteme sind deaktiviert.
- (15) Bei Aufruf START-SDF-A kann nur das globale Subsystem gestartet werden.
- (16) Die Priorität der verschiedenen Subsystemversionen wird zu Gunsten des lokalen Subsystems SDF-A V4.0 geändert.
- (17) Bei Aufruf START-SDF-A wird diese Version nicht gefunden (sie ist nicht im Zustand CREATED).
- (18) Der lokale Subsystemkatalog wird entladen.
- (19) Es werden keine lokalen Subsysteme mehr angezeigt.

3.9.2 Ausgabe in eine S-Variable

Das Beispiel zeigt die Ausgabe in S-Variablen in Abhängigkeit von der Privilegierung des Aufrufers. Es stellt keine reale Situation dar, sondern ist ein konstruiertes, vereinfachtes Beispiel zur Veranschaulichung der Problematik.

Folgende Privilegierungen sind gegeben:

- a) der Aufrufer hat nur das Privileg STD-PROCESSING
- b) der Aufrufer hat die Privilegien STD-PROCESSING sowie SUBSYSTEM-MANAGEMENT oder OPERATING
- c) der Aufrufer hat nicht das Privileg STD-PROCESSING, aber mindestens eines der Privilegien SUBSYSTEM-MANAGEMENT oder OPERATING

Folgende Konfiguration soll gegeben sein:

- das lokale Subsystem AA V04.5 ist im Zustand CREATED
- das globale, nicht-privilegierte Subsystem AA V04.6 ist im Zustand CREATED; es ist eine Task angeschlossen (TID=00010054, TSN=0123)
- das globale, privilegierte Subsystem XX V01.0 ist im Zustand CREATED: es ist eine Task angeschlossen (TID=0002006F, TSN=0BFC)
- das globale, privilegierte Subsystem XX V02.0 ist im Zustand IN-DELETE: es sind zwei Tasks angeschlossen (TID=00070015, TSN=0CMM und TID=00010057, TSN=00AP)

Es wird eine zusammengesetzte Variable vom Typ Liste mit dem Namen DATA deklariert, und dem S-Variablenstrom SYSINF zugewiesen.

```
/DECLARE-VARIABLE VAR-NAME=DATA(TYPE=*STRUCTURE),
MULTIPLE-ELEMENTS=*LIST
/ASSIGN-STREAM STREAM-NAME=SYSINF, TO=*VAT(VAR-NAME=DATA)
/SHOW-STREAM-ASSIGNMENT
```

```
STREAM-NAME    = SYSINF
ASSIGN-LEVEL   = 0
DESTINATION    = *VARIABLE
VARIABLE-NAME  = DATA
VAR-MODE       = *EXTEND
RETURN-VARIABLE-NAME = *NONE
CONTROL-VAR-NAME = *NONE
RET-CONTROL-VAR-NAME = *NONE
STREAM-NAME    = SYSMMSG
:
```

a) der Aufrufer hat nur das Privileg **STD-PROCESSING**

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (1)
```

```
% LOCAL SUBSYSTEM AA      /V04.5      IS CREATED
```

```
% SUBSYSTEM AA          /V04.6      IS CREATED
```

```
/SHOW-VAR DATA _____ (2)
```

```
:
```

```
DATA(*LIST).SUBSYS-TYPE='*LOC'
```

```
DATA(*LIST).SUBSYS-NAME='AA'
```

```
DATA(*LIST).SUBSYS-VERSION='04.5'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

```
DATA(*LIST).SUBSYS-TYPE='*GLB'
```

```
DATA(*LIST).SUBSYS-NAME='AA'
```

```
DATA(*LIST).SUBSYS-VERSION='04.6'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=AA _____ (3)
```

```
% LOCAL SUBSYSTEM AA      /V04.5      IS CREATED
```

```
% SUBSYSTEM AA          /V04.6      IS CREATED
```

```
/SHOW-VAR DATA _____ (4)
```

```
:
```

```
DATA(*LIST).SUBSYS-TYPE='*LOC'
```

```
DATA(*LIST).SUBSYS-NAME='AA'
```

```
DATA(*LIST).SUBSYS-VERSION='04.5'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

```
DATA(*LIST).SUBSYS-TYPE='*GLB'
```

```
DATA(*LIST).SUBSYS-NAME='AA'
```

```
DATA(*LIST).SUBSYS-VERSION='04.6'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

- (1) Es werden der Name, die Version und der Status aller globalen, nicht-privilegierten sowie aller lokalen Subsysteme nach SYSOUT ausgegeben.
Im konstruierten Beispiel ist nur das globale Subsystem AA nicht-privilegiert.
Es wird - zusammen mit dem lokalen Subsystem AA - ausgegeben.
- (2) Die Ausgabe in der S-Variablen DATA enthält explizit den Subsystem-Typ „global“ oder „lokal“.
- (3) Die Beschränkung in der Ausgabe auf das Subsystem AA ergibt keine geänderte Ausgabe nach SYSOUT im Vergleich zu (1).
- (4) Die Beschränkung in der Ausgabe auf das Subsystem AA ergibt keine geänderte Ausgabe in die S-Variable DATA im Vergleich zu (2).

- b) der Aufrufer hat die Privilegien **STD-PROCESSING** sowie **SUBSYSTEM-MANAGEMENT** und/oder **OPERATING**

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (5)
```

```
% LOCAL SUBSYSTEM AA      /V04.5    IS CREATED
% SUBSYSTEM AA           /V04.6    IS CREATED
% SUBSYSTEM XX           /V01.0    IS CREATED
% SUBSYSTEM XX           /V02.0    IS IN DELETE / WAIT-DISCON
```

```
/SHOW-VAR DATA _____ (6)
```

```
:
DATA(*LIST).SUBSYS-TYPE='*LOC'
DATA(*LIST).SUBSYS-NAME='AA'
DATA(*LIST).SUBSYS-VERSION='04.5'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='AA'
DATA(*LIST).SUBSYS-VERSION='04.6'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='XX'
DATA(*LIST).SUBSYS-VERSION='01.0'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='XX'
DATA(*LIST).SUBSYS-VERSION='02.0'
DATA(*LIST).SUBSYS-STA='*IN-DELETE'
DATA(*LIST).SUBSYS-INT-STA='WAIT-DISCON'
```

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=AA _____ (7)
```

```
% LOCAL SUBSYSTEM AA      /V04.5    IS CREATED
% SUBSYSTEM AA           /V04.6    IS USED BY 1 TASK
% TASKID 00010054
% TSN          0123
%   4 CONNECTIONS SINCE STARTUP
```

```
/SHOW-VAR DATA _____ (8)
```

```
:
DATA(*LIST).SUBSYS-TYPE='*LOC'
DATA(*LIST).SUBSYS-NAME='AA'
DATA(*LIST).SUBSYS-VERSION='04.5'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='AA'
DATA(*LIST).SUBSYS-VERSION='04.6'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).CONN-NUM-SINCE-START=4
DATA(*LIST).USED-TASK-LIST(*LIST).TID='00010054'
DATA(*LIST).USED-TASK-LIST(*LIST).TSN='0123'
```

- (5) Es werden der Name, die Version und der Status aller globalen sowie aller lokalen Subsysteme nach SYSOUT ausgegeben. Es wird also über alle Subsysteme informiert, die im laufenden System geladen sind.
Im konstruierten Beispiel sind das die globalen Subsysteme XX (privilegiert, in zwei Versionen) und AA (nicht-privilegiert) sowie das lokale Subsystem AA.
- (6) Die Ausgabe in der S-Variablen DATA enthält explizit den Subsystem-Typ „global“ oder „lokal“.
- (7) Die Beschränkung in der Ausgabe auf das Subsystem AA ergibt folgende Änderungen im Vergleich zu (1): Sind an ein Subsystem Tasks angeschlossen, werden sie mit ihrer TSN und TID angezeigt. Die Anzahl aller Tasks, die seit Startup an dieses Subsystem konnektiert waren, wird ebenfalls nach SYSOUT ausgegeben.
- (8) Die Ausgabe der Subsysteme AA in die S-Variable DATA enthält die gleichen Informationen wie in der SYSOUT-Ausgabe, ergänzt durch den Subsystem-Typ „global“ oder „lokal“.
- c) der Aufrufer hat nicht das Privileg STD-PROCESSING, aber mindestens eines der Privilegien **SUBSYSTEM-MANAGEMENT** oder **OPERATING**

```
/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=*ALL _____ (9)
```

```
% SUBSYSTEM AA      /V04.6   IS CREATED
```

```
% SUBSYSTEM XX      /V01.0   IS CREATED
```

```
% SUBSYSTEM XX      /V02.0   IS IN DELETE / WAIT-CLS-CTRL
```

```
/SHOW-VAR DATA _____ (10)
```

```
:
```

```
DATA(*LIST).SUBSYS-TYPE='*GLB'
```

```
DATA(*LIST).SUBSYS-NAME='AA'
```

```
DATA(*LIST).SUBSYS-VERSION='04.6'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

```
DATA(*LIST).SUBSYS-TYPE='*GLB'
```

```
DATA(*LIST).SUBSYS-NAME='XX'
```

```
DATA(*LIST).SUBSYS-VERSION='01.0'
```

```
DATA(*LIST).SUBSYS-STA='*CREATED'
```

```
DATA(*LIST).SUBSYS-TYPE='*GLB'
```

```
DATA(*LIST).SUBSYS-NAME='XX'
```

```
DATA(*LIST).SUBSYS-VERSION='02.0'
```

```
DATA(*LIST).SUBSYS-STA='*IN-DELETE'
```

```
DATA(*LIST).SUBSYS-INT-STA='WAIT-CLS-CTRL'
```

```

/SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=XX _____ (11)
% SUBSYSTEM XX      /V0.0      IS USED BY 1 TASK
% TASKID 0002006F
% TSN          0BFC
% 7 CONNECTIONS SINCE STARTUP
% SUBSYSTEM XX      /V02.0     IS USED BY 2 TASKS
% TASKID 00070015 00010057
% TSN          0CMM          00AP
%
%                               IS IN DELETE / WAIT-DISCON
% 12 CONNECTIONS SINCE STARTUP

```

```

/SHOW-VAR DATA _____ (12)
:
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='XX'
DATA(*LIST).SUBSYS-VERSION='01.0'
DATA(*LIST).SUBSYS-STA='*CREATED'
DATA(*LIST).CONN-NUM-SINCE-START=7
DATA(*LIST).USED-TASK-LIST(1).TID='0002006F'
DATA(*LIST).USED-TASK-LIST(1).TSN='0BFC'
DATA(*LIST).SUBSYS-TYPE='*GLB'
DATA(*LIST).SUBSYS-NAME='XX'
DATA(*LIST).SUBSYS-VERSION='02.0'
DATA(*LIST).SUBSYS-STA='*IN-DELETE'
DATA(*LIST).SUBSYS-INT-STA='WAIT-DISCON'
DATA(*LIST).CONN-NUM-SINCE-START=12
DATA(*LIST).USED-TASK-LIST(*LIST).TID='00070015'
DATA(*LIST).USED-TASK-LIST(*LIST).TSN='0CMM'
DATA(*LIST).USED-TASK-LIST(*LIST).TID='00010057'
DATA(*LIST).USED-TASK-LIST(*LIST).TSN='00AP'

```

- (9) Es werden der Name, die Version und der Status aller globalen Subsysteme nach SYSOUT ausgegeben.
Im konstruierten Beispiel sind das die globalen Subsysteme XX (privilegiert, in zwei Versionen) und AA (nicht-privilegiert).
- (10) Die Ausgabe in die S-Variable DATA beinhaltet die gleichen Informationen wie nach SYSOUT.
- (11) Die Beschränkung in der Ausgabe auf das Subsystem XX ergibt folgende Änderungen im Vergleich zu (9): Sind an ein Subsystem Tasks angeschlossen, werden sie mit ihrer TSN und TID angezeigt. Die Anzahl aller Tasks, die seit Startup an dieses Subsystem konnektiert waren, wird ebenfalls nach SYSOUT ausgegeben.
- (12) Die Ausgabe der Subsysteme XX in die S-Variable DATA enthält die gleichen Informationen wie in der SYSOUT-Ausgabe, ergänzt durch den Subsystem-Typ „global“.

4 SSCM

Das Subsystem SSCM (Static Subsystem Catalog Manager) ermöglicht eine flexible und anwenderfreundliche Verwaltung des statischen Subsystemkatalogs (SSMCAT).

SSCM V2.3 steht ab BS2000/OSD-BC V3.0 und zur Arbeit ab DSSM V3.8 zur Verfügung.

Zur Installation von SSCM V2.3 siehe [Seite 291](#), zur Koexistenz der SSCM-Versionen siehe [Seite 291](#), zu Fragen der Versionsabhängigkeit von SSCM in Bezug zu DSSM und BS2000/OSD siehe [Seite 28](#).

4.1 Generierung eines Subsystemkatalogs

Der zu erzeugende Subsystemkatalog muss auf das Home-Pubset gebracht und unter der Benutzerkennung TSOS abgelegt werden. Der Name des Katalogs ist frei wählbar und kann über den Parameterservice (siehe [Seite 64](#)) dem System bekannt gemacht werden.

Für die Generierung eines Subsystemkatalogs steht das Programm SSCM zur Verfügung.

Auf der folgenden Seite ist ein funktionaler Ablauf der Generierung eines Subsystemkatalogs skizziert:

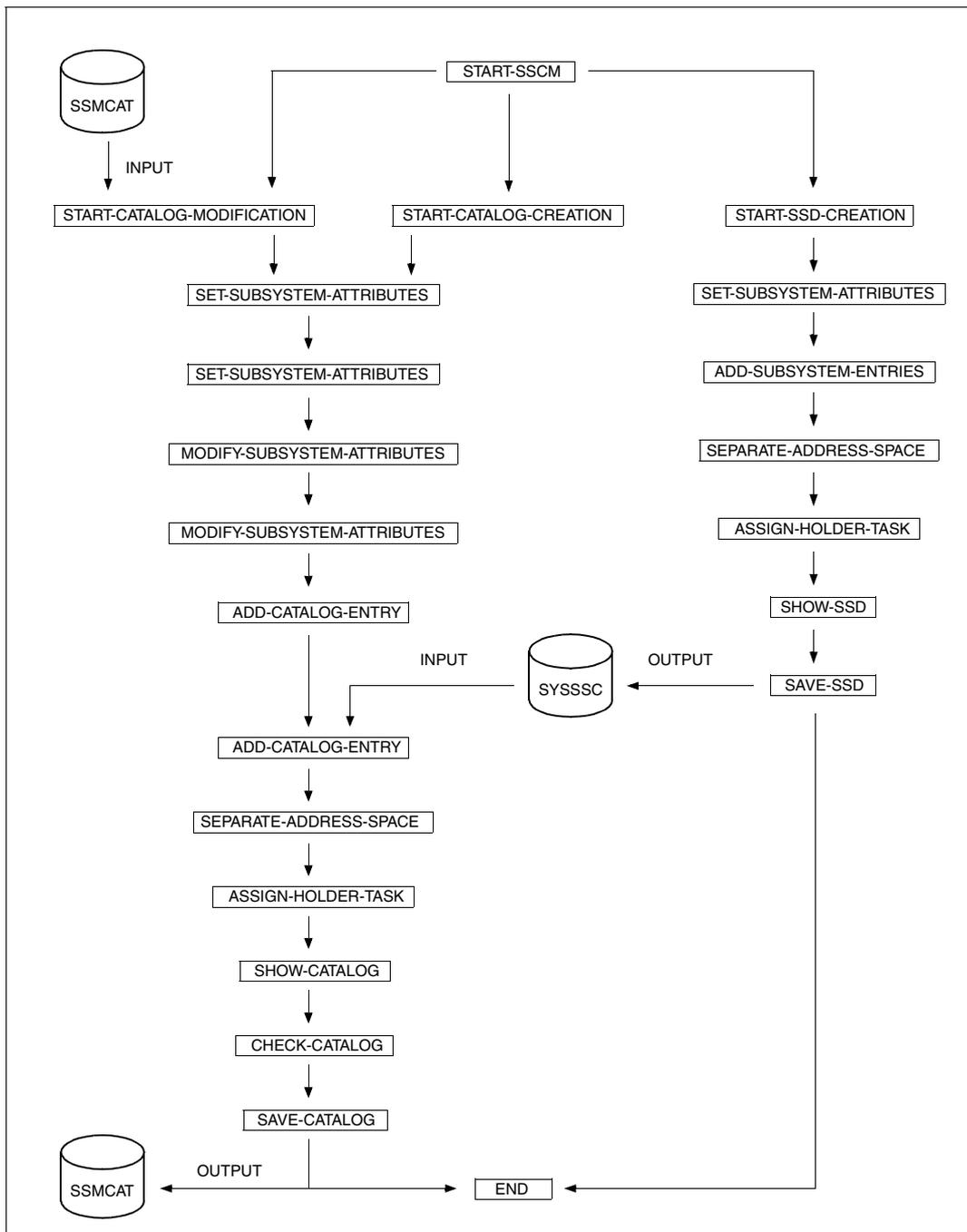


Bild 2: Ablauf einer Subsystemkatalog-Generierung

4.2 Starten und Beenden von SSCM

Das Programm SSCM wird mit dem Kommando **START-SSCM** gestartet.

START-SSCM
MONJV = *NONE / <filename 1..54 without-gen-vers> ,CPU-LIMIT = *JOB-REST / <integer 1..32767>

SSCM kann ebenso mit der Abkürzung **SSCM** aufgerufen werden.

Die Meldungsdatei für SSCM (\$TSOS.SYSMES.SSCM.023) wird aktiviert und das in der Modulbibliothek SYSLNK.SSCM.023 stehende Bindelademodul SSCM wird geladen (Meldung BLS0517).

Die Anweisung **END** beendet SSCM.

Programmüberwachung durch Monitor-Jobvariable

Die programmüberwachende Jobvariable muss im Kommando START-SSCM mit MONJV=<jv-name> vereinbart werden.

Sie kann von SSCM mit folgenden Werten versorgt werden:

- *\$T0000* Programm wurde normal beendet.
- *\$T1010* Anweisung abgewiesen, das Programm wird fortgesetzt.
- *\$A2010* Anweisung abgewiesen, das Programm wurde beendet.
- *\$A2015* Unerwartetes EOF auf SYSDTA, das Programm wurde abgebrochen.
- *\$A3020* SSCM-interner Fehler, das Programm wird abgebrochen.

Zur Programmüberwachung durch Monitor-Jobvariablen siehe auch Handbuch „Kommandos Band 1-5“ [19] und „Jobvariablen“ [18].

Überwachung durch Auftragsschalter

Wird eine Anweisung von SSCM abnormal beendet, wird der Auftragsschalter 31 eingeschaltet.

Ist der Auftragsschalter 1 eingeschaltet, führt SDF nur eine Syntaxprüfung durch.

4.3 Die Anweisungen von SSCM

SSCM wird mit einer SDF-Schnittstelle für die Benutzeroberfläche angeboten. Dem Anwender stehen somit alle Funktionen und Vorzüge, die SDF bietet - geführter Dialog, Hilfe-Texte für Operanden, Einsatz von Default-Werten - auch für die Nutzung von SSCM zur Verfügung.

Die **SDF-Syntaxdarstellung** der Kommandos ist in [Abschnitt „SDF-Syntaxdarstellung“ auf Seite 5](#) erläutert.

Folgende SSCM-Anweisungen stehen der (Sub-)Systembetreuung zur Verfügung:

Anweisung	Bedeutung	Seite
ADD-CATALOG-ENTRY	Subsystemdefinition(en) in den Katalog aufnehmen	183
ADD-SUBSYSTEM-ENTRIES	Zusätzliche Auftragseingänge definieren	186
ASSIGN-HOLDER-TASK	Subsysteme auf Holdertasks verteilen	193
CHECK-CATALOG	Subsystemdefinition(en) auf Konsistenz prüfen	196
GENERATE-CATALOG-SOURCE	SSCM-Anweisungsliste für Generierung erstellen	198
MODIFY-SUBSYSTEM-ATTRIBUTES	Eigenschaften von Subsystemen verändern	200
MODIFY-WORK-TASK-ATTRIBUTE	Arbeitstask-Parameter verändern	233
REMOVE-ADDR-SPACE-SEPARATION	Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher aufheben	235
REMOVE-CATALOG-ENTRY	Subsystemdefinition logisch aus dem Subsystemkatalog löschen	237
SAVE-CATALOG	Subsystemkatalog als PAM-Datei abspeichern	238
SAVE-SSD	Subsystemdefinition(en) abschließen	240
SEPARATE-ADDRESS-SPACE	Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher steuern	241
SET-SUBSYSTEM-ATTRIBUTES	Eigenschaften und Einsprungstellen eines Subsystems definieren	243
SHOW-CATALOG	Subsystem-Konfiguration anzeigen	274
SHOW-SSD	Inhalt eines SSD-Objects (Subsystemdefinitionen) anzeigen	281
START-CATALOG-CREATION	Name eines statischen Subsystemkatalogs vereinbaren	286
START-CATALOG-MODIFICATION	Statischen Subsystemkatalog verändern	287
START-SSD-CREATION	SSD-Object zur Aufnahme von Subsystemdefinitionen generieren	288

Tabelle 13: SSCM-Anweisungen

ADD-CATALOG-ENTRY

Subsystemdefinition(en) in den Subsystemkatalog aufnehmen

Funktionsbeschreibung

Mit dieser Anweisung werden die Definitionen neuer Subsysteme, die in einem SSD-Object hinterlegt sind, in den eröffneten Subsystemkatalog aufgenommen.

Die Anweisung wird zurückgewiesen, wenn kein aktueller Katalog, in den die Definitionen integriert werden sollen, mit der Anweisung START-CATALOG-CREATION oder START-CATALOG-MODIFICATION eröffnet wurde. Die Definition für ein Subsystem, das im eröffneten Katalog bereits enthalten ist, wird ignoriert; die Verarbeitung wird allerdings nach Ausgabe einer entsprechenden Meldung fortgesetzt.

Format

ADD-CATALOG-ENTRY

FROM-FILE = <filename 1..54>

,**INSTALLATION-USERID** = *UNCHANGED / *DEFAULT-USERID / <name 1..8>

,**CORRECTION-STATE** = *UNCHANGED / <c-string 3..3> / <text 3..3>

Operandenbeschreibung

FROM-FILE = <filename 1..54>

Name der Datei, in der die Subsystemdefinition(en) gesucht werden. Die Datei muss ein von SSCM generiertes SSD-Object vom Typ ISAM sein, in der die Eigenschaften eines oder mehrerer Subsysteme hinterlegt sind.

INSTALLATION-USERID =

Vereinbart eine Benutzerkennung, unter der die Nebenkompenten des Subsystems (Rep-Datei, Objektmodulbibliothek, Meldungsdatei, Syntaxdatei und Subsystem-Informationsdatei) erwartet werden, falls diese Dateien noch keiner Benutzerkennung zugeordnet sind.

INSTALLATION-USERID = *UNCHANGED

Voreinstellung: die Dateien werden unter der Benutzerkennung erwartet, die bei der Subsystemdefinition (Anweisung SET-SUBSYSTEM-ATTRIBUTES, [Seite 243](#)) angegeben wurde.

INSTALLATION-USERID = *DEFAULT-USERID

Die Dateien werden unter der System-Standardkennung erwartet (Prefix „\$.“).

INSTALLATION-USERID = <name 1..8>

Benutzerkennung, unter der die Dateien erwartet werden. Eine von dieser Angabe abweichende Vereinbarung, die in der Anweisung SET-SUBSYSTEM-ATTRIBUTES für ein SSD-Object getroffen wurde, wird überschrieben.

CORRECTION-STATE =

Ersetzt im Katalog die letzten drei Zeichen der Subsystemversion aus der SSD-Datei, die den Freigabe- und Korrekturstand darstellen.

Ist die Subsystemversion in der SSD-Datei mit vier Zeichen angegeben, werden die bei CORRECTION-STATE angegebenen drei Zeichen mit diesen verkettet.

CORRECTION-STATE = *UNCHANGED

Voreinstellung: Freigabe- und Korrekturstand bleiben unverändert.

CORRECTION-STATE = <text 3..3>

Angabe des Freigabe- und Korrekturstandes im Format: `aso` . Dabei bedeutet:

- a Freigabestand; alphabetisches Zeichen
- so Korrekturstand; numerische Zeichen

CORRECTION-STATE = <c-string 3..3>

Angabe des Freigabe- und Korrekturstandes als Zeichenkette im Format: `aso`
Zur Bedeutung siehe CORRECTION-STATE = <text 3..3> .

Hinweise

- Enthält eine Subsystemdefinition Dateinamen ohne Benutzerkennung und ist keine Installationskennung angegeben, sucht die zuständige DSSM-Task die Dateien unter der Benutzerkennung TSOS bzw. unter der Benutzerkennung der aufrufenden Task im Fall eines lokalen Subsystems.
Sind die Dateien unter einer anderen Kennung abgelegt, so muss diese beim Operanden INSTALLATION-USERID in einem der Kommandos ADD-CATALOG-ENTRY, SET-SUBSYSTEM-ATTRIBUTES oder MODIFY-SUBSYSTEM-ATTRIBUTES angegeben werden.
- Wird bei FROM-FILE ein SSD-Object angegeben, das mehrere Subsystemdefinitionen enthält, werden die Freigabe- und Korrekturstände aller in diesem SSD-Object definierten Subsysteme durch die Werte der Operanden CORRECTION-STATE und INSTALLATION-USERID beeinflusst.

- Die Dateinamen eines Subsystems werden nicht geändert, wenn Freigabe- und Korrekturstand angegeben sind. Wünscht der Anwender eine Namensänderung, muss er dies mit der Anweisung MODIFY-SUBSYSTEM-ATTRIBUTES veranlassen.
- Wird der Operand CORRECTION-STATE angegeben, kann es bei der Versionsprüfung für Subsysteme, die voneinander abhängig sind, oder für Subsysteme, die untereinander Adressbeziehungen erhalten, zu einem CHECK-CATALOG-Fehler kommen. Das kann durch geeignete Angaben bei der Anweisung MODIFY-SUBSYSTEM-ATTRIBUTES MODIFY-REFER-SUBS=..., MODIFY-RELATED-SUBS=... vermieden werden.

ADD-SUBSYSTEM-ENTRIES

Zusätzliche Auftragseingänge definieren

Funktionsbeschreibung

Mit dieser Anweisung ist es möglich, für ein Subsystem in einem SSD-Object mehr als die bereits bei SET-SUBSYSTEM-ATTRIBUTES angegebenen und maximal erlaubten 100 Auftragseingänge zu definieren.

Mit jeder Anweisung ADD-SUBSYSTEM-ENTRIES (sie kann für ein- und dasselbe Subsystem mehrmals ausgeführt werden) können bis zu 100 weitere Auftragseingänge für ein einzelnes Subsystem definiert werden.

ADD-SUBSYSTEM-ENTRIES wird zurückgewiesen, wenn zuvor die Anweisung START-SSD-CREATION nicht ausgeführt wurde.

Die Anweisung wird abgewiesen, wenn das angegebene Subsystem bei der Anweisung SET-SUBSYSTEM-ATTRIBUTES oder der Anweisung MODIFY-SUBSYSTEM-ATTRIBUTES mit dynamisch zu versorgenden Einsprungstellen (*BY-PROGRAM(...)) definiert wurde.

Die Ausführung der Anweisung wird abgebrochen, wenn

- das mit TO-SUBSYSTEM und VERSION bestimmte Subsystem nicht im aktuellen SSD-Object gefunden wird
 - ein bereits existierender Auftragseingang ein zweites Mal definiert werden soll
- Es wird eine Fehlermeldung ausgegeben.

Format

ADD-SUBSYSTEM-ENTRIES

```

TO-SUBSYSTEM = <structured-name 1..8>(…)
<structured-name 1..8>(…)
  | VERSION = <c-string 3..8> / <text 3..8> “
SUBSYSTEM-ENTRIES = list-poss(100): <text 1..8>(…)
<text 1..8>(…)
  | MODE = *LINK / *ISL(…) / *SVC(…) / *SYSTEM-EXIT(…)
    | *ISL(…)
      | FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
        | <integer 0..255>(…)
          | FUNCTION-VERSION = <integer 1..255>
    | *SVC(…)
      | NUMBER = <integer 0..255>
      | ,CALL-BY-SYSTEM-EXIT = *ALLOWED / *FORBIDDEN
      | ,FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
        | <integer 0..255>(…)
          | FUNCTION-VERSION = <integer 1..255>
    | *SYSTEM-EXIT(…)
      | NUMBER = <integer 0..127>
  | ,CONNECTION-ACCESS = *ALL / *SYSTEM / *SIH
  | ,CONNECTION-SCOPE = *TASK / *PROGRAM / *FREE / *CALL / *OPTIMAL
  | ,FIRST-CONNECTION = *ALLOWED / *FORBIDDEN

```

Operandenbeschreibung

TO-SUBSYSTEM = <structured-name 1..8>(…)

Vereinbart Name und Version des Subsystems, für das weitere Auftragseingänge definiert werden sollen.

VERSION = <c-string 3..8> / <text 3..8>

Die Version des Subsystems ist im Format „[V][m]m.n[aso]“ zu vereinbaren, wobei die Textteile folgende Bedeutung besitzen:

- mm : Hauptversion (numerisch)
- n : Nachtragsversion (numerisch)
- aso : Änderungsstand
(a: Buchstabe, Freigabestand; so: numerisch, Korrekturstand)

SUBSYSTEM-ENTRIES =

Vereinbart weitere Einsprungstellen (Auftragseingänge), die mit dem Subsystem verbunden sind. Es können bis zu 100 Auftragseingänge pro Anweisung vereinbart werden.

SUBSYSTEM-ENTRIES = list-poss(100): <text 1..8>

Vereinbart durch Angabe des Namens der Einsprungstelle maximal 100 Auftragseingänge, deren Typ jeweils in den Unterstrukturen definiert werden muss.

MODE =

Legt den Typ eines vereinbarten Auftragseingangs für das Subsystem fest.

MODE = *LINK

Voreinstellung: der Auftragseingang kann nicht über indirektes Linkage erreicht werden, sondern nur über eine CONNECT-Beziehung mittels eines externen Binder-Symbols.

Bei verschiedenen Versionen des gleichen Subsystems, die das gleiche externe Binder-Symbol nutzen, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

MODE = *ISL(...)

Der Auftragseingang wird durch indirekte Verbindung über System Procedure Linkage (nur für privilegierte Subsysteme) erreicht. Wird zusätzlich noch eine Funktions- und Versionsnummer der ISL-Einsprungstelle spezifiziert, darf sich die Kombination aus Name der Einsprungstelle, Funktions- und Versionsnummer nicht mit einer anderen Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEXISTENCE=*ALLOWED, siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über die selbe ISL-Einsprungstelle erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die dieselbe ISL-Einsprungsstelle nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die dieselbe ISL-Einsprungsstelle nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standard Header der Parameterliste des Aufrufers hinterlegt ist.

Die Angabe CONNECTION-ACCESS=*ALL (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES) ist für ISL-Einsprungsstellen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der ISL-Einsprungsstelle angesprochen werden soll, da die gleiche ISL-Einsprungsstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Voreinstellung: es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der ISL-Einsprungsstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten ISL-Funktionsnummer.

MODE = *SVC(…)

Der Auftragseingang wird durch indirekte Verbindung über Supervisor Call (SVC) erreicht.

Wird zusätzlich noch eine Funktions- und Versionsnummer der SVC-Einsprungsstelle spezifiziert, darf sich die Kombination aus SVC-Nummer, Funktions- und Versionsnummer nicht mit einer anderen Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEXISTENCE=*ALLOWED, siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über den selben SVC erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standardheader der Parameterliste des Aufrufers hinterlegt ist.

Bei Angabe dieses Operandenwertes sollte der Operand CONNECTION-ACCESS vorzugsweise mit dem Wert *SYSTEM statt *ALL belegt werden (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES).

NUMBER = <integer 0..255>

Nummer des SVCs, über den der Auftragseingang erreicht wird. In Verbindung mit CONNECTION-ACCESS=*ALL (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES) ist die Verwendung einer SVC-Nummer größer 191 nicht zulässig.

CALL-BY-SYSTEM-EXIT =

Legt fest, ob die angegebene SVC-Nummer von System-Exit-Routinen aus aufgerufen werden darf.

CALL-BY-SYSTEM-EXIT = *ALLOWED

Voreinstellung: der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen zulässig.

CALL-BY-SYSTEM-EXIT = *FORBIDDEN

Der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der SVC-Einsprungstelle angesprochen werden soll, da die gleiche SVC-Einsprungstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Voreinstellung: es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der SVC-Einsprungstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten SVC-Funktionsnummer.

MODE = SYSTEM-EXIT(…)

Der Auftragseingang wird durch indirekte Verbindung über System-Exit-Routinen erreicht.

In Verbindung mit CONNECTION-ACCESS=*ALL (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES) ist die Verwendung dieses Operanden nicht zulässig.

NUMBER = <integer 0..127>

Nummer der System-Exit-Routine.

CONNECTION-ACCESS =

Vereinbart die Zugriffsberechtigung (Privilegierung) für das Subsystem.

CONNECTION-ACCESS = *ALL

Voreinstellung: privilegierte und nicht-privilegierte Programmläufe sind zugriffsberechtigt.

In Verbindung mit MODE=*SYSTEM-EXIT, MODE=*ISL und MODE=*SVC (mit einer SVC-Nummer größer 191) ist die Verwendung dieses Operandenwertes nicht zulässig (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES).

CONNECTION-ACCESS = *SYSTEM

Nur privilegierte Programmläufe sind zugriffsberechtigt.

CONNECTION-ACCESS = *SIH

Nur Tasks, die im Funktionszustand SIH laufen, sind zugriffsberechtigt.

Das aufgerufene Subsystem läuft ebenfalls im Funktionszustand SIH, d.h. es ist nicht unterbrechbar.

Die Angabe dieses Operandenwertes ist nur für Subsysteme zulässig, deren Auftrags-
eingang definiert wird über

- System Procedure Linkage (MODE=*ISL(FUNCTION-NUMBER=*NONE))
- CONNECTION-SCOPE=*OPTIMAL
- MEMORY-CLASS=*SYSTEM-GLOBAL(SUBSYSTEM-ACCESS=*SYSTEM)

CONNECTION-SCOPE =

Bezeichnet das Ereignis, das die automatische Auflösung des Anschlusses an den angegebenen Subsystem-Auftragseingang hervorruft.

CONNECTION-SCOPE = *TASK

Voreinstellung: der Anschluss wird bei Taskbeendigung aufgehoben.

CONNECTION-SCOPE = *PROGRAM

Der Anschluss wird spätestens bei Programmbeendigung aufgehoben.

Zusammen mit MEMORY-CLASS=*LOCAL-UNPRIVILEGED darf nur CONNECTION-SCOPE=*PROGRAM angegeben werden (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES).

Die Angabe dieses Operandenwertes wird für Subsysteme empfohlen, die mit SUBSYSTEM-ACCESS=*LOW/*HIGH deklariert wurden.

CONNECTION-SCOPE = *FREE

DSSM soll keine Kontrolle von Anschlüssen zu diesen Auftragseingängen durchführen. Der Anschluss wird - außer bei einer expliziten Anforderung - nicht automatisch aufgelöst. Um Probleme oder mögliche Fehler beim Entladen des Subsystems zu vermeiden, müssen die Anschlüsse vom Subsystem selbst verwaltet werden.

CONNECTION-SCOPE = *CALL

Nach Rückkehr aus diesem Auftragseingang führt DSSM automatisch die Auflösung der Anschlüsse durch.

Dieser Operandenwert steht nur für Subsysteme zur Verfügung, deren Auftragseingang über System Procedure Linkage (ISL) oder Supervisor Call (SVC) definiert wird.

CONNECTION-SCOPE = *OPTIMAL

Das Subsystem wird deaktiviert bzw. angehalten, wenn keine Task mehr Anschluss zu diesem Auftragseingang hat.

Eine Routine, deren Einsprungstelle mit *OPTIMAL definiert wird, muss mit RETURN beendet werden.

Wird ein Auftragseingang eines Subsystems mit CONNECTION-SCOPE=*OPTIMAL definiert, sollten alle seine Auftragseingänge im Subsystemkatalog mit MODE=*LINK definiert werden.

Während ein Subsystem deaktiviert oder angehalten wird, wird kein Aufruf des Subsystems mit CONNECTION-SCOPE=*OPTIMAL akzeptiert. Ausnahme: das Subsystem wurde mit CREATION-TIME=*AT-SUBSYSTEM-CALL definiert und die aufrufende Task ist bereits mit ihm verbunden (siehe Anweisung SET-SUBSYSTEM-ATTRIBUTES).

FIRST-CONNECTION =

Bestimmt, ob der Erst-Anschluss der Task an den angegebenen Auftragseingang des Subsystems erlaubt ist. Mindestens ein Auftragseingang eines Subsystems muss mit FIRST-CONNECTION = *ALLOWED definiert werden.

FIRST-CONNECTION = *ALLOWED

Voreinstellung: Der Erst-Anschluss an den angegebenen Auftragseingang ist erlaubt.

FIRST-CONNECTION = *FORBIDDEN

Der Anschluss an den angegebenen Auftragseingang über SVC oder ISL ist nicht erlaubt, wenn die Task nicht bereits an einen anderen Auftragseingang des Subsystems angeschlossen ist.

Die Angabe dieses Operandenwertes ist für Auftragseingänge, die mit MODE=*LINK/*SYSTEM-EXIT oder CONNECTION-ACCESS=*SIH definiert wurden, nicht erlaubt.

ASSIGN-HOLDER-TASK

Subsysteme auf Holdertasks verteilen

Funktionsbeschreibung

Mit dieser Anweisung wird die Verteilung der Subsysteme auf Holdertasks gesteuert. Die in der Anweisung aufgeführten Subsysteme können die Holdertask als Arbeitstask nutzen oder werden gemeinsam in einer Holdertask angelegt. Wird bei der Erstellung eines Katalogs auf diese Anweisung verzichtet, nimmt SSCM eine Standardaufteilung der Subsysteme vor, um die Zahl der Holdertasks zu begrenzen.

Für Subsysteme, die die Holdertask als Arbeitstask nutzen, ist die Anweisung verpflichtend.

Um Arbeitstask zu werden, müssen die Auftragseingänge des Subsystems mit einer der folgenden Kombinationen definiert sein:

CLOSE-CTRL	STOPCOM	DEINIT
*DYNAMIC	*DYNAMIC	*DYNAMIC
*DYNAMIC	*NO	*DYNAMIC
*NO	*DYNAMIC	*DYNAMIC *
*NO	*NO	*DYNAMIC *

* Aus Kompatibilitätsgründen ist es für ein Subsystem ohne CLOSE-CTRL-Routine nicht Pflicht, eine DEINIT-Routine zu definieren. Es ist jedoch darauf zu achten, dass in einem solchen Fall keine Garantie für den ordnungsgemäßen Ablauf des Subsystems gegeben werden kann.

Die Anweisung darf bei einem SSD-Object nur einmal pro Subsystem verwendet werden. Bei zwei aufeinander folgenden Anweisungen für das gleiche Subsystem hat im Konfliktfall - sich widersprechende Vereinbarungen - die Definition Vorrang, wonach die Holdertask als Arbeitstask genutzt werden soll.

Die Vereinbarung bezüglich einer gemeinsamen Holdertask gilt jeweils für alle Versionen des Subsystems mit Ausnahme der Versionen, die die Holdertask als Arbeitstask nutzen.

ASSIGN-HOLDER-TASK wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-SSD-CREATION
- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Format

ASSIGN-HOLDER-TASK
TYPE = *WORK-TASK (...) / *SHARED-HOLDER (...)
*WORK-TASK (...)
SUBSYSTEM-NAME = <structured-name 1..8>
, SUBSYSTEM-VERSION = <c-string 3..8> / <text 3..8>
, TSN = *BY-DSSM / <alphanum-name 1..4>
*SHARED-HOLDER (...)
BY-SUBSYSTEMS = list-poss(15): <structured-name 1..8>
, TSN = *BY-DSSM / <alphanum-name 1..4>

Operandenbeschreibung

TYPE =

Legt fest, ob die Holdertask als Arbeitstask genutzt oder das Subsystem in einer gemeinsamen Holdertask angelegt werden soll.

TYPE = ***WORK-TASK**(...)

Voreinstellung: die Holdertask soll als Arbeitstask genutzt werden.

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

Name des Subsystems, das die Holdertask als Arbeitstask nutzen soll.

SUBSYSTEM-VERSION = <c-string 3..8> / <text 3..8>

Version des Subsystems, das die Holdertask als Arbeitstask nutzen soll.
Die Version muss bereits deklariert sein.

TSN =

Bestimmt, welche Task Sequence Number (TSN) die Arbeitstask des Subsystems erhalten soll.

TSN = ***BY-DSSM**

Voreinstellung: die TSN wird beim Laden der Arbeitstask von DSSM vergeben.

TSN = <alphanum-name 1..4>

TSN, die die Arbeitstask beim Starten erhalten soll. Die angegebene TSN muss eindeutig festgelegt und verwendbar sein, wenn das Subsystem geladen wird.

TYPE = *SHARED-HOLDER(...)

Das Subsystem soll in einer gemeinsamen Holdertask angelegt werden.

BY-SUBSYSTEMS = list-poss(15): <structured-name 1..8>

Name der maximal 15 Subsysteme, die in der gleichen Holdertask angelegt werden müssen.

Das erste in der Liste aufgeführte Subsystem muss bereits deklariert sein.

TSN =

Bestimmt, welche Task Sequence Number (TSN) die gemeinsame Holdertask erhalten soll.

TSN = *BY-DSSM

Voreinstellung: die TSN wird beim Laden der Arbeitstask von DSSM vergeben.

TSN = <alphanum-name 1..4>

TSN, die die gemeinsame Holdertask beim Starten erhalten soll. Die angegebene TSN muss eindeutig festgelegt und verwendbar sein, wenn das Subsystem geladen wird.

CHECK-CATALOG

Subsystemdefinition(en) auf Konsistenz prüfen

Funktionsbeschreibung

Mit dieser Anweisung werden die in einem Katalog zusammengefassten Subsystemdefinitionen auf Konsistenz geprüft.

CHECK-CATALOG wird zurückgewiesen, wenn der anzugebende Dateiname nicht existiert oder der Subsystemkatalog ohne Inhalt ist.

Wenn der angegebene Dateiname nicht dem des aktuell eröffneten Kataloges entspricht, wird folgende Meldung ausgegeben:

