

Deutsch



# LMS V3.4B

SDF-Format

Ausgabe Juni 2013

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

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

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

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

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

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

## **Copyright und Handelsmarken**

Copyright © Fujitsu Technology Solutions GmbH 2013.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

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

---

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>11</b>
1.1	Kurzbeschreibung des Produkts	11
1.2	Zielsetzung und Zielgruppen des Handbuchs	12
1.3	Konzept des Handbuchs	13
1.4	Verwendete Darstellungsmittel	14
1.5	Readme-Datei	15
1.6	Änderungen gegenüber der Vorgänger-Version LMS V3.3A	16
1.7	Voraussetzungen zur Software-Konfiguration	18
<b>2</b>	<b>Merkmale von LMS</b>	<b>19</b>
2.1	Programmbibliotheken	19
2.2	Elemente	19
2.3	Delta-Technik	20
2.4	Eingabe- und Ausgabestrom	20
2.5	Ein- und Ausgabe in S-Variable (SDF-P)	24
2.6	Unterstützung der Programmentwicklung	27
2.7	LMS im Dialog- und Stapelbetrieb	28
2.8	Adressierungsmodus	29

<b>3</b>	<b>Programmbibliotheken</b>	<b>31</b>
<b>3.1</b>	<b>Aufbau einer Bibliothek</b>	<b>32</b>
<b>3.2</b>	<b>Ein- und Ausgabebibliotheken</b>	<b>33</b>
<b>3.3</b>	<b>Mehrfachzugriff auf Bibliotheken</b>	<b>33</b>
<b>3.4</b>	<b>Einschränken des Mehrfachzugriffes</b>	<b>34</b>
<b>4</b>	<b>Elemente</b>	<b>35</b>
<b>4.1</b>	<b>Mehrfachzugriff auf Elemente</b>	<b>36</b>
<b>4.2</b>	<b>Beschreibung der Elementtypen</b>	<b>37</b>
4.2.1	Standardtypen	37
4.2.2	Abgeleitete Typen	40
<b>4.3</b>	<b>Konvention für Elementbezeichnungen</b>	<b>41</b>
4.3.1	Elementbezeichnung in den Anweisungen	41
4.3.2	Protokollierung der Elementbezeichnung	42
4.3.3	Auswahlangabe für Elementbezeichnungen	43
4.3.4	Konstruktionsangabe für Elementbezeichnungen	45
<b>4.4</b>	<b>Elementattribute</b>	<b>47</b>
4.4.1	Attribute, die von der Zugriffsmethode vergeben werden	47
4.4.2	Attribute, die vom Benutzer vergeben werden können	48
4.4.3	Eingabeformat des Datums	48
<b>4.5</b>	<b>Beziehungen zwischen Elementen</b>	<b>49</b>
<b>4.6</b>	<b>Versionsverwaltung</b>	<b>52</b>
4.6.1	Versionshaltung und -speicherung	52
4.6.2	Versionsbezeichnungen	54
4.6.3	Versionskonventionen	57
4.6.3.1	Konvention: NONE	58
4.6.3.2	Konvention: STD-SEQUENCE	58
4.6.3.3	Konvention: STD-TREE	59
4.6.3.4	Konvention: MULTI-SEQUENCE	60
<b>4.7</b>	<b>Elementschutz/Datenschutz</b>	<b>61</b>
4.7.1	Zugriffsschutz für Elemente	61
4.7.2	Datenschutz durch Überschreiben	69
4.7.3	Beweissicherung	70

---

<b>5</b>	<b>Funktionen von LMS</b>	<b>71</b>
<b>5.1</b>	<b>Starten/Beenden von LMS</b>	<b>71</b>
5.1.1	Starten von LMS	71
5.1.2	Überwachung des LMS-Laufs mit Jobvariablen	73
5.1.3	Startdatei	75
5.1.4	Voreinstellungen nach dem Starten von LMS	77
5.1.5	Beenden des LMS-Laufs	78
<b>5.2</b>	<b>Zuweisen von Bibliotheken</b>	<b>79</b>
<b>5.3</b>	<b>Bearbeiten von Elementen</b>	<b>88</b>
5.3.1	Aufnehmen von Elementen in eine Bibliothek	88
5.3.2	Ausgeben von Elementen in eine Datei	91
5.3.3	Auflisten von Elementen	91
5.3.4	Löschen von Elementen	91
5.3.5	Vergleichen von Elementen	92
5.3.6	Korrigieren von Elementen	94
5.3.7	Umbenennen von Elementen	95
5.3.8	Inhaltsverzeichnis einer Bibliothek ausgeben	95
5.3.9	Speichern von Prozeduren	95
<b>5.4</b>	<b>Archivieren mit Delta-Technik</b>	<b>96</b>
5.4.1	Delta als Speicherungsform und Ordnungsmittel	97
5.4.2	Aufnehmen von Delta-Elementen	98
5.4.3	Übersicht über Delta-Elemente	99
5.4.4	Löschen von Delta-Elementen	100
5.4.5	Sperren von Delta-Elementen	100
5.4.6	Besonderheiten der Delta-Technik	100
<b>5.5</b>	<b>Steuern des LMS-Laufs</b>	<b>101</b>
5.5.1	LMS-Protokollparameter	101
5.5.2	Protokollausgabe steuern	102
5.5.3	Bildschirmwechsel steuern	104
5.5.4	Fehlerbehandlung im Dialog- und Prozedurbetrieb	104
5.5.4.1	Spin-off-Mechanismus	104
5.5.4.2	Anweisungs-Returncode-Mechanismus	104
5.5.5	Benutzeranschlüsse	108
5.5.6	Unterbrechen des LMS-Laufs	108
5.5.7	Verwenden von Auftragsschaltern	109
<b>5.6</b>	<b>Pamkey-Eliminierung</b>	<b>110</b>
5.6.1	Bibliotheksddateien	110
5.6.2	Elementverarbeitung	110
5.6.3	Zusammenfassung	115

<b>5.7</b>	<b>Unterstützung von NK4-Platten</b> . . . . .	<b>116</b>
5.7.1	Aufnehmen von Dateien mit ADD-ELEMENT . . . . .	116
5.7.2	Ausgeben von Elementen mit EXTRACT-ELEMENT . . . . .	116
<b>5.8</b>	<b>Behandlung von Aliasnamen (ACS)</b> . . . . .	<b>119</b>
<b>5.9</b>	<b>Verwendung erweiterter Zeichensätze in LMS (XHCS)</b> . . . . .	<b>121</b>
5.9.1	Hard- und Softwarevoraussetzungen . . . . .	122
5.9.2	LMS-spezifische Anwendung erweiterter Zeichensätze . . . . .	123
<b>5.10</b>	<b>Nutzung der LMS-Funktionalität aus dem EDT</b> . . . . .	<b>126</b>
<b>5.11</b>	<b>LMS und EDT V17</b> . . . . .	<b>129</b>
<b>6</b>	<b>Unterstützung des Software-Entwicklungsprozesses</b> . . . . .	<b>131</b>
<b>6.1</b>	<b>Ausleihverfahren</b> . . . . .	<b>132</b>
<b>6.2</b>	<b>make-Funktionalität</b> . . . . .	<b>133</b>
6.2.1	Aktionen . . . . .	137
6.2.2	Verwendung von Variablen . . . . .	137
6.2.3	Auswahl- und Konstruktionsangabe bei make . . . . .	139
6.2.4	Ablaufsteuerung während der Folgeverarbeitung . . . . .	139
6.2.5	TOUCH . . . . .	140
6.2.6	Arbeitsweise von make . . . . .	140
<b>7</b>	<b>Anweisungen</b> . . . . .	<b>145</b>
<b>7.1</b>	<b>SDF-Standardanweisungen für LMS</b> . . . . .	<b>145</b>
<b>7.2</b>	<b>Syntaxbeschreibung</b> . . . . .	<b>146</b>
<b>7.3</b>	<b>Eingaberegeln</b> . . . . .	<b>162</b>
<b>7.4</b>	<b>Anweisungskurznamen</b> . . . . .	<b>163</b>
<b>7.5</b>	<b>Beschreibung der LMS-Anweisungen</b> . . . . .	<b>164</b>
	ACTIVATE-USER-EXIT	
	Benutzerausgang aktivieren . . . . .	166
	ADD-ELEMENT	
	Datei als Element in eine Bibliothek aufnehmen . . . . .	173
	BEGIN-MAKE	
	Einleiten der make-Verarbeitung . . . . .	184
	make-Subanweisungen . . . . .	190
	CALL-EDT	
	Aufrufen des EDT . . . . .	204

CLOSE-LIBRARY	
Schließen einer Bibliothek . . . . .	206
COMPARE-ELEMENT	
Vergleichen von Elementen . . . . .	208
COPY-ELEMENT	
Elemente kopieren . . . . .	220
COPY-LIBRARY	
Bibliothek kopieren . . . . .	233
DEACTIVATE-USER-EXIT	
Benutzerausgang lösen . . . . .	236
DELETE-ELEMENT	
Löschen von Elementen . . . . .	238
EDIT-ELEMENT	
Editieren von Elementen . . . . .	245
EDIT-ELEMENT-ATTRIBUTES	
Geführten Dialog für MODIFY-ELEMENT-ATTRIBUTES aktivieren . . . . .	259
EDIT-ELEMENT-PROTECTION	
Geführten Dialog für MODIFY-ELEMENT-PROTECTION aktivieren . . . . .	262
END	
Beenden des LMS-Laufes . . . . .	264
EXTRACT-ELEMENT	
Ausgeben von Elementen als Dateien . . . . .	265
FIND-ELEMENT	
Elemente mit Muster in Sätzen finden . . . . .	276
MODIFY-ELEMENT	
Ändern von Bindemodulen, Phasen und Bindeladmodulen . . . . .	285
MODIFY-ELEMENT-Subanweisungen für Elementtypen R, C und L . . . . .	295
MODIFY-ELEMENT-Subanweisungen für Textelemente . . . . .	308
MODIFY-ELEMENT-ATTRIBUTES	
Elementattribute ändern . . . . .	313
MODIFY-ELEMENT-PROTECTION	
Elementschutz ändern . . . . .	323
MODIFY-LIBRARY-ATTRIBUTES	
Attribute einer Bibliothek ändern . . . . .	333
MODIFY-LMS-DEFAULTS	
Ändern der globalen Default-Werte . . . . .	339
MODIFY-LOGGING-PARAMETERS	
LMS-Protokollparameter ändern . . . . .	360
MODIFY-TYPE-ATTRIBUTES	
Typattribute ändern . . . . .	365
OPEN-LIBRARY	
Bibliothek eröffnen . . . . .	373
PROVIDE-ELEMENT	
Elemente reservieren . . . . .	377

REORGANIZE-LIBRARY		
Bibliothek reorganisieren . . . . .	386	
RESET-LMS-DEFAULTS		
Rücksetzen der LMS-DEFAULTS . . . . .	388	
RESET-LOGGING-PARAMETERS		
Rücksetzen der LMS-LOGGING-PARAMETERS . . . . .	389	
RESET-TYPE-ATTRIBUTES		
Rücksetzen der Typ-Attribute . . . . .	390	
RETURN-ELEMENT		
Reservierung ausgeliehener Elemente zurückgeben . . . . .	391	
SHOW-ELEMENT		
Anzeigen des Elementinhalts . . . . .	400	
SHOW-ELEMENT-ATTRIBUTES		
Auflisten des Inhaltsverzeichnisses einer Bibliothek . . . . .	414	
SHOW-LIBRARY-ATTRIBUTES		
Bibliotheksattribute anzeigen . . . . .	434	
SHOW-LIBRARY-STATUS		
Zustand der Bibliotheken anzeigen . . . . .	437	
SHOW-LMS-DEFAULTS		
Anzeigen der LMS-Defaultwerte . . . . .	439	
SHOW-LOGGING-PARAMETERS		
LMS-Protokollparameter anzeigen . . . . .	443	
SHOW-STATISTICS		
Ausgeben von Statistiken . . . . .	444	
SHOW-TYPE-ATTRIBUTES		
Typattribute anzeigen . . . . .	448	
SHOW-USER-EXITS		
Benutzerausgänge anzeigen . . . . .	452	
WRITE-COMMENT		
Kommentar schreiben . . . . .	453	
<b>8</b>	<b>Format der LMS-Ausgaben in S-Variable . . . . .</b>	<b>455</b>
<b>8.1</b>	<b>Anweisung COMPARE-ELEMENT . . . . .</b>	<b>455</b>
<b>8.2</b>	<b>Anweisung FIND-ELEMENT . . . . .</b>	<b>456</b>
<b>8.3</b>	<b>Anweisung SHOW-ELEMENT-ATTRIBUTES . . . . .</b>	<b>457</b>
<b>8.4</b>	<b>Anweisung SHOW-LIBRARY-ATTRIBUTES . . . . .</b>	<b>460</b>
<b>8.5</b>	<b>Anweisung SHOW-STATISTICS . . . . .</b>	<b>462</b>
<b>8.6</b>	<b>Anweisung SHOW-TYPE-ATTRIBUTES . . . . .</b>	<b>464</b>



---

<b>9</b>	<b>Beispiele . . . . .</b>	<b>467</b>
<b>9.1</b>	<b>Aufnehmen, Korrigieren und Übersetzen von Quellprogrammen in Bibliotheken . . . . .</b>	<b>467</b>
<b>9.2</b>	<b>Kopieren von Elementen . . . . .</b>	<b>472</b>
<b>9.3</b>	<b>Vergleichen von Elementen . . . . .</b>	<b>476</b>
<b>9.4</b>	<b>Arbeiten mit Delta-Elementen . . . . .</b>	<b>481</b>
<b>9.5</b>	<b>Ändern eines Bindemoduls . . . . .</b>	<b>484</b>
<b>9.6</b>	<b>Erzeugen von SAM-/ISAM-Dateien . . . . .</b>	<b>486</b>
<b>9.7</b>	<b>Ausgeben von Vergleichsstatistiken . . . . .</b>	<b>490</b>
<b>9.8</b>	<b>Verzweigen in ein Benutzerprogramm beim Auflisten eines Elementes . . . . .</b>	<b>493</b>
<b>9.9</b>	<b>Schutzattribute vergeben und anzeigen . . . . .</b>	<b>497</b>
<b>9.10</b>	<b>Automatische Versionserhöhung bei Konvention NONE . . . . .</b>	<b>500</b>
<b>9.11</b>	<b>Automatische Versionserhöhung bei Konvention STD-SEQUENCE . . . . .</b>	<b>502</b>
<b>9.12</b>	<b>Automatische Versionserhöhung bei Konvention STD-TREE . . . . .</b>	<b>505</b>
<b>9.13</b>	<b>make-Lauf . . . . .</b>	<b>508</b>
<b>9.14</b>	<b>Nutzung der Ausgabe in S-Variable . . . . .</b>	<b>515</b>
<b>9.15</b>	<b>Bibliothekslisten . . . . .</b>	<b>516</b>

<b>10</b>	<b>Anhang . . . . .</b>	<b>519</b>
<b>10.1</b>	<b>Zusatzinformationen in LMS-Meldungen . . . . .</b>	<b>519</b>
<b>10.2</b>	<b>Meldungen der Zugriffsroutine AMCB . . . . .</b>	<b>521</b>
<b>10.3</b>	<b>Alte LMS-Unterprogramm-Schnittstelle . . . . .</b>	<b>523</b>
<b>10.4</b>	<b>Migration alter Bibliotheksformate . . . . .</b>	<b>526</b>
<b>10.5</b>	<b>Produktbestandteile . . . . .</b>	<b>527</b>
	<b>Literatur . . . . .</b>	<b>529</b>
	<b>Stichwörter . . . . .</b>	<b>531</b>

---

# 1 Einleitung

Dieses Handbuch beschreibt den Funktionsumfang und die Arbeitsweise des Bibliotheksprogramms LMS.

## 1.1 Kurzbeschreibung des Produkts

Das Bibliotheksverwaltungssystem LMS (Library Maintenance System) erstellt und verwaltet Programmbibliotheken und bearbeitet die darin enthaltenen Elemente.

Programmbibliotheken sind PAM-Dateien des BS2000/OSD, die mit der Bibliotheks-Zugriffsmethode PLAM (Program Library Access Method) bearbeitet werden. Daher werden sie auch als PLAM-Bibliotheken bezeichnet.

Das Bibliotheksverwaltungssystem LMS kann auch von einem Benutzerprogramm als Unterprogramm aufgerufen werden. Dadurch stehen dem Benutzer komfortable Möglichkeiten zur Verfügung, LMS-Bibliotheken und deren Inhalte direkt aus seinem Hauptprogramm zu bearbeiten. Diese Unterprogrammchnittstelle wird in einem gesonderten Handbuch [1] beschrieben.

Das Bibliotheksverwaltungssystem LMS unterstützt den Softwareentwicklungsprozess durch den Ausleihmechanismus und die make-Funktionalität.

## 1.2 Zielsetzung und Zielgruppen des Handbuchs

Dieses Handbuch wendet sich an BS2000/OSD-Anwender, die das Bibliotheksverwaltungssystem LMS für die Verwaltung ihrer Dateien und Elemente einsetzen.

Zum Verständnis dieses Handbuchs sollten Sie über BS2000/OSD-Kenntnisse verfügen, insbesondere sollten Sie mit den wichtigsten Kommandos vertraut sein.

Zudem sollten Sie die BS2000-Kommandosprache SDF (System Dialog Facility) kennen, weil die Benutzeroberfläche, Bildschirmgestaltung und Bedienerführung des LMS durch SDF bestimmt wird. Als Unterlage siehe [3].

## 1.3 Konzept des Handbuchs

Im Einzelnen finden Sie im Kapitel

- **Einleitung**  
eine Kurzbeschreibung des BS2000/OSD-Produkts LMS und Hinweise zur Benutzung des Handbuchs
- **Merkmale von LMS**  
einen Überblick über die bei LMS verwendeten Begriffe wie Programmbibliotheken, Elemente und Elementtypen und über die Ein- und Ausgabemöglichkeiten des LMS
- **Programmbibliotheken**  
Informationen über den Aufbau und den Zugriff von Programmbibliotheken
- **Elemente**  
eine Beschreibung der Elementtypen und der geltenden Konvention für die Elementbezeichnung, Hinweise zur Auswahl- und Konstruktionsangabe, Beschreibung der Elementattribute und der Beziehungen zwischen Elementen, Informationen über die Versionsverwaltung und über den Elementschutz
- **Funktionen von LMS**  
einen Umriss der Möglichkeiten, die LMS bietet
- **Unterstützung des Software-Entwicklungsprozesses**  
die make-Funktionalität und das Ausleihverfahren
- **Anweisungen**  
alle Anweisungen in alphabetischer Reihenfolge
- **Format der LMS-Ausgaben in S-Variable**  
Anweisungen, in denen S-Variablen erzeugt werden können
- **Beispiele**  
ausgewählte Beispiele für die Anwendung von LMS
- **Anhang**  
Hinweise zu Kompatibilitäten und Versionsabhängigkeiten (BS2000/OSD-BC-Produkte)

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

## 1.4 Verwendete Darstellungsmittel



Dieses Symbol weist daraufhin, dass der folgende eingerückte Absatz wichtige bzw. unabdingbare Informationen enthält.

### *Hinweis*

Das Wort „*Hinweis*“ vor einem eingerückten Absatz zeigt an, dass der folgende Absatz wichtige Informationen enthält.

[1]

Zahlen in eckigen Klammern im Text verweisen auf die entsprechende Position im Literaturverzeichnis am Ende des Handbuchs.

[ ]

Eckige Klammern in SDF-Syntaxdarstellungen: Die Zeichen innerhalb der Klammern dürfen weggelassen werden.

### **Fettdruck**

Wo SDF-Syntaxdarstellungen erläutert werden, sind die Zeilen, die aktuell erläutert werden, halbfett dargestellt.

Im Übrigen gelten für Syntaxdarstellungen die Regeln, wie sie in den entsprechenden Kapiteln des Nachschlageteils beschrieben sind.

### SYNTAX/Beispiel

SDF-Syntaxdarstellungen und Beispiel-Eingaben und -Ausgaben werden durch andere Schriften hervorgehoben. Syntaxdarstellungen sind außerdem von einem Rahmen umgeben.

## 1.5 Readme-Datei

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

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

*Informationen unter BS2000/OSD*

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

*Ergänzende Produkt-Informationen*

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

## 1.6 Änderungen gegenüber der Vorgänger-Version LMS V3.3A

### Meldungen

Das Kapitel „Meldungen“ wird nicht mehr fortgeführt.

Die Meldungen finden Sie über eine HTML-Anwendung auf dem Manual-Server (URL: <http://manuals.ts.fujitsu.com>) und auf der DVD „BS2000/OSD SoftBooks“.

Wesentliche neue Leistungen im Produkt sind:

- Änderungen in LMS V3.3B
  - Die Anweisung OPEN-LIBRARY wird um den Operanden SNAPSET erweitert. Ab BS2000/OSD-BC V7.0 kann auf Dateien aus Snapsets zugegriffen werden.
  - Der CCSN ergibt sich ab PLAM V3.4A beim Erzeugen von Elementen implizit aus dem CCSN der Bibliotheksdatei, sofern das das Element erzeugende Programm keine eigenen Vorgaben macht.
- Änderungen in LMS V3.4A
  - Ab BS2000/OSD V8.0 läuft BS2000 auch auf der SQ-Server-Linie mit Intel X86 Prozessoren. Für diese Server-Linie erstellte Elemente vom Typ L können mit //SHOW-ELEMENT angezeigt und mit //MODIFY-ELEMENT geändert werden. Man erkennt sie am HSI-CODE: X86.
  - LMS ruft EDT V17 standardmäßig im Kompatibilitäts-Modus auf. Damit ist gewährleistet, dass existierende Prozeduren weiterhin unverändert ablauffähig sind. Bei der Bearbeitung von Unicode-Elementen schaltet EDT automatisch in den Unicode-Modus.

Um Elemente zu editieren, die Sätze länger als 255 Zeichen enthalten, aber keinen Unicode Zeichensatz haben, geben Sie nach Aufruf des EDT mit der LMS-Anweisung //EDIT-ELEMENT bzw. EDT die folgenden Anweisungen ein:

```
@MODE OP=UNICODE
@OPEN L=<lib>(<elem>)
@CLOSE
```

- In der Anweisung MODIFY-ELEMENT-ATTRIBUTES ist der Operand CODED-CHARACTER-SET um den neuen Wert \*LIBRARY-DEFAULT erweitert worden.
- Die Anweisung MODIFY-LMS-DEFAULTS wird um den neuen Operanden EDT-MODE erweitert.
- Die Anweisung SHOW-LMS-DEFAULTS wird um den neuen Wert EDT-MODE für den Operanden DEFAULTS erweitert.



- Die Anweisung CALL-EDT wird um den neuen Operanden EDT-MODE erweitert.
- Die Anweisung EDIT-ELEMENT wird um den neuen Operanden EDT-MODE erweitert.
- Änderungen in LMS V3.4B
  - Die neue Anweisung EDIT-ELEMENT-ATTRIBUTES startet den geführten Dialog der Anweisung MODIFY-ELEMENT-ATTRIBUTES.
  - Die neue Anweisung EDIT-ELEMENT-PROTECTION startet den geführten Dialog der Anweisung MODIFY-ELEMENT-PROTECTION statement.

## 1.7 Voraussetzungen zur Software-Konfiguration

LMS V3.4B ist ab BS2000/OSD-BC V6.0 ablauffähig. Für einzelne Funktionen wird zusätzlich benötigt:

- BS2000/OSD-BC ab V7.0, wenn Bibliotheken auf Snapsets geöffnet werden sollen. Snapsets sind Kopien eines Pubsets auf Basis von Snap-Units. Diese vom Systemverwalter erzeugten Snapsets können als logische Sicherungen aller Dateien des Pubsets benutzt werden.
- EDT zum Bearbeiten von Textelementen und zur Ausgabe des LMS-Protokolls in eine EDT-Arbeitsdatei. Für die Bearbeitung von Unicode-Elementen wird EDT ab V17.0 benötigt.
- SDF-P für Ein- und Ausgabe in S-Variable und volle Nutzung der make-Funktionalität
- SECOS für Elementschutz mit GUARDS und die SAT-Protokollierung von sicherheitsrelevanten Elementzugriffen

---

## 2 Merkmale von LMS

Dieses Kapitel gibt einen kurzen Überblick über die Merkmale von LMS:

- Definition der Begriffe „Programmbibliothek“, „Element“ und „Delta-Technik“
- Ein- und Ausgabemöglichkeiten des LMS
- Arbeitsweise von LMS anhand eines Beispiels
- Verhalten von LMS im Dialog- und Stapelbetrieb

### 2.1 Programmbibliotheken

LMS benutzt als Bibliotheksformat **Programmbibliotheken**. Eine Programmbibliothek ist eine Datei mit Unterstruktur. Sie enthält Elemente und ein Inhaltsverzeichnis der gespeicherten Elemente. Programmbibliotheken dienen zum Speichern von Quellprogrammen, Makros, Bindemodulen, Phasen, Listen, Prozeduren, Texten usw. (Näheres siehe [Kapitel „Programmbibliotheken“ auf Seite 31](#)).

### 2.2 Elemente

Ein Element ist eine logisch zusammengehörige Datenmenge wie z.B. eine Datei, eine Prozedur, ein Bindemodul oder ein Quellprogramm. Jedes Element ist in der Bibliothek über seine Elementbezeichnung einzeln ansprechbar.

Die Elementbezeichnung identifiziert ein Element und besteht aus den drei Teilen Name, Version und Typ.

Name: Der Bestandteil Name beschreibt den logischen Inhalt des Elements.

Version: Der Bestandteil Version beschreibt den aktuellen Entwicklungsstand des Elementes.

Typ: Der Bestandteil Typ dient der Klassifizierung der Elemente.

Die Beschreibung der Elementtypen finden Sie ab [Seite 37](#).

## 2.3 Delta-Technik

Bei Verwendung der Delta-Technik werden von mehreren Versionen eines Elementes nur die Unterschiede (Deltas) zur Vorgängerversion gespeichert, was zusätzlich Speicherplatz einspart. Beim Lesen solcher Elementversionen werden diese Deltas von LMS wieder an die entsprechenden Stellen eingemischt. Dem Benutzer steht somit wieder das komplette Element zur Verfügung. Weitere Informationen zum Archivieren mit Delta-Technik finden Sie ab [Seite 96](#).

## 2.4 Eingabe- und Ausgabestrom

LMS liest alle Eingaben über die Dialogschnittstelle SDF. Als Unterlage siehe [\[3\]](#).

Das folgende Bild stellt die Ein- und Ausgabemöglichkeiten von LMS dar:

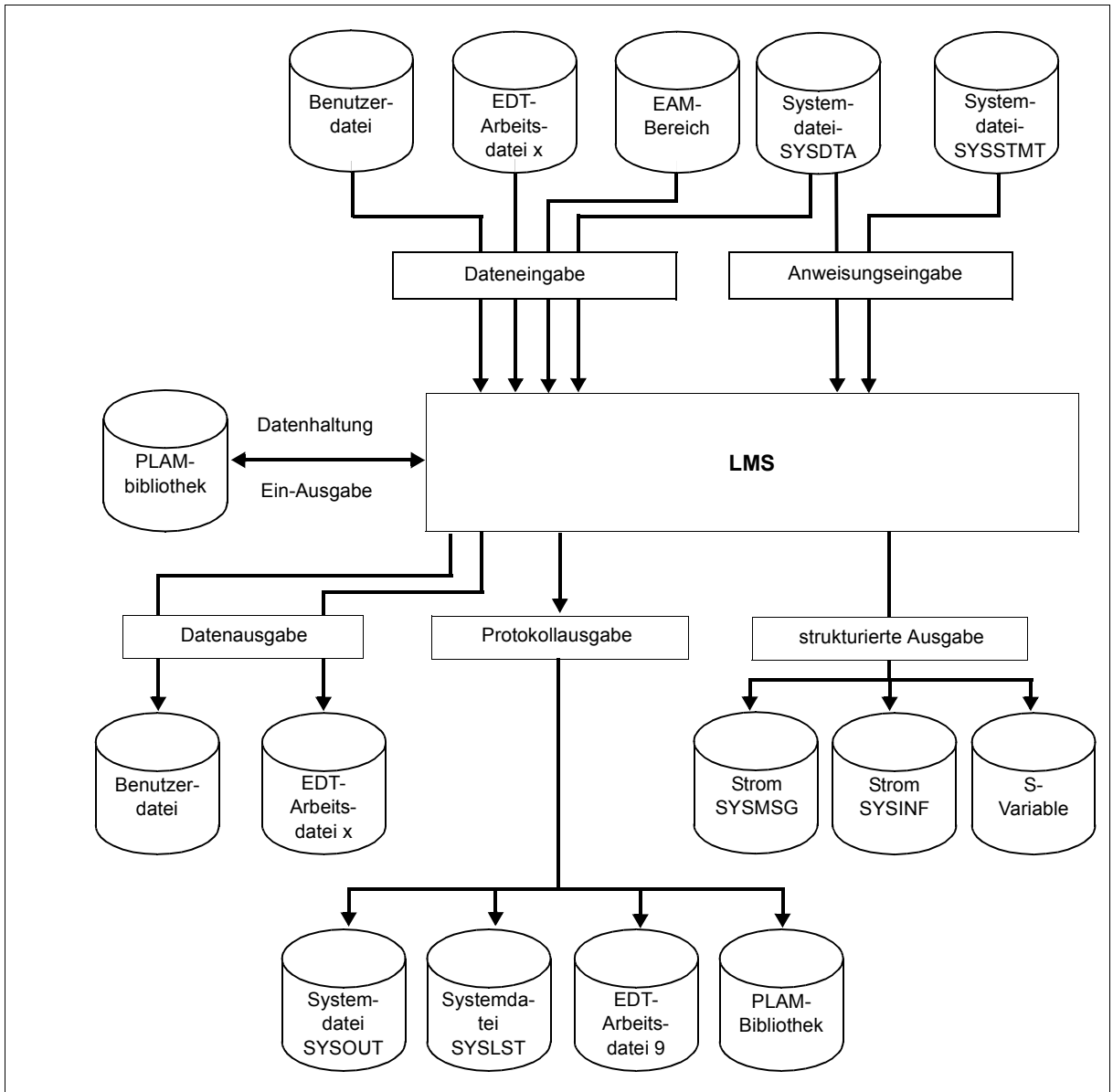


Bild 1: Zugriffsmöglichkeiten von LMS

## Ausgabe umlenken

LMS gibt standardmäßig nach SYSOUT aus. Eine Umlenkung der Ausgabe ist mit der LMS-Anweisung MODIFY-LOGGING-PARAMETERS möglich. Dabei kann die Ausgabe ausgeschaltet, in die Systemdatei SYSLST oder in ein selbst bestimmbares Bibliothekselement umgelenkt werden. Ferner ist eine Umlenkung der Ausgabe in die Arbeitsdatei 9 des EDT möglich.

### *Beispiel für einen LMS-Lauf*

Das folgende Beispiel zeigt einen kurzen LMS-Lauf, um einen ersten Eindruck von der Arbeitsweise des LMS zu vermitteln. Damit das Beispiel ablauffähig ist, müssen unter der Benutzerkennung die zwei Dateien A.EXAMPLE und A.SOURCE.A vorhanden sein. Die Benutzereingaben sind durch Kleinbuchstaben und Fettdruck gekennzeichnet, die LMS-Meldungen durch Großbuchstaben.

Das Beispiel enthält folgende Funktionen:

- Programm LMS starten
- Bibliothek neu eröffnen
- Element A.SOURCE.A in die Bibliothek neu aufnehmen
- Element A.EXAMPLE in die Bibliothek unter gleichzeitigem Umbenennen neu aufnehmen
- Inhaltsverzeichnis der Bibliothek ausgeben
- Programm LMS beenden

```

/show-file-attributes file-name=a.
0000003 :N:$USER.A.EXAMPLE
0000003 :N:$USER.A.SOURCE.A
:N: PUBLIC:      2 FILES. RES=          6, FREE=          4, REL=          0 PAGES
/start-lms _____ (1)
//open-library library=lib1,mode=*update _____ (2)
//add-element from-file=a.source.a,to-elem=*lib(type=s) _____ (3)
//modify-logging-parameters logging=*maximum _____ (4)
//add-element from-file=a.example,to-elem=*lib(elem=examp,type=d) _____ (5)

INPUT FILE
OUTPUT LIBRARY= :N:$USER.LIB1 _____ (6)

ADD :N:$USER.A.EXAMPLE AS (D)EXAMP/(0001)/2012-05-29

```

```

//show-element-attributes _____ (7)
INPUT LIBRARY= :N:$USER.LIB1 _____ (8)

TYP NAME VER (VAR#) DATE

(D) EXAMP @ (0001) 2011-08-02

    1 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS

TYP NAME          VER (VAR#) DATE

(S) A.SOURCE.A @ (0001) 2011-08-02

    1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//end _____ (9)
/

```

- (1) LMS wird aufgerufen.
- (2) LMS richtet LIB1 als globale Bibliothek ein. Die Bibliothek wird mit UPDATE eröffnet, d.h. es kann sowohl in der Bibliothek gelesen als auch in die Bibliothek geschrieben werden.  
Wenn die Bibliothek noch nicht existiert, wird sie neu angelegt.
- (3) Die Datei A.SOURCE.A wird als Element vom Typ S ohne Umbenennen in die Bibliothek aufgenommen.
- (4) Die Voreinstellung der LMS Protokollmenge wird auf maximalen Umfang geändert. LMS gibt ein komplettes Protokoll aus, d.h. nicht nur Fehlermeldungen, sondern auch Erfolgsmeldungen.
- (5) Die Datei A.EXAMPLE wird als Element vom Typ D mit der Elementbezeichnung EXAMP in die Bibliothek aufgenommen.
- (6) Erfolgsmeldung: Da die Ausgabe der Protokollmenge auf maximalen Umfang geändert wurde (vgl. (4)), bestätigt LMS die Aufnahme der Datei A.EXAMPLE als Element EXAMP.
- (7) Das Inhaltsverzeichnis der Programmbibliothek LIB1 soll aufgelistet werden.
- (8) Inhaltsverzeichniseintrag der Programmbibliothek LIB1.
- (9) LMS wird beendet.

## 2.5 Ein- und Ausgabe in S-Variable (SDF-P)

S-Variablenersetzung bei der Eingabe geschieht immer automatisch durch die Dienste der Dialogschnittstelle SDF (siehe [3]), die von LMS für alle Eingaben genutzt wird.

LMS unterstützt die Ausgabe ausgewählter Daten in S-Variable. Es werden stets Listen von Strukturen erzeugt (siehe [12]). Die Ausgabe in S-Variable ist bei den unterstützten Anwendungen lokal durch den Operanden STRUCTURE-OUTPUT= zusätzlich zum (unabhängig vom) Ausgabestrom schaltbar.

Mit folgenden Anweisungen können Informationen in S-Variable ausgegeben werden:

```
COMPARE-ELEMENT
FIND-ELEMENT
SHOW-ELEMENT-ATTRIBUTES
SHOW-LIBRARY-ATTRIBUTES
SHOW-STATISTICS
SHOW-TYPE-ATTRIBUTES
```

Die Variable muss wie folgt deklariert werden:

```
/DECL-VAR varname(TYPE=*STRUC),MULT-ELEM=*LIST
```

"varname" ist ein frei wählbarer Name. Die Variablenausgabe kann auch über den Strom SYSINF erfolgen. Dazu muss vor der Anweisung, die die Variable erzeugen soll, das Kommando /ASSIGN-STREAM STREAM=SYSINF,TO=\*VAR(varname) ausgeführt werden.

Mit dem Kommando /ASSIGN-STREAM STREAM=SYSINF,TO=\*DUMMY wird die Erzeugung von Variablen wieder gestoppt. LMS prüft intern die SYSINF-Zuordnung, falls STRUCTURE-OUTPUT=\*SYSINF, und erzeugt aus Performancegründen nur dann Ausgaben in den SYSINF-Strom, wenn die SYSINF-Zuordnung (direkt oder über SYSVAR) ungleich \*DUMMY ist.

### Unterstützung von MIP-Variablen

Ausgewählte LMS-Meldungen können in MIP-Variable (MIP = Message Improvement Program) abgelegt werden. MIP-Variable sind S-Variable, in die das Produkt MIP Meldungen ablegt. Die MIP-Variablen können in S-Prozeduren zur Steuerung des weiteren Ablaufs ausgewertet werden. Für MIP-Variable sind folgende Meldungen vorgesehen:

```
LMS0302 ELEMENT (&00) NICHT GEFUNDEN
          (&00): Elementbezeichnung der Form
                *LIB-ELEM(library,element(version),type)
LMS0303 ELEMENT (&00) NICHT IM BEREICH DER REFERENZ-BEDINGUNG
          (&00): Elementbezeichnung (Format siehe LMS0302)
```



```
LMS0310 LMS VERSION '(&00)' GESTARTET
        (&00): LMS Version der Form zz.zbzz   z=Ziffer, b=Buchstabe

LMS0311 LMS VERSION '(&00)' NORMAL BEENDET
        (&00): LMS Version (Format siehe LMS0310)

LMS0312 LMS VERSION '(&00)' ABNORMAL BEENDET
        (&00): LMS Version (Format siehe LMS0310)
```

Diese Meldungen sind in der Meldungsdatei mit dem Attribut WARRANTY=\*YES gekennzeichnet. Die Garantie bezieht sich nur auf den Meldungsschlüssel und die Inserts, jedoch nicht auf den Meldungstext.

Der Anwender muss folgende Aktivitäten ausführen, um die MIP-Variable in LMS nutzen zu können :

1. Die MIP-Variable muss wie folgt deklariert werden:

```
/DECL-VAR varname(TYPE=*STRUC),MULT-ELEM=*LIST
```

"varname" ist ein frei wählbarer Name. Mit der obigen Deklaration wird eine Liste von Strukturen vereinbart. Jedes Listenelement stellt den Behälter für genau eine Meldung dar und hat die Form (z.B. i-tes Listenelement):

```
varname#i.MSG-TEXTvollständiger Meldungstext
varname#i.MSG-IDMeldungsschlüssel
varname#i.I0Insert 0
      :
varname#i.In Insert n
```

Die Anzahl der Inserts hängt von der Meldung ab.

2. Mit dem Kommando

```
/ASSIGN-STREAM STREAM=SYSMSG,TO=*VAR(VAR-NAME=varname)
```

wird das Speichern der Meldungen in die MIP-Variable gestartet und mit dem Kommando

```
/ASSIGN-STREAM STREAM=SYSMSG,TO=*DUMMY
```

wieder gestoppt. Alle dazwischen ausgeführten Kommandos und Anweisungen (nicht nur die von LMS) speichern ihre garantierten Meldungen in die MIP-Variable.

*Beispiele*

- Es soll festgestellt werden, ob die Bibliothek LIB leer ist. In dem Fall gibt LMS die Meldung LMS0302 aus. Die Variable soll LMSMIP heißen.

```

/DECL-VAR LMSMIP(TYPE=*STRUC),MULT-ELEM=*LIST
/START-LMS
//HOLD-PROGRAM
/ASSIGN-STREAM SYMSG,TO=*VAR(LMSMIP)
/RESUME-PROGRAM
//SHOW-ELEM-ATTR *LIB(LIB),TEXT-OUTPUT=*NONE
//HOLD-PROGRAM
/ASSIGN-STREAM SYMSG,TO=*DUMMY
/IF (IS-DECLARED('LMSMIP#1.MSG-ID'))
/  IF (LMSMIP#1.MSG-ID='LMS0302')
/  WRITE-TEXT 'Bibliothek leer'
/  END-IF
/END-IF
/RESUME-PROGRAM
:
:

```

- Nicht garantierte LMS-Meldungen können mit Hilfe der SDF-P Built-in Funktion STMT-SPINOFF [12] abgefangen werden.

```

/ASSIGN-SYSDTA TO=*SYSCMD
/BEGIN-BLOCK DATA-INSERTION=YES
/START-LMS
//OPEN-LIB XXX
/A=STMT-SPINOFF()
//STEP
/IF (A EQ 'YES')
// WRITE-COM 'OPEN FEHLER'
/ELSE
// SHOW-ELEM-ATTR
/END-IF
//END
/END-BLOCK

```

## 2.6 Unterstützung der Programmentwicklung

LMS unterstützt den Prozess der Programmentwicklung durch die Funktionen „Ausleihen und Zurückgeben von Elementen“ („Ausleihmechanismus“) und durch ein Verfahren zur effizienten Aktualisierung von Programmsystemen („make-Funktionalität“).

Der **Ausleihmechanismus** ermöglicht den kontrollierten Zugriff auf Elemente, die von mehreren Entwicklern in einem Projekt verändert werden. Das Verfahren soll verhindern, dass gleichzeitig mehrere Personen an einem Element (einer Elementversion) bzw. an einer Folge von Elementen („Entwicklungslinie“) Veränderungen vornehmen. Dazu muss für die entsprechende Bibliothek bzw. für das gewünschte Element der Ausleihmechanismus aktiviert werden. Als Folge kann ein Element nur dann geschrieben werden, wenn der Schreiber vorher als aktueller Halter („holder“) für die Ausgangsversion („Basisversion“) eingetragen ist. Beim Schreiben wird dem neuen Elementstand eine Historie hinzugefügt, die den Zeitstempel der Rückgabe, die Benutzerkennung des Halters und ggf. einen benutzerspezifischen Kommentar umfasst. Der jeweilige Verarbeitungszustand eines Elementes wird als Attribut im Inhaltsverzeichnis ausgegeben und kann auch als Auswahlkriterium benutzt werden.

Die **make-Funktionalität** beschreibt die gegenseitige Abhängigkeit von Bibliothekselementen (inklusive der DVS-Dateien) in einem Programmsystem und die Regeln („Aktionen“) für die Aktualisierung. Dem Anwender stehen in LMS eine Reihe von make-Subanweisungen und die SDF-Standardanweisungen zur Verfügung.

Für eine Zielkomponente („TARGET-OBJECT“) werden die Ausgangskomponenten („FROM-OBJECT“), von denen dieses Ziel abhängt, und die Aktionen („ACTIONS“), die zur Erzeugung der Zielkomponente führen, beschrieben.

Die Aktionen werden unmittelbar oder in Form von sogenannten Standard-Aktionen angegeben; Standard-Aktionen müssen nur einmal für ein Paar von Elementtypen („TARGET-TYPE“, „FROM-TYPE“) vereinbart werden. In den einzelnen Aktionsschritten („Textzeilen“) kann über S-Variable auch Bezug auf die Komponenten genommen werden. Ausgehend von der gewählten Zielkomponente werden alle Ziele neu erzeugt, sofern sich die Ausgangskomponenten eines Zieles seit der letzten Erzeugung des Zieles verändert haben. Neben der gewählten Zielkomponente stellen auch alle zugehörigen Ausgangskomponenten weitere Unterziele dar, die analog behandelt werden. Es wird eine BS2000-Prozedur erzeugt, die synchron oder asynchron gestartet werden kann. Mit make können Programmsysteme effizient aktualisiert werden, da nur unbedingt notwendige Aktionen durchgeführt werden.

Für die wiederholte Anwendung sollte die Folge der make-Subanweisungen in einem eigenem Element, dem sogenannten „makefile“, abgelegt werden.

## 2.7 LMS im Dialog- und Stapelbetrieb

LMS läuft sowohl im Dialog- als auch im Stapelbetrieb ab.

Das LMS-Protokoll wird in die Systemdatei SYSOUT bzw. in das Medium ausgegeben, das mit MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT= vereinbart ist (Systemdatei SYSLST, Bibliothekselement oder Arbeitsdatei 9 des EDT). Wenn LMS außer Fehler- auch Erfolgsmeldungen ausgeben soll, muss die Protokollmenge durch MODIFY-LOGGING-PARAMETERS LOGGING=\*MAXIMUM festgelegt werden.

### Dialogbetrieb

Nachdem Elemente auch durch „wildcard“-Angaben ausgewählt werden können, ist zunächst nicht erkennbar, welches Element gerade bearbeitet wird. Daher ist bei Anweisungen, die Elementdaten löschen oder überschreiben, ein „schrittweises“ Vorgehen ratsam. Dieses wird über einen Dialog mit dem Benutzer gesteuert.

Im Dialog bietet LMS optional einen Fragemechanismus, mit dem der Anwender den Ablauf von elementverarbeitenden Anweisungen steuern kann.

Für jedes Element wird der Anwender gefragt, ob er das Element bearbeiten, überspringen oder die Anweisung abbrechen will.

Der Fragemechanismus wird durch den Operanden DIALOG-CONTROL =\*YES eingeleitet. Der Fragemechanismus ist in den Anweisungen ADD-, COPY-, DELETE-, EDIT-, EXTRACT-, MODIFY-ELEMENT sowie in der Anweisung MODIFY-LMS-DEFAULTS erlaubt.

Desweiteren wird in folgenden Fällen im Dialog ein Fragemechanismus eingeleitet:

- Ein Element ist nicht zugreifbar, z.B. weil es durch einen anderen Benutzer gesperrt ist (vorübergehende exklusive Nutzung eines Elementes z.B. durch Modifizieren).
- Eine Bibliothek ist nicht zugreifbar, z.B. weil die aktuellen Zugriffsrechte keinen Zugriff erlauben (vorübergehende exklusive Nutzung einer Bibliothek).

### Stapelbetrieb

Ist eine Bibliothek, ein Element oder ein Typ gesperrt, kann der Benutzer im Stapelbetrieb die Anzahl der Neuversuche und das Zeitintervall zwischen zwei Versuchen über MODIFY-LMS-DEFAULTS NEXT-ATTEMPT= einstellen. Standardmäßig erfolgt kein Neuversuch.

## 2.8 Adressierungsmodus

LMS läuft im 31-Bit-Adressierungsmodus. Bei Aufruf als Hauptprogramm wird LMS direkt im 31-Bit-Adressierungsmodus gestartet.

Bei Aufruf über die Unterprogrammchnittstellen schaltet LMS bei Bedarf in den 31-Bit-Adressierungsmodus um.

In welchem Adressierungsmodus ein User-Exit aufgerufen wird, hängt vom AMODE-Attribut der aktivierten Benutzerroutine ab:

AMODE = 31 / \*ANY :            31-Bit-Adressierungsmodus

AMODE = 24 :                 24-Bit-Adressierungsmodus



---

## 3 Programmbibliotheken

LMS bearbeitet PLAM-Bibliotheken. PLAM-Bibliotheken sind PAM-Dateien des BS2000, die mit der Bibliotheks-Zugriffsmethode PLAM (Program Library Access Method) bearbeitet werden.

Eine PLAM-Bibliothek enthält Elemente und ein Inhaltsverzeichnis der gespeicherten Elemente. PLAM-Bibliotheken dienen zum Speichern von Quellprogrammen, Makros, Binde-modulen, Phasen, Listen, Prozeduren, Text usw. PLAM-Bibliotheken haben die Eigen-schaft, dass

- alle Elementtypen in einer Bibliothek mit einheitlichen Anweisungen bearbeitet werden können,
- gleichnamige Elemente existieren können, die sich nur durch Typ- oder Versionsbezeichnung unterscheiden,
- auf die Bibliothek von mehreren Benutzern gleichzeitig auch schreibend zugegriffen werden kann,
- für die meisten während eines SW-Entwicklungsprozesses anfallenden Datenelemente eine einheitliche Datenhaltung mit einheitlichen Zugriffsfunktionen existiert und
- die Dienstprogramme und Compiler auf diese Datenhaltung zugreifen und die einzel-nen Elemente direkt bearbeiten können.

Programmbibliotheken werden im Folgenden auch kurz mit „Bibliotheken“ bezeichnet.

## 3.1 Aufbau einer Bibliothek

Eine **Bibliothek** ist eine Datei mit Unterstruktur. Sie enthält Elemente und ein Inhaltsverzeichnis.

Ein Element ist eine logisch zusammengehörige Datenmenge wie z.B. eine Prozedur, ein Bindemodul oder ein Quellprogramm. Jedes Element ist in der Bibliothek einzeln ansprechbar.

Das Speichern von Dateien als Elemente in einer Bibliothek entlastet den Dateinamenskatalog, da jede Bibliothek nur einen Katalogeintrag hat. Speicherplatz wird gespart, da die Elemente in der Bibliothek grundsätzlich in komprimierter Form gespeichert werden und darüberhinaus auch als Delta-Elemente gespeichert werden können.

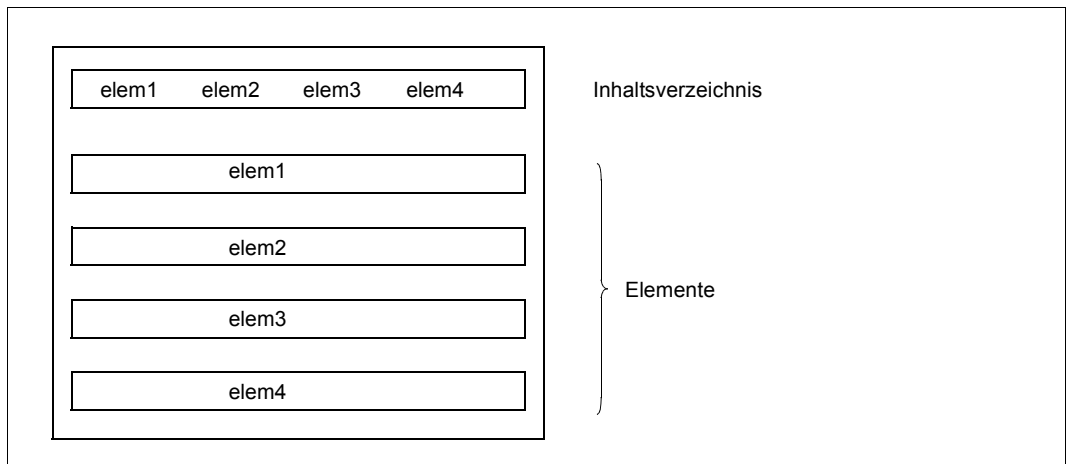


Bild 2: Logischer Aufbau einer Bibliothek

Jede Bibliothek hat einen Eintrag im Systemkatalog. Der Benutzer kann den Namen und andere Dateimerkmale, wie z.B. die Schutzfrist oder die Mehrbenutzbarkeit, festlegen. Der Eintrag im Katalog bzw. Änderungen darin werden vom Benutzer über Systemkommandos vorgenommen.



## 3.2 Ein- und Ausgabebibliotheken

LMS bearbeitet eine Bibliothek als Ein- und/oder Ausgabebibliothek:

Eine **Eingabebibliothek** wird global über die LMS-Anweisung OPEN-LIBRARY oder lokal über den Operanden ELEMENT=\*LIBRARY-ELEMENT(LIBRARY =) zugewiesen.

Eine **Ausgabebibliothek** wird global über die LMS-Anweisung OPEN-LIBRARY oder lokal über den Operanden TO-ELEMENT=\*LIBRARY-ELEMENT(LIBRARY =) zugewiesen.

Weitere Informationen über globale und lokale Bibliotheken siehe [Abschnitt „Zuweisen von Bibliotheken“ auf Seite 79](#).

## 3.3 Mehrfachzugriff auf Bibliotheken

Unter Mehrfachzugriff auf Bibliotheken versteht man den gleichzeitigen mehrfachen lesenden oder schreibenden Zugriff auf eine Bibliothek. Die Zugriffe können aus verschiedenen Tasks erfolgen. Die Tasks können verschiedenen Benutzerkennungen zugeordnet sein.

Mehrfachzugriffe auf Bibliotheken sind nur denjenigen Benutzern erlaubt, die auch das Zugriffsrecht dafür besitzen.

Eine Bibliothek kann von einem oder mehreren Benutzern sowohl lesend als auch schreibend eröffnet werden.

Im Folgenden finden Sie eine Zusammenfassung der Randbedingungen, unter denen Mehrfachzugriff möglich ist:

- Bibliotheken auf Public Volume Set (PVS)  
Mehrfachzugriff ist uneingeschränkt möglich, sofern
  - die für den Zugriff erforderlichen Zugriffsrechte gegeben sind und
  - kein Zugriff über Remote File Access (RFA) erfolgt
- Bibliotheken auf Shared Public Volume Set (SPVS)  
Mehrfachzugriff ist uneingeschränkt möglich, sofern
  - die für den Zugriff erforderlichen Zugriffsrechte gegeben sind,
  - kein Zugriff über Remote File Access (RFA) erfolgt und
  - die zugreifenden Tasks von verschiedenen Rechnern einen HIPLEX-Verbund bilden.

- Bibliotheken auf Shared Private Disk (SPD)  
Mehrfachzugriff ist uneingeschränkt möglich, sofern
  - die für den Zugriff erforderlichen Zugriffsrechte gegeben sind und
  - kein Zugriff über Remote File Access (RFA) erfolgt und
  - alle zugreifenden Tasks am gleichen Rechner ablaufen. Bei zugreifenden Tasks von verschiedenen Rechnern ist gleichzeitig nur lesender Zugriff möglich. Bei schreibendem Zugriff einer Task ist die Bibliothek für Tasks anderer Rechner gesperrt.

### 3.4 Einschränken des Mehrfachzugriffes

Der Benutzer kann den Mehrfachzugriff durch folgende Kommandos einschränken:

```
/ADD-FILE-LINK . . . ,SUPPORT=*DISK(SHARED-UPDATE=*NO)
```

PLAM-Bibliotheken werden standardmäßig mit SHARED-UPDATE=\*YES eröffnet. Ein ADD-FILE-LINK-Kommando mit SHARED-UPDATE=\*NO und ein Eröffnen dieser Bibliothek mit dem angegebenen Linknamen sperrt weitere Update-Zugriffe aus.

```
/SECURE-RESOURCE-ALLOCATION FILE=
```

Die Bibliothek wird exklusiv dieser Task reserviert.

```
/MODIFY-FILE-ATTRIBUTES . . . ,PROTECTION=
```

Nur noch Benutzer, die das erforderliche Zugriffsrecht besitzen, können auf die Bibliothek zugreifen.

Beschreibung dieser Kommandos, siehe [\[4\]](#).

---

## 4 Elemente

Ein Element ist eine logisch zusammengehörige Datenmenge wie z.B. eine Datei, eine Prozedur, ein Bindemodul oder ein Quellprogramm. Jedes Element ist in der Bibliothek über seine Elementbezeichnung einzeln ansprechbar. Die Elementbezeichnung identifiziert ein Element und besteht aus den drei Teilen Name, Version und Typ.

Name: Der Bestandteil Name beschreibt den logischen Inhalt des Elements.

Version: Der Bestandteil Version beschreibt den aktuellen Entwicklungsstand des Elementes.

Typ: Der Bestandteil Typ dient der Klassifizierung der Elemente.

Programmbibliotheken können alle von LMS unterstützten Elementtypen enthalten.

Durch die Eigenschaften einer Programmbibliothek können zu einem Projekt sämtliche Daten, vom Quellprogramm, über Bindemodule und Phasen, Übersetzungsprozeduren, Testdaten bis zur Dokumentation, in den entsprechenden Ablageeinheiten einer Bibliothek gespeichert werden.

### Arten der Speicherung

Für textartige Elemente besteht für den Benutzer die Möglichkeit, zwischen zwei Arten der Speicherungsform zu wählen: Elemente können als Voll- oder als Deltaelement gespeichert werden. Gesteuert wird die Art der Speicherung durch den Operanden STORAGE-FORM in den LMS-Anweisungen ADD-ELEMENT, COPY-ELEMENT, EDIT-ELEMENT, MODIFY-LMS-DEFAULTS, MODIFY-LIBRARY-ATTRIBUTES, MODIFY-TYPE-ATTRIBUTES, PROVIDE-ELEMENT und RETURN-ELEMENT.

Bei Verwendung der Delta-Technik werden von mehreren Versionen eines Elementes nur die Unterschiede (Deltas) zur Vorgängerversion gespeichert, was zusätzlich weiteren Speicherplatz sparen hilft. Beim Lesen solcher Elementversionen werden diese Deltas von LMS wieder an die entsprechenden Stellen eingemischt. Dem Benutzer steht somit wiederum das komplette Element zur Verfügung. Außerdem können hierarchische Beziehungen zwischen den Elementen hergestellt werden (Delta-Sequenz, Delta-Baum).

## 4.1 Mehrfachzugriff auf Elemente

Ein Voll-Element kann gleichzeitig von mehreren Benutzern gelesen, jedoch nur von einem geschrieben werden. Während ein Voll-Element zum Schreiben eröffnet ist, kann kein anderer Zugriff - auch kein lesender - auf dieses Element erfolgen, jedoch auf andere Voll-Elemente der Bibliothek.

Ein Delta-Element kann gleichzeitig von mehreren Benutzern gelesen werden.

Ist ein Delta-Element zum Schreiben eröffnet, so können alle Elemente des jeweiligen Delta-Baums mit gleichem Typ und Namen nicht von anderen Benutzern gelesen oder geschrieben werden.

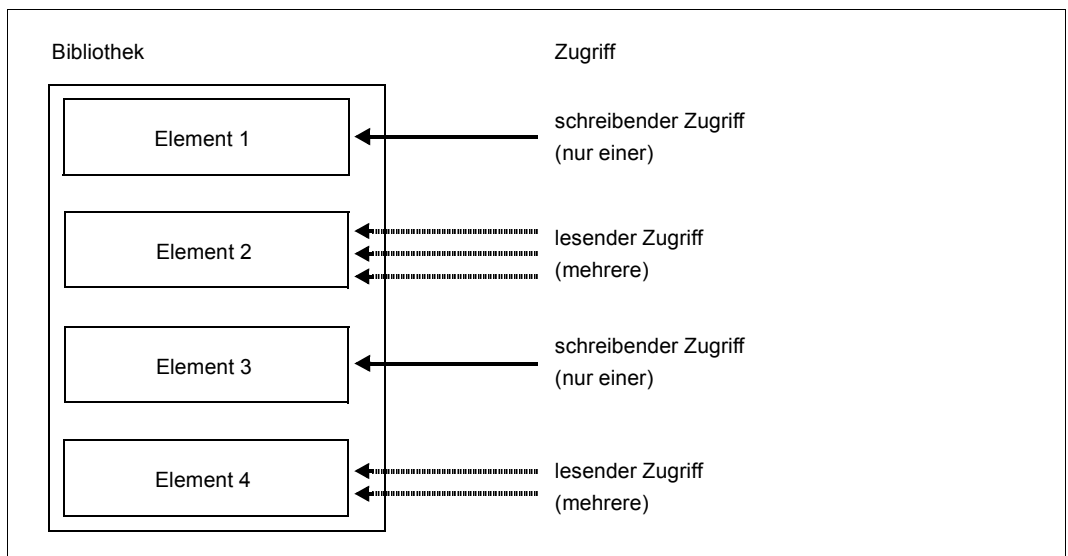


Bild 3: Mehrfachzugriff auf Elemente

Durch den Mehrfachzugriff auf eine Bibliothek ist es möglich, dass ein Element beim Auflisten des Inhaltsverzeichnisses noch vorhanden ist, beim anschließenden Zugriff jedoch nicht mehr existiert. Ein anderer Benutzer hat es in der Zwischenzeit gelöscht. Das Auflisten des Inhaltsverzeichnisses (siehe LMS-Anweisung SHOW-ELEMENT-ATTRIBUTES) zeigt also nur den momentanen Zustand der Eingabebibliothek.

Für die logische Koordination der Zugriffe auf Bibliothekselemente ist der Benutzer selbst verantwortlich.

## 4.2 Beschreibung der Elementtypen

Der Elementtyp gibt an, wie der Inhalt der Elemente von LMS zu interpretieren ist.

### 4.2.1 Standardtypen

Standard- oder vordefinierte Typen sind ein Zeichen lang bzw. beginnen mit \$ oder SYS:

Typ	Bedeutung
\$...	reserviert
C	Phasen
D	Dokumente
F	IFG-Formatmasken
H	Compiler-Ergebnisinformationen
J	Prozeduren und ENTER-Jobs
L	Bindeladmodule (LLM)
M	Makros
P	druckaufbereitete Daten
R	Bindemodule
S	Quellprogramme
SYS...	reserviert
SYSJ..	kompilierte Prozeduren
U	IFG-Benutzerprofile
X	Daten beliebigen Formats

Tabelle 1: Standardtypen

Zur Archivierung können Elemente mit einer Satzlänge von maximal 32 KByte (inkl. Satz-kopf) aufgenommen werden.

Der EDT bearbeitet Textelemente mit einer maximalen Satzlänge von 256 Byte.

**Elementtyp \$... - reservierte Typen**

Mit \$ beginnende Typen sind reserviert.

**Elementtyp C - Phasen**

Eine vom Binder TSOSLNK erzeugte Phase wird standardmäßig in einer Datei abgelegt. Diese Datei kann mit LMS in einer Bibliothek als Element vom Typ C abgelegt werden. Wahlweise können die vom Binder erzeugten Phasen auch direkt in einer Bibliothek abgelegt werden.

**Elementtyp D - Dokumente**

Elemente vom Typ D sind für Dokumente gedacht.

**Elementtyp F - IFG-Formatmasken**

Elemente dieses Typs werden von IFG erzeugt und in Bibliotheken abgelegt.

**Elementtyp H - Compiler-Ergebnisinformationen**

Elemente dieses Typs werden von den Compilern und vom Assembler erzeugt und in Bibliotheken abgelegt.

**Elementtyp J - Prozeduren und ENTER-Jobs**

In diesem Elementtyp werden BS2000-Prozeduren und ENTER-Jobs abgelegt.

**Elementtyp L - Bindelademodule (LLM)**

In Elemente vom Typ L legen sowohl der Binder BINDER (siehe [5]) als auch die Compiler die erzeugten Bindelademodule (LLMs) ab.

**Elementtyp M - Makros**

Der Assembler entnimmt der zugewiesenen Bibliothek die im Programm angesprochenen Makroelemente.

**Elementtyp P - Listenelemente**

Als Listenelemente werden druckaufbereitete Daten bezeichnet. Das erste Zeichen des Satzes muss ein gültiges Vorschubsteuerzeichen sein; dies wird bei der Ausgabe in die Systemdatei SYSLST geprüft.

**Elementtyp R - Bindemodule**

Von Compilern und dem Assembler erzeugte Bindemodule werden standardmäßig im temporären EAM-Bereich abgelegt.

Diese Bindemodule können mit LMS in der Bibliothek als Elemente vom Typ R abgelegt werden. Wahlweise können die von Compilern und dem Assembler erzeugten Bindemodule auch direkt in einer Bibliothek abgelegt werden.

Den Bindern und dem dynamischen Bindelader DBL dienen diese Elemente als Eingabe.

**Elementtyp S - Quellprogramme**

Quellprogramme in Bibliotheken dienen den Compilern und dem Assembler bei Übersetzungsläufen als Eingabe.

**Elementtyp SYS... - reservierte Typen**

Mit SYS beginnende Typen sind reserviert.

**Elementtyp SYSJ.. - kompilierte Prozeduren**

In diesem Elementtyp werden kompilierte Prozeduren abgelegt. Die Elementtypen SYSJ und J werden von LMS gleichwertig behandelt.

**Elementtyp U - IFG-Benutzer-Profile**

Elemente dieses Typs werden von IFG erzeugt und in Bibliotheken abgelegt.

**Elementtyp X - Daten beliebigen Formates**

Der Elementtyp X kann beliebige Daten aufnehmen. Je nach Format der Daten werden sie in Text- oder PAM-Elementen abgelegt.

**Textartige Elementtypen - Textelemente**

Als textartige Elementtypen werden die Typen S, M, J, P, D, X und davon abgeleitete Typen bezeichnet. Textelemente sind Elemente dieser Typen, soweit sie keine blockorientierten Sätze enthalten.

**PAM-Elemente**

Elemente mit blockorientierten Sätzen werden im Folgenden auch als PAM-Elemente bezeichnet (vom Typ X oder daraus abgeleitete Typen), da sie in der Regel durch Aufnehmen einer Datei vom Typ FILE-STRUCTURE=PAM als Element entstanden sind.

## 4.2.2 Abgeleitete Typen

Neben den Standardtypen kann der Anwender auch eigene Typen, sog. benutzerdefinierte Typen, verwenden. Benutzerdefinierte Typen sind zwischen zwei und acht Zeichen lang und beginnen nicht mit \$ oder SYS. Die Verwendung von benutzerdefinierten Typen anstelle von Standardtypen in allen Kommandos und Programmen ist durch einen Umlenkmechanismus möglich (siehe [Abschnitt „Zuweisen von Bibliotheken“ auf Seite 79](#)).

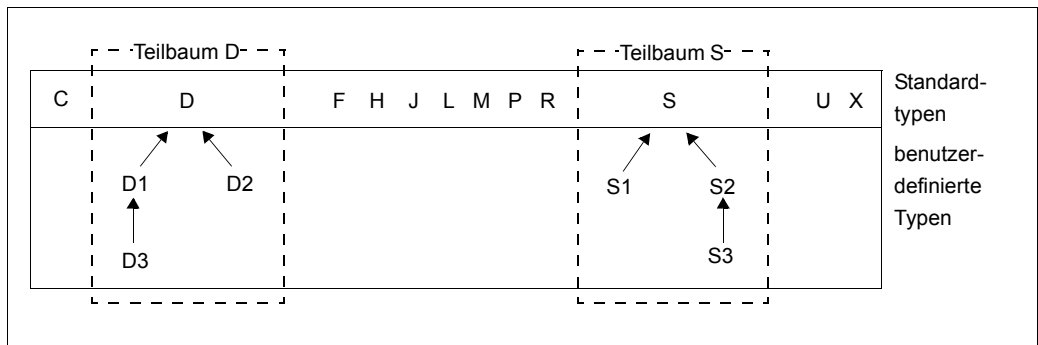
### Supertyp

Jedem benutzerdefinierten Typ kann durch MODIFY-TYPE-ATTRIBUTES ein Supertyp - d.h. ein Typ am Typbaum über ihm - zugeordnet werden. LMS behandelt dann die Elemente des benutzerdefinierten Typs genauso, wie die Elemente seines Supertyps. Der Supertyp kann selbst wiederum ein benutzerdefinierter Typ sein. Damit entstehen u.U. mehrere Typbäume, an deren Spitze jeweils ein Standardtyp stehen sollte.

### Basistyp

Der Typ an der Spitze eines Typbaums heißt Basistyp und sollte ein Standardtyp sein. Er legt fest, wie die von ihm abgeleiteten benutzerdefinierten Typen von LMS behandelt werden sollen.

*Beispiel*



D hat keinen Supertyp und damit sich selbst als Basistyp.

D1 hat D als Supertyp und damit auch D als Basistyp.

D3 hat D1 als Supertyp und damit D als Basistyp.



## 4.3 Konvention für Elementbezeichnungen

Elemente werden in Bibliotheken über eine **Elementbezeichnung** identifiziert. Diese wird im Inhaltsverzeichnis der Bibliothek hinterlegt und kann mit der LMS-Anweisung SHOW-ELEMENT-ATTRIBUTES ausgegeben werden.

Die Elementbezeichnung besteht aus den drei folgenden Teilen:

- Elementname für den logischen Inhalt der Elemente,
- Elementversion für den aktuellen Stand der Elemente,
- Elementtyp für die Klassifizierung der Elemente.

### 4.3.1 Elementbezeichnung in den Anweisungen

Die Elementbezeichnung, d.h. Elementname, -version und -typ, entspricht in den LMS-Anweisungen den Operanden ELEMENT, VERSION und TYPE in der Datenstruktur (LIBRARY-)ELEMENT.

Die Angabe der Version ist wahlfrei. Wird in einer Anweisung kein Wert für Version eingegeben, wird standardmäßig das Element mit der höchsten Version ausgewählt. Soll eine andere Version ausgewählt werden, so ist zu beachten, dass die Versionsangabe eine Unterstruktur des Elementnamens ist:

```

.
.
ELEMENT = <composed-name ...>(....)

    <composed-name ...>(....)
    |
    |  VERSION = ....
    |
    |  ,BASE = .....

,TYPE = .....
.
.

```

Der Elementtyp hat nach dem Aufruf von LMS standardmäßig den Wert \*UNDEFINED. Er muss entweder global mit der Anweisung MODIFY-LMS-DEFAULTS oder lokal in jeder Anweisung vereinbart werden.

### Syntax der Elementbezeichnungen

Die hier verwendete Metasyntax wird ab [Seite 147ff](#) beschrieben. Sie ist für Quell- und Zielelemente gleich.

ELEMENT	<composed-name 1..64 with-underscore_with-wildcards(132)>(…). Der Elementname darf mit einer maximal vierstelligen cat-id beginnen.
VERSION	<composed-name1..24 with-underscore_with-wildcards(52)> Weitere Informationen zu den möglichen Versionsangaben, vor allem über die Verwendung und Bedeutung von Schlüsselwörtern, siehe <a href="#">Seite 54</a> . Die Version '@' ist nicht erlaubt, da sie als Darstellung für die Versionsangabe '*UPPER-LIMIT' dient.
BASE	<composed-name 1..24 with-underscore_with-wildcards> Weitere Informationen zu den möglichen Angaben siehe <a href="#">Seite 56f</a> .
TYPE	<alphanum-name 1..8 with-wildcards(20)> Elementtypen, die nur aus einem Zeichen bestehen oder mit \$ bzw. SYS beginnen, werden als Standardtypen interpretiert. Dies sind reservierte Typen und sollten nicht als benutzerdefinierter Typ verwendet werden. Benutzerdefinierte Typen sind mindestens 2 und maximal 8 Zeichen lang.

### 4.3.2 Protokollierung der Elementbezeichnung

Bei jeder Ausgabe von LMS werden die Elementbezeichnungen wie folgt protokolliert:

(Typ)Elementname/Version[(Variantennummer)]/Datum

Die Variantennummer ist standardmäßig auf (0001) gesetzt und wird durch jeden schreibenden Zugriff um 1 erhöht.

### 4.3.3 Auswahlangabe für Elementbezeichnungen

Sollen in den LMS-Anweisungen bestimmte Elemente zur Bearbeitung ausgewählt werden, so ist dies auf zwei Arten möglich:

- durch die Verwendung von „wildcard“-Angaben in den Operanden ELEMENT, VERSION und TYPE.

Die „wildcard“-Syntax ist innerhalb des Abschnitts „Syntaxbeschreibung“, der auf [Seite 146ff](#) zu finden ist, beschrieben.

Bei den Operanden ELEMENT, VERSION und TYPE kann für den Einzelstern (\*) auch das Schlüsselwort \*ALL verwendet werden, in der Struktur EXCEPT-ELEMENT jedoch nicht.

Eine Negativauswahl, d.h. Ausschließen von Elementen, kann neben dem Minuszeichen auch durch den Operanden EXCEPT-ELEMENT vorgenommen werden. Dieser Operand bietet mehr Möglichkeiten als das Minuszeichen.

- durch die Qualifikation von Attributen, z.B. Datum und Uhrzeit.

*Beispiele für die Auswahlangabe*

- Auswahlangabe

ELEMENT = AB/C\*

Alle Elemente, deren Name mit AB beginnt, an der 3. Stelle ein beliebiges Zeichen, an der 4. Stelle ein C besitzt, werden ausgewählt. Ab der 5. Stelle kann der Elementname beliebigen Inhalt haben.

ELEMENT = <:999>(VERSION=B\*)

Alle Elemente mit einer Namenslänge von maximal 3 Zeichen, die an der ersten Stelle der Version ein B besitzen, werden ausgewählt.

ELEMENT = \*(VERSION=\*), CREATION-DATE = INTERVAL(FROM=2013-01-01)

Alle Elemente, die seit dem 1.1.2013 eingetragen wurden, werden ausgewählt.

ELEMENT = AB\*(VERSION=\*HIGHEST-EXISTING)

Von allen Elementen, deren Name mit AB beginnt, wird jeweils die höchste Version ausgewählt.

## – Auswahlangaben mit Grenzwerten

ELEMENT = A\*(VERSION = \*HIGH), USER-DATE = INTERVAL(TO=12-12-31)

Alle Elemente der höchsten Version, deren Name mit A beginnt, und die älteren Datums als 1.1.2013 sind, werden ausgewählt.

ELEMENT = AB<:9>(V=107)

Alle Elemente, deren Name mit AB beginnt und maximal 3 Zeichen lang ist, und die die Version 107 haben, werden ausgewählt.

## – Auswahlangaben mit auszuschließenden Elementen

ELEMENT = -ABC

Alle Elemente, außer dem Element ABC, werden ausgewählt.

ELEMENT = A\*, EXCEPT-ELEMENT = (ELEMENT = \*ANY(VERSION = B\*))

Alle Elemente, deren Name mit A beginnt, und deren Version nicht mit B beginnt, werden ausgewählt.

ELEMENT = \*(VERSION = <402:>\*), EXCEPT-ELEMENT=(EL=\*(V=402))

Alle Elemente, deren Version größer als 402 ist, werden ausgewählt.

ELEMENT = L<:999>, EXCEPT-ELEMENT = (ELEMENT=\*ANY(VERSION=1))

Alle Elemente, deren Name mit L beginnt und maximal 4 Zeichen lang ist, außer denen, die Version 1 haben, werden ausgewählt.

### 4.3.4 Konstruktionsangabe für Elementbezeichnungen

Bei LMS-Anweisungen, die neben der Auswahlangabe auch eine zweite Elementbezeichnung zulassen, kann die Bezeichnung des zweiten Elementes aus der Bezeichnung der Auswahlangabe konstruiert werden.

Bei den folgenden LMS-Anweisungen ist eine solche Konstruktionsangabe möglich:

- ADD-ELEMENT
- COMPARE-ELEMENT
- COPY-ELEMENT
- EDIT-ELEMENT
- EXTRACT-ELEMENT
- MODIFY-ELEMENT
- MODIFY-ELEMENT-ATTRIBUTES
- PROVIDE-ELEMENT
- RETURN-ELEMENT

Die Konstruktionsangabe beschränkt sich auf die Elementbezeichnung, d.h. auf Elementname, -version und -typ. Dies entspricht in den Anweisungen den Operanden ELEMENT, VERSION und TYPE in der Datenstruktur LIBRARY-ELEMENT. Dabei werden jeweils gleichnamige Operanden, die durch bestimmte Platzhalter gekennzeichnet sind, aufeinander abgebildet:

Auswahlangabe:

ELEMENT    = \*LIBRARY-ELEMENT (ELEMENT=... (VERSION=...), TYPE=...)

Konstruktionsangabe:

TO-ELEMENT = \*LIBRARY-ELEMENT (ELEMENT=... (VERSION=...), TYPE=...)

Die Syntax ist im [Abschnitt „Syntaxbeschreibung“ auf Seite 146ff](#) beschrieben.

*Hinweise*

- Der Operand EXCEPT-ELEMENT wird bei der Konstruktionsangabe nicht berücksichtigt.
- Platzhalter in der Auswahlangabe erfordern mindestens einen Platzhalter in der Konstruktionsangabe. \*ALL in der Auswahlangabe wirkt wie der Einzelstern (\*) und erfordert daher auch einen Platzhalter in der Konstruktionsangabe.  
\*ALL in der Konstruktionsangabe (außer bei COMPARE-ELEMENT) ist nicht möglich.  
(Beispiel: \*ALL → \*B\* ist gleichbedeutend mit \* → \*B\*)

## 4.4 Elementattribute

Alle Elemente besitzen unabhängig von ihrem Typ gewisse Attribute. Dabei wird unterschieden zwischen Attributen, die von der Zugriffsmethode vergeben werden, und Attributen, die vom Benutzer vergeben werden können. Die Werte dieser Attribute werden bei der Ausgabe des Inhaltsverzeichnisses einer Bibliothek angezeigt.

### 4.4.1 Attribute, die von der Zugriffsmethode vergeben werden

Von der Zugriffsmethode werden für Elemente folgende Attribute geführt:

Erstellungsdatum und -zeit (CREATION-DATE und CREATION-TIME)	(1)
Änderungsdatum und -zeit (MODIFICATION-DATE und MODIFICATION-TIME)	(2)
Zugriffsdatum und -zeit (ACCESS-DATE und ACCESS-TIME)	(3)
Speicherungsform (STORAGE-FORM)	
Sekundärnamen und -attribute (SECONDARY-NAME u. SECONDARY-ATTRIBUTE)	
Elementgröße	(4)

- (1): Bei deltagespeicherten Elementen ist dieses Datum für alle Deltas gleich und bezieht sich auf den Zeitpunkt der Erstellung der ersten Daten.
- (2): Bei der Anweisung MODIFY-ELEMENT-ATTRIBUTES kann mit dem Operanden MODIFICATION-DATE gesteuert werden, ob dieser Zeitstempel aktualisiert werden soll.
- (3): Dieser Zeitstempel wird nur geführt, wenn für die Bibliothek MODIFY-LIBRARY-ATTRIBUTES ...,ACCESS-DATE=\*KEEP eingestellt ist. Erst ab diesem Zeitpunkt können Elemente Zugriffsdatum und -zeit besitzen. Außer beim Zugriff über Remote File Access (RFA) ist zum Führen von Zugriffsdatum und -zeit kein Schreibrecht für die Bibliotheksdatei erforderlich. Ist beim Zugriff über RFA das Führen von Zugriffsdatum und -zeit gewünscht, aber kein Schreibrecht für die Bibliotheksdatei gegeben, wird der Zugriff abgewiesen. Für Bibliotheken auf NK-Platten wird der Umfang einer partiellen Sicherung von PLAM festgestellt. Das Sicherungsgranulat ist ein einzelnes Element. Elemente, bei denen sich nur das Zugriffsdatum ändert, werden nicht Bestandteil der partiellen Sicherung. Im Falle eines Restores erhalten solche Elemente damit das Zugriffsdatum zum Zeitpunkt der letzten Vollsicherung der Bibliotheksdatei.
- (4): Bei deltagespeicherten Elementen bezieht sich die Größe auf die komplette Speicherplatzbelegung des Delta-Baumes und nicht auf die der einzelnen Versionen.

## 4.4.2 Attribute, die vom Benutzer vergeben werden können

Der Benutzer kann in den LMS-Anweisungen MODIFY-ELEMENT-ATTRIBUTES bzw. MODIFY-ELEMENT-PROTECTION folgende Attribute bestimmen:

- Organisatorische Attribute:  
Zugriffsrechte (PROTECTION)  
Elementzustand (STATE)
- Inhaltsbeschreibende Attribute:  
Benutzerdatum und -zeit (USER-DATE und USER-TIME)  
Name eines Zeichensatzes (CODED-CHARACTER-SET)

## 4.4.3 Eingabeformat des Datums

Das Eingabeformat eines Datums lautet:

[JJ]JJ-MM-TT	mit [JJ]JJ	: Jahr; wahlweise zwei- oder vierstellig
	MM	: Monat
	TT	: Tag

Bei einer zweistelligen Jahresangabe ergänzt LMS die Jahrhundertangabe durch die Verwendung eines Referenzjahres:

Ist  $JJ < 60$ , dann ergänzt LMS zu 20JJ,

ist  $JJ \geq 60$ , dann ergänzt LMS zu 19JJ.



## 4.5 Beziehungen zwischen Elementen

Für die Beschreibung zeitlicher und logischer Abhängigkeiten, d.h. die Beziehung zwischen den Elementen, gibt es folgende Möglichkeiten:

### Delta-Baum

Ein Delta-Baum ist eine Menge von Elementen, die durch die Beziehung "Element-ist-Nachfolger-von" gebildet wird.

### Namenskonventionen

Innerhalb eines Zweiges eines Delta-Baums haben die Elemente den gleichen Namen und Typ. Sie unterscheiden sich nur durch ihre Version.

### Referenzeinträge

Referenzeinträge sind Einträge im Sekundärinhaltsverzeichnis der Bibliothek. Sie entstehen, wenn der Anwender beim Schreiben eines Elementes Referenzsätze (Satzart 163) der Form <Sekundärname><Sekundärattribut> erzeugt. Das Layout der Satzart 163 ist in [1] beschrieben.

Durch die Referenzeinträge wird die Beziehung „Ein bestimmter <sekundärname> und <sekundärattribut> kommt im Element vor“ dokumentiert. Die typweise Sortierung der Referenzeinträge erlaubt die Abfrage:

„In welchem Element des Typs TYPE kommt ein bestimmter Referenzeintrag vor?“.

Dies wird durch die folgende LMS-Anweisung realisiert:

```
//SHOW-ELEMENT-ATTRIBUTES *LIB(*STD,*,TYPE=...,SECONDARY-NAME=...,  
SECONDARY-ATTRIBUTE=...,)
```

Diese Beziehung wird bei Modulen (Typ R oder L) als Basis der Autolink-Funktion (siehe [5]) verwendet; Referenzeinträge sind z.B. <name><CSECT> bzw. <name><ENTRY>.

PLAM gewährleistet die Konsistenz der Beziehung; für die Erzeugung und Vollständigkeit der Beziehungen ist der Benutzer verantwortlich.