```
SCM0011      WOLLEN SIE WIRKLICH DEN KATALOG IM SPEICHER ÜBERSCHREIBEN ? (Y/N)
```

Bei Antwort **Y** gehen die Definitionen des aktuellen Katalogs verloren.

Bei Antwort **N** wird die Ausführung der Anweisung CHECK-CATALOG abgebrochen. Der Anwender kann mit SAVE-CATALOG die bisher nicht gespeicherten Subsystemdefinitionen in einer Datei ablegen.



Ohne vorherige Prüfung der Link- und Abhängigkeitsbeziehungen kann der Katalog nicht gesichert werden.

Format

CHECK-CATALOG

```
CATALOG-NAME = *CURRENT / <filename 1..54 without-gen-vers>
,DEPENDENCE-RELATION = *YES / *NO
,LINK-RELATION = *YES / *NO
,RELATED-FILES = *NO / *YES
,OUTPUT = *SYSOUT / *SYSLST(...)
  *SYSLST(...)
  | SYSLST-NUMBER = *STD / <integer 1..99>
```

Operandenbeschreibung

CATALOG-NAME =

Vereinbart den Namen des Kataloges, in dem die zu prüfenden Definitionen hinterlegt sind.

CATALOG-NAME = *CURRENT

Voreinstellung: der aktuell geöffnete Katalog (Anweisung START-CATALOG-CREATION bzw. START-CATALOG-MODIFICATION) soll geprüft werden.

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

Vollqualifizierter Name des statischen Subsystemkataloges, dessen Inhalt geprüft wird.

DEPENDENCE-RELATION = *YES / *NO

Vereinbart, ob in die Prüfung der Subsystemdefinitionen auch die verankerten Abhängigkeitsbeziehungen zu anderen Subsystemen einbezogen werden sollen (*YES) oder nicht (*NO). Der Katalog kann nicht gesichert werden, wenn in einer vorhergehenden CHECK-CATALOG-Anweisung *NO angegeben wurde.

LINK-RELATION = *YES / *NO

Vereinbart, ob in die Prüfung der Subsystemdefinitionen auch die verankerten Adressbeziehungen zu anderen Subsystemen einbezogen werden sollen (*YES, Voreinstellung) oder nicht (*NO). Der Katalog kann nicht gesichert werden, wenn in einer vorhergehenden CHECK-CATALOG-Anweisung *NO angegeben wurde.

RELATED-FILES = *NO / *YES

Bestimmt, ob die Existenz von Dateien, die in Abhängigkeitsbeziehungen zu diesen Subsystemen stehen, geprüft werden soll (*YES) oder nicht (*NO, Voreinstellung).

Wurden die Namen von abhängigen Dateien mit dem Wert *INSTALLED(...) definiert, wird auch der dort angegebene DEFAULT-NAME geprüft.

OUTPUT =

Vereinbart, wohin die von der Anweisung generierten Informationen - die Ergebnisse des Prüflaufs - ausgegeben werden sollen.

OUTPUT = *SYSOUT

Voreinstellung: die Meldungen werden auf der Datensichtstation ausgegeben.

OUTPUT = *SYSLST(...)

Die Meldungen sollen nach SYSLST ausgegeben werden.

SYSLST-NUMBER =

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

SYSLST-NUMBER = *STD

Voreinstellung: die Ausgabe soll in die Standard-Systemdatei SYSLST erfolgen.

SYSLST-NUMBER = <integer 1..99>

Die Ausgabe soll in eine Systemdatei aus der Menge SYSLST01 bis SYSLST99 erfolgen, deren Nummer hier anzugeben ist.

GENERATE-CATALOG-SOURCE

SSCM-Anweisungsliste für Generierung erstellen

Funktionsbeschreibung

Mit dieser Anweisung erstellt SSCM eine Datei mit einer Liste aller SSCM-Anweisungen, die für die (Wieder-)Generierung eines Subsystemkatalogs benötigt werden (für bestimmte oder für alle Subsysteme des Eingabe-Katalogs).

Format

GENERATE-CATALOG-SOURCE
<pre> CATALOG-NAME = *<u>CURRENT</u> / <filename 1..54 without-gen-vers> , SUBSYSTEM-NAME = *<u>ALL</u> / <structured-name 1..8>(…) <structured-name 1..8>(…) VERSION = *<u>ALL</u> / <c-string 3..8> / <text 3..8> , OUTPUT = *<u>SYSLST</u>(…) *<u>SYSLST</u>(…) SYSLST-NUMBER = *<u>STD</u> / <integer 1..99> </pre>

Operandenbeschreibung

CATALOG-NAME =

Gibt den Subsystemkatalog an, in dem die Subsystemdefinitionen gespeichert sind.

CATALOG-NAME = *CURRENT

Voreinstellung. Es wird der aktuelle Subsystemkatalog verwendet.

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

Name des Subsystemkatalogs.

SUBSYSTEM-NAME =

Subsysteme, deren Definitionen ausgegeben werden sollen.

SUBSYSTEM-NAME = *ALL

Voreinstellung. Die Definitionen aller Subsysteme sollen ausgegeben werden.

SUBSYSTEM-NAME = <structured-name 1..8>(…)

Name des Subsystems, dessen Definition ausgegeben werden soll.

VERSION = *ALL / <c-string 3..8> / <text 3..8>

Version des Subsystems, dessen Definition ausgegeben werden soll.

OUTPUT = *SYSLST(…)

Systemdatei, an die die generierten Informationen geschickt werden sollen.

SYSLST-NUMBER = *STD

Voreinstellung. Die Informationen werden an die Systemdatei SYSLST geschickt.

SYSLST-NUMBER = <integer 1..99>

Die Informationen werden an die aus dem Bereich SYSLST01 bis SYSLST99 angegebene Systemdatei geschickt.

MODIFY-SUBSYSTEM-ATTRIBUTES

Eigenschaften von Subsystemen verändern

Funktionsbeschreibung

Mit dieser Anweisung können alle Eigenschaften und Einsprungstellen, die mit der Anweisung SET-SUBSYSTEM-ATTRIBUTES definiert wurden, verändert werden.

Bei der Veränderung einer Definition ist auf Folgendes zu achten:

- Das Subsystem - identifiziert durch Name und Version - muss im aktuell geöffneten Katalog gefunden werden
- Der Versuch, einen bereits bestehenden Auftragseingang oder bereits bestehende Beziehungen hinzuzufügen, wird abgewiesen
- Nicht zulässig ist, einen noch nicht bestehenden Auftragseingang oder noch nicht definierte Beziehungen zu verändern oder zu löschen
- Der Modus eines Auftragseingangs darf nur verändert werden, wenn alle Parameter angegeben werden; die Standardwerte *UNCHANGED werden zurückgewiesen
- Die Speicherklasse eines Subsystems darf nur verändert werden, wenn alle Parameter angegeben werden; die Standardwerte *UNCHANGED werden zurückgewiesen

MODIFY-SUBSYSTEM-ATTRIBUTES wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Hinweis zur Syntax

Für die Namen der Einsprungstellen in den folgenden Operanden (im Format ist der Datentyp <name> angegeben) kann auch der spezielle Datentyp <symbol> verwendet werden, der im Handbuch „BLSSERV“ [4] ausführlich beschrieben ist:

- | | |
|-----------------------|-----------------------|
| – LINK-ENTRY | – DEINIT-ROUTINE |
| – DYNAMIC-CHECK-ENTRY | – INTERFACE-VERSION |
| – INIT-ROUTINE | – ADD-SUBS-ENTRIES |
| – CLOSE-CTRL-ROUTINE | – MODIFY-SUBS-ENTRIES |
| – STOPCOM-ROUTINE | – REMOVE-SUBS-ENTRIES |

Format

(Teil 1 von 4)

MODIFY-SUBSYSTEM-ATTRIBUTES

```

SUBSYSTEM-NAME = <structured-name 1..8>(…)
    <structured-name 1..8>(…)
        | VERSION = <c-string 3..8> / <text 3..8>
,INSTALLATION-UNIT = *UNCHANGED / *NONE / *STD / <text 1..30>
,INSTALLATION-USERID = *UNCHANGED / *NONE / *DEFAULT-USERID / <name 1..8>
,COPYRIGHT = *UNCHANGED / *NONE / <c-string 1..54>(…)
    <c-string 1..54>(…)
        | YEAR = *YEAR-1990 / <c-string 4..4>
,LIBRARY = *UNCHANGED / *STD / *CPLINK / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        | LOGICAL-ID = *UNCHANGED / <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = *UNCHANGED / <filename 1..54 without-gen-vers>
,SUBSYSTEM-LOAD-MODE = *UNCHANGED / *STD / *ADVANCED
,REP-FILE = *UNCHANGED / *STD / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        | LOGICAL-ID = *UNCHANGED / <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = *UNCHANGED / <filename 1..54 without-gen-vers> / *NONE
,REP-FILE-MANDATORY = *UNCHANGED / *NO / *YES
,MESSAGE-FILE = *UNCHANGED / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        | LOGICAL-ID = *UNCHANGED / <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = *UNCHANGED / <filename 1..54 without-gen-vers> / *NONE
,SUBSYSTEM-INFO-FILE = *UNCHANGED / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        | LOGICAL-ID = *UNCHANGED / <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = *UNCHANGED / <filename 1..54 without-gen-vers> / *NONE
,SYNTAX-FILE = *UNCHANGED / *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        | LOGICAL-ID = *UNCHANGED / <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = *UNCHANGED / <filename 1..54 without-gen-vers> / *NONE

```

Fortsetzung ➔

```

,DYNAMIC-CHECK-ENTRY = *UNCHANGED / *STD / *NO / <text 1..8 without-sep>
,CREATION-TIME = *UNCHANGED / *AT-CREATION-REQUEST / *AT-SUBSYSTEM-CALL(...) /
    *AT-DSSM-LOAD / *BEFORE-DSSM-LOAD / *MANDATORY-AT-STARTUP /
    *BEFORE-SYSTEM-READY / *AFTER-SYSTEM-READY
    *AT-SUBSYSTEM-CALL(...)
        |   ON-ACTION = *STD / *ISL-CALL / *ANY
,INIT-ROUTINE = *UNCHANGED / *NO / <text 1..8 without-sep>
,CLOSE-CTRL-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,STOPCOM-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,DEINIT-ROUTINE = *UNCHANGED / *NO / *DYNAMIC / <text 1..8 without-sep>
,STOP-AT-SHUTDOWN = *UNCHANGED / *NO / *YES
,INTERFACE-VERSION = *UNCHANGED / *NO / <text 1..8 without-sep>
,SUBSYSTEM-HOLD = *UNCHANGED / *ALLOWED / *FORBIDDEN
,STATE-CHANGE-CMDS = *UNCHANGED / *ALLOWED / *FORBIDDEN / *BY-ADMINISTRATOR-ONLY
,FORCED-STATE-CHANGE = *UNCHANGED / *ALLOWED / *FORBIDDEN
,RESET = *UNCHANGED / *ALLOWED / *FORBIDDEN
,RESTART-REQUIRED = *UNCHANGED / *NO / *YES
,VERSION-COEEXISTENCE = *UNCHANGED / *FORBIDDEN / *ALLOWED
,VERSION-EXCHANGE = *UNCHANGED / *FORBIDDEN / *ALLOWED
,ADD-SUBS-ENTRIES = *NONE / list-poss(100): <text 1..8>(…)
<text 1..8>(…)
    |   MODE = *LINK / *ISL(...) / *SVC(...) / *SYSTEM-EXIT(...)
        |   *ISL(...)
            |   FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
                |   <integer 0..255>(…)
                    |   FUNCTION-VERSION = <integer 1..255>
        |   *SVC(...)
            |   NUMBER = <integer 0..255>
                |   ,CALL-BY-SYSTEM-EXIT = *ALLOWED / *FORBIDDEN
                    |   ,FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
                        |   <integer 0..255>(…)
                            |   FUNCTION-VERSION = <integer 1..255>
    
```

Fortsetzung →

```

*SYSTEM-EXIT(...)
|   NUMBER = <integer 0..127>
,CONNECTION-ACCESS = *ALL / *SYSTEM / *SIH
,CONNECTION-SCOPE = *TASK / *PROGRAM / *FREE / *CALL / *OPTIMAL
,FIRST-CONNECTION = *ALLOWED / *FORBIDDEN
,MODIFY-SUBS-ENTRIES = *NONE / list-poss(100): <text 1..8>(…) / *BY-PROGRAM(…)
<text 1..8>(…)
MODE = *UNCHANGED / *LINK / *ISL(…) / *SVC(…) / *SYSTEM-EXIT(…)
*ISL(…)
|   FUNCTION-NUMBER = *UNCHANGED (…) / *NONE / <integer 0..255>(…)
|   *UNCHANGED(…)
|   |   FUNCTION-VERSION = *UNCHANGED / <integer 1..255>
|   <integer 0..255>(…)
|   |   FUNCTION-VERSION = <integer 1..255>
*SVC(…)
|   NUMBER = *UNCHANGED / <integer 0..255>
|   ,CALL-BY-SYSTEM-EXIT = *UNCHANGED / *ALLOWED / *FORBIDDEN
|   ,FUNCTION-NUMBER = *UNCHANGED (…) / *NONE / <integer 0..255>(…)
|   *UNCHANGED(…)
|   |   FUNCTION-VERSION = *UNCHANGED / <integer 1..255>
|   <integer 0..255>(…)
|   |   FUNCTION-VERSION = <integer 1..255>
*SYSTEM-EXIT(...)
|   NUMBER = <integer 0..127>
,CONNECTION-ACCESS = *UNCHANGED / *ALL / *SYSTEM / *SIH
,CONNECTION-SCOPE = *UNCHANGED / *TASK / *PROGRAM / *FREE / *CALL / *OPTIMAL
,FIRST-CONNECTION = *UNCHANGED / *ALLOWED / *FORBIDDEN
*BY-PROGRAM(…)
|   CONNECTION-SCOPE = *UNCHANGED / *TASK / *PROGRAM
,REMOVE-SUBS-ENTRIES = *NONE / list-poss(100): <text 1..8>
,MEMORY-CLASS = *UNCHANGED / *SYSTEM-GLOBAL(…) / *LOCAL-PRIVILEGED(…) /
*LOCAL-UNPRIVILEGED(…) / *BY-SLICE(…)

```

Fortsetzung ➔