### Abhängigkeiten vom Elementtyp

LMS besitzt Anweisungen,

- die unabhängig vom Elementtyp sind (ohne Bezug auf den Elementinhalt) und
- andere, die sich am Elementtyp orientieren.  
 Im letzteren Falle werden nur einige wenige Standardtypen zugelassen; benutzerdefinierte Typen müssen sich aus diesen Typen ableiten. Elemente mit abweichendem Typ können nicht bearbeitet werden.

Die folgende Tabelle zeigt, welche Elementtypen bei den einzelnen LMS-Anweisungen möglich sind und welche Typprüfungen LMS während des LMS-Laufs vornimmt:

Nur wenn die in der Spalte Typprüfung aufgeführten Bedingungen zutreffen, wird die Anweisung ausgeführt.

Anweisung	Elementtyp		
	Quelle	Ziel	Typprüfung
SHOW-ELEMENT	alphanum-name1..8	-	
SHOW-ELEMENT-ATTRIBUTES	alphanum-name1..8	-	
DELETE-ELEMENT	alphanum-name1..8	-	
MODIFY-ELEMENT-PROTECTION	alphanum-name1..8	-	
FIND-ELEMENT	alphanum-name1..8	-	
MODIFY-ELEMENT-ATTRIBUTES	alphanum-name1..8	alphanum-name1..8	BT(q)=:BT(z)
COPY-ELEMENT	alphanum-name1..8	alphanum-name1..8	BT(q)=:BT(z)
PROVIDE-ELEMENT	alphanum-name1..8	alphanum-name1..8	BT(q)=:BT(z)
RETURN-ELEMENT	alphanum-name1..8	alphanum-name1..8	BT(q)=:BT(z)
COMPARE-ELEMENT	alphanum-name1..8	alphanum-name1..8	BT(p)=:BT(s)
EDIT-ELEMENT	S, M, P, D, J, X, abgeleiteter Typ	S, M, P, D, J, X, abgeleiteter Typ	
MODIFY-ELEMENT	R, C, L S, M, P, D, J, X, abgeleiteter Typ	R, C, L S, M, P, D, J, X, abgeleiteter Typ	BT(q)=:BT(z) BT(q)=:BT(z)
MODIFY-LOGGING-PARAMETERS	-	S, M, P, D, J, X, abgeleiteter Typ	

Anweisung	Elementtyp		
	Quelle	Ziel	Typprüfung
ADD-ELEMENT	„text“ (l)SAM-f „blocks“ PAM-file „module“ file,*OMF „phase“ PAM-file	S, M, P, D, J, X, abgeleiteter Typ X abgeleiteter Typ R C	
EXTRACT-ELEMENT	S, M, P, D, J, X, abgeleiteter Typ X abgeleiteter Typ R C	(l)SAM-file  PAM-file (l)SAM-file PAM-file	

BT Basistyp (Name des Typs des höchsten Knotens)

BT(q)/BT(z) Basistyp des Quell- bzw. Zielelements

BT(p)/BT(s) Basistyp des Primär- bzw. Sekundärelements

Standardtyp ein Zeichen lang bzw. mit \$ oder SYS am Anfang

abgeleiteter von den aufgezählten Typen abgeleitet;

Typ alphanum 2..8 und kein \$ oder SYS am Anfang

:= Die Typen links und rechts dieses Zeichens sind entweder gleich oder beide textartig. Soweit der Typ X betroffen ist, bedeutet textartig, dass nur Textelemente beteiligt sein dürfen.

## 4.6 Versionsverwaltung

Die Version eines Elementes ist in der Elementbezeichnung festgelegt und kennzeichnet den aktuellen Stand des Elementes.

Der folgende Abschnitt beschreibt die möglichen Versionsbezeichnungen, gibt Hinweise zur Versionshaltung und -speicherung und geht auf die Versionskonventionierung ein, die für die automatische Versionserhöhung relevant ist.

### 4.6.1 Versionshaltung und -speicherung

Es ist charakteristisch für Software-Elemente, dass sie leicht änderbar sind und erfahrungsgemäß auch häufig geändert werden. Stabile und relevante Stände eines Entwicklungsobjektes werden daher als Elemente festgehalten.

#### Mehrere Versionen je Elementtyp und -name

In Bibliotheken wird ein Element durch seinen Typ, seinen Namen und eine Versionsbezeichnung eindeutig bestimmt. Weiter ist es möglich, zu einem Elementtyp und -namen mehrere Versionen zu speichern.

Falls der Benutzer bei der Bearbeitung keine Angaben zur Version macht, führt LMS standardmäßig folgende Aktionen aus:

- Beim Lesen  
wird das Element gesucht, das zu dem angegebenen Namen die höchste Versionsbezeichnung besitzt. Das Datum wird dabei nicht berücksichtigt.
- Beim Schreiben  
hängt die Versionsbehandlung von der Anweisung ab:
  - ADD-ELEMENT und MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=  
Das Element wird mit der höchsten Version 'X'FF' erzeugt bzw. überschrieben. LMS stellt diese Version mit @ dar.
  - sonstige Anweisungen  
Das Ausgabeelement erhält die Versionsbezeichnung des Eingabeelementes.

Wird dabei ein gleichnamiges Element überschrieben, wird die interne Variantenummer um 1 erhöht. Sie dient als Schreibzugriffszähler.

Verschiedene Stände eines Entwicklungsobjektes sind in verschiedenen Elementen abgelegt. Die Beziehungen zwischen einzelnen Elementen kennt zunächst nur der Anwender; sie sind nicht in der Bibliothek verankert. Jedes Element ist eine eigenständige Einheit im Kontext einer Bibliothek.

## Delta-Speicherung

Elemente, die aus einem Vorgängerelement durch Änderung entstanden sind, unterscheiden sich im Allgemeinen nur wenig vom Vorgänger. Daher ist es sinnvoll, diese Menge von Elementen kompakter zu speichern. D.h. es werden nur die Unterschiede zum jeweiligen Vorgänger physikalisch gespeichert. Diese kompakte Speicherung der Differenzmenge ergibt den Delta-Baum.

Alle Elemente eines Delta-Baums, die keinen Nachfolger haben, heißen Blätter. Nur Blätter können überschrieben werden.

## Delta-Bäume

Erst die Verwendung von Delta-Bäumen gestattet es, eine Beziehung zwischen den Elementen auch in der Bibliothek festzuhalten. Für solche Elemente gilt

- auf der Elementbezeichnungsebene  
Da das neue Element durch eine Änderung aus einem anderen Element entstand, gehört es immer noch der gleichen Klasse an und beschreibt den gleichen logischen Inhalt. Beide Aspekte sind im Typ und Namen des Elementes festgeschrieben, d.h. alle Elemente eines Delta-Baumes haben den gleichen Typ und Namen.  
Der Bestandteil Version einer Elementbezeichnung sollte den Stand beschreiben.
- auf der physikalischen Ebene  
Die Beziehung der beiden Elemente ist auch auf physikalischer Ebene festzuhalten. Welche Elemente zu verknüpfen sind, wird jedoch nicht anhand des Versionsstrings ermittelt, sondern aus Angaben des Benutzers abgeleitet. Zu diesem Zweck muss der Benutzer ein Element als "Vorgänger" und ein Element als "Nachfolger" deklarieren.

## 4.6.2 Versionsbezeichnungen

Die durch LMS zu bearbeitenden Elementversionen werden in den LMS-Anweisungen durch Angabe der Operanden VERSION und BASE bestimmt. Es wird unterschieden zwischen Bestimmung der Quellversion, der Zielversion und der Basisversion zur Zielversion.

### Quellversion

Wird ein Element als *Eingabe* einer Funktion verwendet, z.B. wenn es kopiert oder verändert werden soll, dann wird die Elementversion *Quellversion* genannt.

Die Quellversion kann wie folgt bestimmt werden:

- composed-name  
Die durch composed-name bestimmte Version wird ausgewählt.
- \*UPPER-LIMIT  
Die höchst mögliche Version (intern durch X'FF' dargestellt) wird ausgewählt.
- \*HIGHEST-EXISTING,BASE=\*STD  
Die höchste existierende Version des angegebenen Elementnamens wird ausgewählt.
- \*HIGHEST-EXISTING,BASE=composed-name\*  
Die höchste existierende Version mit dem Präfix composed-name wird ausgewählt.

### Zielversion

Wird das Element als *Ergebnis* einer Funktion verwendet, z.B. wenn es zurückgeschrieben wird, dann wird die Elementversion *Zielversion* genannt.

Die Zielversion kann wie folgt bestimmt werden:

- \*BY-SOURCE  
Die Quellversion ist auch Zielversion. Ist die Quelle kein Bibliothekselement, ist \*UPPER-LIMIT Zielversion.
- composed-name  
Die durch composed-name bestimmte Version ist Zielversion. Die Angabe von '@' wird zurückgewiesen.
- \*UPPER-LIMIT  
Die absolut höchste Version, intern dargestellt durch X'FF', ist Zielversion.
- \*INCREMENT  
Bestimmt eine Basisversion, abhängig von der unter dem Typ geltenden Konvention und der bei BASE gemachten Angabe. Auf diese Basisversion wird wiederum ein von der geltenden Konvention abhängiges Erhöhungsverfahren angewandt. Die dadurch bestimmte Version ist Zielversion.

Zur ersten Version eines Elementes kann noch keine Basisversion existieren. In diesem Falle ist die von der unter dem Typ geltenden Konvention abhängige Defaultversion Zielversion.

Konvention	Defaultversion
NONE	001
STD-SEQUENCE	EXAMPLE
STD-TREE	001.001
MULTI-SEQUENCE	EXAMPLE

– **\*HIGHEST-EXISTING**

Bestimmt eine Basisversion, abhängig von der unter dem Typ geltenden Konvention und der bei BASE gemachten Angabe. Diese Basisversion ist zugleich Zielversion.

Zur ersten Version eines Elementes kann noch keine Basisversion existieren. In diesem Falle ist die von der unter dem Typ geltenden Konvention abhängige Defaultversion Zielversion.

### Basisversion zur Zielversion

Abgesehen von der ersten Version eines Namens wird eine Zielversion immer zu einer Basisversion erzeugt. Die Basisversion erfüllt mehrere Funktionen:

1. Soll eine Version mit STORAGE-FORM=\*DELTA erzeugt werden, so ist durch die Basisversion festgelegt, zu welcher Version das Delta gebildet werden soll.
2. Soll die Zielversion durch den Versionsautomatismus (VERSION=\*HIGHEST-EXISTING bzw. \*INCREMENT) bestimmt werden, so ist die Basisversion Ausgangspunkt zur Berechnung der Zielversion.
3. Soll eine Version in einem Geltungsbereich (Scope) mit aktivem WRITE-CONTROL erzeugt werden, so ist dies nur möglich, wenn die Basisversion vom Schreiber auch ausgeliehen ist.

Die Basisversion zur Zielversion kann wie folgt bestimmt werden:

**\*STD**

Abhängig von der Angabe bei VERSION, den existierenden Elementversionen und der unter dem Typ geltenden Konvention wird eine Basisversion als Standardbasis bestimmt.

composed-name

Basisversion ist composed-name.

composed-name\*

Basisversion ist die höchste existierende Version mit Präfix composed-name.

Obwohl der Operand BASE= in den Anweisungen den Datentyp <composed-name with-wildcards> hat, ist als einziges Wildcardzeichen nur der \* am Ende der Eingabe erlaubt, um die Eingabe als Präfix zu kennzeichnen.

### Zusammenfassung

Die nachfolgende Tabelle zeigt eine Zusammenfassung zur Bestimmung von Ziel- und Basisversion zur Zielversion.

<b>BASE</b>	<b>VERSION</b>	<b>Zielversion</b>	<b>Basisversion</b>
*STD	composed-name	composed-name	Standardbasis
	*INCREMENT	*INC (Standardbasis) oder Defaultversion	Standardbasis
	*HIGHEST-EXISTING	Standardbasis oder Defaultversion	Standardbasis
composed-name1*	composed-name2	composed-name2	höchste existierende mit Präfix composed-name1
	*INCREMENT	*INC (Basisversion)	höchste existierende mit Präfix composed-name1
	*HIGHEST-EXISTING	höchste existierende mit Präfix composed-name1	höchste existierende mit Präfix composed-name1
composed-name1	composed-name2	composed-name2	composed-name1
	*INCREMENT	*INC (composed-name1)	composed-name1
	*HIGHEST-EXISTING	composed-name1	composed-name1



### 4.6.3 Versionskonventionen

Versionen von Elementen werden immer unter Beachtung einer Konvention erzeugt. Eine Konvention gilt für einen bestimmten Elementtyp innerhalb einer Bibliothek.

Unter einem Typ ist immer eine der folgenden Konventionen gültig:

NONE	keine Einschränkungen
STD-SEQUENCE	nur eine Hauptlinie
STD-TREE	Hauptlinie mit Nebenzweigen
MULTI-SEQUENCE	Jedes Präfix bildet eine Linie

Die Konvention wird in der Anweisung `MODIFY-TYPE-ATTRIBUTES` mit dem Operanden `CONVENTION` eingestellt.

Die Konvention `NONE` kann immer eingestellt werden. Die Änderung der Konvention auf einen anderen Wert ist verboten, solange noch Elemente unter dem Typ vorhanden sind.

Die LMS-Anweisung `SHOW-TYPE-ATTRIBUTES` zeigt die eingestellte Konvention an.

#### **Folgendes wird für jede Konvention dargestellt:**

##### Defaultversion

das ist die Version, die als erste Version eines Namens durch den Versionsautomatismus erzeugt wird.

##### Format

legt das für die Konvention zulässige Format der Versionsbezeichnungen fest.

##### Erhöhungsverfahren

beschreibt das für die Konvention geltende Verfahren zur Bestimmung der Zielversion bei `VERSION=*INCREMENT`.

##### Auswahl der Standardbasis

beschreibt das für die Konvention geltende Verfahren zur Bestimmung der Basisversion zur Zielversion bei `BASE=*STD`.

##### Prüfungen

beschreibt die für die Konvention geltenden Prüfungen als Voraussetzung zur Anweisungsausführung.

#### 4.6.3.1 Konvention: NONE

Defaultversion: 001

Format: beliebig

Erhöhungsverfahren

Die abschließende Zifferngruppe einer Versionsbezeichnung wird um 1 erhöht (Fehler sofern nicht möglich).

Auswahl der Standardbasis

1. Zielversion explizit gegeben (composed-name / \*UPPER-LIMIT)
  - a) Existiert die Zielversion, so ist diese auch Basisversion.
  - b) Die insgesamt höchste der existierenden Versionen ist Basisversion.
2. Zielversion implizit gegeben (\*HIGHEST-EXISTING / \*INCREMENT). Die insgesamt höchste der existierenden Versionen ist Basisversion.

Prüfungen: keine

#### 4.6.3.2 Konvention: STD-SEQUENCE

Defaultversion: durch EXAMPLE von MODIFY-TYPE-ATTRIBUTES festgelegt

Format: [<Präfix>]<Zifferngruppe>

Präfix besteht aus beliebigen zulässigen Zeichen und endet nicht mit einer Ziffer. Präfix ist für alle Versionen des Typs gleich.

Erhöhungsverfahren

Die abschließende Zifferngruppe einer Versionsbezeichnung wird um 1 erhöht (Fehler sofern nicht möglich).

Auswahl der Standardbasis

Die insgesamt höchste der existierenden Versionen ist Basisversion.

Prüfungen

Die Zielversion muss höher oder gleich der höchsten existierenden Version sein.

### 4.6.3.3 Konvention: STD-TREE

Defaultversion: 001.001.

Format: R.L[.B.S]                    R,L,B,S = nnn        n = Ziffer

R.L sind Versionen des Hauptzweiges (Release.Level).

R.L.B.S sind Versionen von Nebenzweigen (Release.Level.Branch.Sequence).

Führende Nullen in den einzelnen Zifferngruppen können bei der Eingabe entfallen, wenn

- in der Versionsbezeichnung keine Wildcard-Symbole angegeben werden, und
- der Elementtyp explizit angegeben wird.

*Beispiel:* '1.1' steht für '001.001'

#### Erhöhungsverfahren

Die abschließende Zifferngruppe einer Versionsbezeichnung wird um 1 erhöht, sofern die Basisversion Blatt eines Zweiges ist. Ansonsten wird ein neuer höherer Branch mit Sequence 1 erzeugt.

#### Auswahl der Standardbasis

1. Zielversion explizit gegeben (composed-name / \*UPPER-LIMIT).
  - a) Existiert die Zielversion, so ist diese auch Basisversion.
  - b) Existieren Versionen, die kleiner als die Zielversion sind und auf dem gleichen Zweig liegen, so ist die höchste dieser Versionen Basisversion.
  - c) Existiert die zur Zielversion gehörige Hauptzweigversion, so ist diese Basisversion.
  - d) Existieren zur Zielversion gehörige Nebenzweigversionen, so ist die höchste dieser Versionen Basisversion.
  - e) Ansonsten gibt es keine Basisversion.
2. Zielversion implizit gegeben (\*HIGHEST-EXISTING / \*INCREMENT).

Die insgesamt höchste der existierenden Versionen ist Basisversion.

#### Prüfungen

Es können nur die Daten der höchsten Version in einem Zweig geändert werden. Es können nur höhere Versionen unter Typ/Name in einem Zweig angefügt oder neue höhere Nebenzweige eingerichtet werden.

#### 4.6.3.4 Konvention: MULTI-SEQUENCE

Defaultversion: durch EXAMPLE von //MODIFY-TYPE-ATTRIBUTES festgelegt.

Format: [<Präfix>]<Zifferngruppe>

Präfix besteht aus beliebigen zulässigen Zeichen und endet nicht mit einer Ziffer. Die Länge von Präfix ist für alle Versionen des Typs gleich.

##### Erhöhungsverfahren

Die abschließende Zifferngruppe einer Versionsbezeichnung wird um 1 erhöht (Fehler, sofern nicht möglich).

##### Auswahl der Standardbasis

1. Zielversion explizit gegeben (composed-name / \*UPPER-LIMIT).
  - a) Die höchste Version mit gleichem Präfix ist Basisversion, sofern eine solche existiert.
  - b) Die insgesamt höchste der existierenden Versionen ist Basisversion.
2. Zielversion implizit gegeben (\*HIGHEST-EXISTING / \*INCREMENT).

Die insgesamt höchste der existierenden Versionen ist Basisversion.

##### Prüfungen

Jeder Präfix bildet einen Unternehmensraum. Die Zielversion muss höher sein als die höchste existierende Version im gleichen Unternehmensraum oder einen neuen Unternehmensraum bilden.

##### *Hinweise*

- Es wird sichergestellt, dass die Strukturen von Versionsbezeichnungen und die Delta-Struktur übereinstimmen.
- Wenn eine Version auf einem Hauptzweig gelöscht wird, von der ein Nebenzweig abhängig ist, kann nicht mehr der gesamte Baum unter einen konventionierten Typ kopiert werden, da für den betreffenden Nebenzweig keine geeignete Basis mehr vorhanden ist.
- composed-name muss, sofern nicht leer, mit einem Punkt „.“ enden, wenn er als BASE=composed-name\* eingegeben wird. Basis ist damit die höchste Version eines Haupt- oder Nebenzweiges.

## 4.7 Elementenschutz/Datenschutz

Dieser Abschnitt beschreibt die Funktionen, mit denen LMS den Element- und Datenschutz unterstützt. LMS kann sowohl Elemente als auch Bibliotheken mit Schutzattributen versehen.

Zur Beweissicherung kann zusätzlich ein AUDIT, d.h. ein Protokoll für die ausgewählten Ereignisse (siehe [7]) erstellt werden.

### 4.7.1 Zugriffsschutz für Elemente

Da Elemente in Bibliotheken enthalten sind, sind diese Elemente zunächst durch den Schutz dieser Bibliothek mitgeschützt. Zusätzlich kann der Schutz für die Elemente weiter vergrößert werden, indem explizit Zugriffsrechte vergeben werden.

#### Zugriffsrechte für Elemente

r : read	Leserecht für Elemente
w : write	Schreibrecht für Elemente
x : execute	Ausführungsrecht für Elemente
h: hold	Ausleihrecht für Elemente
a : administrate	Verwaltungsrecht für Elemente

Die Rechte r, w, x und h werden mit der LMS-Anweisung MODIFY-ELEMENT-PROTECTION (siehe [Seite 323](#)) für einzelne Elemente eingestellt und verändert; mit der Anweisung SHOW-ELEMENT-ATTRIBUTES (siehe [Seite 414](#)) werden sie angezeigt.

Das Verwaltungsrecht wird mit der LMS-Anweisung MODIFY-LIBRARY-ATTRIBUTES (siehe [Seite 333](#)) für alle Elemente der Bibliothek und mit MODIFY-TYPE-ATTRIBUTES (siehe [Seite 365](#)) für alle Elemente dieses Typs eingestellt. Mit SHOW-LIBRARY-ATTRIBUTES (siehe [Seite 434](#)) bzw. SHOW-TYPE-ATTRIBUTES (siehe [Seite 448](#)) wird es angezeigt. Durch das Verwaltungsrecht wird festgelegt, wer Elemente innerhalb einer Bibliothek bzw. eines Typs einrichten, löschen und umbenennen darf.

Die Voreinstellung für das Verwaltungsrecht ist NONE. Dadurch sind Verwaltungsfunktionen dann erlaubt, wenn sie nicht durch den Schutz der Bibliotheksdatei verhindert werden.

Das Ändern von Schutzattributen ist nur dem Eigentümer der Bibliothek vorbehalten.

Für jedes Recht kann der Schutz durch einen der folgenden Mechanismen eingestellt sein:

NONE: kein spezieller Schutz

STD: Standardschutz durch BACL [ + Kennwort ]

BY-GUARD: Schutz durch GUARDS

Es wird explizit darauf hingewiesen, dass der Schutz durch Kennwörter an den BACL-Schutz gekoppelt ist und somit hier beim GUARD-Schutz keine Rolle spielt.

Zu einem Zeitpunkt kann nur ein Mechanismus aktiv sein, für jedes Recht jedoch ein anderer. Beim Wechsel des Mechanismus gehen die Werte des zuletzt aktiven Mechanismus verloren.

Kann aufgrund des aktiven Mechanismus und der aktuellen Systemumgebung die Zugriffsberechtigung nicht entschieden werden, so wird kein Zugriff zugelassen.

Gründe hierfür können sein:

Schutz durch GUARDS, aber Subsystem GUARDS nicht installiert

Abhilfe ist möglich durch:

Ändern der Systemumgebung

Ändern des aktiven Schutzmechanismus (nur durch Bibliothekseigentümer möglich)

Die verschiedenen Mechanismen realisieren Schutz wie folgt:

1. NONE - kein spezieller Schutz

- keine Zugriffsprüfung

2. STD - Standardschutz durch BACL

Jedes der oben aufgeführten Zugriffsrechte wird über Schutzbits und eventuell über Kennwörter an die folgenden Benutzerkreise vergeben:

OWNER: Eigentümer der Bibliotheksdatei

GROUP: Gruppe des Eigentümers der Bibliotheksdatei

OTHERS: alle anderen

Durch Kennwörter kann der Kreis der ausgewählten Benutzer weiter eingeschränkt werden. Kennwörter werden verschlüsselt in der Bibliothek abgelegt. Ein Zugriff auf ein durch ein Kennwort geschütztes Element ist nur erlaubt, wenn das Kennwort in der Kennworttabelle eingetragen ist (durch das BS2000-Kommando ADD-PASSWORD).

Sowohl die Benutzerkreise als auch die Kennwörter werden in der LMS-Anweisung MODIFY-ELEMENT-PROTECTION durch die Operanden USER und PASSWORD festgelegt.

### 3. BY-GUARD - Schutz durch GUARDS

Ein Zugriff wird anhand einer Schutzbeschreibung entschieden, die in einem GUARD abgelegt ist.

Ein GUARD ist ein eigenständiges Objekt des BS2000, das durch BS2000-Kommandos eingerichtet, geändert und gelöscht werden kann. Der GUARD-Eigentümer legt einerseits die Schutzbeschreibung, andererseits den möglichen Nutzerkreis des GUARD fest:

```
SCOPE = *USERID / *USER-GROUP / *HOST-SYSTEM
```

(siehe [6])

Die Schutzbeschreibung enthält Bedingungen, die erfüllt sein müssen, bevor ein Zugriff erlaubt werden kann.

Bedingungen können sein:

- Datum
- Uhrzeit
- Wochentag
- Privileg der Task
- geladenes Programm

Ein Zugriff wird nur erlaubt, wenn der Bibliothekseigentümer den GUARD (noch) nutzen darf und die im GUARDS angegebenen und zu berücksichtigenden Bedingungen erfüllt sind.

Der GUARD, der zur Zugriffsüberprüfung herangezogen werden soll, wird durch den Bibliothekseigentümer festgelegt.

Es sind nur GUARD-Namen zulässig, die vom Subsystem GUARDS akzeptiert wurden. Es wird nicht überprüft, ob der angegebene GUARD überhaupt existiert. Zulässige GUARD-Namen werden genauso in der Bibliothek abgelegt, wie sie angegeben wurden.

GUARD-Namen beziehen sich immer auf die CATID, unter der die Bibliothek katalogisiert ist in der der GUARD-Name abgelegt ist.

GUARD-Namen ohne USERID beziehen sich auf die USERID, unter der die Bibliothek katalogisiert ist in der der GUARD-Name abgelegt ist.

GUARD-Namen werden jedem angezeigt, der auch das Inhaltsverzeichnis der Bibliothek lesen darf.

Beim GUARDS-Schutz von Bibliotheken und Elementen sind Besonderheiten zu beachten. So hat der Schutz einzelner Elemente einer Bibliothek nur Sinn, wenn der Behälter (die Bibliothek) auch geschützt, d.h. im „PROTECTED MODE“ ist.

*Beispiel*

Unter der Kennung USER1 steht die Bibliothek BIBL, die das Element ELEM1 enthält. USER2 möchte zugreifen. Unter der Kennung USER1 ist das GUARD GUARD1, das USER1 und USER2 den Zugriff erlaubt. Folgende Fälle werden betrachtet:

1. Nur die Bibliothek BIBL ist mit dem GUARD GUARD1 lese-geschützt

Dieser Fall wird nicht (!) so behandelt, als wäre BIBL eine Datei. Es wird erkannt, dass es sich um eine Bibliothek handelt und der Zugriffsschutz ein anderer ist, als wenn BIBL eine normale Datei wäre. Die Bibliothek ist im PROTECTED MODE und damit ist nur ein Zugriff über LMS oder über das Kommando COPY-FILE möglich. Ein weiterer lesender Zugriffsversuch von USER1 oder USER2 (z.B. SHOW-FILE BIBL) wird abgewiesen. Man kann aber auf einzelne Elemente zugreifen, z.B. mit SHOW-FILE \*L(BIBL,ELEM1,S).

Das Kopieren ist nicht mehr möglich, wenn mindestens ein Element aus BIBL mit GUARDS geschützt ist (siehe Punkt 3).

2. Nur die Elemente sind mit dem GUARD GUARD1 geschützt

Diese Schutzmethode ist nicht sinnvoll, weil die Bibliothek dann USER-ACC=\*ALL-USERS haben muss, damit USER2 die Bibliothek sehen kann. In diesem Fall kann er die gesamte Bibliothek kopieren und sich so Zugriff auf alle Elemente verschaffen, ganz gleich welcher Schutz für die Elemente angegeben ist.

3. Die Bibliothek und die einzelnen Elemente sind durch GUARDS geschützt

Damit kann man einzelne Elemente einer Bibliothek schützen. Es sollten alle Benutzer, die auf Elemente zugreifen wollen, entsprechende Rechte auf die Bibliothek erhalten. Das bedeutet in dem Fall aber nur, dass sie den Katalogeintrag sehen können und mit LMS darauf zugreifen können.

Auch dann ist es sinnvoll, einzelne Elemente mit GUARDS zu schützen. Die tatsächlichen Zugriffsrechte ergeben sich aus der Schnittmenge der Zugriffsrechte, d.h. wenn ein Benutzer kein Schreibrecht auf die gesamte Bibliothek hat, kann er kein Element schreiben, auch wenn ihm das durch ein entsprechendes GUARD auf Elementebene erlaubt ist.

Wird auf den Schutz einzelner Elemente verzichtet, und schützt man seine Bibliotheken mit GUARDS, dann kann man mit ihnen fast so arbeiten wie mit Dateien.



**Initialer Elementschutz**

Der Eigentümer der Bibliotheksdatei kann für die Bibliothek und/oder einen bestimmten Typ einen initialen Elementschutz festlegen.

Ist ein initialer Elementschutz festgelegt, so wird dieser Schutz bei neuen Elementen eingetragen. Ist kein initialer Elementschutz festgelegt, so werden Elemente ohne zusätzlichen Schutz angelegt. Sie sind nur durch den Schutz der Bibliotheksdatei geschützt.

Ist sowohl für die Bibliothek als auch für den betroffenen Typ ein initialer Elementschutz vereinbart, so wird nur die Vereinbarung für den Typ berücksichtigt. Wird der initiale Elementschutz verändert, so wirkt sich dies nur auf den Schutz künftig neu eingerichteter Elemente aus. Der Schutz bereits existierender Elemente bleibt unverändert.

## Übersicht über die Schutzattribute

<b>Bibliotheksebene:</b>			
a:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
Initial-Element-Protection			
r:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
w:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
x:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
h:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
<b>Typeebene:</b>			
a:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
Initial-Element-Protection			
r:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
w:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
x:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
h:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
<b>Elementebene:</b>			
Element-Protection			
r:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
w:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
x:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD
h:	NONE	/ BACL [+ Kennwort]	/ BY-GUARD

## Übersicht über notwendige Rechte für LMS-Aktionen

Die nachfolgende Übersicht zeigt, welche Rechte für welche LMS-Aktion unter welchen Bedingungen erforderlich sind. Es wird davon ausgegangen, dass die Bibliotheksdatei in geeigneter Weise eröffnet werden kann.

Die in der Übersicht verwendeten Abkürzungen bedeuten:

- Aktion ist nicht zulässig bzw. nicht möglich
- \* Aktion ist von jedem ausführbar
- , Und-Verknüpfung
- / Oder-Verknüpfung
- EATTR = (CCSN, USER-DATE/TIME)
- STATE Der durch MODIFY-ELEMENT-ATTRIBUTES eingestellte Elementzustand.
- WRITE-CONTROL Der im Elementscope wirksame Wert von WRITE-CONTROL (siehe MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES)
- a: Der Rufer muss Administrationsrecht für den Elementscope haben.  
Der Rufer erhält Administrationsrecht auf eine der folgenden Arten:
  1. Für den betroffenen Elementtyp ist Administrationsrecht vergeben und der Rufer gehört zum Kreis der Berechtigten.
  2. Administrationsrecht ist nicht für den Elementtyp, aber für die Bibliothek vergeben, und der Rufer gehört zum Kreis der Berechtigten.
  3. Weder für den Elementtyp noch für die Bibliothek ist Administrationsrecht vergeben.
- r: Der Rufer muss Leserecht für das Element haben.
- w: Der Rufer muss Schreibrecht für das Element haben.
- x: Der Rufer muss Ausführungsrecht für das Element haben.
- h: Der Rufer muss Ausleihrecht für das Element haben.
- E: Der Rufer muss Bibliothekseigentümer sein.
- H: Der Rufer muss als HOLDER des Elementes eingetragen sein.
- B: Explizit oder implizit gegebene Basisversion.

LMS-Aktion	notwendige Bedingungen			
	DEACTIVATED		ACTIVATED	
WRITE-CONTROL=				
LIBRARY-ATTR ändern	E		E	
TYPE-ATTR ändern	E		E	
1. Version erzeugen	a		a	
n. Version erzeugen	a		H(B) <sup>1</sup>	
STATE=	FREE	IN-HOLD	FREE	IN-HOLD
ELEMENT-ATTR anzeigen	*	*	*	*
Version löschen	a,w	-	a,w	-
Version umbenennen <sup>2</sup>	a,w	-	-	-
Version überschreiben	w	w,H	-	w,H
EATTR ändern	a/w	-	a/w	-
STATE=IN-HOLD setzen	h	-	h	-
STATE=FREE setzen	h	H/E	h	H/E
Version lesen	r	r	r	r
Version ausführen	x	x	x	x
ELEMENT-PROT ändern	E	E	E	E

<sup>1</sup> Die neue Version wird mit den Attributen der Basis, also insbesondere STATE=\*IN-HOLD erzeugt.

<sup>2</sup> Soll ein Element auf ein anderes existierendes Element umbenannt werden, so muss für dieses das Recht zum Überschreiben gegeben sein.

*Hinweise*

- In Systemen ohne Gruppenstruktur kann über den Benutzerkreis Gruppe kein Zugriff erfolgen. Durch die Benutzerkreise Eigentümer und alle anderen ist jedoch eine zu USER-ACCESS analoge Einstellung möglich.
- Ein Elementschutz ist nur dann sinnvoll, wenn die Bibliotheksdatei gegen UPAM-Zugriffe geschützt ist. Bibliotheksdateien sind vor UPAM-Zugriffen geschützt, wenn sie durch eine Basic Access Control List (BACL) oder durch GUARDS geschützt sind und ordnungsgemäß eingerichtet sind.

Für Bibliotheken, die durch einen Filetransfer (openFT) überspielt wurden, gilt, dass sie erst dann ordnungsgemäß installiert sind, wenn auf sie durch PLAM schreibend zugegriffen wurde.

- Für Bibliotheken, die gegen UPAM-Zugriffe geschützt sind, gelten folgende Einschränkungen:
  - sie sind nicht mehr durch ADD-ELEMENT in eine Bibliothek aufnehmbar,
  - sie sind nicht mehr über Remote File Access( RFA) bearbeitbar.

Für solche Anwendungen muss für die Bibliotheksdatei der BACL-Schutz deaktiviert werden. Die Bibliotheksdatei ist dann nicht mehr vor UPAM-Zugriffen geschützt. Sie kann jetzt wie jede andere PAM-Datei bearbeitet werden.

- Bibliotheken, die Schutzattribute enthalten und die gegen UPAM-Zugriffe geschützt sind, sind nur vom Bibliothekseigentümer mittels COPY-FILE kopierbar.
- Schutz durch GUARD setzt voraus, dass das Subsystem GUARDS verfügbar ist.

## 4.7.2 Datenschutz durch Überschreiben

Datenschutz durch Überschreiben bedeutet, dass der Anwender gezielt nicht mehr benötigte Dateien durch Überschreiben löscht. Die Daten werden dadurch physikalisch gelöscht, d.h. mit X'00' überschrieben. Das Überschreiben der Daten wird lokal gesteuert durch die LMS-Anweisung DELETE-ELEMENT bzw. global gesteuert in MODIFY-LMS-DEFAULTS, jeweils durch den Operanden DESTROY-DATA.

Der Operand DESTROY-DATA ist einerseits ein Elementattribut, d.h. das Überschreiben wirkt automatisch auf dieses Element und andererseits ein Verarbeitungsparameter der LMS-Anweisung DELETE-ELEMENT. Als Verarbeitungsparameter bewirkt DESTROY-DATA, dass alle Elemente, die durch die Anweisung erfasst sind, beim Löschen überschrieben werden.

Die Daten werden mit X'00' überschrieben, wenn eine der folgenden Angaben das Überschreiben verlangt:

- CLASS2-Option DESTLEV
- Angabe beim Element: Wert von DESTROY-DATA beim letzten Anlegen bzw. beim letzten schreibenden Zugriff auf das Element
- Angabe über den Operanden DESTROY-DATA

### 4.7.3 Beweissicherung

Die von LMS verwendete Zugriffsmethode PLAM besitzt einen Anschluss an das Subsystem SAT (security audit trail) im Sicherheitspaket SECOS (siehe [7]). Bei aktivem SAT können vom Sicherheitsbeauftragten folgende Ereignisse zur Protokollierung ausgewählt werden:

- CREATE-ELEMENT
- MODIFY-ELEMENT
- READ-ELEMENT
- EXECUTE-ELEMENT
- CLOSE-ELEMENT
- DELETE-ELEMENT
- RENAME-ELEMENT
- CREATE-SECURITY-ATTRIBUTES
- MODIFY-SECURITY-ATTRIBUTES
- DELETE-SECURITY-ATTRIBUTES

---

## 5 Funktionen von LMS

Dieses Kapitel gibt einen Überblick über die Funktionen von LMS.

### 5.1 Starten/Beenden von LMS

#### 5.1.1 Starten von LMS

LMS wird durch das Kommando `/START-LMS` bzw. `/LMS` als autonomes Programm aufgerufen.

Anwendungsbereich: **UTILITIES**

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

#### **VERSION =**

Angabe der gewünschten Produktversion.

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

Die durch das Kommando `SELECT-PRODUCT-VERSION` definierte Version wird ausgewählt. Wenn es keine definierte Version gibt, wird vom System die höchstmögliche Version ausgewählt.

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

Angabe der ausgewählten Version.

#### **MONJV = \*NONE / <filename 1..54 without-gen-vers>**

Name der Jobvariablen, die den LMS-Lauf überwachen soll. (Steht nur Benutzern des Software-Produkts `JV` zur Verfügung (siehe [\[8\]](#))).

Während des Ablaufs von LMS setzt das System die Jobvariablen auf folgende Werte:

Wert	Bedeutung/Anlass der Wertzuweisung
\$R	LMS läuft.
\$T	LMS wurde normal beendet.
\$A	LMS wurde abnormal beendet.

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

Es wird keine Jobvariable vereinbart.

### **CPU-LIMIT = \*JOB-REST / <integer 1..32767>**

Maximale CPU-Zeit in Sekunden, die LMS beim Ablauf verbrauchen darf. Wird diese Zeit überschritten, wird der Benutzer im Dialogbetrieb vom System benachrichtigt; im Stapelbetrieb wird der LMS-Lauf beendet.

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

Wurde der Dialogauftrag mit Zeitbeschränkung gestartet, wird für den LMS-Lauf der bei der Systemgenerierung festgelegte Wert als Zeitbeschränkung verwendet. Andernfalls läuft das Programm ohne Zeitbeschränkung.

### **Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	2	LMS0238	Fehler beim Nachladen von LMS
	32	LMS1002	interner Fehler
	130	LMS0041	Systemadressraum erschöpft

### *Hinweis*

Ist eine Startdatei angegeben, so kann der Kommando-Returncode durch die Returncodes der in der Startdatei ausgeführten Anweisungen überschrieben werden.



## 5.1.2 Überwachung des LMS-Laufs mit Jobvariablen

Wenn beim Aufruf von LMS mit `/START-LMS MONJV=name` eine Monitor-Jobvariable angegeben wird, so wird diese beim Beenden von LMS gefüllt. Die MONJV wird unabhängig vom Wert des Operanden PROPAGATE-STMT-RC des SDF-P-Kommandos `/BEGIN-BLOCK` versorgt.

Der MONJV-Wert gliedert sich in eine drei Byte lange Zustandsanzeige und in eine vier Byte lange Rückkehrcode-Anzeige. Die Rückkehrcode-Anzeige gliedert sich in einen ein Byte langen Beendigungscode (BC) und eine drei Byte lange Programminformation (PI). LMS setzt die Zustandsanzeige und den Beendigungscode wie folgt:

### Zustandsanzeige

Zustands- anzeige			Rückkehrcode-Anzeige				Bemerkungen
			Beendigungscode BC		Programm- information PI		
1	2	3	4	5	6	7	Byte
\$T			0	siehe unten			normale Beendigung
			1				
\$A			2				abnormale Beendigung
			3				

**Beendigungsanzeige**

<b>BC</b>	<b>Erläuterung</b>
0	normale Beendigung LMS lief fehlerfrei
1	normale Beendigung Es traten Warnungen oder Fehler auf, die schwächer waren als der mit MAX-ERROR-WEIGHT eingestellte Wert (siehe MODIFY-LMS-DEFAULTS).
2	abnormale Beendigung Es wurde ein durch MAX-ERROR-WEIGHT gesetztes Abbruchkriterium erreicht (siehe MODIFY-LMS-DEFAULTS).
3	abnormale Beendigung Der aufgetretene Fehler ist so schwerwiegend, dass eine Fortsetzung des LMS-Laufs nicht möglich oder nicht mehr sinnvoll ist. Der LMS-Lauf wurde intern beendet.

**Programminformation**

<b>PI</b>	<b>Erläuterungen</b>
000	BC: 0 LMS lief fehlerfrei
001	BC: 1 Es traten höchstens Warnungen auf.
002	BC: 1 oder 2 Es traten höchstens Fehler der Klasse RECOVERABLE auf, d.h. ein Element konnte nicht gefunden oder überschrieben werden.
003	BC: 1 oder 2 Es traten höchstens Fehler der Klasse SIGNIFICANT auf, d.h. nicht schwerwiegende Fehler.
004	BC: 2 Es traten höchstens Fehler der Klasse SERIOUS auf, d.h. schwerwiegende Fehler.
005	BC: 3 Der aufgetretene Fehler ist so schwerwiegend, dass eine Fortsetzung des LMS-Laufs nicht möglich oder nicht mehr sinnvoll ist. Der LMS-Lauf wurde intern beendet.

*Hinweis*

Bei Verwendung der END-Anweisung im Zusammenhang mit der Nutzung der LMS-Funktionalität aus dem EDT sind die Ausführungen auf [Seite 126](#) zu beachten.

### 5.1.3 Startdatei

Der Anwender hat die Möglichkeit, eine SAM oder ISAM Datei einzurichten, die beim Starten von LMS automatisch abgearbeitet wird. Sie kann beliebige LMS Anweisungen enthalten. Die LMS Anweisungen, die beim Starten von LMS automatisch abgearbeitet werden sollen, können auch in ein Bibliothekselement abgelegt werden.

Voraussetzung ist, dass beim Starten von LMS SYSDTA den Wert PRIMARY oder SYSCMD hat. Andernfalls wird die Startanweisungsfolge nicht ausgeführt. SYSDTA wird nach Bearbeitung der Startdatei wieder auf seinen ursprünglichen Wert gesetzt.

Die Startdatei wird nach folgender Hierarchie gesucht:

1. über die S-Variable SYSLMSPAR
2. über den Linknamen \$LMSPAR
3. über den Dateinamen SYSPAR.LMS
4. über die globale Startdatei

Es ist für die Systemadministration möglich, den Installationsort der globalen Startdatei frei zu wählen. Für diese globale Startdatei wird in der SYSSII-Datei die logische Identifikation SYSPAR definiert.

Der Installationsdateiname wird durch das Kommando SET-INSTALLATION-PATH vom System-Administrator dem Installations-Monitor bekanntgegeben. Ist kein Installationsdateiname für die globale Startdatei definiert, wird über den Dateinamen \$.SYSPAR.LMS nach einer globalen Startdatei gesucht.

Die Namen \$LMSPAR, SYSLMSPAR, SYSPAR.LMS und die globale Startdatei dürfen vom Anwender nur im Zusammenhang mit der Startdatei verwendet werden. Anderweitige Verwendungen dieser Namen können beim Starten von LMS auf Fehler führen, z.B. wenn SYSPAR.LMS eine PLAM-Bibliothek ist. Nach Ausgabe einer Fehlermeldung wird der LMS-Lauf fortgesetzt.

Der Inhalt der S-Variablen SYSLMSPAR muss ein gültiger Wert des Operanden TO-FILE= des Kommandos ASSIGN-SYSDTA sein (siehe ASSIGN-SYSDTA Kommando in [4]). Der Inhalt kann also ein Dateiname oder ein Bibliothekselement der Form \*LIB-ELEM(LIB=...,ELEM=...,TYPE=...) sein.

Die S-Variable muss mit SCOPE=\*TASK deklariert werden, wenn sie in einer Prozedur (z.B. LOGON-Prozedur) gesetzt wird und trotzdem außerhalb der Prozedur für jeden LMS-Lauf zugreifbar sein soll. Ansonsten braucht sie nicht deklariert zu werden.

Die Startdatei kann mit /ADD-FILE-LINK FILE-NAME=\*DUMMY,LINK-NAME=\$LMSPAR ausgeschaltet werden, wenn die Startanweisungsfolge nicht über die Variable SYSLMSPAR referenziert wird. Existiert die Variable, so muss sie den String \*DUMMY enthalten, wenn die Ausführung einer Startanweisungsfolge verhindert werden soll.

*Beispiele*

- Es sollen die Werte LOGGING=\*MAXIMUM und TYPE=S in der Datei SYSPAR.LMS voreingestellt werden.

Inhalt der Startdatei:

```
//MODIFY-LOGGING-PARAMETERS LOGGING=*MAXIMUM  
//MODIFY-LMS-DEFAULTS TYPE=S
```

Nach dem Aufruf /START-LMS sind die obigen Werte automatisch eingestellt.

- Diesmal sollen die Startanweisungen im Element START unter Typ S in der Bibliothek X sein.

Inhalt der S-Variablen SYSLMSPAR:

```
/SYSLMSPAR='*LIB-ELEM(LIB=X,ELEM=START,TYPE=S)'
```

Inhalt des Elementes START:

```
//MODIFY-LOGGING-PARAMETERS LOGGING=*MAXIMUM  
//MODIFY-LMS-DEFAULTS TYPE=S
```

Nach dem Aufruf /START-LMS sind die obigen Werte automatisch eingestellt.

### 5.1.4 Voreinstellungen nach dem Starten von LMS

Nach dem Aufrufen von LMS sind Default-Werte gültig. Werden für bestimmte Anweisungen andere Werte als die voreingestellten benötigt, können sie durch die LMS-Anweisung MODIFY-LMS-DEFAULTS oder lokal geändert werden.

Die folgende Tabelle stellt dar, auf welche Anweisung die Default-Werte wirken bzw. ob sie den gesamten LMS-Lauf beeinflussen.

LMS-Default	Default-Wert	Wirkung auf	Bedeutung der Defaults
ELEMENT-ATTRIBUTES TYPE ELEMENT-VERSION TO-ELEMENT-VERSION STORAGE-FORM SOURCE-ATTRIBUTES	*NONE *ALL *BY-SOURCE *STD *STD	Anweisungen mit Typangabe SHOW-ELEMENT-ATTR Zielversion Zielspeicherform ADD-ELEMENT	Festlegen des Elementtyps Festlegung der Quellversion Festlegen der Zielversion Festlegen der Zielspeicherform Behandlung der Dateiattribute
FILE-ATTRIBUTES	*ISAM	EXTRACT-ELEMENT	Attribute für die Zieldatei
DESTROY-DATA	*NO	DELETE-ELEMENT <sup>1</sup>	Überschreiben von Daten
WRITE-MODE	*CREATE	ADD-ELEMENT COPY-ELEMENT EDIT-ELEMENT EXTRACT-ELEMENT MODIFY-ELEMENT MODIFY-ELEMENT-ATTR PROVIDE-ELEMENT RETURN-ELEMENT	Überschreiben der Elemente
DIALOG-CONTROL	*NO	ADD-ELEMENT COPY-ELEMENT DELETE-ELEMENT EDIT-ELEMENT EXTRACT-ELEMENT MODIFY-ELEMENT MODIFY-ELEMENT-ATTR MODIFY-ELEMENT-PROT PROVIDE-ELEMENT RETURN-ELEMENT	dialoggeführter Fragemechanismus, der jedes Element einzeln anbietet.
INFORMATION LAYOUT SORT	*MEDIUM *VARIABLE *BY-NAME	SHOW-ELEMENT-ATTR SHOW-ELEMENT-ATTR SHOW-ELEMENT-ATTR	Informationsmenge Darstellung Sortierreihenfolge
OUTPUT-FORM	*STD	SHOW-ELEMENT	Ausgabeform
DELETE-SOURCE	*NO	ADD-ELEMENT	Quelldatei löschen
PROTECTION	*STD	ADD-ELEMENT COPY-ELEMENT EXTRACT-ELEMENT	Schutzattribute aufnehmen Schutzattribute kopieren Schutzattribute übergeben

LMS-Default	Default-Wert	Wirkung auf	Bedeutung der Defaults
MAX-ERROR-WEIGHT	*SERIOUS	alle Anweisungen	Spinoff-Steuerung
NEXT-ATTEMPT	*NO	alle Anweisungen mit *LIB-ELEM Angabe	steuert die Öffnungsversuche bei Datei-, Typ- oder Element- sperre
COMPARE- PARAMETERS RECORD-PART SPACES INFORMATION LAYOUT JOIN-ELEMENT-SETS	*PARAMETERS  *ALL *IGNORED *MEDIUM *COMPATIBLE *NO	COMPARE-ELEMENT	legt die Vergleichsparameter fest
TEXT-INFORMATION MODULE-INFORMATION PHASE-INFORMATION LLM-INFORMATION	*ALL *ALL *ALL *ALL	SHOW-ELEMENT SHOW-ELEMENT SHOW-ELEMENT SHOW-ELEMENT	Info. für textartige Elemente Modulinformation (Typ R) Phaseninformation (Typ C) LLM-Information (Typ L)

<sup>1</sup> Wenn der Operand DESTROY-DATA mit MODIFY-LMS-DEFAULTS verändert wird, wirkt er indirekt auch auf die Anweisungen ADD-, COPY-, EDIT- und EXTRACT-ELEMENT.

Sollen außer Fehlermeldungen auch Erfolgsmeldungen protokolliert werden, muss in der Anweisung MODIFY-LOGGING-PARAMETERS der Operand LOGGING=\*MAXIMUM gesetzt werden.

### 5.1.5 Beenden des LMS-Laufs

Um LMS zu beenden, muss die Anweisung END angegeben werden.

END

Die END-Anweisung besitzt keine Operanden (Genaueres zur END-Anweisung siehe [Seite 264](#)).

## 5.2 Zuweisen von Bibliotheken

Um Bibliotheken zu bearbeiten, müssen sie zuerst zugewiesen werden. Erst nach dem erfolgreichen Zuweisen können Elemente aufgenommen und/oder bearbeitet werden.

In LMS-Anweisungen werden Bibliotheken durch den Operanden LIBRARY angegeben. Es gibt mehrere Möglichkeiten eine Bibliothek zuzuordnen, die letztlich direkt oder indirekt auf PLAM-Bibliotheken führen:

1. Global vereinbarte Bibliothek (\*STD)
2. Direkter Name einer PLAM-Bibliothek
3. Indirekt über Linkname
4. Bibliotheksliste (Variable "SYSPLAMALT-")
5. Typumlenkung (Variable "SYSPLAMLIB.")

Die global vereinbarte Bibliothek wird durch OPEN-LIBRARY zugewiesen und durch CLOSE-LIBRARY auf undefiniert zurückgesetzt. Die Möglichkeiten 2-5 können zur Vereinbarung der globalen Bibliothek genutzt werden.

Bibliothekslisten ermöglichen die Verwendung von alternate libraries bei Bibliotheksangaben. Eine Bibliotheksliste ist eine Liste, die anzeigt, welche Bibliotheken in welcher Reihenfolge nach einem Element durchsucht werden sollen. Auf Bibliothekslisten kann nur lesend zugegriffen werden. Eine Variable "SYSPLAMALT-" definiert eine Bibliotheksliste (ALT für ALternate library).

Mittels Typumlenkung kann ein Standardtyp auf einen Benutzertyp umgelenkt werden. Damit können existierende Programme, die keine Benutzertypen erlauben, auch Benutzertypen nutzen. Eine Variable "SYSPLAMLIB." definiert eine Typumlenkung. Die Typumlenkung ist ein zentraler Mechanismus, der in allen Kommandos, unabhängig von LMS, verwendet werden kann.

Die LMS-Anweisung SHOW-LIBRARY-STATUS gibt Auskunft über den Zustand der zu bearbeiteten Bibliotheken.

### Direkter Name einer PLAM-Bibliothek

Der angegebene Dateiname bezeichnet eine PLAM-Bibliothek. Diese Angabe wird beim Arbeiten mit LMS häufig benutzt. Normalerweise bezeichnet der angegebene Dateiname auch wirklich eine Datei. Es gibt aber folgende Ausnahmen:

- der Dateiname beginnt mit "SYSPLAMALT-" und eine gleichnamige S-Variable existiert: es liegt eine Bibliotheksliste vor (siehe [Seite 82](#))
- der Dateiname beginnt mit "SYSPLAMLIB." und eine gleichnamige S-Strukturvariable existiert: es liegt eine Typumlenkung vor (siehe [Seite 85](#))

### Indirekt über Linkname

Der unter \*LINK angegebene Name <link> bezeichnet einen Linknamen. Im Allgemeinen ist der Name ein durch /ADD-FILE-LINK vereinbarter Dateikettungsname, dem eine PLAM-Bibliothek zugeordnet ist. Hiervon gibt es folgende Ausnahme:

eine S-Strukturvariable SYSPLAMLIB.<link> existiert: es liegt eine Typumsteuerung vor (siehe dort)

#### Beispiele

##### 1. Zuweisen einer globalen Bibliothek durch OPEN-LIBRARY

Eine Bibliothek wird als **globale** Bibliothek eröffnet, wenn sie durch die LMS-Anweisung OPEN-LIBRARY zugewiesen wird.

Es ist nur eine globale Bibliothek pro LMS-Lauf möglich. Wird durch eine zweite OPEN-LIBRARY-Anweisung eine neue globale Bibliothek eröffnet, wird die erste globale Bibliothek durch ein LMS-internes CLOSE geschlossen und die neue Bibliothek als globale Bibliothek betrachtet.

Eine globale Bibliothek kann bereits bestehen oder neu eingerichtet werden. Wird eine Bibliothek neu eingerichtet, muss sie mit MODE=\*UPDATE erzeugt werden.

Eine globale Bibliothek wird standardmäßig nur zum Lesen eröffnet. Soll sie zum Lesen und Schreiben eröffnet werden, muss der Operand MODE=\*UPDATE in der OPEN-LIBRARY-Anweisung gesetzt werden.

In den LMS-Anweisungen wird eine globale Bibliothek mit LIBRARY=\*STD angesprochen. Dieser Wert ist dort standardmäßig vorgegeben, so dass keine explizite Bibliotheksangabe mehr nötig ist.

```
/START-LMS
//WRITE-COMMENT 'Eröffnen einer globalen Bibliothek zum Lesen und Schreiben'
//OPEN-LIBRARY LIBRARY = global-lib, MODE = *UPDATE
.
.
//WRITE-COMMENT 'Inhaltsverzeichnis der globalen Bibliothek anzeigen'
//SHOW-ELEMENT-ATTRIBUTES
.
.
```



## 2. Zuweisen einer globalen Bibliothek über ihren Dateikettungsnamen

Eine globale Bibliothek kann auch mit ihrem Dateikettungsnamen zugewiesen werden. Bei der Verwendung des Dateikettungsnamens muss jedoch vor dem Aufruf von LMS ein /ADD-FILE-LINK-Kommando gegeben werden, das die Verbindung zum Dateinamen der Bibliothek herstellt.

```
/ADD-FILE-LINK FILE-NAME = global-lib, LINK-NAME = testlib
/START-LMS
//WRITE-COMMENT 'Eröffnen einer globalen Bibliothek über Dateikettungsnamen'
//OPEN-LIBRARY LIBRARY = *LINK(LINK-NAME = testlib)
.
.
```

## 3. Zuweisen einer lokalen Bibliothek in einer Anweisung

Durch die Angabe einer Bibliothek in der Datenstruktur \*LIBRARY-ELEMENT werden mit dem Operanden LIBRARY Bibliotheken anweisungslokal vereinbart. Sie gelten nur für diese Anweisung.

```
/START-LMS
//WRITE-COMMENT 'Beispiel für das Zuweisen einer lokalen Bibliothek '
//WRITE-COMMENT ' in der Anweisung EDIT-ELEMENT '
.
.
//EDIT-ELEMENT *LIBRARY-ELEMENT(LIBRARY = local-lib, ELEMENT = test)
.
.
```

#### 4. Zuweisen einer lokalen Bibliothek über ihren Dateikettungsnamen

Eine lokale Bibliothek kann auch mit ihrem Dateikettungsnamen zugewiesen werden. Bei der Verwendung des Dateikettungsnamens muss jedoch vor dem Aufruf von LMS ein /ADD-FILE-LINK-Kommando gegeben werden, das die Verbindung zum Dateinamen der Bibliothek herstellt.

```

/ADD-FILE-LINK FILE-NAME = lokal-lib, LINK-NAME = testlib
/START-LMS
/WRITE-COMMENT 'Eröffnen einer lokalen Bibliothek über Dateikettungsnamen'
/WRITE-COMMENT ' in der Anweisung EDIT-ELEMENT '
.
//EDIT-ELEMENT (LIBRARY = *LINK(LINK-NAME = testlib), ELEMENT = test)
.
.

```

#### Zuweisen einer Bibliothek aus einer Liste

Bibliothekslisten ermöglichen die Verwendung von alternate libraries bei Bibliotheksangaben. Die Reihenfolge der alternate libraries in der Bibliotheksliste bestimmt, in welcher Reihenfolge Bibliotheken nach Elementen durchsucht werden.

Wenn ein Element in einer Bibliotheksliste gesucht wird, arbeitet PLAM die Liste ab, um zunächst ein Element mit dem gewünschten Typ und Namen zu finden. Die Bibliotheken in der Bibliotheksliste werden dabei der Reihe nach durchsucht, bis der angegebene Elementname mit dem zugehörigen Elementtyp gefunden wird. Das heißt, gibt es mehrere Bibliotheken in der Liste, die Elemente mit gleichem Typ und Namen enthalten, wird das Element stets in der ersten davon gefunden; diese Bibliothek ist die Trefferbibliothek.

Die Versionsangabe zum Element bestimmt, welche Elementversion aus der Trefferbibliothek ausgewählt wird. Damit überdeckt eine Elementversion in einer Bibliothek alle Elementversionen in später genannten Bibliotheken, wodurch der Auswahlvorgang leicht nachvollziehbar ist (siehe Beispiel [Seite 83](#))

Auf Bibliothekslisten kann nur lesend zugegriffen werden. Eine Bibliotheksliste wird über eine S-Variable definiert. Die Variable hat den Typ *String* und der Name beginnt mit 'SYSPLAMALT-' (ALT für ALternate library). Der Variableninhalt wird wie folgt gesetzt:

```
SYSPLAMALT-<name> = '(<lib>,<lib>,...)'
```

<name>: Rest des Variablenbezeichners

<lib>: Bibliotheksangabe

Wird als Name der Bibliotheksliste ein Dateiname "Catid:\$Userid.SYSPLAMALT-" angegeben, werden Catid und Userid ignoriert. Namen, die mit 'SYS' beginnen, sind für zukünftige Entwicklung reserviert. Kleinbuchstaben im Inhalt der Variablen "SYSPLAMALT-" werden als Großbuchstaben interpretiert.

Bibliothekslisten sind einstufig; das heißt, Angaben in der Bibliotheksliste werden immer als PLAM-Bibliotheken oder Typumsteuerungen aufgefasst. Alle Bibliotheken in der Liste müssen vorhanden sein.

### *Hinweis*

Auf Bibliothekslisten kann nur lesend zugegriffen werden, d.h. spezifiziert der Operand LIBRARY eine Ausgabebibliothek, ist keine Bibliotheksliste erlaubt. Bei Bibliothekslisten dürfen keine Wildcard-Angaben verwendet werden.

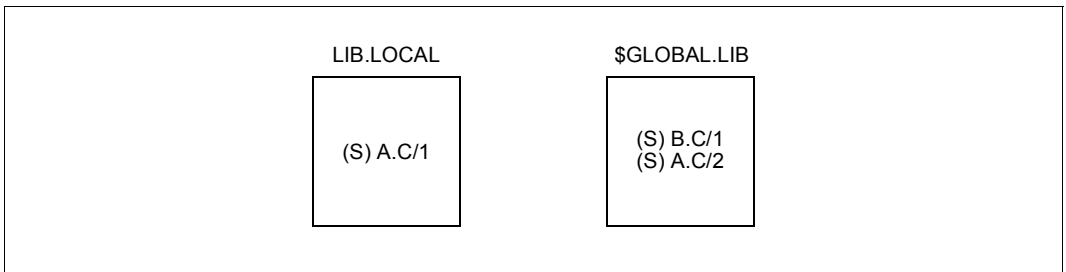


Bibliothekslisten werden von anderen Produkten nur dann unterstützt, wenn dies explizit im jeweiligen Handbuch beschrieben ist. Die Verwendung von Bibliothekslisten bei allen anderen Produkten kann zu unerwartetem Verhalten führen.

### *Beispiel*

Mit Bibliothekslisten kann die make-Funktionalität komfortabel für Projekte verwendet werden, bei denen Sourcen mit dem Ausleihmechanismus verwaltet werden. Die Sourcen befinden sich in einer der Bibliotheken LIB.LOCAL oder \$GLOBAL.LIB. Mit dem Ausleihmechanismus wird zwischen diesen beiden Bibliotheken hin und her kopiert. Im Makefile wird SYSPLAMALT-HUGO direkt als einzige Source-Bibliothek angegeben. Der Inhalt von SYSPLAMALT-HUGO bestimmt, welche Bibliotheken durchsucht werden sollen.

Die Bibliotheken haben folgende Inhalte:



Zusammenfassen der Bibliotheken LIB.LOCAL und \$GLOBAL.LIB für den lokalen Make-Lauf:

```
/set-variable sysplamalt-hugo = '(lib.local,$global.lib)'
```

Es wird A.C aus LIB.LOCAL berücksichtigt, B.C aus \$GLOBAL.LIB.

Bei zentralem Make-Lauf enthält SYSPLAMALT-HUGO nur eine Bibliothek:

```
/set-variable sysplamalt-hugo = '($global.lib)'
```

Es werden nur bereits wieder zurückgegebene Sourcen A.C und B.C aus \$GLOBAL.LIB berücksichtigt. Die Anweisungen für Make sind in beiden Fällen die gleichen:

```
//modify-make-defaults library=sysplamalt-hugo
```

### Gültigkeit von globalen und lokalen Bibliotheken

Eine globale Bibliothek gilt solange für den gesamten LMS-Lauf, bis entweder eine neue globale Bibliothek eröffnet wird oder die Bibliothek durch eine explizite CLOSE-LIBRARY-Anweisung geschlossen wird.

Wird zudem eine lokale Bibliothek eröffnet, dann ist diese nur innerhalb dieser Anweisung gültig. Nach Beendigung dieser Anweisung gilt wieder die globale Bibliothek.

Die LMS-Anweisung SHOW-LIBRARY-STATUS (siehe [Seite 437](#)) gibt Auskunft über den Zustand der Bibliotheken.

### Arbeiten mit mehreren Bibliotheken

Es kann immer nur eine globale Bibliothek zugewiesen werden. Soll gleichzeitig mit zwei Bibliotheken gearbeitet werden, muss die zweite Bibliothek lokal vereinbart werden.

Werden im LMS-Lauf mehrere Bibliotheken global zugewiesen (durch mehrere OPEN-LIBRARY-Anweisungen), verwendet LMS die zuletzt angegebene Bibliothek.

#### *Beispiel*

Das Element xyz, das sich in der Bibliothek lib1 befindet, wird in die bereits existierende Bibliothek lib2 kopiert. Es soll lib1 als globale Bibliothek und lib2 als lokale Bibliothek benutzt werden. Nachdem nach dem Aufruf von LMS der Elementtyp noch nicht definiert ist, wird er als globaler Elementtyp S vereinbart. Das Element soll nicht umbenannt werden:

```
/START-LMS
.
.
//MODIFY-LMS-DEFAULTS TYPE = S
//OPEN-LIBRARY LIBRARY = lib1
//COPY-ELEMENT ELEM = *LIB-ELEM(xyz), TO-ELEM = *LIB-ELEM = (LIBRARY = lib2)
.
.
```

## Umlenkmechanismus

Der Benutzer kann neben den Standard-Elementtypen eigene, von ihm definierte Elementtypen verwenden. Diese benutzerdefinierten Typen sollten von einem Standardtyp abgeleitet sein, d.h. für den einstelligen Standardtyp werden als Synonym ein oder mehrere aussagekräftige Namen definiert (z.B. COBSRC, ASSSRC, ...).

Programme, die nicht auf benutzerdefinierte Typen vorbereitet sind, können trotzdem mit diesen Elementen arbeiten. PLAM bietet einen Umlenkmechanismus an, um auf Bibliotheken mit benutzerdefinierten Elementtypen zugreifen zu können. Alle Kommandos und Programme in BS2000/OSD können die Typumlenkung benutzen.

Der Umlenkmechanismus ist durch die S-Variable SYSPLAMLIB realisiert. Sie ist eine Variable vom Typ *Struktur*. Ihre Strukturelemente bestimmen, was umgelenkt und wohin umgelenkt wird.

Der Name des Strukturelements bestimmt das Was,  
der Inhalt des Strukturelements das Wohin.

Der Name des Strukturelements setzt sich zusammen aus dem Dateikettungsnamen und dem benutzerdefinierten Elementtyp.

Der Inhalt des Strukturelements setzt sich zusammen aus

- dem Bibliotheksnamen und dem Elementtyp oder
- dem Dateikettungsnamen und dem Elementtyp.

Wird kein Elementtyp angegeben, gilt der Wert des Elementtyps aus dem Namen des Strukturelements.

Die S-Variable SYSPLAMLIB, in Form einer Variablenzuweisung geschrieben, hat folgenden Aufbau:

```
SYSPLAMLIB.<link>.<utype>=´<libname>(TYPE=<typename> / *SAME)´
```

oder

```
SYSPLAMLIB.<link>.<utype>=´*LINK(LINK=<linkname>,TYPE=<typename> / *SAME)´
```

link	Dateikettungsname in der Form <structured-name1..8>
utype	benutzerdefinierter Elementtyp in der Form <alphanum-name1..8> Üblicherweise ist ein benutzerdefinierter Elementtyp ein Synonym zu einem real vorhandenen Standardtyp.
typename	Elementtyp in der Form <alphanum-name1..8> Wird der Elementtyp nicht angegeben, gilt: typename=utype
libname	Bibliotheksname
linkname	Dateikettungsname in der Form <structured-name1..8>
*SAME	Der Elementtyp des Inhalts des Strukturelements ist der gleiche wie der Elementtyp im Namen des Strukturelements.

Der Inhalt der S-Variablen ist ein String, der eine Bibliotheks- und eine Typkomponente enthält, die die reale Bibliotheksumgebung widerspiegelt. TYPE=\*SAME im Variableninhalt bedeutet <type>. Die Bibliothekskomponente darf keine Bibliotheksliste sein.

Der Zugriff über die S-Variable SYSPLAMLIB kann auf zwei Arten erfolgen:

- Angabe einer Bibliothek über den Dateinamen SYSPLAMLIB.<link>
- Angabe über den Dateikettungsnamen <link>

Ist SYSPLAMLIB als Variable mit SCOPE=\*VISIBLE oder =\*TASK deklariert, wird der Inhalt des Strukturelements ausgewertet und in die entsprechende Bibliothek umgelenkt.

### Beispiel

COBOL-Quellen vom Elementtyp COBSRC sollen übersetzt werden. Sie stehen in der Bibliothek MY.COBLIB, die durch den Dateikettungsnamen SRCLIB bestimmt ist. Die Copy-Elemente sollen vom Typ COBCOPY sein und ebenfalls in der Bibliothek MY.COBLIB stehen.

```

/DECLARE-VARIABLE SYSPLAMLIB(TYPE=*STRUCTURE),SCOPE=*TASK _____ (1)
/SET-VARIABLE SYSPLAMLIB.SRCLIB.S = 'MY.COBLIB(COBSRC)' _____ (2)
/SET-VARIABLE SYSPLAMLIB.COBLIB.S = 'MY.COBLIB(COBCOPY)' _____ (3)
/ADD-FILE-LINK FILE-NAME=SYSPLAMLIB.SRCLIB,LINK=SRCLIB _____ (4)
/ADD-FILE-LINK FILE-NAME=SYSPLAMLIB.COBLIB,LINK=COBLIB
/MODIFY-JOB-SWITCHES ON=1
/ASSIGN-SYSDTA *SYSCMD
/ASSIGN-SYSLST #L
/START-EXECUTABLE-PROGRAM COBOL85
COMOPT SOURCE-ELEMENT=LMSCOBS
COMOPT SOURCE-VERSION=399
COMOPT MODULE=MY.COBLIB
COMOPT SYSLST=(DIAG,MAP,SOURCE)
END
/STEP
/MODIFY-JOB-SWITCHES OFF=1
/EXIT-PROC

```

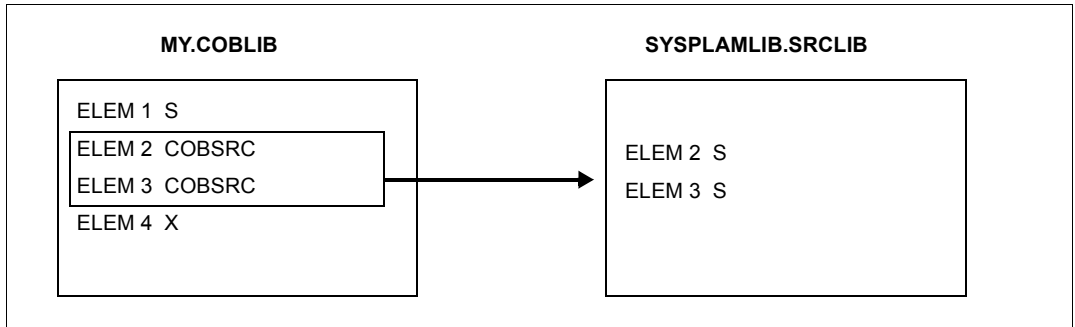
- (1) Deklarieren der zentralen S-Variablen (mit Scope Task)
- (2) Setzen der Umlenkung:  
über den Kettungsnamen SRCLIB oder den Dateinamen SYSPLAMLIB.SRCLIB wird die Umlenkung wirksam.

Wenn z.B. vom Cobol-Compiler auf den Typ S zugegriffen wird, wird dies PLAM-intern auf den Typ COBSRC abgebildet.

- (3) Die Copy-Elemente werden wie unter (2) auf den Typ COBCOPY abgebildet.

- (4) Der Dateikettungsname SRCLIB wird mit einer tatsächlich vorhandenen Bibliothek SYSPLAMLIB.SRCLIB gekoppelt. Das ist notwendig, wenn Programme dezentral das Vorhandensein der Bibliothek testen.

Nach SYSPLAMLIB.SRCLIB.S = 'MY.COBLIB(COBSRC)' entsteht folgende Situation:



Nur auf die Elemente vom Typ COBSRC kann zugegriffen werden und nicht auf alle in der Bibliothek vorhandenen. Der Zugriff erfolgt über den Elementtyp S und bei Programmausgaben wird anstelle des „echten“ Typs COBSRC der Typ S verwendet.

In der Bibliothek MYCOPY werden Elemente mit dem Elementtyp COBCOPY gehalten. In einem LMS-Lauf sollen sie mit dem Elementtyp S bezeichnet werden. Zur Umgehung von LMS-Prüfungen muss die Bibliothek SYSPLAMLIB.COBLIB vorhanden sein.

```

/DECLARE-VARIABLE SYSPLAMLIB(TYPE=*STRUCTURE)
/SET-VARIABLE SYSPLAMLIB.COBLIB.S = 'MYCOPY(COBCOPY)'
/ADD-FILE-LINK SYSPLAMLIB.COBLIB,LINK=COBLIB
/START-LMS
//SHOW-ELEM-ATTR *LIB(SYSPLAMLIB.COBLIB)
    "Alle Elemente vom Typ COBCOPY aus MYCOPY werden angezeigt. Typ S wird
    protokolliert."
//SHOW-ELEM-ATTR *LIB(SYSPLAMLIB.COBLIB,*ALL(*ALL),COBCOPY)
    "Kein Treffer (der Typ COBCOPY existiert nicht in der Benutzersicht)"
//SHOW-ELEM-ATTR *LIB(SYSPLAMLIB.COBLIB,*ALL(*ALL),S)
    "Alle Elemente vom Typ COBCOPY aus MYCOPY werden angezeigt."
//SHOW-ELEM-ATTR (*LINK(COBLIB))
    "Alle Elemente vom Typ COBCOPY aus MYCOPY werden angezeigt. Typ S wird protokolliert."
//END
  
```

## 5.3 Bearbeiten von Elementen

Das folgende Kapitel gibt einen Überblick über die Möglichkeiten, mit LMS Elemente zu bearbeiten.

LMS kann Elemente

- sowohl als Voll- als auch als Delta-Elemente in Bibliotheken aufnehmen,
- in Dateien ausgeben,
- in andere Bibliotheken ausgeben (kopieren),
- auflisten,
- löschen,
- vergleichen,
- umbenennen,
- editieren,
- korrigieren und
- das Inhaltsverzeichnis der Bibliothek ausgeben.

Alle in diesem Abschnitt erwähnten LMS-Anweisungen sind ab [Seite 164ff](#) beschrieben.

### 5.3.1 Aufnehmen von Elementen in eine Bibliothek

Folgende Anweisungen nehmen Elemente in die zugewiesene Bibliothek auf:

ADD-ELEMENT, COPY-ELEMENT und MODIFY-LOGGING-PARAMETERS.

Der Operand WRITE-MODE bestimmt dabei, ob ein gleichbezeichnetes Element in der Ausgabebibliothek überschrieben wird oder nicht.

#### **ADD-ELEMENT**

Die Anweisung ADD-ELEMENT (siehe [Seite 173](#)) nimmt Dateien, Module aus dem EAM-Bereich und Datensätze aus dem LMS-Anweisungsstrom als Elemente in die zugewiesene Bibliothek auf. Wird keine Bibliothek angegeben, wird die durch OPEN-LIBRARY eröffnete Bibliothek genommen.

In dieser Anweisung kann der Benutzer zusätzlich festlegen, ob das Element als Voll- oder Delta-Element gespeichert wird.

Die Satzformate FIXED und UNDEFINED werden in das Satzformat VARIABLE umgewandelt; d.h. mit einem 4 Byte Satzkopf versehen. In Bibliotheken können Dateien mit einer RECORD-SIZE bis maximal 32 KByte (incl. Satzkopf) aufgenommen werden.

Wird eine ISAM-Datei aufgenommen, bestimmt der Operand SOURCE-ATTRIBUTES, ob die Dateiattribute, der ISAM-Schlüssel und Informationen über ISAM-Sekundärschlüssel mit aufgenommen werden.



Es können dann ISAM-Schlüssel bis zu einer Länge von 255 Byte abgelegt werden.

Elemente mit ISAM-Schlüssel eignen sich nur für die Archivierung (siehe Hinweis).

Ist der Operand SOURCE-ATTRIBUTES=\*KEEP gesetzt, ist es auch möglich, Dateien mit RECORD-FORMAT=\*FIXED aufzunehmen, ansonsten ist nur RECORD-FORMAT=\*VARIABLE zulässig.

#### *Hinweise*

- Die ISAM-Schlüssel einer Quellprogrammdatei sollten nicht mit ins Element aufgenommen werden. Der Compiler kann das Quellprogramm, wenn ISAM-Schlüssel enthalten sind, aus diesem Element nicht fehlerfrei übersetzen.
- Wird die Systemdatei SYSDTA einem Element zugewiesen, das die ISAM-Schlüssel gespeichert hat, werden die ISAM-Schlüssel mitgelesen. Die ISAM-Schlüssel müssen dann vom verarbeitenden Programm selbst entfernt werden.

Dateien können unter folgenden Elementtypen abgelegt werden:

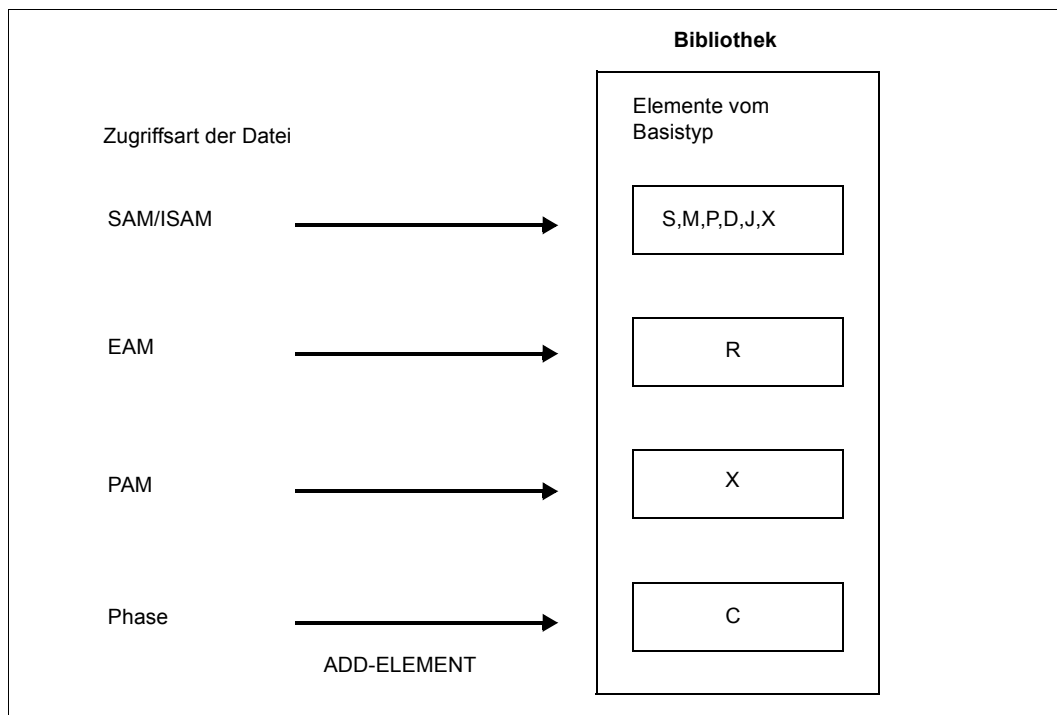


Bild 4: Aufnahmen von Elementen mit ADD-ELEMENT

### COPY-ELEMENT

Die Anweisung COPY-ELEMENT (siehe [Seite 220](#)) kopiert Elemente von der Eingabebibliothek in die Ausgabebibliothek und legt sie dort, falls gewünscht, mit einer anderen Elementbezeichnung ab. Mit dieser Anweisung legt der Benutzer zusätzlich fest, ob das Element als Voll- oder Delta-Element gespeichert wird.

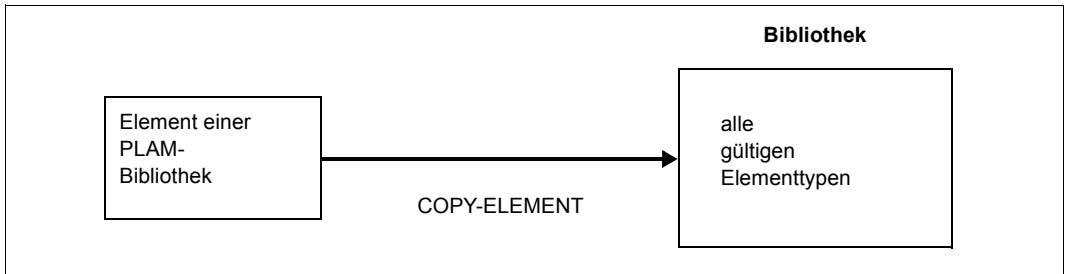


Bild 5: Aufnahmen von Elementen mit COPY-ELEMENT

### MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=\*LIBRARY-ELEMENT

Die Anweisung MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=\*LIBRARY-ELEMENT (siehe [Seite 360](#)) nimmt das LMS-Protokoll (siehe [Seite 102](#)) in das durch \*LIBRARY-ELEMENT spezifizierte Element auf.

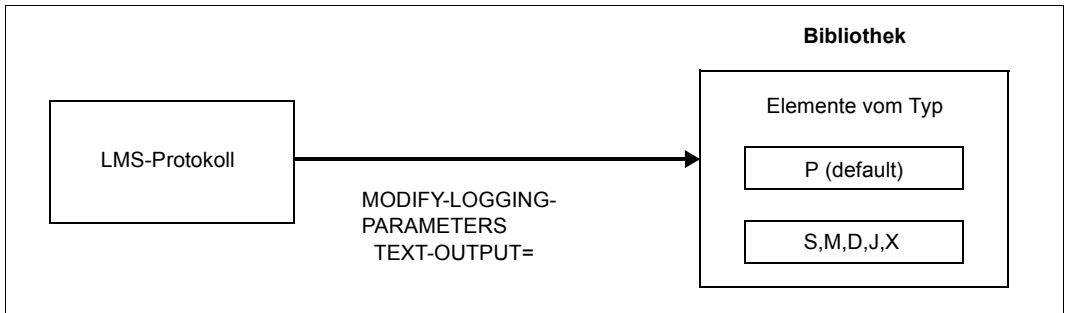


Bild 6: Abspeichern des LMS-Protokolls in ein Element

### 5.3.2 Ausgeben von Elementen in eine Datei

Die Elemente einer Bibliothek werden mit der Anweisung EXTRACT-ELEMENT (siehe [Seite 265](#)) in eine Datei ausgegeben.

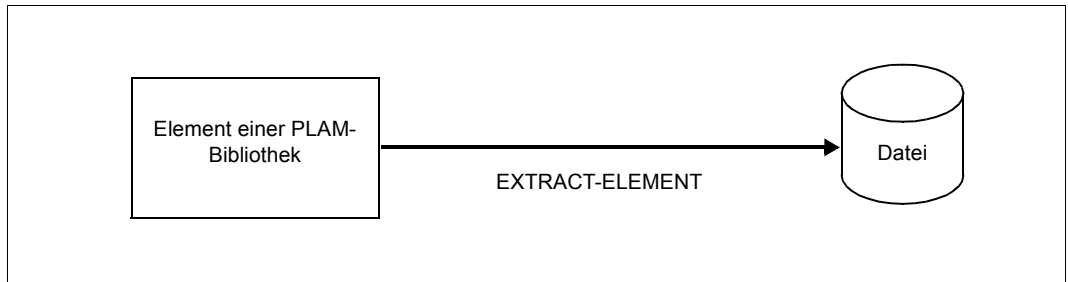


Bild 7: Ausgeben von Elementen

### 5.3.3 Auflisten von Elementen

Das Auflisten von Elementen wird durch die Anweisung SHOW-ELEMENT (siehe [Seite 400](#)) gesteuert. Dabei kann bestimmt werden, in welchem Format und mit welcher Informationsmenge die Elemente ausgegeben werden sollen.

### 5.3.4 Löschen von Elementen

Die Anweisung DELETE-ELEMENT (siehe [Seite 238](#)) löscht Elemente in der zugewiesenen Bibliothek.

Es wird zwischen logischem und physikalischem Löschen unterschieden:

- Logisches Löschen  
Die Einträge im Inhaltsverzeichnis werden gelöscht und der Speicherplatz des entsprechenden Elementes wird freigegeben.
- Physikalisches Löschen  
Zusätzlich zum logischen Löschen wird der Speicherplatz des entsprechenden Elementes mit binär Null überschrieben.

Ein Element einer Bibliothek wird physikalisch gelöscht, wenn der Operand DESTROY-DATA=\*YES gesetzt ist, im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder die CLASS2-Option DESTLEV es verlangt.

Delta-Elemente werden erst dann physikalisch gelöscht, wenn das letzte Delta-Element eines Delta-Baums, d.h. der komplette Delta-Baum, gelöscht wird.

### 5.3.5 Vergleichen von Elementen

Mit der Anweisung COMPARE-ELEMENT (siehe [Seite 208](#)) werden Elemente verglichen. Die Elemente werden dabei Satz für Satz verglichen, wobei über den Operanden RECORD-PART der Vergleichsbereich festgelegt wird. Es wird immer eine Vergleichsstatistik erstellt.

Die Anweisung COMPARE-ELEMENT wird auch dann ausgeführt, wenn nur eines der Vergleichselemente in den angegebenen Bibliotheken gefunden wird. Dies ermöglicht das Zählen von Sätzen in Elementen.

Werden zwei Elemente miteinander verglichen, verwendet LMS die Begriffe Primärelement und Sekundärelement. Dadurch steht dem Anwender frei, ob er das neue oder das alte Element als Basiselement wählt. LMS betrachtet immer das Sekundärelement als Basis des Vergleichs. D.h. LMS kennzeichnet fehlende Sätze im Sekundärelement als eingefügte Sätze und fehlende Sätze im Primärelement als gelöschte Sätze.

Die festgestellten Unterschiede werden auf Anforderung (Operand INFORMATION) protokolliert. Dieses Protokoll bezeichnet man als Vergleichsprotokoll. Anschließend an das Vergleichsprotokoll kann durch die Anweisung SHOW-STATISTICS die Vergleichsstatistik ausgegeben werden. Das ist eine Tabelle, die das Ergebnis des Vergleichs zahlenmäßig wiedergibt.

Bei der Entscheidung, ob zwei Sätze gleich sind, wird unterschieden zwischen formalem und logischem Vergleich.

Beim formalen Vergleich werden alle Zeichen des Satzes verglichen, während beim logischen Vergleich die Leerzeichen *nicht* berücksichtigt werden.

Das Ergebnis der beiden Vergleichsarten (formal bzw. logisch) wird in der gleichen Weise protokolliert.

### Vergleichsprotokoll

Standardmäßig wird der Vergleichsbereich der Sätze protokolliert und nicht der komplette Satz.

Im Vergleichsprotokoll bezieht sich das Vergleichsergebnis auf den Vergleich zweier Vergleichsbereiche, wobei der Unterschied immer gegenüber dem Sekundärelement dargestellt wird.

Vergleichsergebnis	Bedeutung
SAM (same)	Die Vergleichsbereiche der beiden verglichenen Sätze im Primär- und Sekundärelement entsprechen sich.
DEL (deleted)	Der Satz mit diesem Vergleichsbereich ist nur im Sekundärelement vorhanden.
INS (inserted)	Der Satz mit diesem Vergleichsbereich ist nur im Primärelement vorhanden.

### Vergleichsstatistik

Die Vergleichsstatistik wird mit der Anweisung SHOW-STATISTICS (siehe [Seite 444](#)) angezeigt.

Die Vergleichsstatistik gibt über einen Vergleich folgende Auskunft:

- Anzahl aller verglichenen Sätze im Primär- und Sekundärelement
- Anzahl der eingefügten Sätze
- Anzahl der gelöschten Sätze
- Anzahl der gleichen Sätze

Zusätzlich wird ein Ergebnis des gesamten Vergleichs angegeben:

S (same)	Beim Vergleichen wurden keine Unterschiede festgestellt.
C (changed)	Beim Vergleichen wurden Unterschiede festgestellt.
I (inserted)	Das Sekundärelement wurde nicht gefunden.
D (deleted)	Das Primärelement wurde nicht gefunden.
ERR (error)	Während des Vergleichs trat ein Fehler auf.

### 5.3.6 Korrigieren von Elementen

Abhängig von der Art der Elemente besitzt LMS folgende zwei Korrekturanweisungen:

- EDIT-ELEMENT korrigiert Textelemente mit EDT.  
(Elementtyp S, M, J, P, D, X oder davon abgeleitete Typen)
- MODIFY-ELEMENT korrigiert Binde-, Bindelademodule, Phasen und Textelemente über Subanweisungen.  
(Elementtyp R, L, C)  
(Elementtyp S, M, J, P, D, X oder davon abgeleitete Typen)

#### Korrigieren mit EDIT-ELEMENT

Die Anweisung EDIT-ELEMENT (siehe [Seite 245](#)) ruft den Editor EDT als Unterprogramm auf. Das angegebene Element wird dann mit EDT-Anweisungen bearbeitet. Nach Beenden des EDT wird das korrigierte Element in die Ausgabebibliothek geschrieben. Es kann dabei eine neue Elementbezeichnung erhalten.

#### Korrigieren mit MODIFY-ELEMENT

Die Anweisung MODIFY-ELEMENT (siehe [Seite 285](#)) korrigiert Binde-, Bindelademodule, Phasen und Textelemente. Das Korrigieren dieser Elemente wird durch verschiedene Subanweisungen gesteuert. Diese werden unmittelbar hinter MODIFY-ELEMENT bis zur Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

Das korrigierte Element wird dann in die zugewiesene Bibliothek zurückgeschrieben. Es kann dabei eine neue Elementbezeichnung erhalten.

Für die Typen R, L und C sind folgende Funktionen möglich:

- Textsätze korrigieren
- Korrekturen rückgängig machen
- Satzarten aus dem Eingabeelement löschen

Zusätzlich sind nur für Typ R folgende weitere Funktionen möglich:

- REP-Sätze erzeugen
- Merkmale der Programmabschnitte ändern
- Symbole umbenennen

Für Textelemente sind folgende Funktionen möglich:

- Einfügen von Sätzen
- Löschen von Sätzen

### 5.3.7 Umbenennen von Elementen

Die Anweisung MODIFY-ELEMENT-ATTRIBUTES (siehe [Seite 313](#)) benennt die angegebenen Elemente der zugeordneten Bibliothek um. Bei dieser Anweisung können auch Elemente umbenannt werden, deren Bezeichnungen den LMS-Konventionen nicht genügen. Ein Umbenennen von Delta-Elementen ist aus Revisionsgründen nicht zugelassen.

### 5.3.8 Inhaltsverzeichnis einer Bibliothek ausgeben

Die Anweisung SHOW-ELEMENT-ATTRIBUTES (siehe [Seite 414](#)) protokolliert die Inhaltsverzeichniseinträge der angegebenen Elemente oder der gesamten Bibliothek.

Das Inhaltsverzeichnis wird immer nach dem Elementtyp sortiert ausgegeben. Die weitere Sortierreihenfolge bestimmt der Operand SORT. Die Elementbezeichnungen werden standardmäßig nach Typ, Namen, Version und Datum sortiert ausgegeben.

Um das vollständige Inhaltsverzeichnis einer Bibliothek zu erhalten, genügt die Angabe der Anweisung SHOW-ELEMENT-ATTRIBUTES ohne weitere Operanden, sofern über MODIFY-LMS-DEFAULTS kein bestimmter Elementtyp festgelegt wurde.

### 5.3.9 Speichern von Prozeduren

LMS ermöglicht dem Benutzer BS2000-Prozeduren als Elemente in Bibliotheken abzulegen (Elementtyp J).

Bestehende Prozedurdateien können mit ADD-ELEMENT als Elemente in Bibliotheken aufgenommen werden.

Durch das Speichern von Prozeduren kann, insbesondere bei kleinen Kommandodateien, Speicherplatz eingespart werden. Die Zahl der Katalogeinträge wird verringert.

Es ist jedoch darauf zu achten, dass aus den Elementen, die evtl. vorhandene ISAM-Schlüssel mitgespeichert haben (siehe [Seite 88](#)), diese Schlüssel vor dem Aufruf der Prozedur entfernt werden.

Ein Bibliothekselement kann auch als Systemeingabedatei (SYSDTA) mit dem BS2000-Kommando ASSIGN-SYSDTA zugewiesen werden (siehe [\[4\]](#)).

## 5.4 Archivieren mit Delta-Technik

Zum Speichern von mehreren Versionen pro Elementname in Bibliotheken stehen zwei Methoden zur Verfügung:

- die Voll-Speicherung und
- die Delta-Speicherung.

Bei der **Voll-Speicherung** befindet sich genau ein Element, d.h. alle Sätze des Elementes, in einem eigenen Behälter (einer Ablageeinheit der Bibliothek). Wenn unter diesem Namen eine weitere Version aufgenommen wird, werden alle Sätze dieses Elementes ebenfalls in einem eigenen Behälter abgelegt. Eine eventuelle Beziehung der Elemente zueinander ist LMS nicht bekannt.

Im Folgenden werden solche Elemente als Voll-Elemente bezeichnet.

Bei der Bearbeitung, z.B. Lesen eines Voll-Elementes, kann LMS direkt auf alle Sätze des angegebenen Elementes zugreifen und die Aktion ausführen. Diese Methode gilt für alle Elementtypen und eignet sich wegen des schnellen Zugriffs besonders für in Entwicklung bzw. in Änderung befindliche Elemente.

Für die textartigen Elementtypen S, M, J, P, D und X oder davon abgeleitete Typen kann zum Speichern mehrerer Versionen eines Elementes die **Delta-Speicherung** verwendet werden. Bei dieser speicherplatzsparenden Methode wird nur noch das zeitlich erste Element vollständig in einem eigenen Behälter abgelegt. Bei der Aufnahme weiterer Versionen zum gleichen Element werden nicht mehr alle Sätze, sondern nur noch die Differenzmenge zum jeweiligen Vorgänger ermittelt und einsortiert (Vergleich der Elemente). Zusätzlich wird durch die Verknüpfung 'neue Version - Vorgänger-Version' die logische Beziehung der Elemente zueinander bekanntgegeben.

Im Folgenden werden solche Elemente als Delta-Elemente bezeichnet.

Bei der Bearbeitung, z.B. Lesen eines Delta-Elementes, werden die entsprechenden Sätze des angesprochenen Elementes herausgefiltert. Dies kann, wenn eine Vielzahl in Beziehung stehender Delta-Elemente existiert, zu einer Verlangsamung der gesamten Aktion führen. Aus diesen Gründen ist die Delta-Speicherung speziell für die platzsparende und übersichtliche Archivierung von Element-Versionen geeignet.



### 5.4.1 Delta als Speicherungsform und Ordnungsmittel

Delta-Elemente unterscheiden sich von Voll-Elementen nicht nur durch die platzsparende Speicherungsform, sondern auch dadurch, dass zwischen den Delta-Elementen eine eindeutige Beziehung besteht.

Speicherplatz wird gespart durch die Delta-Bildung:

- Redundante Sätze eines Elementes gegenüber dem Vorgänger-Element werden erkannt und nicht erneut gespeichert.
- Um zu erkennen, ob Sätze redundant sind, wird ein formaler Vergleich zwischen den Sätzen der neuen Version und der angegebenen Basisversion vorgenommen.

Eindeutige Beziehungen werden hergestellt durch:

- **Eindeutige Elementnamen**  
Delta-Elemente, die zueinander in Beziehung stehen, werden dadurch charakterisiert, dass sie den gleichen Elementnamen tragen. Sie bilden einen Namensraum. Daher müssen Voll-Elemente und Delta-Elemente unterschiedliche Elementnamen haben. Das bedeutet,
  - ein Voll-Element wird nur dann erstellt, wenn kein gleichnamiges Delta-Element existiert,
  - ein Delta-Element wird nur dann erstellt, wenn kein gleichnamiges Voll-Element existiert.

Alle Delta-Elemente gleichen Namens bilden genau ein logisches Gebilde "Delta-Baum", in seiner einfachsten Form eine „Delta-Sequenz“.

Zu einem Namen gibt es genau einen Delta-Baum.

- **Eindeutige Versionsbezeichnungen**  
Beim Aufnehmen eines Elementes als Delta-Element kann angegeben werden, welches Element der Vorgänger ist, d.h. als (Vergleichs-)Basis für den Vergleich dienen soll. Standardmäßig gilt `BASE=*STD`.

Die bei der Elementaufnahme getroffenen Vereinbarungen hinsichtlich Elementname und Versionsbezeichnung können aus Revisions- und Konsistenzgründen nicht nachträglich manipuliert werden, ggf. ist durch Kopieren des Delta-Baums mit gleichzeitiger Umbenennung ein neuer Delta-Baum zu schaffen.

## 5.4.2 Aufnahmen von Delta-Elementen

Delta-Elemente werden mit Hilfe des Operanden STORAGE-FORM=\*DELTA in den Anweisungen ADD-ELEMENT, COPY-ELEMENT bzw. EDIT-ELEMENT in die zugewiesene Bibliothek aufgenommen.

Soll LMS die Speicherform des neuen Elementes festlegen, muss STORAGE-FORM=\*STD angegeben werden. LMS bestimmt an Hand der Existenz des neuen Namens, ob das neue Element als Voll- oder als Delta-Element aufgenommen wird. Existiert der Name für das neu aufzunehmende Element bereits als Delta, so wird ein neues Delta-Element erzeugt, ansonsten ein Vollelement.

### Aufnahmen eines Elementes in eine Delta-Sequenz

Eine Delta-Sequenz ist ein linearer Delta-Baum, d.h. ein Delta-Baum ohne Verzweigungen.

So wird z.B. durch die Angabe von VERSION=\*INCREMENT und BASE=\*STD ein Element in die Delta-Sequenz aufgenommen:

Der Vorgänger ist das gleichnamige Delta-Element mit der höchsten Versionsbezeichnung. Das neue Delta-Element wird an die vorhandene höchste Version angehängt, die Delta-Menge wird ermittelt und im Behälter abgelegt. Die Versionsbezeichnung des neuen Delta-Elementes sollte höher als die bisher höchste sein.

Existiert kein Vorgänger, wird das Element als erstes Element einer Delta-Sequenz angelegt.

*Beispiel für eine Delta-Sequenz (die Pfeile sollen die Beziehungen darstellen)*

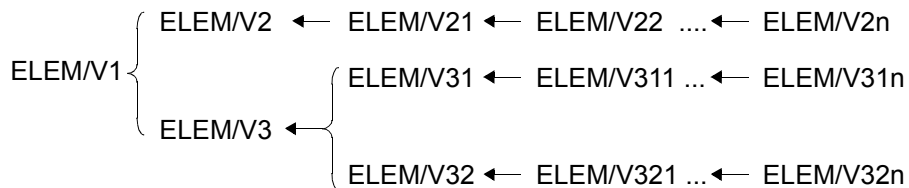
```
ELEM/V1 <-- ELEM/V2 <-- ELEM/V3 ..<-- ELEM/Vn
```

### Aufnehmen eines Elementes in einen Delta-Baum

Durch die Angabe von `VERSION = <text 1..24>` und `BASE = <vorgänger>` kann ein Delta-Baum aufgebaut werden:

Der Vorgänger ist das gleichnamige Delta-Element mit der angegebenen Versionsbezeichnung. Das neue Delta-Element kann seitlich an die angegebene Version angehängt werden. Aus der Versionsbezeichnung des neuen Delta-Elementes sollte die Beziehung zum Vorgänger-Element abzulesen sein.

*Beispiel für einen Delta-Baum (die Pfeile sollen die Beziehungen darstellen)*



### 5.4.3 Übersicht über Delta-Elemente

Eine Übersicht über die vorhandenen Delta-Elemente einer Bibliothek wird ausgegeben durch den Operanden `INFORMATION = *DELTA-STRUCTURE` in der Anweisung `SHOW-ELEMENT-ATTRIBUTES`. Es wird dann immer ein kompletter Delta-Baum aufgelistet, unabhängig davon, welches Element angegeben wurde.

Zusätzlich zur Elementbezeichnung wird die interne Delta-Nummer der Elemente im Feld `DELTA#` und die interne Nummer des Vorgängers im Feld `BASE#` ausgegeben.

Die Delta-Nummer spiegelt die chronologische Reihenfolge der Elementaufnahme wider. Beide Nummern beschreiben die Verkettung der Elemente in einem Baum eindeutig, sie können vom Benutzer nicht beeinflusst werden. Der Benutzer kann die Verkettung der Elemente durch geeignete Versionsvergabe sichtbar festschreiben.

## 5.4.4 Löschen von Delta-Elementen

Es ist zu unterscheiden zwischen dem logischen und physikalischen Löschen:

- **Logisches Löschen**  
Hierbei wird lediglich der Eintrag aus dem Inhaltsverzeichnis entfernt. Die Datensätze und die Beziehung zum Vorgänger bleiben erhalten.
- **Physikalisches Löschen**  
Abhängig vom Operanden DESTROY-DATA (er muss beim Aufnehmen des Elementes gesetzt werden) wird zusätzlich der Speicherplatz des Elementes mit binären Nullen überschrieben.  
Für Delta-Elemente wird dieser Operand jedoch erst dann wirksam, wenn das letzte Element des Delta-Baumes gelöscht wird. Die Angabe von DESTROY-DATA=\*YES genügt bei einem Element, um den gesamten Behälter zu löschen.

## 5.4.5 Sperren von Delta-Elementen

Bei der Bearbeitung eines **Voll-Elementes** wird das angegebene Element gesperrt (Element=Behälter).

Bei der Bearbeitung eines **Delta-Elementes** werden alle Elemente, die im angesprochenen Behälter liegen, gesperrt (Delta-Baum=Behälter).

## 5.4.6 Besonderheiten der Delta-Technik

Im Folgenden sind einige Besonderheiten aufgezählt, die bei der Archivierung von Elementen mit der Delta-Technik zu beachten sind:

- Ein **Umbenennen** von einzelnen Delta-Elementen oder eines ganzen Delta-Baums ist nicht möglich, weil dies die Revisionsfähigkeit eines Archivs unterlaufen würde bzw. die Konsistenz der Daten gefährdet sein kann, wenn eine laufende Aktion, wie auch immer, abgebrochen wird.
- Ein **Überschreiben** von Delta-Elementen ist nur dann erlaubt, wenn die Delta-Elemente Blätter eines Delta-Baumes sind, d.h. keinen Nachfolger besitzen.

## 5.5 Steuern des LMS-Laufs

Der nachfolgende Abschnitt beschreibt die Möglichkeiten, die LMS bietet, um den LMS-Lauf zu steuern. Dazu gehören Protokollparameter, Benutzeranschlüsse, Auftragsschalter usw.

### 5.5.1 LMS-Protokollparameter

Die LMS-Protokollparameter sind globale Parameter. LMS kennt folgende Protokollparameter:

Protokollparameter	Bedeutung der Protokollparameter
LOGGING	Protokollumfang der Meldungen
OUTPUT	Ausgabemedium für das Protokoll
OUTPUT-LAYOUT	Ausgabeumfang und -format für das Protokoll

Alle Protokollparameter besitzen als Standardwert das Schlüsselwort \*UNCHANGED, d.h. die aktuelle Einstellung wird nicht verändert. Zu Beginn des LMS-Laufs haben die Protokollparameter den unmittelbar hinter \*UNCHANGED stehenden Wert. Sie können aber durch die Anweisung MODIFY-LOGGING-PARAMETERS verändert werden.

## 5.5.2 Protokollausgabe steuern

Das LMS-Protokoll enthält alles, was LMS ausgibt, wie z.B. das Ergebnis der Anweisungen, deren Ausführung oder Abbruch, die zugewiesenen Ein- und Ausgabebibliotheken und Listen, die z.B. beim Auflisten oder Vergleichen von Elementen erzeugt wurden.

Das Protokoll wird in die Systemdateien SYSOUT oder SYSLST, in ein Bibliothekselement oder in die Arbeitsdatei 9 des EDT geschrieben. Wohin LMS ausgibt, legt `MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=` fest.

Wenn das Protokoll in ein Element geschrieben wird, dann erzeugt LMS standardmäßig ein Element mit dem Typ P.

War beim Aufruf von LMS der Auftragsschalter 4 gesetzt, werden die Anfangs- und Endmeldungen von LMS unterdrückt.

Fehlermeldungen werden immer ausgegeben.

Die folgende Tabelle zeigt, welche Operanden in welchen Anweisungen die Ausgabe des Protokolls steuern:

Anweisung	Operand	Funktion
MODIFY-LOGGING-PARAMETERS	LOGGING	Legt fest, ob Erfolgsmeldungen protokolliert werden oder nicht.
	TEXT-OUTPUT	Legt das Ausgabemedium für das Protokoll fest.
	OUTPUT-LAYOUT	Legt das Ausgabeformat des Protokolls fest.
	LINES-PER-PAGE	Legt die Anzahl der Zeilen auf einer Protokollseite fest.
	LINE-SIZE	Legt die Länge einer Zeile fest.
	EXTRA-FORM-FEED	Steuert den Seitenvorschub des Protokolls bei Elementwechsel.
	HEADER-LINES	Legt fest, ob Überschriften ausgegeben werden.
SHOW-ELEMENT	OUTPUT-FORM	Legt die Satzdarstellung beim Auflisten eines Elementes fest.
	TEXT-/MODULE-/ PHASE-/ LLM-INFORMATION	Legt den Ausgabeumfang beim Auflisten eines Elements fest.

Anweisung	Operand	Funktion
SHOW-ELEMENT-ATTRIBUTES	SORT	Legt die Sortierung des Inhaltsverzeichnis fest.
	LAYOUT	Legt das Protokollformat des Inhaltsverzeichnisses fest.
	INFORMATION	Legt den Protokollumfang des Inhaltsverzeichnisses fest.
	TEXT-OUTPUT	Steuert die Protokollausgabe.
COMPARE-ELEMENT	LAYOUT	Legt das Protokollformat des Vergleichsprotokolls fest.
	INFORMATION	Legt den Protokollumfang des Vergleichsprotokolls fest.
	TEXT-OUTPUT	Steuert die Protokollausgabe.

### Erfolgs- und Misserfolgsmeldungen

Ist in der LMS-Anweisung MODIFY-LOGGING-PARAMETERS der Operand LOGGING=\*MAXIMUM gesetzt, wird die Ausführung jeder LMS-Anweisung, die ein Element betrifft, protokolliert. Wird die Anweisung fehlerfrei ausgeführt, dann gibt LMS eine Erfolgsmeldung aus.

Kann die Anweisung nicht ausgeführt werden, protokolliert LMS dies mit einer Misserfolgsmeldung und evtl. der entsprechenden LMS-Fehlermeldung.

Die Meldungen finden Sie über eine HTML-Anwendung auf dem Manual-Server (URL: <http://manuals.ts.fujitsu.com>) und auf der DVD „BS2000/OSD SoftBooks“.

Alle Erfolgs- und Misserfolgsmeldungen haben folgendes Format:

[NO] anweisung elem[word elem][ursache]

NO	Die Anweisung wird nicht ausgeführt.
anweisung	Anweisungsname.
elem	Elementbezeichnung oder Dateiname (bei ADD-ELEMENT und EXTRACT-ELEMENT).
word	Schlüsselwort: AS, INTO, WITH.
ursache	Ergebnis: EXISTING, REPLACED, u.ä.

### 5.5.3 Bildschirmwechsel steuern

LMS führt keine eigene Bildschirmsteuerung durch. Die Steuerung übernimmt das System. LMS-Ausgaben können daher nur abgebrochen werden, wenn die Programmunterbrechungstaste (K2) gedrückt wird und anschließend ein SEND-MSG-Kommando (siehe [Seite 346](#)) eingegeben und abgeschickt wird.

### 5.5.4 Fehlerbehandlung im Dialog- und Prozedurbetrieb

LMS unterscheidet zwischen den beiden Ablaufarten Dialog- und Prozedurbetrieb

- Dialogbetrieb  
Im Dialog erfolgt nach der Ausgabe der Fehlermeldung die Aufforderung zur Eingabe der nächsten Anweisung mit //.
- Prozedurbetrieb  
LMS unterstützt im Prozedurbetrieb zwei Fehlerbehandlungs-Mechanismen. Einen Spin-off-Mechanismus und einen Anweisungs-Returncode-Mechanismus. Welcher von den beiden wirkt, hängt von der Einstellung des Operanden PROPAGATE-STMT-RC im Kommando /BEGIN-BLOCK ab.

#### 5.5.4.1 Spin-off-Mechanismus

Standardmäßig wirkt für LMS Anweisungen der Spin-off-Mechanismus, d.h. beim Auftreten eines Fehlers wird zur nächsten //STEP- oder //END-Anweisung gesprungen. Wird zu //END gesprungen, so beendet sich LMS mit TERM UNIT=STEP, MODE=ABNORMAL. Der Spin-off wird dann auf die nachfolgenden Kommandos propagiert. Der Anwender kann selbst bestimmen, in welchen Fehlerfällen LMS den Spin-off-Mechanismus auslösen soll. Dies steuert er über den Operanden MAX-ERROR-WEIGHT (siehe //MODIFY-LMS-DEFAULTS).

#### *Hinweis*

Tritt während der Verarbeitung von Subanweisungen ein Fehler auf, der zum Abbruch der zugehörigen Hauptanweisung führt, so wird hinter einer angegebenen //STEP-Anweisung immer eine Hauptanweisung erwartet.

#### 5.5.4.2 Anweisungs-Returncode-Mechanismus

Analog den Kommando-Returncodes (siehe Handbuch "BS2000/OSD Benutzer-Kommandos" [4]) unterstützt LMS Anweisungs-Returncodes. Der Anweisungs-Returncode ermöglicht es dem Benutzer, nach jeder LMS-Anweisung auf bestimmte Situationen gezielt zu reagieren.



### Aufbau des Anweisungs-Returncodes

Der Anweisungs-Returncode besteht aus drei Teilen:

Maincode: Meldungsschlüssel; die Bedeutung kann mit /HELP-MSG-INFORMATION abgefragt werden.

Subcode1: Fehlerklasse (dezimal); gibt an, wie schwerwiegend der Fehler ist.

Subcode2: Zusatzinformation (dezimal); z.B. Subcode2 = 2 in Verbindung mit Subcode1 = 0 bedeutet Warnung.

### Steuerung der Ausgabe von Anweisungs-Returncodes

Die Ausgabe von Anweisungs-Returncodes wird mit dem Kommando

```
/BEGIN-BLOCK PROGRAM-INPUT=*MIXED-WITH-CMD( -
/                                     PROPAGATE-STMT-RC=*TO-CMD-RC )
```

aktiviert. Danach können die Anweisungs-Returncodes in S-Prozeduren mit SDF-P-Mitteln ausgewertet werden (siehe Handbuch SDF-P "Programmieren in der Kommandosprache" [12]). Zur Auswertung können die SDF-P Builtin-Funktionen MAINCODE() bzw. MC(), SUBCODE1() bzw. SC1() und SUBCODE2() bzw. SC2() herangezogen werden.

Bei der Ausgabe von Anweisungs-Returncodes wird eine Fehlerbehandlung auf Kommandoebene ausgelöst, d.h. Sprung zum nächsten Kommando /IF-CMD-ERROR , /IF-BLOCK-ERROR oder /STEP.

Die Fehlerbehandlung wird genau dann ausgelöst, wenn der Subcode1 des Anweisungs-Returncodes ungleich 0 ist. Dies ist bei jeder Fehlermeldung der Fall. Die Einstellung von MAX-ERROR-WEIGHT ist wirkungslos.

Der Anweisungs-Returncode kann mit dem Kommando /SAVE-RETURNCODE gesichert werden. /IF-CMD-ERROR führt implizit das Kommando /SAVE-RETURNCODE aus.

#### *Hinweis*

Bei den Maincodes LMS1002, LMS1003 und LMS1004 muss der wahre Fehler dem LMS-Protokoll, wenn vorhanden, entnommen werden. Der Maincode LMS1003 nach einer Anweisung mit Auswahlangabe (Wildcard) bedeutet, dass eventuell mehrere Fehler aufgetreten sind.

Für LMS werden folgende mögliche Anweisungs-Returncodes festgelegt:

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
1	0	LMS0036	Bibliothek nicht zugewiesen
2	0	LMS0053	Element- und Dateiattribute unterschiedlich
2	0	LMS0064	GCCSN-Makro-Fehler; kein CCS-Name spezifiziert
2	0	LMS0071	XHCS nicht geladen
2	0	LMS0084	VTSUCB-Makro-Fehler
2	0	LMS0095	Eingabesätze fehlen
2	0	LMS0102	Unvollständiger Modul in EAM Datei
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
2	0	LMS0151	Ein- oder Ausgabemedium auf Standard gesetzt
2	0	LMS0163	Mindestens ein Satz abgeschnitten
2	0	LMS0199	Satzlänge ungültig bei festem Satzformat
2	0	LMS0201	Nur der Vergleichsbereich wird protokolliert
2	0	LMS0274	Blockkontroll-Wert verändert
2	0	LMS0286	Dateiattribute nicht modifiziert
2	0	LMS0712	Touch nicht möglich
2	0	LMS0714	Touch auf leere Datei nicht möglich
2	0	LMS0721	Das angegebene Ziel ist bereits aktuell
	2	CMD0230	Syntaxfehler
	32	LMS0238	Fehler beim Nachladen von LMS
	64	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0035	Elementschutz auf Datei nicht übertragbar
	64	LMS0093	Protokoll-Element existiert bereits
	64	LMS0211	Bibliothek existiert bereits
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0301	Datei nicht gefunden
	64	LMS0302	Element nicht gefunden
	64	LMS0303	Element nicht im Bereich der Referenz-Bedingung
	64	LMS0304	Typ nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
		LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei

(SC2)	SC1	Maincode	Bedeutung
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0081	Kein weiterer Speicherplatz für SYSLST-Datei
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

(SC2) : Subcode2 = 0 wird durch Leerzeichen dargestellt

Bei der Beschreibung der einzelnen LMS-Anweisungen werden die jeweils möglichen Returncodes noch einmal aufgeführt.

### Beispiel

Element A in LIB1 oder LIB2 suchen :

```

/BEGIN-BLOCK PROGRAM-INPUT=*MIXED-WITH-CMD( -
/      PROPAGATE-STMT-RC=*TO-CMD-RC )
/START-LMS
//SHOW-ELEM-ATTR *LIB(LIB = LIB1, ELEM = A, TYP = S),TEXT-OUT=*NONE
/IF-CMD-ERROR
//  SHOW-ELEM-ATTR *LIB(LIB = LIB2, ELEM = A, TYP = S),TEXT-OUT=*NONE
/  IF-CMD-ERROR
/    SET-VAR LIB = 'keiner'
/  ELSE
/    SET-VAR LIB = 'LIB2'
/  END-IF
/ELSE
/  SET-VAR LIB = 'LIB1'
/END-IF
/WR-TEXT 'Element A befindet sich in &LIB Bibliothek'
//END
/END-BLOCK

```

### 5.5.5 Benutzeranschlüsse

LMS bietet dem Benutzer die Möglichkeit, beim Auflisten oder Vergleichen von Elementen in ein Benutzerprogramm zu verzweigen.

Dieses Unterprogramm kann vor der Bearbeitung eines Elementsatzes folgende Aktionen ausführen:

- Manipulieren des aktuellen Elementsatzes.
- Einfügen eigener Sätze vor dem aktuellen Elementsatz bzw. an das Elementende.
- Den aktuellen Elementsatz von der Verarbeitung ausschließen.

Näheres siehe LMS-Anweisung ACTIVATE-USER-EXIT ab [Seite 166](#).

### 5.5.6 Unterbrechen des LMS-Laufs

Der LMS-Lauf kann durch den Benutzer oder durch das Programm unterbrochen werden.

#### Unterbrechen durch den Benutzer

Der Benutzer kann den LMS-Lauf durch Betätigen einer Programmunterbrechungstaste (z.B. K2) unterbrechen.

Die Fortsetzung des LMS-Laufs kann durch das /SEND-MSG-Kommando gesteuert werden, das wahlweise mit einem Eingabetext versehen werden kann. Dieser Eingabetext wird dann in der Unterbrechungsbehandlung von LMS interpretiert. Die gerade laufende Funktion wird von der Art des Abbruchs unterrichtet und beendet sich in der gewünschten Form. Die möglichen Eingaben sind beim Operanden DIALOG-CONTROL der Anweisung MODIFY-LMS-DEFAULTS beschrieben (siehe [Seite 346](#)).

Unter LMS können noch weitere Programmsysteme wie z.B. der EDT aufgerufen werden. Diese Systeme besitzen u.U. eigene STXIT-Routinen. LMS legt immer einen eigenen STXIT-Verwaltungsblock an.

Verwenden andere Systeme andere STXIT-Routinen, so werden die dort zugehörigen Verwaltungsblocks mit denen von LMS verkettet bzw. umgekehrt.

Beim Eintritt eines Ereignisses werden alle zu diesem Ereignis gehörigen STXIT-Routinen aktiviert, d.h. neben den LMS-Routinen werden auch die Routinen anderer Systeme durchlaufen. Das bedeutet, dass die LMS-Routinen auch dann aktiviert werden, wenn das Unterbrechungsereignis nicht in LMS eintritt.

Ein gutes Beispiel dafür ist das SEND-MSG-Kommando. Über das SEND-MSG-Kommando können Meldungen an LMS und an alle anderen STXIT-Routinen, welche dieses Ereignis behandeln, geschickt werden.

*Hinweis*

Das SEND-MSG-Kommando wird von allen zugehörigen STXIT-Routinen des gesamten Programmsystems behandelt, d.h. es kann zu Problemen führen, wenn einzelne Programmteile auf die gleiche Meldung unterschiedlich reagieren. Das Programm wird im Allgemeinen in dem Teilsystem fortgesetzt, in dem das Ereignis eintrat.

**Abbruch des LMS-Laufs durch Fehler**

Die Fehlerbehandlung wird ebenfalls über die STXIT-Routine gesteuert.

Bei Programmbeendigung, Angabe von /START-EXECUTABLE-PROGRAM, /LOAD-EXECUTABLE-PROGRAM, /CANCEL-JOB, /LOGOFF, /CANCEL-PROGRAM, /ABEND, /EXIT-JOB oder Leitungsverlust, wird sichergestellt, dass die Bibliotheken konsistent bleiben.

Für alle Fälle der Programmbeendigung gilt:

- Alle STXIT-Routinen in LMS werden ausgeschaltet, um ein irreguläres Weiterarbeiten durch SEND-MSG zu unterbinden.
- LMS simuliert ein END. Dadurch werden alle offenen Bibliotheken geschlossen.

Falls zum Zeitpunkt der Programmbeendigung noch Bibliotheken offen sind, werden diese geschlossen.

**5.5.7 Verwenden von Auftragsschaltern**

Über BS2000-Auftragsschalter kann der Benutzer auf den Ablauf von LMS Einfluss nehmen. Sie müssen vor dem Laden von LMS mit dem Systemkommando /MODIFY-JOB-SWITCHES ON=(nr,...) gesetzt werden.

Folgender Auftragsschalter nimmt Einfluss auf den Ablauf von LMS:

Auftragsschalter 4:

Wenn Auftragsschalter 4 gesetzt ist, werden die Anfangs- und die Endemeldung von LMS unterdrückt.

Auftragsschalter werden nur bei der Initialisierung abgefragt, nachträgliches Setzen und Löschen hat für LMS keine Wirkung.

## 5.6 Pamkey-Eliminierung

Zur Erhöhung der Netto-Speicherkapazität und zur Erhöhung der effektiven Datenübertragungsgeschwindigkeit wurden neue Plattenformate eingeführt, speziell die NK-Platte. Die NK-Platte ist ohne Pamkey formatiert (Non-Key-Platte). Die Nutzung der NK-Platte erfordert die Konvertierung von K-Dateien zu NK-Dateien (Pamkey-Eliminierung).

Es gibt für SAM-, ISAM- und UPAM-Dateien mehrere unterschiedliche Dateiformate auf Platte: das bisherige PAM-Schlüssel behaftete Format (kurz K) und die PAM-Schlüssel losen Formate (kurz NK2 und NK4).

Das Dateiformat wird durch den BLKCTRL-Wert festgelegt. BLKCTRL kann den Wert PAMKEY, DATA, DATA2K, DATA4K oder NO annehmen. Einzelheiten zu den Dateiformaten siehe [9].

### 5.6.1 Bibliotheksdateien

Die Unterscheidung  $K \leftarrow NK$  ist zunächst eine DVS bedingte Unterscheidung. Sie überträgt sich wie folgt auf die interne Dateiorganisation der PLAM-Bibliothek:

Der PAM-Schlüssel ist nicht notwendig. Dateiseitig ergibt sich dennoch eine Unterscheidung, die durch das Dateiattribut BLKCTRL repräsentiert wird.

PLAM-Bibliotheken brauchen bei der Migration zwischen K- und NK-Welt nicht mit PAMCONV umgesetzt zu werden.

### 5.6.2 Elementverarbeitung

Das nachfolgende Bild zeigt einen Überblick über die möglichen Situationen beim Transfer von Daten zwischen Datei und Bibliothekselementen.

Für die Elemente sind die logischen Informationseinheiten aufgeführt; für die Dateien ist der BLKCTRL-Wert angegeben.

Die Pfeile beschreiben die Transfer-Richtung.

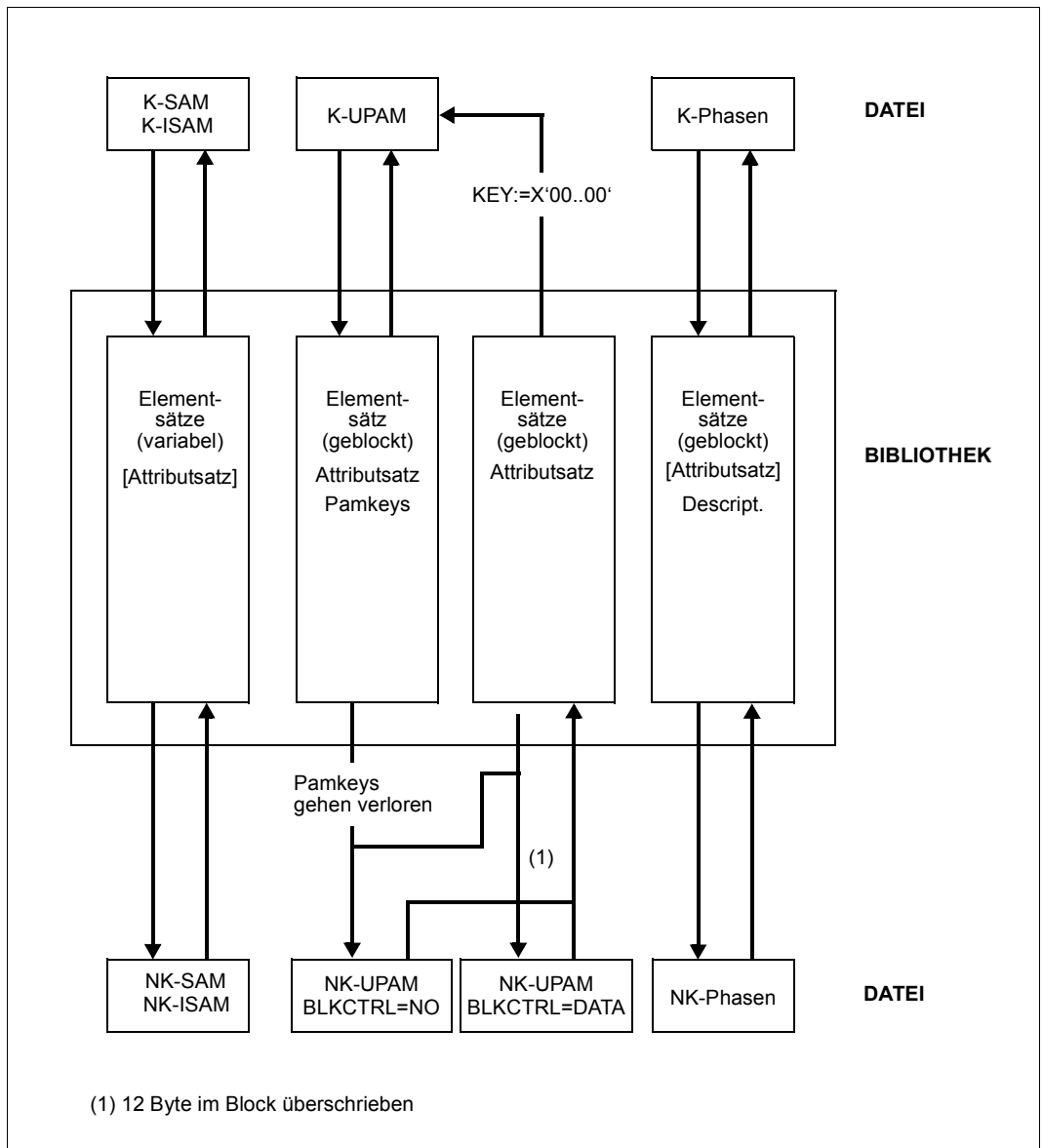


Bild 8: Transfer von Daten zwischen Datei und Bibliothekselementen

### Verhalten bei der ADD-ELEMENT-Anweisung

Über die ADD-ELEMENT-Anweisung werden Datei-Inhalte in Elemente abgelegt. Im Einzelnen gilt für:

– SAM/ISAM-Dateien

Bei Aufnahme von SAM- und ISAM-Dateien wird der BLKCTRL-Wert mit abgespeichert, wenn SOURCE-ATTRIBUTES=\*KEEP gesetzt ist, d.h. die ursprüngliche, vom BLKCTRL-Wert geprägte Blockstruktur der Datei wird im Attributsatz dokumentiert.

Die Daten werden über die logische Zugriffsmethode SAM/ISAM gelesen (Einzelsätze) und unverändert ins Element geschrieben (als Sätze variablen Formates).

Die erzeugte Elementstruktur ist unabhängig von der ursprünglichen BLKCTRL-Eigenschaft.

– PAM-Dateien

Bei Aufnahme von PAM-Dateien wird der BLKCTRL-Wert generell mit abgespeichert. Die Blöcke der Datei werden über die Zugriffsmethode UPAM gelesen und unverändert als Block im Element gespeichert. Sind PAM-Schlüssel vorhanden, d.h. BLKCTRL=PAMKEY, so werden diese PAM-Schlüssel beim Element abgelegt.

Das erzeugte Element behält somit die vom BLKCTRL-Wert geprägte Blockstruktur bei.

– Phasen

Bei Aufnahme von Phasen wird der BLKCTRL-Wert nicht abgespeichert. Die entsprechende Formatangabe ist in der Phaseninformation auf Datei hinterlegt. In der PLAM-Bibliothek haben K-Phasen und NK-Phasen gleiches Format. Die PAM-Schlüssel-Informationen sind in Deskriptoren abgelegt.

ADD datei>element	Datei-Typ	BLKCTRL-Eintrag im Attributsatz	Pamkey-Speicherung
Datei liegt auf NK-Platte	SAM/ISAM	— <sup>1</sup>	
	SAM/ISAM	aus dem Katalog	nein
	UPAM	aus dem Katalog	nein
Datei liegt auf K-Platte	SAM/ISAM	— <sup>1</sup>	
	SAM/ISAM	aus dem Katalog	nein
	UPAM	aus dem Katalog	für BLKCTRL=PAMKEY

<sup>1</sup> die Ablage kann über den Operanden SOURCE-ATTRIBUTES gesteuert werden

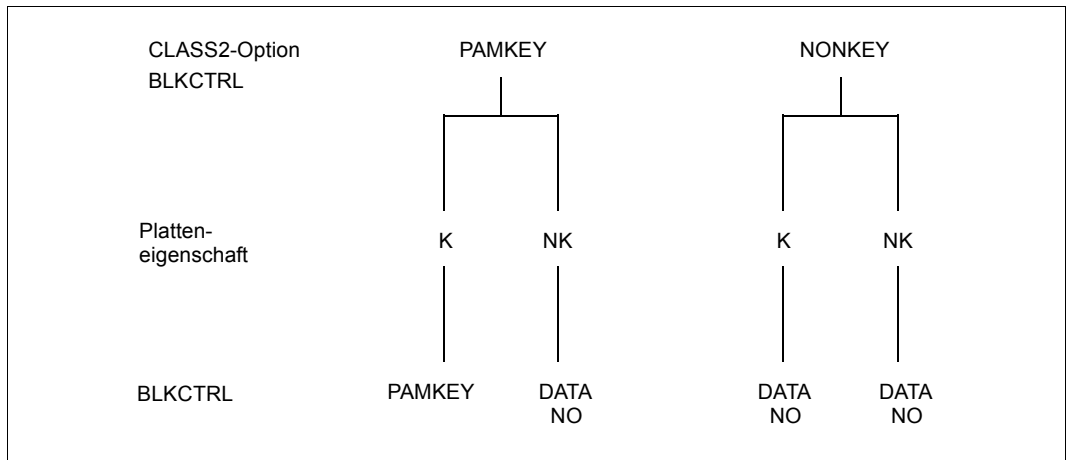


### Verhalten bei der EXTRACT-ELEMENT-Anweisung

Über die EXTRACT-ELEMENT-Anweisung werden Elementinhalte in Dateien ausgegeben. Der BLKCTRL-Wert wird über folgende Hierarchie bestimmt:

1. Angabe im Katalogeintrag oder ADD-FILE-LINK-Kommando.
2. Gespeicherter BLKCTRL-Wert beim Element. Er ist nur bei ursprünglichen PAM-Dateien relevant.
3. Einstellung der CLASS2-Option BLKCTRL: PAMKEY oder NONKEY. Er kann mit dem Kommando SHOW-SYSTEM-PARAMETERS angezeigt werden.
4. Platteneigenschaft K oder NK.

Ist kein Katalogeintrag vorhanden und der BLKCTRL-Wert nicht gespeichert, so entscheidet die CLASS2-Option und die Platteneigenschaft über den BLKCTRL-Wert:



Ist die CLASS2-Option BLKCTRL auf PAMKEY gesetzt, so lässt LMS das System den BLKCTRL-Wert bestimmen, d.h. BLKCTRL = <not specified>.

Ist die CLASS2-Option auf NONKEY gesetzt, so setzt LMS für SAM- und ISAM-Dateien BLKCTRL = DATA und für PAM-Dateien BLKCTRL = NO.

Im Einzelnen gilt für:

- ISAM-Dateien

Die variabel langen Elementsätze werden mit der logischen Zugriffsmethode ISAM geschrieben. Der BLKCTRL-Wert der Datei wird nach dem oben beschriebenen Algorithmus bestimmt, wobei allerdings Punkt 2) entfällt, da der beim Element gespeicherte BLKCTRL-Wert nur dokumentarischen Charakter hat; er wird ignoriert.

- SAM-Dateien

Bei BLKCTRL=DATA tritt ein DVS-Fehler auf, wenn im Element Sätze länger als 32Kbyte - 16Byte sind. In der K-Welt dürfen diese Sätze bis zu 32Kbyte - 4Byte lang sein. LMS reicht beim Selektieren zu lange Sätze ungeprüft an DVS weiter.

Der BLKCTRL-Wert wird wie bei ISAM ermittelt.

- PAM-Dateien (Typ X Elemente)

In der NK-Welt gehen die PAM-Schlüssel verloren. Zusätzlich werden bei BLKCTRL=DATA die ersten 12 Byte eines jeden logischen Blockes durch das System überschrieben. In beiden Fällen gibt LMS eine Warnung aus. Im Falle von Phasen und PLAM-Dateien unterbleibt die Warnung.

- Phasen (Typ C Elemente)

Phasen werden gesondert behandelt.

Neben dem alten Phasenformat (K-Phase) gibt es ein PAM-Schlüssel-freies, neues Phasenformat (NK-Phase) auf Dateiseite.

### 5.6.3 Zusammenfassung

– SAM/ISAM-Dateien

Das Aufnehmen der Dateien ist immer möglich; ebenso der Selektiervorgang. Ein evtl. gespeicherter BLKCTRL-Wert hat nur dokumentarischen Charakter.

Die innere Dateiform wird stets durch die Zugriffsmethode SAM/ISAM bestimmt. Diese übernimmt auch die Konversion der Datensätze in das „innere“ Blockformat der Datei.

– UPAM-Dateien

Eine evtl. notwendige Konversion der Daten kann nicht automatisch erfolgen (weder durch die Zugriffsmethode UPAM noch durch LMS), da in diesen Fällen ein Datenverlust eintritt.

Die Steuerung liegt letztendlich beim Benutzer.

Datei-Type U P A M	Erzeugter / abgespeicherter BLKCTRL-Eintrag im Attributsatz			
	PAMKEY	DATA	NO	—
Datei liegt auf NK-Platte	<sup>1</sup> <sub>2</sub>	ADD EXTRACT	ADD EXTRACT	— EXTRACT
Datei liegt auf K-Platte	ADD EXTRACT	ADD EXTRACT	ADD EXTRACT	— EXTRACT

<sup>1</sup> der Wert BLKCTRL=PAMKEY ist nicht möglich.

<sup>2</sup> der Selektiervorgang muss vom Benutzer gesteuert werden, z.B. über Angabe eines Dateikettungsname in der Anweisung.

## 5.7 Unterstützung von NK4-Platten

Im BS2000 gibt es zwei Formate von PLAM-Bibliotheken, die LMS unterstützt:

- das 2K-orientierte Format (NK2-PLAM-Datei)
- das 4K-orientierte Format (NK4-PLAM-Datei)

Die Konvertierung zwischen diesen beiden Formaten wird mit der Anweisung COPY-LIBRARY vorgenommen (vgl. Beispiel zu COPY-LIBRARY, [Seite 233](#)). Der Anwender bestimmt das jeweilige Format im BS2000-Kommando ADD-FILE-LINK und dem Operanden BUFFER-LENGTH=:

```
NK2-PLAM-Datei : BUFFER-LENGTH=STD(SIZE=1)
NK4-PLAM-Datei : BUFFER-LENGTH=STD(SIZE=2)
```

LMS unterstützt beide Bibliotheksformate. Desweiteren unterstützt LMS NK4-Platten mit den Anweisungen ADD-ELEMENT und EXTRACT-ELEMENT.

### 5.7.1 Aufnahmen von Dateien mit ADD-ELEMENT

Mit ADD-ELEMENT können Dateien beliebiger BUFFER-LENGTH in eine PLAM-Bibliothek aufgenommen werden.

### 5.7.2 Ausgeben von Elementen mit EXTRACT-ELEMENT

Das Verhalten ist verschieden bei Elementen mit Attributsatz und bei Elementen ohne Attributsatz.

#### Element mit Attributsatz

Das Element besitzt einen Attributsatz mit der originalen BUFFER-LENGTH-Angabe. Dies ist der Fall, wenn in der ADD-ELEMENT-Anweisung der Operand SOURCE-ATTRIBUTES=\*KEEP angegeben wurde oder wenn das Element eine ursprüngliche UPAM-Datei, z.B. auch eine PLAM-Bibliothek, war.

Für die Ausgabe dieser Elemente ist folgendes zu beachten:

- Für die Zieldatei ist ein BUFFER-LENGTH-Wert explizit vorgegeben, entweder durch einen Eintrag in der Task-File-Table (TFT) über das ADD-FILE-LINK-Kommando oder direkt im Katalogeintrag. In diesem Fall wird stets der vorgegebene Wert verwendet. Bei der Übernahme des Wertes können folgende Probleme auftreten:

- SAM/ISAM-Datei  
Die Elementsätze sind zu lang für die vorgegebene BUFFER-LENGTH. Dann wird ein DVS-Fehler ausgegeben.

- UPAM-Datei  
Beim Erzeugen von UPAM-Dateien füllt LMS einen logischen Block (bis auf den letzten) dicht mit 2K-Einheiten auf und gibt ihn erst dann mit UPAM aus.

Bei BLOCK-CONTROL-INFORMATION=WITHIN-DATA-BLOCK beginnt jeder logische Block (BUFFER-LENGTH) mit einem 12 Byte langen Blockkontrollfeld (CF). Entspricht die vorgegebene BUFFER-LENGTH nicht der gespeicherten, so können Daten von dem BS2000-Dateiverwaltungssystem (DVS) mit dem CF überschrieben werden. Die Datei ist damit unbrauchbar.

Bei BLOCK-CONTROL-INFORMATION=NO können aber auch unbrauchbare Daten erzeugt werden, wenn die BUFFER-LENGTH verändert wird (z.B. PLAM-Dateien).

Um die oben genannten Fälle zu vermeiden, gibt LMS eine Warnung aus, wenn sich die angegebene BUFFER-LENGTH-Angabe von der gespeicherten Angabe unterscheidet. LMS versucht jedoch immer, die Datei zu erzeugen.

- Für die Zieldatei ist kein BUFFER-LENGTH-Wert explizit vorgegeben oder bekannt. In diesem Fall wird der Wert aus dem Attributsatz verwendet.  
Ist der Wert n bei STD(SIZE=n) ungerade, erhöht LMS auf (n+1).

### Element ohne Attributsatz

Für Elemente, die keinen Attributsatz besitzen, gilt folgendes:

- Ist für die Zieldatei ein BUFFER-LENGTH-Wert explizit vorgegeben, dann behandelt LMS diese Elemente wie Elemente mit Attributsatz (siehe oben).

Bei der Erzeugung von Phasen führen BUFFER-LENGTH-Angaben ungleich STD(SIZE=1) oder STD(SIZE=2) auf Fehler.

- Ist für die Zieldatei kein BUFFER-LENGTH-Wert explizit vorgegeben oder bekannt, dann gilt:
  - Für Phasen ergibt sich die BUFFER-LENGTH aus der aktuellen Umgebung, d.h. auf NK2-Platten BUFFER-LENGTH=STD(SIZE=1) und auf NK4-Platten STD(SIZE=2). Inhaltlich unterscheiden sich die Phasen nicht.
  - Sonst wird die BUFFER-LENGTH aus der maximalen Satzlänge errechnet.

## Empfehlungen für das Konvertieren von Dateien von K/NK2- auf NK4-Platten

Wenn Dateien von einer K/NK2-Platte über eine Bibliothek auf eine NK4-Platte konvertiert werden sollen, gelten folgende Empfehlungen:

1. „Aktionen“ auf der K/NK2-Platte:

Alle „kritischen“ Elemente der Bibliothek als Datei extrahieren. Kritische Elemente sind „PAM“-Elemente unter Typ X, die als Datei eine der folgenden Eigenschaften besitzen:

- BUFFER-LENGTH = STD(SIZE=n) mit n ungerade
- pamkey-behaftete Phasen
- 2K-orientierte PLAM-Dateien

2. Alle Dateien mit ungerader BUFFER-LENGTH (außer PLAM-Dateien) mit PAMCONV in NK4-Dateien konvertieren.

3. Alle pamkey-behafteten Phasen mit LMSCONV über Typ C in NK4-Phasen konvertieren. Die Konvertierung geht nur auf einer K-Platte.

4. NK2-PLAM-Dateien mit der LMS-Anweisung COPY-LIBRARY in NK4-PLAM-Dateien konvertieren.

5. Die NK4-Dateien mit der LMS-Anweisung ADD-ELEMENT in eine NK4-PLAM-Datei aufnehmen und diese auf die NK4-Platte transferieren.

### *Hinweise*

- Eine NK4-Bibliothek, die sich auf einer Key-Platte befindet, kann beispielsweise mit dem BS2000-Kommando COPY-FILE (siehe [5]) oder den BS2000-Produkten FILE-TRANSFER (siehe [14]) bzw. ARCHIVE (siehe [2]) auf eine NK4-Platte übertragen werden. Die jeweiligen Operanden zur Behandlung der BLOCK-CONTROL-Werte sind den entsprechenden Handbüchern zu entnehmen.
- Eine NK2-PLAM-Datei unter Typ X kann durch EXTRACT-ELEMENT nicht in das NK4-PLAM-Format konvertiert werden.

## 5.8 Behandlung von Aliasnamen (ACS)

Der folgende Abschnitt beschreibt die Auswirkungen von ACS auf LMS. ACS (Alias Catalogue Service) ist ein Subsystem des BS2000/OSD, mit dessen Hilfe Aliasnamen für Dateien verwaltet werden. Ein Aliasname ist ein beliebiger Dateiname, der vom Benutzer anstelle des realen Namens der Datei verwendet werden kann.

### Voraussetzungen

Das ACS-Subsystem muss vom Systemverwalter gestartet sein.

Es muss für die gerade aktuelle Task ein Aliaskatalog angelegt sein, aus dem eindeutig die Zuordnung von Aliasnamen zu realen Dateinamen hervorgeht.

Ist ein taskspezifischer Aliaskatalog vorhanden, wird bei allen Zugriffen zum Dateikatalogeintrag einer Datei der vom Aufrufer der Funktion angegebene Name erstmalig als Aliasname angesehen. Erst wenn der Name im Aliaskatalog der Task nicht gefunden wird, gilt er als realer Dateiname.

Ist der angegebene Name als Aliasname definiert, ersetzt der zugeordnete reale Dateiname den angegebenen Namen.

Bei Aliasnamen, die mit einer Katalog- und einer Benutzerkennung versehen sind, können Probleme auftreten. Nähere Informationen zum Arbeiten mit Aliasnamen und zur Verwaltung des Aliaskatalogs siehe [\[9\]](#).

### Behandlung von Aliasnamen in LMS

LMS ersetzt einen Aliasnamen durch den vollständigen Elementnamen bzw. Dateinamen gemäß dem Katalogeintrag in folgenden Fällen:

#### Aufnahmen von Dateien

Wird eine Datei mit ADD-ELEMENT als Element aufgenommen, verwendet LMS für die Konstruktion des Elementnamens immer den vollständigen Dateinamen.

*Beispiel*

Aliasname	Dateiname
X	FILE.X

Die folgende LMS-Anweisung erzeugt das Element S/FILE.X/001:

```
//ADD-ELEMENT FROM-FILE=X,TO-ELEM=*LIB(,ELEM=*BY-SOURCE(VERSION=001),TYPE=S)
```

### **Ausgeben von Elementen**

Wird ein Element mit EXTRACT-ELEMENT in eine Datei ausgegeben, verwendet LMS für die Konstruktion des Dateinamens immer den vollständigen Elementnamen.

*Beispiel*

Aliasname	Dateiname
X	FILE.X

Die folgende LMS-Anweisung erzeugt die Datei FILE.X:

```
//EXTRACT-ELEMENT ELEM=*LIB(,ELEM=X,TYPE=S),TO-FILE=*BY-SOURCE
```

### **Protokollierung von Dateinamen**

LMS protokolliert immer die vollständig umgesetzten Dateinamen.



## 5.9 Verwendung erweiterter Zeichensätze in LMS (XHCS)

Rechenanlagen (Hosts) und Datensichtstationen arbeiten mit je einem Satz von Buchstaben, Ziffern und Zeichen, aus denen Wörter und andere elementare Bestandteile einer Sprache aufgebaut sind, dem sogenannten **Zeichensatz (character set)**.

Durch die Erweiterung dieser Zeichensätze können landesspezifische Zeichendarstellungen, wie z.B. Umlaute (deutsch) oder Akzente (französisch), innerhalb eines Zeichensatzes gleichzeitig angeboten werden.

Ein **codierter Zeichensatz (coded character set, CCS)** ist die eindeutige Darstellung der Zeichen eines Zeichensatzes in binärer Form. Der Inhalt eines codierten Zeichensatzes und seine Regeln, wie z.B. die Sortierreihenfolge und Konvertierungsvorschriften, sind durch internationale Normen festgelegt.

### *Beispiel*

Das Zeichen „ä“ ist im codierten Zeichensatz EBCDIC.DF.03-DRV (deutsche Referenzversion) durch das Byte X'FB', in EBCDIC.DF.04-1 durch das Byte X'43' dargestellt.

Jeder codierte Zeichensatz (kurz: Code) wird durch seinen eindeutigen Namen (**coded character set name, CCSN**) bestimmt.

### *Beispiel*

Der Code EBCDIC.DF.03-IRV (Internationale Referenzversion) hat den Namen „EDF03IRV“.

Eine Liste der existierenden Codes befindet sich im Anhang von [15].

Im BS2000 werden durch das Softwareprodukt XHCS Zeichensätze bereitgestellt. Standardmäßig gehören dazu:

- 7-Bit-Zeichensätze, wie z.B. ISO646 (Internationaler 7-Bit-Zeichensatz, ASCII), EDF03IRV (Internationale Referenz Version, EBCDIC), EDF03DRV (Deutsche Referenz Version, EBCDIC).
- 8-Bit-Zeichensätze, wie z.B. ISO88591 (Lateinisches Alphabet Nr.1, ASCII), EDF041 (Lateinisches Alphabet Nr.1, EBCDIC), EDF04DRV (Erweiterung von EDF03DRV) usw.
- Die 3 Unicode-Zeichensätze UTF16, UTF8 und UTFE.

### 5.9.1 Hard- und Softwarevoraussetzungen

Sollen die erweiterten Codes für LMS benutzt werden, müssen die nachfolgenden Softwarevoraussetzungen erfüllt sein:

- Softwareprodukt XHCS (optionales Subsystem XHCS-SYS)
- Softwarekomponente VTSU

Zur Erstellung erweiterter Codes im Host und zur Datenübertragung zwischen Host und Datensichtstation wird das Softwareprodukt **XHCS** (**eXtended Host Code Support**) benötigt. In [16] befindet sich eine detaillierte Beschreibung der Prinzipien und Funktionen von XHCS sowie eine Auflistung der Code-Tabellen und der Namen von Standard-Codes.

Als Hardware für die Ein-/Ausgabe erweiterter Zeichensätze auf Datensichtstationen werden sogenannte 8-bit-Terminals benötigt.

Die Prüfung der 8-bit-Fähigkeit von Datensichtstationen erfolgt mit der Softwarekomponente VTSU.

## 5.9.2 LMS-spezifische Anwendung erweiterter Zeichensätze

LMS unterstützt die Verwendung spezieller (nationaler) Zeichensätze. Dabei kann jedem Element ein Zeichensatz durch Zuweisung eines Coded-Character-Set-Namens (CCSN) zugeordnet werden. Dieser wird an Schnittstellen weitergegeben und bei Ausgaben berücksichtigt.

Wird XHCS an der entsprechenden Schnittstelle nicht angeboten, so wird stets der Default "kein Code" verwendet.

Um in den Anweisungen MODIFY-ELEMENT-ATTRIBUTES und SHOW-ELEMENT-ATTRIBUTES auch die Auswahl über Wildcards zu ermöglichen, wird die Überlagerung von nam durch `<filename 1..20 without-cat-id-user-id-generation-version with-wildcards>` angeboten.

LMS selbst setzt keinen bestimmten Zeichensatz voraus und wertet auch nicht die Default-Einstellung der Benutzererkennung aus. Interne LMS-Sortiervorgänge, wie z.B. die Elementbezeichnung, erfolgen unabhängig vom eingestellten CCS.

Jedem Element einer PLAM-Bibliothek kann ein Zeichensatz zugeordnet werden. Dabei überträgt LMS immer den CCSN des Quellelements auf das Zielelement. Der CCSN ist ein inhaltsbeschreibendes Attribut und wird im Inhaltsverzeichnis mit ausgegeben.

Der CCSN ergibt sich ab PLAM V3.4A beim Erzeugen von Elementen implizit aus dem CCSN der Bibliotheksdatei, sofern das das Element erzeugende Programm keine eigenen Vorgaben macht.

### CCSN implizit setzen

CCSN werden bei der Elementbearbeitung in folgenden Fällen implizit gesetzt:

- Element mit ADD-ELEMENT aufnehmen  
Wird ein Element mit der LMS-Anweisung ADD-ELEMENT aufgenommen, wird das Katalogattribut CCS für das Element mit übernommen. Der CCSN wird jedoch nicht zusätzlich im Attributsatz (Satzart 164) gespeichert, um Inkonsistenzen zu vermeiden. Das Layout der Satzart 164 ist in [1] beschrieben.  
Wird mit ADD-ELEMENT ein Modul aus der EAM-Datei aufgenommen, erhält dieser das Attribut „kein Code“.
- Element aus SYSDTA übernehmen  
Wird ein Element aus der Systemdatei SYSDTA aufgenommen, wird jeweils der eingestellte Zeichensatz bestimmt. Das Element erhält den Namen des Zeichensatzes als Attribut zugewiesen.

- Element mit EDT bearbeiten  
Der EDT unterstützt erweiterte Zeichensätze. LMS übergibt beim Aufruf den CCSN des jeweiligen Elementes an den EDT und schreibt das Element mit dem im EDT eingestellten Wert zurück. Ist das Eingabeelement nicht angegeben, wird als Eingabe-CCSN der Wert „kein Code“ angenommen.  
  
Mit EDT ab V17.0 ergibt sich der CCSN eines durch EDIT-ELEMENT neu erzeugten Elementes aus dem mit /MODIFY-TERMINAL-OPTIONS eingestellten Wert.  
  
Mit EDT V16.6B kann dieses Verhalten durch den Einsatz des optionalen Reps A0538001 bewirkt werden. Ohne diesen REP hat der CCSN den Wert \*NONE.  
  
Elemente mit ISO Zeichensatz können nicht als ISAM Datei mit EBCDIC ISAM-Schlüssel extrahiert werden.
- Element kopieren  
Beim Kopieren von Elementen mit der LMS-Anweisung COPY-ELEMENT wird dem Zielelement immer der CCSN des Quellelements zugewiesen.
- Element mit WRITE-MODE=\*EXTEND abspeichern  
Wird ein Element mit WRITE-MODE=EXTEND zurückgeschrieben, so prüft LMS den CCSN von Quelle und Ziel. Stimmen diese nicht überein, so wird die Bearbeitung mit der Fehlermeldung LMS0277 abgewiesen.

### CCSN explizit setzen

Der Benutzer setzt einen CCSN explizit in der LMS-Anweisung MODIFY-ELEMENT-ATTRIBUTES. Dabei prüft LMS nicht die Zulässigkeit des angegebenen CCSN im System.

### CCSN protokollieren

Die LMS-Anweisung SHOW-ELEMENT-ATTRIBUTES mit dem Operanden INFORMATION=\*MAXIMUM gibt die Zuordnung von Zeichensätzen zu den Elementen aus. Bei Elementen, die den CCSN „kein Code“ haben, wird dieses Attribut nicht im Inhaltsverzeichnis angezeigt.

#### *Beispiel*

Das Element TEST erhält durch die LMS-Anweisung MODIFY-ELEMENT-ATTRIBUTES den Zeichensatz EDF03IRV zugeordnet und wird unter dem neuen Namen TEST1 abgespeichert. Dann werden die Elementattribute mit SHOW-ELEMENT-ATTRIBUTES ausgegeben.

```
//MOD-ELEM-ATTR (LIB=TESTLIB,ELEM=TEST,TYPE=S), -
//          NEW-ATT=PAR(ELEM=TEST1;CODED-CHAR-SET=EDF03IRV)
OUTPUT LIBRARY= :X:$USERID.TESTLIB
          MODIFY (S)TEST/001(001)2012-06-22 AS (S)TEST1/001(0001)/2012-06-22
//SHOW-ELEM-ATTR (ELEM=TEST1),INFORMATION=*MAXIMUM
```

```

INPUT  LIBRARY= :X:$USERID.TESTLIB
TYPE   = S
NAME   = TEST1
VERSION = 001                                VARIANT = 0001
USER-DATE = 2012-06-22  CRE-DATE = 2012-06-24  MOD-DATE = 2012-06-24
USER-TIME = 13:07:37   CRE-TIME = 14:38:12   MOD-TIME = 14:41:40
STORAGE = FULL        COD-CH-SET = EDF03IRV
STATE   = FREE
ELEM-SIZE = 1
        1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS

```

### CCSN auswerten und weitergeben

Der CCSN wird bei der Ausgabe von Elementen ausgewertet.

- Element in Datei ausgeben  
Die Datei erhält den CCSN des Elementes.
- Element nach SYSOUT ausgeben  
Bei der Ausgabe von Elementsätzen nach SYSOUT (auch in aufbereiteter Form) mit der Anweisung SHOW-ELEMENT wird das CCS des entsprechenden Elementes benutzt.  
  
Ist SYSOUT einer Datei zugewiesen, so muss der Anwender dieser Datei explizit mit dem BS2000-Kommando MODIFY-FILE-ATTRIBUTES den gewünschten Zeichensatz zuweisen.
- Element nach SYSLST ausgeben  
Bei der Ausgabe von Elementsätzen nach SYSLST wird kein CCSN ausgewertet.  
  
Ist SYSLST einer Datei zugewiesen, so muss der Anwender dieser Datei explizit mit dem BS2000-Kommando MODIFY-FILE-ATTRIBUTES den gewünschten Zeichensatz zuweisen.
- Element in Bibliothekselement ausgeben  
Wird die Elementausgabe durch die LMS-Anweisung MODIFY-LOGGING-PARAMETERS in ein Bibliothekselement umgelenkt, erhält dieses Element den CCSN „kein Code“. Diesem Element kann aber mit der LMS-Anweisung MODIFY-ELEMENT-ATTRIBUTES explizit ein Zeichensatz zugewiesen werden.
- Inhaltsverzeichnis ausgeben  
Bei der Ausgabe des Inhaltsverzeichnisses oder sonstiger von LMS erzeugter Elementinformation wird immer der CCSN „kein Code“ angenommen.

## 5.10 Nutzung der LMS-Funktionalität aus dem EDT

Der EDT bietet die Möglichkeit über die @USE-Anweisung eine externe Anweisungsroutine zu definieren, die über ein ebenfalls mit @USE vereinbartes Benutzerfluchtsymbol als Unterprogramm aufgerufen werden kann. Damit können benutzereigene Anweisungen im EDT zugelassen werden.

Um LMS-Anweisungen und somit die Funktionalität von LMS aus dem EDT zu nutzen, muss in der EDT-Anweisungszeile die @USE-Anweisung in folgender Form angegeben werden:

```
@USE COM[MAND]= 'usersymb' (LMSEDT, $.SYSLNK.LMS.034)
```

mit der externen Anweisungsform: *usersymb* LMS-Anweisung

Eine ausführliche Beschreibung der @USE-Anweisung findet sich in [10].

Zu beachten ist, dass die maximale Länge der LMS-Anweisung aus der EDT-Anweisungszeile nur 256 Byte beträgt.

Bei angegebener LMS-Anweisung wird diese vom Kommandoprozessor SDF analysiert und nach Ausführung erfolgt die Rückkehr in die EDT-Anweisungszeile. Unabhängig von der Eingabe werden Kleinbuchstaben vor der Analyse in Großbuchstaben umgesetzt.

### *Hinweise*

- In der EDT-Anweisungszeile werden die LMS-Anweisungen CALL-EDT, EDIT-ELEMENT und BEGIN-MAKE nicht unterstützt.
- Eine möglicherweise vorhandene LMS-Startdatei wird über die externe Anweisungsroutine nicht unterstützt.
- Die Nutzung der LMS-Funktionalität aus dem EDT ist möglich,
  - wenn der EDT als Hauptprogramm aufgerufen wird, oder
  - wenn der EDT im Hauptprogramm LMS über die LMS-Anweisungen CALL-EDT bzw. EDIT-ELEMENT als Unterprogramm aufgerufen wird.
- Ist LMS das Hauptprogramm, wird beim Nutzen der LMS-Funktionalität aus dem EDT kein neuer LMS-Lauf gestartet, d.h. die LMS-Einstellungen bleiben erhalten.
- Läuft EDT im Unicode-Modus, z.B. nach /START-EDTU (ab EDT V17), so muss in der EDT-Anweisung @USE der Entry LMSEDU statt LMSEDT verwendet werden. Verwendet man im Unicode-Modus den Entry LMSEDT, so bekommt ein mit ADD-ELEMENT erzeugtes Element immer den CCSN \*NONE, obwohl im EDT mit CODENAME ein Zeichensatz ungleich \*NONE (z.B. UTF16) eingestellt wurde.

### Verändertes Verhalten der LMS-Anweisungen EXTRACT-ELEMENT, ADD-ELEMENT und END aufgrund der Nutzung der LMS-Funktionalität aus dem EDT

Die Eingabe der LMS-Anweisung erfolgt aus der EDT-Anweisungszeile.

#### – EXTRACT-ELEMENT

```
usersymbEXTRACT-ELEMENT ELEMENT = ... ,
                                TO-FILE = *STD
```

Die Elementsätze des angegebenen Elementes werden in die aktuelle EDT-Arbeitsdatei ans Ende fortgeschrieben.

#### – ADD-ELEMENT

```
usersymbADD-ELEMENT FROM-FILE=*STD ,
                                TO-ELEMENT = ...
```

Die Datensätze werden aus der aktuellen EDT-Arbeitsdatei ins angegebene Element geschrieben.

#### – END

```
usersymbEND
```

##### – EDT wurde als Hauptprogramm aufgerufen:

Das über @USE aufgerufene Unterprogramm LMS wird beendet, d.h. alle noch geöffneten Bibliotheken werden geschlossen. Es erfolgt die Rückkehr in die EDT-Anweisungszeile.

##### – LMS wurde als Hauptprogramm aufgerufen:

Es erfolgt im Dialogbetrieb die Anfrage an den Benutzer:

```
LMS0409 TERMINATE LMS? REPLY (Y=YES, N=NO)
```

Y: Das Hauptprogramm LMS wird beendet.

N: Das Hauptprogramm LMS wird nicht beendet.

### Erweiterung von LMS-Anweisungen

```
MODIFY-LOGGING-PARAMETERS ... ,
                                TEXT-OUTPUT = *EDT (WRITE-MODE=*UNCHANGED)
```

Die LMS-Protokollausgabe wird in die EDT-Arbeitsdatei 9 geschrieben, wobei sich der Operand WRITE-MODE auf den Inhalt der Arbeitsdatei 9 bezieht.

*Beispiel*

```
/START-EDT _____ (1)
```

```
.  
.
```

```
1.00  
2.00  
  
22.00  
23.00  
@USE COM='.'(LMS EDT, $.SYSLNK.LMS.034) 0000.00:001(0) -- (2)  
LTG TAST
```

```
1.00 Elementinhalt von bsp.elem1  
2.00  
  
22.00  
23.00  
.extract-elem (lib=bsp.lib,elem=bsp.elem1,type=s) 0000.00:001(0) -- (3)  
LTG TAST
```

```
1.00 Geaenderter Elementinhalt von bsp.elem1  
2.00  
  
22.00  
23.00  
.add-elem *std,(lib=bsp.lib,elem=bsp.elem2,type=s) 0000.00:001(0) -- (4)  
LTG TAST
```

- (1) Aufruf des EDT
- (2) LMS als externe Anweisungsroutine mit dem Benutzerfluchtsymbol "Punkt" definieren.
- (3) Einlesen des Elementes bsp.elem1 vom Typ=s aus der Bibliothek bsp.lib in die EDT Arbeitsdatei 0.
- (4) Zurückschreiben des Elementes aus der EDT-Arbeitsdatei 0 in die gleiche Bibliothek unter Umbenennen des Elementnamens.



## 5.11 LMS und EDT V17

LMS ruft den EDT V17 standardmäßig im Kompatibilitäts-Modus auf. Damit ist gewährleistet, dass existierende Prozeduren weiterhin unverändert ablauffähig sind. Bei der Bearbeitung von Unicode-Elementen schaltet der EDT automatisch in den Unicode-Modus.

Um Elemente zu editieren, die Sätze länger als 255 Zeichen enthalten, aber keinen Unicode Zeichensatz haben, geben Sie einer der folgenden Anweisungen ein, um den EDT explizit im Unicode-Modus aufzurufen:

```
//CALL-EDT EDT-MODE=*UNICODE
```

oder

```
//EDIT-ELEMENT . . . ,EDT-MODE=*UNICODE
```

Alternativ können Sie den Unicode-Modus auch als Default-Wert für EDT-Aufrufe festlegen:

```
//MODIFY-LMS-DEFAULTS EDT-MODE=*UNICODE
```



---

## 6 Unterstützung des Software-Entwicklungsprozesses

Die folgenden Begriffe werden mit der angegebenen Bedeutung in diesem Kapitel verwendet:

### Programmsystem

Ein Programmsystem ist eine logische Zusammenfassung von Software-Komponenten. Beispiel für ein Programmsystem sind die Komponenten eines Software-Projekts.

### Komponente

Eine Komponente enthält zusammengehörige Daten. Beispiel für eine Komponente ist ein Bibliothekselement, das Text enthält.

### Abhängige Komponente

Eine solche Komponente, die durch maschinelle Verarbeitung aus anderen Komponenten erzeugt wird. Beispiel für eine abhängige Komponente ist ein Übersetzungslisting.

### Source(-Komponente)

Eine solche Komponente, die von keiner anderen Komponente erzeugt wird. Sourcen sind in der Regel die Komponenten, die manuell weiterentwickelt werden. Beispiele für Sourcen sind Quelltexte von Programmen, Makros oder Dokumente.

### Aktualität

Ein Programmsystem ist aktuell, wenn alle Sourcen neueren Datums, aus denen es abgeleitet ist, berücksichtigt sind.

### Abhängigkeit

Eine Komponente ist von denjenigen Komponenten abhängig, aus denen sie abgeleitet ist.

## 6.1 Ausleihverfahren

LMS ermöglicht die kontrollierte Weiterentwicklung von Komponenten durch mehrere Entwickler durch ein Ausleihverfahren. Hierzu ist für eine Bibliothek bzw. einen Typ durch die Anweisung `MODIFY-LIBRARY-ATTRIBUTES` bzw. `MODIFY-TYPE-ATTRIBUTES WRITE-CONTROL` zu aktivieren. Es ist jetzt sichergestellt, dass ein Element nur noch dann geschrieben werden kann, wenn die explizit oder implizit immer gegebene Basisversion vom Benutzer auch ausgeliehen ist. Unter der Konvention `STD-SEQ` ist dem Ausleiher der höchsten Version der gesamte Versionsraum reserviert. Unter der Konvention `STD-TREE` ist dem Ausleiher der höchsten Version eines Zweiges der gesamte Zweig reserviert. Unter der Konvention `MULTI-SEQ` ist dem Ausleiher der höchsten Version eines Unternehmensraumes der gesamte Unternehmensraum reserviert.

### Ausleihen einer Elementversion

Ein Benutzer besorgt sich ein Element durch die Anweisung `PROVIDE-ELEMENT`. Dadurch wird der Zustand des Elementes von `FREE` auf `IN-HOLD` geändert und eine Kopie des Elementes zur Verfügung gestellt. `PROVIDE-ELEMENT` ist für dieses Element jetzt nicht mehr möglich.

Ein Element kann auch durch explizites Setzen des Zustandes durch die Anweisung `MODIFY-ELEMENT-ATTRIBUTES` als ausgeliehen gekennzeichnet werden. Eine Kopie wird in diesem Falle nicht zur Verfügung gestellt. Sie kann später über `COPY-ELEMENT` besorgt werden.

### Zurückgeben eines Elementes

Ein Element wird durch `RETURN-ELEMENT` wieder zurückgegeben. Bei `BASIS=*STD` wird die durch den Benutzer ausgeliehene Elementversion bestimmt. Sind mehrere Versionen durch den Benutzer ausgeliehen, ist eine explizite Angabe der Basis erforderlich. Das Element ist für weitere Ausleihvorgänge wieder verfügbar; der Ausleihvermerk wird gelöscht. Beim Zurückgeben kann ein Kommentartext mitgegeben werden. `HOLDER`, `DATE` und `TIME` zum Rückgabezeitpunkt werden bei aktivem `WRITE-CONTROL` automatisch festgehalten. Man kann sich diese Einträge mit `EDIT-ELEMENT` ansehen.