```

*SYSTEM-GLOBAL(...)
|   SUBSYSTEM-ACCESS = *LOW / *SYSTEM / *HIGH
*LOCAL-PRIVILEGED(...)
|   SIZE = <integer 1..32767>
*LOCAL-UNPRIVILEGED(...)
|   SIZE = *UNCHANGED / <integer 1..32767>
|   ,SUBSYSTEM-ACCESS = *UNCHANGED / *LOW / *HIGH
|   ,START-ADDRESS = *UNCHANGED / *ANY / <x-string 7..8>
*BY-SLICE(...)
|   SIZE = <integer 1..32767>
,LINK-ENTRY = *UNCHANGED / <text 1..8 without-sep>(…)
<text 1..8 without-sep>(…)
|   AUTOLINK = *ALLOWED / *FORBIDDEN
,ADD-REFER-SUBS = *NONE / list-poss(15): <structured-name 1..8>(…)
<structured-name 1..8>(…)
|   LOWEST-VERSION = *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
|   ,HIGHEST-VERSION = *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>
,MODIFY-REFER-SUBS = *NONE / list-poss(15): <structured-name 1..8>(…)
<structured-name 1..8>(…)
|   LOWEST-VERSION = *UNCHANGED / *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
|   ,HIGHEST-VERSION = *UNCHANGED / *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>
,REMOVE-REFER-SUBS = *NONE / list-poss(15): <structured-name 1..8>
,UNRESOLVED-EXTERNALS = *UNCHANGED / *ALLOWED / *FORBIDDEN
,CHECK-REFERENCE = *UNCHANGED / *YES / *NO
,ADD-RELATED-SUBS = *NONE / list-poss(100): <structured-name 1..8>(…)
<structured-name 1..8>(…)
|   LOWEST-VERSION = *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
|   ,HIGHEST-VERSION = *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>
,MODIFY-RELATED-SUBS = *NONE / list-poss(100): <structured-name 1..8>(…)
<structured-name 1..8>(…)
|   LOWEST-VERSION = *UNCHANGED / *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
|   ,HIGHEST-VERSION = *UNCHANGED / *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>
,REMOVE-RELATED-SUBS = *NONE / list-poss(100): <structured-name 1..8>

```

Operandenbeschreibung

Der Standardwert *UNCHANGED bedeutet jeweils, dass der in der Anweisung SET-SUBSYSTEM-ATTRIBUTES eingestellte Wert gültig bleiben soll.

Wird der Typ des vereinbarten Auftragseingangs (Operand MODE) oder der subsystemspezifische Adressraum (Operand MEMORY) verändert, müssen alle Unteroperanden von MODE bzw. MEMORY explizit mit einem Wert versehen werden (der Operandenwert *UNCHANGED (Voreinstellung) wird abgewiesen).

SUBSYSTEM-NAME = <structured-name 1..8>(…)

Vereinbart Name und Version des Subsystems, dessen Definition verändert werden soll.

VERSION = <c-string 3..8> / <text 3..8>

Die Version des Subsystems ist im Format „[V][m]m.n[aso]“ zu vereinbaren, wobei die Textteile folgende Bedeutung besitzen:

- mm : Hauptversion (numerisch)
- n : Nachtragsversion (numerisch)
- aso : Änderungsstand
(a: Buchstabe, Freigabestand; so: numerisch, Korrekturstand)

INSTALLATION-UNIT =

Legt den Namen der installierten Liefergruppe fest. Für alle mit IMON zu installierenden Subsysteme muss ein Wert ungleich *NONE angegeben werden, ebenso, wenn bei den Operanden SUBSYSTEM-LIBRARY, REP-FILE, SUBSYSTEM-INFO-FILE, MESSAGE-FILE und SYNTAX-FILE der Wert *INSTALLED(LOGICAL-ID=…) definiert wurde.

Die im Handbuch „IMON“ [17] dargestellten Syntaxregeln sind bei der Festlegung des Namens zu beachten.

INSTALLATION-UNIT = *NONE

Es wird kein Name vergeben. Für alle mit IMON installierten Subsysteme ist diese Angabe nicht erlaubt.

INSTALLATION-UNIT = *STD

Der beim Operanden SUBSYSTEM-NAME angegebene Name wird als neuer Name der installierten Liefergruppe genutzt.

INSTALLATION-UNIT = <text 1..30>

Neuer Name der installierten Liefergruppe.

INSTALLATION-USERID =

Vereinbart eine Benutzerkennung, unter der die zuständige DSSM-Task die Nebenkompenten des Subsystems (Rep-Datei, Objektmodulbibliothek, Meldungsdatei, Syntaxdatei und Subsystem-Informationsdatei) erwartet, falls diese Dateien noch keiner Benutzerkennung zugeordnet sind, d.h. der Dateiname ohne Benutzerkennung angegeben wurde.

INSTALLATION-USERID = *NONE

Die Dateien werden nicht unter einer bestimmten Benutzerkennung erwartet.

INSTALLATION-USERID = *DEFAULT-USERID

Die Dateien werden unter der System-Standardkennung erwartet (Prefix „\$.“) bzw. unter der Benutzerkennung der aufrufenden Task, wenn es ein lokales Subsystem ist.

INSTALLATION-USERID = <name 1..8>

Benutzerkennung, unter der die NebenkompONENTEN erwartet werden.

Gilt die Anweisung für ein SSD-Object, werden die Dateien nur dann unter der hier angegebenen Benutzerkennung erwartet, wenn in der Anweisung ADD-CATALOG-ENTRY (Übernahme von Subsystemdefinitionen aus dem SSD-Object in den Katalog, [Seite 183](#)) keine Benutzerkennung angegeben wurde. Die Angabe bei ADD-CATALOG-ENTRY hat Vorrang.

COPYRIGHT =

Vereinbart, ob und welche Copyright-Meldung beim Starten des Subsystems ausgegeben werden soll.

COPYRIGHT = *NONE

Es soll keine Copyright-Meldung ausgegeben werden.

COPYRIGHT = <c-string 1..54>(…)

Text der Copyright-Meldung, die zusammen mit dem Datum der Erstellung beim Starten ausgegeben wird.

YEAR = *YEAR-1990 / <c-string 4..4>

Jahreszahl, die in der Copyright-Meldung als Datum der Erstellung erscheinen soll.
Eine semantische Prüfung findet nicht statt.

LIBRARY =

Vereinbart einen neuen Namen für die Programm- oder Bindemodulbibliothek (OML), aus der der Objectcode des Subsystems bei dessen Aktivierung geladen werden soll.

LIBRARY = *STD

Der Objectcode wird beim Starten automatisch aus der Bibliothek

SYSLNK.<subsysname>.<subsysvers#> geladen. Sie ist auf der Benutzerkennung abgelegt, unter der die Holdertask läuft; also auf der Benutzerkennung des Aufrufers bei lokalen Subsystemen und auf TSOS bei globalen Subsystemen.

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

LIBRARY = *CPLINK

Das zu definierende Subsysteme ist mit dem Basissystem des BS2000/OSD verknüpft (CP=Control Program) und muss bereits vor der Aktivierung von DSSM geladen sein.

Der Operand darf nur in Verbindung mit dem Operanden CREATION-TIME=*BEFORE-DSSM-LOAD verwendet werden.

LIBRARY = *INSTALLED(...)

Der Bibliotheksname muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Programm- oder Bindemodulbibliothek.

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

Bibliotheksname bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Vollqualifizierter Dateiname der Bindemodulbibliothek, aus der der Objectcode für das Subsystem geladen werden soll.

SUBSYSTEM-LOAD-MODE =

Bestimmt den Lademodus des Subsystems (über die BLS-DSSM-Schnittstelle \$PBBND1).

SUBSYSTEM-LOAD-MODE = *STD

Das BLS wird im STD-Run-Mode aufgerufen und lädt das Subsystem als Objektmodul.

SUBSYSTEM-LOAD-MODE = *ADVANCED

Das BLS wird im ADVANCED-Run-Mode aufgerufen und lädt das Subsystem als Binde-lademodul (LLM).

REP-FILE =

Legt fest, ob System-Reps für das zu definierende Subsystem benötigt werden und in welcher Datei diese hinterlegt sind. Diese Korrekturanweisungen werden während der Aktivierung des Subsystems ausschließlich auf die in der Bindemodulbibliothek hinterlegten und geladenen Module angewandt, nicht auf andere Subsysteme oder BS2000/OSD-CP. Eine Rep-Datei kann auch für Module eines nicht-privilegierten Subsystems vereinbart werden.

REP-FILE darf nicht zusammen mit LIBRARY=*CPLINK angegeben werden.

REP-FILE = *STD

Standardmäßig werden die System-Reps aus der Rep-Datei mit dem Namen SYSREP.<subsysname>.<subsysvers#> geladen. Diese Datei ist auf der Benutzerkennung abgelegt, unter der die Holdertask läuft; also auf der Benutzerkennung des Aufrufers bei lokalen Subsystemen und auf TSOS bei globalen Subsystemen.

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

REP-FILE = *NO

Für das Subsystem soll keine Rep-Datei verarbeitet werden.

REP-FILE = *INSTALLED(...)

Der Name der Rep-Datei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Rep-Datei.

DEFAULT-NAME =

Name der Rep-Datei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

REP-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Rep-Datei, aus der die Korrekturanweisungen gelesen werden.

REP-FILE-MANDATORY =

Legt fest, ob eine mit dem Operanden REP-FILE deklarierte Rep-Datei beim Laden des Subsystems abgearbeitet werden muss oder nicht.

REP-FILE-MANDATORY = *NO

Der Einsatz einer Rep-Datei ist nicht Pflicht, d.h. weder die Rep-Datei noch deren Einträge sollen beim Aktivieren des Subsystems geprüft werden. Sollte die Rep-Datei nicht zugreifbar oder einzelne Korrekturanweisungen fehlerhaft sein, wird das Subsystem auch in diesem Fall gestartet.

REP-FILE-MANDATORY = *YES

Sollte es bei der Bearbeitung der Rep-Datei zu folgenden Fehlern kommen, wird der Versuch, das Subsystem zu laden, abgebrochen:

- Die Rep-Datei ist nicht katalogisiert oder kann nicht gelesen werden
- Die Prüfung der Korrekturanweisungen zeigt einen Fehler an
- Der Name der Korrekturanweisungen ist fehlerhaft
- Das DMS meldet einen Fehler beim Zugriff auf die NOREF-Datei (diese Datei wird während des Ladens eines Subsystems benutzt, um zu verhindern, dass ungültige System-Reps an der Bedienstation protokolliert werden)

MESSAGE-FILE =

Bestimmt, ob es eine subsystemspezifische Meldungsdatei gibt, die beim Laden des Subsystems automatisch aktiviert wird.

Für Subsysteme, die mit dem Startzeitpunkt AT-DSSM-LOAD definiert werden, ist im Operanden RELATED-SUBSYSTEM eine Abhängigkeitsbeziehung zum Subsystem MIP zu vereinbaren.

MESSAGE-FILE = *NO

Es soll keine Meldungsdatei aktiviert werden. Dieser Wert ist verpflichtend für alle Subsysteme, die mit dem Startzeitpunkt BEFORE-DSSM-LOAD definiert werden (siehe auch Operand CREATION-TIME), da zu diesem Zeitpunkt noch keine Meldungsdatei aktiviert werden kann.

MESSAGE-FILE = *INSTALLED(...)

Der Name der Meldungsdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkomponten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkomponten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Meldungsdatei.

DEFAULT-NAME =

Name der Meldungsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

MESSAGE-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Meldungsdatei. Diese wird beim Laden des Subsystems (Kommando START-SUBSYSTEM) automatisch aktiviert, beim Entladen des Subsystems (Kommando STOP-SUBSYSTEM) automatisch deaktiviert.

SUBSYSTEM-INFO-FILE =

Bestimmt, ob eine Subsysteminformationsdatei (SSINFO) vorhanden ist. In dieser Datei sind subsystemspezifische Daten (Nebenkomponten und Konfigurationsdaten) enthalten, die nicht von DSSM zentral bearbeitet werden können.

SUBSYSTEM-INFO-FILE = *NO

Eine Informationsdatei für das Subsystem ist nicht verfügbar.

SUBSYSTEM-INFO-FILE = *INSTALLED(...)

Der Name der Informationsdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Informationsdatei.

DEFAULT-NAME =

Name der Informationsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird ein neuer Name vergeben.

SUBSYSTEM-INFO-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Informationsdatei. Der Name wird automatisch an die Aktivierungs- und Deaktivierungsroutinen (Operanden INIT/DEINIT/CLOSE-CTRL-ROUTINE) übergeben, wenn diese aufgerufen werden.

SYNTAX-FILE =

Bestimmt, ob eine Syntaxdatei mit dem Subsystem verknüpft ist, die beim Laden des Subsystems automatisch aktiviert wird. Für Subsysteme, die mit dem Startattribut MANDATORY-AT-STARTUP definiert werden, ist im Operanden RELATED-SUBSYSTEM eine Abhängigkeitsbeziehung zum Subsystem SDF zu deklarieren.

SYNTAX-FILE = *NO

Es soll keine Syntaxdatei aktiviert werden. Dieser Wert ist verpflichtend für alle Subsysteme, die mit dem Startzeitpunkt BEFORE-DSSM-LOAD oder AT-DSSM-LOAD definiert werden (siehe auch Operand CREATION-TIME), da zu diesem Zeitpunkt noch keine Syntaxdatei aktiviert werden kann.

SYNTAX-FILE = *INSTALLED(...)

Der Name der Syntaxdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Syntaxdatei.

DEFAULT-NAME =

Name der Syntaxdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

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

Vollqualifizierter Name der Syntaxdatei, die beim Laden des Subsystems automatisch aktiviert werden soll.

DYNAMIC-CHECK-ENTRY =

Vereinbart, ob eine dynamische Identitätsprüfung des Subsystems vorgenommen werden soll. Zu diesem Zweck muss eine Einsprungstelle angegeben werden, an der sowohl der Subsystemname (acht Zeichen) als auch die Versionsnummer (vier bzw. sieben Zeichen) stehen muss. DSSM prüft, ob die bei der Definition vergebene Identifikation mit dem geladenen Subsystem übereinstimmt.

DYNAMIC-CHECK-ENTRY = *STD

Es soll gelten, dass die bei dem Operanden LINK-ENTRY spezifizierte Einsprungstelle für die Identitätsprüfung herangezogen werden soll.

DYNAMIC-CHECK-ENTRY = *NO

Eine Überprüfung soll nicht stattfinden. Dieser Operandenwert darf allerdings für solche Subsysteme, die vor der Aktivierung von DSSM geladen sein sollen (CREATION-TIME= *BEFORE-DSSM-LOAD), nicht verwendet werden.

DYNAMIC-CHECK-ENTRY = <text 1..8 without-sep>

Name der Einsprungstelle, die für die Identitätsprüfung herangezogen werden soll.

CREATION-TIME =

Legt den Zeitpunkt fest, an dem die Aktivierung des Subsystems (CREATE-Routine) angestoßen wird.

Während der Systemeinführung sind zwei Phasen zu unterscheiden, in denen DSSM nach Aufruf durch die Startup-Routine die Steuerung der Systemeinführung übernimmt:

Phase 1: Der DSSM-Code wird geladen, die DSSM-Task generiert und gestartet. Die Task reserviert Klasse-5-Speicher, liest den Subsystemkatalog ein und startet die Subsysteme, die mit den Startattributen BEFORE-DSSM-LOAD und AT-DSSM-LOAD definiert wurden. Nach dem Laden dieser Subsysteme geht die Steuerung der Systemeinführung an die Startup-Routine zurück.

Phase 2: Nach erneutem Aufruf werden alle Subsysteme geladen, die mit den Startattributen MANDATORY-AT-STARTUP, BEFORE-SYSTEM-READY und AFTER-SYSTEM-READY definiert wurden.
 Bei den beiden erstgenannten wird das Laden der Subsysteme mit der Startup-Routine synchronisiert (d.h. das Laden muss abgeschlossen sein), beim letztgenannten wird das Laden asynchron angestoßen.
 Die Steuerung der Systemeinleitung geht an die Startup-Routine zurück.

Sollen verschiedene Versionen eines Subsystems definiert werden, können die Startattribute, die für die Phasen 1 und 2 der Systemeinleitung vorgesehen sind, nur für eine dieser Versionen vergeben werden.

CREATION-TIME = *AT-CREATION-REQUEST

Das Subsystem muss explizit mit dem Kommando START-SUBSYSTEM geladen werden.

CREATION-TIME = *AT-SUBSYSTEM-CALL(...)

Das Subsystem soll automatisch beim ersten SVC- oder ISL-Aufruf geladen werden. Dieser Operandenwert ist reserviert für Subsysteme, die über SVC oder ISL aufgerufen werden.

Sind mehrere Versionen eines Subsystems mit diesem Operandenwert definiert, muss für alle diese Versionen VERSION-COEXISTENCE=*ALLOWED angegeben werden sowie FUNCTION-NUMBER und FUNCTION-VERSION für ihre SVC- bzw. ISL-Einsprungstellen, die mit CONNECTION-ACCESS ungleich *SIH deklariert wurden.

Mindestens eines der angegebenen Subsysteme muss mit SUBSYSTEM-ENTRIES ..., MODE=*SVC oder *ISL deklariert worden sein (übereinstimmend mit dem Wert des Operanden ON-ACTION).

ON-ACTION =

Bestimmt, wodurch das automatische Laden des Subsystems veranlasst wird.

ON-ACTION = *STD

Voreinstellung: das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC-Einsprungstelle.

ON-ACTION = *ISL-CALL

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden ISL-Einsprungstelle.

ON-ACTION = *ANY

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC- oder ISL-Einsprungstelle.

CREATION-TIME = *AT-DSSM-LOAD

Das Subsystem soll während der Systemeinleitung (Phase 1) unter der Kontrolle der DSSM-Task geladen werden.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die ebenfalls mit diesem Startattribut definiert sind oder das Startattribut BEFORE-DSSM-LOAD besitzen.

Die Dateien für dieses Subsystem müssen unter der Benutzerkennung TSOS auf dem Home-Pubset angelegt sein, da zum Startzeitpunkt weder der Benutzerkatalog zugreifbar, noch die IMPORT-PUBSET-Verarbeitung abgeschlossen ist.

Für diese Subsysteme ist das Einbinden einer Syntaxdatei nicht zulässig.

CREATION-TIME = *BEFORE-DSSM-LOAD

Das Subsystem soll während der Systemeinleitung (Phase 1), aber nicht unter der Kontrolle der DSSM-Task geladen werden.

Solche Subsysteme sind mit dem Organisationsprogramm verknüpft und brauchen - bei der Aktivierung - nicht mit der DSSM-Task synchronisiert werden. Nach dem Laden des Subsystems läuft dieses allerdings wieder unter der Kontrolle von DSSM und kann aus Sicht des Anwenders wie andere Subsysteme gesteuert werden.

Adress- oder Abhängigkeitsbeziehungen zu Subsystemen, die mit einem anderen Startattribut definiert wurden, sind nicht möglich. Auch die Einbindung einer Meldungs- oder Syntaxdatei ist nicht zulässig. Alle Auftragseingänge (Operand SUBSYSTEM-ENTRIES) müssen deklariert sein, da DSSM die (privilegierte) Verbindung zu diesen Auftragseingängen herstellt. Es liegt in der Verantwortung des Subsystem-Entwicklers, sicherzustellen, dass zu jedem Zeitpunkt mindestens eine Version dieses Subsystems verfügbar ist.

Der Name des Link-Kontextes für diese Subsysteme muss eindeutig sein, da DSSM eine Entlade-Anforderung erfüllen muss, selbst wenn DSSM den Subsystem-Code nicht geladen hat. Eine Einsprungstelle (Operand DYNAMIC-CHECK-ENTRY) muss angegeben sein.

CREATION-TIME = *BEFORE-SYSTEM-READY

Das Subsystem soll während der Systemeinleitung (Phase 2) geladen werden. Die Aktivierung wird synchron angestoßen; die Steuerung geht erst nach dem vollständigen Laden (oder nach Lade-Fehler) an die Startup-Routine zurück, die dann „SYSTEM READY“ melden kann.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD oder MANDATORY-AT-STARTUP definiert wurden.

Die Dateien für dieses Subsystem müssen auf dem Home-Pubset katalogisiert sein.

Wird ein nicht-privilegiertes Subsystem mit diesem Operandenwert deklariert, bekommt es implizit den Wert *AFTER-SYSTEM-READY zugewiesen. SSCM gibt eine Meldung aus.

CREATION-TIME = *MANDATORY-AT-STARTUP

Das Subsystem muss während der Systemeinleitung (Phase 2) geladen werden. Die Aktivierung wird - wie bei BEFORE-SYSTEM-READY - synchron angestoßen. Im Unterschied zum oben genannten muss das Laden des Subsystems allerdings **erfolgreich** abgeschlossen werden. Andernfalls geht eine Meldung an die Startup-Routine, dass ein verpflichtendes Subsystem nicht geladen werden konnte. Die Startup-Routine entscheidet in diesem Fall, ob die Verarbeitung fortgesetzt oder abgebrochen wird.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD oder AT-DSSM-LOAD definiert wurden. Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

CREATION-TIME = *AFTER-SYSTEM-READY

Das Laden des Subsystems wird während der Systemeinleitung (Phase 2) angestoßen. Die Durchführung dieses Auftrags wird nicht mit der Startup-Routine synchronisiert, die vor dem Abschluss des Ladens „SYSTEM READY“ melden kann.

Das Subsystem darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD, MANDATORY-AT-STARTUP oder BEFORE-SYSTEM-READY definiert wurden. Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

INIT-ROUTINE =

Legt fest, ob eine Initialisierungsroutine für das Subsystem durchlaufen werden soll, wenn es gestartet oder fortgesetzt wird. In diesem Fall muss der Name einer Einsprungstelle bekannt sein und DSSM delegiert die Initialisierung an die Holdertask des betreffenden Subsystems.

Für alle Subsysteme mit dem Startattribut BEFORE-DSSM-LOAD wird die Angabe einer Einsprungstelle unbedingt empfohlen. Beim Laden des Subsystems (d.h. dem Durchlaufen der Initialisierungsroutine) erhält dieses dann Kenntnis davon, dass DSSM die Kontrolle über Anschluss und Abbau von Beziehungen übernehmen kann.

INIT-ROUTINE = *NO

Es soll keine Initialisierungsroutine durchlaufen werden.

INIT-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der Initialisierungsroutine.

Der Initialisierungsroutine wird in der Holdertask die Steuerung übergeben, damit sich das Subsystem initialisieren kann. Dazu werden ihr übergeben:

- der Name und die Version des Subsystems, wie im SSMCAT definiert
- der Name der SSINFO-Datei, falls dieser im Operanden SUBSYSTEM-INFO-FILE spezifiziert wurde
- die Adresse der beim Laden und Binden angegebenen Einsprungstelle (LINK-ENTRY)
- der vom Dynamischen Bindelader verwendete Binder-Kontext-Name
- der Name des Memory-Pools (für Subsysteme im Klasse-5- oder Klasse-6-Speicher), damit sich das Subsystem beim Nachladen eigener Selectable Units/Load Units darauf beziehen kann
- der Name der Meldungsdatei
- die Adresse des Operanden SUBSYSTEM-PARAMETER, falls im Kommando START-SUBSYSTEM ein String angegeben wird

Am Ende der Initialisierung wird eine Rückmeldung des Subsystems erwartet, ob die Initialisierung erfolgreich durchgeführt wurde und ob die Holdertask als Arbeitstask genutzt werden soll (wird in der Anweisung ASSIGN-HOLDER-TASK vereinbart, [Seite 193](#)). Abhängig davon steht die Task dann weiter unter DSSM-Kontrolle oder unter Kontrolle des Subsystems.

CLOSE-CTRL-ROUTINE =

Legt fest, ob eine Routine in das Subsystem eingebunden ist, die das Anhalten/Deaktivieren des Subsystems steuert.

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE. Beim Ansprung dieser Routine wird sichergestellt, dass noch Verbindung zum Subsystem besteht.

Existiert eine CLOSE-CTRL-Routine, tritt bei einem Versionswechsel während der BS2000-Session keine Unterbrechung auf. Es existiert zu jedem Zeitpunkt genau eine gültige Version (entweder die alte Version ist noch verfügbar oder die neue Version ist bereits verfügbar). Ohne eine solche Routine beinhaltet ein Versionswechsel immer eine Anschlussunterbrechung, während die STOPCOM-Routine der alten Version und die INIT-Routine der neuen Version ablaufen (siehe dazu auch [Seite 45](#)).

CLOSE-CTRL-ROUTINE = *NO

Im betreffenden Subsystem ist keine Routine verankert, die das Deaktivieren oder Anhalten des Subsystems steuert.

CLOSE-CTRL-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse oder FITC-Port. Die notwendigen Parameter werden vom Subsystem am Ende der INIT-Routine dynamisch an die CLOSE-CTRL-Routine übergeben und DSSM über die Identifikation der Börse bzw. FITC-Ports in Kenntnis gesetzt.

Voraussetzung für die Nutzung der CLOSE-CTRL-Routine sind die Angabe einer INIT-Routine (Operand INIT-ROUTINE) und einer STOPCOM-Routine (Operand STOPCOM-ROUTINE=*NO/*DYNAMIC).

Beim Aufruf der CLOSE-CTRL-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

CLOSE-CTRL-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

STOPCOM-ROUTINE =

Legt fest, ob eine Routine in das Subsystem eingebunden ist, die das aktive Beenden der Aufträge durchführen kann.

STOPCOM-ROUTINE = *NO

Im betreffenden Subsystem ist keine Routine verankert.

STOPCOM-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse oder FITC. Die notwendigen Parameter werden vom Subsystem am Ende der CLOSE-CTRL-Routine oder (wenn eine solche nicht vorhanden ist) am Ende der INIT-Routine dynamisch an die STOPCOM-Routine übergeben. DSSM wird über die Identifikation der Börse bzw. FITC-Ports in Kenntnis gesetzt. Voraussetzung für die Nutzung der STOPCOM-Routine ist die Angabe einer INIT-Routine (Operand INIT-ROUTINE). Beim Aufruf der STOPCOM-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (siehe ASSIGN-HOLDER-TASK, [Seite 193](#)).

STOPCOM-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

DEINIT-ROUTINE =

Legt fest, ob eine Routine in das Subsystem eingebunden ist, die die Deinitialisierung des Subsystems durchführen kann. Diese Deinitialisierungsroutine realisiert die Rückgabe der vom Subsystem angeforderten Betriebsmittel (Speicher, Dateien, Geräte).

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Wird ein Subsystem mit einer INIT- und einer CLOSE-CTRL-Routine definiert, muss eine DEINIT-Routine - mit dem gleichen Operandenwert wie die CLOSE-CTRL-Routine - angegeben werden. Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE.

Beim Ansprung dieser Routine ist sichergestellt, dass aufrufende Tasks nicht mehr an das Subsystem angeschlossen werden und alle vorhandenen Aufruf-Beziehungen gelöst werden.

DEINIT-ROUTINE = *NO

Im betreffenden Subsystem ist keine Deinitialisierungsroutine verankert, die die Rückgabe der Betriebsmittel veranlasst; diesen Part übernimmt DSSM selbst.

DEINIT-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse bzw. FITC.

Die notwendigen Parameter werden vom Subsystem am Ende der STOPCOM-Routine oder (wenn eine solche nicht vorhanden ist) am Ende der CLOSE-CTRL-Routine oder am Ende der INIT-Routine, wenn weder eine STOPCOM- noch eine CLOSE-CTRL-Routine eingebunden ist, dynamisch an die DEINIT-Routine übergeben. DSSM wird über die Identifikation der Börse bzw. des FITC-Ports in Kenntnis gesetzt.

Voraussetzung für die Nutzung der DEINIT-Routine ist die Angabe einer INIT-Routine (Operand INIT-ROUTINE). Beim Aufruf der DEINIT-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

DEINIT-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

STOP-AT-SHUTDOWN =

Legt fest, ob das Subsystem nach Beendigung der Benutzertasks bei Shutdown automatisch entladen werden soll.

STOP-AT-SHUTDOWN = *NO

Das Subsystem wird nicht automatisch entladen.

Die Angabe sollte nicht für Subsysteme verwendet werden, die Adressbeziehungen zu anderen Subsystemen besitzen, die mit STOP-AT-SHUTDOWN=*YES definiert sind.

STOP-AT-SHUTDOWN = *YES

Das Subsystem wird bei Shutdown automatisch entladen.

Diese Angabe wird ignoriert, wenn keine STOPCOM-, DEINIT- oder CLOSE-CTRL-Routine angegeben wird. SSCM gibt eine Meldung aus.

INTERFACE-VERSION =

Bezeichnet die Einsprungstelle, über die DSSM auf diejenige Schnittstellenversion zugreifen kann, die für den Aufruf der INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routinen benutzt werden soll.

INTERFACE-VERSION = *NO

Das Subsystem ruft keine INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routine auf.

INTERFACE-VERSION = <text 1..8 without-sep>

Name der Einsprungstelle. Die Einsprungstelle verweist auf den Standardheader, in dem u.a. auch die Schnittstellenversion hinterlegt ist. Der Standardheader wird durch den Aufruf des Makros \$ESMINT(I) mit MF=I/L generiert.

Dieser Operand ist Pflicht für Subsysteme, für die eine INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routine definiert wurde.

SUBSYSTEM-HOLD =

Legt fest, ob das geladene Subsystem angehalten oder entladen werden kann.

SUBSYSTEM-HOLD = *ALLOWED

Das geladene Subsystem kann angehalten und entladen werden. Die Kommandos HOLD-SUBSYSTEM und STOP-SUBSYSTEM sind für dieses Subsystem zulässig.

SUBSYSTEM-HOLD = *FORBIDDEN

Die Kommandos HOLD-SUBSYSTEM und STOP-SUBSYSTEM sind für dieses Subsystem nicht zulässig; das Subsystem wird - entsprechend den Angaben im Operanden STOP-AT-SHUTDOWN - bei Shutdown entladen. Wird das Subsystem durch Austausch mit einem anderen Subsystem entladen, so erfolgt der Austausch unterbrechungsfrei.

STATE-CHANGE-CMDS =

Legt fest, ob die DSSM-Kommandos zur Steuerung des Subsystems im laufenden Betrieb (START-SUBSYSTEM, STOP-SUBSYSTEM, HOLD-SUBSYSTEM, RESUME-SUBSYSTEM) verwendet werden dürfen.

Wird von einer Version eines Subsystems in eine andere gewechselt, wird der bei STATE-CHANGE-CMDS angegebene Wert für die auszuwechselnde Version nicht berücksichtigt.

STATE-CHANGE-CMDS = *ALLOWED

Die Kommandos dürfen an der Bedienstation und unter der privilegierten Benutzerkennung (Benutzerkennung, die mit dem Systemprivileg SUBSYSTEM-MANAGEMENT ausgestattet ist) verwendet werden.

STATE-CHANGE-CMDS = *FORBIDDEN

Die Kommandos dürfen generell nicht - weder an der Bedienstation noch unter der privilegierten Benutzerkennung - verwendet werden.

STATE-CHANGE-CMDS = *BY-ADMINISTRATOR-ONLY

Die Kommandos dürfen nur unter der privilegierten Benutzerkennung verwendet werden; für den Operator an der Bedienstation sind die Kommandos gesperrt.

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungsstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE. Beim Ansprung dieser Routine ist sichergestellt, dass aufrufende Tasks nicht mehr an das Subsystem angeschlossen werden. Tasks, die noch in Aufruf-Beziehung zum Subsystem stehen, bleiben davon unberührt.

FORCED-STATE-CHANGE =

Legt fest, ob innerhalb der Kommandos STOP-SUBSYSTEM und HOLD-SUBSYSTEM die Verwendung des Operanden FORCED=*YES zulässig ist. Mit dieser Funktion kann das unbedingte Deaktivieren des Subsystems erzwungen werden.

FORCED-STATE-CHANGE = *FORBIDDEN

Das Deaktivieren des Subsystems kann nicht erzwungen werden. DSSM weist die Verwendung des Operanden FORCED in den entsprechenden Kommandos mit einer Fehlermeldung zurück.

FORCED-STATE-CHANGE = * ALLOWED

Die Verwendung des Operanden FORCED=*YES für dieses Subsystem ist zulässig. Dieser Operandenwert darf nicht zusammen mit SUBSYSTEM-HOLD=*FORBIDDEN angegeben werden.

RESET =

Legt fest, ob innerhalb der Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM die Verwendung des Operanden RESET=*YES zulässig ist. Mit dieser Funktion kann das unbedingte Laden bzw. Fortsetzen des Subsystems erzwungen werden, auch wenn sich das Subsystem im Zustand IN-DELETE bzw. IN-HOLD befindet.

RESET = *FORBIDDEN

Das Aktivieren des Subsystems kann nicht erzwungen werden. DSSM weist die Verwendung des Operanden RESET in den entsprechenden Kommandos mit einer Fehlermeldung zurück.

RESET = *ALLOWED

Die Verwendung des Operanden RESET=*YES für dieses Subsystem ist zulässig. Dieser Operandenwert darf nicht mit SUBSYSTEM-HOLD=*FORBIDDEN angegeben werden.

RESTART-REQUIRED =

Legt fest, ob bei abnormaler Beendigung der Holdertask die Initialisierungsroutine für das Subsystem durchlaufen werden soll.

RESTART-REQUIRED = *NO

Die Initialisierungsroutine wird für einen Wiederanlauf des Subsystems nicht benutzt.

RESTART-REQUIRED = *YES

Die Initialisierungsroutine soll bei abnormaler Beendigung der Holdertask benutzt werden. Voraussetzung ist, dass die Durchführung dieser Routine im Operanden INIT-ROUTINE vorgesehen ist.

VERSION-COEXISTENCE =

Vereinbart, ob mehr als eine Version des gleichen Subsystems gleichzeitig aktiv sein darf.

VERSION-COEXISTENCE = *FORBIDDEN

Die aktuelle Version des Subsystems kann nicht gleichzeitig mit einer anderen Version des gleichen Subsystems koexistieren.

VERSION-COEXISTENCE = *ALLOWED

Die aktuelle Version des Subsystems kann gleichzeitig mit einer anderen Version des gleichen Subsystems koexistieren (Coexistence-Modus).

Bei der Definition des Auftragseingangs (Operand SUBSYSTEM-ENTRIES) darf keine indirekte Verbindung über System-Exit-Routinen gewählt werden. Sind verschiedene Versionen des gleichen Subsystems geladen, die mit dem gleichen Auftragseingang definiert wurden, wird der Anschluss immer an die höchste geladene Version des Subsystems realisiert. Greifen koexistente Subsysteme auf koexistente Syntaxdateien zu, müssen diese im SSD-Object deklariert sein und können nicht von SDF verwaltet werden.

Bei Anschlüssen über SVC und ISL ist jedoch eine Versionsauswahl über die Operanden FUNCTION-NUMBER und FUNCTION-VERSION möglich.

VERSION-EXCHANGE =

Vereinbart, ob das Laden der aktuellen Subsystemversion im Exchange-Modus erlaubt ist. Der Exchange-Modus erlaubt die temporäre Koexistenz zweier Versionen des gleichen Subsystems. Wird die Version B eines Subsystems geladen, wenn bereits Version A des Subsystems aktiv ist, werden alle neuen Aufrufer an die Version B angeschlossen. Die Aufträge, die an die Version A angeschlossen sind, werden noch bearbeitet. Nach Bearbeitung aller Aufträge an Version A wird diese automatisch beendet.

Bei der Definition ist darauf zu achten, dass die zu ersetzende, „alte“ Version nicht abhängig von der ersetzenden, „neuen“ Version sein darf.

VERSION-EXCHANGE = *FORBIDDEN

Die aktuelle Version des Subsystems darf nicht ausgetauscht werden.

VERSION-EXCHANGE = *ALLOWED

Der Exchange-Modus, der die temporäre Koexistenz zweier Subsysteme erlaubt, ist für die aktuelle Subsystemversion zulässig.

ADD-SUBS-ENTRIES / MODIFY-SUBS-ENTRIES =

Bezeichnet, ob neue Auftragseingänge definiert (ADD) oder die Charakteristiken vorhandener Auftragseingänge verändert (MODIFY) werden sollen.

ADD-SUBS-ENTRIES / MODIFY-SUBS-ENTRIES = *NONE

Voreinstellung: es sollen weder neue Auftragseingänge hinzukommen, noch die Eigenschaften vorhandener Auftragseingänge modifiziert werden.

ADD-SUBS-ENTRIES / MODIFY-SUBS-ENTRIES = list-poss(100): <text 1..8>

Vereinbart durch Angabe des Namens der Einsprungstelle entweder maximal 100 neue Auftragseingänge für das Subsystem, deren Typ jeweils in den Unterstrukturen definiert werden muss (ADD), oder modifiziert bereits definierte Auftragseingänge (MODIFY).

MODE =

Legt den Typ eines vereinbarten Auftragseingangs für das Subsystem fest.

Wird der Typ des vereinbarten Auftragseingangs verändert, müssen alle Unteroperanden von MODE explizit mit einem Wert versehen werden; der Operandenwert *UNCHANGED (Voreinstellung bei MODIFY-SUBS-ENTRIES) wird abgewiesen.

MODE = *LINK

Der Auftragseingang kann nicht über indirektes Linkage erreicht werden, sondern nur über eine CONNECT-Beziehung mittels eines externen Binder-Symbols.

Bei verschiedenen Versionen des gleichen Subsystems, die das gleiche externe Binder-Symbol nutzen, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

MODE = *ISL(...)

Der Auftragseingang wird durch indirekte Verbindung über System Procedure Linkage (nur für privilegierte Subsysteme) erreicht. Wird zusätzlich noch eine Funktions- und Versionsnummer der ISL-Einsprungstelle spezifiziert, darf sich die Kombination aus Name der Einsprungstelle, Funktions- und Versionsnummer nicht mit einer anderen Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEEXISTENCE=*ALLOWED) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über die selbe ISL-Einsprungstelle erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die die selbe ISL-Einsprungstelle nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die die selbe ISL-Einsprungstelle nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standard Header der Parameterliste des Aufrufers hinterlegt ist. Die Angabe CONNECTION-ACCESS=*ALL ist für ISL-Einsprungstellen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der ISL-Einsprungstelle angesprochen werden soll, da die gleiche ISL-Einsprungstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der ISL-Einsprungstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten ISL-Funktionsnummer.

MODE = *SVC(...)

Der Auftragseingang wird durch indirekte Verbindung über Supervisor Call (SVC) erreicht.

Wird zusätzlich noch eine Funktions- und Versionsnummer der SVC-Einsprungstelle spezifiziert, darf sich die Kombination aus SVC-Nummer, Funktions- und Versionsnummer nicht mit einer anderen Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEXISTENCE=*ALLOWED) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über den selben SVC erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standard Header der Parameterliste des Aufrufers hinterlegt ist.

NUMBER = <integer 0..255>

Nummer des SVCs, über den der Auftragseingang erreicht wird. In Verbindung mit CONNECTION-ACCESS=*ALL ist die Verwendung einer SVC-Nummer größer 191 nicht zulässig.

CALL-BY-SYSTEM-EXIT =

Legt fest, ob die angegebene SVC-Nummer von System-Exit-Routinen aus aufgerufen werden darf.

CALL-BY-SYSTEM-EXIT = *ALLOWED

Der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen zulässig.

CALL-BY-SYSTEM-EXIT = *FORBIDDEN

Der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der SVC-Einsprungstelle angesprochen werden soll, da die gleiche SVC-Einsprungstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der SVC-Einsprungstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten SVC-Funktionsnummer.

MODE = SYSTEM-EXIT(...)

Der Auftragseingang wird durch indirekte Verbindung über System-Exit-Routinen erreicht.

In Verbindung mit CONNECTION-ACCESS=*ALL ist die Verwendung dieses Operanden nicht zulässig.

NUMBER = <integer 0..127>

Nummer der System-Exit-Routine.

CONNECTION-ACCESS =

Vereinbart die Zugriffsberechtigung (Privilegierung) für das Subsystem.

CONNECTION-ACCESS = *ALL

Privilegierte und nicht-privilegierte Programmläufe sind zugriffsberechtigt.

In Verbindung mit MODE=*SYSTEM-EXIT/*ISL/*SVC (mit einer SVC-Nummer größer 191) ist die Verwendung dieses Operandenwertes nicht zulässig.

CONNECTION-ACCESS = *SYSTEM

Nur privilegierte Programmläufe sind zugriffsberechtigt.

CONNECTION-ACCESS = *SIH

Nur Tasks, die im Funktionszustand SIH laufen, sind zugriffsberechtigt.

Das aufgerufene Subsystem läuft ebenfalls im Funktionszustand SIH, d.h. es ist nicht unterbrechbar.

Die Angabe dieses Operandenwertes ist nur für Subsysteme zulässig, deren Auftrags-
eingang definiert wird über

- System Procedure Linkage (MODE=*ISL(FUNCTION-NUMBER=*NONE))
- CONNECTION-SCOPE=*OPTIMAL
- MEMORY-CLASS=*SYSTEM-GLOBAL(SUBSYSTEM-ACCESS=*SYSTEM)

CONNECTION-SCOPE =

Bezeichnet das Ereignis, das die automatische Auflösung des Anschlusses an den angegebenen Subsystem-Auftragseingang hervorruft.

CONNECTION-SCOPE = *TASK

Der Anschluss wird bei Taskbeendigung aufgehoben.

CONNECTION-SCOPE = *PROGRAM

Der Anschluss wird spätestens bei Programmbeendigung aufgehoben.

Zusammen mit MEMORY-CLASS=*LOCAL-UNPRIVILEGED darf nur CONNECTION-
SCOPE=*PROGRAM angegeben werden.

Die Angabe dieses Operandenwertes wird für Subsysteme empfohlen, die mit
SUBSYSTEM-ACCESS=*LOW/*HIGH oder MEMORY-CLASS=*BY-SLICE deklariert wurden.

CONNECTION-SCOPE = *FREE

DSSM soll keine Kontrolle von Anschlüssen zu diesen Auftragseingängen durchführen. Der Anschluss wird - außer bei einer expliziten Anforderung - nicht automatisch aufgelöst. Um Probleme oder mögliche Fehler beim Entladen des Subsystems zu vermeiden, müssen die Anschlüsse vom Subsystem selbst verwaltet werden.

CONNECTION-SCOPE = *CALL

Nach Rückkehr aus diesem Auftragseingang führt DSSM automatisch die Auflösung der Anschlüsse durch.

Dieser Operandenwert steht nur für Subsysteme zur Verfügung, deren Auftragseingang über System Procedure Linkage (ISL) und Supervisor Call (SVC) definiert wird.

CONNECTION-SCOPE = *OPTIMAL

Das Subsystem wird deaktiviert bzw. angehalten, wenn keine Task mehr Anschluss zu diesem Auftragseingang hat.

Eine Routine, deren Einsprungstelle mit *OPTIMAL definiert wird, muss mit RETURN beendet werden.

Wird ein Auftragseingang eines Subsystems mit CONNECTION-SCOPE=*OPTIMAL definiert, müssen alle seine Auftragseingänge im Subsystemkatalog mit MODE≠*LINK definiert werden.

Während ein Subsystem deaktiviert oder angehalten wird, wird kein Aufruf des Subsystems mit CONNECTION-SCOPE=*OPTIMAL akzeptiert.

FIRST-CONNECTION =

Bestimmt, ob der Erst-Anschluss der Task an den angegebenen Auftragseingang des Subsystems erlaubt ist. Mindestens ein Auftragseingang eines Subsystems muss mit FIRST-CONNECTION=*ALLOWED definiert werden.

FIRST-CONNECTION = *ALLOWED

Der Erst-Anschluss an den angegebenen Auftragseingang ist erlaubt.

Für die Anweisung ADD-SUBS-ENTRIES ist dieser Wert die Voreinstellung.

FIRST-CONNECTION = *FORBIDDEN

Der Anschluss an den angegebenen Auftragseingang über SVC oder ISL ist nicht erlaubt, wenn die Task nicht bereits an einen anderen Auftragseingang des Subsystems angeschlossen ist.

Die Angabe dieses Operandenwertes ist für Auftragseingänge, die mit MODE=*LINK/*SYSTEM-EXIT oder CONNECTION-ACCESS=*SIH definiert wurden, nicht erlaubt.

MODIFY-SUBS-ENTRIES = *NONE / list-poss(100): <text 1..8> / *BY-PROGRAM(...)

Die Werte *NONE und list-poss(100): <text 1..8> sind beim Operanden ADD-SUBS-ENTRIES auf [Seite 220](#) beschrieben.

MODIFY-SUBS-ENTRIES = *BY-PROGRAM(...)

Die Einsprungstellen des angegebenen Subsystems werden nicht statisch aus dem Katalog versorgt, sondern dynamisch zum Ladezeitpunkt aus dem BLS-Namensverzeichnis. Voraussetzung für diese Funktionalität ist der Einsatz von BLSSERV ab Version 2.1, das den Einsatz von EEN-Namen als Einsprungstellen für DSSM-Subsysteme unterstützt.

Die Anweisung wird zurückgewiesen, wenn das Subsystem nicht zuvor bei der Anweisung SET-SUBSYSTEM-ATTRIBUTES auch mit *BY-PROGRAM definiert wurde. MODIFY-SUBS-ENTRIES dient nur zum Ändern der Anschluss-Einstellungen.

Wird *BY-PROGRAM verwendet, müssen die Operanden ADD-SUBS-ENTRIES und REMOVE-SUBS-ENTRIES mit *NONE angegeben werden.

CONNECTION-SCOPE = *TASK / *PROGRAM

Der Anschluss wird bei Task- bzw. Programmbeendigung aufgehoben.

REMOVE-SUBS-ENTRIES =

Bestimmt, ob vorhandene Auftragseingänge, die für das Subsystem vereinbart sind, gelöscht werden sollen.

REMOVE-SUBS-ENTRIES = *NONE

Voreinstellung: Es sollen keine Auftragseingänge gelöscht werden.

REMOVE-SUBS-ENTRIES = list-poss(100): <text 1..8>

Gibt die Namen von maximal 100 Auftragseingängen an, die für das Subsystem nicht mehr gelten sollen.

MEMORY-CLASS =

Vereinbart den subsystemspezifischen Adressraum, in den das Subsystem geladen werden soll. Mit diesem Operanden kann die Systembetreuung die Adressraumvorgaben, die für das jeweilige Subsystem gelten, den speziellen Anforderungen der Installation anpassen.

Wird der subsystemspezifische Adressraum verändert, müssen alle Unteroperanden von MEMORY explizit mit einem Wert versehen werden (der Operandenwert *UNCHANGED (Voreinstellung) wird abgewiesen).

MEMORY-CLASS = *SYSTEM-GLOBAL(...)

Das Subsystem wird in den Klasse-3- oder Klasse-4-Speicher geladen. Residente CSECTs erhalten Klasse-3-Speicher, alle anderen erhalten seitenwechselbaren Klasse-4-Speicher.

SUBSYSTEM-ACCESS =

Bezeichnet die Zugriffsrechte (Privilegierung) und die Lage des angeforderten Speichers.

SUBSYSTEM-ACCESS = *LOW

Es wird nicht-privilegierter Adressraum unterhalb der 16 MByte-Grenze zugewiesen.

SUBSYSTEM-ACCESS = *SYSTEM

Subsysteme, die mit diesem Operandenwert deklariert werden, sind privilegierte Subsysteme, denen privilegierter Adressraum oberhalb der 16 MByte-Grenze zugewiesen wird.

Die Angabe dieses Operandenwertes ist für die Subsysteme verpflichtend, deren Auftragseingang über SVC (MODE=*SVC), oder für die eine INIT-, STOPCOM-, DEINIT- oder CLOSE-CTRL-Routine vereinbart wird.

Der Operandenwert ist unzulässig in Kombination mit CONNECTION-ACCESS=*ALL und MODE=*LINK.

SUBSYSTEM-ACCESS = *HIGH

Es wird nicht-privilegierter Adressraum bis zu 2 GByte zugewiesen.

MEMORY-CLASS = *LOCAL-PRIVILEGED(...)

Das Subsystem erhält einen Memory Pool im nicht-privilegierten Klasse-5-Speicher, der unterhalb der 16 MByte-Grenze angelegt wird.

Die Angabe ist auf nicht-privilegierte Subsysteme zugeschnitten, die relativ viel Adressraum (+/- 1 MByte) beanspruchen und unterhalb der 16 MByte-Grenze anzulegen sind. Die Subsysteme werden in Memory Pools an der gleichen Adresse geladen, um mit dem knapp bemessenen Adressraum unterhalb 16 MByte zu haushalten.

Obwohl solche Subsysteme parallel im gleichen Adressraum geladen werden, können sie nicht simultan von einer Task genutzt werden (siehe auch Anweisung SEPARATE-ADDRESS-SPACE, [Seite 241](#)).

Das Subsystem darf keine residenten CSECTs enthalten, da ansonsten ein späteres Aktivieren abgebrochen wird.

SIZE = <integer 1..32767>

Größe des benötigten Adressraums (in 4KByte-Seiten) für den Memory Pool im Klasse-5-Speicher. Der Wert ist mindestens so groß zu wählen, dass das Subsystem und evtl. von diesem nachgeladene Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

MEMORY-CLASS = *LOCAL-UNPRIVILEGED(...)

Das Subsystem erhält einen Memory Pool im nicht-privilegierten Klasse-6-Speicher. Die Angabe ist für Subsysteme reserviert, die wie ein Programm ausführbar sind. Analog dazu muss deren Zugriffsberechtigung (Privilegierung) im Operanden CONNECTION-ACCESS mit dem Wert *ALL definiert werden.

Dieser Operandenwert darf nicht zusammen mit einer Einsprungstelle angegeben werden, die mit CONNECTION-ACCESS=*SYSTEM definiert wurde.

Das Subsystem darf keine residenten CSECTs enthalten, da ansonsten ein späteres Aktivieren abgebrochen wird.

Wird dieser Operandenwert angegeben, ist zur Auflösung des Anschlusses an den angegebenen Subsystem-Auftragseingang nur CONNECTION-SCOPE=*PROGRAM erlaubt.

SIZE = <integer 1..32767>

Größe des benötigten Adressraums (in 4KByte-Seiten) für den Memory Pool im Klasse-6-Speicher. Der Wert ist mindestens so groß zu wählen, dass das Subsystem und evtl. von diesem nachgeladene Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

SUBSYSTEM-ACCESS =

Bezeichnet die Lage des angeforderten Speichers.

SUBSYSTEM-ACCESS = *LOW

Es wird nicht-privilegierter Adressraum unterhalb der 16 MByte-Grenze zugewiesen. Da diese Angabe auf Subsysteme zugeschnitten ist, die wie Programme ausführbar sind, ist die zusätzliche Angabe CONNECTION-SCOPE=*PROGRAM anzuraten.

SUBSYSTEM-ACCESS = *HIGH

Es wird nicht-privilegierter Adressraum bis zu 2 GByte zugewiesen. Da diese Angabe auf Subsysteme zugeschnitten ist, die wie Programme ausführbar sind, ist die zusätzliche Angabe CONNECTION-SCOPE=*PROGRAM anzuraten.

START-ADDRESS =

Legt die Anfangsadresse im Klasse-6-Speicher fest.

START-ADDRESS = *ANY

Die Lage des Subsystems im Klasse-6-Speicher wird von DSSM festgelegt.

START-ADDRESS = <x-string 7..8>

Start-Adresse im Segment-Raster, an der die Anfangsadresse des Subsystems liegen soll. Als Wert ist eine 8-stellige Hexadezimalkonstante anzugeben, die ein Vielfaches von X'100000' sein muss.

MEMORY-CLASS = *BY-SLICE(...)

Das angegebene Subsystem ist ein nicht-privilegiertes Subsystem und besteht aus einem LLM, das aus einem mehrbenutzbaren Code (Programmbereich) und einem nicht-mehrbenutzbaren Code (Datenbereich) besteht.

Der Programmbereich wird in den gemeinsam benutzbaren Adressraum geladen (das entspricht MEMORY-CLASS=*SYSTEM-GLOBAL). Der Datenbereich wird in den Benutzeradressraum der Holdertask geladen und in die privaten Benutzeradressräume der angeschlossenen Tasks an die selbe Adresse kopiert.

Gemeinsam mit MEMORY-CLASS=*BY-SLICE müssen folgende Werte angegeben werden: SUBSYSTEM-LOAD-MODE=*ADVANCED und CONNECTION-ACCESS=*ALL.

SIZE = <integer 1..32767>

Größe des angeforderten Speicherplatzes für den Datenbereich in 4K-Seiten. Der Wert ist mindestens so groß zu wählen, dass der Datenbereich und evtl. vom Subsystem in den Benutzeradressraum nachzuladende Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

LINK-ENTRY = <text 1..8 without-sep>(…)

Definiert den Namen des zum Laden benötigten Bindemoduls/ENTRY/CSECT (als Operand im Makroaufruf \$PBBND1 an den dynamischen Bindelader DBL). Das Subsystem muss von diesem ENTRY vollständig (ggf. per Autolink) geladen werden.

AUTOLINK =

Steuert den Aufruf der Autolink-Funktion beim Binden und Laden.

Die Autolink-Funktion des Binders ermöglicht das automatische Einfügen von Modulen, die nicht mit entsprechenden Anweisungen explizit eingefügt werden. Die Funktion erspart vor allem den Benutzern der höheren Programmiersprachen, die zahlreiche benötigten Module des Laufzeitsystems (= Run Time System) mit expliziten Anweisungen einzeln einzufügen. Ein nähere Beschreibung der Autolink-Funktion ist im Handbuch „BLSSERV“ [4] zu finden.

Die Autolink-Funktion kann auch implizit umgangen werden, wenn während des Bindens des zu ladenden Objektmoduls der erste Externverweis auf ein vorgebundenes Großmodul zielt. Der Vorteil dieser Vorgehensweise ergibt sich daraus, dass das Seitenwechselverhalten bei der späteren Ausführung bereits im Vorfeld (während des Bindens) optimiert werden kann. Zudem können Fehler während des Bindens auf diese Weise vermieden werden.

AUTOLINK = *ALLOWED

Die Autolink-Funktion wird zugelassen.

AUTOLINK = *FORBIDDEN

Die Autolink-Funktion wird unterdrückt.

ADD-REFER-SUBS / MODIFY-REFER-SUBS =

Legt fest, ob entweder eine Liste von Subsystemen anzulegen ist, zu denen Adressbeziehungen bestehen und die zur Auflösung von Externverweisen benutzt werden sollen (ADD), oder ob eine bereits bestehende Liste existiert, die zu verändern ist (MODIFY).

ADD-REFER-SUBS / MODIFY-REFER-SUBS = *NONE

Voreinstellung: es soll weder eine Liste angelegt, noch eine bereits bestehende Liste verändert werden.

ADD-REFER-SUBS / MODIFY-REFER-SUBS = list-poss(15):<structured-name 1..8>

Es sollen Externverweise angegeben (ADD) bzw. modifiziert (MODIFY) werden.

Es können Externverweise auf maximal 15 andere Subsysteme benannt werden, die zur Auflösung dieser Externverweise benutzt werden müssen. Fehlt eines der hier genannten Subsysteme beim Aktivieren oder Deaktivieren (und ist gleichzeitig eine Überprüfung der Externverweise mit dem Operanden CHECK-REFERENCE=*YES angefordert), wird die Aktion abgebrochen.

Auch das „Basis-Subsystem“ des BS2000/OSD, das Control Program, kann über diese Externverweise - mit dem Namen CP - angesprochen werden. Es kann in der folgenden Unterstruktur sowohl genau eine Version spezifiziert werden, als auch ein Bereich von Versionen angegeben werden, innerhalb dessen alle Versionen herangezogen werden sollen.

Wird die Version des Subsystems CP über eine Versionsbereichsliste eingegrenzt, prüft DSSM die Kompatibilität zwischen der aktuellen CP-Version und Versionen in der Bereichsliste. Nur wenn die Versionen kompatibel sind, wird das Subsystem geladen.

Auf folgende Einschränkungen bei der Angabe von Subsystemen, zu denen Adressbeziehungen bestehen, ist zu achten:

- Es dürfen keine Adressbeziehungen zu Subsystemen vereinbart werden, die das Attribut MEMORY-CLASS=*LOCAL-PRIVILEGED/*LOCAL-UNPRIVILEGED/*BY-SLICE besitzen.
- Ist für das zu definierende Subsystem das Attribut SUBSYSTEM-ACCESS=*SYSTEM vergeben, dürfen keine Subsysteme angesprochen werden, die mit SUBSYSTEM-ACCESS=*LOW bzw. *HIGH definiert sind.
- Subsysteme, die das Attribut STOP-AT-SHUTDOWN=*YES besitzen, dürfen Adressbeziehungen nur zu solchen Subsystemen aufweisen, die ebenfalls dieses Attribut besitzen.
- Ein nicht-privilegiertes Subsystem darf generell keine Adressbeziehung zum Control Program CP aufweisen.
- Wird auf ein Subsystem verwiesen, das zumindest eine Version besitzt, die im Coexistence- oder Exchange-Modus betrieben werden darf, ist die eindeutige Version anzugeben.
- Die Adressbeziehungen müssen in Abhängigkeit der Startattribute (Operand CREATION-TIME) definiert werden, d.h. Subsysteme dürfen Beziehungen nur zu solchen Subsystemen aufnehmen, deren Start zeitgleich oder früher erfolgt.

LOWEST-VERSION =

Spezifiziert den unteren Wert (niedrigste Version) innerhalb der Versionsbereichsliste der Subsysteme.

LOWEST-VERSION = *LOWEST-EXISTING

Die niedrigste Version im Katalog soll angesprochen werden.

LOWEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Untergrenze des Versionsbereiches fungieren soll.

HIGHEST-VERSION =

Spezifiziert den oberen Wert (höchste Version) innerhalb der Versionsbereichsliste der Subsysteme.

HIGHEST-VERSION = *HIGHEST-EXISTING

Die höchste Version im Katalog soll angesprochen werden.

HIGHEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Obergrenze des Versionsbereiches fungieren soll.

MODIFY-REFER-SUBS =

Siehe Beschreibung bei ADD-REFER-SUBS.

REMOVE-REFER-SUBS =

Bestimmt, ob bestehende Externverweise auf andere Subsysteme gelöscht werden sollen.

REMOVE-REFER-SUBS = *NONE

Voreinstellung: Es sollen keine bestehenden Externverweise auf andere Subsysteme gelöscht werden.

REMOVE-REFER-SUBS = list-poss(15): <structured-name 1..8>

Gibt die Namen von maximal 15 Externverweisen an, die nicht mehr gültig sein sollen.

UNRESOLVED-EXTERNALS =

Vereinbart das Verhalten des Ladevorgangs, wenn Externverweise nicht aufgelöst werden können.

UNRESOLVED-EXTERNALS = *FORBIDDEN

Der Ladevorgang wird bei nicht auflösbaren Externverweisen abgebrochen.

UNRESOLVED-EXTERNALS = *ALLOWED

Der Ladevorgang wird fortgesetzt, nicht auflösbare Externverweise werden mit dem Wert 'X'FFFFFFFF' besetzt.

CHECK-REFERENCE =

Vereinbart, ob die im Operanden REFERENCED-SUBSYSTEM angegebenen Subsysteme im Hinblick auf deren Zustand und Verfügbarkeit hin überprüft werden sollen.

CHECK-REFERENCE = *YES

Die Referenz-Subsysteme werden überprüft. Fehlt eines der Subsysteme, bricht DSSM das Aktivieren oder Entladen des Subsystems ab.

CHECK-REFERENCE = *NO

DSSM soll keine Prüfung vornehmen.

Generiert der Anwender allerdings mit dieser Anweisung auch komplexe Subsysteme, führt DSSM die geforderten Funktionen **trotz** möglicher Konflikte durch:

- Das Kommando START-SUBSYSTEM lädt das angegebene Subsystem, auch wenn ein Subsystem, zu dem definierte Beziehungen bestehen, noch nicht vollständig geladen ist.
- Die Kommandos RESUME-SUBSYSTEM, STOP-SUBSYSTEM und HOLD-SUBSYSTEM werden ohne Prüfung von Beziehungen und Abhängigkeiten von DSSM ausgeführt.

ADD-RELATED-SUBS / MODIFY-RELATED-SUBS =

Legt fest, ob entweder eine Liste von Subsystemen aufgebaut werden soll, zu denen Abhängigkeitsbeziehungen bestehen (ADD), oder eine bereits bestehende Liste abhängiger Subsysteme modifiziert werden soll (MODIFY).

ADD-RELATED-SUBS / MODIFY-RELATED-SUBS = *NONE

Voreinstellung: es sollen für das Subsystem weder Abhängigkeitsbeziehungen definiert noch verändert werden.

ADD-RELATED-SUBS / MODIFY-RELATED-SUBS = list-poss(100):**<structured-name 1..8>**

Es sollen Abhängigkeitsbeziehungen zu maximal 100 anderen Subsystemen, ohne die das zu definierende Subsystem nicht funktionsfähig ist, definiert (ADD) oder verändert (MODIFY) werden.

Abhängigkeitsbeziehungen dürfen auch zum BS2000/OSD Control Program CP definiert werden. Die Regeln und Einschränkungen, die hierbei zu beachten sind, gelten analog für Adressbeziehungen und sind dort näher erläutert (siehe Operand ADD-REFER-SUBS). Die Abhängigkeitsbeziehungen zielen jeweils auf eine Version eines Subsystems. In der folgenden Unterstruktur kann sowohl genau eine Version spezifiziert werden, als auch ein Bereich von Versionen angegeben werden, innerhalb dessen alle Versionen herangezogen werden sollen.

Auf folgende Einschränkungen ist allgemein bei der Angabe von abhängigen Subsystemen ist zu achten:

- Die definierte Beziehung muss zyklensfrei sein. Ein Zyklus entsteht, wenn Subsystem A abhängig von B, B abhängig von C und dieses wieder abhängig von A ist.
- Ist für das zu definierende Subsystem das Attribut MEMORY-CLASS=*SYSTEM-GLOBAL vergeben, dürfen keine Subsysteme angesprochen werden, die mit MEMORY-CLASS=*LOCAL-PRIVILEGED oder *LOCAL-UNPRIVILEGED definiert sind.
- Für Subsysteme, die das Attribut SUBSYSTEM-ACCESS=*SYSTEM besitzen, darf keine Abhängigkeitsbeziehung zu Subsystemen definiert werden, für die SUBSYSTEM-ACCESS=*LOW bzw. SUBSYSTEM-ACCESS=*HIGH oder MEMORY-CLASS=*BY-SLICE gilt.
- Die Abhängigkeitsbeziehungen müssen entsprechend den Startattributen (Operand CREATION-TIME) definiert werden, d.h. ein Subsystem darf nur von solchen Subsystemen abhängen, deren Start zeitgleich oder früher erfolgt.

LOWEST-VERSION =

Spezifiziert den unteren Wert (niedrigste Version) innerhalb der Versionsbereichsliste der Subsysteme.

LOWEST-VERSION = *LOWEST-EXISTING

Die niedrigste Version im Katalog soll angesprochen werden.

LOWEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Untergrenze des Versionsbereiches fungieren soll.

HIGHEST-VERSION =

Spezifiziert den oberen Wert (höchste Version) innerhalb der Versionsbereichsliste der Subsysteme.

HIGHEST-VERSION = *HIGHEST-EXISTING

Die höchste Version im Katalog soll angesprochen werden.

HIGHEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Obergrenze des Versionsbereiches fungieren soll.

MODIFY-RELATED-SUBS =

Siehe Beschreibung bei ADD-RELATED-SUBS.

REMOVE-RELATED-SUBS =

Bestimmt, ob bestehende Abhängigkeitsbeziehungen zu anderen Subsystemen gelöscht werden sollen.

REMOVE-RELATED-SUBS = *NONE

Voreinstellung: Es sollen keine bestehenden Abhängigkeitsbeziehungen zu anderen Subsystemen gelöscht werden.

REMOVE-RELATED-SUBS = list-poss(100): <structured-name 1..8>

Gibt die Namen von maximal 100 Subsystemen an, zu denen keine Abhängigkeitsbeziehungen mehr bestehen sollen.

MODIFY-WORK-TASK-ATTRIBUTE

Arbeitstask-Parameter verändern

Funktionsbeschreibung

Mit dieser Anweisung können Subsysteme angegeben werden, die ihre Holdertask nicht mehr als Arbeitstask nutzen sollen. Zudem kann eine neue TSN angegeben werden, die die Holdertask erhalten soll. Die Anweisung wird zurückgewiesen, wenn das angegebene Subsystem nicht existiert oder die Holdertask nicht als Arbeitstask nutzt.

MODIFY-WORK-TASK-ATTRIBUTE wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Format

MODIFY-WORK-TASK-ATTRIBUTE
<p>SUBSYSTEM-NAME = <structured-name 1..8> ,SUBSYSTEM-VERSION = <c-string 3..8> / <text 3..8> ,WORK-TASK = <u>*UNCHANGED</u> (...) / *NO *UNCHANGED(...) TSN = <u>*BY-DSSM</u> / <alphanum-name 1..4></p>

Operandenbeschreibung

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

Name des Subsystems, das die Holdertask als Arbeitstask nutzt.

SUBSYSTEM-VERSION = <c-string 3..8> / <text 3..8>

Version des Subsystems, das die Holdertask als Arbeitstask nutzt.
 Die Version muss bereits deklariert sein.

WORK-TASK =

Legt fest, ob die Holdertask nicht mehr als Arbeitstask genutzt werden soll oder ob lediglich die TSN für die Holdertask neu vergeben wird.

WORK-TASK = *UNCHANGED

Das Subsystem soll die Holdertask nach wie vor als Arbeitstask benutzen; in der folgenden Unterstruktur kann jedoch eine neue TSN für die Holdertask vergeben werden.

TSN =

Bestimmt, welche Task Sequence Number (TSN) die Holdertask des Subsystems erhalten soll.

TSN = *BY-DSSM

Voreinstellung: die TSN wird beim Laden der Arbeitstask von DSSM vergeben.

TSN = <alphanum-name 1..4>

TSN, die die Holdertask beim Starten erhalten soll.

Die angegebene TSN muss eindeutig festgelegt und verwendbar sein, wenn das Subsystem geladen wird.

WORK-TASK = *NO

Die Holdertask des angegebenen Subsystems soll nicht mehr als Arbeitstask verwendet werden.

REMOVE-ADDR-SPACE-SEPARATION

Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher aufheben

Funktionsbeschreibung

Mit dieser Anweisung können die Vereinbarungen, die mit der Anweisung SEPARATE-ADDRESS-SPACE (siehe [Seite 241](#)) bezüglich der disjunkten Verteilung der Subsysteme im Klasse-5-Speicher getroffen wurden, aufgehoben werden.

Die Angabe ist für Subsysteme im Klasse-3- oder Klasse-4-Speicher (Operand MEMORY-CLASS=*SYSTEM-GLOBAL in der Anweisung SET-SUBSYSTEM-ATTRIBUTES) sowie für Subsysteme im Klasse-6-Speicher (MEMORY-CLASS=*LOCAL-UNPRIVILEGED) nicht relevant. Subsysteme im Klasse-6-Speicher werden nie parallel gebraucht, dürfen sich also damit im Adressraum überschneiden, und Subsysteme im Klasse-3- oder Klasse-4-Speicher überschneiden sich grundsätzlich nicht.

REMOVE-ADDR-SPACE-SEPARATION wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Der Name des Subsystems muss im Katalog bekannt sein. Besteht eine in der Liste der Subsysteme angegebene Adressbeziehung nicht, wird die Angabe ignoriert; die Verarbeitung der Anweisung wird fortgesetzt.

Format

REMOVE-ADDR-SPACE-SEPARATION

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

,FROM-SUBSYSTEMS = list-poss(15): <structured-name 1..8>

Operandenbeschreibung

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

Name des Subsystems, das sich vormals nicht mit anderen Subsystemen überschneiden durfte.

Das Subsystem muss SSCM bekannt und bereits definiert sein.

FROM-SUBSYSTEMS = list-poss(15): <structured-name 1..8>

Liste von maximal 15 Subsystemen, für die die Vereinbarung besteht, dass sie sich nicht mit dem im Operanden SUBSYSTEM-NAME angeführten Subsystem überschneiden dürfen. Die Vereinbarung wird für diese Subsysteme aufgehoben.

REMOVE-CATALOG-ENTRY

Subsystemdefinition logisch aus dem Subsystemkatalog löschen

Funktionsbeschreibung

Mit dieser Anweisung wird die Definition eines Subsystems, die im Subsystemkatalog hinterlegt ist, logisch gelöscht.

Die Anweisung wird zurückgewiesen, wenn kein aktueller (nicht leerer) Katalog, in dem die Definition hinterlegt ist, mit der Anweisung START-CATALOG-CREATION bzw. START-CATALOG-MODIFICATION eröffnet wurde.

Ist das angegebene Subsystem im bezeichneten Katalog nicht enthalten, wird die Anweisung ebenfalls zurückgewiesen. Wird das Subsystem aus dem Katalog entfernt, werden auch die disjunkten Beziehungen (die mit der Anweisung SEPARATE-ADDRESS-SPACE vereinbart wurden) gelöscht.



Bei bestehenden Abhängigkeits- oder Ladebeziehungen kann das Entfernen eines Subsystems zu Fehlern bei der Konsistenzprüfung des Kataloges führen!

Format

REMOVE-CATALOG-ENTRY

SUBSYSTEM-NAME = <structured-name 1..8>(…)

<structured-name>(…)

| **VERSION** = *ALL / <c-string 3..8> / <text 3..8>

Operandenbeschreibung

SUBSYSTEM-NAME = <structured-name 1..8>(…)

Name des Subsystems, dessen Definition aus dem Katalog gelöscht werden soll.

VERSION = *ALL / <c-string 3..8> / <text 3..8>

Spezifiziert die Version des Subsystems, dessen Definition aus dem Katalog gelöscht werden soll. Mit der Angabe VERSION=*ALL können die Definitionen aller Versionen des angegebenen Subsystems aus dem Katalog gelöscht werden.

SAVE-CATALOG

Subsystemkatalog als PAM-Datei abspeichern

Funktionsbeschreibung

Mit dieser Anweisung wird der durch zuvor eingegebene Anweisungen im Speicher aufgebaute Subsystemkatalog als PAM-Datei abgespeichert.

Der Dateiname wird entweder aus der letzten Anweisung START-CATALOG-CREATION (siehe [Seite 286](#)) bzw. der Anweisung START-CATALOG-MODIFICATION (siehe [Seite 287](#)) abgeleitet, oder es kann ein neuer, vollqualifizierter Dateiname angegeben werden.

SAVE-CATALOG wird mit Fehlermeldung abgewiesen, wenn:

- vorher keine der Anweisungen START-CATALOG-CREATION oder START-CATALOG-MODIFICATION ausgeführt wurde
- der im Speicher aufgebaute Katalog keine Definitionen enthält, also leer ist
- wenn bereits eine Datei mit dem bei CATALOG-NAME angegebenen Namen (ungleich *CURRENT) katalogisiert ist und REPLACE-OLD-FILE=*NO angegeben wurde

Zu beachten ist auch, dass dieser Anweisung ein Prüflauf auf den Katalog vorausgehen muss. Dieser Prüflauf kann mit der Anweisung CHECK-CATALOG (siehe [Seite 196](#)) eingeleitet werden. Für die Operanden DEPENDENCE-RELATION und LINK-RELATION muß dazu *YES angegeben werden.

Format

SAVE-CATALOG

CATALOG-NAME = *CURRENT / <filename 1..51 without-gen-vers>(…)

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

,**FORCED** = *NO / *YES / *FOR-ADD-SUBSYSTEM

Operandenbeschreibung

CATALOG-NAME =

Dateiname, unter dem der Katalog abgespeichert werden soll.

CATALOG-NAME = *CURRENT

Der Katalog wird unter dem Dateinamen abgespeichert, der in der letzten Anweisung START-CATALOG-CREATION oder START-CATALOG-MODIFICATION angegeben wurde.

CATALOG-NAME = <filename 1..51 without-gen-vers>(…)

Vollqualifizierter Dateiname, unter dem der Katalog gespeichert werden soll.

REPLACE-OLD-FILE= *NO / *YES

Existiert bereits eine katalogisierte Datei mit dem bei CATALOG-NAME angegebenen Namen, legt REPLACE-OLD-FILE fest, ob sie überschrieben werden darf (*YES) oder nicht (*NO).

FORCED =

Bestimmt, wie SSCM bei Fehlern in der Subsystemdefinition verfahren soll.

FORCED = *NO

Voreinstellung: bei Fehlern, die SSCM bei der Analyse der Subsystemdefinition erkennt, soll der Katalog nicht in einer PAM-Datei abgespeichert werden.

Fehler bei der Definition von Subsystemen könnten sich aus einer nicht-zyklenfreien Konstellation an Beziehungen (Adress-, Abhängigkeits- oder Externverweisbeziehungen) ergeben. Ein solcher Zyklus entsteht dann, wenn eine Beziehung („→“) definiert wird, die nach mehreren Schritten wieder auf sich selbst verweist:

```
Subsystem A  → Subsystem B
Subsystem B  → Subsystem C
Subsystem C  → Subsystem A
```

FORCED = *YES

Trotz möglicher Fehler, die SSCM bei der Analyse der Subsystemdefinition erkennen könnte, soll der Katalog in einer PAM-Datei abgespeichert werden.

Die Verantwortung für einen so gespeicherten Katalog obliegt dem Anwender; das Verhalten von DSSM kann nicht gewährleistet werden.

FORCED = *FOR-ADD-SUBSYSTEM

Der angegebene Katalog wird trotz folgender Fehler gesichert:

- ein Subsystem weist Abhängigkeitsbeziehungen zu nicht definierten Subsystemen auf
- ein Subsystem soll sich mit einem nicht definierten Subsystem eine Holdertask teilen

Dieser Wert ist dann sinnvoll, wenn ein Katalog erstellt werden soll, der lediglich neue Subsysteme enthält, die aber z.B. zu anderen, bereits im alten Katalog definierten Subsystemen Beziehungen haben. Zum bestehenden alten Katalog wird dieser neue mit dem Kommando ADD-SUBSYSTEM ...,TYPE=*NEW-SUBSYSTEMS hinzugefügt.

SAVE-SSD

Subsystemdefinition(en) abschließen

Funktionsbeschreibung

Die Anweisung SAVE-SSD schließt die Folge von Anweisungen zur Definition eines oder mehrerer Subsysteme ab, die mit den Anweisungen START-SSD-CREATION und SET-SUBSYSTEM-ATTRIBUTES eingeleitet wurde (siehe [Seite 288](#) bzw. [Seite 243](#)).

Die Anweisung veranlasst SSCM dazu, die Subsystemdefinition(en) in die in der Anweisung START-SSD-CREATION benannte ISAM-Datei abzuspeichern. Diese ISAM-Datei wird SSD-Object genannt.

SAVE-SSD wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-SSD-CREATION
- SET-SUBSYSTEM-ATTRIBUTES

SAVE-SSD wird mit einer Fehlermeldung abgebrochen, wenn das SSD-Object Subsysteme enthält, die

- mit CLOSE-CTRL-ROUTINE=*DYNAMIC definiert wurden, deren Holdertask aber nicht als Arbeitstask genutzt wird
- die verschiedene INSTALLATION-UNIT-Namen (installierte Liefergruppen) haben.

Format

SAVE-SSD

SEPARATE-ADDRESS-SPACE

Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher steuern

Funktionsbeschreibung

Mit dieser Anweisung wird die disjunkte Verteilung der Subsysteme im Klasse-5-Speicher gesteuert. Mithilfe dieser Anweisung ist es möglich, unerwünschte Adressraumüberschneidungen von Subsystemen, die sich aus der Anweisung SET-SUBSYSTEM-ATTRIBUTES ergeben könnten, zu vermeiden.

Die Angabe ist für Subsysteme im Klasse-3- oder Klasse-4-Speicher (Operand MEMORY-CLASS=*SYSTEM-GLOBAL in der Anweisung SET-SUBSYSTEM-ATTRIBUTES) sowie für Subsysteme im Klasse-6-Speicher (MEMORY-CLASS=*LOCAL-UNPRIVILEGED) nicht relevant. Subsysteme im Klasse-6-Speicher werden nie parallel gebraucht, dürfen sich also damit im Adressraum überschneiden, und Subsysteme im Klasse-3- oder Klasse-4-Speicher überschneiden sich grundsätzlich nicht.

SEPARATE-ADDRESS-SPACE wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-SSD-CREATION
- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Format

SEPARATE-ADDRESS-SPACE

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

,FROM-SUBSYSTEMS = list-poss(15): <structured-name 1..8>

Operandenbeschreibung

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

Name des Subsystems, das sich nicht mit anderen Subsystemen überschneiden darf. Das Subsystem muss SSCM bekannt und bereits definiert sein.

FROM-SUBSYSTEMS = list-poss(15): <structured-name 1..8>

Liste von maximal 15 Subsystemen, mit denen sich das im Operanden SUBSYSTEM-NAME angeführte Subsystem nicht überschneiden darf.

Hinweise

Die Anweisung SEPARATE-ADDRESS-SPACE erzeugt immer mindestens zwei disjunkte Beziehungen, wie am nachfolgenden Beispiel zu sehen ist:

```
//SEPARATE-ADDRESS-SPACE SUBSYSTEM-NAME=EINS, FROM-SUBSYSTEMS=(ZWEI, DREI)
```

Die Ausführung dieser Anweisung schafft folgende Beziehungen:

- der Adressraum des Subsystems EINS ist disjunkt zum Adressraum des Subsystems ZWEI
- der Adressraum des Subsystems EINS ist disjunkt zum Adressraum des Subsystems DREI
- der Adressraum des Subsystems ZWEI ist disjunkt zum Adressraum des Subsystems EINS
- der Adressraum des Subsystems DREI ist disjunkt zum Adressraum des Subsystems EINS

SET-SUBSYSTEM-ATTRIBUTES

Eigenschaften und Einsprungstellen eines Subsystems definieren

Funktionsbeschreibung

Mit dieser Anweisung können alle Eigenschaften und Einsprungstellen eines Subsystems definiert werden.

In welcher Weise die Definition eines neuen Subsystems aufgebaut wird, ist abhängig davon, welche Anweisung der Definition vorausging:

- Nach der Anweisung START-CATALOG-CREATION ([Seite 286](#)) kann das Subsystem in den **Katalog** integriert werden.
- Nach der Anweisung START-SSD-CREATION ([Seite 288](#)) kann das Subsystem in das **SSD-Object** integriert werden.
In ein und demselben SSD-Object können jedoch nicht verschiedene Versionen des gleichen Subsystems angelegt werden.

Subsysteme, die mit dem Operanden MEMORY-CLASS=*SYSTEM-GLOBAL ..., SUBSYSTEM-ACCESS=*SYSTEM definiert werden, sind privilegierte Subsysteme.

Bei der Definition ist ferner auf Folgendes zu achten:

- Der Name des Subsystems und seine Version dürfen noch nicht im Subsystemkatalog bekannt sein.
- Zwei verschiedene Versionen ein- und desselben Subsystems können nicht in einem SSD-Object definiert werden.
- Der Subsystem-Name CP ist für DSSM reserviert und darf nicht angegeben werden.
- Die Namen der Einsprungstellen dürfen nicht doppelt genannt werden.
- Der Name des zu definierenden Subsystems darf nicht in der Liste der Subsysteme enthalten sein, zu denen Adress- oder Abhängigkeitsbeziehungen bestehen.
- Innerhalb dieser Liste darf keine Doppelnennung eines Subsystems auftauchen.

SET-SUBSYSTEM-ATTRIBUTES wird abgewiesen, wenn vorher keine der folgenden Anweisungen ausgeführt wurde:

- START-SSD-CREATION
- START-CATALOG-CREATION
- START-CATALOG-MODIFICATION

Hinweis zur Syntax

Für die Namen der Einsprungstellen in den folgenden Operanden (im Format ist der Datentyp <name> angegeben) kann auch der spezielle Datentyp <symbol> verwendet werden, der im Handbuch „BLSSERV“ [4] ausführlich beschrieben ist:

- LINK-ENTRY
- DYNAMIC-CHECK-ENTRY
- INIT-ROUTINE
- CLOSE-CTRL-ROUTINE
- DEINIT-ROUTINE
- INTERFACE-VERSION
- SUBSYSTEM-ENTRIES
- STOPCOM-ROUTINE

Format

(Teil 1 von 4)

SET-SUBSYSTEM-ATTRIBUTES
<p>SUBSYSTEM-NAME = <structured-name 1..8>(…)</p> <p> <structured-name 1..8>(…)</p> <p> VERSION = <c-string 3..8> / <text 3..8></p> <p>,INSTALLATION-UNIT = *NONE / *STD / <text 1..30></p> <p>,INSTALLATION-USERID = *NONE / *DEFAULT-USERID / <name 1..8></p> <p>,COPYRIGHT = *NONE / <c-string 1..54>(…)</p> <p> <c-string 1..54>(…)</p> <p> YEAR = *YEAR-1990 / <c-string 4..4></p> <p>,LIBRARY = *STD / *CPLINK / *INSTALLED(…) / <filename 1..54 without-gen-vers></p> <p> *INSTALLED(…)</p> <p> LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers></p> <p> ,DEFAULT-NAME = <filename 1..54 without-gen-vers></p> <p>,SUBSYSTEM-LOAD-MODE = *ANY / *STD / *ADVANCED</p> <p>,REP-FILE = *STD / *NO / *INSTALLED(…) / <filename 1..54 without-gen-vers></p> <p> *INSTALLED(…)</p> <p> LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers></p> <p> ,DEFAULT-NAME = <filename 1..54 without-gen-vers> / *NONE</p> <p>,REP-FILE-MANDATORY = *NO / *YES</p>

Fortsetzung ➔

```

,MESSAGE-FILE = *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        |
        | LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = <filename 1..54 without-gen-vers> / *NONE
,SUBSYSTEM-INFO-FILE = *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        |
        | LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = <filename 1..54 without-gen-vers> / *NONE
,SYNTAX-FILE = *NO / *INSTALLED(...) / <filename 1..54 without-gen-vers>
    *INSTALLED(...)
        |
        | LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>
        | ,DEFAULT-NAME = <filename 1..54 without-gen-vers> / *NONE
,DYNAMIC-CHECK-ENTRY = *STD / *NO / <text 1..8 without-sep>
,CREATION-TIME = *AT-CREATION-REQUEST / *AT-SUBSYSTEM-CALL(...) / *AT-DSSM-LOAD /
    *BEFORE-DSSM-LOAD / *MANDATORY-AT-STARTUP / *BEFORE-SYSTEM-READY /
    *AFTER-SYSTEM-READY
    *AT-SUBSYSTEM-CALL(...)
        |
        | ON-ACTION = *STD / *ISL-CALL / *ANY
,INIT-ROUTINE = *NO / <text 1..8 without-sep>
,CLOSE-CTRL-ROUTINE = *NO / *DYNAMIC / <text 1..8 without-sep>
,STOPCOM-ROUTINE = *NO / *DYNAMIC / <text 1..8 without-sep>
,DEINIT-ROUTINE = *NO / *DYNAMIC / <text 1..8 without-sep>
,STOP-AT-SHUTDOWN = *NO / *YES
,INTERFACE-VERSION = *NO / <text 1..8 without-sep>
,SUBSYSTEM-HOLD = *ALLOWED / *FORBIDDEN
,STATE-CHANGE-CMDS = *ALLOWED / *FORBIDDEN / *BY-ADMINISTRATOR-ONLY
,FORCED-STATE-CHANGE = *FORBIDDEN / *ALLOWED
,RESET = *FORBIDDEN / *ALLOWED
,RESTART-REQUIRED = *NO / *YES
,VERSION-COEEXISTENCE = *FORBIDDEN / *ALLOWED
,VERSION-EXCHANGE = *FORBIDDEN / *ALLOWED

```

Fortsetzung ➔

```

, SUBSYSTEM-ENTRIES = *NONE / list-poss(100): <text 1..8>(…) / *BY-PROGRAM(…)
<text 1..8>(…)
|
| MODE = *LINK / *ISL(…) / *SVC(…) / *SYSTEM-EXIT(…)
|
|   *ISL(…)
|   |
|   |   FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
|   |   |
|   |   |   FUNCTION-VERSION = <integer 1..255>
|   |
|   | *SVC(…)
|   | |
|   | |   NUMBER = <integer 0..255>
|   | |   , CALL-BY-SYSTEM-EXIT = *ALLOWED / *FORBIDDEN
|   | |   , FUNCTION-NUMBER = *NONE / <integer 0..255>(…)
|   | |   |
|   | |   |   FUNCTION-VERSION = <integer 1..255>
|   |
|   | *SYSTEM-EXIT(…)
|   | |
|   | |   NUMBER = <integer 0..127>
|   |
|   | , CONNECTION-ACCESS = *ALL / *SYSTEM / *SIH
|   | , CONNECTION-SCOPE = *TASK / *PROGRAM / *FREE / *CALL / *OPTIMAL
|   | , FIRST-CONNECTION = *ALLOWED / *FORBIDDEN
|
| *BY-PROGRAM(…)
| |
| |   CONNECTION-SCOPE = *TASK / *PROGRAM
|
| , MEMORY-CLASS = *SYSTEM-GLOBAL (…) / *LOCAL-PRIVILEGED(…) / *LOCAL-UNPRIVILEGED(…) /
|   *BY-SLICE(…)
|
| *SYSTEM-GLOBAL(…)
| |
| |   SUBSYSTEM-ACCESS = *LOW / *SYSTEM / *HIGH
|
| *LOCAL-PRIVILEGED(…)
| |
| |   SIZE = <integer 1..32767>
|
| *LOCAL-UNPRIVILEGED(…)
| |
| |   SIZE = <integer 1..32767>
| |   , SUBSYSTEM-ACCESS = *LOW / *HIGH
| |   , START-ADDRESS = *ANY / <x-string 7..8>

```

Fortsetzung →