### Aufgabe einer Reservierung

Durch `MODIFY-ELEMENT-ATTRIBUTES` kann die Reservierung für ein ausgeliehenes Element aufgegeben werden, etwa wenn auf ein Zurückgeben verzichtet werden soll.

### Vergabe des Ausleihrechtes

Der Bibliothekseigentümer kann mit MODIFY-ELEMENT-PROTECTION für jedes Element festlegen, wer Ausleihrecht hat. Nur die so festgelegten Berechtigten können das betreffende Element ausleihen.

## 6.2 make-Funktionalität

Die make-Funktionalität dient der komfortablen Integration von Programmsystemen. Als Einsatzgebiet für die make-Funktionalität sind kleine bis mittlere Software-Projekte geplant, die ohne großen Verwaltungsaufwand für die Integration auskommen wollen.

Ein Programmsystem besteht aus Komponenten, wie z.B. Makros, Includes, Quellprogramme, Bindemodule, Großmodule, Phasen usw. Eine Komponente heißt *abhängig*, wenn sie aus anderen Komponenten erzeugt wird.

Bei Änderungen im Programmsystem müssen die abhängigen Komponenten neu erzeugt werden, um die Aktualisierung des Programmsystems zu gewährleisten. Häufig betrifft die Änderung mehrere Stufen.

### *Beispiel*

Ein Bindemodul wird aus einem Quellprogramm und den verwendeten Makros erzeugt, ein Großmodul aus einer Reihe von Bindemodulen.

1. Wird ein Quellprogramm geändert, muss daraus der entsprechende Bindemodul neu erzeugt werden (Übersetzung).
2. Anschließend muss der Großmodul, der diesen Bindemodul enthält, neu erzeugt werden.
3. Danach muss die Phase, die diesen Bindemodul enthält, neu erzeugt werden.

Mit der make-Funktionalität kann ein Programmsystem so integriert werden, dass nur die Komponenten des Programmsystems neu erzeugt werden, die von einer Änderung betroffen sind.

Für den make-Lauf sind folgende Voraussetzungen zu überlegen:

- Welche Komponente ist von welcher abhängig?
- Nach welchen Regeln werden die abhängigen Komponenten erzeugt?

Diese Informationen werden in den make-Subanweisungen (siehe nächste Seite) abgelegt. Dort werden auch die Regeln, im Folgenden Aktionen genannt, vereinbart (siehe [Seite 137](#)).

Zulässige Komponenten für die make-Funktionalität sind Bibliothekselemente und Dateien. Eine abhängige Komponente muss neu erzeugt werden, wenn sie älter ist als eine der Komponenten, von denen sie abhängt. Dabei gilt der Zeitpunkt der letzten Änderung einer Komponente:

- bei Bibliothekselementen gilt MODIFICATION-DATE/-TIME als Änderungszeitpunkt,
- bei Dateien gilt LAST-CHANGE-DATE als Änderungszeitpunkt.

Der Änderungszeitpunkt liegt in der ortsüblichen gesetzlichen Landeszeit vor.

### Starten von make

Die make-Funktionalität wird mit der Anweisung BEGIN-MAKE angestoßen. Dort wird die Zielkomponente (TARGET) festgelegt und eine Prozedur erzeugt, die die Folgeverarbeitung regelt. Nach BEGIN-MAKE werden make-Subanweisungen erwartet.

### make-Subanweisungen: Abhängigkeiten beschreiben und Aktionen definieren

Die Subanweisungen legen die Abhängigkeiten unter den Komponenten fest und definieren Aktionen. Die Aktionen beschreiben, wie aus den Ausgangskomponenten eine Zielkomponente erzeugt wird. Sie werden in der bei BEGIN-MAKE angegebenen Prozedur abgelegt.

Folgende Subanweisungen für make stehen zur Verfügung:

MODIFY-MAKE-DEFAULTS	ändert make-lokale Voreinstellungen.
SET-STD-ACTION	legt Abhängigkeiten zwischen einzelnen Objekten fest, indem Standard-Aktionen definiert werden, die aus Komponenten eines bestimmten Elementtyps Komponenten eines anderen Elementtyps erzeugen.
SET-DEPENDENCY	legt die Abhängigkeiten zwischen den einzelnen Komponenten fest. Zudem können Nicht-Standard-Aktionen definiert werden, die später ausgeführt werden sollen, wenn die Zielkomponente nicht aktuell ist. Größere Mengen von Objekten können in SET-DEPENDENCY über S-Variablen eingelesen werden.
SET-PREPROCESSING	definiert die Aktionen, die am Anfang der erzeugten Prozedur stehen sollen.
SET-POSTPROCESSING	definiert die Aktionen, die am Ende der erzeugten Prozedur stehen sollen.

## Beenden von make

Die make-Subanweisung END-MAKE beendet die Folge der Subanweisungen und veranlasst die Folgeverarbeitung. Die Folgeverarbeitung generiert üblicherweise eine Prozedur und führt sie aus.

## Schematische Übersicht über die make-Verarbeitung

Es gibt zwei Wege, ein Programmsystem zu „aktualisieren“:

- durch Neu-Erzeugen der durch die Änderung veralteten Komponenten
- durch „touch“, d.h. bei den veralteten Komponenten wird nur die Änderungszeit aktualisiert (siehe [Seite 140](#)).

[Bild 9](#) zeigt eine schematische Übersicht über die make-Verarbeitung.

Es sei ein Programmsystem mit seinen Komponenten gegeben, in dem die Zielkomponente Z1 aus den Ausgangskomponenten A1 und A2 erzeugt wird.

Wird die Ausgangskomponente A1 verändert, muss die davon abhängige Komponente (Zielkomponente Z1) neu erzeugt werden. Der make-Lauf muss also mit der Zielkomponente Z1 aufgerufen werden. Durch den make-Lauf wird zum einem beschrieben, welche Komponenten von der Ausgangskomponente A1 abhängen und zweitens, welche Aktionen zur Erzeugung der abhängigen Komponente notwendig sind.

Nach Beendigung des make-Laufs wird die erzeugte Prozedur ausgeführt, die die Aktionen enthält. Nachdem die abhängige Komponente Z1 einen älteren Änderungszeitpunkt als die Komponente A1 besitzt, wird sie gemäß den in der Prozedur enthaltenen Aktionen aktualisiert.

In [Bild 9](#) bedeuten:

- (1) Die Komponente Z1 ist abhängig von den Komponenten A1 und A2. Die Komponente A1 wird verändert. Der make-Lauf wird in LMS mit der Zielkomponente Z1 aufgerufen.
- (2) In den make-Subanweisungen während des make-Laufs (BEGIN-MAKE bis END-MAKE) wird festgelegt, dass die Komponente Z1 eine abhängige Komponente von A1 ist. Zudem werden Aktionen vereinbart, wie Z1 aus A1 erzeugt wird. Die zur Aktualisierung des Programmsystems notwendigen Aktionen werden in der bei BEGIN-MAKE definierten Prozedur gesammelt.
- (3) Die Prozedur wird nach Ende des make-Laufs (nach der Eingabe der Anweisung END-MAKE) ausgeführt.
- (4) Durch „touch“ werden die Änderungszeitpunkte der Komponenten auf den neuesten Stand gebracht, ohne deren Inhalt zu ändern.
- (5) Das aktualisierte Programmsystem mit der neu erzeugten Komponente Z1.

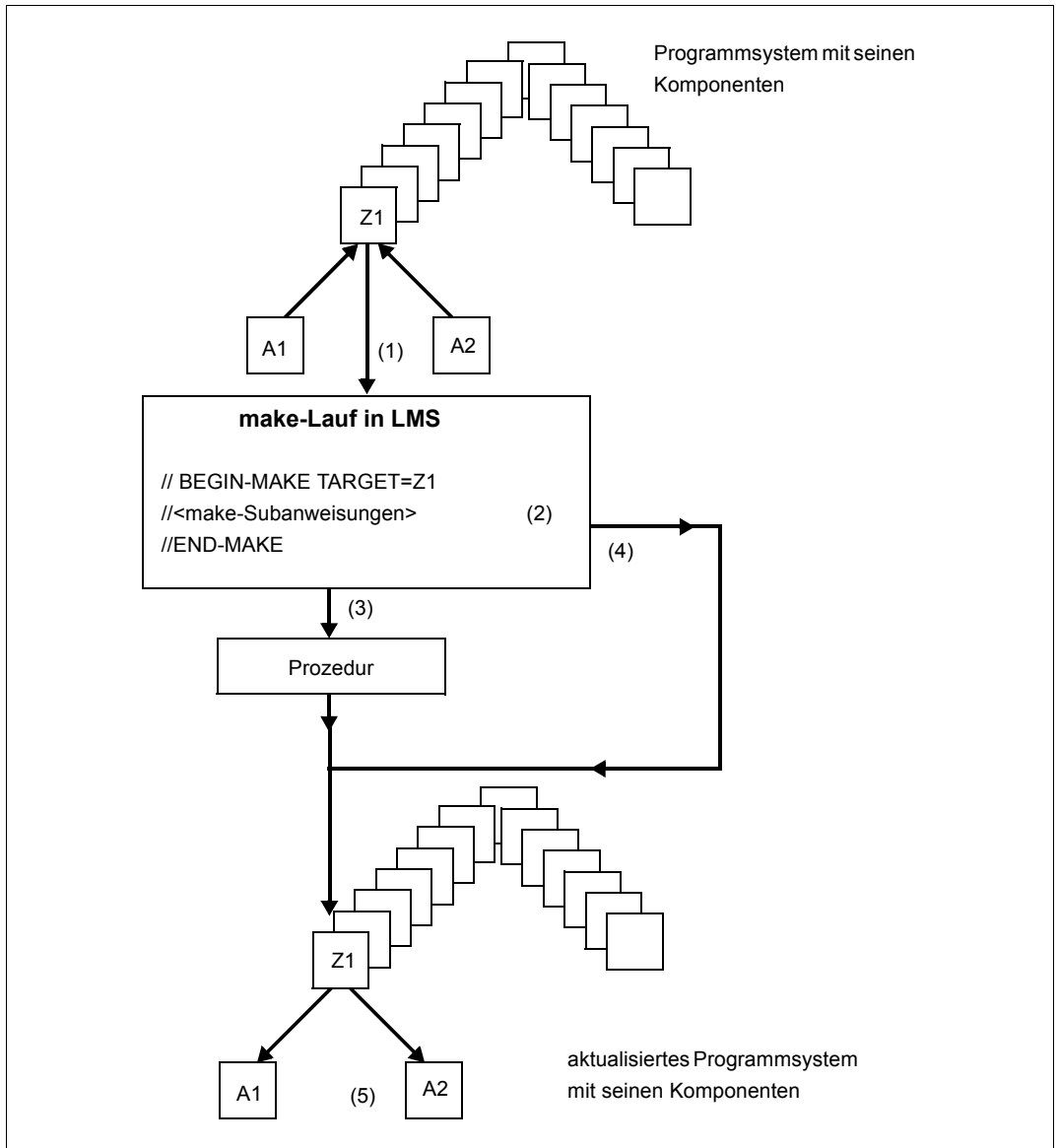


Bild 9: Übersicht über die make-Verarbeitung



## 6.2.1 Aktionen

Aktionen beschreiben, wie aus den Ausgangskomponenten eine Zielkomponente erzeugt wird.

Die Aktionen sind Textzeilen, die bei den make-Subanweisungen SET-DEPENDENCY oder SET-STD-ACTION angegeben werden. Diese Textzeilen werden unverändert in die bei BEGIN-MAKE angegebene Prozedur übernommen.

Jede Aktion einer Aktionsliste wird in der Prozedur eine eigene Zeile. Damit ist möglich, für eine Komponente mehrere Zeilen (Aktionen) anzugeben. Der Ersteller der make-Anweisungen muss selbst auf Fortsetzungszeilen innerhalb von Aktionen achten.

In der Subanweisung SET-STD-ACTION können Standard-Aktionen angegeben werden, mit denen aus Ausgangskomponenten eines bestimmten Typs Zielkomponenten eines anderen Typs erzeugt werden können.

Auf diese Standard-Aktionen kann in der SET-DEPENDENCY-Anweisung Bezug genommen werden.

## 6.2.2 Verwendung von Variablen

Die volle make-Funktionalität benötigt S-Variablen und damit SDF-P (siehe [12]).

In den Aktionen können von make erzeugte S-Variablen verwendet werden (make-S-Variablen). Sie enthalten Angaben über die zu erzeugenden abhängigen Komponenten und die Komponenten, aus denen sie erzeugt werden. Die benötigten Zuweisungen an make-S-Variablen werden von make vor den Aktionen in die erzeugte Prozedur eingefügt.

Während des make-Laufs werden alle in den make-Subanweisungen angegebenen S-Variablen ersetzt. Dies wird bereits durch die SDF-Eingabesprache ermöglicht.

Ein Ersetzen von S-Variablen während des make-Laufs kann bei der Angabe der Aktionen durch Verdoppelung des „&“ verhindert werden.

Große Datenmengen für Komponenten können in SET-DEPENDENCY über S-Variablen eingelesen werden.

Es gibt zwei Kategorien von Variablen, die erst zum Ablaufzeitpunkt der erzeugten Prozedur generiert werden sollen:

- normale S-Variablen, die vom Anwender (in den Aktionen) versorgt und benutzt werden,
- make-S-Variablen, die von make versorgt und in den Aktionen benutzt werden.

Die Namen der make-S-Variablen werden in der Subanweisung MODIFY-MAKE-DEFAULTS definiert. Wenn in Aktionen auf make-S-Variablen zugegriffen werden soll, müssen die Aktionen über SDF verarbeitet werden.

Folgende make-S-Variablen können definiert werden:

- Name des aktuellen Zieles (CURRENT-TARGET-VAR, siehe make-Subanweisung MODIFY-MAKE-DEFAULTS)
- Liste aller vorausgesetzten Komponenten (FROM-OBJECTS-VAR) (siehe make-Subanweisung MODIFY-MAKE-DEFAULTS)
- Liste aller vorausgesetzten Komponenten, die neuer sind als das Ziel (MODIFIED-OBJECTS-VAR) (siehe make-Subanweisung MODIFY-MAKE-DEFAULTS).

Der Wert zu CURRENT-TARGET-VAR ist eine Variable im Ausgabeformat von LMS (siehe Format der Ausgabe in S-Variable, [Seite 455](#)). Die Werte zu FROM-OBJECTS-VAR und MODIFIED-OBJECTS-VAR sind Listen solcher Variablen. Die Reihenfolge in der Listenvariablen entspricht der Reihenfolge, die in der Anweisung SET-DEPENDENCY angegeben wurde.

Für Bibliothekselemente werden folgende Strukturelemente versorgt (Variablenname VAR):

&(VAR.LIB)	STRING
&(VAR.ELEM)	STRING
&(VAR.VERSION)	STRING
&(VAR.TYPE)	STRING

Wird die S-Variable mit einem Wert für eine Datei versorgt, erhält die Variable VAR.LIB den vollständigen Dateinamen, die Variablen VAR.ELEM, VAR.VERSION und VAR.TYPE erhalten einen Leerstring. Die Darstellung von \*NONE enthält nur Leerstrings.

Die Versorgung der S-Variable erfolgt durch Einfügen von Kommandos in die erzeugte Prozedur. Dabei wird die Hilfsvariable SYSLMSMAKE verwendet.

Die Deklaration erfolgt ebenfalls durch eingefügte Kommandos in die erzeugte Prozedur. Deklarationen, die im Preprocessing gemacht wurden, werden nicht überschrieben.

### 6.2.3 Auswahl- und Konstruktionsangabe bei make

Die Auswahl- und Konstruktionsangaben führen zur Verarbeitung mehrerer Komponenten.

#### Auswahlangabe bei make

Mit der Auswahlangabe kann auf Zielkomponenten Bezug genommen werden. Muster in der Auswahlangabe können zur Bildung von Ausgangskomponenten verwendet werden (siehe Konstruktionsangaben). Auswahlangaben beziehen sich auf make bekannte Zielkomponenten, die während des make-Laufs nach und nach durch die angegebenen Abhängigkeiten erzeugt werden.

Auswahlangaben wirken nur auf diese Komponenten und **nicht** auf Inhalte von Bibliotheken. Auswahlangaben sind für ELEMENT, TYPE und Bibliotheksname zulässig.

#### Konstruktionsangabe bei make

Mit Konstruktionsangaben können Bezeichnungen von Ausgangskomponenten aus Zielkomponenten gebildet werden. Dabei korrespondieren ELEMENT mit ELEMENT und TYPE mit TYPE. Angegebene Dateinamen korrespondieren untereinander und mit ELEMENT.

Sind mehrere Ziel- und Ausgangskomponenten angegeben, werden Auswahl und Konstruktion zwischen allen Kombinationen durchgeführt. Angegebene Dateinamen sind untereinander und mit ELEMENT korrespondierend.

Es entfällt die sonst in LMS gültige Einschränkung, dass mindestens ein Platzhalter der Auswahlangabe in der Konstruktionsangabe vorkommen muss.

### 6.2.4 Ablaufsteuerung während der Folgeverarbeitung

Die globale Ablaufsteuerung für make erfolgt mit den Anweisungen MODIFY-MAKE-DEFAULTS und BEGIN-MAKE.

Die Ausgabe des Protokolls der Folgeverarbeitung lässt sich mit BS2000-Kommandos steuern. Als erstes Kommando unter ACTION können die SDF-P-Kommandos MODIFY-PROCEDURE-OPTIONS oder MODIFY-PROCEDURE-TEST-OPTIONS angegeben werden (beim SET-PREPROCESSING für die gesamte Prozedur), die die Protokollierung beeinflussen.

Das Verhalten bei Fehlern wird mit dem Operanden SUPPRESS-ERRORS der make-Subanweisung SET-DEPENDENCY gesteuert. Ist SUPPRESS-ERRORS=\*NONE, wird der Lauf der von make erzeugten Prozedur abgebrochen, wenn nach Ausführung der Aktionen aus der Abhängigkeitsdefinition ein Kommando-Returncode gesetzt ist (siehe [12]). Das entspricht dem herkömmlichen Spin-off-Mechanismus. Das Verhalten bei Fehlern lässt sich

auf diese Weise für jede Abhängigkeitsdefinition getrennt steuern.

Fehlersituationen werden zurückgesetzt, wenn STEP-Kommandos in die erzeugte Prozedur eingefügt werden. Daher darf in diesem Fall am Ende der Aktionen kein Programm geladen sein.

Eine differenziertere Fehlerbehandlung ist durch Verwendung der entsprechenden Kommandos innerhalb der Aktionen möglich.

## 6.2.5 TOUCH

Statt der Prozedurerzeugung und -ausführung kann auch ein „touch“ der Komponenten durchgeführt werden (SUCCESS-PROCESSING=\*TOUCH bei der BEGIN-MAKE-Anweisung). Die bearbeiteten Komponenten erhalten einen aktualisierten Änderungszeitpunkt. In diesem Fall wird keine Prozedur erzeugt. Nach END-MAKE sind die Zeitstempel der Komponenten aktualisiert. Der Inhalt der Komponenten wird nicht verändert.

## 6.2.6 Arbeitsweise von make

Um bestimmen zu können, welche Abhängigkeiten insgesamt bestehen, wird zunächst ein Graph konstruiert, der das gesamte Programmsystem beschreibt. Dabei dient die Komponente als Ausgangspunkt des Graphen, die als Zielkomponente (TARGET bei BEGIN-MAKE) angegeben wurde. Der weitere Aufbau des Graphen erfolgt durch die Auswertung der Abhängigkeiten (make-Subanweisung SET-DEPENDENCY).

Abhängigkeiten haben die folgende Struktur:

```
(target-object1, target-object 2),(from-object1, from-object2,...),
(action1,...)
```

Diese Notation bedeutet:

- die Zielkomponenten (target-objects) sind abhängig von den Ausgangskomponenten (from-objects)
- die Aktionen (actions) beschreiben, wie eine Zielkomponente erzeugt wird.

Eine Zielkomponente ist nicht aktuell, wenn eine zugehörige Ausgangskomponente einen neueren Änderungszeitpunkt aufweist, oder die Zielkomponente nicht vorhanden ist.

Die oben dargestellte Abhängigkeit ist äquivalent zu folgender:

```
target-object1, (from-object1, from-object2,...), (action1,...)
target-object2, (from-object1, from-object2,...), (action1,...)
```

Am Anfang besteht der Graph allein aus dem Ziel des make-Laufs. Danach wird mit Hilfe der Abhängigkeiten der Graph nach und nach erweitert. Ist eine Komponente im Graphen Zielkomponente und hat noch keinen Nachfolger, werden die zugehörigen Ausgangskom-

ponenten an diesen Knoten gehängt.

Wenn make keine Abhängigkeiten mehr auf den Graphen anwenden kann, ist der Aufbau des Graphen abgeschlossen. Die Komponenten im Graphen, die keine Ausgangskomponenten haben, sind die Sourcen des Programmsystems. Sie müssen real vorhanden sein, um das Ziel des make-Laufs erzeugen zu können. Für eine Zielkomponente müssen alle Sourcen vorhanden sein (nicht nur die geänderten), sonst wird der make-Lauf mit Fehler abgebrochen.

Für jede Zielkomponente darf nur eine eindeutig bestimmte Aktionsfolge existieren. Andernfalls wird der make-Lauf mit Fehler abgebrochen.

Von den Sourcen ausgehend werden die Beziehungen zwischen Zielkomponenten und Ausgangskomponenten im Hinblick auf den Änderungszeitpunkt ausgewertet. Ist eine Zielkomponente nicht aktuell, wird die entsprechende Aktion veranlasst.

### *Beispiel*

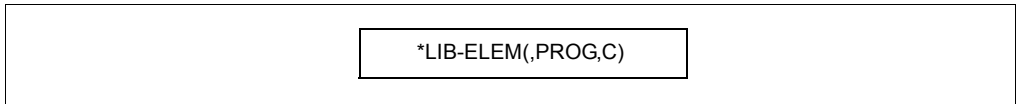
Das nachfolgende Beispiel zeigt die Arbeitsweise der make-Funktionalität unter Verwendung von Wildcards. Die Bibliothek, in der die Komponenten stehen, ist die voreingestellte Bibliothek.

Das Phasenelement PROG (Elementtyp C) wird aus den Objektmodulen TEIL1 und TEIL2 (beide vom Elementtyp R) erzeugt. Diese Objekte werden aus Sourcen vom Typ S mit Suffix „.COB“ übersetzt. Die Aktionen sind nur angedeutet.

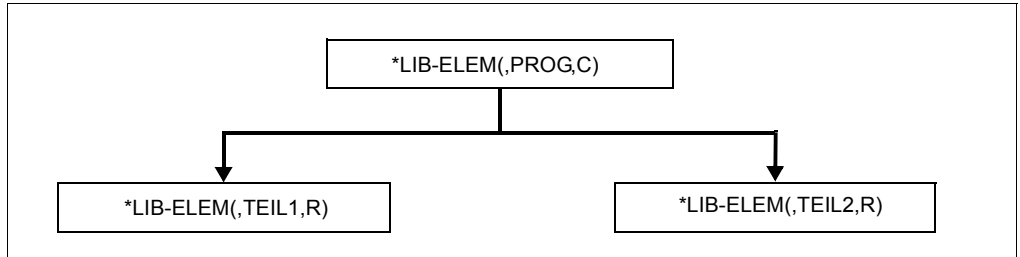
Die Abhängigkeiten (Bibliothekselemente im Format (,NAME,TYPE)):

- (1) TARGET-OBJECT1 = \*LIB-ELEM(,PROG,C), -  
 (FROM-OBJECT1=\*LIB-ELEM(,TEIL1,R), FROM-OBJECT2=\*LIB-ELEM(,TEIL2,R)), -  
 ACTION1 = 'LINK PROG'
- (2) TARGET-OBJECT2 = \*LIB-ELEM(,\*,R) , -  
 FROM-OBJECT = \*LIB-ELEM(,\*\*.COB,S), -  
 ACTION2 = 'COMPILE'

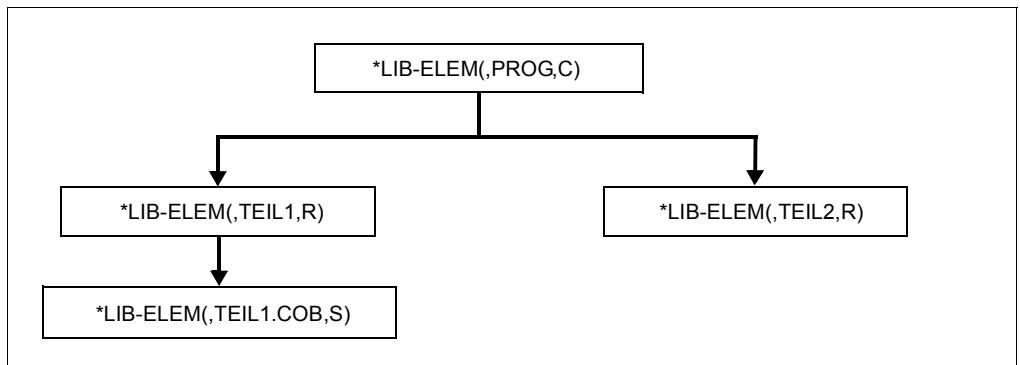
Wenn die Komponente \*LIB-ELEM(,PROG,C) im make-Lauf erzeugt werden soll, ist sie der Anfangsknoten des Graphen..



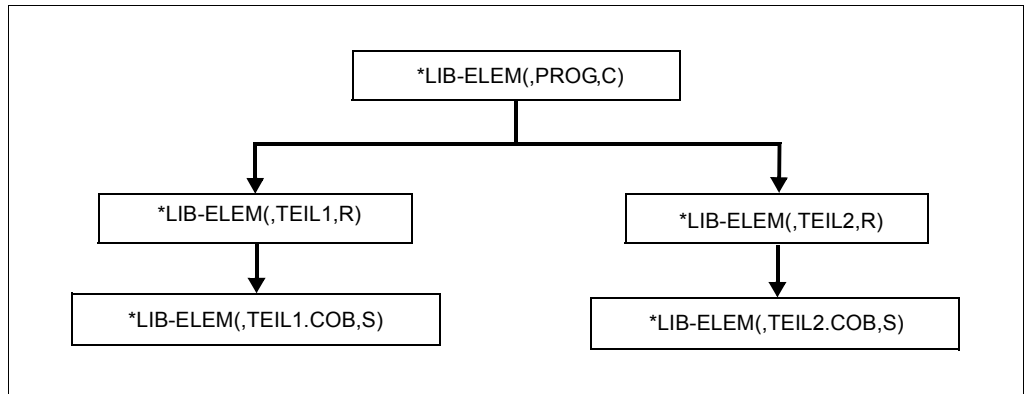
Der Graph kann nur mit der Abhängigkeit (1) erweitert werden. Das Ergebnis:



Die Zielkomponente der Abhängigkeit (2) passt als make-Auswahlangebe auf \*LIB-ELEM(,TEIL1,R), die zugehörige Ausgangskomponente liefert als Ergebnis der Konstruktion \*LIB-ELEM(,TEIL1.COB,S), das im Graphen an \*LIB-ELEM(,TEIL1,R) gehängt wird.



In gleicher Weise wird (,TEIL2.COB,S) an (,TEIL2,R) gehängt.



\*LIB-ELEM(,TEIL1.COB,S) und \*LIB-ELEM(,TEIL2.COB,S) sind die Quellen, die zur Erzeugung von \*LIB-ELEM(,PROG,C) nötig sind; sie müssen real vorhanden sein.





---

## 7 Anweisungen

Dieses Kapitel beschreibt die Anweisungen, die während des LMS-Laufs eingegeben werden können. Nach der Übersicht der SDF-Standardanweisungen und der Syntaxbeschreibung der SDF-Benutzeroberfläche folgt die Beschreibung der LMS-Anweisungen. Um Eindeutigkeit zu gewährleisten, empfiehlt es sich, in Prozeduren vor allen Anweisungen „/ /“ zu schreiben.

### 7.1 SDF-Standardanweisungen für LMS

Die folgenden SDF-Standardanweisungen dürfen während des LMS-Laufs angegeben werden.

Anweisung	Funktion
END	Beenden des LMS-Laufs
EXECUTE-SYSTEM-CMD	Systemkommando ausführen
HELP-MSG-INFORMATION	Text einer Systemmeldung auf SYSOUT ausgeben
HOLD-PROGRAM	Unterbrechen eines Programms
MODIFY-SDF-OPTIONS	Ändern der SDF-Einstellungen
REMARK	Kommentieren von Programmen
RESET-INPUT-DEFAULTS	Taskspezifische Default-Werte zurücksetzen
RESTORE-SDF-INPUT	Wiederanzeigen der letzten Eingabe
SHOW-INPUT-DEFAULTS	Taskspezifische Default-Werte anzeigen
SHOW-INPUT-HISTORY	Inhalt des Eingabepuffers ausgeben
SHOW-SDF-OPTIONS	Anzeigen der SDF-Einstellungen
SHOW-STMT	Syntax einer Anweisung anzeigen
STEP	Definieren eines Wiederaufsetzpunktes
WRITE-TEXT	Text nach SYSOUT ausgeben

Die SDF-Standardanweisungen (bis auf END) werden nicht in diesem Handbuch beschrieben. Eine Beschreibung findet sich in [3].

## 7.2 Syntaxbeschreibung

Diese Syntaxbeschreibung basiert auf der SDF-Version 4.7. Die Syntax der SDF-Kommando-/Anweisungssprache wird im Folgenden in drei Tabellen erklärt.

Zu [Tabelle 2: Metasyntax](#)

In den Kommando-/Anweisungsformaten werden bestimmte Zeichen und Darstellungsformen verwendet, deren Bedeutung in [Tabelle 2](#) erläutert wird.

Zu [Tabelle 3: 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 3](#) beschriebenen Datentypen.

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

Zu [Tabelle 4: 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:

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

Für den Datentyp integer enthält [Tabelle 4](#) 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 4](#) 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 4](#) erläutert.

### Metasyntax

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter (Kommando-, Anweisungs-, Operandennamen, Schlüsselwortwerte) und konstante Operandenwerte. Schlüsselwortwerte beginnen mit *.	<b>HELP-SDF</b>  <b>SCREEN-STEPS = *NO</b>
<b>GROSSBUCHSTABEN in Halbfett</b>	Großbuchstaben in Halbfett kennzeichnen garantierte bzw. vorgeschlagene Abkürzungen der Schlüsselwörter.	<b>GUIDANCE-MODE = *YES</b>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörigen Operandenwerten.	<b>GUIDANCE-MODE = *NO</b>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird (siehe Tabellen <a href="#">3</a> und <a href="#">4</a> ).	<b>SYNTAX-FILE = &lt;filename 1..54&gt;</b>
<u>Unterstreichung</u>	Der Unterstrich kennzeichnet den Default-Wert eines Operanden.	<b>GUIDANCE-MODE = *NO</b>
/	Der Schrägstrich trennt alternative Operandenwerte.	<b>NEXT-FIELD = *NO / *YES</b>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	<b>,UNGUIDED-DIALOG = *YES (...)/ *NO</b>

Tabelle 2: Metasyntax (Teil 1 von 2)

Kennzeichnung	Bedeutung	Beispiele
<p>[ ]</p> <p>Eintrückung</p> <p> </p> <p>,</p> <p>list-poss(n):</p>	<p>Eckige Klammern kennzeichnen struktureinleitende Operandenwerte, deren Angabe optional ist. Die nachfolgende Struktur kann ohne den einleitenden Operandenwert angegeben werden.</p> <p>Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.</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 Strukturtiefe.</p> <p>Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.</p> <p>Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muss sie in runde Klammern eingeschlossen werden.</p>	<p><b>SELECT = [*BY-ATTRIBUTES](...)</b></p> <p><b>,GUIDED-DIALOG = *YES (...)</b></p> <p><b>*YES(...)</b>   <b>SCREEN-STEPS = *NO / *YES</b></p> <p><b>SUPPORT = *TAPE(...)</b></p> <p><b>*TAPE(...)</b>   <b>VOLUME = *ANY(...)</b>   <b>*ANY(...)</b>   ...</p> <p><b>GUIDANCE-MODE = *NO / *YES</b></p> <p><b>,SDF-COMMANDS = *NO / *YES</b></p> <p>list-poss: <b>*SAM / *ISAM</b></p> <p>list-poss(40): &lt;structured-name 1..30&gt;</p> <p>list-poss(256): <b>*OMF / *SYSLST(...)</b> / &lt;filename 1..54&gt;</p>
<p>Kurzname:</p>	<p>Der darauf folgende Name ist ein garantierter Aliasname des Kommando- bzw. Anweisungsnamens.</p>	<p><b>HELP-SDF</b>      Kurzname: <b>HPSDF</b></p>

Tabelle 2: Metasyntax (Teil 2 von 2)

## Datentypen

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

Tabelle 3: Datentypen (Teil 1 von 6)

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

Tabelle 3: Datentypen (Teil 2 von 6)

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

Tabelle 3: Datentypen (Teil 3 von 6)

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

Tabelle 3: Datentypen (Teil 4 von 6)



Datentyp	Zeichenvorrat	Besonderheiten
product-version	A...Z 0...9 Punkt Hochkomma	<p>Eingabeformat: <code>[[C]][V][m]m.naso[']</code></p> <div style="text-align: right; margin-right: 20px;"> <math display="block">\begin{array}{c}   \\   \\ \text{Korrekturstand} \\ \text{Freigabestand} \end{array}</math> </div> <p>wobei m, n, s und o jeweils eine Ziffer und a ein Buchstabe ist. Ob Freigabe- und/oder Korrekturstand angegeben werden dürfen oder ob sie angegeben werden müssen, bestimmen Zusätze zu dem Datentyp (siehe <a href="#">Tabelle 4</a>, Zusätze without-corr, without-man, mandatory-man und mandatory-corr). product-version kann in Hochkommata eingeschlossen werden, wobei der Buchstabe C vorangestellt werden kann. Die Versionsangabe kann mit dem Buchstaben V beginnen.</p>
structured-name	A...Z 0...9 \$, #, @ Bindestrich	<p>alphanumerische Zeichenfolge, die in mehrere durch Bindestrich getrennte Teilzeichenfolgen gegliedert sein kann; erstes Zeichen: A...Z oder \$, #, @</p>
text	beliebig	Das Eingabeformat ist den jeweiligen Operandenbeschreibungen zu entnehmen.
time	0...9 Strukturkennzeichen: Doppelpunkt	<p>Angabe einer Tageszeit</p> <p>Eingabeformat: <math>\left. \begin{array}{l} \text{hh:mm:ss} \\ \text{hh:mm} \\ \text{hh} \end{array} \right\}</math></p> <p>hh: Stunden mm: Minuten ss: Sekunden } führende Nullen können 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 3: Datentypen (Teil 5 von 6)

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

Tabelle 3: Datentypen (Teil 6 von 6)

## Zusätze zu Datentypen

Zusatz	Bedeutung												
<i>x..y unit</i>	<p>beim Datentyp integer: Intervallangabe</p> <p><i>x</i>     Mindestwert, der für integer erlaubt ist. <i>x</i> ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>y</i>     Maximalwert, der für integer erlaubt ist. <i>y</i> ist eine ganze Zahl, die mit einem Vorzeichen versehen werden darf.</p> <p><i>unit</i>    Dimension. Folgende Angaben werden verwendet:</p> <table> <tr> <td><i>days</i></td> <td><i>byte</i></td> </tr> <tr> <td><i>hours</i></td> <td><i>2Kbyte</i></td> </tr> <tr> <td><i>minutes</i></td> <td><i>4Kbyte</i></td> </tr> <tr> <td><i>seconds</i></td> <td><i>Mbyte</i></td> </tr> <tr> <td><i>milliseconds</i></td> <td></td> </tr> </table>	<i>days</i>	<i>byte</i>	<i>hours</i>	<i>2Kbyte</i>	<i>minutes</i>	<i>4Kbyte</i>	<i>seconds</i>	<i>Mbyte</i>	<i>milliseconds</i>			
<i>days</i>	<i>byte</i>												
<i>hours</i>	<i>2Kbyte</i>												
<i>minutes</i>	<i>4Kbyte</i>												
<i>seconds</i>	<i>Mbyte</i>												
<i>milliseconds</i>													
<i>x..y special</i>	<p>bei den übrigen Datentypen: Längenangabe</p> <p>Bei den Datentypen <i>date</i>, <i>device</i>, <i>product-version</i> und <i>time</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> <table> <tr> <td><i>arithm-expr</i></td> <td>arithmetischer Ausdruck (SDF-P)</td> </tr> <tr> <td><i>bool-expr</i></td> <td>logischer Ausdruck (SDF-P)</td> </tr> <tr> <td><i>string-expr</i></td> <td>String-Ausdruck (SDF-P)</td> </tr> <tr> <td><i>expr</i></td> <td>beliebiger Ausdruck (SDF-P)</td> </tr> <tr> <td><i>cond-expr</i></td> <td>bedingter Ausdruck (JV)</td> </tr> <tr> <td><i>symbol</i></td> <td>CSECT- oder Entry-Name (BLS)</td> </tr> </table>	<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)	<i>symbol</i>	CSECT- oder Entry-Name (BLS)
<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)												
<i>symbol</i>	CSECT- oder Entry-Name (BLS)												
<i>with</i>	Erweitert die Angabemöglichkeiten für einen Datentyp.												
<i>-compl</i>	Bei Angaben zu dem Datentyp <i>date</i> ergänzt SDF zweistellige Jahresangaben der Form <i>jj-mm-tt</i> zu: <table> <tr> <td><i>20jj-mm-tt</i></td> <td>falls <i>jj</i> &lt; 60</td> </tr> <tr> <td><i>19jj-mm-tt</i></td> <td>falls <i>jj</i> ≥ 60</td> </tr> </table>	<i>20jj-mm-tt</i>	falls <i>jj</i> < 60	<i>19jj-mm-tt</i>	falls <i>jj</i> ≥ 60								
<i>20jj-mm-tt</i>	falls <i>jj</i> < 60												
<i>19jj-mm-tt</i>	falls <i>jj</i> ≥ 60												
<i>-low</i>	Groß- und Kleinschreibung wird unterschieden.												
<i>-path-compl</i>	Bei Angaben zu dem Datentyp <i>filename</i> ergänzt SDF die Katalog- und/oder die Benutzerkennung, falls diese nicht angegeben werden.												
<i>-under</i>	Erlaubt Unterstriche ' <i>_</i> ' bei den Datentypen <i>name</i> und <i>composed-name</i> .												

Tabelle 4: Zusätze zu Datentypen (Teil 1 von 7)

Zusatz	Bedeutung
with (Forts.) -wild(n)	<p>Teile eines Namens dürfen durch die folgenden Platzhalter ersetzt werden. n bezeichnet die maximale Eingabelänge bei Verwendung von Platzhaltern. Mit Einführung der Datentypen posix-filename und posix-pathname akzeptiert SDF neben den bisher im BS2000 üblichen Platzhaltern auch Platzhalter aus der UNIX-Welt (nachfolgend POSIX-Platzhalter genannt). Da derzeit nicht alle Kommandos POSIX-Platzhalter unterstützen, kann ihre Verwendung bei Datentypen ungleich posix-filename und posix-pathname zu Semantikfehlern führen.</p> <p>Innerhalb einer Musterzeichenfolge sollten entweder nur BS2000- oder nur POSIX-Platzhalter verwendet werden. Bei den Datentypen posix-filename und posix-pathname sind nur POSIX-Platzhalter erlaubt. Ist eine Musterzeichenfolge mehrdeutig auf einen String abbildbar, gilt der erste Treffer.</p>
BS2000-Platzhalter	Bedeutung
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.
Punkt am Ende	Teilqualifizierte Angabe eines Namens. Entspricht implizit der Zeichenfolge „/*“, d.h. nach dem Punkt folgt mindestens ein beliebiges Zeichen.
/	Ersetzt genau ein beliebiges Zeichen.
<s <sub>x</sub> :s <sub>y</sub> >	<p>Ersetzt eine Zeichenfolge, für die gilt,</p> <p>a) wenn s<sub>x</sub> kürzer oder genauso lang wie s<sub>y</sub> ist:</p> <ul style="list-style-type: none"> <li>– Sie ist mindestens so lang wie s<sub>x</sub> und höchstens so lang wie s<sub>y</sub></li> <li>– Sie liegt in der alphabetischen Sortierung im Bereich von s<sub>x</sub> bis s<sub>y</sub></li> <li>– s<sub>x</sub> darf leer sein (=1 Zeichen mit niedrigster Codierung)</li> </ul> <p>b) wenn s<sub>x</sub> länger als s<sub>y</sub> ist:</p> <ul style="list-style-type: none"> <li>– Sie ist mindestens so lang wie s<sub>y</sub> und höchstens so lang wie s<sub>x</sub></li> <li>– Sie liegt in der alphabetischen Sortierung im Bereich von s<sub>x</sub> bis zu einer Zeichenkette, die mit s<sub>y</sub> beginnt und mit Zeichen der höchsten möglichen Codierung auf die Länge von s<sub>x</sub> aufgefüllt ist.</li> <li>– s<sub>y</sub> darf leer sein (=1 Zeichen mit höchster Codierung)</li> </ul>

Tabelle 4: Zusätze zu Datentypen (Teil 2 von 7)

Zusatz	Bedeutung	
with-wild(n) (Forts.)	<s <sub>1</sub> ,...>  -s	Ersetzt alle Zeichenfolgen, auf die eine der mit s angegebenen Zeichenkombinationen zutrifft. s kann auch die leere Zeichenfolge sein. Jede Zeichenfolge s kann auch eine Bereichsangabe „s <sub>x</sub> :s <sub>y</sub> “ sein (siehe <a href="#">Seite 156</a> ).  Ersetzt alle Zeichenfolgen, die der angegebenen Zeichenfolge s nicht entsprechen. Das Minuszeichen darf nur am Beginn der Zeichenfolge stehen. Innerhalb der Datentypen filename bzw. partial-filename kann die negierte Zeichenfolge -s genau einmal verwendet werden, d.h., -s kann einen der drei Namensteile cat, user oder datei ersetzen.
	Platzhalter sind in Generations- und Versionsangaben von Dateinamen nicht erlaubt. In Benutzerkennungen ist die Angabe von Platzhaltern der Systemverwaltung vorbehalten. Platzhalter können nicht die Begrenzer der Namensteile cat (Doppelpunkte) und user (\$ und Punkt) ersetzen.	
POSIX- Platzhalter	Bedeutung	
*	Ersetzt eine beliebige, auch leere Zeichenfolge. Ein * an erster Stelle muss verdoppelt werden, sofern dem * weitere Zeichen folgen und die eingegebene Zeichenfolge nicht mindestens einen weiteren Platzhalter enthält.	
?	Ersetzt genau ein beliebiges Zeichen. Ist als erstes Zeichen außerhalb von Hochkommata nicht zulässig.	
[c <sub>x</sub> -c <sub>y</sub> ]	Ersetzt genau ein Zeichen aus dem Bereich c <sub>x</sub> und c <sub>y</sub> einschließlich der Bereichsgrenzen. c <sub>x</sub> und c <sub>y</sub> müssen einfache Zeichen sein.	
[s]	Ersetzt genau ein Zeichen aus der Zeichenfolge s. Die Ausdrücke [c <sub>x</sub> -c <sub>y</sub> ] und [s] können kombiniert werden zu [s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	
[!c <sub>x</sub> -c <sub>y</sub> ]	Ersetzt genau ein Zeichen, das nicht in dem Bereich c <sub>x</sub> und c <sub>y</sub> einschließlich der Bereichsgrenzen enthalten ist. c <sub>x</sub> und c <sub>y</sub> müssen einfache Zeichen sein. Die Ausdrücke [!c <sub>x</sub> -c <sub>y</sub> ] und [!s] können kombiniert werden zu [!s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	
[!s]	Ersetzt genau ein Zeichen, das nicht in der Zeichenfolge s enthalten ist. Die Ausdrücke [!s] und [!c <sub>x</sub> -c <sub>y</sub> ] können kombiniert werden zu [!s <sub>1</sub> c <sub>x</sub> -c <sub>y</sub> s <sub>2</sub> ]	

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

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

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

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

Tabelle 4: Zusätze zu Datentypen (Teil 5 von 7)

Zusatz	Bedeutung																				
with-wild-constr(n) (Forts.)	Beispiele:																				
	<table border="1"> <thead> <tr> <th>Auswahlmuster</th> <th>Auswahl</th> <th>Konstruktionsmuster</th> <th>neuer Name</th> </tr> </thead> <tbody> <tr> <td>A/*</td> <td>AB1 AB2 A.B.C</td> <td>D&lt;3&gt;&lt;2&gt;</td> <td>D1 D2 D.CB</td> </tr> <tr> <td>C.&lt;A:C&gt;/&lt;D,F&gt;</td> <td>C.AAD C.ABD C.BAF C.BBF</td> <td>G.&lt;1&gt;.&lt;3&gt;.XY&lt;2&gt;</td> <td>G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB</td> </tr> <tr> <td>C.&lt;A:C&gt;/&lt;D,F&gt;</td> <td>C.AAD C.ABD C.BAF C.BBF</td> <td>G.&lt;1&gt;.&lt;2&gt;.XY&lt;2&gt;</td> <td>G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB</td> </tr> <tr> <td>A//B</td> <td>ACDB ACEB AC.B A.CB</td> <td>G//XY/</td> <td>GCXYD GCXYE GCXY.<sup>1</sup> G.XYC</td> </tr> </tbody> </table>	Auswahlmuster	Auswahl	Konstruktionsmuster	neuer Name	A/*	AB1 AB2 A.B.C	D<3><2>	D1 D2 D.CB	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<3>.XY<2>	G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<2>.XY<2>	G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB	A//B	ACDB ACEB AC.B A.CB	G//XY/	GCXYD GCXYE GCXY. <sup>1</sup> G.XYC
	Auswahlmuster	Auswahl	Konstruktionsmuster	neuer Name																	
	A/*	AB1 AB2 A.B.C	D<3><2>	D1 D2 D.CB																	
	C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<3>.XY<2>	G.A.D.XYA G.A.D.XYB G.B.F.XYA G.B.F.XYB																	
C.<A:C>/<D,F>	C.AAD C.ABD C.BAF C.BBF	G.<1>.<2>.XY<2>	G.A.A.XYA G.A.B.XYB G.B.A.XYA G.B.B.XYB																		
A//B	ACDB ACEB AC.B A.CB	G//XY/	GCXYD GCXYE GCXY. <sup>1</sup> G.XYC																		
<sup>1</sup> Punkt am Ende des Namens kann Namenskonvention widersprechen (z.B bei vollqualifizierten Dateinamen)																					
without	Schränkt die Angabemöglichkeiten für einen Datentyp ein.																				
-cat	Die Angabe einer Katalogkennung ist nicht erlaubt.																				
-corr	Eingabeformat: [[C]][V][m].na[ <sup>1</sup> ] Angaben zum Datentyp product-version dürfen den Korrekturstand nicht enthalten.																				
-dig	Der Datentyp name erlaubt keine Ziffern.																				
-gen	Die Angabe einer Dateigeneration oder Dateigenerationsgruppe ist nicht erlaubt.																				
-man	Eingabeformat: [[C]][V][m].n[ <sup>1</sup> ] 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)																				
-spec	Der Datentyp name erlaubt keine Sonderzeichen.																				

Tabelle 4: Zusätze zu Datentypen (Teil 6 von 7)



Zusatz	Bedeutung
without (Forts.)	
-temp-file	Die Angabe einer temporären Datei ist nicht erlaubt (siehe #datei bzw. @datei bei filename).
-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.
case-sensitive	Bei Angaben zu den Datentypen command-rest bzw. text wird Groß- und Kleinschreibung unterschieden. Das gilt auch für Angaben, die nicht in Hochkommata eingeschlossen sind.

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

## 7.3 Eingaberegeln

Die LMS-Anweisungen werden über die SDF-Benutzeroberfläche gelesen und vom Kommandoprozessor SDF (System Dialog Facility) verarbeitet. Damit gibt es verschiedene Formen des geführten oder ungeführten Dialogs mit der Möglichkeit, Hilfemenüs zu den Anweisungen anzufordern. Siehe dazu [3].

### Folgezeilen

Anweisungen können sich über mehrere Sätze erstrecken. Für die Trennung gelten die Konventionen der BS2000-Kommandosprache. Als Trennzeichen wird ein Bindestrich (-) verwendet. Anweisungszeilen können maximal 32763 Zeichen lang sein.

### Abkürzungsmöglichkeiten

Für die Eingabe der LMS-Anweisungen gilt, dass Anweisungsnamen, Operandennamen und Schlüsselwörter abgekürzt werden dürfen.

Es gilt folgende Regel:

Es kann jeweils von rechts nach links bis zur *Eindeutigkeit* abgekürzt werden. Dies betrifft sowohl den ganzen Namen als auch Teilnamen (beginnend mit einem Bindestrich) und schließt ein, dass evtl. der Teilname vollständig weggelassen werden kann.

Die garantierten Abkürzungsmöglichkeiten aller Anweisungen, Operanden und Operandenwerte sind in den Syntaxbeschreibungen der Anweisungen durch halbfette Schreibweise gekennzeichnet. Es können aber auch kürzere Angaben (Abkürzungen bis zur Eindeutigkeit innerhalb einer Struktur) gemacht werden.

Um Doppeldeutigkeiten durch Funktionserweiterungen in kommenden Versionen vorzubeugen und die Lesbarkeit anderen Benutzern zu ermöglichen, sollten in Prozeduren nur Langformen oder allenfalls garantierte Abkürzungen verwendet werden.

### Stellungsoperanden

SDF erlaubt die wahlweise Angabe von Operanden als Schlüsselwort- oder als Stellungsoperanden. Es kann jedoch nicht völlig ausgeschlossen werden, dass sich bei einem Versionswechsel eine Operandenposition ändert. Es wird daher empfohlen, in Prozeduren Stellungsoperanden zu vermeiden.

## 7.4 Anweisungskurznamen

Einige der gebräuchlichsten Anweisungen haben einen zusätzlichen Kurznamen. Dieser Kurzname kann nicht weiter abgekürzt werden.

Anweisung	Kurzname
COPY-ELEMENT	CP / CPE
COPY-LIBRARY	CPL
EDIT-ELEMENT	EDE
EDIT-ELEMENT-ATTRIBUTES	EDEA
EDIT-ELEMENT-PROTECTION	EDEP
MODIFY-ELEMENT	MDE
MODIFY-ELEMENT-ATTRIBUTES	MD / MDEA
MODIFY-ELEMENT-PROTECTION	MDEP
MODIFY-LIBRARY-ATTRIBUTES	MDLA
MODIFY-LMS-DEFAULTS	MDD / MDLMSD
MODIFY-LOGGING-PARAMETERS	MDLGP
MODIFY-TYPE-ATTRIBUTES	MDTA
RESET-LMS-DEFAULTS	RSD / RSLMSD
RESET-LOGGING-PARAMETERS	RSLGP
RESET-TYPE-ATTRIBUTES	RSTA
SHOW-ELEMENT	SHE
SHOW-ELEMENT-ATTRIBUTES	SH / SHEA
SHOW-LIBRARY-ATTRIBUTES	SHLA
SHOW-LIBRARY-STATUS	SHLS
SHOW-LMS-DEFAULTS	SHD / SHLMSD
SHOW-LOGGING-PARAMETERS	SHLGP
SHOW-STATISTICS	SHST
SHOW-TYPE-ATTRIBUTES	SHTA
SHOW-USER-EXITS	SHUE

Tabelle 5: Anweisungskurznamen

## 7.5 Beschreibung der LMS-Anweisungen

In diesem Abschnitt sind die LMS-Anweisungen nach einer tabellarischen Übersicht in alphabetischer Reihenfolge nach folgender Gliederung beschrieben:

- Anweisungsname und Funktion
- Beschreibung der Funktion
- Darstellung des Formates
- Beschreibung der Operanden
- notwendige Zugriffsrechte
- Hinweise
- Beispiele

Die Subanweisungen zur Anweisung MODIFY-ELEMENT sind in alphabetischer Reihenfolge ihrer Namen nach der Anweisung (ab [Seite 308](#)) beschrieben, da sie zusammen eine logische Einheit bilden.

Das gleiche gilt für die make-Subanweisungen, die nach der Anweisung BEGIN-MAKE ab [Seite 190](#) beschrieben sind.

Das Schlüsselwort \*LMS-DEFAULT wird bei den einzelnen Anweisungen nicht mehr beschrieben. Es bedeutet in der Regel den mit der Anweisung MODIFY-LMS-DEFAULTS eingestellten Wert.

Für den Elementtyp gilt zusätzlich:

Die Typangabe \*LMS-DEFAULT führt zu einem Fehler, wenn weder ein globaler noch ein bibliotheksspezifischer Defaulttyp eingestellt ist.

### Übersicht

Anweisung	Funktion
ACTIVATE-USER-EXIT	Anschließen eines Benutzer-Ausgangs
ADD-ELEMENT	Aufnehmen von Dateien als Element in Bibliotheken
BEGIN-MAKE	Einleiten der make-Subanweisungen
CALL-EDT	Aufrufen des EDT
CLOSE-LIBRARY	Schließen einer Bibliothek
COMPARE-ELEMENT	Vergleichen von Elementen
COPY-ELEMENT	Kopieren von Elementen
COPY-LIBRARY	Kopieren von Bibliotheken

<b>Anweisung</b>	<b>Funktion</b>
DEACTIVATE-USER-EXIT	Lösen eines Benutzer-Ausgangs
DELETE-ELEMENT	Löschen von Elementen
EDIT-ELEMENT	Editieren von Elementen mit EDT
EDIT-ELEMENT-ATTRIBUTES	Geführten Dialog für MODIFY-ELEMENT-ATTRIBUTES aktivieren
EDIT-ELEMENT-PROTECTION	Geführten Dialog für MODIFY-ELEMENT-PROTECTION aktivieren
END	Beenden des LMS-Laufs
EXTRACT-ELEMENT	Abspeichern von Elementen in Dateien
FIND-ELEMENT	Element mit Muster in Sätzen finden
MODIFY-ELEMENT	Modifizieren von Elementen mit Subanweisungen
MODIFY-ELEMENT-ATTRIBUTES	Ändern der Elementattribute
MODIFY-ELEMENT-PROTECTION	Ändern des Elementschutzes
MODIFY-LIBRARY-ATTRIBUTES	Ändern der Bibliotheksattribute
MODIFY-LMS-DEFAULTS	Ändern der LMS-Standardwerte
MODIFY-LOGGING-PARAMETERS	Ändern der LMS-Protokollparameter
MODIFY-TYPE-ATTRIBUTES	Ändern der Typattribute
OPEN-LIBRARY	Öffnen einer Bibliothek
PROVIDE-ELEMENT	Reservieren von Elementen
REORGANIZE-LIBRARY	Reorganisieren einer Bibliothek
RESET-LMS-DEFAULTS	Rücksetzen der LMS-DEFAULTS
RESET-LOGGING-PARAMETERS	Rücksetzen der LMS-LOGGING-PARAMETERS
RESET-TYPE-ATTRIBUTES	Rücksetzen der Typ-Attribute
RETURN-ELEMENT	Zurückgeben Reservierung gelöschter Elemente
SHOW-ELEMENT	Auflisten des Inhalts eines Elementes
SHOW-ELEMENT-ATTRIBUTES	Auflisten des Inhaltsverzeichnis einer Bibliothek
SHOW-LIBRARY-ATTRIBUTES	Auflisten der Bibliotheksattribute
SHOW-LIBRARY-STATUS	Anzeigen der zugewiesenen Bibliotheken
SHOW-LMS-DEFAULTS	Ausgeben der aktuellen LMS-Standardwerte
SHOW-LOGGING-PARAMETERS	Ausgeben der aktuellen LMS-Protokollparameter
SHOW-STATISTICS	Auflisten von Statistiken
SHOW-TYPE-ATTRIBUTES	Auflisten der Typattribute
SHOW-USER-EXITS	Anzeigen der aktiven Benutzerausgänge
WRITE-COMMENT	Schreiben von Kommentar

## ACTIVATE-USER-EXIT

### Benutzerausgang aktivieren

Diese Anweisung stellt die Benutzerausgänge für die SHOW-ELEMENT bzw. die COMPARE-ELEMENT Funktion bereit.

Mit ACTIVATE-USER-EXIT verzweigt LMS vor der Bearbeitung eines Elementsatzes in eine Benutzerroutine.

Folgende Aktionen sind vor der Bearbeitung des Elementsatzes durch LMS möglich:

- Verändern des aktuellen Elementsatzes
- Einfügen von Sätzen durch die Benutzerroutine
- Ausschließen des aktuellen Elementsatzes von der Bearbeitung

Der Benutzerroutine werden Beginn und Ende des Elementes mitgeteilt, somit können auch vor dem ersten und nach dem letzten Elementsatz vom Benutzer Sätze eingefügt werden.

Für COMPARE-ELEMENT werden zwei Benutzerausgänge angeboten: einer für das Primärelement und einer für das Sekundärelement. Die Benutzerausgänge für SHOW-ELEMENT und COMPARE-ELEMENT können gleichzeitig definiert werden.

Werden mehrere ACTIVATE-USER-EXIT mit dem gleichen Benutzerausgang angegeben, gilt immer der zuletzt angegebene.

Die aktivierten Benutzerausgänge können mit der Anweisung DEACTIVATE-USER-EXIT wieder deaktiviert werden.

Die ACTIVATE-USER-EXIT-Anweisung erfordert die Angabe der Funktion, für die der Benutzerausgang aktiviert werden soll, und des Einsprungpunktes des Benutzerprogramms. Alle anderen Parameter sind voreingestellt, d.h. das Benutzerprogramm wird standardmäßig aus der TASKLIB-Bibliothek nachgebunden.

#### ACTIVATE-USER-EXIT

**FUNCTION** = \*SHOW-ELEMENT / \*COMPARE-ELEMENT(...)

  \*COMPARE-ELEMENT(...)

    | **ELEMENT** = \*PRIMARY / \*SECONDARY

,**ENTRY** = <name 1..8>

,**LIBRARY** = \*TASKLIB / \*OMF / <filename 1..54>

,**INTERFACE-VERSION** = \*V1 / \*V2

**FUNCTION = \*SHOW-ELEMENT / \*COMPARE-ELEMENT(...)**

Bestimmt diejenige LMS-Anweisung, für die der Benutzerausgang aktiviert werden soll.

**FUNCTION = \*SHOW-ELEMENT**

Benutzerausgang für die SHOW-ELEMENT-Funktion. Sie gilt für alle textartigen Elementtypen und R, F, H und U bzw. davon abgeleitete Elementtypen.

**FUNCTION = \*COMPARE-ELEMENT(...)**

Benutzerausgang für die COMPARE-ELEMENT-Funktion. Sie gilt für alle textartigen Elementtypen und R, F, H und U bzw. davon abgeleitete Elementtypen.

**ELEMENT = \*PRIMARY / \*SECONDARY**

Wird der Benutzerausgang für die Anweisung COMPARE-ELEMENT aktiviert, muss noch festgelegt werden, ob es sich um ein Primär- oder um ein Sekundärelement handelt.

**ENTRY = <name 1..8>**

Maximal 8 Zeichen langer Name des Einsprungpunktes der Benutzerroutine. Der Name darf nicht mit der Zeichenfolge 'LMS' beginnen.

**LIBRARY = \*TASKLIB / \*OMF / <filename 1..54>**

Quelle, in der sich der Benutzermodul befindet.

**LIBRARY = \*TASKLIB**

Der gewünschte Modul wird aus der TASKLIB nachgebunden (siehe Abschnitt „Nachladen des Benutzerprogramms“, [Seite 168](#)).

**LIBRARY = \*OMF**

Der gewünschte Modul wird aus \*OMF nachgebunden.

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

Der gewünschte Modul wird aus der angegebenen Bibliothek nachgebunden.

**INTERFACE-VERSION = \*V1 / \*V2**

Es existieren zwei Schnittstellen-Versionen. Aus Kompatibilitätsgründen werden standardmäßig die Werte BOE, REC und EOE angeboten. Durch die Angabe von V2 wird die Schnittstelle um die Werte C'BOS' (Beginn of Statement) und C'EOS' (End of Statement) erweitert. Dadurch wird dem Benutzerprogramm Anweisungsbeginn und -ende mitgeteilt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

### Notwendige Zugriffsrechte

Das Benutzerprogramm wird dynamisch zugebunden. Die Zugriffsrechte hängen damit vom Verhalten des BIND-Makros ab.

### Nachladen des Benutzerprogramms

Das Benutzerprogramm wird immer in den benutzereigenen Klasse-6-Speicher geladen. Fehlt bei LIBRARY die Angabe einer Bibliothek, wird das Benutzerprogramm zuerst in einer eventuell vorhandenen privaten Tasklib gesucht (mit /SET-TASKLIB LIBRARY=bibliothek zugewiesen) und dann in der Tasklib des Systems \$TASKLIB. Wurde LIBRARY = \*OMF oder eine Datei angegeben, aber das Benutzerprogramm dort nicht gefunden, so wird das Benutzerprogramm in der mit LINK=BLSLIB bzw. in den mit LINK=BLSLIBnn (00 ≤ nn ≤ 99) zugewiesenen Bibliotheken mit aufsteigender Nummer „nn“ gesucht.

### Schnittstelle des Benutzerausgangs

#### – Registerkonventionen

Beim Aufruf des Benutzerprogramms werden von LMS folgende Registerkonventionen berücksichtigt:

- Register 1: Adresse einer Parameterleiste
- Register 13: Adresse des Sicherstellungsbereiches (18 Worte)
- Register 14: Rücksprungadresse
- Register 15: Adresse des Einsprungpunktes

#### – Aufbau der Parameterleiste

Die Parameterleiste besteht aus 5 Worten:

- DC A(Auftragsbeschreibung)
- DC A(Antwortbeschreibung)
- DC A(zu übergebender Satz)
- DC A(Bibliothekensname)
- DC A(Elementbezeichnung)



Die ersten beiden Adressen werden von LMS versorgt, d.h. insbesondere, dass der Benutzer die Antwortbeschreibung nur an der von LMS vorgegebenen Adresse abliefern darf. Die Satzadresse kann sowohl von LMS als auch vom Benutzer versorgt werden.

1. Wort: Auftragsbeschreibung

Die Auftragsbeschreibung an das Benutzerunterprogramm besteht aus 3 Byte und kann folgenden Inhalt haben:

C'BOE'	Beginn des Elementes
C'REC'	Satz steht zur Verarbeitung an
C'EOE'	Ende des Elementes

Wurde die erweiterte Schnittstelle gewählt, dann kann die Auftragsbeschreibung folgenden Inhalt haben:

C'BOS'	Beginn der Anweisung
C'BOE'	Beginn des Elementes
C'REC'	Satz steht zur Verarbeitung an
C'EOE'	Ende des Elementes
C'EOS'	Ende der Anweisung

2. Wort: Antwortbeschreibung

Die Antwortbeschreibung vom Benutzerunterprogramm besteht aus 3 Byte und kann folgenden Inhalt haben:

C'CON'	LMS verarbeitet den Satz, dessen Adresse nach Rückkehr vom Benutzerprogramm in der Parameterleiste steht und bietet danach dem Benutzerprogramm den nächsten Elementsatz oder EOE an.
C'DEL'	LMS übergeht den zuletzt angebotenen Elementsatz und springt mit dem nächsten Elementsatz oder EOE erneut zum Benutzerunterprogramm.
C'INS'	LMS verarbeitet zunächst den vom Benutzer zur Verfügung gestellten Satz und kehrt danach mit dem vorher angebotenen Elementsatz oder EOE an das Benutzerunterprogramm zurück; dies wird solange wiederholt, bis der Benutzer mit DEL oder CON antwortet; d.h. wenn LMS den Satz i anbietet, werden die mit INS zurückgesandten Sätze vor dem Satz i bearbeitet.

Auf die Aufträge kann wie folgt geantwortet werden:

Auftrag	Antwort
BOS	nicht relevant
BOE	nicht relevant
REC	CON, DEL, INS
EOE	CON, INS
EOS	nicht relevant

Bei falscher Antwort wird die LMS-Funktion mit einer Fehlermeldung abgebrochen.

Zur Verdeutlichung der Kommunikation zwischen LMS und dem Benutzerunterprogramm siehe Diagramm.

### 3. Wort: Satz

Dieses Wort enthält die Adresse des Satzes, der von LMS an das Benutzerunterprogramm übergeben wird. Kehrt das Benutzerunterprogramm mit der Antwort CON oder DEL zurück, kann die gleiche Satzadresse verwendet werden. Möchte der Benutzer Sätze einfügen, muss er einen eigenen Satzpuffer verwenden.

Die zwischen LMS und dem Benutzerunterprogramm ausgetauschten Sätze beginnen mit einem 4 Byte langen Satzlängenfeld.

Zum Auflisten von Typ P Elementen mit SHOW-ELEMENT muss das 5. Byte ein gültiges Vorschubsteuerzeichen sein.

### 4. Wort: Bibliotheksname

Dieses Wort enthält die Adresse des Bibliotheksnamens der von LMS aktuell bearbeiteten Bibliothek. Der Bibliotheksname beginnt mit einem 2 Byte langen Satzlängenfeld.

### 5. Wort: Elementbezeichnung

Dieses Wort enthält die Adresse der Elementbezeichnung des von LMS aktuell bearbeiteten Elementes. Die Elementbezeichnung hat die Form: 2 Byte langes Satzlängenfeld, gefolgt von (Typ)Elementname/Version[(Variantennummer)]/Datum

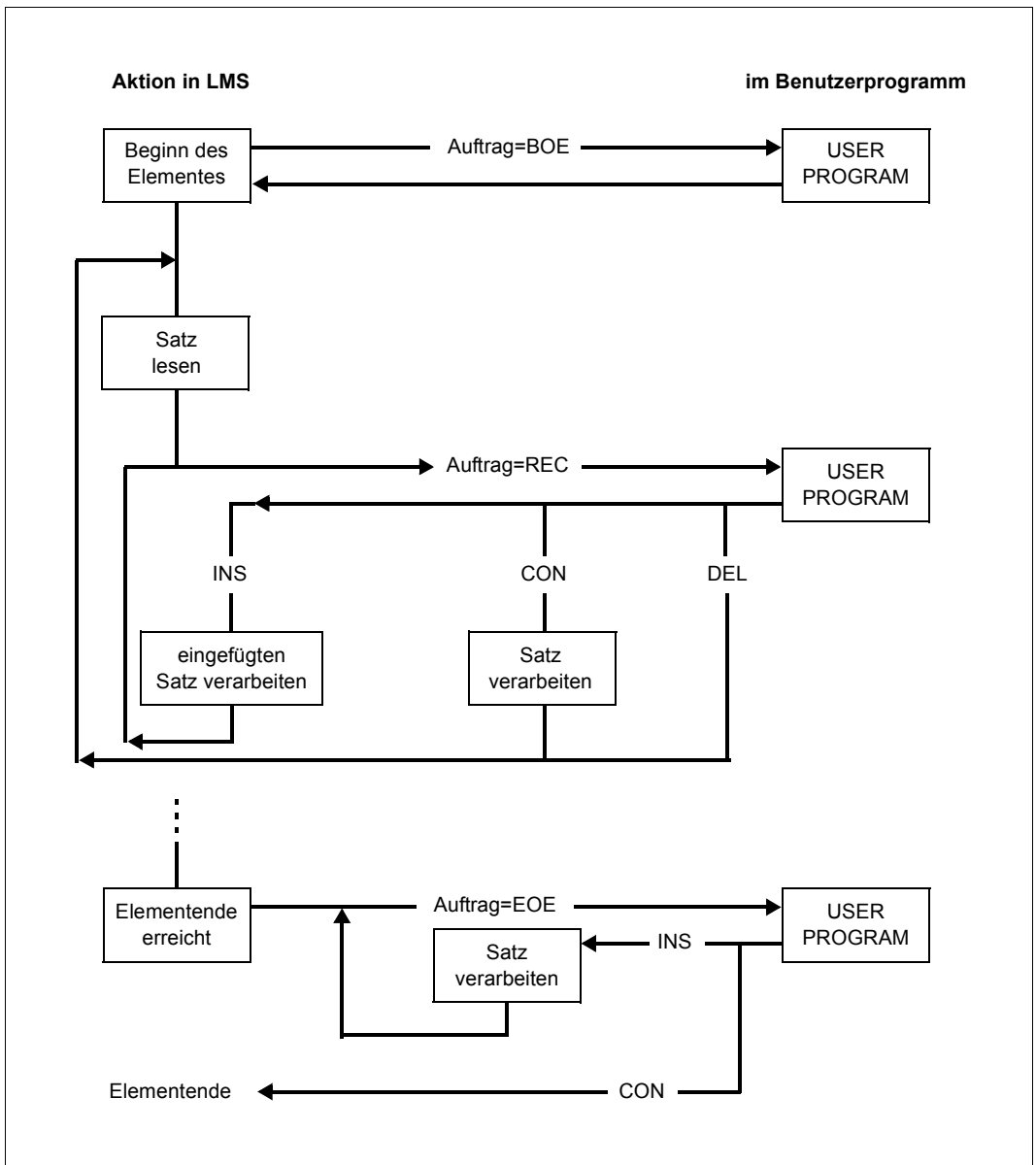


Bild 10: Kommunikation zwischen LMS und Benutzerprogramm

Ein Ablaufbeispiel zum Benutzerausgang siehe [Seite 493](#).

*Beispiele*

- Minimal-Angabe für die Anweisung ACTIVATE-USER-EXIT.  
Das Benutzerprogramm module1 wird für die SHOW-ELEMENT-Funktion als Benutzerausgang bereitgestellt.

```
//activate-user-exit function=*show-element, entry=module1
```

- Das gewünschte Benutzerprogramm befindet sich in der \*OMF-Bibliothek.

```
//activate-user-exit function=*show-element, entry=module1,library=*omf
```

## **ADD-ELEMENT**

### **Datei als Element in eine Bibliothek aufnehmen**

ADD-ELEMENT nimmt Dateien als Voll- bzw. Delta-Elemente in eine Bibliothek auf. Die Elementdaten werden standardmäßig von SYSDTA gelesen. Sie können aber auch von einer explizit angegebenen Datei oder \*OMF gelesen werden. Erfolgt die Anweisung jedoch aus der EDT-Anweisungszeile, werden die Elementdaten standardmäßig aus der aktuellen EDT-Arbeitsdatei gelesen.

Die Dateien werden immer ohne Präfix, d.h. ohne CATID und USERID, als Element in eine Bibliothek aufgenommen, es sei denn, der Benutzer hat in der Konstruktionsangabe explizit ein Präfix angegeben.

Dateien, die mit RECORD-FORMAT=\*UNDEFINED katalogisiert sind, können ebenfalls in Bibliotheken aufgenommen werden. Dateien, die RECORD-FORMAT=\*FIXED haben, können nur über SOURCE-ATTRIBUTES=\*KEEP abgespeichert werden.

Die Satzformate FIXED und UNDEFINED werden in das Satzformat VARIABLE umgewandelt; d.h. mit einem 4 Byte Satzkopf versehen. Die Satzlänge inklusive Satzkopf darf 32KByte nicht überschreiten.

Dateigenerationsgruppen können nur über Linknamen und eine LMS-gültige Elementbezeichnung aufgenommen werden.

LMS übernimmt bei der ADD-ELEMENT-Anweisung das Katalogattribut CCS der Datei als Elementattribut. Werden die Daten von SYSDTA gelesen, erhält das erzeugte Element als Attribut den für SYSDTA eingestellten CCS-Namen. Beim Lesen von \*OMF wird den Elementen „kein Code“ zugeordnet. Werden die Daten aus einer EDT-Arbeitsdatei gelesen, erhält das erzeugte Element als Attribut den aktuell eingestellten CCS-Namen des EDT.

## ADD-ELEMENT

```

FROM-FILE = *STD / *SYSDTA(...) / *ALL / <filename 1..80 without-vers with-wild> / *LINK(...) / *OMF
  *SYSDTA(...)
    | END = *END / <c-string 1..8>
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
,TO-ELEMENT = *LIBRARY-ELEMENT (...)
  *LIBRARY-ELEMENT(...)
    | LIBRARY = *STD / *LINK(...) / <filename 1..54 without-vers>
      *LINK(...)
        | LINK-NAME = <structured-name 1..8>
    ,ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wild-constr>(…)
      *BY-SOURCE(...)
        | VERSION = *LMS-DEFAULT / *HIGHEST-EXISTING / *INCREMENT /
          *UPPER-LIMIT / <composed-name 1..24 with-under>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
        <composed-name 1..132 with-under with-wild-constr>(…)
        | VERSION = *LMS-DEFAULT / *HIGHEST-EXISTING / *INCREMENT /
          *UPPER-LIMIT / <composed-name 1..24 with-under>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
      ,TYPE = *LMS-DEFAULT / <alphanum-name 1..8>
      ,USER-DATE = *TODAY / <date 8..10 with-compl>
      ,STORAGE-FORM = *LMS-DEFAULT / *STD / *FULL / *DELTA
,ELEMENT-ATTRIBUTES = *LMS-DEFAULT / *PARAMETERS(...)
  *PARAMETERS(...)
    | SOURCE-ATTRIBUTES = *LMS-DEFAULT / *STD / *IGNORE / *KEEP
,DELETE-SOURCE = *LMS-DEFAULT / *NO / *YES
,PROTECTION = *LMS-DEFAULT / *STD / *BY-SOURCE
,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *EXTEND / *ANY
,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

**FROM-FILE = \*STD / \*SYSDTA(...) / \*ALL / <filename 1..80 without-vers with-wild>/ \*LINK(..) / \*OMF**

Angabe derjenigen Datei, die als Element in die Bibliothek aufgenommen werden soll.

**FROM-FILE = \*STD**

Standardmäßig werden die Datensätze von der Systemdatei SYSDTA gelesen. Erfolgt die Anweisung jedoch aus der EDT-Anweisungszeile, werden die Datensätze standardmäßig aus der aktuellen EDT-Arbeitsdatei gelesen.

Zulässige Elementtypen:

- für Voll-Elemente: S, M, P, J, D, X oder davon abgeleitete Typen
- für Delta-Elemente: analog

**FROM-FILE = \*SYSDTA(...)**

Die Datensätze werden mit RDATA von der Systemdatei SYSDTA gelesen. Die Datensätze müssen direkt auf die ADD-ELEMENT-Anweisung folgen.

Zulässige Elementtypen:

- für Voll-Elemente: S, M, P, J, D, X, R oder davon abgeleitete Typen
- für Delta-Elemente: S, M, P, J, D, X oder davon abgeleitete Typen

**END = \*END' / <c-string 1..8>**

Endekriterium für die Eingabe. Die Folge der Datensätze muss mit '\*END' oder einem selbst definierten Endekriterium (siehe Beispiel [Seite 183](#)) abgeschlossen werden.

Fehlt das Endekriterium in den Eingabedaten, wird bis EOF gelesen.

#### *Hinweis*

Werden Datensätze von der Systemdatei SYSDTA=(SYSCMD) gelesen, so dürfen sie nicht mit dem Zeichen „/“ beginnen. Der Grund dafür ist, dass der RDATA-Makro solche Sätze als Kommandos interpretiert und daher den Return-Code für EOF übergibt. Es ist damit nicht möglich, System-Kommandos als Datensätze zu übergeben.

**FROM-FILE = \*ALL**

Es wird versucht, alle Dateien der Kennung in die Bibliothek zu übernehmen. Tritt bei einer Datei ein Fehler auf, wird diese übergangen und mit der nächsten Datei fortgefahren.

**FROM-FILE = <filename 1..80 without-vers with-wild>**

Die Daten werden aus der angegebenen Datei gelesen.

Zulässige Elementtypen:

- für Voll-Elemente: S, M, P, J, D, X, R oder davon abgeleitete Typen
- für Delta-Elemente: S, M, P, J, D, X oder davon abgeleitete Typen

Dateien vom Typ PAM können nur als Voll-Elemente unter dem Elementtyp X oder davon abgeleiteten Elementtypen abgelegt werden.

**FROM-FILE = \*LINK(...)**

Die Daten werden aus der über den Dateikettungsnamen angegebenen Datei gelesen.

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

Dateikettungsname, der auf die Datei verweist.

**FROM-FILE = \*OMF**

Gilt nur für Typ R Elemente.

Die Daten werden aus der OMF-Datei gelesen. Es werden alle Module aus der OMF-Datei übernommen. Sind im EAM-Bereich mehrere gleichnamige Module enthalten, übernimmt LMS den zuletzt übersetzten Modul in die Bibliothek.

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

Angabe, wohin und unter welchem Namen das Element aufgenommen werden soll.

**LIBRARY = \*STD / \*LINK(...) / <filename 1..54 without-vers>**

Angabe der Bibliothek, in die das Element aufgenommen werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY global eröffnete Bibliothek.

**LIBRARY = \*LINK(...)**

Die über einen Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

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

Name der Bibliothek, in die die Datei als Element aufgenommen werden soll.

**ELEMENT = \*BY-SOURCE(...) /****<composed-name 1..132 with-under with-wildcard-constr>(...)**

Name, den das neu aufzunehmende Element erhalten soll. Eine Konstruktionsangabe bezieht sich auf den Dateinamen.

**ELEMENT = \*BY-SOURCE(...)**

Der Elementname entspricht dem Dateinamen bzw. dem Modulnamen bei \*OMF.

**VERSION = \*LMS-DEFAULT / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das neu aufzunehmende Element erhalten soll.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.



**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion (siehe auch [Seite 59](#)).

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Wird der Operand BASE in der Form <composed-name 1..23>\* eingegeben, wird er als Präfix interpretiert. Näheres zu den Wirkungen von BASE siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Das Element wird mit dem hier angegebenen Namen abgelegt.

**VERSION = \*LMS-DEFAULT / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das neu aufzunehmende Element erhalten soll. Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

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

Typ, den das neu aufzunehmende Element erhalten soll.

**USER-DATE = \*TODAY / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**STORAGE-FORM = \*LMS-DEFAULT / \*STD / \*FULL / \*DELTA**

Speicherform für das neu anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*STD**

Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

**STORAGE-FORM = \*DELTA**

Das Element wird delta-gespeichert angelegt (Fehler, falls nicht zulässig). Die Angabe ist zulässig für die Elementtypen: S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**ELEMENT-ATTRIBUTES = \*LMS-DEFAULT / \*PARAMETERS(...)**

Bestimmt, ob die Dateieigenschaften und auch der ISAM-Schlüssel in das Ausgabeelement mit übernommen werden.

**ELEMENT-ATTRIBUTES = \*PARAMETERS(...)****SOURCE-ATTRIBUTES = \*LMS-DEFAULT / \*STD / \*IGNORE / \*KEEP**

Speichern von Dateiattributen. Dieser Operand ist wirkungslos, wenn die Daten von SYSDTA, \*OMF oder aus einer EDT-Arbeitsdatei gelesen werden. Es werden keine Ursprungsattribute gespeichert.

Werden die Daten aus einer Datei vom Filetyp UPAM gelesen, so ist die Angabe wirkungslos; es wird immer wie im Fall \*KEEP verfahren.

**SOURCE-ATTRIBUTES = \*STD**

Es werden keine Dateiattribute gespeichert. Im Fall von ISAM-Dateien ist es nur möglich, ISAM-Dateien mit KEY-POSITION = 5, KEY-LENGTH <= 16 und RECORD-FORMAT = VARIABLE in das Element aufzunehmen. Es erfolgt eine Warnung, dass die ISAM-Schlüssel nicht mit aufgenommen wurden.

**SOURCE-ATTRIBUTES = \*IGNORE**

Wie bei SOURCE-ATTRIBUTES = \*STD, jedoch ohne Warnung.

**SOURCE-ATTRIBUTES = \*KEEP**

Folgende Dateiattribute werden unverändert im neu aufzunehmenden Element gespeichert: ACCESS-METHOD, RECORD-FORMAT, RECORD-SIZE, BUFFER-LENGTH, PERFORMANCE, USAGE, ACCESS und USER-ACCESS. Ist ACCESS-METHOD= ISAM, so werden PADDING-FACTOR, LOGICAL-FLAG-LENGTH, VALUE-FLAG-LENGTH, PROPAGATE-VALUE-FLAG und auch die ISAM-Schlüssel sowie Informationen über ISAM-Sekundärschlüssel zusätzlich gespeichert.

**DELETE-SOURCE = \*LMS-DEFAULT / \*NO / \*YES**

Hier kann der Benutzer bestimmen, ob die ursprüngliche Datei erhalten bleiben oder gelöscht werden soll. Dieser Operand ist wirkungslos, wenn die Daten von SYSDTA, \*OMF oder aus einer EDT-Arbeitsdatei gelesen werden.

**DELETE-SOURCE=\*NO**

Die ursprüngliche Datei wird nicht gelöscht.

**DELETE-SOURCE=\*YES**

Die ursprüngliche Datei wird gelöscht.

**PROTECTION= \*LMS-DEFAULT / \*STD / \*BY-SOURCE**

Elementschutz für das aufzunehmende Element. Dieser Operand ist wirkungslos, wenn die Daten von SYSDTA, \*OMF oder aus einer EDT-Arbeitsdatei gelesen werden.

**PROTECTION=\*STD**

Existiert das Element bereits, bleibt der Elementschutz unverändert. Existiert das Element noch nicht und ist für die Bibliothek bzw. den Typ des Elements ein initialer Elementschutz festgelegt, so erhält das Element diesen Schutz.

**PROTECTION=\*BY-SOURCE**

Das Element erhält einen Elementschutz entsprechend den Dateischutzattributen des aktivierten Zugriffsschutzmechanismus der Datei. ADD-ELEMENT wird mit einer Fehlermeldung abgewiesen, wenn die Datei durch den Zugriffsschutzmechanismus Access Control List (ACL) geschützt ist.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

Ist das zu speichernde Element ein Delta-Element, ist darauf zu achten, dass das Element ein Blatt des Delta-Baumes sein muss. Nur Blätter eines Delta-Baumes dürfen überschrieben werden.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Ein Element wird nur dann überschrieben, wenn schon ein Element gleichen Namens vorhanden ist. Ansonsten wird ADD-ELEMENT mit einer Fehlermeldung abgewiesen.

**WRITE-MODE = \*EXTEND**

Ein Element wird jedoch nur dann erweitert, wenn im Element keine ISAM-Schlüssel gespeichert sind und die im Element gespeicherten Dateimerkmale mit den Merkmalen der Datei bis auf den Dateinamen übereinstimmen. Sonst wird ADD-ELEMENT mit einer Fehlermeldung zurückgewiesen.

EXTEND ist nicht für Delta-Elemente und bei Eingabe von SYSDTA erlaubt.

**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL =\*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0064	GCCSN Makro Fehler; kein CCS-Name spezifiziert
2	0	LMS0071	XHCS nicht geladen
2	0	LMS0095	Eingabesätze fehlen
2	0	LMS0102	Unvollständiger Modul in EAM Datei
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0301	Datei nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

## Notwendige Zugriffsrechte

Bei FROM-FILE: Leserecht für die Datei

Sind von der Anweisung mehrere Dateien betroffen, so werden Dateien ohne Leserecht von der Anweisung ausgenommen.

Bei TO-ELEMENT: Lese- und Schreibrecht für LIBRARY Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).

Bei STORAGE-FORM=\*DELTA muss für das durch BASE bestimmte Element Leserecht gegeben sein.

Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Eine neue Version kann nur erzeugt oder diese Basisversion überschrieben werden, wenn Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

### *Hinweise*

- Bei Aufnahme einer Datei in eine Bibliothek gilt nicht das Erzeugungsdatum der Datei, sondern das Erzeugungsdatum des Elementes.
- Wird SOURCE-ATTRIBUTES=\*KEEP angegeben, ist zu beachten:  
Durch eventuell vorhandene ISAM-Schlüssel kann die nachfolgende Verarbeitung wie Übersetzen und /CALL-PROCEDURE behindert werden. Dieser Parameterwert eignet sich vor allem zur Archivierung.
- Beim Anlegen eines Elements ist die unter dem Typ geltende Konvention zu beachten.
- Bei Aufnahme von temporären Dateien mit Wildcards ist keine Konstruktionsangabe des Zielelementnamens erlaubt, d.h. ELEM = darf nur \*BY-SOURCE sein.
- Bei Aufnahme von temporären Dateien mit ELEM = \*BY-SOURCE erhält das Element den internen Dateinamen. Dieses Element kann unter einer anderen TASK ohne explizite Angabe eines Dateinamens nicht wieder in eine Datei ausgegeben werden.
- Bei aktivem WRITE-CONTROL in der Ausgabebibliothek wird dem zu schreibenden Element von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorgangs protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), werden die Elementattribute STATE, HOLDER und alle Rechte von der Basisversion auf die neue Version übernommen. Die CCSN wird von der Quelldatei übernommen. Das USER-DATE wird neu bestimmt.

- Wird PROTECTION=\*BY-SOURCE angegeben, ist zu beachten:  
Die Aufnahme der Datenschutzattribute des aktivierten Zugriffsschutzmechanismus in einen entsprechenden Elementschutz ist für die Mechanismen Standard-Zugriffskontrolle, BACL und GUARDS möglich. Hierbei erhält das Element für die Zugriffsrechte (read, write, execute) beim aktivierten Zugriffsschutzmechanismus der Datei folgenden entsprechenden Schutzmechanismus:

Schutzmechanismus der Datei		Schutzmechanismus des Elementes
Standard-Zugriffskontrolle (ACCESS / USER-ACCESS)	→	Schutz durch BACL (ohne Kennwort)
BACL	→	Schutz durch BACL (ohne Kennwort)
GUARDS <sup>1</sup>	→	Schutz durch GUARD

<sup>1</sup> Sonderfall:

Ist eines der Zugriffsrechte der Datei beim Schutzmechanismus GUARDS mit dem Wert NONE (kein Zugriff für dieses Recht möglich) belegt, erhält das Element für das entsprechende Recht Schutz durch BACL mit USER=NONE (kein Zugriff möglich).

Bestehende Kennwörter auf Dateien gehen nicht in den Elementschutz ein.

Bei der Aufnahme von Dateien auf bestehende Elemente gehen die ggf. bestehenden Elementschutzeinstellungen der Zugriffsrechte (read, write, execute) verloren.

Für das Ausleihrecht (hold) gilt die Verfahrensweise wie bei PROTECTION=\*STD. Das heißt, existiert das Element bereits, bleibt der Elementschutz in Bezug auf das Ausleihrecht unverändert. Existiert das Element noch nicht und ist für die Bibliothek bzw. den Typ des Elementes ein initialer Elementschutz in Bezug auf das Ausleihrecht festgelegt, erhält das Element diese Einstellung.

### Beispiele

- Aufnahmen eines Elementes

Das Element testelem wird in die Bibliothek lib1 unter dem gleichen Namen aufgenommen. Die Typangabe muss hier explizit in der ADD-ELEMENT-Anweisung angegeben werden, da standardmäßig der Typ mit \*NONE voreingestellt ist.

```
/show-file-attributes
TESTELEM
.
//start-lms
//open-library lib1,*update
//add-element from-file=testelem,to-elem=*lib(type=d)
.
.
```

## – Lesen von SYSDTA und der Definition des Endkriteriums

```

/start-lms _____ (1)
//open-library lib1,*update _____ (2)
//modify-lms-defaults (type=d) _____ (3)
//add-element *sysdta(end=c'stop'),to-element=*lib(element=letter.a) _____ (4)
* sehr geehrte ... _____ (5)
.
.
.
*STOP _____ (6)
//end _____ (7)

```

- (1) LMS wird gestartet.
- (2) Die Bibliothek mit dem Namen lib1 wird zum Lesen und Schreiben geöffnet.
- (3) Der LMS-Defaultwert, der für die Typ-Angabe \*NONE ist, wird in die Elementtyp-Angabe „D“ geändert. Diese Einstellung gilt für den gesamten LMS-Lauf, wenn nicht eine neue MODIFY-LMS-DEFAULTS-Anweisung betreffend des Elementtyps gegeben wird oder der Typ lokal in einer Anweisung geändert wird.
- (4) Es sollen Datensätze von SYSDTA eingelesen werden, wobei als Endkriterium das Wort stop vereinbart wird. Die Datensätze sollen unter dem Elementnamen letter.a in die durch OPEN-LIBRARY eröffnete Bibliothek lib1 abgelegt werden. Der Elementtyp muss nicht mehr angegeben werden.
- (5) Eingabe des Textes. Der Text 'Sehr geehrte ...' wird so gespeichert, wie er eingetippt wurde.
- (6) Die Aufnahme der Datensätze wird durch die Angabe von stop beendet.
- (7) LMS wird beendet.

## BEGIN-MAKE

### Einleiten der make-Verarbeitung

Die BEGIN-MAKE-Anweisung leitet die make-Subanweisungen ein. Die make-Subanweisungen werden mit END-MAKE beendet.

Mit BEGIN-MAKE kann folgendes spezifiziert werden:

- Ziel, das mit dem make-Lauf erzeugt werden soll
- Art der make-Verarbeitung (notwendige) oder alle Aktionen ausführen
- Folgeverarbeitung

Wird eine Prozedur synchron ausgeführt, beendet dies LMS.

#### BEGIN-MAKE

```

TARGET = *FIRST-TARGET / <filename 1..54 without-vers> / *LIBRARY-ELEMENT(...)
  *LIBRARY-ELEMENT(...)
    | LIBRARY = <filename 1..54 without-vers> / *LINK(...)
    | *LINK(...)
    | | LINK-NAME = <structured-name 1..8>
    | ,ELEMENT = <composed-name 1..64 with-under>(…)
    | <composed-name 1..64 with-under>(…)
    | | VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under>
    | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    | ,TYPE = <alphanum-name 1..8>
,SELECT = *MODIFIED / *ALL
,SUCCESS-PROCESSING = *INCLUDE-PROCEDURE / *CALL-PROCEDURE / *ENTER-PROCEDURE /
  *CREATE-PROCEDURE / *TOUCH

```

(Teil 1 von 2)



```

,PROCEDURE = SYSPRC.LMS.MAKE / <filename 1..54 without-vers> / *LIBRARY-ELEMENT(...)
  *LIBRARY-ELEMENT(...)
    |
    | LIBRARY = *BY-TARGET / <filename 1..54 without-vers> / *LINK(...)
    |
    | *LINK(...)
    | | LINK-NAME = <structured-name 1..8>
    |
    | ,ELEMENT = <composed-name 1..64 with-under>(…)
    |
    | <composed-name 1..64 with-under>(…)
    | |
    | | VERSION = *HIGHEST-EXISTING / *INCREMENT / *UPPER-LIMIT /
    | | <composed-name 1..24 with-under>
    |
    | ,TYPE = J / <alphanum-name 1..8>
,PROCEDURE-PARAMETERS = *NO / <text 1..1800 with-low>

```

(Teil 2 von 2)

**TARGET = \*FIRST-TARGET** / <filename 1..54 without-vers> /  
\***LIBRARY-ELEMENT**(...)

Zu produzierendes Ziel.

**TARGET = \*FIRST-TARGET**

Das erste Ziel (Bibliothekselement oder Datei) unter den Abhängigkeitsdefinitionen ist das Ziel des gesamten make-Laufs.

**TARGET = <filename 1..54 without-vers>**

Das Ziel ist eine Datei. Der Initialwert für die Default-Bibliothek in MODIFY-MAKE-DEFAULTS ist \*NONE.

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

Das Ziel befindet sich in einem Bibliothekselement. Eine angegebene Zielbibliothek ist der Initialwert für die Default-Bibliothek in MODIFY-MAKE-DEFAULTS.

**LIBRARY = <filename 1..54 without-vers> / \*LINK**(...)

Bibliothek des Zielelements.

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

Bibliothek des Zielelements.

**LIBRARY = \*LINK**(...)

Bibliothek des Zielelements.

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

Linkname der Bibliothek des Zielelements.

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

Name des Zielelements.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /**  
**<composed-name 1..24 with-under>**

Version des Zielelements.

**VERSION = \*HIGHEST-EXISTING**

Es wird die höchste existierende Version als Ziel herangezogen.

**VERSION = \*UPPER-LIMIT**

Es wird die höchstmögliche Version X'FF' als Ziel herangezogen.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis des Zielelements fest.

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

Typ, den das Zielelement besitzt.

**SELECT = \*MODIFIED / \*ALL**

Komponenten, die zur Erzeugung von Zielen herangezogen werden.

**SELECT = \*MODIFIED**

Die geänderten Komponenten (=Komponenten neueren Datums) sollen Einfluss haben.

**SELECT = \*ALL**

Alle Komponenten sollen verwendet werden. Es wird simuliert, dass alle Komponenten, die zum Ziel führen, nicht aktuell sind.

**SUCCESS-PROCESSING = \*INCLUDE-PROCEDURE / \*CALL-PROCEDURE /**  
**\*ENTER-PROCEDURE / \*CREATE-PROCEDURE / \*TOUCH**

Art der abschließenden Bearbeitung. Am Ende des make-Laufs (END-MAKE-Anweisung) wird die entsprechende Verarbeitung ausgeführt. Wenn das Ziel bei TARGET aktuell ist, wird die abschließende Bearbeitung nicht durchgeführt.

**SUCCESS-PROCESSING = \*INCLUDE-PROCEDURE**

Die unter PROCEDURE angegebene Prozedur wird generiert und synchron mit den aktuellen Variablen ausgeführt.

**SUCCESS-PROCESSING = \*CALL-PROCEDURE**

Die unter PROCEDURE angegebene Prozedur wird generiert und synchron mit neuem Variablen-Environment ausgeführt.

**SUCCESS-PROCESSING = \*ENTER-PROCEDURE**

Die unter PROCEDURE angegebene Prozedur wird generiert und asynchron mit neuem Variablen-Environment ausgeführt.

**SUCCESS-PROCESSING = \*CREATE-PROCEDURE**

Die Prozedur wird nur generiert.

**SUCCESS-PROCESSING = \*TOUCH**

Die Komponenten des Programmsystems, die erzeugt werden müssen, werden berührt, d.h. sie erhalten einen neuen Modifikationszeitpunkt als Zeitstempel. Anschließend ist das gesamte Programmsystem bis TARGET aktuell. Es wird keine Prozedur erzeugt. Eine Datei, die mit \*TOUCH behandelt werden soll, darf weder leer noch gegen UPAM-Zugriffe geschützt sein.

**PROCEDURE = SYSRPC.LMS.MAKE / <filename 1..54 without-vers> / \*LIBRARY-ELEMENT(...)**

Prozedur, die durch make erzeugt und eventuell am Ende gestartet wird.

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

Die Prozedurdatei, die erzeugt wird, ist ein Element in einer Bibliothek.

**LIBRARY = \*BY-TARGET / <filename 1..54 without-vers> / \*LINK(...)**

Bibliothek, in der die Prozedur gespeichert wird.

**LIBRARY = \*BY-TARGET**

Bibliothek ist die von TARGET.

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

Bibliothek des Prozedurelements.

**LIBRARY = \*LINK(...)**

Bibliothek des Prozedurelements.

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

Linkname der Bibliothek des Prozedurelements.

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

Name des Prozedurelements.

**VERSION = \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version des Prozedurelements.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version ansonsten eine Defaultversion erzeugt.

**VERSION = \*UPPER-LIMIT**

Es wird die höchstmögliche Version X'FF' erzeugt.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

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

Typ, den das Prozedurelement besitzt.

**PROCEDURE-PARAMETERS = \*NO / <text 1..1800 with-low>**

Parameter, mit denen die Prozedur beim Aufruf versorgt wird. Sie entsprechen dem PROCEDURE-PARAMETERS-Operanden von CALL-PROCEDURE, ENTER-PROCEDURE und INCLUDE-PROCEDURE und werden unverändert weitergegeben.

**PROCEDURE-PARAMETERS = \*NO**

Keine Prozedurparameter.

**PROCEDURE-PARAMETERS = <text 1..1800 with-low>**

Prozedurparameter, die unverändert weitergegeben werden.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweis*

Nur wenn SDF-P im System vorhanden ist, kann ein erzeugtes Prozedurelement mit einer Version ungleich der höchsten ausgeführt werden (und auch nur mit CALL-PROCEDURE und INCLUDE-PROCEDURE).

*Beispiel*

Im Folgenden Teilstück einer Prozedur soll das Ziel aus dem Parameter TARGET mit Hilfe der Definitionen in der Datei MAKEFILE erstellt werden, wenn es nicht aktuell ist. In der Datei MAKEFILE stehen make-Subanweisungen. Zum Abschluss wird die standardmäßig erzeugte Prozedur ausgeführt und LMS beendet.

```
...  
/START-LMS  
/BEGIN-BLOCK DATA-INSERTION=YES  
//BEGIN-MAKE TARGET=&(TARGET), "MAKE-AUFRUF"  
//MODIFY-MAKE-DEFAULTS LIBRARY=MYLIB "DEFAULT-EINSTELLUNGEN"  
/INCLUDE-PROCEDURE NAME=MAKEFILE "AUFRUF DES MAKEFILES"  
/END-BLOCK  
//END-MAKE "ENDE VON MAKE"  
...
```

Der Prozedurrahmen und die Datei MAKEFILE sind im Beispiel für die make-Funktionalität auf [Seite 508](#) abgebildet.

## make-Subanweisungen

Die Subanweisungen führen im Allgemeinen zu einer Prozedur, die alle Aktionen enthält, die notwendig sind, um das angegebene Ziel zu erstellen.

Übersicht der make-Subanweisungen

END-MAKE	Beenden der make-Subanweisungen
MODIFY-MAKE-DEFAULTS	Festlegen globaler Parameter
SET-DEPENDENCY	Festlegen von Abhängigkeiten zwischen Komponenten
SET-POSTPROCESSING	Festlegen von Aktionen, die nach der Folgeverarbeitung laufen
SET-PREPROCESSING	Festlegen von Aktionen, die vor der Folgeverarbeitung laufen
SET-STD-ACTION	Festlegen von Standard-Aktionen, die aus einem Elementtyp einen anderen erzeugen

### *Hinweis*

Als make-Subanweisungen sind auch die SDF-Standardanweisungen zulässig.

**END-MAKE****Beenden der make-Subanweisungen**

END-MAKE kennzeichnet das Ende der make-Subanweisungen. Erst jetzt wird die eigentliche Verarbeitung durchgeführt.

<b>END-MAKE</b>

Wurde bei BEGIN-MAKE für SUCCESS-PROCESSING synchrones Ausführen (\*INCLUDE-PROCEDURE, \*CALL-PROCEDURE) der erzeugten Prozedur angegeben, wird LMS beendet.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0712	Touch nicht möglich
2	0	LMS0714	Touch auf leere Datei nicht möglich
2	0	LMS0721	Das angegebene Ziel ist bereits aktuell
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**MODIFY-MAKE-DEFAULTS****Festlegen globaler Parameter**

Die globale Ablaufsteuerung für den make-Lauf von LMS erfolgt mit der MODIFY-MAKE-DEFAULTS-Anweisung. Mit MODIFY-MAKE-DEFAULTS können Werte immer wieder geändert werden. Die geänderten Werte gelten für die nachfolgenden Anweisungen.

Die MODIFY-MAKE-DEFAULTS-Operanden `...-VAR` definieren die make-S-Variablen, die vor jeder Aktion erneut gesetzt werden.

Die Deklaration der Variablen erfolgt durch LMS. Benutzerdeklarationen im Preprocessing werden nicht überschrieben.

**MODIFY-MAKE-DEFAULTS**

```
LIBRARY = *UNCHANGED / *NONE / <filename 1..54 without-vers> / *LINK(...)
```

```
  *LINK(...)
```

```
    | LINK-NAME = <structured-name 1..8>
```

```
,CURRENT-TARGET-VAR = *UNCHANGED / *NONE / <structured-name 1..20>
```

```
,FROM-OBJECTS-VAR = *UNCHANGED / *NONE / <structured-name 1..20>
```

```
,MODIFIED-OBJECTS-VAR = *UNCHANGED / *NONE / <structured-name 1..20>
```

```
,SUPPRESS-ERRORS = *UNCHANGED / *NONE / *ALL
```

**LIBRARY = \*UNCHANGED / \*NONE / <filename 1..54 without-vers> / \*LINK(...)**

Default-Bibliothek

Bibliothek, die bei Bibliothekselementen für \*MAKE-DEFAULT eingesetzt werden soll. Initialwert ist die Bibliothek, die in der BEGIN-MAKE-Anweisung für TARGET angegeben wurde (bei einer Datei als Ziel ist der Wert \*NONE).

**LIBRARY = \*NONE**

Keine Default-Bibliothek. Verwendung der Default-Bibliothek in den weiteren Anweisungen führt zu Fehler.

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

Angabe einer Default-Bibliothek.

**LIBRARY = \*LINK(...)**

Angabe einer Default-Bibliothek.

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

Dateikettungsname der Default-Bibliothek.



**CURRENT-TARGET-VAR = \*UNCHANGED / \*NONE / <structured-name 1..20>**  
S-Struktur-Variable, der das aktuelle Ziel zugewiesen wird.

**CURRENT-TARGET-VAR = \*NONE**  
Keiner S-Struktur-Variable soll das aktuelle Ziel zugewiesen werden.

**CURRENT-TARGET-VAR = <structured-name 1..20>**  
Der angegebenen S-Struktur-Variable soll das aktuelle Ziel zugewiesen werden.

**FROM-OBJECTS-VAR = \*UNCHANGED / \*NONE / <structured-name 1..20>**  
S-Listen-Variable, der alle Ausgangskomponenten zum aktuellen Ziel zugewiesen werden.

**FROM-OBJECTS-VAR = \*NONE**  
Keiner S-Listen-Variable sollen die Ausgangskomponenten zugewiesen werden.

**FROM-OBJECTS-VAR = <structured-name 1..20>**  
Der angegebene S-Listen-Variable sollen die Ausgangskomponenten zugewiesen werden.

**MODIFIED-OBJECTS-VAR = \*UNCHANGED / \*NONE / <structured-name 1..20>**  
S-Listen-Variable, der neuere Ausgangskomponenten zugewiesen werden. Das sind alle Ausgangskomponenten, die neuer sind als das aktuelle Ziel bzw. alle, wenn bei BEGIN-MAKE SELECT=\*ALL angegeben wurde.

**MODIFIED-OBJECTS-VAR = \*NONE**  
Keiner S-Listen-Variable sollen die neueren Komponenten zugewiesen werden.

**MODIFIED-OBJECTS-VAR = <structured-name 1..20>**  
Der angegebenen S-Listen-Variable sollen die neueren Ausgangskomponenten zugewiesen werden.

**SUPPRESS-ERRORS = \*UNCHANGED / \*NONE / \*ALL**  
Default für die Fehlerunterdrückung bei Aktionen.

**SUPPRESS-ERRORS = \*NONE**  
Fehler führen zum Abbruch der erzeugten Prozedur (Spinoff).

**SUPPRESS-ERRORS = \*ALL**  
Fehler im Ablauf der Aktion werden unterdrückt, führen also nicht zum Abbruch der erzeugten Prozedur. Bei Listen von Aktionen erfolgt die Fehlerunterdrückung nur am Ende. Die Prozedur wird nach der/den Aktion(en) zu dieser Abhängigkeit fortgesetzt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Beispiel*

Für den weiteren make-Lauf sollen die make-S-Variablen sowie die Standardbibliothek neu definiert werden. Die Variable CURT soll jeweils das bearbeitete Ziel, ALLOBJ alle Ausgangskomponenten enthalten.

```
//MODIFY-MAKE-DEFAULTS LIBRARY =BSPLIB,-  
CURRENT-TARGET-VAR=CURT, FROM-OBJECTS-VAR=ALLOBJ
```

**SET-DEPENDENCY****Festlegen von Abhängigkeiten zwischen Komponenten**

Die Festlegung der Abhängigkeiten zwischen Objekten erfolgt mit der SET-DEPENDENCY-Anweisung.

**SET-DEPENDENCY**

```

TARGET-OBJECT = *VARIABLE(...) / list-poss(2000): *LIBRARY-ELEMENT(...) /
    <filename 1..54 without-vers with-wild(80)>

    *VARIABLE(...)
        | VARIABLE-NAME = <composed-name 1..255>
    *LIBRARY-ELEMENT(...)
        | LIBRARY = *MAKE-DEFAULT / <filename 1..54 without-vers> / *LINK(...)
            | *LINK(...)
                | LINK-NAME = <structured-name 1..8>
            ,ELEMENT = <composed-name 1..64 with-under with-wild(132)>(…)
                <composed-name 1..64 with-under with-wild(132)>(…)
                    | VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under>
                    ,BASE = *STD / <composed-name 1..24 with-under with-wild>
                ,TYPE = <alphanum-name 1..8 with-wild(20)>
, FROM-OBJECT = *NONE / *VARIABLE(…) / list-poss(2000): *LIBRARY-ELEMENT(…) /
    <filename 1..54 without-vers with-wild-constr>

    *VARIABLE(…)
        | NAME = <composed-name 1..255>
    *LIBRARY-ELEMENT(…)
        | LIBRARY = *MAKE-DEFAULT / <filename 1..54 without-vers> / *LINK(…)
            | *LINK(…)
                | LINK-NAME = <structured-name 1..8>
            ,ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)
                <composed-name 1..132 with-under with-wildcard-constr>(…)
                    | VERSION = *HIGHEST-EXISTING / *UPPER-LIMIT / <composed-name 1..24 with-under>
                    ,BASE = *STD / <composed-name 1..24 with-under with-wild>
                ,TYPE = <alphanum-name 1..20 with-wild-constr>
, ACTION = *STD / list-poss(2000): <c-string 1..1800 with-low>
, SUPPRESS-ERRORS = *MAKE-DEFAULT / *NONE / *ALL

```

**TARGET-OBJECT = \*VARIABLE(...) / list-poss(2000): \*LIBRARY-ELEMENT(...) / <filename 1..54 without-vers with-wild(80)>**

Zielkomponente.

Ziel der Abhängigkeitsdefinition. Wildcards sind korrespondierend zu den Angaben im FROM-OBJECT-Operanden (siehe auch [Seite 139](#), Auswahlangabe bei make).

**TARGET-OBJECT = \*VARIABLE (...)**

Die Zielobjekte befinden sich in einer S-Listen-Variablen.

Die einzelnen Listenelemente müssen im Format von LMS vorliegen. Die Version \*HIGHEST-EXISTING wird durch \*HIGH-EXIST dargestellt.

**VARIABLE-NAME = <composed-name 1..255>**

Name der S-Listen-Variable.

**TARGET-OBJECT = list-poss(2000): \*LIBRARY-ELEMENT(...)**

Die Zielkomponente ist ein Bibliothekselement. Sie muss explizit eingegeben werden.

**LIBRARY = \*MAKE-DEFAULT / <filename 1..54 without-vers> / \*LINK(...)**

Bibliothek des Zielelements.

**LIBRARY = \*MAKE-DEFAULT**

Die durch die make-Subanweisung MODIFY-MAKE-DEFAULTS voreingestellte Bibliothek.

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

Bibliothek des Zielelements.

**LIBRARY = \*LINK(...)**

Bibliothek des Zielelements.

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

Linkname der Bibliothek des Zielelements.

**ELEMENT = <composed-name 1..64 with-under with-wild(132)>(…)**

Name des Zielelements (Wildcards: siehe [Seite 139](#), Auswahlangabe bei make).

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /**

**<composed-name 1..24 with-under>**

Version des Zielelements.

**VERSION = \*HIGHEST-EXISTING**

Es wird die höchste existierende Version überschrieben. Existiert keine, wird eine Defaultversion analog LMS erzeugt.

**VERSION = \*UPPER-LIMIT**

Es wird die höchstmögliche Version X'FF' erzeugt.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis des Zielelements fest.

**TYPE = <alphanum-name 1..8 with-wild(20)>**

Typ, den das Zielelement besitzt.

**TARGET-OBJECT = <filename 1..54 without-vers with-wild(80)>**

Ziel der Abhängigkeitsdefinition (Wildcards: siehe [Seite 139](#), Auswahlangabe bei make).

**FROM-OBJECT = \*NONE / \*VARIABLE(...) / list-poss(2000): \*LIBRARY-ELEMENT(...) / <filename 1..54 without-vers with-wild-constr>**

Ausgangskomponente der Abhängigkeitsdefinition. Die Wildcards sind korrespondierend zu den Angaben im TARGET-OBJECT-Operanden (Wildcards: siehe [Seite 139](#), Konstruktionsangabe bei make).

**FROM-OBJECT = \*NONE**

Es werden keine Komponenten vorausgesetzt. Die Zielkomponente ist nie aktuell.

**FROM-OBJECT = \*VARIABLE (...)**

Die vorausgesetzten Ausgangsobjekte befinden sich in einer S-Listen-Variablen. Die einzelnen Listenelemente müssen im Format von LMS vorliegen. Die Version \*HIGHEST-EXISTING wird durch \*HIGH-EXIST dargestellt.

**NAME = <composed-name 1..255>**

Name der S-Listen-Variable.

**FROM-OBJECT = list-poss(2000): \*LIBRARY-ELEMENT(...)**

Die Ausgangskomponenten sind Bibliothekselemente.

**LIBRARY = \*MAKE-DEFAULT / <filename 1..54 without-vers> / \*LINK(...)**

Bibliothek des Ausgangselements.

**LIBRARY = \*MAKE-DEFAULT**

Die durch die make-Subanweisung MODIFY-MAKE-DEFAULTS voreingestellte Bibliothek.

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

Bibliothek des Ausgangselements.

**LIBRARY = \*LINK(...)**

Bibliothek des Ausgangselements.

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

Linkname der Bibliothek des Ausgangselements.

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name des Ausgangselements (Wildcards: siehe [Seite 139](#), Konstruktionsangabe bei make).

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version des Ausgangselements.

**VERSION = \*HIGHEST-EXISTING**

Es wird die höchste existierende Version ausgewählt.

**VERSION = \*UPPER-LIMIT**

Es wird die höchstmögliche Version X'FF' ausgewählt.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis des Ausgangselements fest.

**TYPE = <alphanum-name 1..20 with-wild-constr>**

Typ, den das Ausgangselement besitzt.

**FROM-OBJECT = <filename 1..54 without-vers with-wild-constr>**

Ausgangskomponente der Abhängigkeitsdefinition (Wildcards: siehe [Seite 139](#), Konstruktionsangabe bei make).

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

Aktion, die ausgeführt wird, um die Zielkomponente zu erzeugen.

**ACTION = \*STD**

Die bei der make-Subanweisung SET-STD-ACTION für die Kombination aus Ausgangs- und Zieltyp spezifizierte Aktion wird ausgeführt. Sind für TARGET-OBJECT oder FROM-OBJECT Listen angegeben, wird jeweils der Typ des ersten Listenelements zur Bestimmung der Standard-Aktion herangezogen. Für Dateien können keine Standard-Aktionen spezifiziert werden.

**ACTION = list-poss(2000): <c-string 1..1800 with-low>**

Eine oder mehrere Aktionen, die zur Erzeugung der Zielkomponenten aus den Ausgangskomponenten dienen. Jede einzelne Aktion wird eine eigene Zeile in der durch BEGIN-MAKE erzeugten Prozedur.

**SUPPRESS-ERRORS = \*MAKE-DEFAULT / \*NONE / \*ALL**

Fehlerunterdrückung bei Aktionen.

**SUPPRESS-ERRORS = \*MAKE-DEFAULT**

Die durch MODIFY-MAKE-DEFAULTS voreingestellte Fehlerunterdrückung.

**SUPPRESS-ERRORS = \*NONE**

Fehler führen zum Abbruch der erzeugten Prozedur (Spinoff).

**SUPPRESS-ERRORS = \*ALL**

Fehler im Ablauf der Aktion werden unterdrückt, führen also nicht zum Abbruch der erzeugten Prozedur. Bei Listen von Aktionen erfolgt die Fehlerunterdrückung nur am Ende. Die Prozedur wird nach der/den Aktion(en) zu dieser Abhängigkeit fortgesetzt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweise*

- Das erste Ziel der ersten Abhängigkeitsdefinition kann in BEGIN-MAKE unter dem Namen \*FIRST-TARGET als Gesamtziel des make-Laufs bestimmt werden (Voreinstellung).
- Am Ende der Aktion sollte kein Programm geladen sein.

*Beispiel*

Die Phase PROG in der Standardbibliothek besteht aus den Bindemoduln TEIL1 und TEIL2. PROG wird mit Hilfe der Prozedur \*LIB(BSPLIB,LINK) erzeugt. Weiterhin sind die Bindemodule (Elementtyp R) aus gleichnamigen Sourcen (Typ S) mit Hilfe der Prozedur \*LIB(BSPLIB,COMPILE) erzeugt. Die Auswahlangabe \*LIB(\*,R) wirkt auf die im Programmsystem vorhandenen Elemente vom Typ R, also TEIL1 und TEIL2. Die Module sollen aus gleichnamigen Sourcen erzeugt werden. Die Variable ALLOBJ enthalte alle Ausgangskomponenten, die Variable CURT (für current target) das jeweils bearbeitete Ziel.

```
//SET-DEPENDENCY TARGET-OBJECT=PROG, -
// FROM-OBJECT=( *LIB(,TEIL1,R), *LIB(,TEIL2,R)), -
// ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,LINK).(OBJVAR=ALLOBJ) '
//SET-DEPENDENCY -
// TARGET-OBJECT=*LIB(*,R),-
// FROM-OBJECT=( *LIB(*,S), *LIB(,GLOBALDATA,M), *LIB(,HEADERS,M)), -
// ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,COMPILE), (&&(CURT.ELEM)) '
```

In einer dem UNIX-make ähnlichen, abgekürzten Schreibweise:

```
//SET-DEPENDENCY PROG, (*LIB(,TEIL1,R),*LIB(,TEIL2,R)), -
//      /CALL-PROCEDURE *LIB(BSPLIB,LINK),(OBJVAR=ALLOBJ)
//SET-DEPENDENCY *LIB(,*,R), *LIB(,*,S),-
//      /CALL-PROCEDURE *LIB(BSPLIB,COMPILE),(&&(CURT.ELEM))'
```

## SET-POSTPROCESSING

### Festlegen von Aktionen, die nach der Folgeverarbeitung laufen

Die Aktionen, die die erzeugte Prozedur abschließen, werden mit der Anweisung SET-POSTPROCESSING spezifiziert. Die Anweisung SET-POSTPROCESSING darf nur einmal angegeben werden.

<b>SET-POSTPROCESSING</b>
---------------------------

<b>ACTION</b> = list-poss(2000): <c-string 1..1800 with-low>
--------------------------------------------------------------

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

Die angegebenen Aktionen werden als letzte in die zu erzeugende Prozedur geschrieben.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

### *Hinweis*

Die Aktion wird nicht ausgeführt, wenn das Ziel des make-Laufs aktuell ist.

### *Beispiel*

In der erzeugten Prozedur soll am Ende das Kommando  
/CALL-PROCEDURE \*LIB(BSPLIB,STOP) ausgeführt werden.

```
//SET-POSTPROCESSING ACTION='/CALL-PROCEDURE *LIB(BSPLIB,STOP)'
```



**SET-PREPROCESSING****Festlegen von Aktionen, die vor der Folgeverarbeitung laufen**

Die Aktionen, die am Anfang der erzeugten Prozedur stehen sollen, werden mit der Anweisung SET-PREPROCESSING spezifiziert. Die Anweisung SET-PREPROCESSING darf nur einmal angegeben werden.

<b>SET-PREPROCESSING</b>
<b>ACTION</b> = list-poss(2000): <c-string 1..1800 with-low>
<b>SUPPRESS-ERRORS</b> = <u>*MAKE-DEFAULT</u> / *NONE / *ALL

**ACTION = list-poss(2000): <c-string 1..1800 with-low>**

Die angegebenen Aktionen werden als erste in die zu erzeugende Prozedur eingebracht.

**SUPPRESS-ERRORS = \*MAKE-DEFAULT / \*NONE / \*ALL**

Default für die Fehlerunterdrückung bei Aktionen.

**SUPPRESS-ERRORS = \*MAKE-DEFAULT**

Die durch MODIFY-MAKE-DEFAULTS eingestellte Fehlerunterdrückung.

**SUPPRESS-ERRORS = \*NONE**

Fehler führen zum Abbruch der erzeugten Prozedur (Spinoff).

**SUPPRESS-ERRORS = \*ALL**

Fehler im Ablauf der Aktion werden unterdrückt, führen also nicht zum Abbruch der erzeugten Prozedur. Bei Listen von Aktionen erfolgt die Fehlerunterdrückung nur am Ende. Die Prozedur wird nach der/den Aktion(en) zu dieser Abhängigkeit fortgesetzt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweis*

Die Aktion wird nicht ausgeführt, wenn das Ziel des make-Laufs aktuell ist.

*Beispiel*

In der Prozedur soll am Anfang das Kommando  
/CALL-PROCEDURE \*LIB(BSPLIB,INIT) ausgeführt werden.

```
//SET-PREPROCESSING ACTION='/CALL-PROCEDURE *LIB(BSPLIB,INIT)'
```

**SET-STD-ACTION****Festlegen von Standard-Aktionen**

In der Anweisung SET-STD-ACTION können Standard-Aktionen angegeben werden, mit denen aus Ausgangskomponenten eines bestimmten Typs Zielkomponenten eines anderen Typs erzeugt werden können. Auf diese Standard-Aktionen kann in der SET-DEPENDENCY-Anweisung Bezug genommen werden.

Die Anweisung SET-STD-ACTION kann nur für Abhängigkeiten zwischen Bibliothekselementen wirken und darf pro Kombination von Ausgangs- und Ziel-Typ nur einmal angegeben werden. Dateien können keine Standard-Aktionen zugewiesen werden, da sie keine Typen haben.

**SET-STD-ACTION**

```
TARGET-TYPE = <alphanum-name 1..8>
,FROM-TYPE = <alphanum-name 1..8>
,ACTION = list-poss(2000): <c-string 1..1800 with-low>
```

**TARGET-TYPE = <alphanum-name 1..8>**

Element-Typ der Zielkomponente.

**FROM-TYPE = <alphanum-name 1..8>**

Element-Typ der Ausgangskomponente.

**ACTION = list-poss(2000): <c-string 1..1800 with-low>**

Standard-Aktion zur Erzeugung von Komponenten des Ziel-Typs aus Komponenten des Ausgangs-Typs.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweis*

Am Ende der Aktion sollte kein Programm geladen sein.

*Beispiel*

In der SET-DEPENDENCY-Anweisung ohne explizite Aktion sollen Elemente vom Elementtyp S mit Hilfe der spezifizierten Standard-Aktionen Typ-R-Elemente erzeugen.

```
//SET-STD-ACTION -  
//  TARGET-TYPE=R, FROM-TYPE=S,-  
//  ACTION='/CALL-PROCEDURE *LIB(BSPLIB,COMPILE,(&&(CURT.ELEM)))'  
//SET-DEPENDENCY -  
//  TARGET-OBJECT=*LIB(*,R), FROM-OBJECT=*LIB(*,S)
```

In einer dem UNIX-make ähnlichen, abgekürzten Schreibweise (//SET-STD-ACTION entspricht der Verwendung der .Suffixes des UNIX-make:

```
//SET-STD-ACTION R,S,-  
//  '/CALL-PROCEDURE *LIB(BSPLIB,COMPILE),(&&(CURT.ELEM))'  
//SET-DEPENDENCY *LIB(*,R), *LIB(*,S)
```

## CALL-EDT

### Aufrufen des EDT

Die Anweisung CALL-EDT ruft das Dateibearbeitungsprogramm EDT auf und positioniert in die Arbeitsdatei 0.

LMS unterstützt EDT-Versionen ab V16.2A (siehe [10]).

Für die Bearbeitung von Unicode-Elementen wird EDT ab V17.0 benötigt (siehe [11]).

Beenden des EDT:

Der EDT kann mit RETURN oder HALT beendet werden, wobei die EDT-Daten (Dateien im virtuellen Speicher, Variablen,...) erhalten bleiben. Nur bei einem schwerwiegenden EDT-Fehler werden diese Daten freigegeben und es erfolgt eine LMS-Meldung.

#### CALL-EDT

```
EDT-MODE = *LMS-DEFAULT / *COMPATIBLE / *UNICODE
,EDITOR-COMMANDS = *NONE / <c-string 1..251> / *LOWER-CASE(...)
  *LOWER-CASE(...)
    | EDITOR-COMMANDS = <c-string 1..251 with-low>
```

#### EDT-MODE = \*LMS-DEFAULT / \*COMPATIBLE / \*UNICODE

Legt den Betriebsmodus fest, in dem der EDT aufgerufen wird.

#### EDT-MODE = \*COMPATIBLE

Der EDT wird im Kompatibilitätsmodus aufgerufen.

#### EDT-MODE = \*UNICODE

Der EDT wird im Unicode-Modus aufgerufen.

#### EDITOR-COMMANDS = \*NONE / <c-string 1..251> / \*LOWER-CASE(...)

Es ist keine Editor-Kommandofolge spezifiziert.

#### EDITOR-COMMANDS = <c-string 1..251>

Editor-Kommandofolge, die nach Aufruf des EDT ausgeführt wird. Die einzelnen Kommandos müssen durch „;“ getrennt eingegeben werden. Mit Ausnahme von EDIT und RETURN dürfen alle Kommandos angegeben werden, die sowohl im F-Modus als auch im L-Modus des EDT zulässig sind. Das Kommando EDIT ist nur in der Form EDIT ONLY erlaubt. Zu beachten ist, dass das Kommando HALT in der EDT-Kommandofolge das Beenden des EDT und somit die Freigabe der EDT-Daten (Dateien im virtuellen Speicher, Variablen,...) bedeutet.

Kleinbuchstaben werden in Großbuchstaben umgesetzt.

**EDITOR-COMMANDS = \*LOWER-CASE(...)**

Keine Umsetzung der Kleinbuchstaben.

**EDITOR-COMMANDS = <c-string 1..251 with-low>**

Editor-Kommandofolge, wie oben, nur dass Kleinbuchstaben nicht in Großbuchstaben umgesetzt werden.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0297	Wechsel des EDT-Betriebsmodus nicht möglich
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Hinweise*

Nach CALL-EDT können durch die EDT-Anweisung OPEN F= oder OPEN L= Dateien oder Elemente geöffnet und in EDT-Arbeitsbereiche eingelesen werden. Diese Objekte bleiben auch nach Rückkehr zu LMS durch HALT oder RETURN gesperrt. Erst mit Beenden von LMS wird der OPEN-Indikator zurückgesetzt. Wenn allerdings vor der Eingabe von HALT das Kommando CLOSE abgesetzt wird, wird das Element oder die Datei zurückgeschrieben und geschlossen. Ein Zugriff von einer anderen Task ist dann möglich, ohne LMS zu beenden.

Wenn der EDT-MODE explizit in der Anweisung CALL-EDT festgelegt ist, aber der EDT in diesem Modus nicht starten kann, wird die Anweisung abgebrochen und die Meldung LMS0297 ausgegeben.

Wenn der EDT-MODE nicht explizit in der Anweisung CALL-EDT festgelegt ist, startet der EDT im EDT-Default-Modus, kann bei Bedarf jedoch selbst den Modus wechseln. Der EDT-Default-Modus ist zu Beginn des LMS-Laufs COMPATIBLE und kann mit der Anweisung MODIFY-LMS-DEFAULTS geändert werden.

## CLOSE-LIBRARY

### Schließen einer Bibliothek

Diese Anweisung schließt die angegebene(n) Bibliothek(en).

Liegt die LMS-Ausgabe auf einem Bibliothekselement, so wird die zugehörige Bibliothek nicht geschlossen und eine Fehlermeldung wird ausgegeben.

Wird diese Anweisung ohne Parameter angegeben, werden alle offenen Bibliotheken geschlossen.

#### CLOSE-LIBRARY

```
LIBRARY = *ALL / *STD / <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
```

#### LIBRARY = \*ALL / \*STD/ <filename 1..54 without-vers> / \*LINK(...)

Angabe derjenigen Bibliothek bzw. Bibliotheken, die geschlossen werden sollen.

#### LIBRARY = \*ALL

Es werden alle offenen Bibliotheken geschlossen.

#### LIBRARY = \*STD

Die durch OPEN-LIBRARY geöffnete Bibliothek wird geschlossen.

#### LIBRARY = <filename 1..54 without-vers>

Name der Bibliothek, die geschlossen werden soll.

#### LIBRARY = \*LINK(...)

Die über einen Dateikettungsname zugewiesene Bibliothek.

#### LINK-NAME = <structured-name 1..8>

Dateikettungsname der Bibliothek, der vor dem LMS-Lauf mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
1	0	CMD0001	Ohne Fehler
	0	LMS0036	Bibliothek nicht zugewiesen
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Beispiele*

- Alle offenen Bibliotheken werden geschlossen.  
`//CLOSE-LIBRARY`
- Die Bibliothek, die über den Dateikettungsnamen LIB2 zugewiesen wurde, wird geschlossen.  
`//CLOSE-LIBRARY LIBRARY=*LINK(LINK-NAME=LIB2)`

## COMPARE-ELEMENT

### Vergleichen von Elementen

Mit COMPARE-ELEMENT werden Elemente Satz für Satz verglichen, wobei über den Operanden RECORD-PART der Vergleichsbereich festgelegt wird. Die dabei festgestellten Unterschiede werden in einem Vergleichsprotokoll und einer Vergleichsstatistik dargestellt. Die Elemente können sich in verschiedenen Bibliotheken befinden.

Die Anweisung COMPARE-ELEMENT wird auch dann ausgeführt, wenn nur eines der Vergleichselemente in den angegebenen Bibliotheken gefunden wird. Dies ermöglicht das Zählen der Sätze in Elementen.

Werden zwei Elemente miteinander verglichen, verwendet LMS die Begriffe Primärelement und Sekundärelement. Dadurch steht dem Anwender frei, ob er das neue oder das alte Element als Basiselement wählt. LMS betrachtet immer das Sekundärelement als Basis des Vergleichs. D.h. LMS kennzeichnet fehlende Sätze im Sekundärelement als eingefügte Sätze und fehlende Sätze im Primärelement als gelöschte Sätze.

Der Basistyp für Primär- und Sekundärelement darf unterschiedlich sein, wenn Textelemente verglichen werden.

Mit dem Operanden COMPARE-PARAMETERS wird die Vergleichsart (formaler oder logischer Vergleich) und die Protokollmenge (Umfang und Format) festgelegt.

COMPARE-ELEMENT erstellt immer eine Vergleichsstatistik in den internen Speicher C0. Nach Ausführung der Anweisung wird C0 auf den Speicher C1 addiert. Der Speicher C0 wird vor jeder COMPARE-ELEMENT-Anweisung neu initialisiert. Die Speicher können mit der Anweisung SHOW-STATISTICS ausgegeben werden.

Mit ACTIVATE-USER-EXIT kann über ein Benutzerunterprogramm vor dem eigentlichen Vergleich auf die Elementsätze zugegriffen werden.



**COMPARE-ELEMENT****PRIMARY-ELEMENT** = \*LIBRARY-ELEMENT (...)**\*LIBRARY-ELEMENT**(...)**LIBRARY** = \*STD / <filename 1..54 without-vers> / **\*LINK**(...)**\*LINK**(...)**LINK-NAME** = <structured-name 1..8>**,ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**\*ALL**(...)**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

&lt;composed-name 1..64 with-under with-wild(132)&gt;(...)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>**,TYPE** = \*LMS-DEFAULT / **\*ALL** / <alphanum-name 1..8 with-wild(20)>**,USER-DATE** = \*ANY / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)**\*INTERVAL**(...)**FROM** = 1900-01-01 / <date 8..10 with-compl>**,TO** = \*TODAY / <date 8..10 with-compl>**,CREATION-DATE** = \*ANY / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)**\*INTERVAL**(...)**FROM** = 1900-01-01 / <date 8..10 with-compl>**,TO** = \*TODAY / <date 8..10 with-compl>**,MODIFICATION-DATE** = \*ANY / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)**\*INTERVAL**(...)**FROM** = 1900-01-01 / <date 8..10 with-compl>**,TO** = \*TODAY / <date 8..10 with-compl>

(Teil 1 von 3)



<pre> <b>,ELEMENT</b> = <b>*BY-SOURCE</b> (...) / <b>*ALL</b>(...) /               &lt;composed-name 1..132 with-under with-wild-constr&gt;(…)  <b>*BY-SOURCE</b>(…)         <b>VERSION</b> = <b>*HIGHEST-EXISTING</b> / <b>*BY-SOURCE</b> / <b>*ALL</b> / <b>*UPPER-LIMIT</b> /               &lt;composed-name 1..52 with-under with-wild-constr&gt;         <b>,BASE</b> = <b>*STD</b> / &lt;composed-name 1..24 with-under with-wild&gt;     <b>*ALL</b>(…)         <b>VERSION</b> = <b>*HIGHEST-EXISTING</b> / <b>*BY-SOURCE</b> / <b>*ALL</b> / <b>*UPPER-LIMIT</b> /               &lt;composed-name 1..52 with-under with-wild-constr&gt;         <b>,BASE</b> = <b>*STD</b> / &lt;composed-name 1..24 with-under with-wild&gt;         &lt;composed-name 1..132 with-under with-wild-constr&gt;(…)         <b>VERSION</b> = <b>*HIGHEST-EXISTING</b> / <b>*BY-SOURCE</b> / <b>*ALL</b> / <b>*UPPER-LIMIT</b> /               &lt;composed-name 1..52 with-under with-wild-constr&gt;         <b>,BASE</b> = <b>*STD</b> / &lt;composed-name 1..24 with-under with-wild&gt;     <b>,TYPE</b> = <b>*BY-SOURCE</b> / <b>*LMS-DEFAULT</b> / <b>*ALL</b> / &lt;alphanum-name 1..20 with-wild-constr&gt;  <b>,COMPARE-PARAMETERS</b> = <b>*LMS-DEFAULT</b> / <b>*PARAMETERS</b>(…)  <b>*PARAMETERS</b>(…)         <b>RECORD-PART</b> = <b>*LMS-DEFAULT</b> / <b>*ALL</b> / <b>*PART</b>(…)         <b>*PART</b>(…)             <b>START</b> = <b>*LMS-DEFAULT</b> / &lt;integer 1..32764&gt;             <b>,LENGTH</b> = <b>*LMS-DEFAULT</b> / <b>*REST</b> / &lt;integer 1..32764&gt;             <b>,SPACES</b> = <b>*LMS-DEFAULT</b> / <b>*STD</b> / <b>*IGNORED</b> / <b>*RELEVANT</b>             <b>,INFORMATION</b> = <b>*LMS-DEFAULT</b> / <b>*MEDIUM</b> / <b>*MINIMUM</b> / <b>*MAXIMUM</b> / <b>*SUMMARY</b> / <b>*STATISTICS</b>                 <b>/ *NONE</b>             <b>,LAYOUT</b> = <b>*LMS-DEFAULT</b> / <b>*COMPATIBLE</b> / <b>*COMPRESSED</b>             <b>,JOIN-ELEMENT-SETS</b> = <b>*LMS-DEFAULT</b> / <b>*NO</b> / <b>*YES</b>     <b>,TEXT-OUTPUT</b> = <b>*LOGGING-PARAMETERS</b> / <b>*NONE</b> / <b>*SYSOUT</b> / <b>*SYSLST</b>(…) / <b>*EDT</b>(…)  <b>*SYSLST</b>(…)         <b>SYSLST-NUMBER</b> = <b>*STD</b> / &lt;INTEGER 1..99&gt;     <b>*EDT</b>(…)         <b>WRITE-MODE</b> = <b>*EXTEND</b> / <b>*REPLACE</b>  <b>,STRUCTURE-OUTPUT</b> = <b>*SYSINE</b> / <b>*NONE</b> / &lt;composed-name 1..255&gt;(…)          &lt;composed-name 1..255&gt;(…)         <b>WRITE-MODE</b> = <b>*REPLACE</b> / <b>*EXTEND</b> </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Teil 3 von 3)

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

Angaben des ersten Vergleichselementes (Primärelement).

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Primärelement befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Primärelement befindet.

**LIBRARY = \*LINK(..)**

Die über einen Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem LMS-Lauf mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL (...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Alle Elemente der Bibliothek werden zum Vergleich herangezogen bzw. der Name des Elementes, das als Primärelement verwendet wird.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das Primärelement besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. der Basis wird als Primärelement verwendet.

**VERSION = \*ALL**

Alle Versionen werden beim Vergleich berücksichtigt.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version 'X'FF' wird als Primärelement verwendet.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das als Primärelement verwendet wird.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Alle Typen werden beim Vergleich berücksichtigt.

**TYPE = <alphanum-name 1..8 with-wild(20)>**

Typ, den das Primärelement besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das Primärelement besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird als Primärelement verwendet.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird als Primärelement verwendet.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden als Primärelement verwendet.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe oben bei USER-DATE.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe oben bei USER-DATE.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die nicht als Primärelement verwendet werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe \*LIBRARY-ELEMENT.

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

Angaben des zweiten Vergleichselementes (Sekundärelement). Das hier ausgewählte Element wird als Basis des Vergleichs hergenommen.

**LIBRARY = \*STD / \*BY-SOURCE / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Sekundärelement befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

**LIBRARY = \*BY-SOURCE**

Das Sekundärelement befindet sich in der selben Bibliothek wie das Primärelement.

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

Name der Bibliothek, in der sich das Sekundärelement befindet.

**LIBRARY = \*LINK(...)**

Die über einen Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem LMS-Lauf mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*BY-SOURCE(...) / \*ALL(...) /**

**<composed-name 1..132 with-under with-wild-constr>(...)**

Name des Elementes, das als Sekundärelement verwendet wird.

**VERSION = \*HIGHEST-EXISTING / \*BY-SOURCE / \*ALL / \*UPPER-LIMIT /**

**<composed-name 1..52 with-under with-wild-constr>**

Version, die das Sekundärelement besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. der Basis wird als Sekundärelement verwendet.

**VERSION = \*BY-SOURCE**

Die Version des Sekundärelementes ist gleich der Version des Primärelementes oder X'FF', falls sie nicht existiert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird als Sekundärelement verwendet.

**VERSION = <composed-name 1..52 with-under with-wild-constr>**

Explizite Angabe der Version für das Element, das als Sekundärelement verwendet wird.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Sekundärelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name des Elementes, das als Sekundärelement verwendet wird.

**VERSION = \*HIGHEST-EXISTING / \*BY-SOURCE / \*ALL / \*UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>**

Version, die das Sekundärelement besitzt.

Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / \*ALL /**

**<alphanum-name 1..20 with-wild-constr>**

Typ, den das Sekundärelement besitzt.

**TYPE = \*BY-SOURCE**

Das Sekundärelement besitzt den gleichen Typ wie das Primärelement.

**COMPARE-PARAMETERS = \*LMS-DEFAULT / \*PARAMETERS(…)**

Bestimmt die Vergleichsparameter. Die Art des Vergleichs (formal oder logisch), Umfang und Format der Protokollierung kann festgelegt werden.

**RECORD-PART = \*LMS-DEFAULT / \*ALL / \*PART(…)**

Bestimmt den Vergleichsbereich im Satz.

**RECORD-PART = \*ALL**

Der ganze Satz wird verglichen.

**RECORD-PART = \*PART(…)**

Bereichsangabe für den zu vergleichenden Teil des Satzes.

**START = <integer 1..32764>**

Bereichsanfang für den zu vergleichenden Teil des Satzes. Wird kein Wert eingetragen, wird der Satz von Anfang an genommen.

**LENGTH = \*REST / <integer 1..32764>**

Länge des zu vergleichenden Bereichs im Satz. Wird kein Wert eingetragen, wird der Satz vom Startanfang genommen.

**SPACES = \*LMS-DEFAULT / \*STD / \*IGNORED / \*RELEVANT**

Behandlung von Leerzeichen im Satz.

**SPACES = \*STD**

Wirkt für Textelemente wie \*IGNORED, sonst wie \*RELEVANT.

**SPACES = \*IGNORED**

Logischer Vergleich. Die Vergleichsfelder werden Zeichen für Zeichen verglichen, Zwischenräume werden übergangen.

**SPACES = \*RELEVANT**

Formaler Vergleich. Die Vergleichsfelder werden zuerst auf gleiche Länge überprüft. Sind die Längen gleich, werden die Felder vollständig verglichen. Sind die Längen unterschiedlich, werden die Sätze als ungleich protokolliert.

**INFORMATION = \*LMS-DEFAULT / \*MEDIUM / \*MINIMUM / \*MAXIMUM / \*SUMMARY / \*STATISTICS / \*NONE**

Umfang der Protokollierung.

**INFORMATION = \*MEDIUM**

Standard-Vergleichsprotokoll. Der Vergleichsbereich der ungleichen Sätzen wird vollständig protokolliert. Bei gleichen Sätzen werden nur Bereichsangaben (Satznummern) protokolliert. Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*MINIMUM**

Minimales Vergleichsprotokoll. Für gleiche und ungleiche Sätze werden nur Bereichsangaben (Satznummern) protokolliert. Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*MAXIMUM**

Ausführliches Vergleichsprotokoll. Alle Sätze werden protokolliert. Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*SUMMARY**

Kein Vergleichsprotokoll. Nur die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*STATISTICS**

Kein Vergleichsprotokoll. Die Vergleichsstatistik wird in komprimierter Form ausgegeben. Die Ausgabe ist auf Zeilen der Länge 132 ausgerichtet.

**INFORMATION = \*NONE**

Keine Protokollierung, weder Vergleichsprotokoll noch Vergleichsstatistik. \*NONE ist nur sinnvoll bei Verwendung der SHOW-STATISTICS-Anweisung.

**LAYOUT = \*LMS-DEFAULT / \*COMPATIBLE / \*COMPRESSED**

Format der Protokollierung.

**LAYOUT = \*COMPATIBLE**

Das Vergleichsprotokoll wird im Standardformat ausgegeben. Dieses Format ist kompatibel zu früheren LMS-Versionen.

**LAYOUT = \*COMPRESSED**

Das Vergleichsprotokoll wird in einem komprimierten Format ausgegeben.

**JOIN-ELEMENT-SETS = \*LMS-DEFAULT / \*NO / \*YES**

Legt die Elementmenge fest, die verglichen werden soll.

**JOIN-ELEMENT-SETS = \*NO**

Es werden nur die Primärelemente und die durch Konstruktion ermittelten Sekundärelemente zum Vergleich herangezogen.



**JOIN-ELEMENT-SETS = \*YES**

Es werden alle Primär- und Sekundärelemente zum Vergleich herangezogen.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(…)**

Strukturierte Ausgabe.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch /ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe [12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variable die strukturierte Ausgabe erfolgen soll. Diese Variable muss als dynamische Listenvariable deklariert sein.

(Kommando: DECLARE-VARIABLE NAME=... (TYPE=\*STRUCTURE),  
MULTIPLE-ELEMENTS=\*LIST)

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
2	0	LMS0201	Nur der Vergleichsbereich wird protokolliert
2	0	LMS0313	Überlauf eines Statistikzählers
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0412	Element gesperrt

### Hinweise

- Der Operand INFORMATION beeinflusst die strukturierte Ausgabe nicht.

- Für jeden Vergleich zweier Elemente wird ein Listenelement erzeugt. Die einzelnen Variablen-Elemente sind im [Kapitel „Format der LMS-Ausgaben in S-Variable“](#) beschrieben (siehe [Seite 455](#)).
- Bei der Vergleichsstatistik ist der Maximalwert für Elementzähler 99 999, der für Zeilenzähler ist 999 999 999. Falls die Grenze von 9 Ziffern überschritten wird, wird die Meldung LMS0313 ausgegeben und die betroffenen Zähler zählen Modulo  $10^9$  weiter.

### Notwendige Zugriffsrechte

Bei PRIMARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

Bei SECONDARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

#### *Beispiel*

Die Elemente TEST1 der Bibliotheken BIBU und PLIB werden verglichen. Der Vergleichsbereich ist das 5. bis 30. Byte des Elementsatzes.

```
/ADD-FILE-LINK FILE=BIBU;LINK-NAME=LIB%
/START-LMS
//OPEN-LIBRARY LIBRARY=PLIB
//COMPARE-ELEMENT PRIMARY-ELEMENT=*LIB-ELEM(ELEM=TEST1,TYPE=S),-
    SECONDARY-ELEMENT=*LIB-ELEM(LIB=*LINK=LIB5)), -
    COMPARE-PARAMETERS=*PARAMETERS(INFORMATION=*MAXIMUM, -
    RECORD-PART=*PART(START=5;LENGTH=26))
...
//END
```

## COPY-ELEMENT

### Elemente kopieren

COPY-ELEMENT kopiert Elemente und Bibliotheken im Verhältnis eins zu eins. Die kopierten Elemente können neue Elementbezeichnungen erhalten. Der Basistyp für Quell- und Zielelement darf unterschiedlich sein, wenn Textelemente kopiert werden.

Es gibt folgende Kopiermöglichkeiten:

- Kopieren von einem oder mehreren Elementen in die gleiche Bibliothek
- Kopieren von einem oder mehreren Elementen in eine andere Bibliothek
- Kopieren einer kompletten Bibliothek (siehe Beispiel [Seite 232](#))

Die kopierten Elemente können entweder als Voll-Element oder als Delta-Element gespeichert werden. Falls beim Kopieren von Delta-Elementen die Eingabebibliothek mit der Ausgabebibliothek identisch ist, müssen die kopierten Delta-Elemente neue Elementnamen erhalten. Blätter von Delta-Bäumen dürfen überschrieben werden.

Das Quellelement kann mit DELETE-SOURCE = \*YES nach dem Kopieren gelöscht werden. Damit kann ein Element verschoben werden.

### Strukturerhaltendes Kopieren (STORAGE-FORM = \*BY-SOURCE)

LMS erkennt bei diesem Format die Form der Elementabspeicherung in den PLAM-Bibliotheken. Dementsprechend werden Delta-Bäume als Delta-Bäume und alle andere Elemente als Vollelemente in die Ausgabedatei kopiert.

#### *Hinweise*

- Wird der Kopiervorgang vorzeitig abgebrochen, bleibt der schon kopierte Teil eines Delta-Baumes erhalten.
- VERSION, BASE und alle Datumsoperanden müssen auf ihren Default-Wert gesetzt sein. Es wird immer der komplette Namensraum, d.h. wie Version=\*, kopiert.
- Die Angabe von Bibliothekslisten führt auf Fehler.

#### *Beispiel*

In der Bibliothek sind die Elemente A/1 und A/2 enthalten. Dann wird durch die Anweisung „Kopiere Element A nach B“ bei Angabe von STORAGE-FORM=\*STD das Element B/2 erzeugt und bei Angabe von STORAGE-FORM=\*BY-SOURCE die Elemente B/1 und B/2.

## Überschreiben des Zielnamensraumes (WRITE-MODE=\*SUBSTITUTE)

Mit WRITE-MODE=\*SUBSTITUTE kann erreicht werden, dass das kopierte Element in der Zielbibliothek das einzige Element mit gleichem Typ und Namen ist. LMS löscht vor dem Kopieren in der Zielbibliothek alle Elemente mit dem Typ und dem Namen des Zielelementes. D.h. alle Benutzerangaben bei TO-ELEMENT (wie z.B. VERSION = \*INCREMENT) wirken erst auf den leeren Zielnamensraum. Bei \*INCREMENT wird z.B. immer die Defaultversion erzeugt.

### Einschränkungen

1. Der Operand STORAGE-FORM= darf nicht den Wert \*BY-SOURCE haben.
2. Eingabe- und Ausgabebibliothek müssen unterschiedlich sein.

Tritt während des Löschens des Zielnamensraumes ein Fehler auf (z.B. Element schreibgeschützt), so wird die COPY-ELEMENT-Anweisung abgebrochen.

COPY-ELEMENT
<pre> ELEMENT = *<u>LIBRARY-ELEMENT</u> (...)   *LIBRARY-ELEMENT(...)     LIBRARY = *<u>STD</u> / &lt;filename 1..54 without-vers&gt; / *LINK(...)       *LINK(...)           LINK-NAME = &lt;structured-name 1..8&gt;       ,ELEMENT = *<u>ALL</u>(...) / &lt;composed-name 1..64 with-under with-wild(132)&gt;(…)         *<u>ALL</u>(…)             <u>VERSION</u> = *<u>HIGHEST-EXISTING</u> / *<u>ALL</u> / *<u>UPPER-LIMIT</u> /               &lt;composed-name 1..24 with-under with-wild(52)&gt;           ,<u>BASE</u> = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt;             &lt;composed-name 1..64 with-under with-wild(132)&gt;(…)             <u>VERSION</u> = *<u>HIGHEST-EXISTING</u> / *<u>ALL</u> / *<u>UPPER-LIMIT</u> /               &lt;composed-name 1..24 with-under with-wild(52)&gt;           ,<u>BASE</u> = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt;         ,<u>TYPE</u> = *<u>LMS-DEFAULT</u> / *<u>ALL</u> / &lt;alphanum-name 1..8 with-wild(20)&gt; </pre>

(Teil 1 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    | *ANY(...)
    | |
    | | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    | | <composed-name 1..24 with-under with-wild(52)>
    | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    | | <composed-name 1..64 with-under with-wild(132)>(…)
    | | |
    | | | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    | | | <composed-name 1..24 with-under with-wild(52)>
    | | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    | | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
    | | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    | | *INTERVAL(...)
    | | |
    | | | FROM = 1900-01-01 / <date 8..10 with-compl>
    | | | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)

```

,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,TO-ELEMENT = *LIBRARY-ELEMENT (...)
    *LIBRARY-ELEMENT(...)
        |
        | LIBRARY = *STD / *BY-SOURCE / <filename 1..54 without-vers> / *LINK(...)
        | *LINK(...)
        | | LINK-NAME = <structured-name 1..8>
,ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wild-constr>(…)
    *BY-SOURCE(…)
        |
        | VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
        | *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
        | <composed-name 1..132 with-under with-wild-constr>(…)
        | | VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
        | | *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
        | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
,TYPE = *BY-SOURCE / *LMS-DEFAULT / <alphanum-name 1..20 with-wild-constr>
,USER-DATE = *BY-SOURCE / *TODAY / <date 8..10 with-compl>
,STORAGE-FORM = *LMS-DEFAULT / *BY-SOURCE / *STD / *FULL / *DELTA
,PROTECTION = *LMS-DEFAULT / *STD / *BY-SOURCE
,DELETE-SOURCE = *NO / *YES
,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *EXTEND / *SUBSTITUTE / *ANY
,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich die zu kopierenden Elemente befinden.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, aus der die Elemente kopiert werden sollen.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL (...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, das kopiert werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das zu kopierende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird kopiert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird kopiert.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*.

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das zu kopierende Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das zu kopierende Element besitzt ein beliebiges Datum.



**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird kopiert.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird kopiert.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden kopiert.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl >**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element vom Kopieren ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die vom Kopieren ausgeschlossen sind. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

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

Angabe, wohin und unter welchem Namen das Element aufgenommen werden soll.

**LIBRARY = \*STD / \*BY-SOURCE / <filename 1..54 without-vers> / \*LINK(...)**

Angabe des neuen Bibliotheksnamens oder Bibliothek, in die das Element aufgenommen werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

**LIBRARY = \*BY-SOURCE**

Das Element wird in die Bibliothek kopiert, in der sich auch das zu kopierende Element befindet.

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

Name der Bibliothek, in die die Datei als Element aufgenommen werden soll. Existiert die Bibliothek noch nicht, wird sie neu eingerichtet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*BY-SOURCE(...) /****<composed-name 1..132 with-under with-wild-constr>(…)**

Name, den das neu aufzunehmende Element erhalten soll.

**ELEMENT = \*BY-SOURCE(...)**

Der neue Name entspricht dem alten Namen.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /****<composed-name 1..52 with-under with-wild-constr>**

Version, die das neu aufzunehmende Element erhalten soll.

**VERSION = \*BY-SOURCE**

Das neue Element erhält die gleiche Version wie das ursprüngliche Element. Hat das ursprüngliche Element keine Versionsangabe, erhält das neue Element X'FF' als Versionsangabe.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion (siehe auch Abschnitt „Versionskonventionen“ [Seite 57](#)).

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

**VERSION = <composed-name 1..52 with-under with-wild-constr>**

Das Zielelement erhält die hier angegebene Version.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name des neu aufzunehmenden Elementes. Er kann auch mit Wildcards eingegeben werden.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /**

**<composed-name 1..52 with-under with-wild-constr>**

Version, die das neu aufzunehmende Element erhalten soll.

Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / <alphanum-name 1..20 with-wild-constr>**

Typ, den das neu aufzunehmende Element erhalten soll.

**TYPE = \*BY-SOURCE**

Das neue Element erhält die gleiche Typbezeichnung wie das ursprüngliche Element.

**USER-DATE = \*BY-SOURCE / \*TODAY / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*BY-SOURCE**

Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**STORAGE-FORM = \*LMS-DEFAULT / \*BY-SOURCE / \*STD / \*FULL / \*DELTA**

Speicherform für das anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*BY-SOURCE**

VERSION, BASE und alle Datumsoperanden müssen auf ihren Standardwert gesetzt sein. Es wird immer der komplette Namensraum, d.h. wie Version=\*, kopiert.

Das neu aufzunehmende Element wird strukturerhaltend kopiert, d.h. Delta-Bäume werden als solche wieder abgelegt und Vollelemente werden als Vollelemente übernommen. Werden Delta-Bäume strukturerhaltend kopiert, darf der Zielname im Zieltyp nicht existieren (d.h. WRITE-MODE wirkt nicht).

**STORAGE-FORM = \*STD**

Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**

Das neue Element wird als Voll-Element angelegt (Fehler, falls nicht zulässig).

**STORAGE-FORM = \*DELTA**

Das neue Element wird als Delta-Element angelegt (Fehler, falls nicht zulässig). Zulässige Elementtypen: S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**PROTECTION = \*LMS-DEFAULT / \*STD / \*BY-SOURCE**

Elementschutz für das kopierte Element.

**PROTECTION = \*STD**

Existiert das Element bereits, so bleibt der Elementschutz unverändert. Existiert das Element noch nicht und ist für die Bibliothek bzw. den Typ des Elementes ein initialer Elementschutz festgelegt, so erhält das kopierte Element diesen Schutz.

**PROTECTION = \*BY-SOURCE**

Das kopierte Element erhält den selben Schutz wie das ursprüngliche Element.

**DELETE-SOURCE = \*NO / \*YES**

Hier kann der Benutzer bestimmen, ob das ursprüngliche Element erhalten bleiben oder gelöscht werden soll.

**DELETE-SOURCE = \*NO**

Das ursprüngliche Element wird nicht gelöscht.

**DELETE-SOURCE = \*YES**

Das ursprüngliche Element wird gelöscht.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*EXTEND / \*SUBSTITUTE / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

Ist das zu speichernde Element ein Delta-Element, ist darauf zu achten, dass das Element ein Blatt des Delta-Baumes sein muss. Nur Blätter eines Delta-Baumes dürfen überschrieben werden.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*EXTEND**

Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt. \*EXTEND ist nicht für Delta-Elemente erlaubt.

**WRITE-MODE = \*SUBSTITUTE**

Alle Elemente mit Typ und Namen des Zielelementes werden in der Zielbibliothek gelöscht. Danach wird das Quellelement kopiert.

**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei LIBRARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgenommen.

Beim Kopieren von Deltas mit STORAGE-FORM=\*BY-SOURCE wird der Delta-Baum nur kopiert, wenn für alle seine Elemente Leserecht besteht.

Bei TO-ELEMENT: Lese- und Schreibrecht für LIBRARY

Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).

Bei STORAGE-FORM=\*DELTA muss für das durch BASE bestimmte Element Leserecht gegeben sein.

Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Es kann nur eine neue Version erzeugt oder diese Basisversion überschrieben werden, sofern Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

Bei PROTECTION=\*BY-SOURCE kann nur der Eigentümer der Bibliotheksdatei diese Funktionalität nutzen.

### *Hinweise*

- Beim Anlegen eines Elements ist die unter dem Typ geltende Konvention zu beachten. Insbesondere dann, wenn der Zieltyp die Konvention STD-TREE hat, können Probleme auftreten, wenn auf der Quellseite Nebenzweigversionen existieren, deren zugehörige Hauptzweigversion gelöscht ist. In diesem Fall können die betreffenden Nebenzweige nicht kopiert werden; es erfolgt jedoch eine Fehlermeldung von LMS.
- Ausnahme: Das Kopieren eines kompletten Delta-Baumes mit STORAGE-FORM=\*BY-SOURCE ist immer möglich.
- STATE und HOLDER des Quellelements werden nicht auf das Zielelement übernommen, auch nicht bei PROTECTION=\*BY-SOURCE.
- Bei aktivem WRITE-CONTROL in der Ausgabebibliothek wird dem zu schreibenden Element von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorganges protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Bereits vorhandene Kommentarsätze werden dahinter kopiert. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), so werden die Elementattribute STATE, HOLDER und alle Rechte von der Basisversion auf die neue Version übernommen. Die CCSN wird von dem Quellelement übernommen. Das USER-DATE wird neu bestimmt.

### *Beispiele mit WRITE-MODE=\*SUBSTITUTE*

- In der Eingabebibliothek X ist unter dem Typ S das Element A/1 enthalten. In der Ausgabebibliothek Y ist unter Typ S das Element A/2 enthalten.

```
//COPY-ELEM ELEMENT= *LIB(LIB=X,ELEM=A,TYPE=S),-
      TO ELEMENT= *LIB(LIB=Y),WRITE-MODE=*SUBSTITUTE
```

Nach dieser Anweisung ist in der Ausgabebibliothek unter Typ S und Namen A nur das Element A/1 vorhanden. Das Element A/2 wurde gelöscht.

- In der Eingabebibliothek X befinden sich alle Elemente einer Produktversion. Diese Elemente sollen in eine existierende Ausgabebibliothek Y so kopiert werden, dass Y nach dem Kopieren nur die kopierte Produktversion enthält und sonst keine andere Version. Das erreicht man mit folgender Anweisung.

```
//COPY-ELEM ELEMENT= *LIB(LIB=X,ELEM=*,TYPE=*),-
      TO ELEMENT= *LIB(LIB=Y),WRITE-MODE=*SUBSTITUTE
```

*Beispiel für das Kopieren einer kompletten Bibliothek*

Die Bibliothek lib1 wird komplett kopiert und erhält den Namen lib2. Durch die Angabe von '\*\*' bei Element und Typ ist keine Kenntnis über die enthaltenen Elemente erforderlich, d.h. alle Elemente werden im Verhältnis eins zu eins in die Bibliothek lib2 kopiert.

```

/START-LMS
//OPEN-LIBRARY LIB1
//SHOW-ELEMENT-ATTRIBUTES
  INPUT LIBRARY= :10SQ:$USER.LIB1
TYP NAME      VER (VAR#) DATE      NAME      VER (VAR#) DATE
(D) LETTER.A @ (0001) 2011-04-12  TESTELEM @ (0001) 2011-04-12
      2 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYP NAME      VER (VAR#) DATE
(S) TEST3 @   (0001) 2011-04-12
      1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
-----
      3 ELEMENT(S) IN THIS TABLE OF CONTENTS
//COPY-ELEMENT ( ,ELEM=*,TYPE=*),TO-ELEM=*LIB(LIBRARY=LIB2)
//SHOW-LIBRARY-STATUS
  STATUS FILENAME                                MODE      LINK DEF-TYPE
OPEN :10SQ:$USER.LIB2
OPEN :10SQ:$USER.LIB1                            READ
//SHOW-ELEMENT-ATTRIBUTES *LIB(LIBRARY=LIB2)
  INPUT LIBRARY= :N:$USER.LIB2
TYP NAME      VER (VAR#) DATE      NAME      VER (VAR#) DATE
(D) LETTER.A @ (0001) 2011-04-12  TESTELEM @ (0001) 2011-04-12
      2 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYP NAME      VER (VAR#) DATE
(S) TEST3 @   (0001) 2011-04-12
      1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
-----
      3 ELEMENT(S) IN THIS TABLE OF CONTENTS
//END

```



## COPY-LIBRARY

### Bibliothek kopieren

Die COPY-LIBRARY-Anweisung kopiert eine Bibliothek komplett mit allen Bibliotheks-, Typ- und Elementattributen. Die Zielbibliothek darf nicht existieren oder muss FILE-STRUCTURE=NONE haben. Die Zielbibliothek erhält das Bibliotheksformat entsprechend deren Wert für BUFFER-LENGTH. Die Anweisung ist damit zum Konvertieren des Bibliotheksformats geeignet.

Die Dateischutzattribute der Quellbibliothek können auf die Zielbibliothek übernommen werden. Die Anweisung ist damit zum Reorganisieren von Bibliotheken geeignet. Die Zielbibliothek ist logisch mit dem Original identisch und belegt nur noch den minimal notwendigen Plattenplatz.

Tritt während der COPY-LIBRARY-Verarbeitung ein Fehler auf (z.B. unzureichender Plattenplatz), ist die Zielbibliothek nicht vollständig.

#### COPY-LIBRARY

```

LIBRARY = <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
, TO-LIBRARY = <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
, FILE-ATTRIBUTES = *STD / *BY-SOURCE

```

**LIBRARY = <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, die kopiert werden soll.

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

Die Bibliothek mit dem hier angegebenen Namen wird kopiert.

**LIBRARY = \*LINK(...)**

Die über einen Dateikettungsamen zugewiesene Bibliothek wird kopiert.

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

Dateikettungsname der Bibliothek, der mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**TO-LIBRARY = <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Ziel-Bibliothek.

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

Die Bibliothek mit dem hier angegebenen Namen wird erstellt.

**TO-LIBRARY = \*LINK(...)**

Die über einen Dateikettungsamen zugewiesene Bibliothek wird erstellt.

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

Dateikettungsname der Bibliothek, der mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**FILE-ATTRIBUTES = \*STD / \*BY-SOURCE**

Attribute der Ziel-Bibliotheksdatei.

**FILE-ATTRIBUTES = \*STD**

Die Dateiattribute der Zielbibliothek werden nicht verändert. Neue Dateien werden mit den vom Dateiverwaltungssystem festgelegten Standardwerten erzeugt.

**FILE-ATTRIBUTES = \*BY-SOURCE**

Die Dateischutzattribute der Quellbibliothek werden auf die Zielbibliothek übernommen (analog /COPY-FILE . . . , PROTECTION=\*SAME).

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt

### Notwendige Zugriffsrechte

Leserecht für LIBRARY

Lese- und Schreibrecht für TO-LIBRARY

Eigentümerschaft für LIBRARY oder

LIBRARY ist nicht gegen TU-UPAM-Zugriffe geschützt oder

es existieren keine Schutzattribute auf Bibliotheks-, Typ- und Elementebene.

#### *Hinweis*

Bibliothekslisten sind nicht erlaubt.

*Beispiele*

## – Kopieren einer Bibliothek auf einen NK4-Pubset

```
/start-lms  
//copy-library library=lib,to-library=:nk4:lib  
//end
```

## – Vorab erstellen des NK4-Bibliotheksformates

```
/add-file-link file-name=nk4.lib,link-name=nk4,buffer-length=*std(2)  
/start-lms  
//copy-library library=nk2lib,to-library=*link(nk4)  
//end
```

## – Reorganisation einer Bibliothek mit Zwischenspeicher

```
/delete-file file-name=tolib  
/start-lms  
//copy-library library=lib,to-library=tolib,file-attributes=*by-source  
//end  
/copy-file from-file=tolib,to-file=lib  
/delete-file file-name=tolib
```

## DEACTIVATE-USER-EXIT

### Benutzerausgang lösen

Die Anweisung DEACTIVATE-USER-EXIT löst die Benutzerausgänge, die durch ACITIVATE-USER-EXIT bereitgestellt wurden. Die Benutzerausgänge werden bei dem nächsten entsprechenden SHOW-ELEMENT oder COMPARE-ELEMENT nicht mehr benutzt. Die Benutzerroutine wird jedoch noch nicht entladen, da sie eventuell noch anderweitig benötigt wird. Das bedeutet, dass bei dem nächsten ACTIVATE-USER-EXIT mit demselben Einsprungpunkt das Benutzerprogramm nicht neu dazugebunden werden muss.

Die DEACTIVATE-USER-EXIT-Anweisung erfordert die Angabe der Funktion, für die der Benutzerausgang gelöst werden soll.

#### DEACTIVATE-USER-EXIT

**FUNCTION = \*SHOW-ELEMENT / \*COMPARE-ELEMENT(...)**

**\*COMPARE-ELEMENT(...)**

**ELEMENT = \*PRIMARY / \*SECONDARY**

#### **FUNCTION = \*SHOW-ELEMENT / \*COMPARE-ELEMENT(...)**

Bestimmt diejenige LMS-Anweisung, für die der Benutzerausgang gelöst werden soll.

#### **FUNCTION = \*SHOW-ELEMENT**

Benutzerausgang für die SHOW-ELEMENT-Funktion lösen.

#### **FUNCTION = \*COMPARE-ELEMENT(...)**

Benutzerausgang für die COMPARE-ELEMENT-Funktion lösen.

#### **ELEMENT = \*PRIMARY / \*SECONDARY**

Wird der Benutzerausgang für die Anweisung COMPARE-ELEMENT gelöst, muss noch festgelegt werden, ob es sich um ein Primär- oder um ein Sekundärelement handelt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**Notwendige Zugriffsrechte**

Es sind keine Zugriffsrechte erforderlich.