```

*BY-SLICE(...)
  |  SIZE = <integer 1..32767>
,LINK-ENTRY = <text 1..8 without-sep>(…)
  <text 1..8 without-sep>(…)
  |  AUTOLINK = *FORBIDDEN / *ALLOWED
,REFERENCED-SUBSYSTEM = *NONE / list-poss(15): <structured-name 1..8>(…)
  <structured-name 1..8>(…)
  |  LOWEST-VERSION = *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
  |  ,HIGHEST-VERSION = *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>
,UNRESOLVED-EXTERNALS = *FORBIDDEN / *ALLOWED
,CHECK-REFERENCE = *YES / *NO
,RELATED-SUBSYSTEM = *NONE / list-poss(100): <structured-name 1..8>(…)
  <structured-name 1..8>(…)
  |  LOWEST-VERSION = *LOWEST-EXISTING / <c-string 3..8> / <text 3..8>
  |  ,HIGHEST-VERSION = *HIGHEST-EXISTING / <c-string 3..8> / <text 3..8>

```

Operandenbeschreibung

SUBSYSTEM-NAME = <structured-name 1..8>(…)

Vereinbart Name und Version des zu definierenden Subsystems.

VERSION = <c-string 3..8> / <text 3..8>

Die Version des Subsystems ist im Format „[V][m]m.n[aso]“ zu vereinbaren, wobei die Textteile folgende Bedeutung besitzen:

mm : Hauptversion (numerisch)

n : Nachtragsversion (numerisch)

aso : Änderungsstand

(a: Buchstabe, Freigabestand; so: numerisch, Korrekturstand)

INSTALLATION-UNIT =

Legt den Namen der installierten Liefergruppe fest. Für alle mit IMON zu installierenden Subsysteme muss ein Wert ungleich *NONE angegeben werden, ebenso, wenn bei den Operanden SUBSYSTEM-LIBRARY, REP-FILE, SUBSYSTEM-INFO-FILE, MESSAGE-FILE und SYNTAX-FILE der Wert *INSTALLED(LOGICAL-ID=…) definiert wurde.

Die im Handbuch „IMON“ [17] dargestellten Syntaxregeln sind bei der Festlegung des Namens zu beachten.

INSTALLATION-UNIT = *NONE

Voreinstellung: Es wird kein Name vergeben. Für alle mit IMON installierten Subsysteme ist diese Voreinstellung nicht erlaubt.

INSTALLATION-UNIT = *STD

Der beim Operanden SUBSYSTEM-NAME angegebene Name wird als Name der installierten Liefergruppe genutzt.

INSTALLATION-UNIT = <text 1..30>

Name der installierten Liefergruppe.

INSTALLATION-USERID =

Vereinbart eine Benutzerkennung, unter der die zuständige DSSM-Task die NebenkompONENTEN des Subsystems (Rep-Datei, Objektmodulbibliothek, Meldungsdatei, Syntaxdatei und Subsystem-Informationsdatei) erwartet, falls diese Dateien noch keiner Benutzerkennung zugeordnet sind, d.h. der Dateiname ohne Benutzerkennung angegeben wurde.

INSTALLATION-USERID = *NONE

Die Dateien werden nicht unter einer bestimmten Benutzerkennung erwartet.

INSTALLATION-USERID = *DEFAULT-USERID

Die Dateien werden unter der System-Standardkennung erwartet (Prefix „\$.“) bzw. unter der Benutzerkennung der aufrufenden Task, wenn es sich um ein lokales Subsystem handelt.

INSTALLATION-USERID = <name 1..8>

Benutzerkennung, unter der die NebenkompONENTEN erwartet werden.

Gilt die Anweisung für ein SSD-Object, werden die Dateien nur dann unter der hier angegebenen Benutzerkennung erwartet, wenn in der Anweisung ADD-CATALOG-ENTRY (Übernahme von Subsystemdefinitionen aus dem SSD-Object in den Katalog, [Seite 183](#)) keine Benutzerkennung angegeben wurde. Die Angabe bei ADD-CATALOG-ENTRY hat Vorrang.

COPYRIGHT =

Vereinbart, ob und welche Copyright-Meldung beim Starten des Subsystems ausgegeben werden soll.

COPYRIGHT = *NONE

Voreinstellung: es soll keine Copyright-Meldung ausgegeben werden.

COPYRIGHT = <c-string 1..54>(…)

Text der Copyright-Meldung, die zusammen mit dem Datum der Erstellung beim Starten ausgegeben wird.

YEAR = *YEAR-1990 / <c-string 4..4>

Jahreszahl, die in der Copyright-Meldung als Datum der Erstellung erscheinen soll. Eine semantische Prüfung findet nicht statt.

LIBRARY =

Vereinbart den Namen der Programm- oder Bindemodulbibliothek (OML), aus der der Objectcode des Subsystems bei dessen Aktivierung geladen werden soll.

LIBRARY = *STD

Voreinstellung: der Objectcode wird beim Starten automatisch aus der Bibliothek SYSLNK.<subsysname>.<subsysvers#> geladen. Sie ist auf der Benutzerkennung abgelegt, unter der die Holdertask läuft; also auf der Benutzerkennung des Aufrufers bei lokalen Subsystemen und auf TSOS bei globalen Subsystemen.

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

LIBRARY = *CPLINK

Das zu definierende Subsysteme ist mit dem Basissystem des BS2000/OSD (CP=Control Program) verknüpft und muss bereits vor der Aktivierung von DSSM geladen sein.

Der Operand darf nur in Verbindung mit dem Operanden CREATION-TIME =*BEFORE-DSSM-LOAD verwendet werden.

LIBRARY = *INSTALLED(...)

Der Bibliotheksname muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompontenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompontenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Programm- oder Bindemodulbibliothek.

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

Neuer Bibliotheksname bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Vollqualifizierter Dateiname der Bindemodulbibliothek, aus der der Objectcode für das Subsystem geladen werden soll.

SUBSYSTEM-LOAD-MODE =

Bestimmt den Lademodus des Subsystems (über die BLS-DSSM-Schnittstelle \$PBBND1).

SUBSYSTEM-LOAD-MODE = *ANY

Voreinstellung: das BLS wird im STD-Run-Mode aufgerufen und lädt das Subsystem als Objektmodul. Tritt während des Ladens ein Fehler ein, wird BLS ein zweites Mal aufgerufen. Der Aufruf erfolgt im ADVANCED-Run-Mode und das Subsystem wird als Bindelademodul (LLM) geladen.

Dieser Operandenwert wird nur aus Kompatibilitätsgründen weiter unterstützt.

SUBSYSTEM-LOAD-MODE = *STD

Das BLS wird im STD-Run-Mode aufgerufen und lädt das Subsystem als Objektmodul.

SUBSYSTEM-LOAD-MODE = *ADVANCED

Das BLS wird im ADVANCED-Run-Mode aufgerufen und lädt das Subsystem als Bindelademodul (LLM).

REP-FILE =

Legt fest, ob System-Reps für das zu definierende Subsystem benötigt werden und in welcher Datei diese hinterlegt sind. Diese Korrekturanweisungen werden während der Aktivierung des Subsystems ausschließlich auf die in der Bindemodulbibliothek hinterlegten und geladenen Module angewandt, nicht auf andere Subsysteme oder BS2000/OSD CP. Eine Rep-Datei kann auch für Module eines nicht-privilegierten Subsystems vereinbart werden.

REP-FILE darf nicht zusammen mit LIBRARY=*CPLINK angegeben werden.

REP-FILE = *STD

Standardmäßig werden die System-Reps aus der Rep-Datei mit dem Namen SYSREP.<subsysname>.<subsysvers#> geladen. Die Datei ist auf der Benutzerkennung abgelegt, unter der die Holdertask läuft; also auf der Benutzerkennung des Aufrufers bei lokalen Subsystemen und auf TSOS bei globalen Subsystemen.

Der Wert von „subsysvers#“ ist dreistellig und setzt sich aus den beim Operanden SUBSYSTEM-NAME=...(VERSION=...) angegebenen Teilen „mmn“ zusammen.

REP-FILE = *NO

Für das Subsystem soll keine Rep-Datei verarbeitet werden.

REP-FILE = *INSTALLED(...)

Der Name der Rep-Datei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der NebenkompONENTEN mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden NebenkompONENTEN logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Rep-Datei.

DEFAULT-NAME =

Name der Rep-Datei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

REP-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Rep-Datei, aus der die Korrekturanweisungen gelesen werden sollen.

REP-FILE-MANDATORY =

Legt fest, ob eine mit dem Operanden REP-FILE deklarierte Rep-Datei beim Laden des Subsystems abgearbeitet werden muss oder nicht.

REP-FILE-MANDATORY = *NO

Voreinstellung: der Einsatz einer Rep-Datei ist nicht Pflicht, d.h. weder die Rep-Datei noch deren Einträge sollen beim Aktivieren des Subsystems geprüft werden. Sollte die Rep-Datei nicht zugreifbar oder einzelne Korrekturanweisungen fehlerhaft sein, wird das Subsystem auch in diesem Fall gestartet.

REP-FILE-MANDATORY = *YES

Sollte es bei der Bearbeitung der Rep-Datei zu folgenden Fehlern kommen, wird der Versuch, das Subsystem zu laden, abgebrochen:

- Die Rep-Datei ist nicht katalogisiert oder kann nicht gelesen werden.
- Die Prüfung der Korrekturanweisungen zeigt einen Fehler an.
- Der Name der Korrekturanweisungen ist fehlerhaft.
- Das DMS meldet einen Fehler beim Zugriff auf die NOREF-Datei (diese Datei wird während des Ladens eines Subsystems benutzt, um zu verhindern, dass ungültige System-Reps an der Bedienstation protokolliert werden).

MESSAGE-FILE =

Bestimmt, ob es eine subsystemspezifische Meldungsdatei gibt, die beim Laden des Subsystems automatisch aktiviert wird. Für Subsysteme, die mit dem Startzeitpunkt AT-DSSM-LOAD definiert werden, ist im Operanden RELATED-SUBSYSTEM eine Abhängigkeitsbeziehung zum Subsystem MIP zu vereinbaren.

MESSAGE-FILE = *NO

Voreinstellung: es soll keine Meldungsdatei aktiviert werden. Dieser Wert ist verpflichtend für alle Subsysteme, die mit dem Startzeitpunkt BEFORE-DSSM-LOAD definiert werden (siehe auch Operand CREATION-TIME), da zu diesem Zeitpunkt noch keine Meldungsdatei aktiviert werden kann.

MESSAGE-FILE = *INSTALLED(...)

Der Name der Meldungsdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Meldungsdatei.

DEFAULT-NAME =

Name der Meldungsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

MESSAGE-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Meldungsdatei. Diese wird beim Laden des Subsystems automatisch aktiviert, beim Entladen automatisch deaktiviert (START-/ STOP-SUBSYSTEM).

SUBSYSTEM-INFO-FILE =

Bestimmt, ob eine Subsysteminformationsdatei (SSINFO) vorhanden ist. In dieser Datei sind subsystemspezifische Daten (Nebenkomponenten und Konfigurationsdaten) enthalten, die nicht von DSSM zentral bearbeitet werden können.

SUBSYSTEM-INFO-FILE = *NO

Voreinstellung: eine Informationsdatei für das Subsystem ist nicht verfügbar.

SUBSYSTEM-INFO-FILE = *INSTALLED(...)

Der Name der Informationsdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkomponenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkomponenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Informationsdatei.

DEFAULT-NAME =

Name der Informationsdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

SUBSYSTEM-INFO-FILE = <filename 1..54 without-gen-vers>

Vollqualifizierter Name der Informationsdatei. Der Name wird automatisch an die Aktivierungs- und Deaktivierungsroutinen (Operanden INIT-/DEINIT-/STOPCOM-/CLOSE-CTRL-ROUTINE) übergeben, wenn diese aufgerufen werden.

SYNTAX-FILE =

Bestimmt, ob eine Syntaxdatei mit dem Subsystem verknüpft ist, die beim Laden des Subsystems automatisch aktiviert wird. Für Subsysteme, die mit dem Startattribut MANDATORY-AT-STARTUP definiert werden, ist im Operanden RELATED-SUBSYSTEMS eine Abhängigkeitsbeziehung zum Subsystem SDF zu deklarieren.

SYNTAX-FILE = *NO

Voreinstellung: es soll keine Syntaxdatei aktiviert werden. Dieser Wert ist verpflichtend für alle Subsysteme, die mit dem Startzeitpunkt BEFORE-DSSM oder AT-DSSM-LOAD definiert werden (siehe auch Operand CREATION-TIME), da zu diesem Zeitpunkt noch keine Syntaxdatei aktiviert werden kann.

SYNTAX-FILE = *INSTALLED(...)

Der Name der Syntaxdatei muss durch den Aufruf von IMON-GPN (Verwaltung von Installationspfaden) bestimmt werden.

Wird eine der Nebenkompenten mit einem logischen Namen angesprochen, müssen bei allen zu diesem Subsystem gehörenden Nebenkompenten logische Namen angegeben werden. Wird ein logischer Name vergeben, muss beim Operanden INSTALLATION-UNIT ein Wert ungleich *NONE vergeben werden.

LOGICAL-ID = <filename 1..30 without-catid-userid-gen-vers>

Logischer Name der Syntaxdatei.

DEFAULT-NAME =

Name der Syntaxdatei bei Nichtverfügbarkeit von IMON-GPN oder wenn der logische Name unbekannt ist.

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

Es wird ein neuer Name vergeben.

DEFAULT-NAME = *NONE

Es wird kein neuer Name vergeben.

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

Vollqualifizierter Name der Syntaxdatei, die beim Laden des Subsystems automatisch aktiviert werden soll.

DYNAMIC-CHECK-ENTRY =

Vereinbart, ob eine dynamische Identitätsprüfung des Subsystems vorgenommen werden soll. Zu diesem Zweck muss eine Einsprungstelle angegeben werden, an der sowohl der Subsystemname (acht Zeichen) als auch die Versionsnummer (vier bzw. sieben Zeichen) stehen muss. DSSM prüft, ob die bei der Definition vergebene Identifikation mit dem geladenen Subsystem übereinstimmt.

DYNAMIC-CHECK-ENTRY = *STD

Als Voreinstellung gilt, dass die bei dem Operanden LINK-ENTRY spezifizierte Einsprungstelle für die Identitätsprüfung herangezogen werden soll.

DYNAMIC-CHECK-ENTRY = *NO

Eine Überprüfung soll nicht stattfinden. Dieser Operandenwert darf allerdings für solche Subsysteme, die vor der Aktivierung von DSSM geladen sein sollen (CREATION-TIME= *BEFORE-DSSM-LOAD), nicht verwendet werden.

DYNAMIC-CHECK-ENTRY = <text 1..8 without-sep>

Name der Einsprungstelle, die für die Identitätsprüfung herangezogen werden soll.

CREATION-TIME =

Legt den Zeitpunkt fest, an dem die Aktivierung des Subsystems (CREATE-Routine) angestoßen wird.

Während der Systemeinführung sind zwei Phasen zu unterscheiden, in denen DSSM nach Aufruf durch die Startup-Routine die Steuerung der Systemeinführung übernimmt:

Phase 1: Der DSSM-Code wird geladen, die DSSM-Task generiert und gestartet. Die Task reserviert Klasse-5-Speicher, liest den Subsystemkatalog ein und startet die Subsysteme, die mit den Startattributen BEFORE-DSSM-LOAD und AT-DSSM-LOAD definiert wurden. Nach dem Laden dieser Subsysteme geht die Steuerung der Systemeinführung an die Startup-Routine zurück.

Phase 2: Nach erneutem Aufruf werden alle Subsysteme geladen, die mit den Startattributen MANDATORY-AT-STARTUP, BEFORE-SYSTEM-READY und AFTER-SYSTEM-READY definiert wurden.

Bei den beiden erstgenannten wird das Laden der Subsysteme mit der Startup-Routine synchronisiert (d.h. das Laden muss abgeschlossen sein), beim letztgenannten wird das Laden asynchron angestoßen.

Die Steuerung der Systemeinführung geht an die Startup-Routine zurück.

Sollen verschiedene Versionen eines Subsystems definiert werden, können die Startattribute, die für die Phasen 1 und 2 der Systemeinführung vorgesehen sind, nur für eine dieser Versionen vergeben werden.

CREATION-TIME = *AT-CREATION-REQUEST

Voreinstellung: das Subsystem muss explizit mit dem Kommando START-SUBSYSTEM geladen werden.

CREATION-TIME = *AT-SUBSYSTEM-CALL(...)

Das Subsystem soll automatisch beim ersten SVC- oder ISL-Aufruf geladen werden. Dieser Operandenwert ist reserviert für Subsysteme, die über SVC oder ISL aufgerufen werden.

Sind mehrere Versionen eines Subsystems mit diesem Operandenwert definiert, muss für alle diese Versionen VERSION-COEEXISTENCE=*ALLOWED angegeben werden sowie FUNCTION-NUMBER und FUNCTION-VERSION für ihre SVC- bzw. ISL-Einsprungstellen, die mit CONNECTION-ACCESS ungleich *SIH deklariert wurden.

Mindestens eines der angegebenen Subsysteme muss mit SUBSYSTEM-ENTRIES ..., MODE=*SVC/*ISL deklariert worden sein (übereinstimmend mit dem Wert des Operanden ON-ACTION).

ON-ACTION =

Bestimmt, wodurch das automatische Laden des Subsystems veranlasst wird.

ON-ACTION = *STD

Voreinstellung: das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC-Einsprungstelle.

ON-ACTION = *ISL-CALL

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden ISL-Einsprungstelle.

ON-ACTION = *ANY

Das Laden beginnt beim Aufruf einer beliebigen, zum Subsystem gehörenden SVC- oder ISL-Einsprungstelle.

CREATION-TIME = *AT-DSSM-LOAD

Das Subsystem soll während der Systemeinleitung (Phase 1) unter der Kontrolle der DSSM-Task geladen werden.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die ebenfalls mit diesem Startattribut oder dem Startattribut BEFORE-DSSM-LOAD definiert sind.

Die Dateien für dieses Subsystem müssen unter der Benutzerkennung TSOS auf dem Home-Pubset angelegt sein, da zum Startzeitpunkt weder der Benutzerkatalog zugreifbar, noch die IMPORT-PUBSET-Verarbeitung abgeschlossen ist.

Für diese Subsysteme ist die Angabe einer Syntaxdatei nicht zulässig.

CREATION-TIME = *BEFORE-DSSM-LOAD

Das Subsystem soll während der Systemeinleitung (Phase 1), aber nicht unter der Kontrolle der DSSM-Task geladen werden.

Solche Subsysteme sind mit dem Organisationsprogramm verknüpft und brauchen - bei der Aktivierung - nicht mit der DSSM-Task synchronisiert werden. Nach dem Laden des Subsystems läuft dieses allerdings wieder unter der Kontrolle von DSSM und kann aus Sicht des Anwenders wie andere Subsysteme gesteuert werden.

Anschluss- oder Abhängigkeitsbeziehungen zu Subsystemen, die mit einem anderen Startattribut definiert wurden, sind nicht möglich. Auch die Einbindung einer Meldungs- oder

Syntaxdatei ist nicht zulässig. Alle Auftragseingänge (Operand SUBSYSTEM-ENTRIES) müssen deklariert sein, da DSSM die (privilegierte) Verbindung zu diesen Auftragseingängen herstellt. Es liegt in der vollen Verantwortung des Subsystem-Entwicklers, sicherzustellen, dass zu jedem Zeitpunkt mindestens eine Version dieses Subsystems verfügbar ist (z.B. PLAM). Der Name des Link-Kontextes für diese Subsysteme muss eindeutig sein, da DSSM auch eine Entlade-Anforderung erfüllen muss, selbst wenn DSSM den Subsystem-Code nicht geladen hat. Eine Einsprungstelle (Operand DYNAMIC-CHECK-ENTRY) muss angegeben sein.

CREATION-TIME = *BEFORE-SYSTEM-READY

Das Subsystem soll während der Systemeinleitung (Phase 2) geladen werden. Die Aktivierung wird synchron angestoßen; die Steuerung geht erst nach dem vollständigen Laden (oder nach Lade-Fehler) an die Startup-Routine zurück, die dann „SYSTEM READY“ melden kann.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD oder MANDATORY-AT-STARTUP definiert wurden. Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

Wird ein nicht-privilegiertes Subsystem mit diesem Operandenwert deklariert, bekommt es implizit den Wert *AFTER-SYSTEM-READY zugewiesen. SSCM gibt eine Meldung aus.

CREATION-TIME = *MANDATORY-AT-STARTUP

Das Subsystem muss während der Systemeinleitung (Phase 2) geladen werden. Die Aktivierung wird - wie bei BEFORE-SYSTEM-READY - synchron angestoßen. Im Unterschied zum oben genannten muss das Laden des Subsystems allerdings **erfolgreich** abgeschlossen werden. Andernfalls geht eine Meldung an die Startup-Routine, dass ein verpflichtendes Subsystem nicht geladen werden konnte. Die Startup-Routine entscheidet in diesem Fall, ob die Verarbeitung fortgesetzt oder abgebrochen wird.

Das Subsystem muss privilegiert sein und darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD oder AT-DSSM-LOAD definiert wurden. Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

CREATION-TIME = *AFTER-SYSTEM-READY

Das Laden des Subsystems wird während der Systemeinleitung (Phase 2) angestoßen. Die Durchführung dieses Auftrags wird nicht mit der Startup-Routine synchronisiert, die vor dem Abschluss des Ladens „SYSTEM READY“ melden kann.

Das Subsystem darf nur Adress- oder Abhängigkeitsbeziehungen zu Subsystemen aufweisen, die mit dem gleichen oder den Startattributen BEFORE-DSSM-LOAD, AT-DSSM-LOAD, MANDATORY-AT-STARTUP oder BEFORE-SYSTEM-READY definiert wurden.

Die Dateien für dieses Subsystem müssen auf dem Home-Pubset liegen.

INIT-ROUTINE =

Legt fest, ob eine Initialisierungsroutine für das Subsystem durchlaufen werden soll, wenn es gestartet oder fortgesetzt wird. In diesem Fall muss der Name einer Einsprungstelle bekannt sein und DSSM delegiert die Initialisierung an die Holdertask des betreffenden Subsystems. Für alle Subsysteme mit dem Startattribut BEFORE-DSSM-LOAD wird die Angabe einer Einsprungstelle unbedingt empfohlen. Beim Laden des Subsystems (d.h. dem Durchlaufen der Initialisierungsroutine) erhält dieses dann Kenntnis davon, dass DSSM die Kontrolle über Anschluss und Abbau von Beziehungen übernehmen kann. Eine INIT-Routine muss mit RESTART-REQUIRED=*YES deklariert werden.

INIT-ROUTINE = *NO

Voreinstellung: es wird keine Initialisierungsroutine durchlaufen.

INIT-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der Initialisierungsroutine.

Der Initialisierungsroutine wird in der Holdertask die Steuerung übergeben, damit sich das Subsystem initialisieren kann. Dazu werden ihr übergeben:

- der Name und die Version des Subsystems, wie im SSCMCAT definiert
- der Name der SSINFO-Datei, falls dieser im Operanden SUBSYSTEM-INFO-FILE spezifiziert wurde
- die Adresse der beim Laden und Binden angegebenen Einsprungstelle (LINK-ENTRY)
- der vom Dynamischen Bindelader verwendete Binder-Kontext-Name
- der Name des Memory-Pools (für Subsysteme im Klasse-5- oder Klasse-6-Speicher), damit sich das Subsystem beim Nachladen eigener Selectable Units/Load Units darauf beziehen kann
- der Name der Meldungsdatei
- die Adresse des Operanden SUBSYSTEM-PARAMETER, falls im Kommando START-SUBSYSTEM ein String angegeben wird

Am Ende der Initialisierung wird eine Rückmeldung des Subsystems erwartet, ob die Initialisierung erfolgreich durchgeführt wurde und ob die Holdertask als Arbeitstask genutzt werden soll (wird in der Anweisung ASSIGN-HOLDER-TASK vereinbart, [Seite 193](#)). Abhängig davon steht die Task dann weiter unter DSSM-Kontrolle oder unter Kontrolle des Subsystems.

CLOSE-CTRL-ROUTINE =

Legt fest, ob eine spezielle Routine für das Subsystem durchlaufen werden soll, wenn es angehalten oder deaktiviert wird.

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE. Beim Ansprung dieser Routine wird sichergestellt, dass noch Verbindung zum Subsystem besteht.

Existiert eine CLOSE-CTRL-Routine, tritt bei einem Versionswechsel während der BS2000-Session keine Unterbrechung auf. Es existiert zu jedem Zeitpunkt genau eine gültige Version (entweder die alte Version ist noch verfügbar oder die neue Version ist bereits verfügbar). Ohne eine solche Routine beinhaltet ein Versionswechsel immer eine Anschlussunterbrechung, während die STOPCOM-Routine der alten Version und die INIT-Routine der neuen Version ablaufen (siehe dazu auch [Seite 45](#)).

CLOSE-CTRL-ROUTINE = *NO

Voreinstellung: im betreffenden Subsystem ist keine Routine verankert, die beim Deaktivieren oder Anhalten des Subsystems durchlaufen werden soll.

CLOSE-CTRL-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse oder FITC. Die notwendigen Parameter werden vom Subsystem am Ende der INIT-Routine dynamisch an die CLOSE-CTRL-Routine übergeben und DSSM über die Identifikation der Börse bzw. FITC-Ports in Kenntnis gesetzt.

Voraussetzung für die Nutzung der CLOSE-CTRL-Routine sind die Angabe einer INIT-Routine (Operand INIT-ROUTINE) und einer STOPCOM-Routine (Operand STOPCOM-ROUTINE=*NO/*DYNAMIC).

Beim Aufruf der CLOSE-CTRL-Routine muss die Holdertask des Subsystems Arbeitstask sein (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

CLOSE-CTRL-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

STOPCOM-ROUTINE =

Legt fest, ob eine Routine in das Subsystem eingebunden ist, die das aktive Beenden der Aufträge durchführen kann.

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE.

Beim Ansprung dieser Routine ist sichergestellt, dass aufrufende Tasks nicht mehr an das Subsystem angeschlossen werden. Tasks, die noch in Aufruf-Beziehung zum Subsystem stehen, bleiben davon unberührt.

Wird CLOSE-CTRL-ROUTINE=*DYNAMIC vereinbart, muss der Operand STOPCOM-ROUTINE=*NO/*DYNAMIC angegeben werden.

STOPCOM-ROUTINE = *NO

Voreinstellung: im betreffenden Subsystem ist keine Routine verankert.

STOPCOM-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse oder FITC. Die notwendigen Parameter werden vom Subsystem am Ende der CLOSE-CTRL-Routine oder (wenn eine solche nicht vorhanden ist) am Ende der INIT-Routine dynamisch an die STOPCOM-Routine übergeben. DSSM wird über die Identifikation der Börse bzw. FITC-Ports in Kenntnis gesetzt.

Voraussetzung für die Nutzung der STOPCOM-Routine ist die Angabe einer INIT-Routine (Operand INIT-ROUTINE).

Beim Aufruf der STOPCOM-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

STOPCOM-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

DEINIT-ROUTINE =

Legt fest, ob eine Routine in das Subsystem eingebunden ist, die die Deinitialisierung des Subsystems durchführen kann. Diese Deinitialisierungsroutine realisiert die Rückgabe der vom Subsystem angeforderten Betriebsmittel (Speicher, Dateien, Geräte).

Wird ein Subsystem (mit STOP-SUBSYSTEM oder HOLD-SUBSYSTEM) deaktiviert, so erhält es von DSSM in der Holdertask an der bezeichneten Einsprungstelle die Kontrolle oder es wird (bei *DYNAMIC) über Börsen- bzw. FITC-Linkage benachrichtigt (gesteuert von Rückmeldungen bei der Initialisierung).

Wird ein Subsystem mit einer INIT- und einer CLOSE-CTRL-Routine definiert, muss eine DEINIT-Routine - mit dem gleichen Operandenwert wie die CLOSE-CTRL-Routine - angegeben werden.

Die übergebenen Parameter sind die gleichen wie bei INIT-ROUTINE. Beim Ansprung dieser Routine ist sichergestellt, dass aufrufende Tasks nicht mehr an das Subsystem angeschlossen werden und alle vorhandenen Aufruf-Beziehungen gelöst werden.

DEINIT-ROUTINE = *NO

Voreinstellung: im betreffenden Subsystem ist keine Deinitialisierungsroutine verankert, die die Rückgabe der Betriebsmittel veranlasst.

DEINIT-ROUTINE = *DYNAMIC

Der Aufruf dieser Routine erfolgt über Börse bzw. FITC. Die notwendigen Parameter werden vom Subsystem am Ende der STOPCOM-Routine oder (wenn eine solche nicht vorhanden ist) am Ende der CLOSE-CTRL-Routine oder am Ende der INIT-Routine, wenn weder eine STOPCOM- noch eine CLOSE-CTRL-Routine eingebunden ist, dynamisch an die DEINIT-Routine übergeben. DSSM wird über die Identifikation der Börse bzw. des FITC-Ports in Kenntnis gesetzt.

Voraussetzung für die Nutzung der DEINIT-Routine ist die Angabe einer INIT-Routine (Operand INIT-ROUTINE).

Beim Aufruf der DEINIT-Routine muss die Holdertask des Subsystems als Arbeitstask genutzt werden (Anweisung ASSIGN-HOLDER-TASK, [Seite 193](#)).

DEINIT-ROUTINE = <text 1..8 without-sep>

Name der Einsprungstelle der betreffenden Subsystemroutine.

STOP-AT-SHUTDOWN =

Legt fest, ob das Subsystem nach Beendigung der Benutzertasks bei Shutdown automatisch entladen werden soll.

STOP-AT-SHUTDOWN = *NO

Voreinstellung: das Subsystem wird nicht automatisch entladen.

Die Angabe sollte nicht für Subsysteme verwendet werden, die Adressbeziehungen zu anderen Subsystemen besitzen, die mit STOP-AT-SHUTDOWN=*YES definiert sind.

STOP-AT-SHUTDOWN = *YES

Das Subsystem wird bei Shutdown automatisch entladen.

Diese Angabe wird ignoriert, wenn keine STOPCOM-, DEINIT- oder CLOSE-CTRL-Routine angegeben wird.

INTERFACE-VERSION =

Bezeichnet die Einsprungstelle, über die DSSM auf diejenige Schnittstellenversion zugreifen kann, die für den Aufruf der INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routinen benutzt werden soll.

Dieser Operand ist Pflicht für Subsysteme, für die eine Einsprungstelle angegeben wurde (INIT-, CLOSE-CTRL-, STOPCOM-, DEINIT-Routine).

INTERFACE-VERSION = *NO

Voreinstellung: das Subsystem ruft keine INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routine auf.

INTERFACE-VERSION = <text 1..8 without-sep>

Name der Einsprungstelle. Die Einsprungstelle verweist auf den Standardheader, in dem u.a. auch die Schnittstellenversion hinterlegt ist. Der Standardheader wird durch den Aufruf des Makros \$ESMINT(!) mit MF=I/L generiert.

Dieser Operand ist Pflicht für Subsysteme, für die eine INIT-, DEINIT-, STOPCOM- oder CLOSE-CTRL-Routine definiert wurde.

SUBSYSTEM-HOLD =

Legt fest, ob das geladene Subsystem angehalten oder entladen werden kann.

SUBSYSTEM-HOLD = *ALLOWED

Voreinstellung: das geladene Subsystem kann angehalten und entladen werden.

Die Kommandos HOLD-SUBSYSTEM und STOP-SUBSYSTEM sind für dieses Subsystem zulässig.

SUBSYSTEM-HOLD = *FORBIDDEN

Die Kommandos HOLD-SUBSYSTEM und STOP-SUBSYSTEM sind für dieses Subsystem nicht zulässig; das Subsystem wird - entsprechend den Angaben im Operanden STOP-AT-SHUTDOWN - bei Shutdown entladen.

Wird das Subsystem durch Austausch mit einem anderen Subsystem entladen, so erfolgt der Austausch unterbrechungsfrei.

STATE-CHANGE-CMDS =

Legt fest, ob die DSSM-Kommandos zur Steuerung des Subsystems im laufenden Betrieb (START-SUBSYSTEM, STOP-SUBSYSTEM, HOLD-SUBSYSTEM und RESUME-SUBSYSTEM) verwendet werden dürfen.

Wird von einer Version eines Subsystems in eine andere gewechselt, wird der bei STATE-CHANGE-CMDS angegebene Wert für die auszuwechselnde Version nicht berücksichtigt.

STATE-CHANGE-CMDS = *ALLOWED

Voreinstellung: Die Kommandos dürfen an der Bedienstation und unter der privilegierten Benutzerkennung (Benutzerkennung, die mit dem Systemprivileg SUBSYSTEM-MANAGEMENT ausgestattet ist) verwendet werden.

STATE-CHANGE-CMDS = *FORBIDDEN

Die Kommandos dürfen generell nicht - weder an der Bedienstation noch unter der privilegierten Benutzerkennung - verwendet werden.

STATE-CHANGE-CMDS = *BY-ADMINISTRATOR-ONLY

Die Kommandos dürfen nur unter der privilegierten Benutzerkennung verwendet werden; für den Operator an der Bedienstation sind die Kommandos gesperrt.

FORCED-STATE-CHANGE =

Legt fest, ob innerhalb der Kommandos STOP-SUBSYSTEM und HOLD-SUBSYSTEM die Verwendung des Operanden FORCED=*YES zulässig ist. Mit dieser Funktion kann das unbedingte Deaktivieren des Subsystems erzwungen werden.

FORCED-STATE-CHANGE = *FORBIDDEN

Voreinstellung: das Deaktivieren des Subsystems kann nicht erzwungen werden. DSSM weist die Verwendung des Operanden FORCED in den entsprechenden Kommandos mit einer Fehlermeldung zurück.

FORCED-STATE-CHANGE = *ALLOWED

Die Verwendung des Operanden FORCED=*YES für dieses Subsystem ist zulässig. Dieser Operandenwert darf nicht zusammen mit SUBSYSTEM-HOLD=*FORBIDDEN angegeben werden.

RESET =

Legt fest, ob innerhalb der Kommandos START-SUBSYSTEM und RESUME-SUBSYSTEM die Verwendung des Operanden RESET=*YES zulässig ist. Mit dieser Funktion kann das unbedingte Laden bzw. Fortsetzen des Subsystems erzwungen werden, auch wenn sich das Subsystem im Zustand IN-DELETE bzw. IN-HOLD befindet.

RESET = *FORBIDDEN

Voreinstellung: das Aktivieren des Subsystems kann nicht erzwungen werden. DSSM weist die Verwendung des Operanden RESET in den entsprechenden Kommandos mit einer Fehlermeldung zurück.

RESET = *ALLOWED

Die Verwendung des Operanden RESET=*YES für dieses Subsystem ist zulässig. Dieser Operandenwert darf nicht mit SUBSYSTEM-HOLD=*FORBIDDEN angegeben werden.

RESTART-REQUIRED =

Legt fest, ob bei abnormaler Beendigung der Holdertask die Initialisierungsroutine für das Subsystem durchlaufen werden soll.

RESTART-REQUIRED = *NO

Voreinstellung: die Initialisierungsroutine wird für einen Wiederanlauf des Subsystems nicht benutzt.

RESTART-REQUIRED = *YES

Die Initialisierungsroutine soll bei abnormaler Beendigung der Holdertask benutzt werden. Voraussetzung ist, dass die Durchführung dieser Routine im Operanden INIT-ROUTINE vorgesehen ist.

VERSION-COEXISTENCE =

Vereinbart, ob mehr als eine Version des gleichen Subsystems gleichzeitig aktiv sein darf.

VERSION-COEXISTENCE = *FORBIDDEN

Voreinstellung: Die aktuelle Version des Subsystems kann nicht gleichzeitig mit einer anderen Version des gleichen Subsystems koexistieren.

VERSION-COEXISTENCE = *ALLOWED

Die aktuelle Version des Subsystems kann gleichzeitig mit einer anderen Version des gleichen Subsystems koexistieren (Coexistence-Modus).

Bei der Definition des Auftragseingangs (Operand SUBSYSTEM-ENTRIES) darf keine indirekte Verbindung über System-Exit-Routinen gewählt werden. Sind verschiedene Versionen des gleichen Subsystems geladen, die mit dem gleichen Auftragseingang definiert wurden, wird der Anschluss immer an die höchste geladene Version realisiert.

Greifen koexistente Subsysteme auf koexistente Syntaxdateien zu, müssen diese im SSD-Object deklariert sein und können nicht von SDF verwaltet werden.

Bei Anschlüssen über SVC und ISL ist jedoch eine Versionsauswahl über die Operanden FUNCTION-NUMBER und FUNCTION-VERSION möglich.

VERSION-EXCHANGE =

Vereinbart, ob das Laden eines Subsystems im Exchange-Modus erlaubt ist.

Der Exchange-Modus erlaubt die temporäre Koexistenz zweier Versionen des gleichen Subsystems. Wird die Version B eines Subsystems geladen, wenn bereits Version A des Subsystems aktiv ist, werden alle neuen Aufrufer an die Version B angeschlossen. Die Aufträge, die an die Version A angeschlossen sind, werden noch bearbeitet. Nach Bearbeitung aller Aufträge an Version A wird diese automatisch beendet.

Bei der Definition ist darauf zu achten, dass die zu ersetzende, „alte“ Version nicht abhängig von der ersetzenden, „neuen“ Version sein darf.

VERSION-EXCHANGE = *FORBIDDEN

Voreinstellung: Die aktuelle Version des Subsystems darf nicht ausgetauscht werden.

VERSION-EXCHANGE = *ALLOWED

Der Exchange-Modus, der die temporäre Koexistenz zweier Subsysteme erlaubt, ist für die aktuelle Subsystemversion zulässig.

SUBSYSTEM-ENTRIES =

Vereinbart die Einsprungstellen (Auftragseingänge), die mit dem Subsystem verbunden sind. Es können bis zu 100 Auftragseingänge vereinbart werden. Werden mehr als 100 Auftragseingänge für ein Subsystem gewünscht, können diese mit den Anweisungen MODIFY-SUBSYSTEM-ATTRIBUTES (für ein Subsystem im Katalog) bzw. ADD-SUBSYSTEM-ENTRIES (für ein Subsystem in einem SSD-Object) definiert werden.

SUBSYSTEM-ENTRIES = *NONE

Voreinstellung: es werden keine Einsprungstellen vereinbart.

SUBSYSTEM-ENTRIES = list-poss(100): <text 1..8>

Vereinbart durch Angabe des Namens der Einsprungstelle maximal 100 Auftragseingänge, deren Typ jeweils in den Unterstrukturen definiert werden muss.

MODE =

Legt den Typ eines vereinbarten Auftragseingangs für das Subsystem fest.

MODE = *LINK

Voreinstellung: der Auftragseingang kann nicht über indirektes Linkage erreicht werden, sondern nur über eine CONNECT-Beziehung mittels eines externen Binder-Symbols.

Bei verschiedenen Versionen des gleichen Subsystems, die das gleiche externe Binder-Symbol nutzen, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

MODE = *ISL(...)

Der Auftragseingang wird durch indirekte Verbindung über System Procedure Linkage (nur für privilegierte Subsysteme) erreicht. Wird zusätzlich noch eine Funktions- und Versionsnummer der ISL-Einsprungstelle spezifiziert, darf sich die Kombination aus Name der Einsprungstelle, Funktions- und Versionsnummer nicht mit einer anderen

Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEXISTENCE=*ALLOWED) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über die selbe ISL-Einsprungsstelle erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die die selbe ISL-Einsprungsstelle nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die die selbe ISL-Einsprungsstelle nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standardheader der Parameterliste des Aufrufers hinterlegt ist.

Der Wert *ALL des Operanden CONNECTION-ACCESS ist für ISL-Einsprungsstellen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der ISL-Einsprungsstelle angesprochen werden soll, da die gleiche ISL-Einsprungsstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Voreinstellung: es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der ISL-Einsprungsstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten ISL-Funktionsnummer.

MODE = *SVC(…)

Der Auftragseingang wird durch indirekte Verbindung über Supervisor Call (SVC) erreicht.

Wird zusätzlich noch eine Funktions- und Versionsnummer der SVC-Einsprungsstelle spezifiziert, darf sich die Kombination aus SVC-Nummer, Funktions- und Versionsnummer nicht mit einer anderen Kombination für die verschiedenen Subsysteme eines Kataloges oder die verschiedenen Versionen des gleichen Subsystems (bei Angabe von VERSION-COEXISTENCE=*ALLOWED) decken.

Bei ungleichen Subsystemen, deren Auftragseingang über den selben SVC erreicht werden soll, muss zur eindeutigen Unterscheidung die Funktions- und Versionsnummer angegeben werden.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen - ohne Angaben zur Funktions- oder Versionsnummer -, stellt DSSM automatisch den

Anschluss an die höchste geladene Version des Subsystems her.

Bei verschiedenen Versionen des gleichen Subsystems, die den selben SVC nutzen und deren Funktions- und Versionsnummer ungleich *NONE ist, ist die Auswahl, an welche Version der Anschluss erfolgen soll, von der Funktions- und Versionsnummer abhängig, die im Standardheader der Parameterliste des Aufrufers hinterlegt ist.

Bei Angabe dieses Operandenwertes sollte der Operand CONNECTION-ACCESS vorzugsweise mit dem Wert *SYSTEM statt *ALL belegt werden.

NUMBER = <integer 0..255>

Nummer des SVCs, über den der Auftragseingang erreicht wird. In Verbindung mit CONNECTION-ACCESS=*ALL ist die Verwendung einer SVC-Nummer größer 191 nicht zulässig.

CALL-BY-SYSTEM-EXIT =

Legt fest, ob die angegebene SVC-Nummer von System-Exit-Routinen aus aufgerufen werden darf.

CALL-BY-SYSTEM-EXIT = *ALLOWED

Voreinstellung: der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen zulässig.

CALL-BY-SYSTEM-EXIT = *FORBIDDEN

Der Aufruf der angegebenen SVC-Nummer ist für System-Exit-Routinen nicht zulässig.

FUNCTION-NUMBER =

Vereinbart, ob eine bestimmte Funktions- und Versionsnummer der SVC-Einsprungstelle angesprochen werden soll, da die gleiche SVC-Einsprungstelle von verschiedenen Funktionen genutzt werden kann.

FUNCTION-NUMBER = *NONE

Voreinstellung: es soll keine bestimmte Funktions- oder Versionsnummer angesprochen werden.

FUNCTION-NUMBER = <integer 0..255>(…)

Nummer der SVC-Einsprungstelle. Die Version ist in der folgenden Unterstruktur zu benennen.

FUNCTION-VERSION = <integer 1..255>

Version der spezifizierten SVC-Funktionsnummer.

MODE = SYSTEM-EXIT(…)

Der Auftragseingang wird durch indirekte Verbindung über System-Exit-Routinen erreicht.

In Verbindung mit CONNECTION-ACCESS=*ALL ist die Verwendung dieses Operanden nicht zulässig.

NUMBER = <integer 0..127>

Nummer der System-Exit-Routine.

CONNECTION-ACCESS =

Vereinbart die Zugriffsberechtigung (Privilegierung) für das Subsystem.

CONNECTION-ACCESS = *ALL

Voreinstellung: privilegierte und nicht-privilegierte Programmläufe sind zugriffsberechtigt. In Verbindung mit MODE=*SYSTEM-EXIT/*ISL/*SVC (mit einer SVC-Nummer größer 191) ist die Verwendung dieses Operandenwertes nicht zulässig.

CONNECTION-ACCESS = *SYSTEM

Nur privilegierte Programmläufe sind zugriffsberechtigt.

CONNECTION-ACCESS = *SIH

Nur Tasks, die im Funktionszustand SIH laufen, sind zugriffsberechtigt. Das aufgerufene Subsystem läuft ebenfalls im Funktionszustand SIH, d.h. es ist nicht unterbrechbar.

Die Angabe dieses Operandenwertes ist nur für Subsysteme zulässig, deren Auftragseingang definiert wird über

- System Procedure Linkage (MODE=*ISL(FUNCTION-NUMBER=*NONE))
- CONNECTION-SCOPE=*OPTIMAL
- MEMORY-CLASS=*SYSTEM-GLOBAL(SUBSYSTEM-ACCESS=*SYSTEM)

CONNECTION-SCOPE =

Bezeichnet das Ereignis, das die automatische Auflösung des Anschlusses an den angegebenen Subsystem-Auftragseingang hervorruft.

CONNECTION-SCOPE = *TASK

Voreinstellung: der Anschluss wird bei Taskbeendigung aufgehoben.

CONNECTION-SCOPE = *PROGRAM

Der Anschluss wird spätestens bei Programmbeendigung aufgehoben. Zusammen mit MEMORY-CLASS=*LOCAL-UNPRIVILEGED darf nur CONNECTION-SCOPE=*PROGRAM angegeben werden.

Die Angabe dieses Operandenwertes wird für Subsysteme empfohlen, die mit SUBSYSTEM-ACCESS=*LOW/*HIGH oder MEMORY-CLASS=*BY-SLICE deklariert wurden.

CONNECTION-SCOPE = *FREE

DSSM soll keine Kontrolle von Anschlüssen zu diesen Auftragseingängen durchführen. Der Anschluss wird - außer bei einer expliziten Anforderung - nicht automatisch aufgelöst.

Um Probleme oder mögliche Fehler beim Entladen des Subsystems zu vermeiden, müssen die Anschlüsse vom Subsystem selbst verwaltet werden.

CONNECTION-SCOPE = *CALL

Nach Rückkehr aus diesem Auftragseingang führt DSSM automatisch die Auflösung der Anschlüsse durch.

Dieser Operandenwert steht nur für Subsysteme zur Verfügung, deren Auftragseingang über System Procedure Linkage (ISL) oder Supervisor Call (SVC) definiert wird.

CONNECTION-SCOPE = *OPTIMAL

Das Subsystem wird deaktiviert bzw. angehalten, wenn keine Task mehr Anschluss zu diesem Auftragseingang hat.

Eine Routine, deren Einsprungstelle mit *OPTIMAL definiert wird, muss mit RETURN beendet werden.

Wird ein Auftragseingang eines Subsystems mit CONNECTION-SCOPE=*OPTIMAL definiert, sollten alle seine Auftragseingänge im Subsystemkatalog mit MODE≠*LINK definiert werden.

Während ein Subsystem deaktiviert oder angehalten wird, wird kein Aufruf des Subsystems mit CONNECTION-SCOPE=*OPTIMAL akzeptiert.

FIRST-CONNECTION =

Bestimmt, ob der Erst-Anschluss der Task an den angegebenen Auftragseingang des Subsystems erlaubt ist. Mindestens ein Auftragseingang eines Subsystems muss mit FIRST-CONNECTION=*ALLOWED definiert werden.

FIRST-CONNECTION = *ALLOWED

Voreinstellung: Der Erst-Anschluss an den angegebenen Auftragseingang ist erlaubt.

FIRST-CONNECTION = *FORBIDDEN

Der Anschluss an den angegebenen Auftragseingang über SVC oder ISL ist nicht erlaubt, wenn die Task nicht bereits an einen anderen Auftragseingang des Subsystems angeschlossen ist.

Die Angabe dieses Operandenwertes ist für Auftragseingänge, die mit MODE=*LINK/*SYSTEM-EXIT oder CONNECTION-ACCESS=*SIH definiert wurden, nicht erlaubt.

SUBSYSTEM-ENTRIES = *BY-PROGRAM(...)

Die Einsprungstellen des angegebenen Subsystems werden nicht statisch aus dem Katalog versorgt, sondern dynamisch zum Ladezeitpunkt aus dem BLS-Namensverzeichnis.

Voraussetzung für diese Funktionalität ist der Einsatz von BLSSERV ab Version 2.1, das den Einsatz von Extended External Names (EEN-Namen) als Einsprungstellen für DSSM-Subsysteme unterstützt.

Der Wert kann nicht zusammen angegeben werden mit dem Operanden SUBSYSTEM-ACCESS=*SYSTEM.

Wenn die Einsprungstellen mit diesem Wert definiert sind, werden die Operanden MODE, CONNECTION-ACCESS und FIRST-CONNECTION mit den Standardwerten versorgt.



Der Wert *BY-PROGRAM ist nicht nur auf EEN-Namen beschränkt. Er kann auch für Subsysteme angegeben werden, die eine Vielzahl von Einsprungstellen aufweisen.

CONNECTION-SCOPE = *TASK / *PROGRAM

Der Anschluss wird bei Task- bzw. Programmbeendigung aufgehoben.

MEMORY-CLASS =

Vereinbart den subsystemspezifischen Adressraum, in den das Subsystem geladen werden soll. Mit diesem Operanden kann die Systembetreuung die Adressraumvorgaben, die für das jeweilige Subsystem gelten, den speziellen Anforderungen der Installation anpassen.

MEMORY-CLASS = *SYSTEM-GLOBAL(...)

Voreinstellung: das Subsystem wird in den Klasse-3- oder Klasse-4-Speicher geladen. Residente CSECTs erhalten Klasse-3-Speicher, alle anderen erhalten seitenwechselbaren Klasse-4-Speicher.

SUBSYSTEM-ACCESS =

Bezeichnet die Zugriffsrechte (Privilegierung) und die Lage des angeforderten Speichers.

SUBSYSTEM-ACCESS = *LOW

Voreinstellung: Es wird nicht-privilegierter Klasse-3- oder Klasse-4-Speicher unterhalb der 16 MByte-Grenze zugewiesen.

SUBSYSTEM-ACCESS = *SYSTEM

Subsysteme, die mit diesem Operandenwert deklariert werden, sind privilegierte Subsysteme, denen privilegierter Adressraum oberhalb der 16 MByte-Grenze zugewiesen wird.

Die Angabe dieses Operandenwertes ist für die Subsysteme verpflichtend, deren Auftragseingang über SVC (MODE=*SVC), oder für die eine INIT-, STOPCOM-, DEINIT- oder CLOSE-CTRL-Routine vereinbart wird.

Der Wert kann nicht zusammen mit einer Einsprungstelle angegeben werden, die zusammen mit CONNECTION-ACCESS=*ALL oder MODE=*LINK angegeben wurde.

Der Wert kann nicht zusammen angegeben werden mit dem Operanden SUBSYSTEM-ENTRIES=*BY-PROGRAM.

SUBSYSTEM-ACCESS = *HIGH

Es wird nicht-privilegierter Adressraum bis zu 2 GByte zugewiesen.

MEMORY-CLASS = *LOCAL-PRIVILEGED(...)

Das Subsystem erhält einen Memory Pool im nicht-privilegierten Klasse-5-Speicher, der unterhalb der 16 MByte-Grenze angelegt wird.

Die Angabe ist auf nicht-privilegierte Subsysteme zugeschnitten, die relativ viel Adressraum (+/- 1 MByte) beanspruchen und unterhalb der 16 MByte-Grenze anzulegen sind. Die Subsysteme werden in Memory Pools an der gleichen Adresse geladen, um mit dem knapp bemessenen Adressraum unterhalb 16 MByte zu haushalten.

Obwohl solche Subsysteme parallel im gleichen Adressraum geladen werden, können sie nicht simultan von einer Task genutzt werden (siehe auch Anweisung SEPARATE-ADDRESS-SPACE, [Seite 241](#)).

Das Subsystem darf keine residenten CSECTs enthalten, da ansonsten ein späteres Aktivieren abgebrochen wird.

SIZE = <integer 1..32767>

Größe des benötigten Adressraums (in 4KByte-Seiten) für den Memory Pool im Klasse-5-Speicher. Der Wert ist mindestens so groß zu wählen, dass das Subsystem und evtl. von diesem nachgeladene Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

MEMORY-CLASS = *LOCAL-UNPRIVILEGED(...)

Das Subsystem erhält einen Memory Pool im nicht-privilegierten Klasse-6-Speicher. Die Angabe ist für Subsysteme reserviert, die wie ein Programm ausführbar sind. Analog dazu muss deren Zugriffsberechtigung (Privilegierung) im Operanden CONNECTION-ACCESS mit dem Wert *ALL definiert werden.

Dieser Operandenwert darf nicht zusammen mit einer Einsprungstelle angegeben werden, die mit CONNECTION-ACCESS=*SYSTEM definiert wurde.

Das Subsystem darf keine residenten CSECTs enthalten, da ansonsten ein späteres Aktivieren abgebrochen wird.

Wird dieser Operandenwert angegeben, ist zur Auflösung des Anschlusses an den angegebenen Subsystem-Auftragseingang nur CONNECTION-SCOPE=*PROGRAM erlaubt.

SIZE = <integer 1..32767>

Größe des benötigten Adressraums (in 4KByte-Seiten) für den Memory Pool im Klasse-6-Speicher. Der Wert ist mindestens so groß zu wählen, dass das Subsystem und evtl. von diesem nachgeladene Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

SUBSYSTEM-ACCESS =

Bezeichnet die Lage des angeforderten Speicherplatzes.

SUBSYSTEM-ACCESS = *LOW

Voreinstellung: Es wird nicht-privilegierter Klasse-6-Speicher unterhalb der 16 MByte-Grenze zugewiesen.

Da diese Angabe auf Subsysteme zugeschnitten ist, die wie Programme ausführbar sind, ist die zusätzliche Angabe CONNECTION-SCOPE=*PROGRAM anzuraten.

SUBSYSTEM-ACCESS = *HIGH

Es wird nicht-privilegierter Klasse-6-Speicher bis zu 2 GByte zugewiesen.

Da diese Angabe auf Subsysteme zugeschnitten ist, die wie Programme ausführbar sind, ist die zusätzliche Angabe CONNECTION-SCOPE=*PROGRAM anzuraten.

START-ADDRESS =

Legt die Anfangsadresse im Klasse-6-Speicher fest.

START-ADDRESS = *ANY

Voreinstellung: die Lage des Subsystems im Klasse-6-Speicher wird von DSSM festgelegt.

START-ADDRESS = <x-string 7..8>

Start-Adresse im Segment-Raster, an der die Anfangsadresse des Subsystems liegen soll. Als Wert ist eine 8-stellige Hexadezimalkonstante anzugeben, die ein Vielfaches von X'100000' sein muss.

MEMORY-CLASS = *BY-SLICE(...)

Das angegebene Subsystem ist ein nicht-privilegiertes Subsystem und besteht aus einem LLM, das aus einem mehrbenutzbaren Code (Programmbereich) und einem nicht-mehrbenutzbaren Code (Datenbereich) besteht.

Der Programmbereich wird in den gemeinsam benutzbaren Adressraum geladen (das entspricht MEMORY-CLASS=*SYSTEM-GLOBAL). Der Datenbereich wird in den Benutzeradressraum der Holdertask geladen und in die privaten Benutzeradressräume der angeschlossenen Tasks an die selbe Adresse kopiert.

Gemeinsam mit MEMORY-CLASS=*BY-SLICE müssen folgende Werte angegeben werden: SUBSYSTEM-LOAD-MODE=*ADVANCED und CONNECTION-ACCESS=*ALL.

Der Wert kann nur zusammen angegeben werden mit den Operanden MODE=*LINK und CONNECTION-SCOPE=*TASK oder *PROGRAM.

SIZE = <integer 1..32767>

Größe des angeforderten Speicherplatzes für den Datenbereich in 4K-Seiten.

Der Wert ist mindestens so groß zu wählen, dass der Datenbereich und evtl. vom Subsystem in den Benutzeradressraum nachzuladende Selectable Units/Load Units in vollem Umfang geladen werden können. Die obere Grenze ist generierungsabhängig.

LINK-ENTRY = <text 1..8 without-sep>(…)

Definiert den Namen des zum Laden benötigten Bindemoduls/ENTRY/CSECT (als Operand im Makroaufruf \$PBBND1 an den dynamischen Bindelader DBL). Das Subsystem muss von diesem ENTRY vollständig (ggf. per Autolink) geladen werden.

AUTOLINK =

Steuert den Aufruf der Autolink-Funktion beim Binden und Laden.

Die Autolink-Funktion des Binders ermöglicht das automatische Einfügen von Modulen, die nicht mit entsprechenden Anweisungen explizit eingefügt werden.

Die Funktion erspart vor allem den Benutzern der höheren Programmiersprachen, die zahlreich benötigten Module des Laufzeitsystems (= Run Time System) mit expliziten Anweisungen einzeln einzufügen. Ein nähere Beschreibung der Autolink-Funktion ist im Handbuch „BLSSERV“ [4] zu finden.

Die Autolink-Funktion kann auch implizit umgangen werden, wenn während des Bindens des zu ladenden Objektmoduls der erste Externverweis auf ein vorgebundenes Großmodul zielt. Der Vorteil dieser Vorgehensweise ergibt sich daraus, dass das Seitenwechselverhalten bei der späteren Ausführung bereits im Vorfeld (während des Bindens) optimiert werden kann. Zudem können Fehler während des Bindens auf diese Weise vermieden werden.

AUTOLINK = *FORBIDDEN /*ALLOWED

Voreinstellung: Die Autolink-Funktion wird unterdrückt oder zugelassen.

REFERENCED-SUBSYSTEM =

Legt fest, ob es eine Liste von Subsystemen gibt, zu denen Adressbeziehungen bestehen und die zur Auflösung von Externverweisen benutzt werden.

REFERENCED-SUBSYSTEM = *NONE

Voreinstellung: das zu definierende Subsystem besitzt keine Externverweise.

REFERENCED-SUBSYSTEM = list-poss(15): <structured-name 1..8>

Das zu definierende Subsystem besitzt Externverweise auf maximal 15 andere Subsysteme, die zur Auflösung dieser Externverweise benutzt werden müssen. Fehlt eines der hier genannten Subsysteme beim Aktivieren oder Deaktivieren (und ist gleichzeitig eine Überprüfung der Externverweise mit dem Operanden CHECK-REFERENCE=*YES angefordert), wird die Aktion abgebrochen.

Auch das „Basis-Subsystem“ des BS2000/OSD, das Control Program, kann über diese Externverweise - mit dem Namen CP - angesprochen werden. Es kann in der folgenden Unterstruktur sowohl genau eine Version spezifiziert werden, als auch ein Bereich von Versionen angegeben werden, innerhalb dessen alle Versionen herangezogen werden sollen. Wird die Version des Subsystems CP über eine Versionsbereichsliste eingegrenzt, prüft DSSM die Kompatibilität zwischen der aktuellen CP-Version und Versionen in der Bereichsliste. Nur wenn die Versionen kompatibel sind, wird das Subsystem geladen.

Auf folgende Einschränkungen bei der Angabe von Subsystemen, zu denen Adressbeziehungen bestehen, ist zu achten:

- Es dürfen keine Adressbeziehungen zu Subsystemen vereinbart werden, die das Attribut MEMORY-CLASS=*LOCAL-PRIVILEGED/*LOCAL-UNPRIVILEGED/*BY-SLICE besitzen.
- Subsysteme, die das Attribut STOP-AT-SHUTDOWN=*NO besitzen, dürfen Adressbeziehungen nur zu solchen Subsystemen aufweisen, die auch dieses Attribut besitzen.
- Ist für das zu definierende Subsystem das Attribut SUBSYSTEM-ACCESS=*SYSTEM vergeben, dürfen keine Subsysteme angesprochen werden, die mit SUBSYSTEM-ACCESS=*LOW bzw. SUBSYSTEM-ACCESS=*HIGH definiert sind.
- Ein nicht-privilegiertes Subsystem darf generell keine Adressbeziehung zum Control Program CP aufweisen.
- Wird auf ein Subsystem verwiesen, das zumindest eine Version besitzt, die im Coexistence- oder Exchange-Modus betrieben werden darf, ist die eindeutige Version anzugeben.
- Die Adressbeziehungen müssen in Abhängigkeit der Startattribute (Operand CREATION-TIME) definiert werden; d.h. Subsysteme dürfen Beziehungen nur zu solchen Subsystemen aufnehmen, deren Start zeitgleich oder früher erfolgt.

UNRESOLVED-EXTERNALS =

Vereinbart das Verhalten des Ladevorgangs, wenn Externverweise nicht aufgelöst werden können.

UNRESOLVED-EXTERNALS = *FORBIDDEN

Voreinstellung: der Ladevorgang wird bei nicht auflösbaren Externverweisen abgebrochen.

UNRESOLVED-EXTERNALS = *ALLOWED

Der Ladevorgang wird fortgesetzt; nicht auflösbare Externverweise werden mit dem Wert 'X'FFFFFFF' besetzt.

CHECK-REFERENCE =

Vereinbart, ob die im Operanden REFERENCED-SUBSYSTEM angegebenen Subsysteme im Hinblick auf deren Zustand und Verfügbarkeit hin überprüft werden sollen.

CHECK-REFERENCE = *YES

Voreinstellung: die Referenz-Subsysteme werden überprüft. Fehlt eines der Subsysteme, bricht DSSM das Aktivieren oder Entladen des Subsystems ab.

CHECK-REFERENCE = *NO

DSSM soll keine Prüfung vornehmen.

Generiert der Anwender allerdings mit dieser Anweisung auch komplexe Subsysteme, führt DSSM die geforderten Funktionen **trotz** möglicher Konflikte durch:

- Das Kommando START-SUBSYSTEM lädt das angegebene Subsystem, auch wenn ein Subsystem, zu dem definierte Beziehungen bestehen, noch nicht vollständig geladen ist.
- Die Kommandos RESUME-SUBSYSTEM, STOP-SUBSYSTEM oder HOLD-SUBSYSTEM werden ohne Prüfung von Beziehungen und Abhängigkeiten von DSSM ausgeführt.

RELATED-SUBSYSTEM =

Legt fest, ob es eine Liste von Subsystemen gibt, zu denen Abhängigkeitsbeziehungen bestehen.

RELATED-SUBSYSTEM = *NONE

Voreinstellung: das zu definierende Subsystem besitzt keine Abhängigkeitsbeziehungen.

RELATED-SUBSYSTEM = list-poss(100): <structured-name 1..8>

Das zu definierende Subsystem besitzt Abhängigkeitsbeziehungen zu maximal 100 anderen Subsystemen, ohne die das zu definierende Subsystem nicht funktionsfähig ist.

Abhängigkeitsbeziehungen dürfen auch zum BS2000/OSD Control Program CP definiert werden. Die Regeln und Einschränkungen, die hierbei zu beachten sind, gelten analog für Adressbeziehungen (Operand REFERENCED-SUBSYSTEM) und sind dort näher erläutert.

Die Abhängigkeitsbeziehungen zielen jeweils auf eine Version eines Subsystems. In der folgenden Unterstruktur kann sowohl genau eine Version spezifiziert werden, als auch ein Bereich von Versionen angegeben werden, innerhalb dessen alle Versionen herangezogen werden sollen.

Auf folgende Einschränkungen ist allgemein bei der Angabe von abhängigen Subsystemen ist zu achten:

- Die definierte Beziehung muss zyklensfrei sein. Ein Zyklus entsteht, wenn Subsystem A abhängig von B, B abhängig von C und dieses wieder abhängig von A ist.
- Ist für das zu definierende Subsystem das Attribut MEMORY-CLASS=*SYSTEM-GLOBAL vergeben, dürfen keine Subsysteme angesprochen werden, die mit MEMORY-CLASS=*LOCAL-PRIVILEGED oder *LOCAL-UNPRIVILEGED definiert sind.
- Für Subsysteme, die das Attribut SUBSYSTEM-ACCESS=*SYSTEM besitzen, darf keine Abhängigkeitsbeziehung zu Subsystemen definiert werden, für die SUBSYSTEM-ACCESS=*LOW bzw. SUBSYSTEM-ACCESS=*HIGH oder MEMORY-CLASS=*BY-SLICE gilt.
- Die Abhängigkeitsbeziehungen müssen entsprechend den Startattributen (Operand CREATION-TIME) definiert werden; d.h. ein Subsystem darf nur von solchen Subsystemen abhängen, deren Start zeitgleich oder früher erfolgt.

LOWEST-VERSION =

Spezifiziert den unteren Wert (niedrigste Version) innerhalb der Versionsbereichsliste der Subsysteme.

LOWEST-VERSION = *LOWEST-EXISTING

Die niedrigste Version im Katalog soll angesprochen werden.

LOWEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Untergrenze des Versionsbereiches fungieren soll.

HIGHEST-VERSION =

Spezifiziert den oberen Wert (höchste Version) innerhalb der Versionsbereichsliste der Subsysteme.

HIGHEST-VERSION = *HIGHEST-EXISTING

Die höchste Version im Katalog soll angesprochen werden.

HIGHEST-VERSION = <c-string 3..8> / <text 3..8>

Version eines Subsystems, die als Obergrenze des Versionsbereiches fungieren soll.

SHOW-CATALOG

Subsystem-Konfiguration anzeigen

Funktionsbeschreibung

Mit dieser Anweisung wird der Inhalt einer Subsystem-Konfiguration, die in einem Subsystemkatalog gespeichert ist, wahlweise auf Bildschirm oder in eine Datei ausgegeben. Die Ausgabe beinhaltet immer den Namen des Subsystemkatalogs und die Information, ob und wie im Fehlerfall der Katalog abgespeichert wird (dieses Verhalten wird in der Anweisung SAVE-CATALOG ..., FORCED=... eingestellt).

SHOW-CATALOG wird zurückgewiesen, wenn der angegebene Datei- oder Subsystemname oder die angegebene Subsystemversion nicht existiert. Ist der Subsystemkatalog ohne Inhalt, wird eine Warnung ausgegeben und die Anweisung abgebrochen.

Wenn der angegebene Dateiname nicht dem des aktuell eröffneten Kataloges entspricht, wird folgende Meldung ausgegeben:

```
SCM0011      WOLLEN SIE WIRKLICH DEN KATALOG IM SPEICHER ÜBERSCHREIBEN ? (Y/N)
```

Bei Antwort **Y** gehen die virtuellen Definitionen des aktuellen Kataloges verloren.

Bei Antwort **N** wird die Ausführung der Anweisung SHOW-CATALOG abgebrochen. Der Anwender kann mit SAVE-CATALOG die bisher nicht gespeicherten Subsystemdefinitionen in einer Datei ablegen.

Wurde im Vorfeld die Anweisung CHECK-CATALOG nicht zur Konsistenzprüfung eingesetzt, kann SHOW-CATALOG Fehler bzgl. Konsistenz im Subsystemkatalog aufdecken.

Format

SHOW-CATALOG
<p>CATALOG-NAME = <u>*CURRENT</u> / <filename 1..54 without-gen-vers></p> <p>,CONTAINED-SUBSYSTEMS = <u>*NO</u> / *YES</p> <p>,SUBSYSTEM-NAME = <u>*ALL</u> / <structured-name 1..8>(…)</p> <p> <structured-name 1..8>(…)</p> <p> VERSION = <u>*ALL</u> / <c-string 3..8> / <text 3..8></p> <p>,GENERAL-ATTRIBUTES = <u>*YES</u> / *NO</p> <p>,INTERNAL-ENTRIES = <u>*YES</u> / *NO</p> <p>,MEMORY-ATTRIBUTES = <u>*YES</u> / *NO</p> <p>,RELATED-FILES = <u>*YES</u> / *NO</p> <p>,LINK-ATTRIBUTES = <u>*YES</u> / *NO</p> <p>,REFERENCE-RELATION = <u>*YES</u> / *NO</p> <p>,DEPENDENCE-RELATION = <u>*YES</u> / *NO</p> <p>,ADDR-SPACE-RELATION = <u>*YES</u> / *NO</p> <p>,HOLDER-TASK-INFO = <u>*YES</u> / *NO</p> <p>,SUBSYSTEM-ENTRIES = <u>*YES</u> / *NO</p> <p>,OUTPUT = <u>*SYSLST</u>(…) / *SYSOUT</p> <p> *SYSLST(…)</p> <p> SYSLST-NUMBER = <u>*STD</u> / <integer 1..99></p> <p> SUMMARY = <u>*YES</u> / *NO</p>

Operandenbeschreibung

CATALOG-NAME =

Vereinbart den Namen des Kataloges, in dem die anzuzeigenden Definitionen hinterlegt sind. Existiert der Katalog-Dateiname nicht, oder ist der angegebene Katalog ohne Inhalt, wird die Anweisung zurückgewiesen.

CATALOG-NAME = *CURRENT

Voreinstellung: der Inhalt des aktuell geöffneten Kataloges (Anweisung START-CATALOG-CREATION bzw. START-CATALOG-MODIFICATION) soll ausgegeben werden.

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

Vollqualifizierter Name des statischen Subsystemkataloges, dessen Inhalt angezeigt werden soll.

CONTAINED-SUBSYSTEMS = *NO / *YES

Vereinbart, ob die Liste der DSSM-Subsysteme, die als Definitionen im Katalog enthalten sind, ausgegeben werden soll (*YES) oder nicht (*NO).

SUBSYSTEM-NAME =

Vereinbart, über welche Subsysteme Informationen angefordert werden.

Wenn der Name des Subsystems oder eine Version, über die Informationen angefordert werden, nicht existiert, wird die Anweisung zurückgewiesen. Der Umfang der Informationen, die über die Subsysteme ausgegeben werden sollen, kann in den folgenden Operanden durch Angabe entsprechender Kriterien eingeschränkt werden.

SUBSYSTEM-NAME = *ALL

Voreinstellung: es sollen Informationen über alle Subsysteme angefordert werden, die im Katalog verzeichnet sind.

SUBSYSTEM-NAME = <structured-name 1..8>(…)

Name des Subsystems, über das SSCM die Informationen aus dem Katalog bereitstellt.

VERSION =

Spezifiziert die Version des ausgewählten Subsystems.

VERSION = *ALL

Alle im Katalog gespeicherten Versionen des Subsystems sollen in die Informationsausgabe eingeschlossen werden.

VERSION = <c-string 3..8> / <text 3..8>

Nur über diese Version des ausgewählten Subsystems sollen Informationen aus dem Katalog bereitgestellt werden.

GENERAL-ATTRIBUTES = *YES / *NO

Vereinbart, ob folgende generelle Eigenschaften der genannten Subsysteme aus dem Katalog gelesen werden sollen (*YES) oder nicht (*NO):

- Wann soll das Subsystem nach Systemeinleitung gestartet werden? (CREATION-TIME)
- In welchem Lademodus soll das Subsystem geladen werden? (SUBSYSTEM-LOAD-MODE)
- Soll das Subsystem bei Shutdown automatisch entladen werden? (STOP-AT-SHUTDOWN)
- Darf das geladene Subsystem angehalten oder entladen werden? (SUBSYSTEM-HOLD)

- Dürfen die Kommandos zur Steuerung des Subsystems verwendet werden? (STATE-CHANGE-CMDS)
- Ist die FORCE-Option zugelassen? (FORCED-STATE-CHANGE)
- Ist die RESET-Option zugelassen? (RESET)
- Muss bei abnormaler Beendigung der Holdertask die Initialisierungsroutine durchlaufen werden? (RESTART-REQUIRED)
- Darf mehr als eine Version des Subsystems gleichzeitig aktiv sein? (VERSION-COEEXISTENCE)
- Dürfen zwei Versionen eines Subsystems dynamisch ausgetauscht werden? (VERSION-EXCHANGE)
- Wie lautet der Name der INSTALLATION-UNIT des Subsystems? (INSTALLATION-UNIT)
- Wie lautet das Copyright (Text und Datum) des Subsystems? (COPYRIGHT)

INTERNAL-ENTRIES = *YES / *NO

Vereinbart, ob folgende Informationen über die Einsprungstellen der angegebenen Subsysteme von SSCM bereitgestellt werden sollen (*YES) oder nicht (*NO):

- die Namen der Einsprungstellen für die Subsystemroutinen INIT-, STOPCOM-, DEINIT- und CLOSE-CTRL-Routine
- der Name der Einsprungstelle, die für die dynamische Identitätsprüfung herangezogen wird (DYNAMIC-CHECK-ENTRY)
- der Name der Schnittstellenversion für den Aufruf der INIT-, STOPCOM-, DEINIT- oder CLOSE-CTRL-Routinen (INTERFACE-VERSION)

MEMORY-ATTRIBUTES = *YES / *NO

Vereinbart, ob folgende Speicher-relevanten Informationen, die im Katalog über die Subsysteme gespeichert sind, ausgegeben werden sollen (*YES) oder nicht (*NO):

- Speicherklasse (MEMORY-CLASS)
- Größe des benötigten Adressraums (SIZE)
- Anfangsadresse des Subsystemcodes (START-ADDRESS)
- Privilegierung und Zugriffsberechtigung bezüglich Adressraum (SUBSYSTEM-ACCESS)

RELATED-FILES = *YES / *NO

Vereinbart, ob Informationen über die NebenkompONENTEN des Subsystems geliefert werden sollen (*YES) oder nicht (*NO). In die Ausgabe eingeschlossen ist die Information, ob die Verwendung einer Rep-Datei für dieses Subsystem verpflichtend ist (REP-FILE-MANDATORY) und unter welcher Benutzerkennung die NebenkompONENTEN katalogisiert sind (INSTALLATION-USERID).

Unter dem Begriff NebenkompONENTEN werden zusammengefasst:

- die Objektmoduldatei des Subsystems (LIBRARY)
- die Meldungsdatei (MESSAGE-FILE)
- die Syntaxdatei (SYNTAX-FILE)
- die Informationsdatei des Subsystems (SUBSYSTEM-INFO-FILE)
- die Rep-Datei (REP-FILE)

LINK-ATTRIBUTES = *YES / *NO

Vereinbart, ob die Informationen aus dem Katalog zu lesen sind (*YES) oder nicht zu lesen sind (*NO), die in Bezug auf das Binden und Laden des Subsystems gespeichert wurden:

- der Name des zum Laden benötigten Bindemoduls/ENTRY/CSECT (LINK-ENTRY)
- die Einbindung der Autolink-Funktion (AUTOLINK)
- die Informationen über das Verhalten bei nicht auflösbaren Externverweisen (UNRESOLVED)
- die Einbindung des Prüflaufs für Referenz-Subsysteme (CHECK-REFERENCE)

REFERENCE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, zu denen Adressbeziehungen bestehen, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

DEPENDENCE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, zu denen Abhängigkeitsbeziehungen bestehen, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

ADDR-SPACE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, mit denen eine Adressraumüberschneidung vermieden werden muss, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

HOLDER-TASK-INFO = *YES / *NO

Vereinbart, ob die Identifikation der Holdertask und die Liste der Subsysteme, die in einer gemeinsamen Holdertask anzulegen sind, bei der Ausgabe der Katalog-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

SUBSYSTEM-ENTRIES = *YES / *NO

Vereinbart, ob die Liste der bei der Definition des Subsystems vereinbarten Auftragseingänge und deren folgende Eigenschaften aus dem Katalog gelesen werden soll (*YES) oder nicht (*NO):

- Typ des vereinbarten Auftragseingangs (MODE)
- Nummer der Routine (bei SVC oder System-Exit) (NUMBER)
- die Funktionsnummer der Einsprungstelle (FUNCTION-NUMBER)
- die Version der Funktionsnummer (FUNCTION-VERSION)
- die Information über Aufruf durch System-Exit-Routinen (CALL-BY-SYSTEM-EXIT)
- die Privilegierung und Zugriffsberechtigung bezüglich Einsprungstellen (CONNECTION-ACCESS und CONNECTION-SCOPE)

OUTPUT =

Vereinbart, wohin die von der Anweisung generierten Informationen ausgegeben werden sollen.

OUTPUT = *SYSLST(...)

Voreinstellung: die Meldungen sollen nach SYSLST ausgegeben werden.

SYSLST-NUMBER =

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

SYSLST-NUMBER = *STD

Voreinstellung: die Ausgabe soll in die Standard-Systemdatei SYSLST erfolgen.

SYSLST-NUMBER = <integer 1..99>

Die Ausgabe soll in eine Systemdatei aus der Menge SYSLST01 bis SYSLST99 erfolgen, deren Nummer hier anzugeben ist.

SUMMARY = *YES / *NO

Legt fest, ob eine zusammenfassende Übersicht über alle Subsysteme, die im Katalog verzeichnet sind, der geforderten Ausgabe vorangestellt werden soll (*YES) oder nicht (*NO). Für OUTPUT=*SYSLST(SUMMARY=*YES) siehe unten: Abkürzungs-Übersicht.

OUTPUT = *SYSOUT

Die Meldungen werden auf der Datensichtstation ausgegeben.

Bei der Ausgabe einer **Übersicht** über alle Subsysteme im Katalog (siehe Operand OUTPUT=*SYSLST(SUMMARY=*YES)) werden folgende Abkürzungen verwendet:

für CREATION-TIME

ACR : *AT-CREATION-REQUEST
 ASC : *AT-SUBSYSTEM-CALL
 ADL : *AT-DSSM-LOAD
 BDL : *BEFORE-DSSM-LOAD
 MAS : *MANDATORY-AT-STARTUP
 BSR : *BEFORE-SYSTEM-READY
 ASR : *AFTER-SYSTEM-READY

für MEMORY-CLASS

S : *SYSTEM-GLOBAL
 P : *LOCAL-PRIVILEGED
 U : *LOCAL-UNPRIVILEGED
 B : *BY-SLICE

für SUBSYSTEM-ACCESS

SYS : *SYSTEM
 ALL : *LOW / *HIGH

für INTERNAL-ENTRIES

DYN : *DYNAMIC
 YES : name
 NO : *NO

für CONNECTION-ACCESS

SYS : *SYSTEM

für STATE-CHANGE-CMDS

ADM : *BY-ADMINISTRATOR-ONLY

für REP-FILE

MAN : REP-FILE = *STD / dateiname und REP-FILE-MANDATORY = *YES

für GENERAL-ATTRIBUTES

YES : *ALLOWED
 NO : *FORBIDDEN

für RELATED-FILES

YES : *STD / dateiname
 NO : *NO
 IMO : *INSTALLED

SHOW-SSD

Inhalt eines SSD-Objects (Subsystemdefinitionen) anzeigen

Funktionsbeschreibung

Mit dieser Anweisung wird der Inhalt eines SSD-Objects wahlweise auf Bildschirm oder in eine andere Datei ausgegeben. In einem SSD-Object sind die Definitionen eines oder mehrerer Subsysteme hinterlegt.

Die Ausgabe erfolgt separat für alle Subsystemdefinitionen, die im angegebenen SSD-Object (ISAM-Datei) gespeichert sind.

Name und Version des SSD-Objects werden immer ausgegeben, ebenso der Name der Domäne für das SSD-Object und ob PULS-Problemmeldungen in das SSD-Object eingearbeitet sind.

Es ist zu beachten, dass SHOW-SSD nicht den vollständigen Inhalt des SSD-Objects anzeigt, sondern nur die Anweisungen zur Subsystemdefinition, die nach der letzten Anweisung ADD-CATALOG-ENTRY eingegeben wurden.

Format

SHOW-SSD
<pre> SSD-FILE-NAME = *<u>CURRENT</u> / <filename 1..54 without-gen-vers> ,GENERAL-ATTRIBUTES = *<u>YES</u> / *NO ,INTERNAL-ENTRIES = *<u>YES</u> / *NO ,MEMORY-ATTRIBUTES = *<u>YES</u> / *NO ,RELATED-FILES = *<u>YES</u> / *NO ,LINK-ATTRIBUTES = *<u>YES</u> / *NO ,REFERENCE-RELATION = *<u>YES</u> / *NO ,DEPENDENCE-RELATION = *<u>YES</u> / *NO ,ADDR-SPACE-RELATION = *<u>YES</u> / *NO ,HOLDER-TASK-INFO = *<u>YES</u> / *NO ,SUBSYSTEM-ENTRIES = *<u>YES</u> / *NO ,OUTPUT = *<u>SYSLST</u>(...) / *SYSOUT *SYSLST(...) SYSLST-NUMBER = *<u>STD</u> / <integer 1..99> </pre>

Operandenbeschreibung

SSD-FILE-NAME =

Gibt den Namen des SSD-Objects (ISAM-Datei) an, in dem die anzuzeigenden Definitionen hinterlegt sind. Existiert keine ISAM-Datei mit diesem Namen oder ist die angegebene Datei ohne Inhalt, wird die Anweisung zurückgewiesen.

SSD-FILE-NAME = *CURRENT

Voreinstellung: der Inhalt des aktuell geöffneten SSD-Objects (Anweisung START-SSD-CREATION) soll ausgegeben werden.

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

Vollqualifizierter Name der SSD-Objects, dessen Inhalt angezeigt werden soll.

GENERAL-ATTRIBUTES = *YES / *NO

Vereinbart, ob folgende generelle Eigenschaften der SSD-Object definierten Subsysteme angezeigt werden sollen (*YES) oder nicht (*NO):

- Wann soll das Subsystem nach Systemeinleitung gestartet werden?
(CREATION-TIME)
- In welchem Lademodus soll das Subsystem geladen werden?
(SUBSYSTEM-LOAD-MODE)
- Soll das Subsystem bei Shutdown automatisch entladen werden?
(STOP-AT-SHUTDOWN)
- Darf das geladene Subsystem angehalten oder entladen werden?
(SUBSYSTEM-HOLD)
- Dürfen die Kommandos zur Steuerung des Subsystems verwendet werden?
(STATE-CHANGE-CMDS)
- Ist die FORCE-Option zugelassen?
(FORCED-STATE-CHANGE)
- Ist die RESET-Option zugelassen?
(RESET)
- Muss bei abnormaler Beendigung der Holdertask die Initialisierungsroutine durchlaufen werden? (RESTART-REQUIRED)
- Darf mehr als eine Version des Subsystems gleichzeitig aktiv sein?
(VERSION-COEXISTENCE)
- Dürfen zwei Versionen eines Subsystems dynamisch ausgetauscht werden?
(VERSION-EXCHANGE)
- Wie lautet der Name der INSTALLATION-UNIT des Subsystems?
(INSTALLATION-UNIT)
- Wie lautet das Copyright (Text und Datum) des Subsystems?
(COPYRIGHT)

INTERNAL-ENTRIES = *YES / *NO

Vereinbart, ob folgende Informationen über die Einsprungstellen der enthaltenen Subsysteme von SSCM bereitgestellt werden sollen (*YES) oder nicht (*NO):

- die Namen der Einsprungstellen für die Subsystemroutinen INIT-, STOPCOM-, DEINIT- und CLOSE-CTRL-Routine.
- der Name der Einsprungstelle, die für die dynamische Identitätsprüfung herangezogen wird (DYNAMIC-CHECK-ENTRY)
- der Name der Schnittstellenversion für den Aufruf der INIT-, STOPCOM-, DEINIT- oder CLOSE-CTRL-Routinen (INTERFACE-VERSION)

MEMORY-ATTRIBUTES = *YES / *NO

Vereinbart, ob folgende Speicher-relevanten Informationen, die als Teil der Subsystemdefinition SSD-Object gespeichert sind, ausgegeben werden sollen (*YES) oder nicht (*NO):

- Speicherklasse (MEMORY-CLASS)
- Größe des benötigten Adressraums (SIZE)
- Anfangsadresse des Subsystemcodes (START-ADDRESS)
- Privilegierung und Zugriffsberechtigung bzgl. Adressraum (SUBSYSTEM-ACCESS)

RELATED-FILES = *YES / *NO

Vereinbart, ob Informationen über die Nebenkompenten des Subsystems geliefert werden sollen (*YES) oder nicht (*NO). In die Ausgabe eingeschlossen ist die Information, ob die Verwendung einer Rep-Datei für dieses Subsystem verpflichtend ist (REP-FILE-MANDATORY) und unter welcher Benutzerkennung die Nebenkompenten katalogisiert sind (INSTALLATION-USERID). Unter dem Begriff Nebenkompenten werden zusammengefasst:

- die Objektmoduldatei des Subsystems (LIBRARY)
- die Meldungsdatei (MESSAGE-FILE)
- die Syntaxdatei (SYNTAX-FILE)
- die Informationsdatei des Subsystems (SUBSYSTEM-INFO-FILE)
- die Rep-Datei (REP-FILE)

LINK-ATTRIBUTES = *YES / *NO

Vereinbart, ob die Informationen aus dem SSD-Object zu lesen sind (*YES) oder nicht (*NO), die in Bezug auf das Binden und Laden des Subsystems gespeichert sind:

- der Name des zum Laden benötigten Bindemoduls/ENTRY/CSECT (LINK-ENTRY)
- die Einbindung der Autolink-Funktion (AUTOLINK)
- die Informationen über das Verhalten bei nicht auflösbaren Externverweisen (UNRESOLVED)
- die Einbindung des Prüflaufs für Referenz-Subsysteme (CHECK-REFERENCE)

REFERENCE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, zu denen Adressbeziehungen bestehen, bei der Ausgabe der SSD-Datei-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

DEPENDENCE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, zu denen Abhängigkeitsbeziehungen bestehen, bei der Ausgabe der SSD-Object-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

ADDR-SPACE-RELATION = *YES / *NO

Vereinbart, ob die Liste der Subsysteme, mit denen eine Adressraumüberschneidung vermieden werden muss, bei der Ausgabe der SSD-Object-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

HOLDER-TASK-INFO = *YES / *NO

Vereinbart, ob die Identifikation der Holdertask und die Liste der Subsysteme, die in einer gemeinsamen Holdertask anzulegen sind, bei der Ausgabe der SSD-Object-Informationen berücksichtigt werden soll (*YES) oder nicht (*NO).

SUBSYSTEM-ENTRIES = *YES / *NO

Vereinbart, ob die Liste der bei der Definition des Subsystems vereinbarten Auftragseingänge und deren folgende Eigenschaften aus dem SSD-Object gelesen werden soll (*YES) oder nicht (*NO):

- Typ des vereinbarten Auftragseingangs (MODE)
- Nummer der Routine (bei SVC oder System-Exit) (NUMBER)
- die Funktionsnummer der Einsprungstelle (FUNCTION-NUMBER)
- die Version der Funktionsnummer (FUNCTION-VERSION)
- die Information über Aufruf durch System-Exit-Routinen (CALL-BY-SYSTEM-EXIT)
- die Privilegierung und Zugriffsberechtigung bezüglich Einsprungstellen (CONNECTION-ACCESS und CONNECTION-SCOPE)

OUTPUT =

Vereinbart den Ausgabeort der von der Anweisung generierten Informationen.

OUTPUT = *SYSLST(...)

Voreinstellung: die Meldungen sollen nach SYSLST ausgegeben werden.

SYSLST-NUMBER =

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

SYSLST-NUMBER = *STD

Voreinstellung: die Ausgabe soll in die Standard-Systemdatei SYSLST erfolgen.

SYSLST-NUMBER = <integer 1..99>

Die Ausgabe soll in eine Systemdatei aus der Menge SYSLST01 bis SYSLST99 erfolgen, deren Nummer hier anzugeben ist.

OUTPUT = SYSOUT

Die Meldungen werden auf der Datensichtstation ausgegeben.

START-CATALOG-CREATION

Name eines statischen Subsystemkatalogs vereinbaren

Funktionsbeschreibung

Mit dieser Anweisung wird der Name eines neuen statischen Subsystemkataloges vereinbart. In diese Datei können dann die SSD-Objects, die die Definitionen der Subsysteme beinhalten, eingebunden werden.

START-CATALOG-CREATION wird mit einer Fehlermeldung abgewiesen, wenn eine Datei gleichen Namens bereits existiert oder zuvor ein gleich lautender Katalogeintrag mit CREATE-FILE erzeugt wurde. Die Definition wird mit der Sicherung des neuen Kataloges (Anweisung SAVE-CATALOG, [Seite 238](#)) abgeschlossen.

Folgen allerdings zwei Anweisungen START-CATALOG-CREATION oder START-CATALOG-MODIFICATION aufeinander, erscheint in Dialogaufträgen die Meldung:

```
SCM0011  WOLLEN SIE WIRKLICH DEN KATALOG IM SPEICHER ÜBERSCHREIBEN? (Y/N)
```

Auf die Antwort **Y** wird der belegte Speicherplatz freigegeben und zuvor vereinbarte Eigenschaften des Katalogs werden verworfen.

Auf die Antwort **N** wird die zweite START-CATALOG-CREATION-Anweisung ungültig und die erste Anweisung kann mit SAVE-CATALOG abgeschlossen werden.

Bei Stapelaufträgen wird implizit die Antwort **Y** gegeben.

Format

START-CATALOG-CREATION
CATALOG-NAME = * <u>STD</u> / <filename 1..51 without-gen-vers>

Operandenbeschreibung

CATALOG-NAME =

Name des zu erzeugenden statischen Subsystemkataloges.

Existiert kein Katalog mit diesem Namen, wird die Anweisung abgewiesen.

CATALOG-NAME = *STD

Voreinstellung ist die Datei SYS.SSD.CAT.X auf dem Home-Pubset.

CATALOG-NAME = <filename 1..51 without-gen-vers>

Vollqualifizierter Dateiname.

START-CATALOG-MODIFICATION

Statischen Subsystemkatalog verändern

Funktionsbeschreibung

Mit dieser Anweisung kann ein bereits vorhandener statischer Subsystemkatalog modifiziert werden. In diese Datei können dann neue SSD-Objects, die die Definitionen der Subsysteme beinhalten, eingebunden werden.

Die Definition eines veränderten Katalogs wird mit den Anweisungen CHECK-CATALOG (siehe [Seite 196](#)) und SAVE-CATALOG (siehe [Seite 238](#)) abgeschlossen.

START-CATALOG-MODIFICATION wird abgewiesen, wenn der Name der angegebenen Datei nicht existiert oder zuvor nicht mit der Anweisung START-CATALOG-CREATION erzeugt wurde.

Folgen zwei Anweisungen START-CATALOG-MODIFICATION aufeinander oder wird START-CATALOG-MODIFICATION nach der Anweisung START-CATALOG-CREATION angegeben ohne ein abschließendes SAVE-CATALOG, erscheint in Dialogaufträgen die Meldung:

```
SCM0011 WOLLEN SIE WIRKLICH DEN KATALOG IM SPEICHER ÜBERSCHREIBEN ? (Y/N)
```

Auf die Antwort **Y** wird der belegte Speicherplatz freigegeben und zuvor veränderte Eigenschaften des Kataloges werden verworfen.

Auf die Antwort **N** wird die zweite START-CATALOG-MODIFICATION-Anweisung ungültig und die erste Anweisung kann mit SAVE-CATALOG abgeschlossen werden.

Bei Stapelaufträgen wird implizit die Antwort **Y** gegeben.

Format

START-CATALOG-MODIFICATION
CATALOG-NAME = <filename 1..54 without-gen-vers>

Operandenbeschreibung

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

Vollqualifizierter Dateiname des Subsystemkataloges, der verändert werden soll.

START-SSD-CREATION

SSD-Object zur Aufnahme von Subsystemdefinitionen generieren

Funktionsbeschreibung

Mit der Anweisung START-SSD-CREATION wird die Generierung eines SSD-Objects zur Aufnahme von Subsystemdefinitionen angestoßen. Der Anwender muss den Namen des SSD-Objects, den Namen der Datei, in der es gespeichert werden soll sowie die Namen der vom Subsystem benötigten/referenzierten Dateien angeben.

START-SSD-CREATION wird mit einer Fehlermeldung abgewiesen, wenn die bei SSD-FILE-NAME anzugebende Datei bereits existiert.

Die Anweisung SAVE-SSD (siehe [Seite 240](#)) zur Sicherung aller vereinbarten Eigenschaften kann nur nach erfolgreicher Abarbeitung der Anweisung SET-SUBSYSTEM-ATTRIBUTES eingegeben werden.

Format

START-SSD-CREATION

```

SSD-NAME = <name 1..8>
, VERSION = <integer 1..999>
, DOMAIN = <structured-name 1..13>
, CORRECTION = *NONE / list-poss(20): <alphanumeric-name 8..8>
, SSD-FILE-NAME = <filename 1..51 without-userid without-gen-vers>
, BLOCK-CONTROL-INFO = *STD / *WITHIN-DATA-BLOCK

```

Operandenbeschreibung

SSD-NAME = <name 1..8>

Name des SSD-Objects.

Der Name muss folgender Konvention genügen: die ersten drei Buchstaben entsprechen der Meldungsklasse des Produkts, die letzten drei Buchstaben lauten „SSC“.

VERSION = <integer 1..999>

Version des SSD-Objects.

DOMAIN = <structured-name 1..13>

Bezeichnung der MONSYS-Domäne des SSD-Objects. Über den Namen dieser Domäne ist systemintern eine eindeutige, konsistente Zuordnung aller Komponenten zu dem Subsystem zu realisieren.

CORRECTION =

Gibt an, ob im SSD-Object für die Subsysteme PULS-Problemmeldungen eingearbeitet worden sind.

CORRECTION = *NONE

Voreinstellung: in das SSD-Object wurden keine PULS-Problemmeldungen eingearbeitet.

CORRECTION = list-poss(20) <alphanum-name 8..8>

Liste der maximal 20 Problemmeldungen, die im SSD-Object korrigiert sind.

SSD-FILE-NAME = <filename 1..51 without-userid without-gen-vers>

Name der neu anzulegenden ISAM-Datei, in der das SSD-Object gesichert werden soll. Existiert der Dateiname bereits, wird die Anweisung zurückgewiesen.

BLOCK-CONTROL-INFO =

Gibt an, in welchem Dateiformat die Datei für das SSD-Object angelegt werden soll.

BLOCK-CONTROL-INFO = *STD

Voreinstellung: SSCM versucht zuerst, das SSD-Object als K-Datei (Blockformat PAMKEY) mit der Blocklänge 1 anzulegen (BUFFER-LENGTH=*STD(SIZE=1)).

Im Fehlerfall wird versucht, das SSD-Object mit der Blocklänge 2 anzulegen.

Ist auch dieser Versuch nicht erfolgreich, wird das SSD-Object als NK-Datei mit dem Blockformat DATA und der Blocklänge 2 angelegt.

BLOCK-CONTROL-INFO = *WITHIN-DATA-BLOCK

SSCM legt das SSD-Object als NK-Datei im Blockformat DATA und der Blocklänge 2 an.

Hinweise

Im Anschluss an eine fehlerfreie Bearbeitung der Anweisung START-SSD-CREATION sind nur die folgenden Anweisungen möglich. Die Anweisungen mit (*) können auch mehrmals verwendet werden, um jeweils unterschiedliche Subsysteme (aber nicht unterschiedliche Versionen eines Subsystems) zu definieren.

- ADD-SUBSYSTEM-ENTRIES (*)
- ASSIGN-HOLDER-TASK (*)
- SEPARATE-ADDRESS-SPACE (*)
- SET-SUBSYSTEM-ATTRIBUTES (*)
- SHOW-SSD

4.4 Installation von SSCM

Es wird empfohlen, SSCM mit IMON zu installieren.

Mit SSCM V2.3 werden folgende Dateien ausgeliefert:

- die Modulbibliothek SYSLNK.SSCM.023, die das Bindelademodul SSCM enthält
- die System-Syntaxdatei SYSSDF.SSCM.023 mit Anweisungen für SSCM
- die Benutzer-Syntaxdatei SYSSDF.SSCM.023.USER mit Anweisungen für SSCM
- die Prozedur SYSPRC.SSCM.023, die SSCM mit START-PROGRAM startet
- die Subsystem-Deklarationsdatei SYSSSC.SSCM.023.120, für den Einsatz in BS2000/OSD-BC V3.0
- die Meldungsdatei SYSMES.SSCM.023 für den Einsatz ab BS2000/OSD-BC V3.0
- die Rep-Datei SYSREP.SSCM.023 (sofern vorhanden)

Folgende Bedingungen müssen für die Installation von SSCM V2.3 erfüllt sein:

1. Die SSCM-Subsystemdefinition muss im Subsystemkatalog aufgenommen sein.
2. Die Bibliothek SYSLNK.SSCM.023 muss unter der Installations-Benutzerkennung katalogisiert sein, die mit ADD-CATALOG-ENTRY oder SET-SUBSYSTEM-ATTRIBUTES bei der Aufnahme von SSCM in den Subsystemkatalog benannt wurde (standardmäßig die System-Benutzerkennung, deren Dateien i.a. mit „\$.“ beginnen).
3. Mit dem Kommando START-SUBSYSTEM SSCM wird das Subsystem SSCM aktiviert und mit dem Kommando START-SSCM gestartet.
Es gehören auch die Meldungsdatei, die Syntaxdatei und die Rep-Datei dazu; sie werden automatisch aktiviert. Die System-Syntaxdatei SYSSDF.SSCM.023 muss das Attribut SHARE=SPECIAL besitzen.

Wird die Prozedur SYSPRC.SSCM.023 zum Aufruf von SSCM V2.3 benutzt, entfallen die Punkte 1 und 3.

Koexistenz von SSCM-Versionen

Die SSCM-Versionen V1.0 bis V2.3 können gleichzeitig im Subsystemkatalog definiert sein. Die Version kann mit dem Kommando SELECT-PRODUCT-VERSION ausgewählt werden. Das Kommando START-SSCM startet dann die gewählte Version.

Zu Fragen über Kompatibilität und Portabilität zwischen den verschiedenen BS2000/OSD-, SSCM- und DSSM-Versionen siehe [Seite 28](#).

4.5 Beispiele

Die nachfolgenden Beispiele stellen Folgen von SSCM-Anweisungen dar, die die Vorgehensweise zur Erstellung und Änderung bestimmter Objekte veranschaulichen sollen.

4.5.1 Erstellung eines SSD-Objects

```

/START-SSCM
//START-SSD-CREATION  SSD-NAME=SS1SSC,VERSION=001,DOMAIN=DSSM,
    CORRECTION=*NONE,SSD-FILE-NAME=SYSSSC.SS1.001 _____ (1)
//SET-SUBSYSTEM-ATTRIBUTES  SUBSYSTEM-NAME=SS1(VERSION=1.0),
    SUBSYSTEM-ENTRIES=ENTRY1(CONNECTION-ACCESS=*ALL,
        CONNECTION-SCOPE=*TASK),
    REFERENCED-SUBSYSTEM=SS2(LOWEST-VERS=01.0,HIGHEST-VERSION=02.5),
    RELATED-SUBSYSTEM=SS3(LOWEST-VERSION=02.0,HIGHEST-VERSION=02.0),
    LINK-ENTRY=E1 _____ (2)
//SEPARATE-ADDRESS-SPACE  SUBSYSTEM-NAME=SS1,
    FROM-SUBSYSTEMS=(SS5,SS6,SS9) _____ (3)
//ASSIGN-HOLDER-TASK  TYPE=*SHARED-HOLDER(
    BY-SUBSYSTEMS=(SS1,SS2,SS3),TSN=*BY-DSSM) _____ (4)
//SAVE-SSD _____ (5)

```

- (1) Deklaration des SSD-Objects.
- (2) Definition der wichtigsten Attribute des Subsystems SS1: der Auftragseingänge, der Referenzen und Abhängigkeiten.
- (3) Bestimmung der Subsysteme, die keinen gemeinsamen Adressraum mit SS1 haben dürfen; im Beispiel die Subsysteme SS5, SS6 und SS9.
- (4) Bestimmung der Holdertask-Eigenschaften von SS1, SS2 und SS3.
- (5) Sichern des SSD-Objects, das die Definition des Subsystems SS1 enthält. Zusätzlich wird das SSD-Object in die bei (1) angegebene ISAM-Datei SYSSSC.SS1.001 abgespeichert.

4.5.2 Erstellung eines statischen Subsystemkatalogs

```
/START-SSCM  
//START-CATALOG-CREATION CATALOG-NAME=KAT1 _____ (1)  
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSC.SS1.001,  
    INSTALLATION-USERID=*UNCHANGED,CORRECTION-STATE=*UNCHANGED _____ (2)  
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSD.SS2.001,  
    INSTALLATION-USERID=*UNCHANGED,CORRECTION-STATE=*UNCHANGED _____ (3)  
//SET-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=SS3(VERSION=V2.5A00),  
    :  
    : _____ (4)  
//CHECK-CATALOG _____ (5)  
//SAVE-CATALOG CATALOG-NAME=*CURRENT, FORCED=*NO _____ (6)
```

- (1) Deklaration des statischen Subsystemkatalogs.
- (2) Einbringen der Subsystemdefinition in der Object-Form für SS1 in den Katalog.
- (3) Einbringen der Subsystemdefinition im UGEN-Format für SS2 in den Katalog.
- (4) Definition der Attribute des Subsystems SS3.
- (5) Prüfen des Katalogs.
- (6) Sichern des erzeugten statischen Subsystemkatalogs.

4.5.3 Änderung eines statischen Subsystemkatalogs