## **DELETE-ELEMENT**

### **Löschen von Elementen**

Die Anweisung DELETE-ELEMENT löscht die angegebenen Elemente in der zugewiesenen Bibliothek (logisches Löschen). Dabei werden die Inhaltsverzeichniseinträge gelöscht und der Speicherplatz freigegeben.

Ein Element einer Bibliothek wird physikalisch gelöscht, wenn

- das Element ein Kennzeichen für physikalisches Löschen enthält,
- der Operand DESTROY-DATA=\*YES gesetzt ist,
- die CLASS2-Option DESTLEV es verlangt.

Delta-Elemente werden erst dann physikalisch gelöscht, wenn das letzte Delta-Element eines Delta-Baums, d.h. der komplette Delta-Baum, gelöscht wird.

Die Anweisung wird nur ausgeführt, wenn in der Anweisung explizit eine Bibliothek angegeben wurde oder die bei OPEN-LIBRARY angegebene Bibliothek mit MODE=\*UPDATE eröffnet wurde.

Die DELETE-ELEMENT-Anweisung ist für alle Elementtypen erlaubt.

**DELETE-ELEMENT****ELEMENT** = \*LIBRARY-ELEMENT (...)\*LIBRARY-ELEMENT(...)**LIBRARY** = \*STD / <filename 1..54 without-vers> / \*LINK(...)\*LINK(...)| **LINK-NAME** = <structured-name 1..8>, **ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)\*ALL(...)| **VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
| <composed-name 1..24 with-under with-wild(52)>, **BASE** = \*STD / <composed-name 1..24 with-under with-wild>

&lt;composed-name 1..64 with-under with-wild(132)&gt;(…)

| **VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
| <composed-name 1..24 with-under with-wild(52)>, **BASE** = \*STD / <composed-name 1..24 with-under with-wild>, **TYPE** = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>, **USER-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)\*INTERVAL(...)| **FROM** = 1900-01-01 / <date 8..10 with-compl>, **TO** = \*TODAY / <date 8..10 with-compl>, **CREATION-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)\*INTERVAL(...)| **FROM** = 1900-01-01 / <date 8..10 with-compl>, **TO** = \*TODAY / <date 8..10 with-compl>, **MODIFICATION-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)\*INTERVAL(...)| **FROM** = 1900-01-01 / <date 8..10 with-compl>, **TO** = \*TODAY / <date 8..10 with-compl>

(Teil 1 von 2)

```

,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
      *ANY(...)
        VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
          <composed-name 1..24 with-under with-wild(52)>
        ,BASE = *STD / <composed-name 1..24 with-under with-wild>
      <composed-name 1..64 with-under with-wild(132)>(…)
        VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
          <composed-name 1..24 with-under with-wild(52)>
        ,BASE = *STD / <composed-name 1..24 with-under with-wild>
      ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
      ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
        *INTERVAL(...)
          FROM = 1900-01-01 / <date 8..10 with-compl>
          ,TO = *TODAY / <date 8..10 with-compl>
        ,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
          *INTERVAL(...)
            FROM = 1900-01-01 / <date 8..10 with-compl>
            ,TO = *TODAY / <date 8..10 with-compl>
        ,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
          *INTERVAL(...)
            FROM = 1900-01-01 / <date 8..10 with-compl>
            ,TO = *TODAY / <date 8..10 with-compl>
      ,DESTROY-DATA = *LMS-DEFAULT / *NO / *YES
      ,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES

```

(Teil 2 von 2)



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

Angaben, welches Element gelöscht werden soll..

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, aus der das Element gelöscht werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, aus der das Element gelöscht werden soll.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, welches gelöscht werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das zu löschende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird gelöscht.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird gelöscht.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das gelöscht werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das zu löschende Element besitzt.

**USER-DATE = \*ANY /\*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das zu löschende Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird gelöscht.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird gelöscht.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden gelöscht.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element vom Löschen ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die vom Löschen ausgeschlossen sind. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieses Kommandos.

**DESTROY-DATA =\*LMS-DEFAULT / \*NO / \*YES**

Löschen der Daten für alle Elemente, die durch \*LIBRARY-ELEMENT bestimmt wurden.

**DESTROY-DATA = \*NO**

Ein Element einer Bibliothek wird nur physikalisch gelöscht, wenn im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder die CLASS2-Option DESTLEV es verlangt.

**DESTROY-DATA = \*YES**

Nach dem logischen Löschen werden die Daten, falls vorhanden, physikalisch gelöscht, d.h. mit X'00' überschrieben.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS, wobei der dort ggf. eingestellte Wert \*ERROR wie \*NO wirkt. Ebenso wirkt der beim /SEND-MSG-Kommando ggf. angegebene Wert für DIALOG-CONTROL= \*ERROR bei DELETE-ELEMENT wie \*NO.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht vorhanden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Lese- und Schreibrecht für LIBRARY

Administrationsrecht und Schreibrecht für ELEMENT.

*Hinweise*

- Bei jedem UPDATE auf einen Delta-Baum wird die Delta-Struktur reorganisiert, d.h. nicht mehr benötigte Datensätze werden gelöscht und der nicht mehr benötigte Speicher freigegeben.
- Das Löschen einer Hauptzweigversion unter der Konvention STD-TREE, von der Neben-zweigversionen abhängen, kann bei späteren Kopiervorgängen zu Problemen führen.

*Beispiel*

Aus der Bibliothek LIB1 wird das Element TEST3 gelöscht.

```
/START-LMS
//OPEN-LIBRARY LIB1,MODE=*UPDATE
//SHOW-ELEMENT-ATTRIBUTES
TYP NAME      VER (VAR#) DATE      NAME      VER (VAR#) DATE
(D) LETTER.A @ (0001) 2012-04-12  TESTELEM @ (0001) 2012-04-12
      2 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYP NAME      VER (VAR#) DATE
(S) TEST3 @   (0001) 2012-04-12
      1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
-----
      3 ELEMENT(S) IN THIS TABLE OF CONTENTS
//DELETE-ELEMENT *LIB(ELEM=TEST$,TYPE=S)
//SHOW-ELEMENT-ATTRIBUTES
INPUT LIBRARY= :10SQ:$USER.LIB1
TYP NAME      VER (VAR#) DATE      NAME      VER (VAR#) DATE
(D) LETTER.A @ (0001) 2012-04-12  TESTELEM @ (0001) 2012-04-12
      2 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//END
```

## **EDIT-ELEMENT**

### **Editieren von Elementen**

Die Anweisung EDIT-ELEMENT ruft das Dateibearbeitungsprogramm EDT auf, um entweder Textelemente zu erstellen, zu korrigieren oder anzuschauen (siehe auch [10] und [11]).

Der Basistyp für Quell- und Zielelement darf unterschiedlich sein.

### **Erstellen und Korrigieren von Textelementen**

Die Anweisung EDIT-ELEMENT ruft den EDT auf und liest das eben angegebene Element aus der zugewiesenen Bibliothek in die Arbeitsdatei 0 ein. Der Inhalt der Arbeitsdatei 0 wird vorher gelöscht. Wenn der EDT beendet wird, wird das eventuell korrigierte bzw. erstellte Element in die zugewiesene Bibliothek geschrieben.

LMS unterstützt EDT-Versionen ab V16.2A.

Ab der EDT-Version V16.4A wird XHCS unterstützt. LMS übergibt beim Aufruf den Coded-Character-Set-Namen des jeweiligen Elementes an den EDT und schreibt das Element mit dem im EDT eingestellten Wert zurück. Ist das Eingabeelement nicht angegeben, so wird als Eingabe-CCSN „kein Code“ angenommen.

Für EDT-Versionen < V16.4A wird als CCSN des Zielelementes der CCSN des Eingabeelementes eingetragen. Ist kein Eingabeelement angegeben, so erhält das Zielelement den CCSN „kein Code“.

Ist im Attributsatz (Satzart 164) des Eingabeelementes RECORD-FORMAT=FIXED, KEY-POSITION > 5 oder KEY-LENGTH > 8 gespeichert, so ist ein Editiervorgang nicht möglich.

Für die Bearbeitung von Unicode-Elementen wird EDT ab V17.0 benötigt.

### **Hilfsdatei**

LMS erzeugt beim Aufruf des EDT eine Hilfsdatei mit dem Dateikettungsnamen EDTISAM, sofern ISAM-Schlüssel im Element gespeichert sind.

Name der Hilfsdatei:

S.LMS.TSNnnnn.datum.uhrzeit.elem

elem hat eine maximale Länge von neun Zeichen. Längere Elementebezeichnungen werden bis auf die ersten neun Zeichen abgekürzt. Ergeben sich durch das Anhängen des Elementnamens unzulässige BS2000-Dateinamen (z.B. das 9. Zeichen ist ein „.“), bildet LMS den Hilfsnamen ohne elem.

**Ablauf des Editieraufrufs:**

- Die Elementsätze werden von LMS an den EDT weitergereicht. Das Element steht dann im virtuellen Speicher zur Verfügung.
- Sind im Element ISAM-Schlüssel gespeichert, zeigt die Zeilennummernanzeige des EDT die ersten sechs Stellen dieser Schlüssel an.
- Das zu bearbeitende Element ist für andere Benutzer gesperrt.

**Beenden des EDT-Laufes****RETURN aus Arbeitsdatei 0:**

Die Arbeitsdatei 0 wird als Element in die Ausgabebibliothek aufgenommen, wenn diese nicht leer ist.. Die EDT-Daten (Dateien im virtuellen Speicher, Variablen,...) bleiben erhalten. Nur bei einem schwerwiegenden EDT-Fehler werden diese Daten freigegeben.

**HALT aus Arbeitsdatei 0:**

Es erfolgt die Abfrage:

```
LMS0420: EDITED ELEMENT (Typ)Elementname/Version[(Variantennummer)]/Datum TO  
BE ADDED? REPLY (Y=YES, N=NO or R=RETURN TO EDITOR)?
```

Entsprechend der Antwort wird die aktuelle Arbeitsdatei als Element aufgenommen oder zum EDT zurückgekehrt. Die EDT-Daten bleiben erhalten. Nur bei einem schwerwiegenden EDT-Fehler werden diese Daten freigegeben.

**HALT/RETURN aus Arbeitsdatei ≠ 0:**

Es erfolgt die Abfrage:

```
LMS0420: EDITED ELEMENT (Typ)Elementname/Version[(Variantennummer)]/Datum TO  
BE ADDED? REPLY (Y=YES, N=NO or R=RETURN TO EDITOR)?
```

Wird „N“ eingegeben, wird das Element nicht aufgenommen. Bei der Antwort „Y“ wird als nächstes ein Dialog mit dem Anwender geführt:

```
LMS0421: WORKFILE TO BE ADDED (0 = WORKFILE(0),..., N = NONE)
```

Wird hier die Antwort „N“ gegeben, kehrt LMS in die gerade bearbeitete Arbeitsdatei des EDT zurück.

**EDT im Stapelbetrieb****@RETURN:**

Das korrigierte Element wird aus der Arbeitsdatei 0 in die Ausgabebibliothek aufgenommen, wenn diese nicht leer ist.

**@HALT:**

Das korrigierte Element wird nicht in die Ausgabebibliothek aufgenommen.

**Information über das zu editierende Element aus dem EDT im Dialogbetrieb****RETURN ? / HALT ?**

Die Namen der Zielbibliothek und des Zielelements mit Version und Typ des zu editierenden Elementes aus der Arbeitsdatei 0 werden angezeigt. Ferner wird der bearbeitete Teilbereich des Elementes ausgegeben. Danach wird der Editiervorgang fortgesetzt. Wenn ELEMENT=\*NONE bzw. TO-ELEMENT=\*NONE angegeben sind, sind RETURN ? und HALT ? wirkungslos.

**EDIT-ELEMENT**

**ELEMENT = \*NONE / \*LIBRARY-ELEMENT(...)**

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

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

**\*LINK(...)**

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

**,ELEMENT = \*ALL(...)/ <composed-name 1..64 with-under with-wild(132)>(...**

**\*ALL(...)**

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
<composed-name 1..24 with-under with-wild(52)>**

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**<composed-name 1..64 with-under with-wild(132)>(...**

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
<composed-name 1..24 with-under with-wild(52)>**

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**,TYPE = \*LMS-DEFAULT / <alphanum-name 1..8>**

(Teil 1 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(...)
    | *ANY(...)
    |   |
    |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   <composed-name 1..24 with-under with-wild(52)>
    |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | <composed-name 1..64 with-under with-wild(132)>(...)
    |   |   |
    |   |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   |   <composed-name 1..24 with-under with-wild(52)>
    |   |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8>
    |   | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)



	<p>,<b>CREATION-DATE</b> = <u>*ANY</u> / *<b>TODAY</b> / &lt;date 8..10 with-compl&gt; / *<b>INTERVAL</b>(...)</p> <p style="padding-left: 20px;">*<b>INTERVAL</b>(...)</p> <p style="padding-left: 40px;">  <b>FROM</b> = <u>1900-01-01</u> / &lt;date 8..10 with-compl&gt;</p> <p style="padding-left: 40px;">  ,<b>TO</b> = <u>*TODAY</u> / &lt;date 8..10 with-compl&gt;</p> <p>,<b>MODIFICATION-DATE</b> = <u>*ANY</u> / *<b>TODAY</b> / &lt;date 8..10 with-compl&gt; / *<b>INTERVAL</b>(...)</p> <p style="padding-left: 20px;">*<b>INTERVAL</b>(...)</p> <p style="padding-left: 40px;">  <b>FROM</b> = <u>1900-01-01</u> / &lt;date 8..10 with-compl&gt;</p> <p style="padding-left: 40px;">  ,<b>TO</b> = <u>*TODAY</u> / &lt;date 8..10 with-compl&gt;</p> <p>,<b>TO-ELEMENT</b> = *<u>LIBRARY-ELEMENT</u>(...) / *<b>NONE</b></p> <p>*<b>LIBRARY-ELEMENT</b>(...)</p> <p style="padding-left: 20px;">  <b>LIBRARY</b> = *<u>STD</u> / *<b>BY-SOURCE</b> / &lt;filename 1..54 without-vers&gt; / *<b>LINK</b>(...)</p> <p style="padding-left: 40px;">  *<b>LINK</b>(...)</p> <p style="padding-left: 60px;">  <b>LINK-NAME</b> = &lt;structured-name 1..8&gt;</p> <p>,<b>ELEMENT</b> = *<u>BY-SOURCE</u> (...) / &lt;composed-name 1..132 with-under with-wild-constr&gt;(...)</p> <p style="padding-left: 20px;">*<b>BY-SOURCE</b>(...)</p> <p style="padding-left: 40px;">  <b>VERSION</b> = *<u>LMS-DEFAULT</u> / *<b>BY-SOURCE</b> / *<b>HIGHEST-EXISTING</b> / *<b>INCREMENT</b> /</p> <p style="padding-left: 60px;">  *<b>UPPER-LIMIT</b> / &lt;composed-name 1..52 with-under with-wild-constr&gt;</p> <p style="padding-left: 40px;">  ,<b>BASE</b> = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt;</p> <p style="padding-left: 20px;">&lt;composed-name 1..132 with-under with-wild-constr&gt;(...)</p> <p style="padding-left: 40px;">  <b>VERSION</b> = *<u>LMS-DEFAULT</u> / *<b>BY-SOURCE</b> / *<b>HIGHEST-EXISTING</b> / *<b>INCREMENT</b> /</p> <p style="padding-left: 60px;">  *<b>UPPER-LIMIT</b> / &lt;composed-name 1..52 with-under with-wild-constr&gt;</p> <p style="padding-left: 40px;">  ,<b>BASE</b> = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt;</p> <p>,<b>TYPE</b> = *<u>BY-SOURCE</u> / *<u>LMS-DEFAULT</u> / &lt;alphanum-name 1..8&gt;</p> <p>,<b>USER-DATE</b> = *<u>TODAY</u> / *<b>BY-SOURCE</b> / &lt;date 8..10 with-compl&gt;</p> <p>,<b>STORAGE-FORM</b> = *<u>LMS-DEFAULT</u> / *<b>STD</b> / *<b>FULL</b> / *<b>DELTA</b></p> <p>,<b>EDT-MODE</b> = *<u>STD</u> / *<b>COMPATIBLE</b> / *<b>UNICODE</b></p> <p>,<b>EDITOR-COMMANDS</b> = *<u>NONE</u> / &lt;c-string 1..251&gt; / *<b>LOWER-CASE</b>(...)</p> <p style="padding-left: 20px;">*<b>LOWER-CASE</b>(...)</p> <p style="padding-left: 40px;">  <b>EDITOR-COMMANDS</b> = &lt;c-string 1..251 with-low&gt;</p> <p>,<b>INFORMATION</b> = *<u>TEXT</u> / list-poss(2): *<b>TEXT</b> / *<b>COMMENT</b></p> <p>,<b>WRITE-MODE</b> = *<u>LMS-DEFAULT</u> / *<b>CREATE</b> / *<b>REPLACE</b> / *<b>EXTEND</b> / *<b>ANY</b></p> <p>,<b>DIALOG-CONTROL</b> = *<u>LMS-DEFAULT</u> / *<b>NO</b> / *<b>YES</b> / *<b>ERROR</b></p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Teil 3 von 3)

**ELEMENT = \*NONE / \*LIBRARY-ELEMENT(...)**

Angabe des Elementes, das editiert werden soll.

**ELEMENT = \*NONE**

Es wird kein Eingabeelement angegeben. Das zu editierende Element wird neu eingerichtet bzw. mit neuen Daten überschrieben.

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, aus der das Element editiert werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, aus der das Element genommen werden soll.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, welches editiert werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /**

**<composed-name 1..24 with-under with-wild(52)>**

Version, die das zu editierende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird verwendet.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version 'X'FF' wird editiert.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das editiert werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

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

Typ, den das zu editierende Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das zu editierende Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird editiert.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird editiert.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden editiert.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element vom Editieren ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die vom Editieren ausgeschlossen sind. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieses Kommandos.

**TO-ELEMENT = \*LIBRARY-ELEMENT(...) / \*NONE**

Angabe, wohin und unter welchem Namen das Element zurückgeschrieben werden soll.

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / \*BY-SOURCE / <filename 1..54 without-vers> / \*LINK(...)**  
Angabe der Bibliothek, in die das Element aufgenommen werden soll.

**LIBRARY = \*STD**  
Die durch OPEN-LIBRARY eröffnete Bibliothek.

**LIBRARY = \*BY-SOURCE**  
Das Element wird in die Bibliothek aufgenommen, aus der es auch genommen wurde.

**LIBRARY = <filename 1..54 without-vers>**  
Name der Bibliothek, in die die Datei als Element aufgenommen werden soll. Existiert die Bibliothek noch nicht, wird sie neu eingerichtet.

**LIBRARY = \*LINK(...)**  
Die über den Dateikettungsamen zugewiesene Bibliothek.

**LINK-NAME = <structured-name 1..8>**  
Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*BY-SOURCE(...) /**  
**<composed-name 1..132 with-under with-wild-constr>(...)**  
Name, den das neu aufzunehmende Element erhalten soll.

**ELEMENT = \*BY-SOURCE(...)**  
Der neue Name entspricht dem alten Namen.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING /**  
**\*INCREMENT / \*UPPER-LIMIT /**  
**<composed-name 1..52 with-under with-wild-constr>**  
Version, die das neu aufzunehmende Element erhalten soll.

**VERSION = \*BY-SOURCE**  
Das neue Element erhält die gleiche Version wie das ursprüngliche Element. Hat das ursprüngliche Element keine Versionsangabe, erhält das neue Element X'FF' als Versionsangabe.

**VERSION = \*HIGHEST-EXISTING**  
Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**  
Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion (siehe auch [Seite 54](#)).

**VERSION = \*UPPER-LIMIT**  
Die höchstmögliche Version X'FF' wird erzeugt.

**VERSION = <composed-name 1..52 with-under with-wild-constr>**

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name des neu aufzunehmenden Elementes. Er kann auch mit Wildcards eingegeben werden.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /**

**<composed-name 1..52 with-under with-wild-constr>**

Version, die das neu aufzunehmende Element erhalten soll.

Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / <alphanum-name 1..8>**

Typ, den das neu aufzunehmende Element erhalten soll.

**TYPE = \*BY-SOURCE**

Das neue Element erhält die gleiche Typbezeichnung wie das ursprüngliche Element.

**USER-DATE = \*TODAY / \*BY-SOURCE / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = \*BY-SOURCE**

Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**STORAGE-FORM = \*LMS-DEFAULT / \*STD / \*FULL / \*DELTA**

Speicherform für das neu anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen, und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*STD**

Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

**STORAGE-FORM = \*DELTA**

Das Element wird delta-gespeichert angelegt (Fehler, falls nicht zulässig). Die Angabe ist zulässig für die Elementtypen: S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**TO-ELEMENT = \*NONE**

Das editierte Element wird nicht zurückgeschrieben.

**EDT-MODE = \*STD / \*COMPATIBLE / \*UNICODE**

Legt den Betriebsmodus fest, in dem der EDT aufgerufen wird.

**EDT-MODE = \*STD**

Der EDT wird in dem Modus aufgerufen, der mit der Anweisung MODIFY-LMS-DEFAULTS festgelegt ist, er ist jedoch berechtigt, bei Bedarf selbst den Modus zu wechseln.

**EDT-MODE = \*COMPATIBLE**

Der EDT wird im Kompatibilitätsmodus aufgerufen.

**EDT-MODE = \*UNICODE**

Der EDT wird im Unicode-Modus aufgerufen.

**EDITOR-COMMANDS = \*NONE / <c-string 1..251> / \*LOWER-CASE(...)**

Gibt eine Editor-Kommandofolge an.

**EDITOR-COMMANDS = \*NONE**

Keine Editor-Kommandofolge spezifiziert.

**EDITOR-COMMANDS = <c-string 1..251>**

Editor-Kommandofolge, die nach Aufruf des EDT ausgeführt wird. Die einzelnen Kommandos müssen durch „;“ getrennt eingegeben werden. Mit Ausnahme von EDIT und RETURN dürfen alle Kommandos angegeben werden, die sowohl im F-Modus als auch im L-Modus des EDT zulässig sind. Das Kommando EDIT ist nur in der Form EDIT ONLY erlaubt. Zu beachten ist, dass das Kommando HALT in der EDT-Kommandofolge das Beenden des EDT bedeutet und somit die Freigabe der EDT-Daten (Dateien im virtuellen Speicher, Variablen, ...).

Nach Ausführung dieser EDT-Kommandofolge wird das erstellte bzw. korrigierte Element direkt zurückgeschrieben.

Kleinbuchstaben werden in Großbuchstaben umgesetzt.

**EDITOR-COMMANDS = \*LOWER-CASE(...)**

Kleinbuchstaben werden nicht umgesetzt.

**EDITOR-COMMANDS = <c-string 1..251 with-low>**

Editor-Kommandofolge, wie oben, nur dass Kleinbuchstaben nicht in Großbuchstaben umgesetzt werden.

*Beispiel für ein EDT-Kommando*

```
EDITOR-COM = 'ON&C''A''TO''B'''
```

*Beispiel für ein EDT-Kommando ohne Umsetzung der Kleinbuchstaben*

```
EDITOR-COM = (('LOWER ON;ON&C''a''TO''b''')
```

**INFORMATION = \*TEXT / list-poss(2): \*TEXT / \*COMMENT**

der zu bearbeitende Teilbereich des Elementes

**INFORMATION = \*TEXT**

Der eigentliche Text, d.h. Satzart 1 wird editiert.

**INFORMATION = \*COMMENT**

Der separat gespeicherte Kommentar, d.h. Satzart 2 wird editiert.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

Überschreiben eines Elementes gleichen Namens. Ist Quellelement gleich Zielelement, so wird der Operand WRITE-MODE ignoriert. Ist das zu speichernde Element ein Delta-Element, ist darauf zu achten, dass das Element ein Blatt des Delta-Baumes sein muss. Nur Blätter eines Delta-Baumes dürfen überschrieben werden.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*EXTEND**

Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt. \*EXTEND ist nicht bei Delta-Elementen erlaubt.

**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
2	0	LMS0163	Mindestens ein Satz abgeschnitten
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei LIBRARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgeschlossen.

Bei TO-ELEMENT: Lese- und Schreibrecht für LIBRARY

Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).



Bei STORAGE-FORM=\*DELTA muss für das durch BASE bestimmte Element Leserecht gegeben sein.

Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Es kann nur eine neue Version erzeugt oder diese Basisversion überschrieben werden, sofern Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

#### *Hinweise*

- Siehe Hinweis bei CALL-EDT.
- Beim Anlegen eines Elementes ist die unter dem Typ geltende Konvention zu beachten.
- Die Angabe einer Liste bei INFORMATION führt für jede Angabe zu einem Editiervorgang in Arbeitsdatei 0. Es ergeben sich also zeitlich aufeinanderfolgende Editiervorgänge der einzelnen Elementbereiche. Das Beenden des EDT-Laufes und somit die Entscheidung, ob das editierte Element (alle ausgewählten Teilbereiche) in die Bibliothek aufgenommen werden soll, erfolgt dabei erst beim letzten ausgewählten Teilbereich.
- Bei aktivem WRITE-CONTROL in der Ausgabebibliothek wird dem zu schreibenden Element von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorganges protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Bereits vorhandene oder durch INFORMATION \*COMMENT editierte Kommentarsätze werden dahinter kopiert. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), so werden die Elementattribute STATE, HOLDER und alle Rechte von der Basisversion auf die neue Version übernommen. Die CCSN wird von dem Quellelement übernommen. Das USER-DATE wird neu bestimmt.
- Wenn der EDT-MODE explizit in der Anweisung EDIT-ELEMENT festgelegt ist, aber der EDT in diesem Modus nicht starten kann, wird die Anweisung abgebrochen und die Meldung LMS0297 ausgegeben.
- Wenn der EDT-MODE nicht explizit in CALL-EDT festgelegt ist, startet der EDT im EDT-Default-Modus, kann bei Bedarf jedoch selbst den Modus wechseln. Der EDT-Default-Modus ist zu Beginn des LMS-Laufs COMPATIBLE und kann mit der Anweisung MODIFY-LMS-DEFAULTS geändert werden.

*Beispiele*

Das Element test4 soll mit EDT editiert werden:

```
//edit-elem *lib(elem=test4,type=s)
```

Das Element test4 soll mit EDT editiert werden, wobei der String 'AltWort' in den String 'NeuWort' umgewandelt wird:

```
//edit-elem *lib(elem=test4,type=s),ed-com=*low-case('on 1-10 c''AltWort''t''NeuWort''')
```

Das Element test4 soll mit EDT angeschaut, aber nicht zurückgeschrieben werden:

```
//edit-elem (elem=test4,type=s),to-elem=*none
```

Die Bibliothek LIBCCSN enthält das Element UTFE, das im Zeichensatz UTFE codiert ist. Das Element LONG-LINES ist im normalen 7-Bit-Zeichensatz EDF03IRV codiert, hat aber Zeilen, die länger als 256 Zeichen sind. Der EDT-Default-Modus ist COMPATIBLE.

```
//modify-lms-defaults edt-mode=*compatible
//open-lib libccsn,*upd
//edit-elem (libccsn,utf,s),edt-mode=*compatible
% LMS0297 WECHSEL DES EDT-BETRIEBSMODUS NICHT MOEGlich _____ (1)
//edit-elem (libccsn,utf,s) _____ (2)
```

```
1.00 LINE 1
2.00 LINE 2

22.00
23.00
ret                                0000.00:00001(00)
LTG                                TAST
```

```
//edit-elem (libccsn,long-lines,s),edt-mode=*unicode _____ (3)
```

```
1.00 LINE 1
2.00 very long line ... longer than 256 characters

22.00
23.00
ret                                0000.00:00001(00)
LTG                                TAST
```

- (1) EDT kann nicht starten, da im Kompatibilitätsmodus Unicode-Zeichensätze nicht unterstützt werden.
- (2) EDT wechselt automatisch in den Unicode-Modus
- (3) Unicode-Modus muss explizit festgelegt werden

## EDIT-ELEMENT-ATTRIBUTES

### Geführten Dialog für MODIFY-ELEMENT-ATTRIBUTES aktivieren

Die Anweisung EDIT-ELEMENT-ATTRIBUTES aktiviert den geführten Dialog für die Anweisung MODIFY-ELEMENT-ATTRIBUTES. Soweit technisch möglich und sinnvoll, werden dabei die vordefinierten Default-Werte der einzelnen Operanden durch Werte ersetzt, die für das angegebene Element aktuell gültig sind.

#### EDIT-ELEMENT-ATTRIBUTES

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

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

**LIBRARY** = \*STD / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = <composed-name 1..64 with-under>(…)

<composed-name 1..64 with-under>(…)

**VERSION** = \*HIGHEST-EXISTING / **\*UPPER-LIMIT** /

<composed-name 1..24 with-under>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

**,TYPE** = \*LMS-DEFAULT / <alphanum-name 1..8>

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

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

Name des Elementes, dessen Attribute geändert werden sollen.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /****<composed-name 1..24 with-under>**

Version, die das Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Die Attribute des Elementes mit der höchsten bestehenden Version werden geändert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version 'X'FF' des Elementes wird verwendet.

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

Explizite Angabe der Version des Elementes.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>**

Typ, den das Element besitzt.

**Anweisungs-Returncode**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0303	Element nicht im Bereich der Referenz-Bedingung
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

## EDIT-ELEMENT-PROTECTION

### Geführten Dialog für MODIFY-ELEMENT-PROTECTION aktivieren

Die Anweisung EDIT-ELEMENT-PROTECTION aktiviert den geführten Dialog für die Anweisung MODIFY-ELEMENT-PROTECTION. Soweit technisch möglich und sinnvoll, werden dabei die vordefinierten Default-Werte der einzelnen Operanden durch Werte ersetzt, die für das angegebene Element aktuell gültig sind.

#### EDIT-ELEMENT-PROTECTION

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

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

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

**\*LINK(...)**

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

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

**<composed-name 1..64 with-under>(...)**

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /**

**<composed-name 1..24 with-under>**

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**,TYPE = \*LMS-DEFAULT / <alphanum-name 1..8>**

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

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

Name des Elementes, dessen Schutz-Attribute geändert werden soll.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Die Schutz-Attribute des Elementes mit der höchsten bestehenden Version werden geändert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' des Elementes wird verwendet.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

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

Typ, den das Element besitzt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS0303	Element nicht im Bereich der Referenz-Bedingung
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

## END

### Beenden des LMS-Laufes

Mit END wird das Programm LMS beendet. Alle noch geöffneten Bibliotheken werden geschlossen.

END

Die END-Anweisung besitzt keine Operanden.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	LMS normal beendet
	64	CMD0205	LMS abnormal beendet



## EXTRACT-ELEMENT

### Ausgeben von Elementen als Dateien

Die Anweisung EXTRACT-ELEMENT gibt Bibliothekselemente standardmäßig in Dateien aus. Wird die Anweisung in der EDT-Anweisungszeile gegeben, werden die Elementdaten standardmäßig in die aktuelle EDT-Arbeitsdatei geschrieben.

LMS erzeugt die Dateien entsprechend

- dem Eintrag in der TASK-FILE-TABLE (TFT), wenn die Datei mit dem Dateikettungsnamen zugeordnet ist,
- den gespeicherten Dateimerkmalen und dem Operanden FILE-ATTRIBUTES,
- dem Katalogeintrag.

Die Dateien können RECORD-FORMAT=UNDEFINED und beliebige Angaben zu BUFFER-LENGTH und RECORD-SIZE haben. Die maximale Satzlänge von 32 KByte (inkl. Satzkopf) darf allerdings nicht überschritten werden.

Wurden die ISAM-Schlüssel einer ISAM-Datei mit in das Element aufgenommen, werden die ISAM-Schlüssel mit EXTRACT-ELEMENT auch wieder mit ausgegeben.

Wurden bei der Aufnahme der Datei Informationen über ISAM-Sekundärschlüssel gespeichert, werden die Sekundärschlüssel wieder eingerichtet. Ist das Einrichten einzelner oder aller Sekundärschlüssel nicht möglich, wird die Datei ohne oder nicht mit allen Sekundärschlüsseln erzeugt.

Die Anweisung EXTRACT-ELEMENT ist für die Elementtypen S, M, R, J, P, D, X, C und davon abgeleitete Elementtypen erlaubt.

Elemente vom Typ C und PAM-Dateien unter dem Typ X bzw. davon abgeleitete Typen werden als PAM-Dateien erzeugt.

Die erzeugte Datei erhält als Katalogattribut CCS den CCS-Namen des Quellelements. Werden die Elementdaten in eine EDT-Arbeitsdatei geschrieben, so ist die Anweisung EXTRACT-ELEMENT nur für textartige Elementtypen oder davon abgeleitete Typen erlaubt, wobei der CCSN des jeweiligen Elements an den EDT übergeben wird.

#### *Hinweis*

Zulässige Elementnamen sind nicht in jedem Fall als Dateinamen zulässig.

### Erzeugen von ISAM-Dateien

Beim Ausgeben von Elementen in ISAM-Dateien erzeugt LMS die ISAM-Schlüssel wie folgt:

- Wurden bei der Aufnahme einer ISAM-Datei als Bibliothekselement die ISAM-Schlüssel mit aufgenommen, erzeugt LMS die ISAM-Datei mit diesen gespeicherten ISAM-Schlüsseln.
- Sind im Eingabeelement keine ISAM-Schlüssel gespeichert, wird eine ISAM-Datei mit KEY-POSITION = 5 und KEY-LENGTH = 8 erstellt. Standardmäßig erzeugt LMS dann ISAM-Schlüssel mit einem Anfangswert von 1000 und einer Schrittweite von 1000. Ist das Element zu groß für diese Schrittweite (mehr als 100000 Sätze), so wird die Schrittweite aus der Anzahl der Sätze berechnet.

#### *Hinweise*

- Elemente vom Typ R werden bis zum END-Satz ausgegeben. Eventuell nachfolgende Sätze werden ignoriert.
- Korrekturjournalsätze (TXTP) werden bei Elementen vom Typ C nicht mit ausgegeben.
- Die RECORD-SIZE wird nur bei RECORD-FORMAT=FIXED versorgt; sie hat bei RECORD-FORMAT=VARIABLE den Wert 0.

## EXTRACT-ELEMENT

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

\*LIBRARY-ELEMENT(...)

**LIBRARY** = \*STD / <filename 1..54 without-vers> / \*LINK(...)

\*LINK(...)

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

,**ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)

\*ALL(...)

| **VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
| <composed-name 1..24 with-under with-wild(52)>

| ,**BASE** = \*STD / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(…)

| **VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
| <composed-name 1..24 with-under with-wild(52)>

| ,**BASE** = \*STD / <composed-name 1..24 with-under with-wild>

,**TYPE** = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>

,**USER-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)

\*INTERVAL(...)

| **FROM** = 1900-01-01 / <date 8..10 with-compl>

| ,**TO** = \*TODAY / <date 8..10 with-compl>

,**CREATION-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)

\*INTERVAL(...)

| **FROM** = 1900-01-01 / <date 8..10 with-compl>

| ,**TO** = \*TODAY / <date 8..10 with-compl>

,**MODIFICATION-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)

\*INTERVAL(...)

| **FROM** = 1900-01-01 / <date 8..10 with-compl>

| ,**TO** = \*TODAY / <date 8..10 with-compl>

(Teil 1 von 3)

**,EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

**\*ELEMENT(...)**

**ELEMENT = \*ANY (...) / <composed-name 1..64 with-under with-wild(132)>(...)**

**\*ANY(...)**

**VERSION = \*ANY / \*HIGHEST-EXISTING / \*UPPER-LIMIT /  
<composed-name 1..24 with-under with-wild(52)>**

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**<composed-name 1..64 with-under with-wild(132)>(...)**

**VERSION = \*ANY / \*HIGHEST-EXISTING / \*UPPER-LIMIT /  
<composed-name 1..24 with-under with-wild(52)>**

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**,TYPE = \*ANY / \*LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>**

**,USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

**,CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

**,MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

(Teil 2 von 3)

```

,TO-FILE = *STD / *BY-SOURCE / <filename 1..54 without-gen-vers with-wild-constr> / *LINK(...)
    *LINK(...)
        |   LINK-NAME = <structured-name 1..8>
,FILE-ATTRIBUTES = *BY-ELEMENT / *BY-CATALOG / *LMS-DEFAULT / *PARAMETERS(...)
    *PARAMETERS(...)
        |   ACCESS-METHOD = *LMS-DEFAULT / *ISAM / *SAM
,INFORMATION = *TEXT / list-poss(2): *TEXT / *COMMENT
,PROTECTION = *LMS-DEFAULT / *STD / *BY-SOURCE
,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *EXTEND / *ANY
,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(..)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, welches aus der Bibliothek in eine Datei übernommen werden soll.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /****<composed-name 1..24 with-under with-wild(52)>**

Version, die das auszugebende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird verwendet.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird ausgegeben.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das ausgegeben werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das zu auszugebende Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das auszugebende Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird ausgegeben.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird ausgegeben.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden ausgegeben.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die nicht ausgegeben werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieses Kommandos.

**TO-FILE = \*STD / \*BY-SOURCE /**

**<filename 1..54 without-gen-vers with-wild-constr> / \*LINK(...)**

Name der Zieldatei. Eine Konstruktionsangabe bezieht sich auf den Elementnamen.

**TO-FILE = \*STD**

Standardmäßig werden die Elementdaten in Dateien ausgegeben, wobei die Datei den gleichen Namen wie das Element erhält. Erfolgt die EXTRACT-ELEMENT-Anweisung jedoch aus der EDT-Anweisungszeile, so werden die Daten standardmäßig in die aktuelle EDT-Arbeitsdatei geschrieben.

**TO-FILE = \*BY-SOURCE**

Der Dateiname ist der gleiche Name wie der Elementname.

**TO-FILE = \*LINK(...)**

Das Element wird in die Datei ausgegeben, die über den Dateikettungsnamen zugewiesen wurde.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**FILE-ATTRIBUTES = \*BY-ELEMENT / \*BY-CATALOG / \*LMS-DEFAULT / \*PARAMETERS(...)**

Dateiattribute, die bei der Erzeugung der Datei festgelegt werden. Dieser Operand ist wirkungslos, wenn die Elementdaten in die aktuelle EDT-Arbeitsdatei geschrieben werden. LMS bestimmt die Dateiattribute nach folgender Hierarchie:

1. LINK-Eintrag
2. im Element gespeicherte Dateiattribute
3. Katalogeintrag
4. LMS Defaultwerte

Die nachfolgenden Angaben wirken nur, wenn nicht TO-FILE=\*LINK angegeben wurde.

**FILE-ATTRIBUTES = \*BY-ELEMENT**

Die im Element gespeicherten Dateiattribute haben Vorrang.

**FILE-ATTRIBUTES = \*BY-CATALOG**

Die im Katalogeintrag gespeicherten Attribute haben Vorrang. Fehlt der Katalogeintrag, wirkt die Angabe \*BY-CATALOG wie die Angabe \*BY-ELEMENT.

**FILE-ATTRIBUTES = \*PARAMETERS(...)****ACCESS-METHOD = \*LMS-DEFAULT / \*ISAM / \*SAM**

Legt die Zugriffsmethode ISAM oder SAM für die Zieldatei fest.

**INFORMATION = \*TEXT / list-poss(2): \*TEXT / \*COMMENT**

der zu bearbeitende Teilbereich des Elementes.

**INFORMATION = \*TEXT**

Der eigentliche Text, d.h. Satzart 1 wird ausgegeben.

**INFORMATION = \*COMMENT**

Der separat gespeicherte Text, d.h. Satzart 2 wird ausgegeben.

**PROTECTION = \*LMS-DEFAULT / \*STD / \*BY-SOURCE**

Einstellung und Aktivierung eines Zugriffsschutzmechanismus für die zu erzeugende Datei, der dem wirksamen Elementenschutz entspricht. Dieser Operand ist wirkungslos, wenn die Elementdaten in die aktuelle EDT-Arbeitsdatei geschrieben werden.

**PROTECTION = \*STD**

Der wirksame Elementschutz des Elementes geht nicht in die Einstellung des Zugriffsschutzmechanismus für die zu erzeugende Datei ein.

**PROTECTION = \*BY-SOURCE**

Die zu erzeugende Datei erhält als Dateischutz einen Zugriffsschutzmechanismus, der dem für das Element wirksamen Elementschutz entspricht.

*Hinweis*

Bei PROTECTION=\*BY-SOURCE erhält die Datei als Dateischutz mindestens den Zugriffsschutzmechanismus Einfache-Zugriffskontrollliste (BACL), auch wenn auf den Zugriffsrechten des Elementes (read, write, execute) kein zusätzlicher Schutz wirkt. Die möglicherweise im Element gespeicherten Werte der Dateiattribute ACCESS und USER-ACCESS (Satzart 164) werden, falls gewünscht (siehe FILE-ATTRIBUTES), für die Einstellung der Werte der Standard-Zugriffskontrolle herangezogen, unabhängig von der Belegung des Operanden PROTECTION.



**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

Überschreiben einer Datei gleichen Namens.

Existiert die Datei nicht unter diesem Namen, wird sie neu angelegt.

Werden die Elementdaten in die aktuelle EDT-Arbeitsdatei geschrieben, so ist die Angabe wirkungslos, d.h. bei existierenden Daten in der aktuellen Arbeitsdatei werden diese um die Elementdaten erweitert bzw. bei einer leeren Arbeitsdatei werden die Elementdaten an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*CREATE**

Die neue Datei darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Die Datei muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*EXTEND**

Die Datei wird erweitert, wenn sie bereits existiert. Ansonsten wird sie neu angelegt.

**WRITE-MODE = \*ANY**

Die Datei wird ersetzt, wenn sie bereits existiert. Ansonsten wird sie neu angelegt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0053	Element- und Dateiattribute unterschiedlich
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
2	0	LMS0199	Satzlänge ungültig bei festem Satzformat
2	0	LMS0274	Blockkontroll-Wert verändert
2	0	LMS0286	Dateiattribute nicht modifiziert
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0035	Elementschutz auf Datei nicht übertragbar
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei LIBRARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgenommen.

Bei TO-FILE: Lese- und Schreibrecht für Datei

Bei PROTECTION=\*BY-SOURCE ist eine Ausgabe von Elementen als Dateien nur für die eigene Kennung möglich, und nur der Eigentümer der Bibliotheksdatei kann diese Funktionalität nutzen.

*Hinweis*

Wird PROTECTION=\*BY-SOURCE angegeben, ist zu beachten:

Die Funktionalität wird abgewiesen, wenn für ein Element auf mindestens einer Zugriffsart (read, write, execute) der Mechanismus Schutz durch GUARD wirkt und auf mindestens einer Zugriffsart (r, w, x) kein spezieller Schutz oder Schutz durch BACL abweichend von USER=NONE wirkt.

Bestehende Kennwörter im Elementschutz werden nicht berücksichtigt.

Bei der Ausgabe von Elementen auf bestehende Dateien, die durch Schutzattribute des Kennwortschutzes zusätzlich geschützt sind, werden bei Anwendung des Operanden PROTECTION=\*BY-SOURCE diese Attribute auf NONE (löscht ein ggf. vorhandenes Kennwort) gesetzt.

*Beispiele*

- Das Element ELEM1 wird mit EXTRACT-ELEMENT in die Datei TEST mit den spezifizierten Dateieigenschaften ausgegeben.

```
/ADD-FILE-LINK FILE-NAME=TEST, LINK-NAME=OUT, ACCESS-METHOD=*SAM, -
/          RECORD-FORMAT=*VARIABLE
/START-LMS
//MOD-LOGG-PAR LOGG=*MAX
//OPEN-LIBRARY LIBRARY=LIBIN
//EXTRACT-ELEMENT *LIB(LIBIN, ELEM1, S), TO-FILE=*LINK(LINK-NAME-OUT)
INPUT  LIBRARY= :10SQ:$USER.LIBIN
OUTPUT FILE
        EXTRACT (S)ELEM1/@(0001)/2011-03-27 AS :10SQ:$USER.TEST
        , REPLACED EMPTY FILE
```

- Sollen alle Elemente einer Bibliothek unter ihrem Namen ausgegeben werden, ist folgende Anweisung anzugeben:

```
//EXTRACT-ELEMENT *LIB(ELEM=*ALL, TYPE=*ALL)
INPUT  LIBRARY= :10SQ:$USER.LIBIN
OUTPUT FILE
        EXTRACT (S)ELEM1/@(0001)/2011-03-27 AS :10SQ:$USER.ELEM1
        EXTRACT (S)EXT.FILE.1/@(0001)/2011-03-27 AS :10SQ:$USER.EXT.FILE.1
        EXTRACT (S)EXT.FILE.2/@(0001)/2011-03-27 AS :10SQ:$USER.EXT.FILE.2
        EXTRACT (S)EXT.FILE.3/@(0001)/2011-03-27 AS :10SQ:$USER.EXT.FILE.3
        EXTRACT (S)EXT.FILE.4/@(0001)/2011-03-27 AS :10SQ:$USER.EXT.FILE.4
        EXTRACT (S)EXT.FILE.5/@(0001)/2011-03-27 AS :10SQ:$USER.EXT.FILE.5
//END
```

## FIND-ELEMENT

### Elemente mit Muster in Sätzen finden

Die Anweisung FIND-ELEMENT durchsucht Elemente nach dem angegebenen Muster und protokolliert die gefundenen Inhaltsverzeichniseinträge wahlweise mit oder ohne Trefersätze.

#### FIND-ELEMENT

```

ELEMENT = *LIBRARY-ELEMENT (...)
  *LIBRARY-ELEMENT(...)
    LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
      *LINK(...)
        | LINK-NAME = <structured-name 1..8> / <filename 1..8>
    ,ELEMENT = *ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)
      *ALL(…)
        | VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT /
          <composed-name 1..24 with-under with-wild(52)>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
          <composed-name 1..64 with-under with-wild(132)>(…)
        | VERSION = *HIGHEST-EXISTING / *ALL / *UPPER-LIMIT /
          <composed-name 1..24 with-under with-wild(52)>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    ,TYPE = *LMS-DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
    ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(…)
      *INTERVAL(…)
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
    ,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(…)
      *INTERVAL(…)
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 1 von 3)

```

,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    | *ANY(...)
    |   |
    |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   <composed-name 1..24 with-under with-wild(52)>
    |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | <composed-name 1..64 with-under with-wild(132)>(…)
    |   |   |
    |   |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   |   <composed-name 1..24 with-under with-wild(52)>
    |   |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
    |   | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>
    |   | ,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>
    |   | ,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)

```

,PATTERN = <c-string 1..256> / *LOWER-CASE(...)
  *LOWER-CASE(...)
    | PATTERN = <c-string 1..256 with-low>
,SHOW-RECORDS = *YES (...) / *NO
  *YES(...)
    | RECORD-NUMBER = *NO / *YES
,INFORMATION = *STD / *ALL / list-poss(2): *TEXT / *COMMENT
TEXT-OUTPUT = *LOGGING-PARAMETERS / *NONE / *SYSOUT / *SYSLST(...) / *EDT(...)
  *SYSLST(...)
    | SYSLST-NUMBER = *STD / <INTEGER 1..99>
  *EDT(...)
    | WRITE-MODE = *EXTEND / *REPLACE
,STRUCTURE-OUTPUT = *SYSINF / *NONE / <composed-name 1..255>(...)
  <composed-name 1..255>(...)
    | WRITE-MODE = *REPLACE / *EXTEND

```

(Teil 3 von 3)

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

Angabe des Elementes, das durchsucht werden soll.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, die durchsucht werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, aus der das Element genommen werden soll.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek.

**ELEMENT = \*ALL (...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Angabe des Elementes, das durchsucht werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /****<composed-name 1..24 with-under with-wild(52)>**

Version, die das zu durchsuchende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird verwendet.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird durchsucht.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das durchsucht werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das zu durchsuchende Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das zu durchsuchende Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird durchsucht.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird durchsucht.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden durchsucht.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieses Kommandos.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element vom Durchsuchen ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die vom Durchsuchen ausgeschlossen sind. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe \*LIBRARY-ELEMENT.

**PATTERN = <c-string 1..256> / \*LOWER-CASE(...)**

Suchmuster als regulärer Ausdruck (siehe Handbuch „C-Bibliotheksfunktionen (BS2000/OSD) für POSIX-Anwendungen“ [16]). Die Groß- und Kleinschreibung wird nicht beachtet.

**PATTERN = \*LOWER-CASE(...)**

Keine Umsetzung der Kleinbuchstaben.

**PATTERN = <c-string 1..256 with-low>**

Suchmuster wie oben, jedoch wird die Groß- und Kleinschreibung beachtet.

**SHOW-RECORDS = \*YES(...) / \*NO**

Bestimmt, ob die Treffersätze gezeigt werden oder nicht.

**SHOW-RECORDS = \*YES(...)**

Die Treffersätze werden gezeigt. Jedes gefundene Element wird in einer Überschrift, gefolgt von den Treffersätzen, ausgegeben (ähnlich der SHOW-ELEMENT Ausgabe). Zusätzlich wird die Anzahl der gefundenen und die Anzahl der durchsuchten Sätze pro Element ausgegeben. Zum Schluss wird die Anzahl der gefundenen und die Anzahl der durchsuchten Elemente ausgegeben.

**RECORD-NUMBER = \*NO / \*YES**

Bestimmt die Ausgabe der Satznummern.

**RECORD-NUMBER = \*NO**

Es werden keine Satznummern ausgegeben.

**RECORD-NUMBER = \*YES**

Die Satznummer wird zusätzlich ausgegeben.

**SHOW-RECORDS = \*NO**

Die Treffersätze werden unterdrückt. Die Ausgabe der gefundenen Elemente erfolgt wie bei SHOW-ELEMENT-ATTRIBUTES ... INFORMATION=\*MEDIUM. Zusätzlich wird die Anzahl der gefundenen und die Anzahl der durchsuchten Elemente ausgegeben.



**INFORMATION = \*STD / \*ALL / list-poss(2): \*TEXT / \*COMMENT**

Der zu bearbeitende Teilbereich des Elementes.

**INFORMATION = \*STD**

Wirkt bei Textelementen wie \*TEXT, sonst wie \*ALL.

**INFORMATION = \*ALL**

Es wird das ganze Element, d.h. Satzart 1-159 und 164 durchsucht.

**INFORMATION = \*TEXT**

Es wird der eigentliche Text, d.h. Satzart 1 durchsucht.

**INFORMATION = \*COMMENT**

Es wird der separat gespeicherte Kommentar, d.h. Satzart 2 durchsucht

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(…)**

Strukturierte Ausgabe der gefundenen Elemente.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch /ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe [12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variable die strukturierte Ausgabe erfolgen soll. Diese Variable muss als dynamische Listenvariable deklariert sein.

(Kommando: /DECLARE-VARIABLE NAME=... (TYPE =\*STRUCTURE), MULTIPLE-ELEMENTS =\*LIST)

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

**Anweisungs-Returncode**

<b>(SC2)</b>	<b>SC1</b>	<b>Maincode</b>	<b>Bedeutung</b>
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

## Notwendige Zugriffsrechte

Bei ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgeschlossen.

### Beispiele

- Die Elemente der Bibliothek X sollen nach der Zeichenfolge „abc“ durchsucht werden.

```
//FIND-ELEMENT *LIB(LIB=X,ELEM=*,TYPE=S),PATTERN='abc',SHOW-RECORDS=*YES
INPUT LIBRARY= :10SQ:$USER.X
INPUT ELEMENT= (S)FIND/@(0001)/2012-10-09
abc ist in diesem Satz.
```

```
Dies ist ein Satz mit abcdefg.
2 OUT OF 127 RECORD(S) FOUND
```

```
INPUT LIBRARY= :10SQ:$USER.X
1 OUT OF 2 (S)-ELEMENT(S) FOUND
```

```
//FIND-ELEMENT *LIB(LIB=X,ELEM=*,TYPE=S),PATTERN='abc', -
SHOW-RECORDS=*YES(,RECORD-NUMBER=*YES)
```

```
INPUT LIBRARY= :10SQ:$USER.X
INPUT ELEMENT= (S)FIND/@(0001)/2012-10-09
#7 > abc ist in diesem Satz.
```

```
#123 >Dies ist ein Satz mit abcdefg.
2 OUT OF 127 RECORD(S) FOUND
```

```
INPUT LIBRARY= :10SQ:$USER.X
1 OUT OF 2 (S)-ELEMENT(S) FOUND
```

```
//FIND-ELEMENT *LIB(LIB=X,ELEM=*,TYPE=S),PATTERN='abc',SHOW-RECORDS=*NO
```

```
INPUT LIBRARY= :10SQ:$USER.X
TYP NAME VER (VAR#) DATE
(S) FIND @ (0001) 2012-10-09
1 OUT OF 2 (S)-ELEMENT(S) FOUND
```

- Das Element LLM vom Typ L der Bibliothek X soll nach 'SYSLNK' durchsucht werden, und 5 Zeichen vor dem Treffer und alle Zeichen hinter dem Treffer (maximal eine Bildschirmzeile) sollen ausgegeben werden.

```
//FIND-ELEMENT *LIB(LIB=X,ELEM=LLM,TYPE=L),PATTERN='.....SYSLNK.*'
INPUT LIBRARY= :1SOSQ:$USER.X
```

```
INPUT ELEMENT= (L)LLM/@(0001)/2012-10-09
RECORD-TYPE: 160
$.SYSLNK.LMS.034
```

```
MSLMSSYSLNK.LMS
2 OUT OF 7 RECORD(S) FOUND
```

```
INPUT LIBRARY= :10SQ:$USER.X
1 OUT OF 1 (L)-ELEMENT(S) FOUND
```

## MODIFY-ELEMENT

### Ändern von Bindemodulen, Phasen und Bindelademodulen

Die Anweisung MODIFY-ELEMENT leitet die Modifikation von Elementen ein. Die Änderungen selber werden über MODIFY-ELEMENT-Subanweisungen gesteuert (siehe [Seite 308](#)).

Durch MODIFY-ELEMENT werden die Elemente ausgewählt, die geändert werden sollen. Der Basistyp für Quell- und Zielelement darf unterschiedlich sein, wenn Textelemente modifiziert werden.

Wurde die MODIFY-ELEMENT-Anweisung abgeschickt, erwartet LMS als nächste Anweisung eine Subanweisung. Wird statt einer Subanweisung eine andere Anweisung eingegeben, wird eine Fehlermeldung ausgegeben.

#### MODIFY-ELEMENT

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

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

**LIBRARY** = \*STD / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)

**\*ALL**(...)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

**,TYPE** = \*LMS-DEFAULT / <alphanum-name 1..8>

(Teil 1 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(...)
    | *ANY(...)
    |   |
    |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   <composed-name 1..24 with-under with-wild(52)>
    |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | <composed-name 1..64 with-under with-wild(132)>(...)
    |   |   |
    |   |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   |   <composed-name 1..24 with-under with-wild(52)>
    |   |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8>
    |   | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |   |
    |   |   | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |   | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)

```

,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    |   FROM = 1900-01-01 / <date 8..10 with-compl>
    |   ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    |   FROM = 1900-01-01 / <date 8..10 with-compl>
    |   ,TO = *TODAY / <date 8..10 with-compl>
,TO-ELEMENT = *LIBRARY-ELEMENT (...)
  *LIBRARY-ELEMENT(...)
    |
    |   LIBRARY = *STD / *BY-SOURCE / <filename 1..54 without-vers> / *LINK(...)
    |   *LINK(...)
    |     |
    |     |   LINK-NAME = <structured-name 1..8>
,ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wild-constr>(…)
  *BY-SOURCE(...)
    |
    |   VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
    |             *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
    |   ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   <composed-name 1..132 with-under with-wildcard-constr>(…)
    |     |
    |     |   VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
    |             *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
    |     |   ,BASE = *STD / <composed-name 1..24 with-under with-wild>
,TYPE = *BY-SOURCE / *LMS-DEFAULT / <alphanum-name 1..8>
,USER-DATE = *TODAY / *BY-SOURCE / <date 8..10 with-compl>
,TEXT-PARAMETERS = *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    |   INPUT-RECORD-ID = *NONE / *RECORD-PART(...)
    |   *RECORD-PART(...)
    |     |
    |     |   START = <integer 1..251>
    |     |   ,LENGTH = <integer 1..16>
,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *ANY
,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD/ <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das zu ändernde Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das zu ändernde Element befindet.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Angabe des zu ändernden Elements.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das zu ändernde Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird geändert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird geändert.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elements, das geändert werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / <alphanum 1..8>**

Typ, den das zu ändernde Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das zu ändernde Element besitzt ein beliebiges Datum.



**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird geändert.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird geändert.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden geändert.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elements. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element vom Korrigieren ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die vom Ändern ausgeschlossen sind. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

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

Angabe, wohin und unter welchem Namen das korrigierte Element zurückgeschrieben werden soll.

**LIBRARY = \*STD / \*BY-SOURCE / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in die das korrigierte Element zurückgeschrieben werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

**LIBRARY = \*BY-SOURCE**

Das korrigierte Element wird in die ursprüngliche Bibliothek zurückgeschrieben.

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

Name der Bibliothek, in die das korrigierte Element aufgenommen werden soll.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*BY-SOURCE(...) /****<composed-name 1..132 with-under with-wild-constr>(…)**

Name, den das korrigierte Element erhalten soll.

**ELEMENT = \*BY-SOURCE(...)**

Der neue Name entspricht dem alten Namen.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /****<composed-name 1..52 with-under with-wild-constr>**

Version, die das korrigierte Element erhalten soll.

**VERSION = \*BY-SOURCE**

Das korrigierte Element erhält die gleiche Version wie das ursprüngliche Element. Hat das ursprüngliche Element keine Versionsangabe, erhält das korrigierte Element 'X'FF' als Versionsangabe.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version 'X'FF' wird erzeugt.

**VERSION = <composed-name 1..52 with-under with-wild-constr>**

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name des korrigierten Elements. Er kann auch mit Wildcards eingegeben werden.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /**

**<composed-name 1..52 with-under with-wild-constr>**

Version, die das korrigierte Element erhalten soll.

Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe

[Seite 54](#).

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / <alphanum-name 1..8>**

Typ, den das korrigierte Element erhalten soll.

**TYPE = \*BY-SOURCE**

Das korrigierte Element erhält die gleiche Typbezeichnung wie das ursprüngliche Element.

**USER-DATE = \*TODAY / \*BY-SOURCE / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = \*BY-SOURCE**

Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**TEXT-PARAMETERS = \*NONE / \*PARAMETERS(…)**

Legt Parameter für Textelemente fest.

**TEXT-PARAMETERS = \*NONE**

Es werden keine Parameter für Textelemente festgelegt.

**TEXT-PARAMETERS = \*PARAMETERS(…)**

Es werden Parameter für Textelemente festgelegt.

**INPUT-RECORD-ID = \*NONE / \*RECORD-PART(…)**

Legt den Bereich der Satzkennung (siehe Subanweisungen für Textelemente auf [Seite 308](#)) im Eingabesatz fest.

**INPUT-RECORD-ID = \*NONE**

Es wird keine Satzkennung im Eingabesatz festgelegt.

**INPUT-RECORD-ID = \*RECORD-PART(...)**

Beginn und Länge des Satzkennungsbereichs werden festgelegt.

Es muss gelten: Beginn + Länge ≤ 252

**START = <integer 1..251>**

Beginn des Satzkennungsbereichs durch Angabe des ersten Zeichens im Satz.

**LENGTH = <integer 1..16>**

Länge der Satzkennung

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

Ist Quellelement = Zielelement so wird der Operand WRITE-MODE ignoriert.

**WRITE-MODE = \*CREATE**

Der Name des korrigierten Elements darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das korrigierte Element muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*ANY**

Das korrigierte Element wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS (siehe [Seite 339](#)).

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgeschlossen.

Bei TO-ELEMENT: Lese- und Schreibrecht für LIBRARY

Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).

Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Es kann nur eine neue Version erzeugt oder diese Basisversion überschrieben werden, sofern Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

*Hinweise*

- Beim Anlegen eines Elementes ist die unter dem Typ geltende Konvention zu beachten.
- Bei aktivem WRITE-CONTROL in der Ausgabebibliothek wird dem zu schreibenden Element von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorganges protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Bereits vorhandene oder durch INFORMATION=\*COMMENT editierte Kommentarsätze werden dahinter kopiert. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), so werden die Elementattribute STATE, HOLDER und alle Rechte von der Basisversion auf die neue Version übernommen. Die CCSN wird von dem Quellelement übernommen. Das USER-DATE wird neu bestimmt.

## MODIFY-ELEMENT-Subanweisungen für Elementtypen R, C und L

Diese Subanweisungen gelten für Elemente vom Basistyp R, C und L und müssen direkt auf die MODIFY-ELEMENT-Anweisung folgen. Sie werden bis zur MODIFY-ELEMENT-Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

Die MODIFY-ELEMENT-Subanweisungen führen Änderungen in Bindemodulen, Phasen und Bindelademodulen durch.

LMS sammelt zunächst diese Subanweisungen und führt sie erst nach Eingabe der Subanweisung END-MODIFY aus.

### Übersicht der MODIFY-ELEMENT-Subanweisungen

Diese Subanweisungen sind abhängig vom gewählten Elementtyp und nur für Elemente der Typen R, C oder L erlaubt.

MODIFY-ELEMENT-Subanweisung	Elementtyp	Funktion
ADD-REP-RECORD	R	Erzeugen von REP-Sätzen
ADD-TEXT-MODIFICATION	R,C,L	Ändern von Textsätzen
DELETE-RECORD-TYPE	R,C,L	Löschen von Satzarten
END-MODIFY	R,C,L	Beenden der Änderungen
MODIFY-CSECT-ATTRIBUTES	R	Ändern der CSECT-Attribute
MODIFY-MODIFICATION-DEFAULTS	R,C,L	Festlegen der globalen Parameter in der MODIFY-ELEMENT-Anweisung
REMOVE-MODIFICATION	R,C,L	Zurücknehmen von Korrekturen
RENAME-SYMBOLS	R	Umbenennen von CSECT-, ENTRY-, EXTRN- und COMMON-Namen

#### *Hinweis*

Als MODIFY-ELEMENT-Subanweisungen sind auch die SDF-Standardanweisungen (siehe [Seite 145](#)) zulässig.

## ADD-REP-RECORD

### REP-Sätze erzeugen

Die MODIFY-ELEMENT-Subanweisung ADD-REP-RECORD fügt REP-Sätze in den Bindemodul ein. Diese REP-Sätze wertet der Dynamische Bindelader (DBL) aus.

ADD-REP-RECORD ist nur für Bindemodule (Elemente des Typs R) zulässig.

ADD-REP-RECORD
<pre> ADDRESS = &lt;x-string 1..8&gt;(…)   &lt;x-string 1..8&gt;(…)         <b>BASE-ADDRESS</b> = *<u>MODIFICATION-DEFAULT</u> / &lt;x-string 1..8&gt; ,NEW-CONTENTS = &lt;x-string 1..100&gt; / &lt;c-string 1..50 with-low&gt; </pre>

### ADDRESS = <x-string 1..8>(…)

Angabe der Adresse, an der das durch MODIFY-ELEMENT gewählte Element geändert werden soll.

**BASE-ADDRESS = \*MODIFICATION-DEFAULT / <x-string 1..8>**

Basisadresse. BASE-ADDRESS wird auf ADDRESS addiert. Die so entstehende Korrekturadresse muss bei Großmoduln großmodul-relativ (nicht CSECT-relativ) sein.

### NEW-CONTENTS = <x-string 1..100> / <c-string 1..50 with-low>

Ersetzungstext, der zeichenweise oder sedezial angegeben wird.

Wird der Text zeichenweise angegeben, darf er höchstens 50 Zeichen lang sein. Ein Hochkomma im Text muss doppelt angegeben werden.

Wird der Text sedezial angegeben, darf er höchstens 100 Zeichen lang sein.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler



**ADD-TEXT-MODIFICATION****Textsätze ändern**

Die MODIFY-ELEMENT-Subanweisung ADD-TEXT-MODIFICATION korrigiert Textsätze eines Bindeladomoduls und Phasen. Die Subanweisung erzeugt einen Korrekturjournal-satz (TXTP-Satz), der den ursprünglichen Inhalt des Textbereiches enthält. Mit der Anweisung MODIFY-MODIFICATION-DEFAULTS kann eingestellt werden, dass kein Korrekturjournal-satz erzeugt wird. Korrekturen ohne Korrekturjournal-satz sind über die Subanweisung REMOVE-MODIFICATION nicht rückgängig zu machen.

Sie ist für Elemente der Typen R, C und L zulässig.

**ADD-TEXT-MODIFICATION**

```

ADDRESS = <x-string 1..8>(…)
  <x-string 1..8>(…)
    | BASE-ADDRESS = *MODIFICATION-DEFAULT / <x-string 1..8>
,NEW-CONTENTS = <x-string 1..100>(…) / <c-string 1..50 with-low>(…)
  <x-string 1..100>(…)
    | OLD-CONTENTS = *ANY / <x-string 1..100> / <c-string 1..50 with-low>
  <c-string 1..50 with-low>(…)
    | OLD-CONTENTS = *ANY / <x-string 1..100> / <c-string 1..50 with-low>
,MODIFICATION-ID = *MODIFICATION-DEFAULT / *SPACES / <c-string 1..12 with-low>

```

**ADDRESS = <x-string 1..8>(…)**

Angabe der Adresse, an der das durch MODIFY-ELEMENT gewählte Element geändert werden soll.

**BASE-ADDRESS = \*MODIFICATION-DEFAULT / <x-string 1..8>**

Basisadresse.

Die Basisadresse wird auf ADDRESS addiert. Die so entstehende Korrekturadresse ist:

bei ...	relativ zu:
Moduln	CSECT-Anfang (das gewünschte CSECT wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben.)
Phasen	Phasen-Anfang
LLMs	CSECT-Anfang, wenn ein CSECT angegeben wurde
	SUB-LLM-Anfang, wenn ein Sub-LLM angegeben wurde (der gewünschte SUB-LLM wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben)
	SLICE-Anfang, wenn eine SLICE angegeben wurde (die gewünschte SLICE wird über die Subanweisung MODIFY-MODIFICATION-DEFAULTS angegeben)
	LLM-Anfang, wenn nichts angegeben wurde und das LLM nur aus einem SLICE besteht. Besteht das LLM aus mehreren SLICES, ist die Angabe einer CSECT, eines Sub-LLM oder eines SLICES nötig.

**NEW-CONTENTS = <x-string 1..100>(…) / <c-string 1..50 with-low>(…)**

Ersetzungstext, der zeichenweise oder sedezimal angegeben wird.

**OLD-CONTENTS = \*ANY / <x-string 1..100> / <c-string 1..50 with-low>**

Originaltext des Elementes. Der Originaltext muss immer in der Länge des Ersetzungstextes angegeben werden.

**MODIFICATION-ID = \*MODIFICATION-DEFAULT / \*SPACES / <c-string 1..12 with-low>**

Identifikation, die im Änderungs Journalsatz (TXTP-Satz) gehalten wird. Wird SPACES angegeben, so werden als Identifikation Blanks verwendet.

Für Elementtyp R und C sind nur 8 Zeichen erlaubt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

**DELETE-RECORD-TYPE****Löschen von Satzarten**

Die MODIFY-ELEMENT-Subanweisung DELETE-RECORD-TYPE schließt aus dem Eingabeelement folgende Satzarten aus:

- ISD-Sätze (gilt nur für Typ R Elemente)
- LSD-Sätze (gilt nur für Typ R Elemente)
- REP-Sätze (gilt nur für Typ R Elemente)
- INCLUDE-Sätze (gilt nur für Typ R Elemente)
- TXTP-Sätze (gilt für Typ R, C und L Elemente)
- DSDD-Sätze (gilt nur für Typ R Elemente)

Sie ist für Elemente der Typen R, C und L zulässig.

**DELETE-RECORD-TYPE**

**TYPE = \*TXTP(...)/ list-poss(5): \*ISD / \*LSD / \*REP / \*DSDD / \*INCLUDE**

**\*TXTP(...)**

    | **MODIFICATION-ID = \*ALL / \*SPACES / <c-string 1..12 with-low>**

**TYPE = \*TXTP(...)/ list-poss(5): \*ISD / \*LSD / \*REP / \*DSDD / \*INCLUDE**

Legt die Satzart fest, die nicht vom Eingabeelement ins Ausgabeelement übernommen werden soll.

**MODIFICATION-ID = \*ALL / \*SPACES / <c-string 1..12 with-low>**

Es werden nur die TXTP-Sätze mit der angegebenen Identifizierung gelöscht.

Für Elementtyp R und C sind nur 8 Zeichen erlaubt.

Diese Identifizierung gilt nur für dieses DELETE-RECORD-TYPE.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweis*

Gelöschte Satzarten sind unwiderruflich verloren.

**END-MODIFY****Beenden der MODIFY-ELEMENT-Subanweisungen für Elementtypen R, C und L**

END-MODIFY schließt die Folge der MODIFY-ELEMENT-Subanweisungen ab. Anschließend prüft LMS alle Anweisungen auf Durchführbarkeit und führt die Anweisungsfolge aus.

END-MODIFY

Diese Subanweisung besitzt keine Operanden.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**MODIFY-CSECT-ATTRIBUTES****Verändern von Programmabschnittsmerkmalen**

Die MODIFY-ELEMENT-Subanweisung MODIFY-CSECT-ATTRIBUTES verändert Programmabschnittsmerkmale.

Sie ist nur für Bindemodule (Elemente des Typs R) zulässig.

**MODIFY-CSECT-ATTRIBUTES**

```

NAME = *ALL / <c-string 1..8 with-low> / <text 1..8>
,VISIBLE = *UNCHANGED / *YES / *NO
,READ-ONLY = *UNCHANGED / *YES / *NO
,PAGE-ALIGNMENT = *UNCHANGED / *YES / *NO
,RESIDENCY-MODE = *UNCHANGED / 24 / *ANY
,ADDRESSING-MODE = *UNCHANGED / 24 / 31 / *ANY

```

Zu Beginn der Anweisung MODIFY-ELEMENT sind die Operanden mit dem unmittelbar auf \*UNCHANGED folgenden Wert voreingestellt.

**NAME = \*ALL / <c-string 1..8 with-low> / <text 1..8>**

Name des CSECTs, dessen Attribute geändert werden sollen. Es können alle CSECTs oder ein spezielles CSECT angegeben werden.

**VISIBLE = \*UNCHANGED / \*YES / \*NO**

Maskierung (Sichtbarkeit) der Programmabschnitte.

**VISIBLE = \*YES**

Die angegebenen Programmabschnitte werden nicht maskiert (siehe [5]).

Für diese Abschnitte wird ein Sekundärnamenssatz angelegt und die Namen werden im Sekundärnamensverzeichnis eingetragen.

**VISIBLE = \*NO**

Die angegebenen Programmabschnitte werden maskiert. Für sie wird weder ein Sekundärnamenssatz angelegt noch werden die Namen im Sekundärnamensverzeichnis eingetragen. Ein eventuell vorhandener Sekundärnamenssatz wird gelöscht.

Werden alle Programmabschnitte eines Bindemoduls maskiert, wird ein Bibliothekselement ohne Sekundärnamenseintrag erzeugt. Dieser Bindemodul ist nur über Primärnamen auffindbar.

Der Modulname kann jedoch aus dem ersten Programmabschnittsnamen mit Hilfe aller ESD-Sätze abgeleitet werden, da dazu auch maskierte Programmabschnitte verwendet werden.

*Hinweis*

Bindemodule, die nur maskierte Programmabschnitte besitzen, kann der Binder nicht bearbeiten, z.B. Ausschließen eines Bindemoduls bei der Autolink-Funktion.  
Der Operand **VISIBLE** ist auch auf **ENTRY**s anwendbar.

**READ-ONLY = \*UNCHANGED / \*YES / \*NO**

Schreibschutz

**READ-ONLY = \*YES**

Legt fest, dass die angegebenen Programmabschnitte zur Ablaufzeit des Programms nur gelesen werden dürfen.

**READ-ONLY = \*NO**

Erlaubt, dass während des Programmlaufs auch in die angegebenen Programmabschnitte geschrieben werden darf.

**PAGE-ALIGNMENT = \*UNCHANGED / \*YES / \*NO**

Seitenausrichtung

**PAGE-ALIGNMENT = \*YES**

Legt fest, dass die angegebenen Programmabschnitte auf Seitengrenze ausgerichtet werden sollen, d.h. die Ladeadresse soll ein Vielfaches von dezimal 4096 bzw. sedezimal 1000 sein.

**PAGE-ALIGNMENT = \*NO**

Lässt Seitengrenzen unberücksichtigt. Die Programmabschnitte beginnen jeweils bei der nächsten Doppelwortadresse, die sich beim Binden ergibt.

**RESIDENCY-MODE = \*UNCHANGED / 24 / \*ANY**

Lademodus

**RESIDENCY-MODE = 24**

Legt fest, dass die angegebenen Programmabschnitte in den Adressraum unterhalb der 16 MB Grenze zu laden sind.

**RESIDENCY-MODE = \*ANY**

Es existiert keine Einschränkung.

**ADDRESSING-MODE = \*UNCHANGED / 24 / 31 / \*ANY**

Ablaufmodus

**ADDRESSING-MODE = 24**

Legt fest, dass die angegebenen Programmabschnitte im 24-Bit-Modus ablauffähig sein sollen.

**ADDRESSING-MODE = 31**

Legt fest, dass die angegebenen Programmabschnitte im 31-Bit-Modus ablauffähig sein sollen.

**ADDRESSING-MODE = \*ANY**

Der Ablaufmodus ist beliebig.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

## MODIFY-MODIFICATION-DEFAULTS

### Festlegen von globalen Parametern

Die MODIFY-ELEMENT-Subanweisung MODIFY-MODIFICATION-DEFAULTS legt innerhalb der MODIFY-ELEMENT-Anweisung die globalen Defaultwerte fest.

Sie ist für Elemente der Typen R, C und L zulässig.

MODIFY-MODIFICATION-DEFAULTS
<pre> <b>CSECT-NAME</b> = <u>*UNCHANGED</u> / *NONE / &lt;c-string 1..32 with-low&gt; / &lt;text 1..32&gt; , <b>PHASE-SEGMENT</b> = <u>*UNCHANGED</u> / *ROOT / &lt;name 1..8&gt; , <b>LLM-PART</b> = <u>*UNCHANGED</u> / *NONE / *SLICE(...) / *SUB-LLM(...)   *SLICE(...)       <b>NAME</b> = &lt;structured-name 1..32&gt;   *SUB-LLM(...)       <b>PATH-NAME</b> = &lt;c-string 1..255 with-low&gt; / &lt;text 1..255&gt; , <b>MODIFICATION-LOGGING</b> = <u>*UNCHANGED</u> / *YES(...) / *NO   *YES(...)       <b>MODIFICATION-ID</b> = <u>*UNCHANGED</u> / *SPACES / &lt;c-string 1..12 with-low&gt; , <b>BASE-ADDRESS</b> = <u>*UNCHANGED</u> / &lt;x-string 1..8&gt; </pre>

Zu Beginn der Anweisung MODIFY-ELEMENT sind die Operanden mit dem unmittelbar auf \*UNCHANGED folgenden Wert voreingestellt.

**CSECT-NAME = \*UNCHANGED / \*NONE / <c-string 1..32 with-low> / <text 1..32>**

Name des zu korrigierenden CSECTs. (Nur relevant für Typ R und L.)

**CSECT-NAME = \*NONE**

Wird kein CSECT-Name angegeben, wird bei R-Moduln der erste CSECT-Name verwendet.

**PHASE-SEGMENT = \*UNCHANGED / \*ROOT / <name 1..8>**

Angabe des Phasen-Segmentes, das korrigiert werden soll. Wird kein Segment angegeben, wird das erste Segment (\*ROOT) verwendet.



**LLM-PART = \*UNCHANGED / \*NONE / \*SLICE(...) / \*SUB-LLM(...)**

Wird kein LLM-Teil angegeben, wird der gesamte LLM genommen.

**LLM-PART = \*SLICE(...)**

Angabe der SLICE, die korrigiert werden soll.

**NAME = <structured-name 1..32>**

Name der SLICE, die korrigiert werden soll.

**LLM-PART = \*SUB-LLM(...)**

Angabe des SUB-LLM, der korrigiert werden soll.

**PATH-NAME = <c-string 1..255 with-low> / <text 1..255>**

Der zu korrigierende SUB-LLM wird durch seinen Pfadnamen bestimmt.

**MODIFICATION-LOGGING = \*UNCHANGED / \*YES(...) / \*NO**

Bestimmt die Erzeugung der TXTP-Sätze.

**MODIFICATION-LOGGING = \*YES(...)**

Es sollen TXTP-Sätze erzeugt werden.

**MODIFICATION-ID = \*UNCHANGED / \*SPACES / <c-string 1..12 with-low>**

Identifikation, die im Änderungsjournalatz (TXTP-Satz) gehalten wird. Wird SPACES angegeben, so werden als Identifikation Blanks verwendet.

Für Elementtyp R und C sind nur 8 Zeichen erlaubt

**MODIFICATION-LOGGING = \*NO**

Es sollen keine TXTP-Sätze erzeugt werden.

**BASE-ADDRESS = \*UNCHANGED / <x-string 1..8>**

Sedezimale Angabe der Basisadresse. Zu Beginn der Anweisung MODIFY-ELEMENT ist die Basisadresse 0 eingestellt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

**REMOVE-MODIFICATION****Rückgängigmachen von Korrekturen**

Die MODIFY-ELEMENT-Subanweisung REMOVE-MODIFICATION macht Korrekturen aus einem früheren Korrekturlauf unter den folgenden Voraussetzungen rückgängig:

Es wurde ein Korrekturjournalsatz mit der MODIFY-ELEMENT-Subanweisung ADD-TEXT-MODIFICATION erstellt, d.h. der Operand MODIFICATION-LOGGING=\*YES (siehe [Seite 305](#)) war gesetzt.

Sie ist für Elemente der Typen R, C und L zulässig.

<b>REMOVE-MODIFICATION</b>
<b>MODIFICATION-ID = <u>*ALL</u> / *SPACES / &lt;c-string 1..12 with-low&gt;</b>

**MODIFICATION-ID = \*ALL / \*SPACES / <c-string 1..12 with-low>**

Für Elementtyp R und C sind nur 8 Zeichen erlaubt.

Es werden nur die Korrekturen mit der angegebenen Identifikation rückgängig gemacht. Wird eine Identifikation angegeben, müssen dafür Korrekturjournalsätze existieren. Wird keine Identifikation (\*ALL) angegeben, werden alle Korrekturen, für die ein Korrekturjournalsatz vorliegt, rückgängig gemacht.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

**RENAME-SYMBOLS****Umbenennen von Symbolen**

Die MODIFY-ELEMENT-Subanweisung RENAME-SYMBOLS ändert den Namen von einer CSECT, einem ENTRY, COMMON, EXTRN, WXTRN oder einer V-Konstante. Jede Umbenennung verursacht eine Änderung von ESD-Sätzen. LMS prüft die Eindeutigkeit der Namen innerhalb aller ESD-Sätze und weist eine Umbenennung zurück, wenn der neue Name bereits existiert.

In LSD-Sätzen wird eine Umbenennung nicht durchgeführt. Deswegen kann mit AID (siehe [13]) nach einer Änderung des CSECT-Namens nicht mehr symbolisch getestet werden. Wird maschinennah getestet, müssen bei Qualifikationen in AID die neuen Namen verwendet werden.

Die MODIFY-ELEMENT-Subanweisung RENAME-SYMBOLS ist nur für Bindemodule (Elemente des Typs R) zulässig.

**RENAME-SYMBOLS**

**SYMBOL-NAME** = <text 1..8>

**,SYMBOL-TYPE** = \*CSECT / \*ENTRY / \*COMMON / \*EXTRN / \*VCON / \*WXTRN

**,NEW-NAME** = <text 1..8>

**SYMBOL-NAME = <text 1..8>**

Legt den Symbolnamen fest, der umbenannt werden soll.

**SYMBOL-TYPE = \*CSECT / \*ENTRY / \*COMMON / \*EXTRN / \*VCON / \*WXTRN**

Legt den Typ des Symbols fest, dessen Name geändert werden soll.

**NEW-NAME = <text 1..8>**

Neuer Symbolname.

Der Name sollte den BINDER-Konventionen für den Sonderdatentyp <symbol> genügen (siehe [5]). Allerdings prüft LMS nicht auf diese Konvention.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

*Hinweis*

Auch maskierte (invisible) CSECT/ENTRY-Namen können umbenannt werden.

## MODIFY-ELEMENT-Subanweisungen für Textelemente

Diese Subanweisungen führen Änderungen in Textelementen durch. Sie werden bis zur MODIFY-ELEMENT-Subanweisung END-MODIFY aus dem Anweisungsstrom gelesen.

### Übersicht der MODIFY-ELEMENT-Subanweisungen

Diese Subanweisungen sind nur für Textelemente erlaubt.

MODIFY-ELEMENT-Subanweisung	Funktion
ADD-RECORD	Einfügen von Sätzen
END-MODIFY	Beenden der Änderungen
REMOVE-RECORD	Ausfügen von Sätzen

#### *Hinweise*

- Als MODIFY-ELEMENT-Subanweisungen sind auch die SDF-Standardanweisungen (siehe [Seite 145](#)) zulässig.
- Es werden nur Elementsätze der Länge  $\leq 251$  bearbeitet. Längere Sätze werden abgeschnitten. In diesem Fall gibt LMS eine Warnung aus.

### Definition der Satzidentifikation für Textelemente

Die Satzidentifikation kann eine Satznummer oder eine Satzkennung sein.

**Satznummer:** Die Satznummer ist die relative Position des Elementsatzes bezogen auf den Elementanfang. Ist die angegebene Satznummer größer als die höchste Satznummer des Elements, werden die Änderungen nach dem letzten Elementsatz fortgesetzt, d.h. Sätze an das Element angefügt.

**Satzkennung:** Lage und Länge der Satzkennung werden mit dem Operanden INPUT-RECORD-ID (siehe Anweisung MODIFY-ELEMENT) festgelegt. Deshalb darf die Angabe einer Satzkennung in den Subanweisungen nur bei INPUT-RECORD-ID ungleich \*NONE verwendet werden. Die Satzkennung muss in der Länge, die bei INPUT-RECORD-ID vereinbart wurde, angegeben werden. Nur führende Nullen dürfen entfallen. Wenn die Satzkennung im Eingabeelement nicht vorkommt, wird vor dem ersten Satz mit größerer Satzkennung modifiziert.

Satznummer und Satzkenung können in den Subanweisungen gemischt angegeben werden. Sie müssen in den Subanweisungen und Datensätzen immer aufsteigend angegeben werden.

Wurde im Dialog ein Fehler erkannt, muss die Korrektur mit END-MODIFY beendet und die Korrektur neu gestartet werden. Nach einer ADD-RECORD-Subanweisung muss zusätzlich ein \*END eingegeben werden.

## ADD-RECORD

### Einfügen von Sätzen

Die MODIFY-ELEMENT-Subanweisung ADD-RECORD fügt die der Anweisung folgenden Sätze an der angegebenen Position ein. Die einzufügenden Sätze müssen mit einem \*END-Satz abgeschlossen werden.

<b>ADD-RECORD</b>
<b>RECORD-ID = *NONE</b> / <integer 0..99999999> / <c-string 1..16 with-low>

**RECORD-ID = \*NONE** / <integer 0..99999999> / <c-string 1..16 with-low>

Angabe des Datensatzes, der eingefügt werden soll.

### **RECORD-ID = \*NONE**

Ist der Operand INPUT-RECORD-ID der MODIFY-ELEMENT-Anweisung ungleich \*NONE, werden die der ADD-RECORD-Subanweisung folgenden Datensätze gemäß ihrer Satz-  
kennung in das zu modifizierende Element aufgenommen.

Gibt es einen Satz mit der angegebenen Satz-  
kennung, wird er durch den Datensatz  
ersetzt. Gibt es keinen Satz mit der angegebenen Satz-  
kennung, wird der Datensatz vor  
dem ersten Satz mit der höheren Satz-  
kennung eingefügt. Datensätze mit leerer Satz-  
kennung, oder wenn INPUT-RECORD-ID=\*NONE gesetzt ist, werden an aktueller Position ein-  
geordnet.

**RECORD-ID = <integer 0..99999999> / <c-string 1..16 with-low>**

Angabe der Elementposition, hinter der die der Anweisung folgenden Datensätze eingefügt  
werden sollen. Ist die angegebene Satz-  
nummer oder Satz-  
kennung nicht vorhanden, wer-  
den die Datensätze vor dem ersten Satz eingefügt, dessen Satz-  
nummer/Satz-  
kennung größer ist als die angegebene Satz-  
nummer/Satz-  
kennung.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler

**END-MODIFY****Beenden der MODIFY-ELEMENT-Subanweisungen für Textelemente**

END-MODIFY schließt die Folge der MODIFY-ELEMENT-Subanweisungen ab.

END-MODIFY

Diese Subanweisung besitzt keine Operanden.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

## REMOVE-RECORD

### Ausfügen von Sätzen

Die MODIFY-ELEMENT-Subanweisung REMOVE-RECORD löscht den angegebenen Satz oder Satzbereich im Element.

#### REMOVE-RECORD

**RECORD-ID** = <integer 0..99999999> / <c-string 1..16 with-low> / \***RANGE**(...)

\***RANGE**(...)

**FROM** = <integer 0..99999999> / <c-string 1..16 with.low>

**TO** = <integer 0..99999999> / <c-string 1..16 with.low>

**RECORD-ID** = <integer 0..99999999> / <c-string 1..16 with-low> / \***RANGE**(...)

Satznummer oder Satzkenung des zu löschenden Satzes.

**RECORD-ID** = \***RANGE**(...)

Der zu löschende Satzbereich wird bestimmt.

**FROM** = <integer 0..99999999> / <c-string 1..16 with-low>

    Beginn des Bereichs durch Angabe der ersten Satznummer oder Satzkenung.

**TO** = <integer 0..99999999> / <c-string 1..16 with-low>

    Ende des Bereichs durch Angabe der letzten Satznummer oder Satzkenung.

#### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler



## MODIFY-ELEMENT-ATTRIBUTES

### Elementattribute ändern

Mit dieser Anweisung können der Elementname, die Version, das Benutzerdatum, der CCS-Name und der Elementzustand geändert werden. Die Anweisung ist für alle Elementtypen zulässig.

Der Basistyp für Quell- und Zielelement darf unterschiedlich sein, wenn Textelemente bearbeitet werden.

Das Ändern einer Elementbezeichnung ist für Delta-Elemente oder bei aktivem WRITE-CONTROL nicht zugelassen.

#### MODIFY-ELEMENT-ATTRIBUTES

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

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

**LIBRARY** = \*STD / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)

**\*ALL**(...)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

**,TYPE** = \*LMS-DEFAULT / **\*ALL** / <alphanum-name 1..8 with-wild(20)>

(Teil 1 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CODED-CHARACTER-SET = *ANY / *NONE / <name 1..8 with-wild(20)>
,STATE = *ANY / *FREE / *IN-HOLD(...)
  *IN-HOLD(...)
    |
    | HOLDER = *ANY / <name 1..8 with-wild(20)>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    | *ANY(...)
    |   |
    |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   <composed-name 1..24 with-under with-wild(52)>
    |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | <composed-name 1..64 with-under with-wild(132)>(…)
    |   |   |
    |   |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   |   <composed-name 1..24 with-under with-wild(52)>
    |   |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>

```

(Teil 2 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,NEW-ATTRIBUTES = PARAMETERS (...)
    *PARAMETERS(...)
        |
        | ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wild-constr>(…)
        | *BY-SOURCE(...)
        | |
        | | VERSION = *BY-SOURCE / *LMS-DEFAULT / *HIGHEST-EXISTING / *INCREMENT /
        | | *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
        | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
        | | <composed-name 1..132 with-under with-wild-constr>(…)
        | | |
        | | | VERSION = *BY-SOURCE / *LMS-DEFAULT / *HIGHEST-EXISTING / *INCREMENT /
        | | | *UPPER-LIMIT / <composed-name 1..52 with-under with-wild-constr>
        | | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
        | | ,TYPE = *BY-SOURCE / *LMS-DEFAULT / <alphanum-name 1..20 with-wild-constr>
        | | ,USER-DATE = *BY-SOURCE / *TODAY / <date 8..10 with-compl>
        | | ,MODIFICATION-DATE = *BY-SOURCE / *SYSTEM-DATE
        | | ,CODED-CHARACTER-SET = *BY-SOURCE / *LIBRARY-DEFAULT / *NONE / <name 1..8>
        | | ,STATE = *BY-SOURCE / *FREE / *IN-HOLD
        | ,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *ANY
        | ,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, dessen Attribute geändert werden sollen.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Die Attribute des Elementes mit der höchsten bestehenden Version werden geändert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version 'X'FF' des Elementes wird verwendet.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird verwendet.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird verwendet.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden verwendet.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**CODED-CHARACTER-SET = \*ANY / \*NONE / <name 1..8 with-wild(20)>**

Zeichensatz, der dem Element zugeordnet ist.

**CODED-CHARACTER-SET = \*ANY**

Die Elemente werden unabhängig von dem zugeordneten Zeichensatz ausgewählt.

**CODED-CHARACTER-SET = \*NONE**

Die Elemente, denen kein Zeichensatz zugeordnet ist, werden ausgewählt.

**CODED-CHARACTER-SET = <name 1..8 with-wild(20)>**

Die Elemente, denen der angegebene Zeichensatz zugeordnet ist, werden ausgewählt.

**STATE = \*ANY / \*FREE / \*IN-HOLD(...)**

Zustand, der dem Element zugeordnet ist.

**STATE = \*ANY**

Es werden Elemente unabhängig von STATE ausgewählt.

**STATE = \*FREE**

Nur Elemente mit STATE=FREE werden ausgewählt.

**STATE = \*IN-HOLD(...)**

Nur Elemente mit STATE=IN-HOLD werden ausgewählt.

**HOLDER = \*ANY / <name 1..8 with-wild(20)>**

HOLDER, der dem Element zugeordnet ist.

**HOLDER = \*ANY**

Die Elemente werden, unabhängig vom HOLDER, ausgewählt.

**HOLDER = <name 1..8 with-wild(20)>**

Nur Elemente, denen ein zum Muster passender HOLDER zugeordnet ist, werden ausgewählt.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, deren Attribute von einer Änderung ausgeschlossen werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

**NEW-ATTRIBUTES = \*PARAMETERS(...)**

Angabe der Attribute, die das Zielelement erhalten soll.

**ELEMENT = \*BY-SOURCE(...) /**

**<composed-name 1..132 with-under with-wild-constr>(...)**

Neuer Name, den das Element erhalten soll.

**ELEMENT = \*BY-SOURCE(...)**

Der neue Name entspricht dem alten Namen. Das Element wird nicht umbenannt.

**VERSION = \*BY-SOURCE / \*LMS-DEFAULT / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /**

**<composed-name 1..52 with-under with-wild-constr>**

Version, die das Element erhalten soll. Das Element wird nicht umbenannt, sondern nur die Version.

**VERSION = \*BY-SOURCE**

Das neue Element erhält die gleiche Version wie das ursprüngliche Element.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion (siehe auch [Seite 59](#)).

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

**VERSION = <composed-name 1..52 with-under with-wild-constr>**

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Der neue Name des Elementes kann auch mit Wildcards eingegeben werden.

**VERSION = \*BY-SOURCE / \*LMS-DEFAULT / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT /****<composed-name 1..52 with-under with-wild-constr>**

Version, die das umzubennende Element erhalten soll.

Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / <alphanum-name 1..20 with-wild-constr>**

Typ, den das umzubennende Element erhalten soll.

**TYPE = \*BY-SOURCE**

Das umzubennende Element erhält die gleiche Typbezeichnung wie das ursprüngliche Element.

**USER-DATE = \*BY-SOURCE / \*TODAY / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*BY-SOURCE**

Das umzubennende Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**MODIFICATION-DATE = \*BY-SOURCE / \*SYSTEM-DATE**

Steuert die Aktualisierung des Änderungsdatums.

**MODIFICATION-DATE = \*BY-SOURCE**

Das Änderungsdatum wird nicht aktualisiert.

**MODIFICATION-DATE = \*SYSTEM-DATE**

Das Änderungsdatum wird aktualisiert.

**CODED-CHARACTER-SET = \*BY-SOURCE / \*LIBRARY-DEFAULT / \*NONE / <name 1..8>**

Zeichensatz, der dem Element zugeordnet werden soll.

**CODED-CHARACTER-SET = \*BY-SOURCE**

Das Element erhält den Zeichensatz des Quellelementes zugeordnet.

**CODED-CHARACTER-SET = \*LIBRARY-DEFAULT**

Das Element erhält den Zeichensatz der Bibliothek zugeordnet, die das Element enthält.

**CODED-CHARACTER-SET = \*NONE**

Dem Element wird kein Zeichensatz zugeordnet.

**CODED-CHARACTER-SET = <name 1..8>**

Das Element erhält den hier angegebenen Zeichensatz zugeordnet.

**STATE = \*BY-SOURCE / \*FREE / \*IN-HOLD**

Zustand, der dem Element zugeordnet wird.

**STATE = \*BY-SOURCE**

Der Zustand des Elementes bleibt unverändert.

**STATE = \*FREE**

Der neue Zustand des Elementes ist FREE. Nur der HOLDER und der Bibliothekseigentümer dürfen den Zustand auf FREE ändern.

**STATE = \*IN-HOLD**

Der neue Zustand des Elementes ist IN-HOLD. Die USERID desjenigen, der den Zustandswechsel von FREE auf IN-HOLD änderte, wird als HOLDER eingetragen. War der Zustand bereits IN-HOLD, wird die Anweisung abgewiesen. Nur wer Ausleihrecht hat, kann HOLDER werden.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt. Bleiben Typ, Name und Version des Elementes unverändert, so wirkt WRITE-MODE immer wie WRITE-MODE=\*ANY.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.



**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

### Notwendige Zugriffsrechte

Lese- und Schreibrecht für LIBRARY

Soll die Elementbezeichnung geändert werden, ist zusätzlich Administrationsrecht erforderlich.

Soll die Elementbezeichnung geändert werden und existiert diese Elementbezeichnung bereits, so ist für das dadurch überschriebene Element zusätzlich Schreibrecht erforderlich.

Soll der Elementzustand von FREE auf IN-HOLD geändert werden, ist Ausleihrecht erforderlich. Nur der HOLDER oder der Bibliothekseigentümer dürfen den Zustand wieder auf FREE ändern.

Andere Attribute dürfen mit Administrationsrecht oder Schreibrecht für das Element geändert werden, nicht jedoch bei STATE=\*IN-HOLD und aktivem WRITE-CONTROL.

#### *Hinweise*

- Beim Anlegen eines Elements ist die unter dem Typ geltende Konvention zu beachten.
- Bei aktivem WRITE-CONTROL ist das Ändern der Elementbezeichnung nicht erlaubt. Damit wird verhindert, dass Elemente erzeugt oder überschrieben werden, ohne dass dokumentiert wird durch wen und wann.

#### *Beispiele*

In den folgenden Beispielen sei der Elementtyp bereits voreingestellt.

- Ändern des Elementnamens von OLD in NEW:

```
//modify-element-attributes *lib(library=lib,elem=old), -  
                             new-attr=*par(elem=new)
```

- Ändern der Versionsbezeichnung des Elementes old von V1 in V2:

```
//modify-element-attributes *lib(library=lib,elem=old(version=v1)), -  
                             new-attr=*par(*by-source(version=v2))
```

## MODIFY-ELEMENT-PROTECTION

### Elementschutz ändern

Diese Anweisung ändert den Schutz für die angegebenen Elemente der zugewiesenen Bibliothek. Die Anweisung ist für alle Elementtypen zulässig.

\*UNCHANGED bedeutet, dass die jeweiligen Elementschutzattribute unverändert bleiben.

#### MODIFY-ELEMENT-PROTECTION

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

\*LIBRARY-ELEMENT(...)

**LIBRARY** = \*STD / <filename 1..54 without-vers> / \*LINK(...)

        \*LINK(...)

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

**ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)

        \*ALL(...)

**VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
                            <composed-name 1..24 with-under with-wild(52)>

**BASE** = \*STD / <composed-name 1..24 with-under with-wild>

    <composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT /  
                            <composed-name 1..24 with-under with-wild(52)>

**BASE** = \*STD / <composed-name 1..24 with-under with-wild>

**TYPE** = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>

**USER-DATE** = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)

        \*INTERVAL(...)

**FROM** = **1900-01-01** / <date 8..10 with-compl>

**TO** = \*TODAY / <date 8..10 with-compl>

(Teil 1 von 4)

```

,CREATION-DATE = ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,PROTECTION = ANY / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | READ = ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |   *BY-GUARD(...)
    |     |
    |     | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |     |
    |     | *PARAMETERS(...)
    |     |   |
    |     |   | USER = ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |     |   | ,PASSWORD = ANY / *YES / *NO
    |   ,WRITE = ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |     *BY-GUARD(...)
    |       |
    |       | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |       |
    |       | *PARAMETERS(...)
    |       |   |
    |       |   | USER = ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |       |   | ,PASSWORD = ANY / *YES / *NO
    |   ,EXEC = ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |     *BY-GUARD(...)
    |       |
    |       | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |       |
    |       | *PARAMETERS(...)
    |       |   |
    |       |   | USER = ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |       |   | ,PASSWORD = ANY / *YES / *NO

```

(Teil 2 von 4)

```

, HOLD = *ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
  *BY-GUARD(...)
    |   GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
  *PARAMETERS(...)
    |   USER = *ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |   , PASSWORD = *ANY / *YES / *NO
, EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |   ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    |   *ANY(...)
    |     |   VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |     |     <composed-name 1..24 with-under with-wild(52)>
    |     |   , BASE = *STD / <composed-name 1..24 with-under with-wild>
    |     |   <composed-name 1..64 with-under with-wild(132)>(…)
    |     |     |   VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |     |     |     <composed-name 1..24 with-under with-wild(52)>
    |     |     |   , BASE = *STD / <composed-name 1..24 with-under with-wild>
    |     |   , TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
    |     |   , USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |     |     *INTERVAL(...)
    |     |       |   FROM = 1900-01-01 / <date 8..10 with-compl>
    |     |       |   , TO = *TODAY / <date 8..10 with-compl>
    |     |     , CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |     |       *INTERVAL(...)
    |     |         |   FROM = 1900-01-01 / <date 8..10 with-compl>
    |     |         |   , TO = *TODAY / <date 8..10 with-compl>
    |     |     , MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |     |       *INTERVAL(...)
    |     |         |   FROM = 1900-01-01 / <date 8..10 with-compl>
    |     |         |   , TO = *TODAY / <date 8..10 with-compl>

```

(Teil 3 von 4)

```

,NEW-PROTECTION = *UNCHANGED / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    READ = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,WRITE = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,EXEC = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,HOLD = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
  ,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES

```

(Teil 4 von 4)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, dessen Schutz-Attribute geändert werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Die Schutz-Attribute des Elementes mit der höchsten bestehenden Version werden geändert.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' des Elementes wird verwendet.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*. Näheres zur Angabe der Basis siehe [Seite 54](#).

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird verwendet.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird verwendet.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden verwendet.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**PROTECTION = \*ANY / \*NONE / \*PARAMETERS(...)**

Elementschutz für die ausgewählten Elemente.

**PROTECTION = \*ANY**

Die Elemente haben beliebigen Elementenschutz.

**PROTECTION = \*NONE**

Die Elemente haben keinen zusätzlichen Elementenschutz.

**PROTECTION = \*PARAMETERS(...)**

Angabe, mit welchem Schutz die auszuwählenden Elemente versehen sind.

**READ = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Schutzeinstellung bezüglich Lesen, die dem Element zugeordnet ist.

**READ = \*ANY**

Es werden Elemente unabhängig vom Leseschutz ausgewählt.

**READ = \*NONE**

Es werden nur Elemente ausgewählt, für die kein Leseschutz eingestellt ist.

**READ = \*BY-GUARD(...)**

Nur Elemente, für die Leseschutz durch GUARD eingestellt ist, werden ausgewählt.

**GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>**

Nur Elemente, für die Leseschutz durch einen zum Muster passenden GUARD-NAMEN eingestellt ist, werden ausgewählt.



**READ = \*PARAMETERS(...)**

Nur Elemente, für die Leseschutz durch BACL und/oder Kennwort eingestellt ist, werden ausgewählt.

**USER = \*ANY / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Leseberechtigten, der dem Element zugeordnet ist.

**USER = \*ANY**

Es werden Elemente unabhängig vom Kreis der Leseberechtigten ausgewählt.

**USER = \*NONE**

Nur Elemente, für die an niemanden Leserecht vergeben ist, werden ausgewählt.

**USER = \*OWNER**

Nur Elemente, für die der Eigentümer der Bibliotheksdatei Leserecht hat, werden ausgewählt.

**USER = \*GROUP**

Nur Elemente, für die die Gruppe des Eigentümers der Bibliotheksdatei Leserecht hat, werden ausgewählt.

**USER = \*OTHERS**

Nur Elemente, für die Leserecht für OTHERS vergeben ist, werden ausgewählt.

**PASSWORD = \*ANY / \*YES / \*NO**

Kennwort bezüglich Lesen, das dem Element zugeordnet ist.

**PASSWORD = \*ANY**

Es werden Elemente unabhängig vom Kennwort ausgewählt.

**PASSWORD = \*YES**

Nur Elemente, die durch ein Lesekennwort geschützt sind, werden ausgewählt.

**PASSWORD = \*NO**

Nur Elemente, die durch kein Lesekennwort geschützt sind, werden ausgewählt.

**WRITE = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Schreibrecht. Nur Elemente werden ausgewählt, für die dieses Recht in der angegebenen Art vergeben ist.

Beschreibung der Operanden analog zu READ.

**EXEC = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ausführungsrecht. Nur Elemente werden ausgewählt, für die dieses Recht in der angegebenen Art vergeben ist.

Beschreibung der Operanden analog zu READ.

**HOLD = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ausleihrecht. Nur Elemente werden ausgewählt, für die dieses Recht in der angegebenen Art vergeben ist.

Beschreibung der Operanden analog zu READ.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, dessen Schutz-Attribute von einer Änderung ausgeschlossen werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

**NEW-PROTECTION = \*UNCHANGED / \*NONE / \*PARAMETERS(...)**

Neuer Elementschutz für die ausgewählten Elemente.

**NEW-PROTECTION = \*NONE**

Das Element bekommt keinen neuen oder zusätzlichen Schutz. Es ist nur durch den Schutz der Bibliotheksdatei geschützt.

**NEW-PROTECTION = \*PARAMETERS(...)**

Angabe, welcher Schutz jetzt für das Element gelten soll.

**READ = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Neue Vergabe des Leseschutzes für das Element.

**READ = \*NONE**

Es wird keine weitere Zugriffseinschränkung vorgesehen.

**READ = \*BY-GUARD(...)**

Angabe des GUARDS für das Leserecht.

**GUARD-NAME = <filename 1..18 without-cat-gen-vers>**

Name des GUARDS.

**READ = \*PARAMETERS(...)**

Änderung der Benutzerkreise für das Leserecht.

**USER = \*UNCHANGED / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Leseberechtigten wird explizit aufgezählt.

**USER = \*NONE**

Keiner darf in der angegebenen Art zugreifen.

**USER = \*ALL**

Alle dürfen in der angegebenen Art zugreifen (komplette Aufzählung).

**USER = \*OWNER**

Der Eigentümer der Bibliotheksdatei darf zugreifen.

**USER = \*GROUP**

Die Angehörigen der Gruppe des Eigentümers der Bibliotheksdatei dürfen zugreifen.

**USER = \*OTHERS**

Alle anderen dürfen zugreifen.

**PASSWORD = \*UNCHANGED / \*SECRET / \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>**

Der Kreis der Berechtigten wird weiter eingeschränkt. Zusätzlich zum notwendigen Zugriffsrecht wird das richtige Kennwort verlangt. Die Angabe 0 bzw. X'00000000' führt nicht zu einer Änderung des letzten Wertes.

Die Angabe von \*SECRET bzw. ^ ermöglicht die verdeckte Eingabe des gewünschten Kennwortes. Wird der „geheime“ Wert als c-string eingegeben, muss er in Hochkommata stehen. Wird er als x-string eingegeben, muss er ebenfalls in Hochkommata angegeben und ein X vorangestellt werden.

**WRITE = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Neue Vergabe des Schreibrechts für das Element.

Beschreibung der Operanden analog zu READ ([Seite 330](#)).

**EXEC = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Neue Vergabe des Ausführungsrechts für das Element.

Beschreibung der Operanden analog zu READ ([Seite 330](#)).

**HOLD = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Neue Vergabe des Ausleihrechts für das Element.

Beschreibung der Operanden analog zu READ ([Seite 330](#)).

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS, wobei der dort ggf. eingestellte Wert \*ERROR wie \*NO wirkt. Ebenso wirkt der beim /SEND-MSG-Kommando ggf. angegebene Wert für DIALOG-CONTROL= \*ERROR bei MODIFY-ELEMENT-PROTECTION wie \*NO.

*Hinweis*

Der bei MODIFY-LMS-DEFAULTS ggf. eingestellte Wert \*ERROR wirkt wie \*NO.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Lese- und Schreibrecht für LIBRARY

Schutzattribute können nur vom Eigentümer der Bibliotheksdatei verändert werden.

*Beispiele*

- Zusätzlich zum Leserecht erhält die Gruppe auch das Schreibrecht für alle Elemente, auf die die Gruppe bisher lesend zugreifen darf.

```
//modify-element-protection -
//          *lib(elem=*,type=*,protection=(read=(user=*group))),-
//          new-protection=(write=(user=*group))
```

Bei der Angabe von new-protection muss das Leserecht nicht nochmal bestimmt werden. Es gilt die bisherige Einstellung (UNCHANGED) für das Leserecht.

- Die Eingabe des Kennwortes soll dabei nicht sichtbar sein.

```
//modify-element-protection *lib(elem=test,type=s), -
//          new-protection=(exec=(user=*group,password=*secret))
%ENTER SECRET OPERAND (NEW-PROTECTION=:EXEC=:PASSWORD):
```

## MODIFY-LIBRARY-ATTRIBUTES

### Attribute einer Bibliothek ändern

Diese Anweisung ändert die Attribute der angegebenen Bibliothek. Dies sind

- Schutzattribute
- bevorzugte Speicherform und
- Führen eines Zugriffsdatums

\*UNCHANGED bedeutet, dass die jeweiligen Bibliotheksattribute unverändert bleiben.

#### MODIFY-LIBRARY-ATTRIBUTES

```

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
,STORAGE-FORM = *UNCHANGED / *NONE / *STD / *FULL / *DELTA
,WRITE-CONTROL = *UNCHANGED / *NONE / *DEACTIVATE / *ACTIVATE
,ACCESS-DATE = *UNCHANGED / *NONE / *KEEP
,ADMINISTRATION = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
  *BY-GUARD(...)
    | GUARD-NAME = <filename 1..18 without-cat-gen-vers>
  *PARAMETERS(...)
    | USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    | ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
      <x-string 1..8> / <integer -2147483648..2147483647>
,INIT-ELEM-PROTECTION = *UNCHANGED / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    | READ = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        | GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        | USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        | ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
          <x-string 1..8> / <integer -2147483648..2147483647>

```

(Teil 1 von 2)

```

,WRITE = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
    *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
,EXEC = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
    *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
,HOLD = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
    *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>

```

(Teil 2 von 2)

**LIBRARY = \*STD / <full-filename 1..54 without-vers> / \*LINK(...)**  
 Angabe der Bibliothek, deren Schutzattribute geändert werden sollen.

**LIBRARY = \*STD**  
 Die durch OPEN-LIBRARY eröffnete Bibliothek.

**LIBRARY = <filename 1..54 without-vers>**  
 Name der Bibliothek, deren Schutzattribute geändert werden sollen.

**LIBRARY = \*LINK(...)**  
 Die über den Dateikettungsname zugewiesene Bibliothek.

**LINK-NAME = <structured-name 1..8>**  
 Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem  
 /ADD-FILE-LINK-Kommando vereinbart wurde.

**STORAGE-FORM = \*UNCHANGED / \*NONE / \*STD / \*FULL / \*DELTA**

Zulässige Speicherform für Elemente der Bibliothek, sofern durch MODIFY-TYPE-ATTRIBUTES für den Elementtyp nichts anderes verlangt wird. Außerdem müssen alle Elemente eines Typs und Namens die gleiche Speicherform haben.

**STORAGE-FORM = \*NONE**

Wirkung wie \*STD.

**STORAGE-FORM = \*STD**

Sowohl Voll- als auch Deltaspeicherung sind zulässig.

**STORAGE-FORM = \*FULL**

Nur Vollspeicherung ist zulässig.

**STORAGE-FORM = \*DELTA**

Nur Deltaspeicherung ist zulässig.

**WRITE-CONTROL = \*UNCHANGED / \*NONE / \*DEACTIVATE / \*ACTIVATE**

Attribut zur Steuerung zusätzlicher Prüfungen.

**WRITE-CONTROL = \*NONE**

Es gibt keine zusätzlichen Prüfungen beim Erzeugen oder Überschreiben von Versionen.

**WRITE-CONTROL = \*DEACTIVATE**

Es gibt keine zusätzlichen Prüfungen beim Erzeugen oder Überschreiben von Versionen.

**WRITE-CONTROL = \*ACTIVATE**

Eine Version darf nur geschrieben werden, wenn

- die USERID desjenigen, der schreiben will, bei der explizit oder implizit immer gegebenen Basisversion als HOLDER eingetragen ist, und
- entweder eine neue Version erzeugt oder die Basisversion überschrieben wird.

Bei der ersten Version eines Namens gibt es noch keine Basis; sie kann nur von ADMIN-Berechtigten erzeugt werden. Beim Erzeugen oder Überschreiben von Versionen wird automatisch ein Satz der Satzart 2 zugefügt, der HOLDER=Schreiber, DATE und TIME des Vorgangs dokumentiert. Ferner werden die Attribute STATE, HOLDER und alle Rechte auf die neue Version übernommen, falls durch die aktuelle Anweisung nicht das Eintragen anderer Werte verlangt wird.

**ACCESS-DATE = \*UNCHANGED / \*NONE / \*KEEP**

Attribut zum Führen des Zugriffsdatums.

**ACCESS-DATE = \*NONE**

In der Bibliothek wird kein Zugriffsdatum für Elemente geführt.

**ACCESS-DATE = \*KEEP**

In der Bibliothek wird das Zugriffsdatum für Elemente geführt. Erst ab diesem Zeitpunkt kann ein Element Zugriffsdatum und -zeit besitzen.

**ADMINISTRATION = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Administrationsrecht. Der Kreis der Administrationsberechtigten wird für die Bibliothek explizit festgelegt. Nur diese dürfen Elemente anlegen, löschen und umbenennen.

**ADMINISTRATION = \*NONE**

Es wird kein Administrationsrecht vergeben.

**ADMINISTRATION = \*BY-GUARD(...)**

Das Verwaltungsrecht der angegebenen Bibliothek wird über einen GUARD geregelt.

**GUARD-NAME = <filename 1..18 without-cat-gen-vers>**

Name des GUARDS.

**ADMINISTRATION = \*PARAMETERS(...)**

Angabe der Benutzerkreise und evtl. eines Kennwortes für die Verwaltung der angegebenen Bibliothek.

**USER = \*UNCHANGED / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Verwaltungsberechtigten wird explizit aufgezählt.

**USER = \*NONE**

Keiner darf verwalten.

**USER = \*ALL**

Alle dürfen verwalten (komplette Aufzählung).

**USER = \*OWNER**

Nur der Eigentümer der Bibliotheksdatei darf sie verwalten.

**USER = \*GROUP**

Die Angehörigen der Gruppe des Eigentümers der Bibliotheksdatei dürfen verwalten.

**USER = \*OTHERS**

Alle anderen dürfen verwalten.

**PASSWORD = \*UNCHANGED / \*SECRET / \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>**

Der Kreis der Berechtigten wird weiter eingeschränkt. Zusätzlich zum notwendigen Zugriffsrecht wird das richtige Kennwort verlangt. Die Angabe 0 bzw. X'00000000' führt nicht zu einer Änderung des letzten Wertes.

Die Angabe von \*SECRET bzw. ^ ermöglicht die verdeckte Eingabe des gewünschten Kennwortes. Wird der „geheime“ Wert als c-string eingegeben, muss er in Hochkommata stehen. Wird er als x-string eingegeben, muss er ebenfalls in Hochkommata angegeben und ein X vorangestellt werden.



**INIT-ELEM-PROTECTION = \*UNCHANGED / \*NONE / \*PARAMETERS(...)**

**INIT-ELEM-PROTECTION = \*NONE**

Es wird kein initialer Schutz für die in der Bibliothek enthaltenen Elemente vereinbart.

**INIT-ELEM-PROTECTION = \*PARAMETERS(...)**

Angabe der Schutzrechte, die die neu erstellten Elemente erhalten sollen.

**READ = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Leserecht wird explizit angegeben.

**READ = \*NONE**

Es ist keine weitere Zugriffseinschränkung vorgesehen.

**READ = \*BY-GUARD(...)**

Angabe des GUARDs für das Leserecht.

**GUARD-NAME = <filename 1..18 without-cat-gen-vers>**

Name des GUARDs.

**READ = \*PARAMETERS(...)**

Angabe der Benutzerkreise für das Leserecht.

**USER = \*UNCHANGED / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Leseberechtigten wird explizit aufgezählt.

**USER = \*NONE**

Keiner darf in der angegebenen Art zugreifen.

**USER = \*ALL**

Alle dürfen in der angegebenen Art zugreifen (komplette Aufzählung).

**USER = \*OWNER**

Der Eigentümer der Bibliotheksdatei darf zugreifen.

**USER = \*GROUP**

Die Angehörigen der Gruppe des Eigentümers der Bibliotheksdatei dürfen zugreifen.

**USER = \*OTHERS**

Alle anderen dürfen zugreifen.

**PASSWORD = \*UNCHANGED / \*SECRET / \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>**

Der Kreis der Berechtigten wird weiter eingeschränkt. Zusätzlich zum notwendigen Zugriffsrecht wird das richtige Kennwort verlangt.

Die Angabe 0 bzw. X'00000000' führt nicht zu einer Änderung des letzten Wertes.

Die Angabe von \*SECRET bzw. ^ ermöglicht die verdeckte Eingabe des gewünschten Kennwortes. Wird der „geheime“ Wert als c-string eingegeben, muss er in Hochkommata stehen. Wird er als x-string eingegeben, muss er ebenfalls in Hochkommata angegeben und ein X vorangestellt werden.

**WRITE = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Schreibrecht wird explizit festgelegt.

Beschreibung der Operanden analog zu READ.

**EXEC = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Ausführungsrecht wird explizit festgelegt.

Beschreibung der Operanden analog zu READ.

**HOLD = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Neue Vergabe des Ausleihrechts für das Element.

Beschreibung der Operanden analog zu READ.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt

### Notwendige Zugriffsrechte

Lese- und Schreibrecht für LIBRARY

Nur der Eigentümer der Bibliotheksdatei kann Attribute der Bibliothek ändern.

## MODIFY-LMS-DEFAULTS

### Ändern der globalen Default-Werte

Mit der Anweisung MODIFY-LMS-DEFAULTS können die Defaultwerte geändert werden. Wird in einer LMS-Anweisung lokal ein expliziter Wert benutzt, hat dieser Vorrang vor dem Defaultwert.

Der Bezug in den LMS-Anweisungen zu den hier eingestellten Werten ist die Angabe von \*LMS-DEFAULT.

Zu Beginn des LMS-Laufs gelten die unmittelbar auf \*UNCHANGED folgenden Werte. Wird einer dieser Werte durch die MODIFY-LMS-DEFAULTS-Anweisung verändert, wird diese neue Einstellung zur aktuellen Einstellung. Diese bleibt solange für den LMS-Lauf gültig (\*UNCHANGED), bis eine neue Anweisung MODIFY-LMS-DEFAULTS für diesen Wert oder RESET-LMS-DEFAULTS gegeben wird.

#### MODIFY-LMS-DEFAULTS

**ELEMENT-ATTRIBUTES = \*UNCHANGED / \*PARAMETERS(...)**

**\*PARAMETERS(...)**

**TYPE = \*UNCHANGED / \*NONE / <alphanum-name 1..8>**

**,ELEMENT-VERSION = \*UNCHANGED / \*ALL / \*HIGHEST-EXISTING**

**,TO-ELEMENT-VERSION = \*UNCHANGED / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT**

**,STORAGE-FORM = \*UNCHANGED / \*STD / \*FULL / \*DELTA**

**,SOURCE-ATTRIBUTES = \*UNCHANGED / \*STD / \*IGNORE / \*KEEP**

**,FILE-ATTRIBUTES = \*UNCHANGED / \*PARAMETERS(...)**

**\*PARAMETERS(...)**

**ACCESS-METHOD = \*UNCHANGED / \*ISAM / \*SAM**

**,DESTROY-DATA = \*UNCHANGED / \*NO / \*YES / \*BY-SOURCE**

**,WRITE-MODE = \*UNCHANGED / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

**,DIALOG-CONTROL = \*UNCHANGED / \*NO / \*YES / \*ERROR**

(Teil 1 von 5)

```

,INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM /
                                     *SUMMARY / *DELTA-STRUCTURE / *PARAMETERS(...)

*PARAMETERS(...)
  |
  | GENERAL = *UNCHANGED / *NO / *YES
  |
  | ,HISTORY = *UNCHANGED / *NO / *YES
  |
  | ,SECURITY = *UNCHANGED / *NO / *YES
  |
,LAYOUT = *UNCHANGED / *VARIABLE / *FIXED
,SORT = *UNCHANGED / *BY-NAME / *BY-VERSION / *BY-USER-DATE / *BY-CREATION-DATE /
      *BY-MODIFICATION-DATE / *BY-ACCESS-DATE / *BY-ELEMENT-SIZE / *BY-SECONDARY-NAME
,OUTPUT-FORM = *UNCHANGED / *STD / *CHARACTER / *HEXADECIMAL / *DUMP
,DELETE-SOURCE = *UNCHANGED / *NO / *YES
,PROTECTION = *UNCHANGED / *STD / *BY-SOURCE
,MAX-ERROR-WEIGHT = *UNCHANGED / *SERIOUS / *SIGNIFICANT / *RECOVERABLE
,EDT-MODE = *UNCHANGED / *COMPATIBLE / *UNICODE
,NEXT-ATTEMPT = *UNCHANGED / *NO / *YES(...)

*YES(...)
  |
  | NUMBER-OF-ATTEMPTS = *UNCHANGED / <integer 1..2147483647>
  |
  | ,PERIOD = *UNCHANGED / <integer 1..21599>
  |
,COMPARE-PARAMETERS = *UNCHANGED / *PARAMETERS(...)

*PARAMETERS(...)
  |
  | RECORD-PART = *UNCHANGED / *ALL / *PART(...)
  |
  | *PART(...)
  |   |
  |   | START = *UNCHANGED / <integer 1..32764>
  |   |
  |   | ,LENGTH = *UNCHANGED / *REST / <integer 1..32764>
  |   |
  |   | ,SPACES = *UNCHANGED / *STD / *IGNORED / *RELEVANT
  |   |
  |   | ,INFORMATION = *UNCHANGED / *MEDIUM / *MINIMUM / *MAXIMUM / *SUMMARY / *STATISTICS
  |   |                 / *NONE
  |   |
  |   | ,LAYOUT = *UNCHANGED / *COMPATIBLE / *COMPRESSED
  |   |
  |   | ,JOIN-ELEMENT-SETS = *UNCHANGED / *NO / *YES
  |

```

(Teil 2 von 5)

```

,TEXT-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *FILE-ATTRIBUTES / *PARAMETERS(...)
  *PARAMETERS(...)
    INFORMATION = *UNCHANGED / *ALL / list-poss(2): *TEXT / *COMMENT
  ,RECORD-RANGE = *UNCHANGED / *ALL / *RANGE(...)
    *RANGE(...)
      FROM = *UNCHANGED / <integer 1..2147483647>
      ,TO = *UNCHANGED / *LAST / <integer 1..2147483647>
  ,RECORD-PART = *UNCHANGED / *ALL / *PART(...)
    *PART(...)
      START = *UNCHANGED / <integer 1..262144>
      ,LENGTH = *UNCHANGED / *REST / <integer 1..262144>
  ,RECORD-NUMBER = *UNCHANGED / *BY-OUTPUT / *YES / *NO
,MODULE-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *PARAMETERS(...)
  *PARAMETERS(...)
    INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) /
      list-poss(9): *ESD / *ISD / *LSD / *RLD / *REP / *INCLUDE / *DSDD / *REF / *END
    *TXT(...)
      CSECT-NAME = *UNCHANGED / *ALL / <c-string 1..32 with-low> / <text 1..32>
    ,ADDRESS = *UNCHANGED (...) / <x-string 1..8>(…)
      *UNCHANGED(...)
        | BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
      <x-string 1..8>(…)
        | BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
    ,LENGTH = *UNCHANGED / *REST / <integer 1..2147483647> / <x-string 1..8>
    *TXTP(...)
      MODIFICATION-ID = *UNCHANGED / *ALL / <c-string 1..8 with-low> / *RANGE(...)
      *RANGE(...)
        | FROM = *UNCHANGED / *LOWEST / <c-string 1..8 with-low>
        ,TO = *UNCHANGED / *HIGHEST / <c-string 1..8 with-low>

```

(Teil 3 von 5)

```

,PHASE-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | SEGMENT = *UNCHANGED / *ALL / *ROOT / <name 1..8>
    | ,INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) /
    | list-poss(4): *ESD / *ISD / *LSD / *RLD
    | *TXT(...)
    | | ADDRESS = *UNCHANGED (...) / <x-string 1..8>(…)
    | | *UNCHANGED(…)
    | | | BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
    | | | <x-string 1..8>(…)
    | | | BASE-ADDRESS = *UNCHANGED / <x-string 1..8>
    | | ,LENGTH = *UNCHANGED / *REST / <integer 1..2147483647> / <x-string 1..8>
    | *TXTP(...)
    | | MODIFICATION-ID = *UNCHANGED / *ALL / <c-string 1..8 with-low> / *RANGE(...)
    | | *RANGE(...)
    | | | FROM = *UNCHANGED / *LOWEST / <c-string 1..8 with-low>
    | | | ,TO = *UNCHANGED / *HIGHEST / <c-string 1..8 with-low>
    | ,LLM-INFORMATION = *UNCHANGED / *ALL / *STATISTICS / *PARAMETERS(...)
    | *PARAMETERS(...)
    | | LLM-PART = *UNCHANGED / *ALL / *SLICE(...) / *SUB-LLM(...)
    | | *SLICE(...)
    | | | NAME = *UNCHANGED / <structured-name 1..32>
    | | *SUB-LLM(...)
    | | | PATH-NAME = *UNCHANGED / <c-string 1..255 with-low> / <text 1..255>
    | | ,INFORMATION = *UNCHANGED / *ALL / *TXT(...) / *TXTP(...) / *LOGICAL(...) / *PHYSICAL / *REF /
    | | list-poss(4): *RELOCATION / *ESVD / *ESVR / *LRLD
    | | *TXT(...)
    | | | CSECT-NAME = *UNCHANGED / *ALL / <c-string 1..32 with-low> / <text 1..32>

```

(Teil 4 von 5)

<pre> ,ADDRESS = *<u>UNCHANGED</u> (...) / &lt;x-string 1..8&gt;(…)   *<u>UNCHANGED</u>(…)         <b>BASE-ADDRESS</b> = *<u>UNCHANGED</u> / &lt;x-string 1..8&gt;       &lt;x-string 1..8&gt;(…)             <b>BASE-ADDRESS</b> = *<u>UNCHANGED</u> / &lt;x-string 1..8&gt;           ,LENGTH = *<u>UNCHANGED</u> / *<u>REST</u> / &lt;integer 1..2147483647&gt; / &lt;x-string 1..8&gt;         *TXTP(...)               <b>CSECT-NAME</b> = *<u>UNCHANGED</u> / *<u>ALL</u> / &lt;c-string 1..32 with-low&gt; / &lt;text 1..32&gt;             ,MODIFICATION-ID = *<u>UNCHANGED</u> / *<u>ALL</u> / &lt;c-string 1..12 with-low&gt; / *<u>RANGE</u>(…)               *<u>RANGE</u>(…)                     <b>FROM</b> = *<u>UNCHANGED</u> / *<u>LOWEST</u> / &lt;c-string 1..12 with-low&gt;                   ,<b>TO</b> = *<u>UNCHANGED</u> / *<u>HIGHEST</u> / &lt;c-string 1..12 with-low&gt;                 *<u>LOGICAL</u>(…)                       <b>LEVEL</b> = *<u>UNCHANGED</u> / *<u>ALL</u> / *<u>NEXT</u> </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Teil 5 von 5)

**ELEMENT-ATTRIBUTES = \*UNCHANGED / \*PARAMETERS(...)**

Bestimmt den Elementtyp, die Elementversion und die Speicherform sowie die Dateieigenschaften.

**TYPE = \*UNCHANGED / \*NONE / <alphanum-name 1..8>**

Festlegung des Elementtyps.

**TYPE = \*NONE**

Es wird kein globaler Elementtyp festgelegt, d.h. die Typangaben müssen anweisungslokal erfolgen, sofern die in der Anweisung betroffene Bibliothek keinen bibliotheksspezifischen Defaulttyp besitzt.

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

Der hier angegebene Name wird in den Anweisungen als Typ verwendet, sofern die in der Anweisung betroffene Bibliothek keinen bibliotheksspezifischen Defaulttyp besitzt. Sonst ersetzt dieser bibliotheksspezifische Defaulttyp die anweisungslokale Typangabe \*LMS-DEFAULT.

**ELEMENT-VERSION = \*UNCHANGED / \*ALL / \*HIGHEST-EXISTING**

Festlegung der Elementversion für SHOW-ELEMENT-ATTRIBUTES.

**ELEMENT-VERSION = \*ALL**

Es werden alle Versionen zu einem Element ausgegeben.

**ELEMENT-VERSION = \*HIGHEST-EXISTING**

Es wird nur die höchste, zu BASE passende Version eines Elementes ausgegeben.

**TO-ELEMENT-VERSION = \*UNCHANGED / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT**

Festlegung der Version des Zielelementes.

**TO-ELEMENT-VERSION = \*BY-SOURCE**

Das Zielelement erhält die gleiche Version wie das Quellelement.

**TO-ELEMENT-VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**TO-ELEMENT-VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version, ansonsten eine Defaultversion erzeugt.

**STORAGE-FORM = \*UNCHANGED / \*STD / \*FULL / \*DELTA**

Speicherform für das anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*STD**

Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**

Das Element wird vollgespeichert angelegt (Fehler, falls nicht zulässig).

**STORAGE-FORM = \*DELTA**

Das Element wird delta-gespeichert angelegt (Fehler, falls nicht zulässig). Die Angabe ist zulässig für die Elementtypen: S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**SOURCE-ATTRIBUTES = \*UNCHANGED / \*STD / \*IGNORE / \*KEEP**

Wirkt nur bei der Anweisung ADD-ELEMENT ausgewertet.

**SOURCE-ATTRIBUTES = \*STD**

Es werden keine Datei-Attribute und auch kein ISAM-Schlüssel gespeichert. Für ISAM-Dateien erfolgt eine Warnung, dass die ISAM-Schlüssel nicht mit aufgenommen wurden

**SOURCE-ATTRIBUTES = \*IGNORE**

Wie **SOURCE-ATTRIBUTES = \*STD**, jedoch ohne Warnung.



**SOURCE-ATTRIBUTES = \*KEEP**

Folgende Dateiattribute werden unverändert im neu aufzunehmenden Element gespeichert: ACCESS-METHOD, RECORD-FORMAT, RECORD-SIZE, BUFFER-LENGTH, PERFORMANCE, USAGE, ACCESS und USER-ACCESS.

Ist ACCESS-METHOD=ISAM, so werden PADDING-FACTOR, LOGICAL-FLAG-LENGTH, VALUE-FLAG-LENGTH, PROPAGATE-VALUE-FLAG und auch die ISAM-Schlüssel sowie Informationen über ISAM-Sekundärschlüssel zusätzlich gespeichert.

**FILE-ATTRIBUTES = \*UNCHANGED / \*PARAMETERS(...)**

Dateiattribute, die bei der Erzeugung der Datei festgelegt werden.

**ACCESS-METHOD = \*UNCHANGED / \*ISAM / \*SAM**

Festlegung der Dateizugriffsmethode.

**ACCESS-METHOD = \*ISAM**

ISAM-Datei erzeugen.

**ACCESS-METHOD = \*SAM**

SAM-Datei erzeugen.

**DESTROY-DATA = \*UNCHANGED / \*NO / \*YES / \*BY-SOURCE**

Bestimmt, ob die Daten physikalisch gelöscht, d.h. mit X'00' überschrieben werden.

**DESTROY-DATA = \*NO**

Ein Element einer Bibliothek wird nur physikalisch gelöscht, wenn im Element ein Kennzeichen für physikalisches Löschen vorhanden ist oder die CLASS2-Option DESTLEV es verlangt.

**DESTROY-DATA = \*YES**

Nach dem logischen Löschen werden die Daten, falls vorhanden, physikalisch gelöscht.

**DESTROY-DATA = \*BY-SOURCE**

Das Kennzeichen zum Überschreiben der Daten wird dem Quellelement bzw. der Quelldatei entnommen und dem Zielelement bzw. der Zieldatei zugeordnet. Fehlt die Quelle, dann wirkt \*BY-SOURCE wie DESTROY-DATA=\*NO.

**WRITE-MODE = \*UNCHANGED / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

Ist das zu speichernde Element ein Delta-Element, ist darauf zu achten, dass das Element ein Blatt des Delta-Baumes sein muss. Nur Blätter eines Delta-Baumes dürfen überschrieben werden.

Ist der ausgewählte Wert für WRITE-MODE anweisungslokal nicht möglich, gilt die Einstellung WRITE-MODE= \*CREATE.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*EXTEND**

Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**DIALOG-CONTROL = \*UNCHANGED / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht. (Dieser Operand wirkt nicht im Prozedur- oder Stapelbetrieb.)

**DIALOG-CONTROL = \*NO**

Alle Elemente werden ohne Nachfrage bearbeitet.

**DIALOG-CONTROL = \*YES**

LMS fragt bei jedem Element nach, wie es sich verhalten soll, z.B. ob das Element bearbeitet, übersprungen oder die Anweisung abgebrochen werden soll.

**DIALOG-CONTROL = \*ERROR**

Tritt während der Elementbearbeitung ein behebbarer Fehler auf, z.B. Überschreiben eines Elementes, wird der Anwender gefragt, wie LMS sich verhalten soll.

Der Benutzer hat ggf. folgende Eingriffsmöglichkeiten:

YES	Das Element soll bearbeitet werden.
NO	Das Element soll nicht bearbeitet werden.
ALL	Die Anweisung soll ohne Dialog beendet werden.
TERMINATE	Die Anweisung soll abgebrochen werden.

Nach Drücken der Taste K2 und Eingabe von „/SEND-MSG“ hat der Anwender die Möglichkeit, den Wert des Operanden DIALOG-CONTROL zu verändern. Befindet sich LMS mitten in der Elementverarbeitung, so kann der Anwender mit dem folgenden /SEND-MSG-Kommando die Weiterverarbeitung wie folgt beeinflussen:

```
/SEND-MSG PROG, '[N-I / N-E / C][,DIALOG-CONTROL= NO / YES / ERROR]'
```

```
SEND-MSG PROG, NEXT-INPUT (N-I)
```

Die aktuelle Anweisung wird abgebrochen; LMS liest eine weitere Anweisung ein, sobald es wieder aktiv ist. Ein unbekannter oder fehlender Text wirkt wie NEXT-INPUT.

```
SEND-MSG PROG, NEXT-ELEMENT (N-E)
```

Die Bearbeitung des aktuellen Elements in der laufenden Anweisung wird abgebrochen; LMS setzt die Verarbeitung mit dem nächsten Element (sofern vorhanden) fort. Ist kein nächstes Element vorhanden, wirkt NEXT-ELEMENT wie NEXT-INPUT.

SEND-MSG PROG, CONTINUE (C)

Die LMS-Verarbeitung wird regulär fortgesetzt.

C,NO	Das Element soll bearbeitet, aber der Dialog nicht eingeschaltet werden.
N-E/N-I,NO	Die Elementverarbeitung bzw. die Anweisung soll abgebrochen, aber der Dialog nicht eingeschaltet werden.
C,YES/ERROR	Das Element soll bearbeitet und danach der Dialog eingeschaltet werden.
N-E/N-I, YES/ERROR	Die Elementverarbeitung bzw. die Anweisung soll abgebrochen und danach der Dialog eingeschaltet werden.

### *Hinweis*

Der mit /SEND-MSG eingestellte Wert für DIALOG-CONTROL gilt nur für die aktuelle Anweisung.

**INFORMATION = \*UNCHANGED / \*MEDIUM / \*MINIMUM/ \*MAXIMUM / \*SUMMARY / \*DELTA-STRUCTURE / \*PARAMETERS(...)**

Dieser Parameter bestimmt den Umfang des auszugebenden Inhaltsverzeichnisses.

**INFORMATION = \*MEDIUM**

Es wird Typ, Name, Version, Variantennummer und abhängig vom Operanden SORT das Datum bzw. die Elementgröße der ausgewählten Elemente ausgegeben.

**INFORMATION = \*MINIMUM**

Es wird nur Typ, Name und Version der ausgewählten Elemente ausgegeben.

**INFORMATION = \*MAXIMUM**

Es wird die gesamte Information der ausgewählten Elemente ausgegeben.

**INFORMATION = \*SUMMARY**

Es wird nur die Anzahl der ausgewählten Elemente pro Typ ausgegeben.

**INFORMATION = \*DELTA-STRUCTURE**

Die Beziehung „Vorgänger - Nachfolger“ bei Delta-Elementen wird ausgegeben. Zusätzlich zur Elementbezeichnung werden die interne Delta-Nummer (DELTA#, sie spiegelt die chronologische Reihenfolge wider) und die entsprechende Nummer der Basis (BASE#) ausgegeben. Diese internen Delta-Nummern sind innerhalb eines Baumes eindeutig und beschreiben die Verkettung der Elemente im Baum (unabhängig von der externen benutzereigenen Versionsbezeichnung). Die Ausgabe eines Baumes ist immer nach DELTA# sortiert, d.h. der Operand SORT (siehe unten) wirkt nicht innerhalb eines Baumes. Verschiedene Bäume sind durch einen durchgehenden Strich voneinander getrennt. Bei Voll-Elementen sind die Ausgabefelder DELTA# und BASE# leer.

**INFORMATION = \*PARAMETERS(...)**

Zusätzliche Ausgaben.

**GENERAL = \*UNCHANGED / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein allgemeiner Informationsblock bestehend aus Speicherformat, Zustand, Elementgröße und ggf. Zeichensatz und HOLDER der ausgewählten Elemente ausgegeben werden soll.

**HISTORY = \*UNCHANGED / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein HISTORY-Block bestehend aus Benutzerdatum/-zeit, Erzeugungsdatum/-zeit, Änderungsdatum /-zeit und ggf. Zugriffsdatum/-zeit der ausgewählten Elemente ausgegeben werden soll.

**SECURITY = \*UNCHANGED / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein SECURITY-Block der ausgewählten Elemente ausgegeben werden soll, sofern für ein Zugriffsrecht bzw. Ausleihrecht ein zusätzlicher Elementschutz vereinbart ist.

**LAYOUT = \*UNCHANGED / \*VARIABLE / \*FIXED**

Bestimmt das Format des auszugebenden Inhaltsverzeichnisses.

**LAYOUT = \*VARIABLE**

Die Anzahl der Druckspalten ist abhängig von der längsten Elementbezeichnung innerhalb eines Elementtyps. Bei der Ausgabe auf den Bildschirm und keiner speziellen Sortierung richtet sich das Layout nach der Elementbezeichnung innerhalb des Ausgabepuffers, wobei nachfolgende Ausgaben innerhalb eines Elementtyps sich nur verändern, falls längere Elementbezeichnungen vorkommen.

**LAYOUT = \*FIXED**

Das Inhaltsverzeichnis wird einspaltig im festen Format gedruckt. Eine Spalte bedeutet die untereinander stehenden Einträge im Inhaltsverzeichnis.

**SORT = \*UNCHANGED / \*BY-NAME / \*BY-VERSION / \*BY-USER-DATE / \*BY-CREATION-DATE / \*BY-MODIFICATION-DATE / \*BY-ACCESS-DATE / \*BY-ELEMENT-SIZE / \*BY-SECONDARY-NAME**

Sortierkriterium für die Inhaltsverzeichniseinträge der ausgewählten Elemente. Als erstes Sortierkriterium wird immer der Typ verwendet.

**SORT = \*BY-NAME**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Name und Version.

**SORT = \*BY-VERSION**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Version und Name.

**SORT = \*BY-USER-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Benutzerdatum, Name und Version.

**SORT = \*BY-CREATION-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Erzeugungsdatum, Name und Version.

**SORT = \*BY-MODIFICATION-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Änderungsdatum, Name und Version.

**SORT = \*BY-ACCESS-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Zugriffsdatum, Name und Version.

**SORT = \*BY-ELEMENT-SIZE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Elementgröße, Name und Version.

**SORT = \*BY-SECONDARY-NAME**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Sekundärname, Sekundärattribut, Name und Version.

*Hinweis*

Weitere Hinweise siehe Anweisung SHOW-ELEMENT-ATTRIBUTES.

**OUTPUT-FORM = \*UNCHANGED / \*STD / \*CHARACTER / \*HEXADECIMAL / \*DUMP**

Festlegung des Darstellungsformates für die Ausgabe.

**OUTPUT-FORM = \*STD**

Das Darstellungsformat wird abhängig vom Typ des Elementes gewählt. Für Textelemente wirkt dieser Operand wie OUTPUT-FORM = \*CHARACTER.

**OUTPUT-FORM = \*CHARACTER**

Die Ausgabe wird alphanumerisch dargestellt.

**OUTPUT-FORM = \*HEXADECIMAL**

Die Ausgabe wird hexadezimal übereinander dargestellt.

**OUTPUT-FORM = \*DUMP**

Die Ausgabe wird alphanumerisch und hexadezimal dargestellt.

**DELETE-SOURCE = \*UNCHANGED / \*NO / \*YES**

Hier kann der Benutzer bestimmen, ob die ursprüngliche Datei erhalten bleiben (Default-Wert \*NO) oder gelöscht (Parameter \*YES) werden soll. Dieser Operand ist wirkungslos, wenn die Daten von \*OMF gelesen werden.

**PROTECTION = \*UNCHANGED / \*STD / \*BY-SOURCE**

Elementenschutz für das Element oder Dateischutz für die Datei.

**PROTECTION = \*STD**

Für Elementschutz:

Existiert das Element bereits, bleibt der Elementschutz unverändert. Existiert das Element noch nicht und ist für die Bibliothek bzw. den Typ des Elements ein initialer Elementschutz festgelegt, so erhält das Element diesen Schutz.

Für Dateischutz:

Der wirksame Elementschutz geht nicht in den Dateischutz der zu erzeugenden Datei ein.

**PROTECTION = \*BY-SOURCE**

Das kopierte Element erhält den selben Schutz wie das ursprüngliche Element bzw.

das aufzunehmende Element erhält einen Elementschutz entsprechend den Dateischutzattributen des aktivierten Zugriffsschutzmechanismus der Datei bzw.

die zu erzeugende Datei erhält als Dateischutz einen Zugriffsschutzmechanismus, der dem für das Element wirksamen Elementschutz entspricht.

**MAX-ERROR-WEIGHT = \*UNCHANGED / \*SERIOUS / \*SIGNIFICANT / \*RECOVERABLE**

Dieser Operand bestimmt, in welchen Fehlerfällen LMS den Spin-off Mechanismus auslösen soll.

**MAX-ERROR-WEIGHT = \*SERIOUS**

Der Spin-off-Mechanismus wird bei schwerwiegenden Fehlern, d.h. Fehlern, bei denen eine Fortsetzung der Anweisung nicht sinnvoll ist, ausgelöst.

**MAX-ERROR-WEIGHT = \*SIGNIFICANT**

Der Spin-off-Mechanismus wird wie bei \*SERIOUS und zusätzlich bei sonstigen Fehlern (außer Element konnte nicht gefunden oder überschrieben werden) ausgelöst.

**MAX-ERROR-WEIGHT = \*RECOVERABLE**

Der Spinoff-Mechanismus wird bei allen Fehlern ausgelöst.

**EDT-MODE = \*UNCHANGED / \*COMPATIBLE / \*UNICODE**

Legt den Betriebsmodus fest, in dem der EDT aufgerufen wird.

**EDT-MODE = \*COMPATIBLE**

Der EDT wird im Kompatibilitätsmodus aufgerufen.

**EDT-MODE = \*UNICODE**

Der EDT wird im Unicode-Modus aufgerufen.

**NEXT-ATTEMPT = \*UNCHANGED / \*NO / \*YES(...)**

Steuert die weiteren Öffnungsversuche bei Datei-, Typ- oder Elementsperrung im Prozedur- oder Stapelbetrieb.

**NEXT-ATTEMPT = \*NO**

Keine weiteren Öffnungsversuche werden vorgenommen.

**NEXT-ATTEMPT = \*YES(...)**

Weitere Öffnungsversuche werden vorgenommen.

**NUMBER-OF-ATTEMPTS =\*UNCHANGED / <integer 1..2147483647>**

Anzahl der weiteren Öffnungsversuche (Standard sind 9 Versuche).

**PERIOD =\*UNCHANGED / <integer 1..21599>**

Wartezeit in Sekunden zwischen zwei Versuchen (Standard sind 6 Sekunden).

**COMPARE-PARAMETERS =\*UNCHANGED / \*PARAMETERS(...)**

Bestimmt die Vergleichsparameter. Die Art des Vergleichs (formal oder logisch), Umfang und Format der Protokollierung können festgelegt werden.

**RECORD-PART =\*UNCHANGED / \*ALL / \*PART(...)**

Bestimmt den Vergleichsbereich im Satz.

**RECORD-PART = \*ALL**

Der ganze Satz wird verglichen.

**RECORD-PART = \*PART(...)**

Bereichsangabe für den zu vergleichenden Teil des Satzes.

**START = \*UNCHANGED / <integer 1..32764>**

Bereichsanfang für den zu vergleichenden Teil des Satzes. Wird kein Wert angegeben, wird der Satz von Anfang an genommen.

**LENGTH = \*UNCHANGED / \*REST / <integer 1..32764>**

Länge des zu vergleichenden Bereichs im Satz. Wird kein Wert angegeben, wird der Satz vom Startanfang an genommen.

**SPACES = \*UNCHANGED / \*STD / \*IGNORED / \*RELEVANT**

Behandlung von Leerzeichen im Satz.

**SPACES = \*STD**

Wirkt für Textelemente wie \*IGNORED, sonst wie \*RELEVANT.

**SPACES = \*IGNORED**

Logischer Vergleich; die Vergleichsfelder werden Zeichen für Zeichen verglichen, Zwischenräume werden übergangen.

**SPACES = \*RELEVANT**

Formaler Vergleich; die Vergleichsfelder werden zuerst auf gleiche Länge überprüft. Sind die Längen gleich, werden die Felder vollständig verglichen. Sind die Längen unterschiedlich, werden die Sätze als ungleich protokolliert.

**INFORMATION = \*UNCHANGED / \*MEDIUM / \*MINIMUM / \*MAXIMUM / \*SUMMARY / \*STATISTICS / \*NONE**

Umfang der Protokollierung.

**INFORMATION = \*MEDIUM**

Standard-Vergleichsprotokoll. Der Vergleichsbereich der ungleichen Sätze wird vollständig protokolliert. Bei gleichen Sätzen werden nur Bereichsangaben (Satznummern) protokolliert. Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*MINIMUM**

Minimales Vergleichsprotokoll.

Für gleiche und ungleiche Sätze werden nur Bereichsangaben (Satznummern) protokolliert. Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*MAXIMUM**

Ausführliches Vergleichsprotokoll.

Alle Sätze werden protokolliert.

Die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*SUMMARY**

Kein Vergleichsprotokoll.

Nur die Vergleichsstatistik wird ausgegeben.

**INFORMATION = \*STATISTICS**

Kein Vergleichsprotokoll.

Die Vergleichsstatistik wird in komprimierter Form ausgegeben.

**INFORMATION = \*NONE**

Keine Protokollierung, weder Vergleichsprotokoll noch Vergleichsstatistik.

\*NONE ist nur sinnvoll bei Verwendung der Anweisung SHOW-STATISTICS.

**LAYOUT = \*UNCHANGED / \*COMPATIBLE / \*COMPRESSED**

Format der Protokollierung.

**LAYOUT = \*COMPATIBLE**

Das Vergleichsprotokoll wird im Standardformat ausgegeben.

Dieses Format ist kompatibel zu früheren LMS-Versionen.

**LAYOUT = \*COMPRESSED**

Das Vergleichsprotokoll wird in einem komprimierten Format ausgegeben.

**JOIN-ELEMENT-SETS = \*UNCHANGED / \*NO / \*YES**

Legt die Elementmenge fest, die verglichen werden soll.

**JOIN-ELEMENT-SETS = \*NO**

Nur die Primärelemente und die durch Konstruktion ermittelten Sekundärelemente werden zum Vergleich herangezogen.

**JOIN-ELEMENT-SETS = \*YES**

Alle Primär- und Sekundärelemente werden zum Vergleich herangezogen.



**TEXT-INFORMATION = \*UNCHANGED / \*ALL / \*STATISTICS / \*FILE-ATTRIBUTES / \*PARAMETERS(...)**

Legt die Informationsmenge für alle Elemente außer den Elementtypen R, C und L fest.

**TEXT-INFORMATION = \*ALL**

Alles wird ausgegeben.

**TEXT-INFORMATION = \*STATISTICS**

Die Anzahl der Sätze pro Satzart und deren Summe werden ausgegeben.

**TEXT-INFORMATION = \*FILE-ATTRIBUTES**

Nur die gespeicherten Dateiattribute werden ausgegeben.

**TEXT-INFORMATION = \*PARAMETERS(...)**

Legt einen Elementausschnitt fest, der ausgegeben werden soll.

**INFORMATION = \*UNCHANGED / \*ALL / list-poss(2): \*TEXT / \*COMMENT**

Der anzuzeigende Teilbereich des Elementes.

**INFORMATION = \*ALL**

Es wird alles ausgegeben.

**INFORMATION = \*TEXT**

Der eigentliche Text, d.h. Satzart 1 wird angezeigt.

**INFORMATION = \*COMMENT**

Der separat gespeicherte Text, d.h. Satzart 2 wird angezeigt.

**RECORD-RANGE = \*UNCHANGED / \*ALL / \*RANGE(...)**

Der zu bearbeitende Teilbereich des Elementes.

**RECORD-RANGE = \*ALL**

Es werden alle Sätze bearbeitet.

**RECORD-RANGE = \*RANGE(...)**

Der zu bearbeitende Satznummernbereich wird bestimmt. Die Satznummern beziehen sich nicht auf eine Satzart, sondern auf den durch INFORMATION= festgelegten Teilbereich des Elementes. Innerhalb dieses Teilbereiches werden die Sätze von 1 bis n durchnummeriert.

**FROM = \*UNCHANGED / <integer 1..2147483647>**

Beginn des Bereichs durch Angabe der ersten Satznummer. Standardmäßig wird die Satznummer 1 genommen.

**TO = \*UNCHANGED / \*LAST / <integer 1..2147483647>**

Ende des Bereichs durch Angabe der letzten Satznummer. Standardmäßig wird die letzte Satznummer genommen.

**RECORD-PART = \*UNCHANGED / \*ALL / \*PART(...)**

Der zu bearbeitende Satzbereich wird bestimmt.

**RECORD-PART = \*ALL**

Der ganze Satz wird bearbeitet.

**RECORD-PART = \*PART(...)**

Der zu bearbeitende Satzbereich wird bestimmt. Werden die Standardwerte nicht verändert, wird der ganze Satz bearbeitet.

**START = \*UNCHANGED / <integer 1..262144>**

Beginn des Bereichs durch Angabe der ersten Zeichens im Satz. Standardmäßig wird das erste Zeichen genommen.

**LENGTH = \*UNCHANGED / \*REST / <integer 1..262144>**

Länge des Bereichs. Standardmäßig wird der Rest des Satzes genommen.

LENGTH = 0 bedeutet, dass nur die Satzköpfe ausgegeben werden.

**RECORD-NUMBER = \*UNCHANGED / \*BY-OUTPUT / \*YES / \*NO**

Bestimmt die Ausgabe der Satznummern.

**RECORD-NUMBER = \*BY-OUTPUT**

Nur wenn die Ausgabe nach SYSOUT erfolgt, werden keine Satznummern ausgegeben. Bei jedem anderen Ausgabemedium werden sie mit ausgegeben.

**RECORD-NUMBER = \*YES**

Die Satznummern werden auch nach SYSOUT ausgegeben.

**RECORD-NUMBER = \*NO**

Es werden keine Satznummern mit ausgegeben.

**MODULE-INFORMATION = \*UNCHANGED / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Bindemodule (Elemente vom Typ R) fest.

**MODULE-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**MODULE-INFORMATION = \*STATISTICS**

Name, Länge und Adresse der CSECTS sowie die Gesamtlänge des Moduls werden ausgegeben.

**MODULE-INFORMATION = \*PARAMETERS(...)**

Legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden sollen.

**INFORMATION = \*UNCHANGED / \*ALL / \*TXT(...) / \*TXTP(...) / list-poss(9): \*ESD / \*ISD / \*LSD / \*RLD / \*REP / \*INCLUDE / \*DSDD / \*REF / \*END**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Textsätze werden ausgewählt.

**CSECT-NAME = \*UNCHANGED / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die Textsätze können auf eine CSECT eingeschränkt werden.

**ADDRESS = \*UNCHANGED(...) / <x-string 1..8>(...**

Anfangsadresse des Textes.

**BASE-ADDRESS = \*UNCHANGED / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

**LENGTH = \*UNCHANGED / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes.

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgeben.

**MODIFICATION-ID = \*UNCHANGED / \*ALL / <c-string 1..8 with-low> / \*RANGE(...)**

Die TXTP-Sätze mit der angegebenen Identifikation werden ausgewählt.

**MODIFICATION-ID = \*RANGE(...)**

Mehrere TXTP-Sätze, die in einem Bereich liegen, können ausgewählt werden.

**FROM = \*UNCHANGED / \*LOWEST / <c-string 1..8 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**TO = \*UNCHANGED / \*HIGHEST / <c-string 1..8 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**PHASE-INFORMATION = \*UNCHANGED / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Phasen (Elemente vom Typ C) fest.

**PHASE-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**PHASE-INFORMATION = \*STATISTICS**

Name, Länge und Adresse des Segmentes sowie die Gesamtlänge des Segmentes werden ausgegeben.

**PHASE-INFORMATION = \*PARAMETERS(...)**

Legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

**SEGMENT = \*UNCHANGED / \*ALL / \*ROOT / <name 1..8>**

Phasensegment, das ausgewählt wird.

**INFORMATION = \*UNCHANGED / \*ALL / \*TXT(...) / \*TXTP(...) / list-poss(4): \*ESD / \*ISD / \*LSD / \*RLD**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Textsätze werden ausgewählt.

**ADDRESS = \*UNCHANGED(...) / <x-string 1..8>(...**

Anfangsadresse des Textes.

**BASE-ADDRESS = \*UNCHANGED / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

**LENGTH = \*UNCHANGED / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes.

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgeben.

**MODIFICATION-ID = \*UNCHANGED / \*ALL / <c-string 1..8 with-low> / \*RANGE(...)**

Die TXTP-Sätze mit der angegebenen Identifikation werden ausgewählt.

**MODIFICATION-ID = \*RANGE(...)**

Mehrere TXTP-Sätze, die in einem Bereich liegen, können ausgewählt werden.

**FROM = \*UNCHANGED / \*LOWEST / <c-string 1..8 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**TO = \*UNCHANGED / \*HIGHEST / <c-string 1..8 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**LLM-INFORMATION = \*UNCHANGED / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Bindeladmodule (Elemente vom Typ L) fest.

**LLM-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**LLM-INFORMATION = \*STATISTICS**

Allgemeine Informationen über das Bindeladmodul (Name, Copyright, ...) werden ausgegeben.

**LLM-INFORMATION = \*PARAMETERS(...)**

Legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

**LLM-PART = \*UNCHANGED / \*ALL / \*SLICE(...) / \*SUB-LLM(...)**

Angabe des LLM-Teils, der ausgewählt werden soll. Standardmäßig wird der gesamte LLM-Teil ausgewählt.

**LLM-PART = \*SLICE(...)**

Angabe der Slice, die ausgegeben werden soll.

**NAME = \*UNCHANGED / <structured-name 1..32>**

Name der Slice, die ausgegeben werden soll.

**LLM-PART = \*SUB-LLM(...)**

Angabe des SUB-LLM, der ausgegeben werden soll.

**PATH-NAME = \*UNCHANGED / <c-string 1..255 with-low> / <text 1..255>**

Der auszugebende SUB-LLM wird durch seinen Pfadnamen bestimmt.

**INFORMATION = \*UNCHANGED / \*ALL / \*TXT(...) / \*TXTP(...) / \*LOGICAL(...) / \*PHYSICAL / \*REF / list-poss(4): \*RELOCATION / \*ESVD / \*ESVR / \*LRLD**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Textsätze werden ausgewählt.

**CSECT-NAME = \*UNCHANGED / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die Textsätze können auf eine CSECT eingeschränkt werden.

**ADDRESS = \*UNCHANGED(...) / <x-string 1..8>(...**

Anfangsadresse des Textes.

**BASE-ADDRESS = \*UNCHANGED / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

**LENGTH = \*UNCHANGED / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes.

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgeben.

**CSECT-NAME = \*UNCHANGED / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die TXTP-Sätze können auf eine CSECT eingeschränkt werden.

**MODIFICATION-ID = \*UNCHANGED / \*ALL / <c-string 1..12 with-low> / \*RANGE(...)**

Die TXTP-Sätze mit der angegebenen Identifikation werden ausgewählt.

**MODIFICATION-ID = \*RANGE(...)**

Mehrere TXTP-Sätze, die in einem Bereich liegen, können ausgewählt werden.

**FROM = \*UNCHANGED / \*LOWEST / <c-string 1..12 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**TO = \*UNCHANGED / \*HIGHEST / <c-string 1..12 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder die hier eingegebene.

**INFORMATION = \*LOGICAL(...)**

Die logische Struktur des LLM wird ausgegeben.

**LEVEL = \*UNCHANGED / \*ALL / \*NEXT**

Es werden standardmäßig alle oder nur die nächste Unterstruktur ausgegeben.

**INFORMATION = \*PHYSICAL**

Die physikalische Struktur des LLM wird ausgegeben.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Beispiel*

Es soll die Datei TEST1 als Elemente vom Typ D in die Bibliothek LIB3 aufgenommen werden.

Die standardmäßige Einstellung für den Elementtyp wird mit der Anweisung MODIFY-LMS-DEFAULTS auf den gewünschten Wert geändert. Dadurch ist es nicht mehr nötig, bei den nachfolgenden ADD-ELEMENT-Anweisungen jeweils den Elementtyp anzugeben. Damit LMS die erfolgreiche Aufnahme der Dateien meldet, wird durch die Anweisung MODIFY-LOGGING-PARAMETERS der Umfang der Protokollmenge auf das komplette LMS-Protokoll gesetzt.

```
//START-LMS
//OPEN-LIBRARY LIB3,*UPDATE
//MODIFY-LMS-DEFAULTS TYPE=D
//MODIFY-LOGGING-PARAMETERS LOGGING=*MAXIMUM
//ADD-ELEMENT TEST1
INPUT FILE
OUTPUT LIBRARY= :10SQ:$USER.LIB3
        ADD :10SQ:$USER.TEST1 AS (D)TEST1/@(0001)/2013-03-01
//END
```

Die Bibliothek LIBCCSN enthält ein Element LONGR im EDF03IRV-Code mit Sätzen länger als 256 Byte. Damit diese langen Sätze im EDT bearbeitet werden können, muss EDT im Unicode-Modus aufgerufen werden.

Deshalb wird die Default-Einstellung für den EDT-Betriebsmodus (COMPATIBLE) mit der Anweisung MODIFY-LMS-DEFAULTS in den gewünschten Unicode-Modus geändert. Nun ruft die Anweisung EDIT-ELEMENT den EDT im Unicode-Modus auf und liest das Element LONGR in die Arbeitsdatei 0 ein.