```

/START-SSCM
//START-CATALOG-MODIFICATION CATALOG-NAME=KAT1 _____ (1)
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSC.SS5,
    INSTALLATION-USERID=*UNCHANGED,CORRECTION-STATE=*UNCHANGED _____ (2)
//ADD-CATALOG-ENTRY FROM-FILE=SYSSSD.SS4,
    INSTALLATION-USERID=*UNCHANGED,CORRECTION-STATE=*UNCHANGED _____ (3)
//SET-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=SS1(V01.0A10),
    LINK-ENTRY=SSIENT,MEMORY-CLASS=*SYSTEM-GLOBAL(SUBSYSTEM-ACCESS=
    *SYSTEM),INIT-ROUTINE=INITROUT,DEINIT-ROUTINE=*DYNAMIC,
    INTERFACE-VERSION=INTVERS _____ (4)
//SEPARATE-ADDRESS-SPACE SUBSYSTEM-NAME=SS1,
    FROM-SUBSYSTEMS=(SS5,SS6) _____ (5)
//ASSIGN-HOLDER-TASK TYPE=*WORK-TASK(SUBSYSTEM-NAME=SS1,
    SUBSYSTEM-VERSION=V01.0A10) _____ (6)
//REMOVE-CATALOG-ENTRY SUBSYSTEM-NAME=SS7(VERSION=V03.2A00) _____ (7)
//MODIFY-SUBSYSTEM-ATTRIBUTES SUBSYSTEM-NAME=SS2(VERSION=V11.0A10),
    INSTALLATION-USERID=UIDXYZ,....
    : _____ (8)
//REMOVE-ADDR-SPACE-SEPARATION SUBSYSTEM-NAME=SS1,
    FROM-SUBSYSTEMS=SS2 _____ (9)
//MODIFY-WORK-TASK-ATTRIBUTE SUBSYSTEM-NAME=SS3,
    SUBSYSTEM-VERSION=00.1,WORK-TASK=*NO _____ (10)
//CHECK-CATALOG _____ (11)
//SAVE-CATALOG CATALOG-NAME=*CURRENT,FORCED=*NO _____ (12)

```

- (1) Angabe des Katalogs, dessen Inhalt geändert werden soll.
- (2) Hinzufügen einer Subsystemdefinition, die in einem SSD-Object enthalten ist.
- (3) Hinzufügen einer Subsystemdefinition, die in der alten DSSMGEN-Syntax (UGEN-Format) geschrieben wurde.
- (4)-(6) Hinzufügen eines neuen Subsystems unter Bestimmung seiner Attribute.
- (7) Löschen einer Subsystemdefinition aus dem Katalog.
- (8) Ändern einer Subsystemdefinition, die sich bereits im Katalog befindet.
- (9) Aufheben der strikten Adressraumteilung für beide Subsysteme.
- (10) Ändern der Arbeitstask-Attribute eines als Arbeitstask arbeitenden Subsystems.
- (11) Prüfen des Katalogs.
- (12) Sichern des geänderten statischen Subsystemkatalogs.

Literatur

Wenden Sie sich zum Bestellen von Handbüchern bitte an Ihre zuständige Geschäftsstelle.

- [1] **ADAM** (BS2000/OSD)
Abstract Device Access Method
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an Assembler-Programmierer

Inhalt

Mit der Zugriffsmethode ADAM können in einem ASSEMBLER-Programm Geräte angesprochen werden, die nicht durch die logischen Zugriffsmethoden des BS2000/OSD unterstützt werden. Im Handbuch sind die zur Programmierung benötigten ADAM-Makros beschrieben.

- [2] **AID** (BS2000)
Advanced Interactive Debugger
Basishandbuch
Benutzerhandbuch

Zielgruppe

Programmierer im BS2000

Inhalt

- Überblick über AID
- Beschreibung der Sachverhalte und Operanden, die für alle Programmiersprachen gleich sind
- Meldungen
- Gegenüberstellung von AID-IDA

Einsatz

Testen von Programmen im Dialog- und Stapelbetrieb

- [3] **ARCHIVE** (BS2000/OSD)
Benutzerhandbuch

Zielgruppe

- BS2000/OSD-Benutzer
- BS2000/OSD-Systemverwalter
- BS2000/OSD-Operator

Inhalt

Funktionen und Anweisungen des Programms ARCHIVE zur logischen Datensicherung. ARCHIVE dient zur Sicherung, Rekonstruktion und zum Transfer von Dateien und Jobvariablen.

- [4] **BLSSERV**
Bindelader-Starter
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an Software-Entwickler und geübte BS2000/OSD-Benutzer.

Inhalt

Es beschreibt die Funktionen, die Unterprogrammchnittstelle, die XS-Unterstützung und den Aufruf des Bindeladers DBL als Bestandteil des Subsystems BLSSERV. Daran anschließend sind die Kommandos zum Aufruf des Laders ELDE und die Migration vom DLL zum DBL beschrieben.

- [5] **BINDER** (BS2000/OSD)
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an Software-Entwickler

Inhalt

Es beschreibt die BINDER-Funktionen und enthält Beispiele dazu. Im Nachschlageteil sind die BINDER-Anweisungen und der Makroaufruf BINDER beschrieben.

- [6] **CALENDAR** (BS2000/OSD)
Benutzerhandbuch

Zielgruppe

Anwender und Systembetreuung.

Inhalt

Das Handbuch beschreibt das Produkt CALENDAR, mit dem Kalender mit komplexen Datumsangaben erzeugt werden können. Die Oberfläche bietet eine maskengesteuerte Dialogschnittstelle (Kalender-Editor) und eine Programmschnittstelle für Assembler und C.

- [7] **CRTE (BS2000/OSD)**
Common RunTime Environment
Benutzerhandbuch
Zielgruppe
Programmierer und Systemverwalter im BS2000/OSD
Inhalt
Beschreibung der gemeinsamen Laufzeitumgebung für COBOL85-, COBOL2000-, C-, und C++-Objekte sowie für „Fremdsprachenmix“:
– Komponenten des CRTE
– Programmkommunikationsschnittstelle ILCS
– Bindebeispiele
- [8] **DAB (BS2000/OSD)**
Disk Access Buffer
Benutzerhandbuch
Zielgruppe
Das Handbuch wendet sich an die Systembetreuung.
Inhalt
Nach einführenden Kapiteln über das DAB-Caching, die DAB-Cache-Medien und -Funktionen, werden die DAB-Kommandos ausführlich beschrieben.
Inhaltlicher Überblick:
– DAB-Caching, DAB-Medien, DAB-Funktionen
– DAB-Einsatzhinweise, Leistungsverhalten, Installation, Starten und Beenden
– DAB-Kommandos und -Meldungen
- [9] **BS2000/OSD-BC**
Dienstprogramme
Benutzerhandbuch
Zielgruppe
Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.
Inhalt
Es beschreibt die mit dem BS2000-Grundausbau BS2000/OSD-BC ausgelieferten Dienstprogramme.

[10] **Distributed Print Services** (BS2000/OSD)

Drucken in Computernetzen

Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an nichtprivilegierte Anwender, Geräteverwalter und die Systembetreuung des BS2000/OSD.

Inhalt

Beschrieben werden für die drei Anwendergruppen Konzept, Nutzung und Verwaltung von Distributed Print Services. Beispiele zeigen die Einsatzmöglichkeiten von Distributed Print Services.

[11] **DRV** (BS2000/OSD)

Dual Recording by Volume

Benutzerhandbuch

Zielgruppe

Systembetreuung, Operating und nichtprivilegierte Anwender

Inhalt

Das Manual beschreibt das Daten-Aufzeichnungsverfahren DRV (Dual Recording by Volume). Der Einsatz von DRV im RZ erhöht die Verfügbarkeit der auf Platten gespeicherten Daten. Jeder Schreibauftrag wird auf beiden Platten und jeder Leseauftrag auf der Platte mit der kürzesten Zugriffszeit ausgeführt. Im Handbuch sind alle für den Einsatz, die Steuerung und Überwachung von DRV notwendigen Kommandos und Makroerweiterungen beschrieben, nebst Kapiteln über die Installation von DRV, das Arbeiten mit DRV, Fehlerbehandlung, DRV-Abrechnungssatz usw.

[12] **EDT** (BS2000/OSD)

Anweisungen

Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an EDT-Einsteiger und EDT-Anwender.

Inhalt

Das Handbuch beschreibt das Bearbeiten von SAM- und ISAM-Dateien, Elementen aus Programm-Bibliotheken und POSIX-Dateien. Es enthält weiter eine Beschreibung der Arbeitsmodi, Kurzanweisungen, EDT-Prozeduren und Anweisungen des EDT.

- [13] **BS2000/OSD-BC**
Einführung in das DVS
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an den nichtprivilegierten Anwender und an die Systembetreuung.

Inhalt

Es beschreibt die Dateiverwaltung und -verarbeitung im BS2000.

Themenschwerpunkte:

- Datenträger und Dateien
- Datei- und Katalogverwaltung
- Datei- und Datenschutz
- OPEN-, CLOSE-, EOVS-Verarbeitung
- DVS-Zugriffsmethoden (SAM, ISAM,...)

- [14] **BS2000/OSD-BC**
Einführung in die Systembetreuung
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.

Inhalt

Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbaus enthalten: Systemeinleitung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei-, Pubset- und Systemzeit-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

- [15] **FDDRL (BS2000/OSD)**
Benutzerhandbuch

Zielgruppe

BS2000/OSD-Systembetreuer und -Operateur

Inhalt

FDDRL sichert und restauriert physikalisch den Inhalt ganzer Platten und Pubsets. Das Handbuch beschreibt die Funktionen und Anweisungen des Programms FDDRL zur physikalischen Datensicherung im Rechenzentrum.

- [16] **HSMS / HSMS-SV (BS2000/OSD)**
Hierarchisches Speicher Management System
2 Bände
Benutzerhandbücher
- Zielgruppe*
- BS2000/OSD-Benutzer
 - BS2000/OSD-Systembetreuer
 - HSMS-Verwalter
- Band 1** enthält die Beschreibung der Funktionen, der Verwaltung und der Installation
- Beschreibung der Funktionen Datensicherung, Langzeitarchivierung, Verdrängung (Migration) und Datentransfer
 - Verwaltung, Aufruf, Ablauf und Installation von HSMS
 - Meldungen von HSMS
- Band 2** enthält die Beschreibung der HSMS-Anweisungen in alphabetischer Reihenfolge
- [17] **IMON (BS2000/OSD)**
Installationsmonitor
Benutzerhandbuch
- Zielgruppe*
- Das Handbuch wendet sich an die Systembetreuung des Betriebssystems BS2000/OSD.
- Inhalt*
- Das Handbuch beschreibt die Installation und Verwaltung von BS2000-Software mit dem Installationsmonitor IMON und seinen drei Komponenten IMON-BAS, IMON-GPN und IMON-SIC. In zwei Beispielkapiteln wird die Installation (standard und kundenspezifisch) mit der Komponente IMON-BAS für Systeme mit BS2000/OSD V2.0 und ab BS2000/OSD V3.0/V4.0 ausführlich dargestellt.
- [18] **JV (BS2000/OSD)**
Jobvariablen
Benutzerhandbuch
- Zielgruppe*
- Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.
- Inhalt*
- Es beschreibt die Verwaltung und die verschiedenen Einsatzmöglichkeiten von Jobvariablen. Die Kommandobeschreibungen sind getrennt nach den Funktionsbereichen der JVs aufgeführt. Die Makroaufrufe sind in einem eigenen Kapitel beschrieben.

- [19] **BS2000/OSD-BC
Kommandos (Band 1-5)**
Benutzerhandbücher

Zielgruppe

Die Handbücher wenden sich an nicht-privilegierte Anwender und die Systembetreuung.

Inhalt

Sie enthalten die Kommandos (Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien. Die Einleitung gibt Hinweise zur Kommandoeingabe.

- [20] **BS2000/OSD-BC
Kommandos Band 6, Ausgabe in S-Variablen und SDF-P-BASYS**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an Programmierer und Anwender, die Prozeduren erstellen.

Inhalt

Band 6 enthält die tabellarische Darstellung aller S-Variablen, die von den SHOW-Kommandos bei einer strukturierten Ausgabe mit Werten versorgt werden. Weitere Kapitel:

- Einführung in das Arbeiten mit S-Variablen
- SDF-P-BASYS V2.2A

- [21] **LMS (BS2000/OSD)
SDF-Format**
Benutzerhandbuch

Zielgruppe

BS2000-Anwender

Inhalt

Beschreibung der Anweisungen zum Erstellen und Verwalten von PLAM-Bibliotheken und darin enthaltenen Elementen.

Häufige Anwendungsfälle werden an Hand von Beispielen erklärt.

[22] **MAREN (BS2000/OSD)**

Band 1 und 2

Benutzerhandbücher

Zielgruppe

Das Handbuch wendet sich an alle BS2000-Anwender, RZ-Organisatoren, Operatoren und Systembetreuer.

Inhalt

Band 1 enthält eine Einführung in die Arbeit mit MAREN. Es werden u.a. folgende Themen beschrieben:

- Der MAREN-Katalog und seine Migration
- Die Verwaltung von MBKs und Lagerorten
- Die Unterstützung von Programmen zur Datensicherung durch MAREN
- Die Unterstützung von Archivsystemen durch MAREN

Band 2 enthält Übersichten, Schnittstellenbeschreibungen und Beispiele zur Arbeit mit MAREN, aufgeteilt in einen privilegierten und einen nichtprivilegierten Teil.

- Im privilegierten Teil werden die Verwaltung von MAREN mit MARENADM, die Konfiguration und Installation von MAREN, die Erstellung und Migration des MAREN-Katalogs sowie die Fehlerbehandlung beschrieben.
- Im nichtprivilegierten Teil werden die MAREN-Kommandos und das Benutzerprogramm MAREN beschrieben.

Der Anhang enthält u.a. die Meldungsangaben und globale Parameter.

[23] **HIPLEX MSCF (BS2000/OSD)**

BS2000-Rechner im Verbund

Benutzerhandbuch

Zielgruppe

Systembetreuung, Operating und nichtprivilegierte Anwender

Inhalt

HIPLEX MSCF (BS2000) ermöglicht, mehrere Zentraleinheiten des BS2000/OSD zu einem LCS-, CCS-, SPVS- bzw. XCS-Rechnerverbund zusammenzuschließen. Das Handbuch informiert über HIPLEX MSCF (BS2000), seine Anwendungsmöglichkeiten, Einsatzvoraussetzungen und Kommandos.

[24] **PCA (BS2000/OSD)**

Peripheral Cache Administrator

Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an die Systembetreuung des Betriebssystems BS2000/OSD.

Inhalt

Das Handbuch erläutert die PCA-Hardware, die Caching-Modi und -Verfahren, den effizienten Einsatz von PCA, sein Leistungsverhalten, seine Funktionen, Kommandos und Meldungen.

- [25] **PCS** (BS2000/OSD)
Performance Control Subsystem
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an die Systembetreuung.

Inhalt

Das Handbuch beschreibt den Einsatz des Performance Control Subsystems (PCS) zur optimalen Leistungssteuerung einer Rechenanlage gemäß dem Task-Kategorien-Konzept. Nach einer Einführung in das PCS-Konzept wird die Bedienung von PCS beschrieben. Die Einstellungen der PCS-Parameter werden ausführlich dargestellt. Zahlreiche Tabellen liefern wichtige Einstellhinweise.

- [26] **POSIX** (BS2000/OSD)
Grundlagen für Anwender und Systemverwalter
Benutzerhandbuch

Zielgruppe

BS2000-Systemverwalter, POSIX-Verwalter, BS2000-Benutzer, Benutzer von UNIX-/SINIX-Workstations

Inhalt

- Einführung und Arbeiten mit POSIX
- BS2000-Softwareprodukte im Umfeld von POSIX
- POSIX installieren
- POSIX steuern und Dateisysteme verwalten
- POSIX-Benutzer verwalten
- BS2000-Kommandos für POSIX

- [27] **POSIX** (BS2000/OSD)
Kommandos
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an alle Benutzer der POSIX-Shell.

Inhalt

Dieses Handbuch ist ein Nachschlagewerk. Es beschreibt das Arbeiten mit der POSIX-Shell sowie die Kommandos der POSIX-Shell in alphabetischer Reihenfolge.

- [28] **PRM (BS2000/OSD)**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an SPOOL-Anwender, Systembetreuung und RSO-Geräteverwalter.

Inhalt

Das Handbuch beschreibt das Dienstprogramm PRM zur Erstellung und Verwaltung von Druckressourcen für BS2000-SPOOL. Es werden die zwei Benutzeroberflächen des Dienstprogramms beschrieben: die SDF-Anweisungen für Dialog- und Stapelbetrieb und die FHS-basierte Menüoberfläche für den Dialogbetrieb.

- [29] **PROP-XT (BS2000/OSD)**
Programmiertes Operating mit komfortablen Sprachmitteln von SDF-P
Produktthandbuch

Zielgruppe

Systemverwalter, Systembediener, Arbeitsvorbereiter und mit gleichartigen Aufgaben betraute Personen.

Inhalt

PROP-XT ermöglicht programmiertes Operating durch vom Anwender mit den komfortablen Sprachmitteln von SDF-P erstellte Administrationsprozeduren sowie die Automatisierung einfacher Administrationstätigkeiten über spezielle Administrationskommandos.

- [30] **RFA (BS2000/OSD)**
Remote File Access
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an alle BS2000/OSD-Anwender.

Inhalt

Das Handbuch erläutert die Grundlagen des RFA-Konzepts. Alle RFA-Kommandos sind ausführlich beschrieben, ebenso die Besonderheiten von DVS-Kommandos beim Zugriff auf Dateien in einem fernen System mit RFA.

- [31] **RSO (BS2000/OSD)**
Remote SPOOL Output
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an nichtprivilegierte Benutzer, RSO-Geräteverwalter, SPOOL-Verwalter und Systembetreuer des BS2000/OSD.

Inhalt

Für die einzelnen Anwendergruppen werden die Aufgaben und Möglichkeiten zur Nutzung und Steuerung von dezentralen Druckern (RSO-Drucker) beschrieben. Das Handbuch enthält die Druckermerkmale aller RSO-Drucker.

- [32] **SDF-A (BS2000/OSD)**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an erfahrene BS2000-Benutzer und an die Systemverwaltung.

Inhalt

Es beschreibt, wie Syntaxdateien bearbeitet werden und erklärt die SDF-A-Funktionen anhand von Beispielen. Die SDF-A-Anweisungen sind in alphabetischer Reihenfolge aufgeführt.

Über die im Handbuch beschriebene SDF-Programmschnittstelle können auch Benutzerprogramme die SDF-Benutzeroberfläche nutzen. Mit dem Dienstprogramm SDF-SIM kann die Syntax von Kommandos und Anweisungen getestet werden.

- [33] **SDF-CONV (BS2000/OSD)**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an alle BS2000-Anwender.

Inhalt

Prozedurformat und Kommandosprache von Prozeduren können wie folgt umgewandelt werden:

- von ISP- in SDF-Kommandosprache
- von Nicht-S- in S-Prozedurformat
- gleichzeitige Umwandlung von Kommandosprache und Prozedurformat.

- [34] **SDF-P (BS2000/OSD)**
Programmieren in der Kommandosprache
Benutzerhandbuch
- Zielgruppe*
BS2000-Anwender und Systembetreuung.
- Inhalt*
SDF-P ist eine strukturierte Prozedursprache im BS2000. Nach einführenden Kapiteln zum Prozedur- und Variablenkonzept werden Kommandos, Funktionen und Makros ausführlich beschrieben.
- [35] **SDF (BS2000/OSD)**
SDF-Verwaltung
Benutzerhandbuch
- Zielgruppe*
Das Handbuch wendet sich an die Systemverwaltung und an erfahrene BS2000-Benutzer.
- Inhalt*
Es beschreibt, wie SDF mit Hilfe von SDF-Kommandos und den Dienstprogrammen SDF-I, SDF-U und SDF-PAR installiert und verwaltet wird. Die Anweisungen von SDF-I, SDF-U und SDF-PAR sind vollständig beschrieben.
- [36] **SECOS (BS2000/OSD)**
Security Control System
Benutzerhandbuch
- Zielgruppe*
- BS2000-Systemverwalter
 - BS2000-Anwender, die den erweiterten Zugriffsschutz für Dateien nutzen
- Inhalt*
Leistung und Anwendung der Funktionseinheiten:
- SRPM (Privilegien und Betriebsmittel verwalten)
 - SRPMSSO (Single Sign On)
 - GUARDS (Zugriffsbedingungsverwaltung und -auswertung für Objekte)
 - GUARDEF (Default Protection, Standardschutz)
 - GUARDCOO (Co-owner Protection, Miteigentümerschutz)
 - SAT (Protokollierung und Auswertung sicherheitsrelevanter Daten, Ereignisüberwachung mit Alarmfunktion).

- [37] **SM2 (BS2000/OSD)**
Software Monitor
Band 1: Verwaltung und Bedienung
- Zielgruppe*
Anwender und Systembetreuung
- Inhalt*
Das Messsystem SM2 liefert dem Benutzer statistische Daten über die Leistung des DV-Systems und die Auslastung der Betriebsmittel. Im Band 1 werden die Bedienung des Messmonitors SM2, die SM2-Messprogramme und die SM2-Bildschirmreports beschrieben.
Zur Auswertung und Darstellung der SM2-Messwerte siehe Band 2 des Handbuchs.
- [38] **SPOOL (BS2000/OSD)**
Benutzerhandbuch
- Zielgruppe*
Das Handbuch wendet sich an nichtprivilegierte Anwender, den Spool & Print - Verwalter, den RSO-Geräteverwalter und die Systembetreuung.
- Inhalt*
Es wird der Betrieb von SPOOL beschrieben.
- [39] **SPOOLAPA PRINTING SYSTEM**
Drucken mit APA
Allgemeine Beschreibung
- Zielgruppe*
Systembetreuer, DV-Organisatoren und DV-Manager
- Inhalt*
Dargestellt wird das grundlegend neue Druckkonzept APA (All Points Adressable) für das BS2000. APA ermöglicht im Bereich der Hochleistungsdrucker völlig neue Gestaltungsmöglichkeiten wie freies Positionieren von vorgefertigten Bausteinen usw.
- [40] **BS2000/OSD-BC**
Systeminstallation
Benutzerhandbuch
- Zielgruppe*
Das Handbuch wendet sich an die BS2000/OSD-Systemverwaltung.
- Inhalt*
Beschrieben wird die Generierung der Hardware-Konfiguration mit UGEN und die Installationsdienste. Letztere beinhalten die Plattenorganisation mit MPVS, die Installation von Datenträgern mit dem Dienstprogramm SIR und das Subsystem IOCFCOPY.

[41] **BS2000/OSD-BC V5.0**
Rechenzentrum-Taschenbuch, Band 1

Zielgruppe

Das Taschenbuch wendet sich an die Systembetreuung im BS2000-Rechenzentrum.

Inhalt

Das Taschenbuch soll die Systembetreuung bei der täglichen Arbeit im BS2000-RZ unterstützen. Es enthält in Form von Übersichten und Auszügen aus anderen Handbüchern knappe, konzentrierte Informationen zum BS2000-Betrieb; u.a. die Abrechnungssätze, den Parameterservice und die Syntax der RZ-Dienstprogramme.

[42] **BS2000/OSD**
Technische Beschreibung

Zielgruppe

BS2000-Anwender und -Betreiber, die sich für den technischen Hintergrund ihres Systems interessieren (Softwareentwickler, Systemanalytiker, RZ-Leiter, Systemverwalter).

Informatiker, die ein konkretes „General-Purpose“-Betriebssystem studieren wollen.

Stichwörter

\$ESMINT(I) (Makro) [217](#), [260](#)
\$PBBND1 (Makro) [207](#), [228](#), [249](#), [270](#)
\$TSOS.SYS.SSD.CAT.X (Standardname des
Subsystemkatalogs) [65](#)

A

Abbruch von Subsystem-belegenden Tasks [80](#),
[162](#)

Abhängigkeitsbeziehungen
festlegen und verändern [231](#)
löschen [232](#)
zwischen Subsystemen [48](#), [272](#)

Abnormale Beendigung von DSSM [71](#)

Abrechnung
Satzbeschreibung [67](#)
Satzstruktur [67](#)

Abrechnungssätze [66](#)

Abrechnungssystem [siehe Abrechnung](#)

Accounting [siehe Abrechnung](#)

ADD-CATALOG-ENTRY (SSCM-
Anweisung) [182](#), [183](#)

ADD-SUBSYSTEM (DSSM-Kommando) [74](#), [75](#)

ADD-SUBSYSTEM-ENTRIES (SSCM-
Anweisung) [182](#), [186](#)

Adressbeziehungen
angeben und bearbeiten [228](#)
ausgeben [278](#), [284](#)
prüfen [197](#)
zwischen Subsystemen [47](#), [271](#)

Adressraum
Adressraum-Haushalt von DSSM [57](#)
bestimmen [53](#)
eines Subsystems freigeben [110](#)
eines Subsystems vereinbaren [225](#), [268](#)

Überschneidungen [235](#), [241](#)
Verlagerung [58](#)
Verteilung [241](#)

Adressraum-Streifen [56](#)

Aktivieren
eines lokalen Subsystems [150](#)
von Subsystemen [39](#), [153](#), [211](#)

Aliasname [8](#)
alphanum-name (Datentyp) [9](#)

Anhalten
von Subsystemen [44](#), [80](#)
von Subsystemen (CLOSE-CTRL-
Routine) [215](#), [257](#)

Anschluss an ein Subsystem [42](#)

Anweisungen
Syntaxdarstellung [5](#)
von SSCM [182](#)

Arbeitstask [49](#), [58](#), [193](#)

ASSIGN-HOLDER-TASK (SSCM-
Anweisung) [60](#), [182](#), [193](#)

asynchrone Aktivitäten [38](#)

Auftragsbeziehungen zwischen Subsystemen [47](#)

Auftragseingänge
ausgeben [279](#), [284](#)
definieren [263](#)
definieren oder verändern [220](#)
eines Subsystems [24](#), [186](#)
löschen [225](#)

Auftragssteuerung [142](#)

Austausch von Subsystemversionen [45](#)

AUTOLINK [105](#)

B

- Bedienstation 218
- Beenden
 - eines Subsystems 162
 - von Aufträgen (STOPCOM-Routine) 215, 258
 - von SSCM 181
- Beispiele
 - Ausgabe in eine S-Variable 173
 - Laden eines lokalen Subsystemkatalogs 84
 - MODIFY-SUBSYSTEM-PARAMETER 109
 - Nutzung von lokalen Subsystemen 170
 - REMOVE-SUBSYSTEM 112
 - SAVE-SUBSYSTEM-CATALOG 119
 - SHOW-SUBSYSTEM-INFO 140
 - SHOW-SUBSYSTEM-STATUS 149
 - START-LOCAL-SUBSYSTEM 152
 - STOP-LOCAL-SUBSYSTEM 161
 - UNLOAD-LOCAL-SUBSYSTEM-CATALOG 167
 - Verteilung von Subsystemen auf Holdertasks 78
- Benutzeradressraum 56
 - Laden eines lokalen Subsystemkatalogs 83
 - lokalen Subsystemkatalog entladen 166
 - lokales Subsystem laden 150
- Beziehungen zu anderen Subsystemen 47, 77
- BIND (Makro) 48, 59
- Bindelademodul SSCM 291
- Bindemodul 130
- Binden und Laden eines Subsystems 53

C

- cat (Zusatz zu Datentypen) 20
- cat-id (Datentyp) 9
- CHECK-CATALOG (SSCM-Anweisung) 182, 196
- Coexistence-Modus 219, 262
- command-rest (Datentyp) 9
- compl (Zusatz zu Datentypen) 15
- composed-name (Datentyp) 9
- CONNECTION-ACCESS (Ausgabefeld) 139, 280
- CONSLOG-Protokolldatei 63

- Control Program (CP) 228, 271
- Copyright 128
- corr (Zusatz zu Datentypen) 20, 21
- CREATED (Subsystem-Zustand) 50
- CREATION-TIME (Ausgabefeld) 139, 280
- CSECT 130
- c-string (Datentyp) 9

D

- Darstellungsmittel 4
- date (Datentyp) 9
- Dateien zur Inbetriebnahme von DSSM 63
- Datenbereich 57
 - eines Subsystems 41
- Datentypen SDF 5, 9
 - Zusätze 6
- Deaktivieren
 - eines lokalen Subsystems 159
 - von Subsystemen 44, 162
 - von Subsystemen (CLOSE-CTRL-Routine) 215, 257
- Definition
 - der Subsystem-Konfiguration 38
 - eines Subsystems (im SSD-Object) 26
 - eines Subsystems in Katalog aufnehmen 183
 - eines Subsystems löschen 237
- Deinitialisieren von Subsystemen (DEINIT-Routine) 216, 259
- Deklaration eines Subsystems 38
- Deklarationsdatei 38
 - Installation für SSCM 291
- Deklarations-Zustand 25
- device (Datentyp) 9
- DSSM 26
 - Abnormale Beendigung 71
 - Adressraum-Haushalt 57
 - Arbeitstask 58
 - Aufruf durch die Startup-Routine 211
 - Begriffsbestimmung 35
 - Fehlerbehandlung 71
 - Funktionsübersicht 53
 - Holdertask 58
 - Inbetriebnahme 63

- DSSM (Fortsetzung)
- Kommandoübersicht [36, 74](#)
 - Produkte entbündelt [33](#)
 - Produkte im Grundausbau [31](#)
 - Protokollierung [120](#)
 - Share-Programme [59](#)
 - Speicherkonzept [56](#)
 - Subsystemdefinition [23](#)
 - Subsystemkatalog siehe
 - Subsystemkatalog [25](#)
 - Subsystemkomponente [24](#)
 - Subsystem-Konfiguration [38](#)
 - Taskkonzept [58](#)
 - Versionsabhängigkeiten [28](#)
- DSSM (Parametersatz zur Inbetriebnahme) [64](#)
- DSSMLOG-Protokolldatei [120](#)
- DSSM-Task-Error [71](#)
- dynamische Änderungen von Subsystem-Attributen [62](#)
- E**
- Eigenschaften von Subsystemen dynamisch ändern [62](#)
- Einschränkungen
 - bei abhängigen Subsystemen [231](#)
 - bei Adressbeziehungen [229](#)
- Einsprungstelle [132](#)
- Einsprungstelle [siehe Auftragseingänge eines Subsystems](#)
- Entfernen eines inaktiven Subsystems aus dem Katalog [111](#)
- Entladen
 - eines lokalen Subsystemkatalogs [166](#)
 - eines Subsystems [162](#)
- ENTRY [130](#)
- Entsperren eines Subsystems [168](#)
- Error-Logging-Datei SERSLOG [73](#)
- Erweitern des Subsystemkatalogs [60, 75](#)
- ESMC (DSSM-Abrechnungssatz) [69](#)
- ESMD (DSSM-Abrechnungssatz) [70](#)
- Exchange-Modus [220, 263](#)
- Externverweise [133, 230](#)
 - angeben und bearbeiten [228](#)
 - auf Subsysteme [271](#)
 - löschen [230](#)
- F**
- Fehlerbehandlung in DSSM [71](#)
- Fehlerdatei SERSLOG [73](#)
- filename (Datentyp) [10](#)
- fixed (Datentyp) [9](#)
- FORCE
 - Option zum Anhalten von Subsystemen [80](#)
 - Subsystem zwangsdeaktivieren [162](#)
- full-filename
 - siehe Datentyp file-name [10](#)
- Funktionsübersicht von DSSM/SSCM [53](#)
- G**
- gen (Zusatz zu Datentypen) [20](#)
- GENERAL-ATTRIBUTES (Ausgabefeld) [139, 280](#)
- GENERATE-CATALOG-SOURCE (SSCM-Anweisung) [182, 198](#)
- Generierung eines Subsystemkatalogs [179, 286](#)
- H**
- Holdertask [38, 49, 58, 129, 193](#)
 - beim Aktivieren des Subsystems [41](#)
 - Nutzung aufheben [233](#)
 - Subsysteme verteilen [193](#)
- Holdertask-Error [72](#)
- HOLD-SUBSYSTEM (DSSM-Kommando) [74, 80](#)
- I**
- Identifikation eines Subsystems [53](#)
- IMON [291](#)
- Inbetriebnahme von DSSM [63, 64](#)
- IN-CREATE (Subsystem-Zustand) [49](#)
- IN-DELETE (Subsystem-Zustand) [50](#)
- Index
 - global [18](#)
 - Konstruktionszeichenfolge [18](#)
 - platzhalter-spezifisch [18](#)

- Index (Fortsetzung)
 - Schreibweise 19
 - Information
 - über Subsysteme 49, 142
 - über Subsystem-Konfiguration 138
 - Informationsdatei 48
 - IN-HOLD (Subsystem-Zustand) 50
 - Initialisieren
 - von Subsystemen 153
 - von Subsystemen (INIT-Routine) 214, 257
 - IN-RESUME (Subsystem-Zustand) 50
 - Installation von SSCM 291
 - integer (Datentyp) 11
 - INTERNAL-ENTRIES (Ausgabefeld) 139, 280
 - ISAM-Datei, SSD-Object 26
 - ISL-Einsprungsstelle 48, 188, 221
- K**
- Katalog [siehe Subsystemkatalog](#)
 - Klasse-x-Speicher (Laden von Subsystemen) 56
 - Koexistenz
 - von SSCM-Versionen 30, 291
 - von Subsystemversionen (Coexistence-Modus) 47, 219, 262
 - Kommandos
 - Syntaxdarstellung 5
 - von DSSM 74
 - Kommunikation zwischen Subsystem und DSSM 49
 - Komponenten eines Subsystems 24
 - Konfiguration
 - eines Subsystems 38
 - von Subsystemen ermitteln 138
 - Konstruktionsangabe 19
 - Konstruktionszeichenfolge 18
 - Kurzname 8
- L**
- Ladeadresse von Subsystemen ermitteln 138
 - Lademodule für DSSM 63
 - Lademodus 133
 - eines Subsystems 207, 249
 - Laden eines lokalen Subsystemkatalogs 83
 - Lade-Zustand 25
 - Liefergruppe 88
 - Liefergruppen-Name 205, 247
 - Linkage (indirektes) 188, 220
 - Literaturhinweise 4
 - LOAD-LOCAL-SUBSYSTEM-CATALOG (DSSM-Kommando) 74, 83
 - LOCKED (Subsystem-Zustand) 50, 71, 168
 - lokaler Subsystemkatalog
 - entladen 74, 166
 - laden 74, 83
 - lokales Subsystem
 - aktivieren 74, 150
 - Beispiel zur Nutzung 170
 - deaktivieren 74, 159
 - low (Zusatz zu Datentypen) 15
- M**
- man (Zusatz zu Datentypen) 20, 21
 - mandatory (Zusatz zu Datentypen) 21
 - Meldungsdatei 48, 63, 130, 209, 251
 - Installation für SSCM 291
 - SYSTEMS.SSCM.023 291
 - Memory Pool 56
 - Memory Pool-Kollision 62
 - MEMORY-CLASS (Ausgabefeld) 139, 280
 - Metasyntax SDF 5, 7
 - MODIFY-SUBSYSTEM-ATTRIBUTES (SSCM-Anweisung) 182, 200
 - MODIFY-SUBSYSTEM-PARAMETER (DSSM-Kommando) 74, 85
 - MODIFY-WORK-TASK-ATTRIBUTE (SSCM-Anweisung) 182, 233
 - Modulbibliothek 48
 - Installation für SSCM 291
 - SYSLNK.SSCM.023 291
 - Monitor-Jobvariable (SSCM) 181
 - Monitor-Jobvariablen 52
- N**
- name (Datentyp) 11
 - Nebenkomponenten eines Subsystems 48, 53
 - Nichtverfügbarkeit von Subsystemen bei Versionsaustausch 45
 - NOREF-Datei 63

NOT-CREATED (Subsystem-Zustand) [49](#)
 NOT-RESUMED (Subsystem-Zustand) [50](#)

O

Objectcode-Bibliothek [206](#)
 Objektmoduldatei [133](#)
 odd (Zusatz zu Datentypen) [20](#)

P

PAM-Datei mit Subsystemkatalog [238](#)
 Parameter von Subsystemen anzeigen lassen [142](#)
 Parameterservice zur Inbetriebnahme von DSSM [63](#)
 partial-filename (Datentyp) [12](#)
 path-compl (Zusatz zu Datentypen) [15](#)
 Performance-Verbesserung [41](#)
 Performance-Verschlechterung [56](#), [120](#)
 Phasen der Systemeinleitung [211](#)
 posix-filename (Datentyp) [12](#)
 posix-pathname (Datentyp) [12](#)
 POSIX-Platzhalter [16](#)
 product-version (Datentyp) [13](#)
 Produkte (DSSM-fähig)
 entbündelt [33](#)
 im BS2000-Grundausbau [31](#)
 Produktversion [130](#)
 Programmbereich [57](#)
 eines Subsystems [41](#)
 Programmüberwachung durch Monitor-Jobvariable (SSCM) [181](#)
 Protokolldatei
 CONSLOG [63](#)
 DSSMLOG [120](#)
 Protokollierung
 der Parameterverarbeitung [63](#)
 von DSSM ein-/ausschalten [120](#)
 Protokollierungsfunktion ein-/ausschalten [74](#)
 Prüffunktion für Subsystemkatalog [196](#)
 PULS-Problemmeldungen [289](#)

Q

quotes (Zusatz zu Datentypen) [21](#)

R

Readme-Datei [3](#)
 reentrant-fähige Subsysteme [57](#)
 Referenz-Subsystem [128](#)
 RELATED-FILES (Ausgabefeld) [140](#), [280](#)
 Release Unit [88](#)
 RELEASE-SUBSYSTEM-SPACE (DSSM-Kommando) [74](#), [110](#)
 REMOVE-ADDR-SPACE-SEPARATION (SSCM-Anweisung) [182](#), [235](#)
 REMOVE-CATALOG-ENTRY (SSCM-Anweisung) [182](#), [237](#)
 REMOVE-SUBSYSTEM (DSSM-Kommando) [74](#), [111](#)
 Rep-Datei [48](#), [131](#), [250](#)
 Installation für SSCM [291](#)
 SYSREP.SSCM.023 [291](#)
 REP-FILE (Ausgabefeld) [139](#), [280](#)
 Rep-Korrekturdatei [63](#)
 RESET-Option [131](#)
 RESUME-SUBSYSTEM (DSSM-Kommando) [74](#), [113](#)

S

SAVE-CATALOG (SSCM-Anweisung) [182](#), [238](#)
 SAVE-SSD (SSCM-Anweisung) [182](#), [240](#)
 SAVE-SUBSYSTEM-CATALOG (DSSM-Kommando) [74](#), [116](#)
 SDF-Syntaxdarstellung [5](#)
 SDF-Syntaxdatei [63](#)
 sep (Zusatz zu Datentypen) [20](#)
 SEPARATE-ADDRESS-SPACE (SSCM-Anweisung) [61](#), [182](#), [241](#)
 SERSLOG-Fehlerdatei [73](#)
 SET-DSSM-OPTIONS (DSSM-Kommando) [74](#), [120](#)
 SET-SUBSYSTEM-ATTRIBUTES (SSCM-Anweisung) [53](#), [61](#), [182](#), [243](#)
 Share-Programme verwalten [59](#)
 SHOW-CATALOG (SSCM-Anweisung) [182](#), [274](#)
 SHOW-SSD (SSCM-Anweisung) [182](#), [281](#)
 SHOW-SUBSYSTEM-ATTRIBUTES (DSSM-Kommando) [74](#), [122](#)

- SHOW-SUBSYSTEM-INFO (DSSM-Kommando) [74](#), [138](#)
- SHOW-SUBSYSTEM-STATUS (DSSM-Kommando) [74](#), [142](#)
- Sichern von Änderungen im Subsystemkatalog [116](#)
- Speicherklasse [59](#), [130](#)
- Speicherkonzept von DSSM [56](#)
- SSCM [26](#), [179](#)
 - Anweisungen [182](#)
 - Dateien zur Installation [291](#)
 - Funktionsübersicht [53](#)
 - Installation [291](#)
 - Katalog überprüfen [196](#)
 - Koexistenz von SSCM-Versionen [30](#), [291](#)
 - mit Prozedur starten [291](#)
 - Programmüberwachung durch Monitor-Jobvariable [181](#)
 - SSMCAT [25](#), [39](#), [179](#)
 - Starten und Beenden [181](#)
 - Subsystemkatalog [siehe Subsystemkatalog](#)
 - Versionsabhängigkeiten [28](#)
- SSD-Object [26](#), [183](#), [186](#), [243](#)
- SSINFO (Informationsdatei) [209](#), [252](#)
- SSMCAT (statischer Subsystemkatalog) [25](#)
- SSMCAT (Subsystemkatalog) [26](#), [39](#), [179](#)
- Stand-alone-Prinzip [77](#)
- START-CATALOG-CREATION (SSCM-Anweisung) [182](#), [286](#)
- START-CATALOG-MODIFICATION (SSCM-Anweisung) [182](#), [287](#)
- Starten
 - eines Subsystems [153](#)
 - Startzeitpunkt für Subsysteme [40](#), [96](#)
 - von SSCM [181](#)
- START-LOCAL-SUBSYSTEM (DSSM-Kommando) [74](#), [150](#)
- START-SSD-CREATION (SSCM-Anweisung) [182](#), [288](#)
- START-SUBSYSTEM (DSSM-Kommando) [74](#), [153](#)
- Startup [71](#)
- Startup-gebundene Subsysteme [40](#)
- STATE-CHANGE-CMDS (Ausgabefeld) [139](#), [280](#)
- Static Subsystem Catalog Manager (SSCM) [179](#)
- STOP-LOCAL-SUBSYSTEM (DSSM-Kommando) [74](#), [159](#)
- STOP-SUBSYSTEM (DSSM-Kommando) [74](#), [162](#)
- structured-name (Datentyp) [13](#)
- Subsystem [23](#)
 - (inaktiv) aus dem Katalog entfernen [111](#)
 - Abhängigkeitsbeziehungen [131](#)
 - Abhängigkeitsbeziehungen definieren und verändern [230](#)
 - Abhängigkeitsbeziehungen löschen [232](#)
 - Adressbeziehung [131](#)
 - Adressraum freigeben [74](#), [110](#)
 - Adressraum vereinbaren [268](#)
 - Adressraumüberschneidungen vermeiden [241](#)
 - Adressraumverteilung [235](#), [241](#)
 - aktivieren [39](#), [74](#), [153](#), [254](#)
 - anhalten [44](#), [80](#)
 - Anschluss aufheben [43](#)
 - Anschluss herstellen [42](#)
 - Anzahl der Auftragseingänge für ein SSD-Object erhöhen [186](#)
 - Attribute (Eigenschaften) [24](#)
 - Attribute anzeigen [36](#), [74](#)
 - Attribute dynamisch ändern [62](#)
 - Auftragsbelegung ermitteln [142](#)
 - Auftragseingänge [220](#)
 - Auftragseingänge bestimmen [24](#)
 - Auftragseingänge löschen [225](#)
 - aus Katalog löschen [237](#)
 - Austausch von Subsystemversionen [45](#)
 - automatisch laden [212](#), [255](#)
 - beenden [162](#)
 - Benutzeradressraum [56](#)
 - Beziehungen zu anderen Subsystemen [24](#)
 - Beziehungen zwischen Subsystemen [47](#)
 - Datenbereich [41](#), [57](#)
 - deaktivieren [44](#), [74](#), [162](#)
 - definieren [243](#)
 - Definition (im SSD-Object) [26](#)
 - Definition abspeichern [240](#)
 - Definition ändern [200](#)

Subsystem (Fortsetzung)

Definition aufnehmen 183
 Definition löschen 237
 Deklaration 38, 53
 DSSM-Protokollierung aus- oder einschalten 120
 dynamische Identitätprüfung 129
 Eigenschaften (Attribute) 24
 Eigenschaften dynamisch ändern 62
 entladen 162
 entsperren 74, 168
 Externverweise 228
 Externverweise definieren und verändern 228
 Externverweise löschen 230
 gesperrt 168
 globale Verwaltung 54
 Identifizierung gegenüber DSSM 23
 im Memory Pool 58, 226
 in Katalog aufnehmen 183
 in Wartezustand versetzen 74
 in Wartezustand versetzen 80
 inaktiv 74
 Information 54
 Information über Konfiguration abfragen 138
 Informationen über Attribute anzeigen siehe SHOW-SUBSYSTEM-ATTRIBUTES 122
 Informationen über den Zustand abfragen 49
 Informationsdatei 132
 Inhalt einer Objektmoduldatei ausgeben 281
 Interner Status 133
 Katalog [siehe Subsystemkatalog](#)
 Koexistenz von Versionen 47
 Kommunikation mit DSSM 49
 Komponenten 24, 153
 Konfiguration 24, 49, 54
 Konfiguration anzeigen 74
 Ladeadresse ermitteln 138
 Lademodus 207, 249
 lokales 27
 lokales [siehe lokales Subsystem](#)
 Name 129
 NebenkompONENTEN 48
 nicht-privilegiertes (TU-Subsystem) 26

Subsystem (Fortsetzung)

Nichtverfügbarkeit bei Versionsaustausch 45
 Parameter verändern 74, 85
 privilegiertes (TPR-Subsystem) 26
 Programmbereich 41, 57
 reentrant-fähig 57
 Schnittstellenversion 130
 Speicherkonzept 56
 starten 153
 Startzeitpunkt 129
 Startzeitpunkt bestimmen 39, 96, 254
 Status 133
 Systemadressraum 56
 Unverträglichkeiten bei Beziehungen 79
 vereinbarte Adressraumüberschneidungen aufheben 235
 Version 129
 Version ermitteln 142
 Versionen austauschen 45
 Verteilung auf Holdertasks steuern 193
 Wartezustand aufheben 74, 113
 Wiederanlauf 42
 Worktask-Nutzung aufheben 233
 Zustand abfragen 49, 74
 Zustandswechsel 51
 zwangsdeaktivieren 162
 SUBSYSTEM-ACCESS (Ausgabefeld) 139, 280
 Subsystem-Beendigungs-Abrechnungssatz (ESMD) 70
 Subsystem-Initialisierungs-Abrechnungssatz (ESMC) 69
 Subsystemkatalog 25, 38
 abspeichern 238
 ändern 85
 Änderungen sichern 74, 116
 dynamisch erweitern 74
 erweitern 60, 75
 generieren 179, 286
 Inhalt ausgeben 274
 lokaler 27
 lokaler [siehe lokaler Subsystemkatalog](#)
 mehr als 100 Auftragseingänge definieren 186
 modifizieren 287

Subsystemkatalog (Fortsetzung)

- SSMCAT 39
 - Standarddateiname 65, 117, 286
 - statischen Katalog erweitern 183
 - Subsystem löschen 237
 - Subsystem neu aufnehmen 183
 - überprüfen 196
 - Unverträglichkeiten 79
 - verändern 287
 - verwalten 60
- ## Subsystem-Konfiguration, lokale 27
- ## Subsystemroutine
- CLOSE-CTRL-ROUTINE 128
 - DEINIT 129
 - INIT 129
 - STOPCOM 132
- ## SVC-Einsprungstelle 48, 189, 222
- ## synchrone Aktivitäten 38
- ## Syntaxdarstellung SDF 5
- ## Syntaxdatei 48, 133
- beim Laden zu aktivierende 210, 253
 - Installation für SSCM 291
 - SYSSDF.SSCM.023 291
 - SYSSDF.SSCM.023.USER 291
- ## SYS...-Dateien zur Installation von SSCM 291
- ## SYSLST (Standard-Ausgabedatei) 279, 285
- ## SYSPRC.SSCM.023 (Prozedur zum Start von SSCM) 291
- ## SYSSSC.SSCM.023 (Subsystem-Deklarationsdatei) 291
- ## System Procedure Linkage 188
- ## Systemadressraum 56
- ## Systemeinleitung 63, 211
- ## System-Exit-Routinen 48, 190, 223

T

Task

- Abbruch bei Anhalten von Subsystemen 80
 - Belegung von Subsystemen ermitteln 142
 - DSSM-Taskkonzept 58
- ## taskeigener Benutzeradressraum 83
- ## Task-Error 71
- ## Taskkonzept 58
- ## temp-file (Zusatz zu Datentypen) 20

- text (Datentyp) 13
- TID von belegenden Tasks 142
- time (Datentyp) 13
- TPR-Subsystem 26
- TSN
 - der Arbeitstask des Subsystems 194
 - der gemeinsamen Holdertask 195
 - von belegenden Tasks 142
- TU-Subsystem 26

U

Übersicht

- über DSSM-Kommandos 36, 74
 - über SSCM-Anweisungen 182
- ## under (Zusatz zu Datentypen) 15
- ## UNLOAD-LOCAL-SUBSYSTEM-CATALOG (DSSM-Kommando) 74, 166
- ## UNLOCK-SUBSYSTEM (DSSM-Kommando) 74, 168
- ## Unverträglichkeiten bei Beziehungen zwischen Subsystemen 79
- ## user (Zusatz zu Datentypen) 21

V

- Verändern von Subsystem-Parametern 85
- Verbindungsaufbau zu Subsystemen nach Aufhebung des Wartezustands 113
- vers (Zusatz zu Datentypen) 21
- Versionsabhängigkeiten zwischen BS2000, DSSM und SSCM 28
- Versionsaustausch bei Subsystemen 45
 - ohne Unterbrechung der Verfügbarkeit 45
- Versionskoexistenz 47
- Verwalten
 - des dynamischen Subsystemkatalogs 60
 - von Share-Programmen 59
- vsn (Datentyp) 13

W

Wartezustand

- für ein Subsystem aufheben 113
 - für ein Subsystem vereinbaren 80
- ## Wiederanlauf von Subsystemen 39, 42
- ## wild(n) (Zusatz zu Datentypen) 16

with (Zusatz zu Datentypen) 15
without (Zusatz zu Datentypen) 20

X

x-string (Datentyp) 14
x-text (Datentyp) 14

Z

Zeitpunkt der Aktivierung von Subsystemen 40
Zusätze zu Datentypen 6, 15
Zustände eines Subsystems 49
Zustandswechsel 51

Inhalt

1	Einführung	1
1.1	Zielgruppe und Konzept des Handbuchs	1
1.2	Änderungen gegenüber der Ausgabe Dezember 1996 (DSSM V3.6 und SSCM V2.1) .	2
1.3	Readme-Datei - Änderungen zu den aktuellen Versionen	3
1.4	Darstellungsmittel	4
1.5	SDF-Syntaxdarstellung	5
2	Das Subsystem-Konzept in BS2000/OSD	23
2.1	Erläuterung wichtiger Begriffe	23
2.2	Verwaltung von Subsystemen mit DSSM und SSCM	26
2.3	Lokale Subsystemverwaltung	27
2.4	Versionsabhängigkeiten zwischen BS2000/OSD, DSSM und SSCM	28
2.5	Übersicht über wichtige DSSM-fähige Produkte im BS2000-Grundausbau (GA)	31
2.6	Übersicht über ausgewählte entbündelte, DSSM-fähige Produkte	33
3	DSSM	35
3.1	Aufgaben und Funktionen von DSSM	38
3.1.1	Subsystem-Deklaration (SSC)	38
3.1.2	Aktivieren bzw. Wiederanlauf	39
3.1.3	Anschluss herstellen und wieder aufheben	42
3.1.4	Subsystem deaktivieren oder anhalten	44
3.1.5	Austausch von Subsystemversionen	45
3.1.6	Koexistenz von Versionen	47
3.1.7	Beziehungen zwischen Subsystemen	47
3.1.8	Beziehungen zwischen Nebenkomponten und Subsystemen	48
3.1.9	Kommunikation zwischen Subsystem und DSSM	49
3.1.10	Informieren über Subsysteme	49
3.1.11	Zustände eines Subsystems	49
3.1.12	Subsystem-Überwachung mittels Monitor-Jobvariablen	52
3.1.13	Funktionsübersicht	53
3.2	Speicher- und Taskkonzept	56
3.3	Verwaltung von Share-Programmen	59
3.4	Verwaltung des dynamischen Subsystemkatalogs	60
3.5	Inbetriebnahme der dynamischen Subsystemverwaltung	63

3.6	Abrechnungssätze des DSSM (Accounting)	66
	ESMC - Subsystem-Initialisierungs-Abrechnungssatz	69
	ESMD - Subsystem-Beendigungs-Abrechnungssatz	70
3.7	Fehlerbehandlung in DSSM	71
3.8	Die Kommandos von DSSM	74
	ADD-SUBSYSTEM	
	Dynamischen Subsystemkatalog erweitern	75
	HOLD-SUBSYSTEM	
	Subsystem in den Wartezustand versetzen	80
	LOAD-LOCAL-SUBSYSTEM-CATALOG	
	Lokalen Subsystemkatalog laden	83
	MODIFY-SUBSYSTEM-PARAMETER	
	Subsystem-Parameter verändern	85
	RELEASE-SUBSYSTEM-SPACE	
	Reservierten Adressraum von Subsystemen freigeben	110
	REMOVE-SUBSYSTEM	
	Inaktives Subsystem aus dynamischem Katalog entfernen	111
	RESUME-SUBSYSTEM	
	Wartezustand für ein Subsystem aufheben	113
	SAVE-SUBSYSTEM-CATALOG	
	Änderungen des dynamischen Subsystemkatalogs sichern	116
	SET-DSSM-OPTIONS	
	Protokollierungsfunktion von DSSM aus- oder einschalten	120
	SHOW-SUBSYSTEM-ATTRIBUTES	
	Informationen über Subsystem-Attribute anzeigen	122
	SHOW-SUBSYSTEM-INFO	
	Informationen über aktuelle Subsystem-Konfiguration anfordern	138
	SHOW-SUBSYSTEM-STATUS	
	Informationen über den Zustand von Subsystemen anfordern	142
	START-LOCAL-SUBSYSTEM	
	Lokales Subsystem im Benutzeradressraum aktivieren	150
	START-SUBSYSTEM	
	Subsystem aktivieren	153
	STOP-LOCAL-SUBSYSTEM	
	Lokales Subsystem im Benutzeradressraum deaktivieren	159
	STOP-SUBSYSTEM	
	Subsystem deaktivieren	162
	UNLOAD-LOCAL-SUBSYSTEM-CATALOG	
	Lokalen Subsystemkatalog entladen	166
	UNLOCK-SUBSYSTEM	
	Subsystem im Zustand LOCKED in den Zustand NOT-CREATED überführen (Subsystem entsperren)	168

3.9	Beispiele	170
3.9.1	Nutzung von lokalen Subsystemen	170
3.9.2	Ausgabe in eine S-Variable	173
4	SSCM	179
4.1	Generierung eines Subsystemkatalogs	179
4.2	Starten und Beenden von SSCM	181
4.3	Die Anweisungen von SSCM	182
	ADD-CATALOG-ENTRY	
	Subsystemdefinition(en) in den Subsystemkatalog aufnehmen	183
	ADD-SUBSYSTEM-ENTRIES	
	Zusätzliche Auftragseingänge definieren	186
	ASSIGN-HOLDER-TASK	
	Subsysteme auf Holdertasks verteilen	193
	CHECK-CATALOG	
	Subsystemdefinition(en) auf Konsistenz prüfen	196
	GENERATE-CATALOG-SOURCE	
	SSCM-Anweisungsliste für Generierung erstellen	198
	MODIFY-SUBSYSTEM-ATTRIBUTES	
	Eigenschaften von Subsystemen verändern	200
	MODIFY-WORK-TASK-ATTRIBUTE	
	Arbeitstask-Parameter verändern	233
	REMOVE-ADDR-SPACE-SEPARATION	
	Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher aufheben	235
	REMOVE-CATALOG-ENTRY	
	Subsystemdefinition logisch aus dem Subsystemkatalog löschen	237
	SAVE-CATALOG	
	Subsystemkatalog als PAM-Datei abspeichern	238
	SAVE-SSD	
	Subsystemdefinition(en) abschließen	240
	SEPARATE-ADDRESS-SPACE	
	Disjunkte Verteilung der Subsysteme im Klasse-5-Speicher steuern	241
	SET-SUBSYSTEM-ATTRIBUTES	
	Eigenschaften und Einsprungstellen eines Subsystems definieren	243
	SHOW-CATALOG	
	Subsystem-Konfiguration anzeigen	274
	SHOW-SSD	
	Inhalt eines SSD-Objects (Subsystemdefinitionen) anzeigen	281
	START-CATALOG-CREATION	
	Name eines statischen Subsystemkatalogs vereinbaren	286
	START-CATALOG-MODIFICATION	
	Statischen Subsystemkatalog verändern	287
	START-SSD-CREATION	
	SSD-Object zur Aufnahme von Subsystemdefinitionen generieren	288

4.4	Installation von SSCM	291
4.5	Beispiele	292
4.5.1	Erstellung eines SSD-Objects	292
4.5.2	Erstellung eines statischen Subsystemkatalogs	293
4.5.3	Änderung eines statischen Subsystemkatalogs	294
	Literatur	295
	Stichwörter	309

DSSM V4.0/SSCM V2.3

Verwaltung von Subsystemen in BS2000/OSD

Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich an die Systembetreuung und die Softwareberatung des BS2000/OSD.

Inhalt

Es werden das Subsystemkonzept des BS2000/OSD, die Dynamische Subsystemverwaltung DSSM und die Subsystemkatalog-Verwaltung SSCM mit den dazugehörigen Kommandos und Anweisungen beschrieben.

Ausgabe: März 2002

Datei: dssm.pdf

Copyright © Fujitsu Siemens Computers GmbH, 2002.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

Dieses Handbuch wurde erstellt von
cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Fujitsu Siemens Computers GmbH
Handbuchredaktion
81730 München

Kritik Anregungen Korrekturen

Fax: 0 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
<http://manuals.fujitsu-siemens.com>

Absender

Kommentar zu DSSM V4.0/SSCM V2.3
Verwaltung von Subsystemen in BS2000/OSD



Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/) and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009