```
//START-LMS
//MOD-LMS-DEF EDT-MODE=*UNICODE
//OPEN-LIB LIBCCSN, *UPD
//EDIT-ELEM (,LONGR,S)
//END
```

## MODIFY-LOGGING-PARAMETERS

### LMS-Protokollparameter ändern

Die Anweisung MODIFY-LOGGING-PARAMETERS ändert die globale Einstellung für die Protokollmenge, das Ausgabemedium und das Protokollformat.

Wird einer dieser Werte durch die MODIFY-LOGGING-PARAMETERS-Anweisung verändert, wird diese neue Einstellung zur aktuellen Einstellung. Diese bleibt solange für den LMS-Lauf gültig (\*UNCHANGED), bis eine neue Anweisung MODIFY-LOGGING-PARAMETERS für diesen Wert oder RESET-LOGGING-PARAMETERS gegeben wird.

Zu Beginn des LMS-Laufs gelten die unmittelbar auf \*UNCHANGED folgenden Werte. Die Standardwerte bzw. die aktuellen Werte können mit der entsprechenden SHOW-Anweisung ausgegeben werden..

#### MODIFY-LOGGING-PARAMETERS

```

LOGGING = *UNCHANGED / *MINIMUM / *MAXIMUM
,TEXT-OUTPUT = UNCHANGED / *SYSOUT / *SYSLST(...) / *NONE / *EDT(...) / *LIBRARY-ELEMENT(...)
  *SYSLST(...)
    | SYSLST-NUMBER = *STD / <integer 1..99>
  *EDT(...)
    | WRITE-MODE = UNCHANGED / *EXTEND / *REPLACE
  *LIBRARY-ELEMENT(...)
    | LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
      *LINK(...)
        | LINK-NAME = <structured-name 1..8>
      ,ELEMENT = <composed-name 1..64 with-under>(…)
        <composed-name 1..64 with-under>(…)
          | VERSION = UPPER-LIMIT / *HIGHEST-EXISTING / *INCREMENT /
            <composed-name 1..24 with-under>
          ,BASE = *STD / <composed-name 1..24 with-under with-wild>
      ,TYPE = P / <alphanum-name 1..8>
    ,WRITE-MODE = UNCHANGED / *CREATE / *REPLACE / *EXTEND / *ANY

```

(Teil 1 von 2)



```

,OUTPUT-LAYOUT = *UNCHANGED / *PARAMETERS(...)
  *PARAMETERS(...)
    LINES-PER-PAGE = *UNCHANGED / <integer 1..9999>
    ,LINE-SIZE = *UNCHANGED / 132 / 80
    ,EXTRA-FORM-FEED = *UNCHANGED / *NO / *YES
    ,HEADER-LINES = *UNCHANGED / *YES / *NO

```

(Teil 2 von 2)

**LOGGING = \*UNCHANGED / \*MINIMUM / \*MAXIMUM**

Festlegen der LMS-Protokollmenge.

**LOGGING = \*MINIMUM**

Es werden nur Fehlermeldungen und Misserfolgsmeldungen ausgegeben.

**LOGGING = \*MAXIMUM**

Es wird ein komplettes LMS-Protokoll ausgegeben.

**TEXT-OUTPUT = \*UNCHANGED / \*SYSOUT / \*SYSLST(...) / \*NONE / \*EDT(...) / \*LIBRARY-ELEMENT(...)**

Dieser Parameter bestimmt das Ausgabemedium. Bei Medienwechsel und bei WRITE-MODE=\*EXTEND beginnt die Seitennumerierung immer mit 1.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*NONE**

Die Ausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT.

Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*UNCHANGED / \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe in bezug auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren in der Arbeitsdatei 9 Daten, werden sie durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

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

Die Ausgabe wird in ein Bibliothekselement gespeichert.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in die die Ausgabe gespeichert werden soll. Es wird entweder standardmäßig die global durch OPEN-LIBRARY eingestellte Bibliothek oder die explizit angegebene oder die durch den Dateikettungsamen zugewiesene Bibliothek verwendet.

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

Dateikettungsname der Bibliothek.

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

Angabe des Elementes, in das die Ausgabe gespeichert werden soll.

**VERSION = \*UPPER-LIMIT / \*HIGHEST-EXISTING / \*INCREMENT / <composed-name 1..24 with-under>**

Angabe der Version, die das Element erhalten soll.

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version erzeugt, ansonsten eine Defaultversion (siehe auch [Seite 59](#)).

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-underscore with-wild>**

Legt die Basis für das Zielelement fest. Näheres zur Angabe der Basis siehe [Seite 54](#).

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

Angabe des Elementtyps. Standardmäßig erhält das Element, in das die Ausgabe gespeichert wird, den Typ P für druckaufbereitete Dateien.

**WRITE-MODE = \*UNCHANGED / \*CREATE / \*REPLACE / \*EXTEND / \*ANY**

Überschreiben eines Elementes gleichen Namens. Existiert das Element nicht unter diesem Namen, wird es neu angelegt.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren und wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*EXTEND**

Das Zielelement wird erweitert, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**WRITE-MODE = \*ANY**

Das Zielelement wird ersetzt, wenn es bereits existiert. Ansonsten wird es neu angelegt.

**OUTPUT-LAYOUT = \*UNCHANGED / \*PARAMETERS(...)**

Dieser Parameter bestimmt das LMS-Protokollformat.

**LINES-PER-PAGE = \*UNCHANGED / <integer 1..9999>**

Dieser Parameter legt die Seitenlänge fest.  
Standardwert: 64 Zeilen

**LINE-SIZE = \*UNCHANGED / 132 / 80**

Dieser Parameter legt die Zeilenlänge fest.

**LINE-SIZE = 132**

Die Zeile soll 132 Zeichen lang sein.

**LINE-SIZE = 80**

Die Zeile soll 80 Zeichen lang sein.

**EXTRA-FORM-FEED = \*UNCHANGED / \*NO / \*YES**

Dieser Parameter steuert einen Extra-Seitenvorschub.

**EXTRA-FORM-FEED = \*NO**

Es gibt nur dann einen Seitenvorschub, wenn die Seite voll ist.

**EXTRA-FORM-FEED = \*YES**

Es gibt einen Seitenvorschub, wenn entweder die Seite voll ist oder wenn ein Anweisungswechsel oder Elementwechsel stattfindet.

**HEADER-LINES = \*UNCHANGED / \*YES / \*NO**

Dieser Parameter steuert die Ausgabe von Überschriften.

**HEADER-LINES = \*YES**

Es werden Überschriften mit der Bibliotheks- und Elementbezeichnung ausgegeben.

**HEADER-LINES = \*NO**

Es werden keine Überschriften ausgegeben.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0151	Ein- oder Ausgabemedium auf Standard gesetzt
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0093	Protokoll-Element existiert bereits
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0510	Basis nicht gefunden
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Es sind keine Zugriffsrechte erforderlich.

## MODIFY-TYPE-ATTRIBUTES

### Typattribute ändern

Mit dieser Anweisung können für den angegebenen Typ bestimmte Attribute eingestellt/geändert werden:

- der Supertyp
- die geltende Versionskonvention
- die Speicherungform der Elemente
- Zusatzprüfungen beim Erzeugen oder Überschreiben von Elementen
- der Berechtigtenkreis zum Erzeugen, Löschen und Umbenennen von Elementen und der initiale Elementschutz

Zu Beginn des LMS-Laufs gelten die unmittelbar auf \*UNCHANGED folgenden Werte. Die Standardwerte bzw. die aktuellen Werte können mit der entsprechenden SHOW-Anweisung ausgegeben werden.

#### MODIFY-TYPE-ATTRIBUTES

```

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
,TYPE = *LMS-DEFAULT / <alphanum-name 1..8>
,SUPER-TYPE = *UNCHANGED / *NONE / <alphanum-name 1..8>
,CONVENTION = *UNCHANGED / *NONE / *STD-TREE / *STD-SEQUENCE(...) / *MULTI-SEQUENCE(...)
  *STD-SEQUENCE(...)
    | EXAMPLE = 001 / <composed-name 1..24 with-under>
  *MULTI-SEQUENCE(...)
    | EXAMPLE = <composed-name 1..24 with-under>
,STORAGE-FORM = *UNCHANGED / *NONE / *STD / *FULL / *DELTA
,WRITE-CONTROL = *UNCHANGED / *NONE / *DEACTIVATE / *ACTIVATE
,ADMINISTRATION = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
  *BY-GUARD(...)
    | GUARD-NAME = <filename 1..18 without-cat-gen-vers>
  *PARAMETERS(...)
    | USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    | ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
      <x-string 1..8> / <integer -2147483648..2147483647>

```

(Teil 1 von 2)

```

,INIT-ELEM-PROTECTION = *UNCHANGED / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    READ = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,WRITE = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,EXEC = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>
    ,HOLD = *UNCHANGED / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
      *BY-GUARD(...)
        |   GUARD-NAME = <filename 1..18 without-cat-gen-vers>
      *PARAMETERS(...)
        |   USER = *UNCHANGED / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
        |   ,PASSWORD = *UNCHANGED / *SECRET / *NONE / <c-string 1..4> /
        |               <x-string 1..8> / <integer -2147483648..2147483647>

```

(Teil 2 von 2)

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der die Typattribute festgelegt werden sollen.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der die Typattribute festgelegt werden sollen.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

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

Elementtyp, dessen Attribute eingestellt oder geändert werden sollen.

**SUPER-TYPE = \*UNCHANGED / \*NONE / <alphanum-name 1..8>**

Name des übergeordneten Typs

**SUPER-TYPE = \*NONE**

Es gibt keinen übergeordneten Typ.

**SUPER-TYPE = <alphanum-name 1..8>**

Name des übergeordneten Typs. Der entstehende Graph muss zyklensfrei (Baum) sein. Für Standardtypen (ein Zeichen lang bzw. Typen, die mit \$ oder SYS beginnen) kann kein SUPER-TYPE vereinbart werden. Die Werte R, C und L sind nicht als SUPER-TYPE erlaubt.

**CONVENTION = \*UNCHANGED / \*NONE / \*STD-TREE / \*STD-SEQUENCE(...) / \*MULTI-SEQUENCE(...)**

Versionskonvention, die von nun an für den spezifizierten Typ gelten soll.

**CONVENTION = \*NONE**

Die Versionskonvention wird von dem spezifizierten Typ entfernt. Dies ist jederzeit möglich.

**CONVENTION = \*STD-TREE**

Die Versionskonvention \*STD-TREE gilt für den betreffenden Typ. Es ist nicht erlaubt, \*STD-TREE anzugeben, wenn ein Element unter dem spezifizierten Typ existiert.

**CONVENTION = \*STD-SEQUENCE(...)**

Die Versionskonvention \*STD-SEQUENCE gilt für den betreffenden Typ. Alle Elemente des Typs besitzen das gleiche Versionsformat, das durch ein Beispiel spezifiziert wird. Es ist nicht erlaubt, \*STD-SEQUENCE anzugeben, wenn ein Element unter dem spezifizierten Typ existiert.

**EXAMPLE = 001 / <composed-name 1..24 with-under>**

Beispiel für das Versionsformat. Das Format besteht aus einem, eventuell leeren, führenden Teil und der maximal langen abschließenden Zifferngruppe, die nicht leer sein darf. Bei allen Versionen unter dem betreffenden Typ müssen diese beiden Teile die gleiche Länge haben wie das Beispiel. Die abschließende Zifferngruppe wird für die automatische Versionserhöhung verwendet.

Wird hier keine explizite Angabe gemacht, wird der Wert „001“ als Beispiel für das Versionsformat verwendet.

**CONVENTION = \*MULTI-SEQUENCE(...)**

Die Versionskonvention \*MULTI-SEQUENCE gilt für den betreffenden Typ. Alle Elemente des Typs besitzen das gleiche Versionsformat, das durch ein Beispiel spezifiziert wird. Es ist nicht erlaubt, \*MULTI-SEQUENCE anzugeben, wenn ein Element unter dem spezifizierten Typ existiert.

**EXAMPLE = <composed-name 1..24 with-under>**

Beispiel für das Versionsformat. Das Format besteht aus einem eventuell leeren, führenden Teil und der maximal langen abschließenden Zifferngruppe, die nicht leer sein darf. Bei allen Versionen unter dem betreffenden Typ müssen diese beiden Teile die gleiche Länge haben wie das Beispiel. Die abschließende Zifferngruppe wird für die automatische Versionserhöhung verwendet.

**STORAGE-FORM = \*UNCHANGED / \*NONE / \*STD / \*FULL / \*DELTA**

Zulässige Speicherform für Elemente dieses Typs. Alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*NONE**

Es ist sowohl Voll- als auch Deltaspeicherung zulässig.

**STORAGE-FORM = \*STD**

Sowohl Voll- als auch Deltaspeicherung sind zulässig.

**STORAGE-FORM = \*FULL**

Es ist nur Vollspeicherung zulässig.

**STORAGE-FORM = \*DELTA**

Nur Deltaspeicherung ist zulässig.

**WRITE-CONTROL = \*UNCHANGED / \*NONE / \*DEACTIVATE / \*ACTIVATE**

Attribut zur Steuerung zusätzlicher Prüfungen.

**WRITE-CONTROL = \*NONE**

Es wirkt die Einstellung von WRITE-CONTROL für die Bibliothek.

**WRITE-CONTROL = \*DEACTIVATE**

Es gibt keine zusätzlichen Prüfungen beim Erzeugen oder Überschreiben von Versionen.



**WRITE-CONTROL = \*ACTIVATE**

Eine Version darf nur geschrieben werden, wenn

- die USERID desjenigen, der schreiben will, bei der explizit oder implizit immer gegebenen Basisversion als HOLDER eingetragen ist, und
- entweder eine neue Version erzeugt oder die Basisversion überschrieben wird.

Bei der ersten Version eines Namens gibt es noch keine Basis; sie kann nur von ADMIN-Berechtigten erzeugt werden. Beim Erzeugen oder Überschreiben von Versionen wird automatisch ein Satz der Satzart 2 zugefügt, der HOLDER=Schreiber, DATE und TIME des Vorgangs dokumentiert. Ferner werden die Attribute STATE, HOLDER und alle Rechte auf die neue Version übernommen, falls durch die aktuelle Anweisung nicht das Eintragen anderer Werte verlangt wird.

**ADMINISTRATION = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Administrationsrecht. Der Kreis der Administrationsberechtigten wird für diesen Typ explizit festgelegt. Nur diese dürfen Elemente anlegen, löschen und umbenennen.

**ADMINISTRATION = \*NONE**

Es wird kein Administrationsrecht für diesen Typ vergeben. Es gilt die Einstellung für die Bibliothek.

**ADMINISTRATION = \*BY-GUARD(...)**

Das Verwaltungsrecht für diesen Typ wird über einen GUARD geregelt.

**GUARD-NAME = <filename 1..18 without-cat-gen-vers>**

Name des GUARDS.

**ADMINISTRATION = \*PARAMETERS(...)**

Angabe der Benutzerkreise und evtl. eines Kennwortes für die Verwaltung des angegebenen Typs.

**USER = \*UNCHANGED / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Verwaltungsberechtigten wird explizit aufgezählt.

**USER = \*NONE**

Keiner darf verwalten.

**USER = \*ALL**

Alle dürfen verwalten (komplette Aufzählung).

**USER = \*OWNER**

Nur der Eigentümer der Bibliotheksdatei darf verwalten.

**USER = \*GROUP**

Die Angehörigen der Gruppe des Eigentümers der Bibliotheksdatei dürfen verwalten.

**USER = \*OTHERS**

Alle anderen dürfen verwalten.

**PASSWORD = \*UNCHANGED / \*SECRET / \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>**

Der Kreis der Berechtigten wird weiter eingeschränkt. Zusätzlich zum notwendigen Zugriffsrecht wird das richtige Kennwort verlangt. Die Angabe 0 bzw. X'00000000' führt nicht zu einer Änderung des letzten Wertes.

Die Angabe von \*SECRET bzw. ^ ermöglicht die verdeckte Eingabe des gewünschten Kennwortes. Wird der „geheime“ Wert als c-string eingegeben, muss er in Hochkommata stehen. Wird er als x-string eingegeben, muss er ebenfalls in Hochkommata angegeben und ein X vorangestellt werden.

**INIT-ELEM-PROTECTION = \*UNCHANGED / \*NONE/ \*PARAMETERS(...)**

Angabe des initialen Elementschutzes, der bei den unter obigen Typ neu angelegten Elementen eingetragen werden soll.

**INIT-ELEM-PROTECTION = \*NONE**

Es wird kein initialer Schutz für diesen Elementtyp vereinbart.

**INIT-ELEM-PROTECTION = \*PARAMETERS(...)**

Angabe der Schutzrechte, die der oben spezifizierte Typ erhalten soll.

**READ = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Leserecht wird explizit festgelegt.

**READ = \*NONE**

Es wird keine weitere Zugriffseinschränkung vorgesehen.

**READ = \*BY-GUARD(...)**

Angabe des GUARDS für das Leserecht.

**GUARD-NAME = <filename 1..18 without-cat-gen-vers>**

Name des GUARDS.

**READ = \*PARAMETERS(...)**

Angabe der Benutzerkreise für das Leserecht.

**USER = \*UNCHANGED / \*NONE / \*ALL/ list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Leseberechtigten wird explizit aufgezählt.

**USER = \*NONE**

Keiner darf in der angegebenen Art zugreifen.

**USER = \*ALL**

Alle dürfen in der angegebenen Art zugreifen (komplette Aufzählung).

**USER = \*OWNER**

Der Eigentümer der Bibliotheksdatei darf zugreifen.

**USER = \*GROUP**

Die Angehörigen der Gruppe des Eigentümers der Bibliotheksdatei dürfen zugreifen.

**USER = \*OTHERS**

Alle anderen dürfen zugreifen.

**PASSWORD = \*UNCHANGED / \*SECRET / \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647>**

Der Kreis der Berechtigten wird weiter eingeschränkt. Zusätzlich zum notwendigen Zugriffsrecht wird das richtige Kennwort verlangt.

Die Angabe von \*SECRET bzw. ^ ermöglicht die verdeckte Eingabe des gewünschten Kennwortes. Wird der „geheime“ Wert als c-string eingegeben, muss er in Hochkommata stehen. Wird er als x-string eingegeben, muss er ebenfalls in Hochkommata angegeben und ein X vorangestellt werden.

**WRITE = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Schreibrecht wird explizit festgelegt.

Beschreibung der Operanden analog zu READ (siehe [Seite 370](#)).

**EXEC = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Ausführungsrecht wird explizit festgelegt.

Beschreibung der Operanden analog zu READ.

**HOLD = \*UNCHANGED / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ein initiales Ausleihrecht wird explizit festgelegt.

Beschreibung der Operanden analog zu READ.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0413	Typ gesperrt

### Notwendige Zugriffsrechte

Lese- und Schreibrecht für LIBRARY

Nur der Eigentümer der Bibliotheksdatei kann Attribute des Typs ändern.

*Hinweis*

Die Typattribute bleiben erhalten, auch wenn alle Elemente des Typs gelöscht werden.

*Beispiel*

Für eine Bibliothek X.1 soll gelten:

- Es dürfen nur vorgegebene Typen verwendet werden.
- Für alle Elemente des Typs S wird die Konvention STD-SEQUENCE mit dem Versions-Beispiel V001 eingestellt.
- Dem benutzerdefinierten Typ USER1 soll der SUPER-TYPE 'S' zugewiesen werden.

```
//MODIFY-LIBRARY-ATTRIBUTES LIBRARY=X.1,-                "Bibliothek X"
//                ADMIN=*PAR(USER=*NONE)                "typespez. Einstellung"
//MODIFY-TYPE-ATTRIBUTES TYPE=S,-                        "Standardtyp S"
//                SUPER-TYPE=*NONE,-                    "kein Supertyp"
//                CONVENTION=*STD-SEQUENCE -            "Versionskonvention"
//                (EXAMPLE=V001),-                       "Defaultversion"
//                ADMIN=*PAR(USER=*ALL)                  "jeder darf verwalten"
//MODIFY-TYPE-ATTRIBUTES TYPE=USER1,-                    "pro benutzerdef. Typ"
//                SUPER-TYPE=S,-                          "geeigneter Supertyp"
//                ADMIN=*PAR(USER=*UNCHANGED)            "selektive Einstellung"
//SHOW-TYPE-ATTRIBUTES TYPE=S
INPUT  LIBRARY= :10SQ:$USER.X.1
TYPE   = S
SUPER-TYPE = *NONE
BASE-TYPE  = S
CONVENTION = *STD-SEQUENCE
EXAMPLE    = V001
INIT-ELEM-P= *NONE
ADMIN-PASS = *NONE          ADMIN-USER = *OWNER *GROUP *OTHERS
STORAGE   = *NONE          WR-CONTROL = *NONE
```

Ein weiteres Beispiel für den Elementschutz und Einstellung der Typattribute siehe [Seite 497](#).

## OPEN-LIBRARY

### Bibliothek eröffnen

Mit OPEN-LIBRARY wird eine globale Bibliothek festgelegt und eröffnet. Diese wird in den anderen Anweisungen mit LIBRARY = \*STD angesprochen.

Werden in einer Anweisung zwei Bibliotheken benötigt, so muss die zweite Bibliothek bei der Anweisung explizit oder über Dateikettungsnamen angegeben werden.

Globale Bibliotheken bleiben solange geöffnet, bis sie explizit durch die CLOSE-LIBRARY-Anweisung geschlossen werden oder eine neue OPEN-LIBRARY-Anweisung gegeben wird.

Globale Bibliotheken werden standardmäßig nur zum Lesen eröffnet. Sollen sie zum Lesen und Schreiben eröffnet werden, muss der Operand MODE=\*UPDATE gesetzt werden.

Wird eine Bibliothek neu eingerichtet, muss sie mit MODE=\*UPDATE erzeugt werden.

#### OPEN-LIBRARY

```

LIBRARY = <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
,MODE = *READ / *UPDATE(...)
  *UPDATE(...)
    | STATE = *ANY / *OLD / *NEW
,DEFAULTS = *UNCHANGED / *PARAMETERS (...)
  *PARAMETERS(...)
    | TYPE = *UNCHANGED / *NONE / <alphanum-name 1..8>
,SNAPSET = *NONE / *LATEST / <name 1..1 with-low> / <integer -52..-1>

```

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

Die Bibliothek mit dem hier angegebenen Namen wird als globale Bibliothek eingerichtet und eröffnet.

#### **LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek wird als globale Bibliothek eingerichtet und eröffnet.

#### **LINK-NAME = <structured-name 1..8>**

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**MODE = \*READ / \*UPDATE(...)**

Öffnungsmodus der Bibliothek.

**MODE = \*READ**

Die Bibliothek wird nur zum Lesen eröffnet. Sie muss bereits existieren.

**MODE = \*UPDATE(...)**

Die Bibliothek wird zum Lesen und Schreiben eröffnet.

**STATE = \*ANY / \*OLD / \*NEW**

Status der zu öffnenden Bibliothek.

**STATE = \*ANY**

Die Bibliothek kann existieren. Existiert sie nicht, so wird sie neu angelegt.

**STATE = \*OLD**

Die Bibliothek muss existieren.

**STATE = \*NEW**

Die Bibliothek darf nicht existieren. Sie wird neu angelegt.

**DEFAULTS = \*UNCHANGED / \*PARAMETERS(...)**

Bibliotheksspezifische Defaultwerte für einen LMS-Lauf.

**TYPE = \*UNCHANGED**

Keine Änderung bezüglich des bibliotheksspezifischen Defaulttyps. Standardmäßig wird der Bibliothek kein Defaulttyp zugewiesen.

**TYPE = \*NONE**

Der Bibliothek wird kein bibliotheksspezifischer Defaulttyp zugewiesen, bzw. ein zugewiesener bibliotheksspezifischer Defaulttyp wird zurückgenommen.

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

Name des bibliotheksspezifischen Defaulttyps. Dieser ersetzt bei allen Anweisungen, die diese Bibliothek betreffen, die Typangabe \*LMS-DEFAULT.

**SNAPSET = \*NONE / \*LATEST / <name 1..1 with-low> / <integer -52...-1>**

Der Operand ermöglicht das Eröffnen einer Bibliothek aus einem Snapset.

Die Angabe eines Snapset ist nur in Verbindung mit MODE=\*READ (Default) erlaubt. Bibliotheken auf Snapsets können in den LMS-Anweisungen nur über LIBRARY=\*STD angesprochen werden. Andere LIBRARY-Angaben beziehen sich immer auf das Originalpubset. (Beispiel siehe unten)

Beim Eröffnen einer Snapset-Bibliothek muss auf dem Originalpubset zur Ermittlung diverser Attribute eine Bibliothek gleichen Namens existieren. Fehlt die Bibliothek, lässt sich diese mit /RESTORE-FILE- FROM-SNAPSET als Ganzes wieder herstellen. Ersatzweise kann eine leere Bibliothek mit OPEN-LIBRARY *library,U* eingerichtet werden.

**SNAPSET = \*NONE**

Die Bibliothek liegt nicht auf einem Snapset.

**SNAPSET = \*LATEST**

Die Bibliothek liegt auf dem zuletzt erzeugten Snapset.

\*LATEST entspricht der Angabe -1.

**SNAPSET = <name 1..1 with-low>**

Die Bibliothek liegt auf einem Snapset mit der Snapset-Id a-z,A-Z.

Snapsets mit Grossbuchstaben A-Z werden erst ab BS2000/OSD-BC V8.0 unterstützt.

Die Snapset-Id der Bibliothek kann mit den Kommandos /LIST-FILE-FROM-SNAPSET und /SHOW-SNAPSET-CONFIGURATION ermittelt werden.

**SNAPSET = <integer -52..-1>**

Die Bibliothek liegt auf einem Snapset in der zeitlichen Reihenfolge der Snapset-Erzeugung mit -1 für den neuesten Snapset.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0211	Bibliothek existiert bereits
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt

**Notwendige Zugriffsrechte**

bei MODE=\*READ : Leserecht für LIBRARY

bei MODE=\*UPDATE : Lese- und Schreibrecht für LIBRARY

*Hinweis*

Wird die Anweisung mit LMS0024 und PLA0203 abgebrochen, so könnte ein altes Bibliotheksformat vorliegen. Wie dies geprüft und in eine PLAM-Bibliothek konvertiert werden kann, ist im Anhang im [Abschnitt „Migration alter Bibliotheksformate“](#) auf [Seite 526](#) beschrieben.

*Beispiele*

- Eröffnen einer existierenden Bibliothek lib1:

```
//open-library library=lib1
```

- Eröffnen einer existierenden Bibliothek über Dateikettungsname:

```
//add-file-link link-name=glob-lib,file-name=lib1
```

```
//start-lms
```

```
...
```

```
//open-library library=*link(link-name=glob-lib)
```

- Neuanlegen einer Bibliothek:

```
//open-library library=lib1new, mode=*update
```

- Eröffnen einer Bibliothek auf einem Snapset:

Die Bibliothek X liege auf dem Originalpubset und ein Snap von X auf dem Snapset a.

```
//open-lib x,snapset=a
```

Mit folgender Anweisung können die Elemente vom Snapset a auf den Originalpubset kopiert werden:

```
//copy-elem (*std,*),(x)
```

Im Gegensatz dazu kopiert die folgende Anweisung nur innerhalb des Originalpubsets:

```
//copy-elem (x,*),(y)
```



## PROVIDE-ELEMENT

### Elemente reservieren

PROVIDE-ELEMENT reserviert Elemente einer Quellbibliothek und stellt Kopien dieser Elemente in einer Ausgabebibliothek zur Verfügung. Die Quellbibliothek ist explizit zu benennen. Bei aktivem WRITE-CONTROL sind die reservierten Elemente vor anderweitiger Weiterentwicklung geschützt. Bei zusätzlich eingestellter Konvention ist der gesamte Versionsraum über der angegebenen Version für den Holder reserviert.

Der Basistyp für Quell- und Zielelement darf unterschiedlich sein, wenn Textelemente reserviert werden.

#### PROVIDE-ELEMENT

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

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

**LIBRARY** = <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = **\*ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(…)

**\*ALL**(...)

**VERSION** = **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
                    <composed-name 1..24 with-under>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

    <composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
                    <composed-name 1..24 with-under>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

**,TYPE** = **\*LMS-DEFAULT** / **\*ALL** / <alphanum-name 1..8 with-wild(20)>

(Teil 1 von 3)

```

,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    | *ANY(...)
    | |
    | | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    | | <composed-name 1..24 with-under>
    | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    | | <composed-name 1..64 with-under with-wild(132)>(…)
    | | |
    | | | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    | | | <composed-name 1..24 with-under>
    | | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    | | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
    | | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    | | *INTERVAL(...)
    | | |
    | | | FROM = 1900-01-01 / <date 8..10 with-compl>
    | | | ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)

```

,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 1900-01-01 / <date 8..10 with-compl>
        | ,TO = *TODAY / <date 8..10 with-compl>
,TO-ELEMENT = *LIBRARY-ELEMENT (...)
    *LIBRARY-ELEMENT(...)
        |
        | LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
        | *LINK(...)
        | | LINK-NAME = <structured-name 1..8>
,ELEMENT = *BY-SOURCE (...) / <composed-name 1..132 with-under with-wild-constr>(...)
    *BY-SOURCE(...)
        |
        | VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
        | | *UPPER-LIMIT / <composed-name 1..24 with-under>
        | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
        | <composed-name 1..132 with-under with-wildcard-constr>(...)
        | | VERSION = *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING / *INCREMENT /
        | | | *UPPER-LIMIT / <composed-name 1..24 with-under>
        | | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
,TYPE = *BY-SOURCE / *LMS-DEFAULT / <alphanum-name 1..20 with-wild-constr>
,USER-DATE = *BY-SOURCE / *TODAY / <date 8..10 with-compl>
,STORAGE-FORM = *LMS-DEFAULT / *STD / *FULL / *DELTA
,WRITE-MODE = *LMS-DEFAULT / *CREATE / *REPLACE / *ANY
,DIALOG-CONTROL = *LMS-DEFAULT / *NO / *YES / *ERROR

```

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich die bereitzustellenden Elemente befinden.

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

Name der Bibliothek

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek

**ELEMENT = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(...)**

Namen der Elemente, die bereitgestellt werden sollen.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version der bereitzustellenden Elemente.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird bereitgestellt.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird bereitgestellt.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ der bereitzustellenden Elemente.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Es werden Elemente unabhängig von diesem Datum bereitgestellt.

**USER-DATE = \*TODAY**

Nur Elemente mit dem aktuellen Tagesdatum werden bereitgestellt.

**USER-DATE = <date 8..10 with-compl>**

Nur Elemente mit dem angegebenen Datum werden bereitgestellt.

**USER-DATE = \*INTERVAL(...)**

Nur Elemente mit einem in dem angegebenen Zeitraum liegenden Datum werden bereitgestellt.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die ausgeschlossen werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

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

Angabe, wo und unter welchem Namen das Element bereitgestellt werden soll.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in die das Element kopiert werden soll.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in die das Element kopiert werden soll. Existiert die Bibliothek noch nicht, wird sie neu eingerichtet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek.

**ELEMENT = \*BY-SOURCE (...) /**

**<composed-name 1..132 with-under with-wildcard-constr>(…)**

Name, den das Zielelement erhalten soll.

**ELEMENT = \*BY-SOURCE(...)**

Das Zielelement erhält den Namen des Quellelementes.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, den das Zielelement erhalten soll.

**VERSION = \*BY-SOURCE**

Das Zielelement erhält die gleiche Version wie das Quellelement.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version, ansonsten eine Defaultversion erzeugt.

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

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

Das Zielelement erhält die hier angegebene Version.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest.

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name, den das Zielelement erhalten soll. Er kann auch mit Wildcards eingegeben werden.

**VERSION = \*LMS-DEFAULT / \*BY-SOURCE / \*HIGHEST-EXISTING / \*INCREMENT / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das Zielelement erhalten soll. Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest.

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT /**

**<alphanum-name 1..20 with-wild-constr>**

Typ, den das bereitgestellte Element erhalten soll.

**TYPE = \*BY-SOURCE**

Das Zielelement erhält die gleiche Typbezeichnung wie das Quellelement.

**USER-DATE = \*BY-SOURCE / \*TODAY / <date 8..10 with-compl>**

Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*BY-SOURCE**

Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = \*TODAY**

Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = <date 8..10 with-compl>**

Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**STORAGE-FORM = \*LMS-DEFAULT / \*STD / \*FULL / \*DELTA**

Speicherform für das anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*STD**

Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**

Das Element wird vollgespeichert angelegt (Fehler falls nicht zulässig).

**STORAGE-FORM = \*DELTA**

Das Element wird deltagespeichert angelegt (Fehler falls nicht zulässig). Die Angabe ist zulässig für die Elementtypen S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*ANY**

Steuerung des Überschreibverhaltens.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren, es wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*ANY**

Existiert das Zielelement wird dieses ersetzt, ansonsten wird das Zielelement neu erzeugt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt



### Notwendige Zugriffsrechte

Bei ELEMENT:            Leserecht für LIBRARY  
                              Lese- und Ausleihrecht für ELEMENT

Bei TO-ELEMENT:        Lese- und Schreibrecht für LIBRARY

Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).

Bei Deltaspeicherung muss für das durch BASE bestimmte Element Leserecht gegeben sein. Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Es kann nur eine neue Version erzeugt oder diese Basisversion überschrieben werden, sofern Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

#### *Hinweise*

- Beim Anlegen eines Elements ist die unter dem Typ geltende Konvention zu beachten. Insbesondere dann, wenn der Zieltyp die Konvention STD-TREE hat, können Probleme auftreten, wenn auf der Quellseite Nebenzweigversionen existieren, deren zugehörige Hauptzweigversion gelöscht ist. In diesem Fall können die betreffenden Nebenzweige nicht kopiert werden; es erfolgt eine Fehlermeldung von LMS.
- Bei aktivem WRITE-CONTROL in der Ausgabebibliothek wird dem zu schreibenden Element von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorganges protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Bereits vorhandene Kommentarsätze werden dahinter kopiert. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), so werden die Elementattribute STATE, HOLDER und alle Rechte von der Basisversion auf die neue Version übernommen. Die CCSN wird von dem Quellelement übernommen. Das USER-DATE wird neu bestimmt.
- Der Benutzer wird als HOLDER des Quellelementes eingetragen. Das Quellelement befindet sich im Status 'IN-HOLD'. PROVIDE-ELEMENT ist damit für dieses Element solange nicht mehr möglich, bis es sich wieder im Status FREE befindet. Dies kann durch RETURN-ELEMENT oder MODIFY-ELEMENT-ATTRIBUTES erfolgen.

## REORGANIZE-LIBRARY

### Bibliothek reorganisieren

REORGANIZE-LIBRARY reorganisiert eine Bibliothek derart, dass möglichst viel für Elemente nicht genutzter Plattenplatz am Ende der Bibliotheksdatei anfällt, der dann freigegeben wird. Der für eine Bibliothek erforderliche Plattenplatzbedarf kann damit vermindert werden.

#### REORGANIZE-LIBRARY

```
LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
```

#### LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)

Angabe der Bibliothek, die reorganisiert werden soll.

#### LIBRARY = \*STD

Die durch OPEN-LIBRARY global eröffnete Bibliothek wird reorganisiert.

#### LIBRARY = <filename 1..54 without-vers>

Die Bibliothek mit dem hier angegebenen Namen wird reorganisiert.

#### LIBRARY = \*LINK(...)

Die über den Dateikettungsamen zugewiesene Bibliothek wird reorganisiert.

#### LINK-NAME = <structured-name 1..8>

Dateikettungsname der Bibliothek, der mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt

### Notwendige Zugriffsrechte

Eigentümerschaft, Lese- und Schreibrecht für LIBRARY

*Hinweise*

- Während einer Reorganisation werden keine weiteren Bibliothekszugriffe zugelassen (Öffnungsmodus: SHARUPD=NO). Die Reorganisation wird daher abgewiesen, wenn die Bibliothek bereits für eine Bibliotheksanwendung geöffnet ist, z.B. durch eine andere Task. Wurde die zu reorganisierende Bibliothek zuvor im gleichen LMS-Lauf geöffnet, wird sie von LMS vor der Reorganisation implizit geschlossen, es sei denn, das LMS-Protokoll wird in diese Bibliothek geschrieben. Dann muss der Anwender selbst die Bibliothek durch Umweisen des Protokollstroms schließen.
- Der nach der Reorganisation noch belegte, aber nicht genutzte Plattenplatz kann so groß wie das größte in der Bibliothek vorhandene Element sein. Wird dieses Minimum nicht erreicht, z.B. weil nach Systemfehler hintere Blöcke zwar bereits als belegt markiert, aber noch nicht benutzt sind, oder soll die Bibliothek die absolute Minimalgröße erreichen, ist das Kopieren über Zwischenspeicher anzuwenden (COPY-LIBRARY).

## RESET-LMS-DEFAULTS

### Rücksetzen der LMS-DEFAULTS

Mit dieser Anweisung werden die mit MODIFY-LMS-DEFAULTS gesetzten Defaults auf die zu Beginn des LMS-Laufs geltenden Werte zurückgesetzt.

RESET-LMS-DEFAULTS

Die Anweisung RESET-LMS-DEFAULTS besitzt keine Operanden.

#### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

## RESET-LOGGING-PARAMETERS

### Rücksetzen der LMS-LOGGING-PARAMETERS

Mit dieser Anweisung werden die mit MODIFY-LOGGING-PARAMETERS gesetzten Parameter auf die zu Beginn des LMS-Laufs geltenden Werte zurückgesetzt.

RESET-LOGGING-PARAMETERS

Die Anweisung RESET-LOGGING-PARAMETERS besitzt keine Operanden.

#### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

## RESET-TYPE-ATTRIBUTES

### Rücksetzen der Typ-Attribute

Mit dieser Anweisung werden für den angegebenen Typ alle Typattribute auf die Defaultwerte der MODIFY-TYPE-ATTRIBUTES-Anweisung zurückgesetzt.

RESET-TYPE-ATTRIBUTES
<pre> <b>LIBRARY</b> = <u>*STD</u> / &lt;filename 1..54 without-vers&gt; / *LINK(...)   *LINK(...)         <b>LINK-NAME</b> = &lt;structured-name 1..8&gt; / &lt;filename 1..8&gt;   ,<b>TYPE</b> = &lt;alphanum-name 1..8&gt; </pre>

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der die Typattribute zurückgesetzt werden sollen.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der die Typattribute zurückgesetzt werden sollen.

**LIBRARY = \*LINK(..)**

Die über den Dateikettungsname zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

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

Elementtyp, dessen Attribute zurückgesetzt werden sollen.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0413	Typ gesperrt

## RETURN-ELEMENT

### Reservierung ausgeliehener Elemente zurückgeben

RETURN-ELEMENT kopiert Elemente einer Quellbibliothek in eine Ausgabebibliothek, sofern die zur Zielversion angegebene Basis in der Ausgabebibliothek vom Benutzer reserviert ist, löscht die Elemente in der Quellbibliothek und gibt die Reservierungen in der Ausgabebibliothek wieder zurück.

Zur ersten Version eines Elementes gibt es noch keine Basis. RETURN-ELEMENT erfordert in diesem Falle Administrationsrecht.

Der Basistyp für Quell- und Zielelement darf unterschiedlich sein, wenn Textelemente zurückgegeben werden.

Dem zu schreibenden Element wird von der Zugriffsmethode ein Kommentarsatz (Satzart 2) zugefügt, der HOLDER, DATE und TIME des Vorganges protokolliert. Der Satz wird als erster Satz der Satzart geschrieben. Dahinter wird der bei COMMENT angegebene Text als Kommentarsatz geschrieben. Bereits vorhandene Kommentarsätze werden dahinter kopiert. Wird das Element überdies zur Basis einer anderen Version geschrieben (also nicht erste Version unter einem Namen), so werden alle Rechte von der Basisversion auf die neue Version übernommen. STATE wird bei beiden Versionen auf FREE gesetzt. Die CCSN wird vom Quellelement übernommen. USER-DATE wird neu bestimmt.

#### RETURN-ELEMENT

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

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

**LIBRARY** = **\*STD** / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = **\*ALL**(...) / <composed-name 1..64 with-under with-wild(132)>(...)

**\*ALL**(...)

**VERSION** = **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(...)

**VERSION** = **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

(Teil 1 von 3)

```

,TYPE = LMS-DEFAULT / *ALL / <alphanum-name 1..8 with-wild(20)>
,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    FROM = 1900-01-01 / <date 8..10 with-compl>
    ,TO = *TODAY / <date 8..10 with-compl>
,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    FROM = 1900-01-01 / <date 8..10 with-compl>
    ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    FROM = 1900-01-01 / <date 8..10 with-compl>
    ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
      *ANY(...)
        VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
          <composed-name 1..24 with-under>
        ,BASE = *STD / <composed-name 1..24 with-under with-wild>
          <composed-name 1..64 with-under with-wild(132)>(…)
        VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
          <composed-name 1..24 with-under>
        ,BASE = *STD / <composed-name 1..24 with-under with-wild>
      ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
      ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
        *INTERVAL(...)
          FROM = 1900-01-01 / <date 8..10 with-compl>
          ,TO = *TODAY / <date 8..10 with-compl>

```

(Teil 2 von 3)



<pre> ,CREATION-DATE = <u>*ANY</u> / *TODAY / &lt;date 8..10 with-compl&gt; / *INTERVAL(...)     *INTERVAL(...)                     FROM = <u>1900-01-01</u> / &lt;date 8..10 with-compl&gt;           ,TO = *TODAY / &lt;date 8..10 with-compl&gt; ,MODIFICATION-DATE = <u>*ANY</u> / *TODAY / &lt;date 8..10 with-compl&gt; / *INTERVAL(...)     *INTERVAL(...)                     FROM = <u>1900-01-01</u> / &lt;date 8..10 with-compl&gt;           ,TO = *TODAY / &lt;date 8..10 with-compl&gt; ,TO-ELEMENT = *<u>LIBRARY-ELEMENT</u> (...)     *LIBRARY-ELEMENT(...)                     LIBRARY = &lt;filename 1..54 without-vers&gt; / *LINK(...)           *LINK(...)             LINK-NAME = &lt;structured-name 1..8&gt; ,ELEMENT = *<u>BY-SOURCE</u> (...) / &lt;composed-name 1..132 with-under with-wild-constr&gt;(…)     *BY-SOURCE(...)                     VERSION = *<u>INCREMENT</u> / *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING /             *UPPER-LIMIT / &lt;composed-name 1..24 with-under&gt;           ,BASE = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt;           &lt;composed-name 1..132 with-under with-wildcard-constr&gt;(…)             VERSION = *<u>INCREMENT</u> / *LMS-DEFAULT / *BY-SOURCE / *HIGHEST-EXISTING /               *UPPER-LIMIT / &lt;composed-name 1..24 with-under&gt;             ,BASE = *<u>STD</u> / &lt;composed-name 1..24 with-under with-wild&gt; ,TYPE = *<u>BY-SOURCE</u> / *LMS-DEFAULT / &lt;alphanum-name 1..20 with-wild-constr&gt; ,USER-DATE = *<u>BY-SOURCE</u> / *TODAY / &lt;date 8..10 with-compl&gt; ,STORAGE-FORM = *<u>LMS-DEFAULT</u> / *STD / *FULL / *DELTA ,COMMENT = *<u>BY-SOURCE</u> / &lt;c-string 1..72 with-low&gt; ,WRITE-MODE = *<u>LMS-DEFAULT</u> / *CREATE / *REPLACE / *ANY ,DIALOG-CONTROL = *<u>LMS-DEFAULT</u> / *NO / *YES / *ERROR </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Teil 3 von 3)

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in der sich die betroffenen Elemente befinden.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek.

**ELEMENT = \*ALL / <composed-name 1..64 with-under with-wild(132)>(...)**

Namen der Elemente, die zurückgegeben werden sollen.

**VERSION = \*HIGHEST-EXISTING / \*UPPER-LIMIT /****<composed-name 1..24 with-under>**

Version der zurückzugebenden Elemente.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird zurückgegeben.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird zurückgegeben.

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

Der hier angegebene Text wird als Versionsbezeichnung interpretiert.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*.

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ der zurückzugebenden Elemente.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Es werden Elemente unabhängig von diesem Datum bereitgestellt.

**USER-DATE = \*TODAY**

Nur Elemente mit dem aktuellen Tagesdatum werden zurückgegeben.

**USER-DATE = <date 8..10 with-compl>**

Nur Elemente mit dem angegebenen Datum werden zurückgegeben.

**USER-DATE = \*INTERVAL(...)**

Nur Elemente mit einem in dem angegebenen Zeitraum liegenden Datum werden zurückgegeben.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden siehe Operand USER-DATE dieser Anweisung.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die ausgeschlossen werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

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

Angabe, wohin und unter welchem Namen das Element kopiert werden soll.

**LIBRARY = <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in die das Element kopiert werden soll.

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

Name der Bibliothek, in die das Element zurückgegeben werden soll. Existiert die Bibliothek noch nicht, wird sie neu eingerichtet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek.

**ELEMENT = \*BY-SOURCE(...) /**

**<composed-name 1..132 with-under with-wild-constr>(…)**

Name, den das Zielelement erhalten soll.

**ELEMENT = \*BY-SOURCE(…)**

Das Zielelement erhält den Namen des Quellelementes.

**VERSION = \*INCREMENT / \*LMS-DEFAULT / \*BY-SOURCE /**

**\*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das Zielelement erhalten soll.

**VERSION = \*INCREMENT**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens eine neue, höhere Version ansonsten eine Defaultversion erzeugt.

**VERSION = \*BY-SOURCE**

Das Zielelement erhält die gleiche Version wie das Quellelement.

**VERSION = \*HIGHEST-EXISTING**

Abhängig von der unter dem Typ geltenden Konvention wird unter existierenden Elementen gleichen Typs und Namens die höchste zu BASE passende Version überschrieben, ansonsten wird eine Defaultversion erzeugt.

**VERSION = \*UPPER-LIMIT**

Die höchstmögliche Version X'FF' wird erzeugt.

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

Das Zielelement erhält die hier angegebene Version.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest. Bei BASE=\*STD ist die vom Benutzer ausgeliehene Version Basisversion (Fehler bei Mehrdeutigkeit).

**ELEMENT = <composed-name 1..132 with-under with-wild-constr>(…)**

Name, den das Zielelement erhalten soll. Er kann auch mit Wildcards eingegeben werden.

**VERSION =\*INCREMENT / \*LMS-DEFAULT / \*BY-SOURCE /**

**\*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under>**

Version, die das Zielelement erhalten soll. Beschreibung der Operanden siehe oben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Legt die Basis für das Zielelement fest.

**TYPE = \*BY-SOURCE / \*LMS-DEFAULT / <alphanum-name 1..20 with-wildcard-constr>**  
Typ, den das Zielelement erhalten soll.

**TYPE = \*BY-SOURCE**  
Das Zielelement erhält die gleiche Typbezeichnung wie das Quellelement.

**USER-DATE = \*BY-SOURCE / \*TODAY / <date 8..10 with-compl>**  
Datum, das vom Benutzer vergeben wird.

**USER-DATE = \*BY-SOURCE**  
Das neue Element erhält das gleiche Datum wie das ursprüngliche Element.

**USER-DATE = \*TODAY**  
Das aktuelle Tagesdatum wird vergeben.

**USER-DATE = <date 8..10 with-compl>**  
Das Datum muss in der Form [JJ]JJ-MM-TT eingegeben werden.

**STORAGE-FORM = \*LMS-DEFAULT / \*STD / \*FULL / \*DELTA**  
Speicherform für das anzulegende Element. Die Speicherform darf den Einstellungen durch MODIFY-TYPE-ATTRIBUTES bzw. MODIFY-LIBRARY-ATTRIBUTES nicht widersprechen und alle Elemente eines Typs und Namens müssen die gleiche Speicherform haben.

**STORAGE-FORM = \*STD**  
Das Element wird gemäß der für den Elementscope geforderten Speicherform angelegt. Widersprüchliche Anforderungen führen auf Fehler. Ist nichts besonderes gefordert, wird Vollspeicherung gewählt.

**STORAGE-FORM = \*FULL**  
Das Element wird vollgespeichert angelegt (Fehler falls nicht zulässig).

**STORAGE-FORM = \*DELTA**  
Das Element wird deltagespeichert angelegt (Fehler falls nicht zulässig). Die Angabe ist zulässig für die Elementtypen S, P, D, J, M, X bzw. davon abgeleitete Elementtypen.

**COMMENT = \*BY-SOURCE / <c-string1..72 with-low>**  
Steuert die Angabe eines Kommentartextes.

**COMMENT = \*BY-SOURCE**  
Der Kommentartext wird dem Quellelement entnommen.

**COMMENT = <c-string 1..72 with-low>**  
Kommentartext, der in das Zielelement eingefügt wird. Der Kommentartext der Quelle bleibt zusätzlich erhalten.

**WRITE-MODE = \*LMS-DEFAULT / \*CREATE / \*REPLACE / \*ANY**

Steuerung des Überschreibverhaltens.

**WRITE-MODE = \*CREATE**

Das Zielelement darf noch nicht existieren; es wird neu erzeugt.

**WRITE-MODE = \*REPLACE**

Das Zielelement muss bereits existieren und wird dann ersetzt.

**WRITE-MODE = \*ANY**

Existiert das Zielelement, wird es ersetzt, ansonsten wird das Zielelement neu erzeugt.

**DIALOG-CONTROL = \*LMS-DEFAULT / \*NO / \*YES / \*ERROR**

Dieser Operand legt fest, ob während der Ausführung einer Anweisung ein Dialog mit dem Benutzer geführt werden soll oder nicht.

Nähere Erläuterungen zur Dialogführung unter MODIFY-LMS-DEFAULTS.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0020	Ziel-Element oder Ziel-Datei nicht vorhanden
	64	LMS0213	Name existiert als Delta-Element
	64	LMS0214	Name existiert als Voll-Element
	64	LMS0302	Element nicht gefunden
	64	LMS0509	Ziel-Element oder Ziel-Datei schon vorhanden
	64	LMS0510	Basis nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0223	Nur Blätter eines Delta-Baumes überschreibbar
	64	PLA0224	Speicherform nicht erlaubt
	64	PLA0229	Kein Zugriffsrecht für das Element
	64	PLA0233	Ausleihstatus verhindert Elementzugriff
	64	PLA0475	Funktion verletzt Versionsautomatismus
	64	PLA0476	Version passt nicht zur geltenden Konvention
	64	PLA0478	Versionsüberlauf bei Erhöhung
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei ELEMENT: Leserecht für LIBRARY und ELEMENT

Bei TO-ELEMENT: Lese- und Schreibrecht für LIBRARY

Administrationsrecht, sofern die angegebene Elementbezeichnung neu ist. Ansonsten nur Schreibrecht für das unter der angegebenen Elementbezeichnung existierende Element (kein Administrationsrecht mehr erforderlich).

Bei Deltaspeicherung muss für das durch BASE bestimmte Element Leserecht gegeben sein.

Bei aktivem WRITE-CONTROL muss die USERID des Benutzers bei dem durch BASE bestimmten Element als HOLDER eingetragen sein, sofern eine Basisversion existiert. Es kann nur eine neue Version erzeugt oder diese Basisversion überschrieben werden, sofern Schreibrecht gegeben ist. Administrationsrecht ist in diesem Falle nicht mehr erforderlich.

*Hinweis*

Beim Anlegen eines Elements ist die unter dem Typ geltende Konvention zu beachten. Insbesondere dann, wenn der Zieltyp die Konvention STD-TREE hat, können Probleme auftreten, wenn auf der Quellseite Nebenzweigversionen existieren, deren zugehörige Hauptzweigversion gelöscht ist. In diesem Fall können die betreffenden Nebenzweige nicht kopiert werden; es erfolgt eine Fehlermeldung von LMS.

## SHOW-ELEMENT

### Anzeigen des Elementinhalts

SHOW-ELEMENT zeigt den Elementinhalt abhängig vom Typ des angegebenen Elementes. Es können die Inhalte von textartigen Elementen, Moduln, Phasen und Bindelademoduln ausgegeben werden. Das Darstellungsformat der Ausgabe wird über den Operanden OUTPUT-FORM gesteuert. Die Bedeutung der Attribute bei Moduln und Bindelademoduln ist [5] zu entnehmen.

Die Anweisung ist für alle Elementtypen zulässig. Benutzertypen werden dabei entsprechend ihrem Basistyp behandelt. Ist der Basistyp LMS nicht bekannt, so wirken nur die Operanden TEXT-INFORMATION und OUTPUT-FORM.

#### SHOW-ELEMENT

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

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

**LIBRARY** = \*STD / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = \*ALL(...) / <composed-name 1..64 with-under with-wild(132)>(…)

**\*ALL**(...)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = \*HIGHEST-EXISTING / **\*ALL** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = \*STD / <composed-name 1..24 with-under with-wild>

**,TYPE** = \*LMS-DEFAULT / **\*ALL** / <alphanum-name 1..8 with-wild(20)>

**,USER-DATE** = \*ANY / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)

**\*INTERVAL**(...)

**FROM** = **1900-01-01** / <date 8..10 with-compl>

**,TO** = \*TODAY / <date 8..10 with-compl>

(Teil 1 von 5)



```

,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,EXCEPT-ELEMENT = *NONE / *ELEMENT(...)
  *ELEMENT(...)
    |
    | ELEMENT = *ANY (...) / <composed-name 1..64 with-under with-wild(132)>(…)
    | *ANY(...)
    |   |
    |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   <composed-name 1..24 with-under with-wild(52)>
    |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | <composed-name 1..64 with-under with-wild(132)>(…)
    |   |   |
    |   |   | VERSION = *ANY / *HIGHEST-EXISTING / *UPPER-LIMIT /
    |   |   |   <composed-name 1..24 with-under with-wild(52)>
    |   |   | ,BASE = *STD / <composed-name 1..24 with-under with-wild>
    |   | ,TYPE = *ANY / *LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>
    |   | ,USER-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>
    |   | ,CREATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>
    |   | ,MODIFICATION-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
    |   |   *INTERVAL(...)
    |   |     |
    |   |     | FROM = 1900-01-01 / <date 8..10 with-compl>
    |   |     | ,TO = *TODAY / <date 8..10 with-compl>

```

```

,TEXT-INFORMATION = *LMS-DEFAULT / *ALL / *STATISTICS / *FILE-ATTRIBUTES / *PARAMETERS(...)
  *PARAMETERS(...)
    INFORMATION = *LMS-DEFAULT / *ALL / list-poss(2): *TEXT / *COMMENT
  ,RECORD-RANGE = *LMS-DEFAULT / *ALL / *RANGE(...)
    *RANGE(...)
      FROM = *LMS-DEFAULT / <integer 1..2147483647>
      ,TO = *LMS-DEFAULT / *LAST / <integer 1..2147483647>
  ,RECORD-PART = *LMS-DEFAULT / *ALL / *PART(...)
    *PART(...)
      START = *LMS-DEFAULT / <integer 1..262144>
      ,LENGTH = *LMS-DEFAULT / *REST / <integer 0..262144>
  ,RECORD-NUMBER = *LMS-DEFAULT / *BY-OUTPUT / *YES / *NO
,MODULE-INFORMATION = *LMS-DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
  *PARAMETERS(...)
    INFORMATION = *LMS-DEFAULT / *ALL / *TXT(...) / *TXTP(...) /
      list-poss(9): *ESD / *ISD / *LSD / *RLD / *REP / *INCLUDE / *DSDD / *REF / *END
    *TXT(...)
      CSECT-NAME = *LMS-DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>
      ,ADDRESS = *LMS-DEFAULT (...) / <x-string 1..8>(...)
        *LMS-DEFAULT(...)
          BASE-ADDRESS = *LMS-DEFAULT / <x-string 1..8>
          <x-string 1..8>(...)
            BASE-ADDRESS = *LMS-DEFAULT / <x-string 1..8>
          ,LENGTH = *LMS-DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
    *TXTP(...)
      MODIFICATION-ID = *LMS-DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)
        *RANGE(...)
          FROM = *LMS-DEFAULT / *LOWEST / <c-string 1..8 with-low>
          ,TO = *LMS-DEFAULT / *HIGHEST / <c-string 1..8 with-low>

```

(Teil 3 von 5)

```

,PHASE-INFORMATION = *LMS-DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | SEGMENT = *LMS-DEFAULT / *ALL / *ROOT / <name 1..8>
    | ,INFORMATION = *LMS-DEFAULT / *ALL / *TXT(...) / *TXTP(...) /
    | list-poss(4): *ESD / *ISD / *LSD / *RLD
    |
    | *TXT(...)
    |   |
    |   | ADDRESS = *LMS-DEFAULT (...) / <x-string 1..8>(…)
    |   | *LMS-DEFAULT(...)
    |   | | BASE-ADDRESS = *LMS-DEFAULT / <x-string 1..8>
    |   | | <x-string 1..8>(…)
    |   | | BASE-ADDRESS = *LMS-DEFAULT / <x-string 1..8>
    |   | ,LENGTH = *LMS-DEFAULT / *REST / <integer 1..2147483647> / <x-string 1..8>
    |   |
    |   | *TXTP(...)
    |   |   |
    |   |   | MODIFICATION-ID = *LMS-DEFAULT / *ALL / <c-string 1..8 with-low> / *RANGE(...)
    |   |   | *RANGE(...)
    |   |   | | FROM = *LMS-DEFAULT / *LOWEST / <c-string 1..8 with-low>
    |   |   | | ,TO = *LMS-DEFAULT / *HIGHEST / <c-string 1..8 with-low>
    |   |
    |   | ,LLM-INFORMATION = *LMS-DEFAULT / *ALL / *STATISTICS / *PARAMETERS(...)
    |   |   *PARAMETERS(...)
    |   |     |
    |   |     | LLM-PART = *LMS-DEFAULT / *ALL / *SLICE(...) / *SUB-LLM(...)
    |   |     | *SLICE(...)
    |   |     | | NAME = *LMS-DEFAULT / <structured-name 1..32>
    |   |     | *SUB-LLM(...)
    |   |     | | PATH-NAME = *LMS-DEFAULT / <c-string 1..255 with-low> / <text 1..255>
    |   |     | ,INFORMATION = *LMS-DEFAULT / *ALL / *TXT(...) / *TXTP(...) / *LOGICAL(...) / *PHYSICAL /
    |   |     | *REF / list-poss(4): / *RELOCATION / *ESVD / *ESVR / *LRLD
    |   |     | *TXT(...)
    |   |     | | CSECT-NAME = *LMS-DEFAULT / *ALL / <c-string 1..32 with-low> / <text 1..32>

```

(Teil 4 von 5)

<pre> ,ADDRESS = *<u>LMS-DEFAULT</u> (...) / &lt;x-string 1..8&gt;(…)   *<u>LMS-DEFAULT</u>(…)         <b>BASE-ADDRESS</b> = *<u>LMS-DEFAULT</u> / &lt;x-string 1..8&gt;         &lt;x-string 1..8&gt;(…)         <b>BASE-ADDRESS</b> = *<u>LMS-DEFAULT</u> / &lt;x-string 1..8&gt; ,LENGTH = *<u>LMS-DEFAULT</u> / *<u>REST</u> / &lt;integer 1..2147483647&gt; / &lt;x-string 1..8&gt;  *TXTP(…)       <b>CSECT-NAME</b> = *<u>LMS-DEFAULT</u> / *<u>ALL</u> / &lt;c-string 1..32 with-low&gt; / &lt;text 1..32&gt; ,MODIFICATION-ID = *<u>LMS-DEFAULT</u> / *<u>ALL</u> / &lt;c-string 1..12 with-low&gt; / *<u>RANGE</u>(…)   *<u>RANGE</u>(…)         <b>FROM</b> = *<u>LMS-DEFAULT</u> / *<u>LOWEST</u> / &lt;c-string 1..12 with-low&gt;         <b>TO</b> = *<u>LMS-DEFAULT</u> / *<u>HIGHEST</u> / &lt;c-string 1..12 with-low&gt;  *LOGICAL(…)       <b>LEVEL</b> = *<u>LMS-DEFAULT</u> / *<u>ALL</u> / *<u>NEXT</u>  ,OUTPUT-FORM = *<u>LMS-DEFAULT</u> / *<u>STD</u> / *<u>CHARACTER</u> / *<u>HEXADECIMAL</u> / *<u>DUMP</u> ,TEXT-OUTPUT = *<u>LOGGING-PARAMETERS</u> / *<u>SYSOUT</u> / *<u>SYSLST</u>(…) / *<u>EDT</u>(…)   *<u>SYSLST</u>(…)         <b>SYSLST-NUMBER</b> = *<u>STD</u> / &lt;INTEGER 1..99&gt;   *<u>EDT</u>(…)         <b>WRITE-MODE</b> = *<u>EXTEND</u> / *<u>REPLACE</u> </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(Teil 5 von 5)

**ELEMENT = \*LIBRARY-ELEMENT(…)**

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(…)**

Angabe der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Name der Bibliothek, in der sich das Element befindet.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsnamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL / <composed-name 1..64 with-under with-wild(132)>(…)**

Name des Elementes, welches angezeigt werden soll.

**VERSION = \*HIGHEST-EXISTING / \*ALL / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das auszugebende Element besitzt.

**VERSION = \*HIGHEST-EXISTING**

Das Element mit der höchsten bestehenden Version bzgl. BASE wird ausgegeben.

**VERSION = \*UPPER-LIMIT**

Die in der Bibliothek unter dem angegebenen TYPE und Namen höchstmögliche Version X'FF' wird angezeigt.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Explizite Angabe der Version des Elementes, das angezeigt werden soll.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Typ, den das auszugebende Element besitzt.

**USER-DATE = \*ANY / \*TODAY / <date8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das auszugebende Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Das Element mit dem aktuellen Tagesdatum wird angezeigt.

**USER-DATE = <date 8..10 with-compl>**

Das Element, dessen Datum explizit in der Form [JJ]JJ-MM-TT eingegeben wird, wird angezeigt.

**USER-DATE = \*INTERVAL(...)**

Alle Elemente, die in dem angegebenen Zeitraum liegen, werden angezeigt.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden analog zu USER-DATE.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden analog zu USER-DATE.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die nicht angezeigt werden sollen. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen. Zur Beschreibung der Operanden siehe Operand \*LIBRARY-ELEMENT dieser Anweisung.

**TEXT-INFORMATION = \*LMS-DEFAULT / \*ALL / \*STATISTICS / \*FILE-ATTRIBUTES / \*PARAMETERS(...)**

Legt die Informationsmenge für alle Elemente außer den Elementtypen R, C und L fest.

**TEXT-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**TEXT-INFORMATION = \*STATISTICS**

Es wird die Anzahl der Sätze pro Satzart und die Summe der Sätze ausgegeben. Pro Satzart wird die Summe der Satzlängen (ohne Satzlängfelder) ausgegeben und deren Summe über alle Satzarten.

**TEXT-INFORMATION = \*FILE-ATTRIBUTES**

Es werden nur die gespeicherten Dateiattribute ausgegeben.

**TEXT-INFORMATION = \*PARAMETERS(...)**

Es wird ein Elementausschnitt festgelegt, der ausgegeben werden soll.

**INFORMATION = \*LMS-DEFAULT / \*ALL / list-poss(2): \*TEXT / \*COMMENT**

Der anzuzeigende Teilbereich des Elementes

**INFORMATION = \*ALL**

Es wird alles ausgegeben.

**INFORMATION = \*TEXT**

Der eigentliche Text, d.h. Satzart 1 wird angezeigt.

**INFORMATION = \*COMMENT**

Der separat gespeicherte Text, d.h. Satzart 2 wird angezeigt.

**RECORD-RANGE = \*LMS-DEFAULT / \*ALL / \*RANGE(...)**

Der zu bearbeitende Teilbereich des Elementes.

**RECORD-RANGE = \*ALL**

Es werden alle Sätze bearbeitet.

**RECORD-RANGE = \*RANGE(...)**

Der zu bearbeitende Satznummernbereich wird bestimmt. Die Satznummern beziehen sich nicht auf eine Satzart, sondern auf den durch INFORMATION= festgelegten Teilbereich des Elementes. Innerhalb dieses Teilbereiches werden die Sätze von 1 bis n durchnummeriert.

**FROM = \*LMS-DEFAULT / <integer 1..2147483647>**

Beginn des Bereichs durch Angabe der ersten Satznummer. Standardmäßig wird die Satznummer 1 genommen.

**TO = \*LMS-DEFAULT / \*LAST / <integer 1..2147483647>**

Ende des Bereichs durch Angabe der letzten Satznummer. Standardmäßig wird die letzte Satznummer genommen.

**RECORD-PART = \*LMS-DEFAULT / \*ALL / \*PART(...)**

Der zu bearbeitende Satzbereich wird bestimmt.

**RECORD-PART = \*ALL**

Der ganze Satz wird bearbeitet.

**RECORD-PART = \*PART(...)**

Der zu bearbeitende Satzbereich wird bestimmt. Werden die Standardwerte nicht verändert, wird der ganze Satz bearbeitet.

**START = \*LMS-DEFAULT / <integer 1..262144>**

Beginn des Bereichs durch Angabe des ersten Zeichens im Satz. Standardmäßig wird das erste Zeichen genommen.

**LENGTH = \*LMS-DEFAULT / \*REST / <integer 0..262144>**

Länge des Bereichs. Standardmäßig wird der Rest des Satzes genommen.

**RECORD-NUMBER = \*LMS-DEFAULT / \*BY-OUTPUT / \*YES / \*NO**

Bestimmt die Ausgabe der Satznummern.

**RECORD-NUMBER = \*BY-OUTPUT**

Nur wenn die Ausgabe nach SYSOUT erfolgt, werden keine Satznummern ausgegeben. Bei jedem anderen Ausgabemedium werden sie mit ausgegeben.

**RECORD-NUMBER = \*YES**

Die Satznummern werden auch nach SYSOUT ausgegeben.

**RECORD-NUMBER = \*NO**

Es werden keine Satznummern mit ausgegeben.

**MODULE-INFORMATION = \*LMS-DEFAULT / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Bindemodule (Elemente vom Typ R) fest.

**MODULE-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**MODULE-INFORMATION = \*STATISTICS**

Es werden Name, Länge und Adresse der CSECTS sowie die Gesamtlänge des Moduls ausgegeben.

**MODULE-INFORMATION = \*PARAMETERS(...)**

Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

**INFORMATION = \*LMS-DEFAULT / \*ALL / \*TXT(...) / \*TXTP(...) / list-poss(9): \*ESD / \*ISD / \*LSD / \*RLD / \*REP / \*INCLUDE / \*DSDD / \*REF / \*END**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Es werden Textsätze ausgewählt.

**CSECT-NAME = \*LMS-DEFAULT / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die Textsätze können auf eine CSECT eingeschränkt werden.

**ADDRESS = \*LMS-DEFAULT / <x-string 1..8>(...**

Anfangsadresse des Textes. Die Standardeinstellung ist X'00000000'.

**BASE-ADDRESS = \*LMS-DEFAULT / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert. Die Standardeinstellung ist X'00000000'.

**LENGTH = \*LMS-DEFAULT / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes.

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgegeben.

**MODIFICATION-ID = \*LMS-DEFAULT / \*ALL / <c-string 1..8 with-low> / \*RANGE(...)**

Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.



**MODIFICATION-ID = \*RANGE(...)**

Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

**FROM = \*LMS-DEFAULT / \*LOWEST / <c-string 1..8 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**TO = \*LMS-DEFAULT / \*HIGHEST / <c-string 1..8 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**PHASE-INFORMATION = \*LMS-DEFAULT / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Phasen (Elemente vom Typ C) fest.

**PHASE-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**PHASE-INFORMATION = \*STATISTICS**

Es werden Name, Länge und Adresse des Segmentes sowie die Gesamtlänge des Segmentes ausgegeben.

**PHASE-INFORMATION = \*PARAMETERS(...)**

Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

**SEGMENT = \*LMS-DEFAULT / \*ALL / \*ROOT / <name 1..8>**

Phasensegment, das ausgewählt wird.

**INFORMATION = \*LMS-DEFAULT / \*ALL / \*TXT(...) / \*TXTP(...) /****list-poss(4): \*ESD / \*ISD / \*LSD / \*RLD**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Es werden Textsätze ausgewählt.

**ADDRESS = \*LMS-DEFAULT(...) / <x-string 1..8>(...**

Anfangsadresse des Textes

**BASE-ADDRESS = \*LMS-DEFAULT / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

**LENGTH = \*LMS-DEFAULT / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes.

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgegeben.

**MODIFICATION-ID = \*LMS-DEFAULT / \*ALL / <c-string 1..8 with-low> / \*RANGE(...)**

Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.

**MODIFICATION-ID = \*RANGE(...)**

Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

**FROM = \*LMS-DEFAULT / \*LOWEST / <c-string 1..8 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**TO= \*LMS-DEFAULT / \*HIGHEST / <c-string 1..8 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**LLM-INFORMATION = \*LMS-DEFAULT / \*ALL / \*STATISTICS / \*PARAMETERS(...)**

Legt die Informationsmenge für Bindelademodule (Elemente vom Typ L) fest.

**LLM-INFORMATION = \*ALL**

Es wird alles ausgegeben.

**LLM-INFORMATION =\*STATISTICS**

Allgemeine Informationen über das Bindelademodul (Name, Copyright, ...) werden ausgegeben.

**LLM-INFORMATION = \*PARAMETERS(...)**

Dieser Parameter legt fest, ob alle Satzarten oder nur ausgewählte Satzarten ausgegeben werden.

**LLM-PART = \*LMS-DEFAULT / \*ALL / \*SLICE(...) / \*SUB-LLM(...)**

Angabe des LLM-Teils, das ausgewählt werden soll. Standardmäßig wird der gesamte LLM ausgewählt.

**LLM-PART = \*SLICE(...)**

Angabe der SLICE, die ausgegeben werden soll.

**NAME = \*LMS-DEFAULT / <structured-name 1..32>**

Name der SLICE, die ausgegeben werden soll.

**LLM-PART = \*SUB-LLM(...)**

Angabe des SUB-LLM, der ausgegeben werden soll.

**PATH-NAME = \*LMS-DEFAULT / <c-string 1..255 with-low> / <text 1..255>**

Der auszugebende SUB-LLM wird durch seinen Pfadnamen bestimmt.

**INFORMATION = \*LMS-DEFAULT / \*ALL / \*TXT(...) / \*TXTP(...) / \*LOGICAL(...) / \*PHYSICAL / \*REF / list-poss(4): \*RELOCATION / \*ESVD / \*ESVR / \*LRLD**

Die hier aufgezählten Satzarten können ausgewählt werden.

**INFORMATION = \*TXT(...)**

Es werden Textsätze ausgewählt.

**CSECT-NAME = \*LMS-DEFAULT / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die Textsätze können auf eine CSECT eingeschränkt werden.

**ADDRESS = \*LMS-DEFAULT(...) / <x-string 1..8>(...**

Anfangsadresse des Textes

**BASE-ADDRESS = \*LMS-DEFAULT / <x-string 1..8>**

Die hier angegebene Basisadresse wird auf die Anfangsadresse addiert.

**LENGTH = \*LMS-DEFAULT / \*REST / <integer 1..2147483647> / <x-string 1..8>**

Länge des Textes

**INFORMATION = \*TXTP(...)**

TXTP-Sätze ausgegeben.

**CSECT-NAME = \*LMS-DEFAULT / \*ALL / <c-string 1..32 with-low> / <text 1..32>**

Die TXTP-Sätze können auf eine CSECT eingeschränkt werden.

**MODIFICATION-ID = \*LMS-DEFAULT / \*ALL / <c-string 1..12 with-low> /**

**\*RANGE(...)**

Es werden die TXTP-Sätze mit der angegebenen Identifikation ausgewählt.

**MODIFICATION-ID = \*RANGE(...)**

Es können mehrere TXTP-Sätze ausgewählt werden, die in einem Bereich liegen.

**FROM = \*LMS-DEFAULT / \*LOWEST / <c-string 1..12 with-low>**

Der Bereichsbeginn ist standardmäßig die niedrigste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**TO = \*LMS-DEFAULT / \*HIGHEST / <c-string 1..12 with-low>**

Das Bereichsende ist standardmäßig die höchste Identifikation für die TXTP-Sätze oder sonst die hier eingegebene.

**INFORMATION = \*LOGICAL(...)**

Es wird die logische Struktur des LLM ausgegeben.

**LEVEL = \*LMS-DEFAULT / \*ALL / \*NEXT**

Es werden standardmäßig alle oder sonst nur die nächste Unterstruktur ausgegeben.

**INFORMATION = \*PHYSICAL**

Es wird die physikalische Struktur des LLM ausgegeben.

**OUTPUT-FORM = \*LMS-DEFAULT / \*STD / \*CHARACTER / \*HEXADECIMAL / \*DUMP**

Festlegung der Darstellungsformates für die Ausgabe.

**OUTPUT-FORM = \*STD**

Das Darstellungsformat wird abhängig vom Typ des Elementes gewählt. Für Textelemente wirkt dieser Operand wie OUTPUT-FORM = \*CHARACTER.

**OUTPUT-FORM = \*CHARACTER**

Die Ausgabe wird alphanumerisch dargestellt.

**OUTPUT-FORM = \*HEXADECIMAL**

Die Ausgabe wird hexadezimal dargestellt.

**OUTPUT-FORM = \*DUMP**

Die Ausgabe wird alphanumerisch und hexadezimal dargestellt.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0084	VTSUCB-Makro-Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	64	PLA0229	Kein Zugriffsrecht für das Element
	130	LMS0041	Systemadressraum erschöpft
	130	LMS0411	Bibliothek gesperrt
	130	LMS0412	Element gesperrt
	130	LMS0413	Typ gesperrt

**Notwendige Zugriffsrechte**

Bei LIBRARY-ELEMENT: Leserecht für LIBRARY und ELEMENT

Sind von der Anweisung mehrere Elemente betroffen, so werden Elemente ohne Leserecht von der Anweisung ausgenommen.

*Beispiel*

Das Element LETTER.A, das den Text 'Sehr geehrte ...' enthält, soll aufgelistet werden.

```
//show-element (element=letter.a,type=d)
INPUT  LIBRARY= :10SQ:$USER.LIB.SHOW
INPUT  ELEMENT= (D)LETTER.A/@(0001)/2011-04-09
        SEHR GEEHRTE ...
```

```
NUMBER OF PROCESSED RECORDS IS      1
```

## SHOW-ELEMENT-ATTRIBUTES

### Auflisten des Inhaltsverzeichnisses einer Bibliothek

SHOW-ELEMENT-ATTRIBUTES gibt die Inhaltsverzeichniseinträge der angegebenen Elemente oder der gesamten Bibliothek aus. Die Ausgabe erfolgt auf das durch MODIFY-LOGGING-PARAMETERS eingestellte Ausgabemedium. Durch den Operanden STRUCTURE-OUTPUT kann zusätzlich die Ausgabe in eine strukturierte S-Variable (Liste) eingeschaltet werden.

Das Inhaltsverzeichnis wird immer nach dem Typ sortiert ausgegeben. Die weitere Sortierreihenfolge bestimmt der Operand SORT. Standardmäßig gilt die Sortierreihenfolge Typ, Name und Version.

Durch die Operanden INFORMATION und LAYOUT wird der Umfang und das Format der Ausgabe des Inhaltsverzeichnisses festgelegt. Standardmäßig werden Typ, Name, Version, Variantennummer und Datum ausgegeben.

Mit Hilfe der Operanden SECONDARY-NAME und -ATTRIBUTE kann das Inhaltsverzeichnis auf die Elemente beschränkt werden, die einen bestimmten Referenzeintrag enthalten.

#### *Hinweis*

Um den gesamten Inhalt einer Bibliothek zu erhalten (alle Elemente mit allen Versionen), genügt die Angabe von SHOW-ELEMENT-ATTRIBUTES ohne Angabe von Operanden, sofern über MODIFY-LMS-DEFAULTS kein bestimmter Elementtyp bzw. keine bestimmte Elementversion festgelegt wurde.

## SHOW-ELEMENT-ATTRIBUTES

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

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

**LIBRARY** = **\*STD** / <filename 1..54 without-vers> / **\*LINK**(...)

**\*LINK**(...)

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

**,ELEMENT** = **\*ALL** (...) / <composed-name 1..64 with-under with-wild(132)>(…)

**\*ALL**(...)

**VERSION** = **\*ALL** / **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

<composed-name 1..64 with-under with-wild(132)>(…)

**VERSION** = **\*ALL** / **\*HIGHEST-EXISTING** / **\*UPPER-LIMIT** /  
<composed-name 1..24 with-under with-wild(52)>

**,BASE** = **\*STD** / <composed-name 1..24 with-under with-wild>

**,TYPE** = **\*LMS-DEFAULT** / **\*ALL** / <alphanum-name 1..8 with-wild(20)>

**,USER-DATE** = **\*ANY** / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)

**\*INTERVAL**(...)

**FROM** = **1900-01-01** / <date 8..10 with-compl>

**,TO** = **\*TODAY** / <date 8..10 with-compl>

**,CREATION-DATE** = **\*ANY** / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)

**\*INTERVAL**(...)

**FROM** = **1900-01-01** / <date 8..10 with-compl>

**,TO** = **\*TODAY** / <date 8..10 with-compl>

**,MODIFICATION-DATE** = **\*ANY** / **\*TODAY** / <date 8..10 with-compl> / **\*INTERVAL**(...)

**\*INTERVAL**(...)

**FROM** = **1900-01-01** / <date 8..10 with-compl>

**,TO** = **\*TODAY** / <date 8..10 with-compl>

(Teil 1 von 5)

```

,ACCESS-DATE = *ANY / *TODAY / <date 8..10 with-compl> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 1900-01-01 / <date 8..10 with-compl>
    | ,TO = *TODAY / <date 8..10 with-compl>
,USER-TIME = *ANY / <time 1..8> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 00:00:00 / <time 1..8>
    | ,TO = 23:59:59 / <time 1..8>
,CREATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 00:00:00 / <time 1..8>
    | ,TO = 23:59:59 / <time 1..8>
,MODIFICATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 00:00:00 / <time 1..8>
    | ,TO = 23:59:59 / <time 1..8>
,ACCESS-TIME = *ANY / <time 1..8> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 00:00:00 / <time 1..8>
    | ,TO = 23:59:59 / <time 1..8>
,CODED-CHARACTER-SET = *ANY / *NONE / <name 1..8 with-wild(20)>
,STATE = *ANY / *FREE / *IN-HOLD(...)
  *IN-HOLD(...)
    |
    | HOLDER = *ANY / <name 1..8 with-wild(20)>
,STORAGE-FORM = *ANY / *FULL / *DELTA
,SECONDARY-NAME = *ANY / <alphanum-name 1..32 with-wild(68)>
,SECONDARY-ATTRIBUTE = *ANY / *CSECT / *ENTRY

```

(Teil 2 von 5)



```

,ELEMENT-SIZE = *ANY / <integer 0..2147483647> / *INTERVAL(...)
  *INTERVAL(...)
    |
    | FROM = 0 / <integer 1..2147483647>
    | ,TO = 2147483647 / <integer 1..2147483647>
,PROTECTION = *ANY / *NONE / *PARAMETERS(...)
  *PARAMETERS(...)
    |
    | READ = *ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |   *BY-GUARD(...)
    |     |
    |     | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |     |
    |     | *PARAMETERS(...)
    |     |   |
    |     |   | USER = *ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |     |   | ,PASSWORD = *ANY / *YES / *NO
    |     |
    |     | ,WRITE = *ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |     |   *BY-GUARD(...)
    |     |     |
    |     |     | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |     |     |
    |     |     | *PARAMETERS(...)
    |     |     |   |
    |     |     |   | USER = *ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |     |     |   | ,PASSWORD = *ANY / *YES / *NO
    |     |
    |     | ,EXEC = *ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |     |   *BY-GUARD(...)
    |     |     |
    |     |     | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |     |     |
    |     |     | *PARAMETERS(...)
    |     |     |   |
    |     |     |   | USER = *ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |     |     |   | ,PASSWORD = *ANY / *YES / *NO
    |     |
    |     | ,HOLD = *ANY / *NONE / *BY-GUARD(...) / *PARAMETERS(...)
    |     |   *BY-GUARD(...)
    |     |     |
    |     |     | GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>
    |     |     |
    |     |     | *PARAMETERS(...)
    |     |     |   |
    |     |     |   | USER = *ANY / *NONE / *ALL / list-poss(3): *OWNER / *GROUP / *OTHERS
    |     |     |   | ,PASSWORD = *ANY / *YES / *NO

```

(Teil 3 von 5)

**,EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

**\*ELEMENT(...)**

**ELEMENT = \*ANY (...) / <composed-name 1..64 with-under with-wild(132)>(...**

**\*ANY(...)**

**VERSION = \*ANY / \*HIGHEST-EXISTING / \*UPPER-LIMIT /**  
 <composed-name 1..24 with-under with-wild(52)>

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

<composed-name 1..64 with-under with-wild(132)>(...

**VERSION = \*ANY / \*HIGHEST-EXISTING / \*UPPER-LIMIT /**  
 <composed-name 1..24 with-under with-wild(52)>

**,BASE = \*STD / <composed-name 1..24 with-under with-wild>**

**,TYPE = \*ANY / \*LMS-DEFAULT / <alphanum-name 1..8 with-wild(20)>**

**,USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

**,CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

**,MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

**,ACCESS-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

**\*INTERVAL(...)**

**FROM = 1900-01-01 / <date 8..10 with-compl>**

**,TO = \*TODAY / <date 8..10 with-compl>**

(Teil 4 von 5)

```

,USER-TIME = *ANY / <time 1..8> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 00:00:00 / <time 1..8>
        | ,TO = 23:59:59 / <time 1..8>
,CREATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 00:00:00 / <time 1..8>
        | ,TO = 23:59:59 / <time 1..8>
,MODIFICATION-TIME = *ANY / <time 1..8> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 00:00:00 / <time 1..8>
        | ,TO = 23:59:59 / <time 1..8>
,ACCESS-TIME = *ANY / <time 1..8> / *INTERVAL(...)
    *INTERVAL(...)
        |
        | FROM = 00:00:00 / <time 1..8>
        | ,TO = 23:59:59 / <time 1..8>
,INFORMATION = *LMS-DEFAULT / *MEDIUM / *MINIMUM / *MAXIMUM /
    *SUMMARY / *DELTA-STRUCTURE / *PARAMETERS(...)
    *PARAMETERS(...)
        |
        | GENERAL = *LMS-DEFAULT / *NO / *YES
        | ,HISTORY = *LMS-DEFAULT / *NO / *YES
        | ,SECURITY = *LMS-DEFAULT / *NO / *YES
,LAYOUT = *LMS-DEFAULT / *VARIABLE / *FIXED
,SORT = *LMS-DEFAULT / *BY-NAME / *BY-VERSION / *BY-USER-DATE / *BY-CREATION-DATE /
    *BY-MODIFICATION-DATE / *BY-ACCESS-DATE / *BY-ELEMENT-SIZE /
    *BY-SECONDARY-NAME
,TEXT-OUTPUT = *LOGGING-PARAMETERS / *NONE / *SYSOUT / *SYSLST(...) / *EDT(...)
    *SYSLST(...)
        |
        | SYSLST-NUMBER = *STD / <INTEGER 1..99>
    *EDT(...)
        |
        | WRITE-MODE = *EXTEND / *REPLACE
,STRUCTURE-OUTPUT = *SYSINF / *NONE / <composed-name 1..255>(…)
    <composed-name 1..255>(…)
        |
        | WRITE-MODE = *REPLACE / *EXTEND

```

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

Angaben für die gewünschte Elementbezeichnung.

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, deren Inhaltsverzeichnis ausgegeben wird.

**LIBRARY = \*STD**

Die durch OPEN-LIBRARY eröffnete Bibliothek.

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

Das Inhaltsverzeichnis der hier angegebenen Bibliothek wird ausgegeben.

**LIBRARY = \*LINK(...)**

Die über den Dateikettungsamen zugewiesene Bibliothek.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**ELEMENT = \*ALL / <composed-name 1..64 with-under with-wild(132)>(...)**

Name des Elementes, dessen Bibliothekseintrag ausgegeben wird.

Wird der Standardwert „\*ALL“ angegeben, werden die Bibliothekseinträge aller Elemente passend zur Version und zum Typ ausgegeben.

**VERSION = \*ALL / \*HIGHEST-EXISTING / \*UPPER-LIMIT / <composed-name 1..24 with-under with-wild(52)>**

Version, die das Element besitzt.

**VERSION = \*ALL**

Es werden die Bibliothekseinträge aller oben ausgewählten Elemente ausgegeben, unabhängig von ihrer Version.

**VERSION = \*HIGHEST-EXISTING**

Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der höchsten bestehenden Version ausgegeben.

**VERSION = \*UPPER-LIMIT**

Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der Version X'FF' ausgegeben.

**VERSION = <composed-name 1..24 with-under with-wild(52)>**

Es werden die Bibliothekseinträge aller oben ausgewählten Elemente mit der hier angegebenen Version ausgegeben.

**BASE = \*STD / <composed-name 1..24 with-under with-wild>**

Präfix für die Versionsauswahl. Im Zusammenhang mit VERSION=\*HIGHEST-EXISTING kann damit die höchste Version mit einem bestimmten Präfix angesprochen werden. BASE=\*STD wirkt wie BASE=\*

Der Operand wird ignoriert, falls in einem der Operanden SECONDARY-NAME oder SECONDARY-ATTRIBUTE etwas anderes als \*ANY angegeben ist.

**TYPE = \*LMS-DEFAULTS / \*ALL / <alphanum-name 1..8 with-wild(20)**

Typ, den das auszugebende Element besitzt.

Ist der LMS-Default-Wert für TYPE \*UNDEFINED, so wirkt \*LMS-DEFAULT wie \*ALL.

**USER-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum, das vom Benutzer vergeben wurde.

**USER-DATE = \*ANY**

Das Element besitzt ein beliebiges Datum.

**USER-DATE = \*TODAY**

Die Bibliothekseinträge aller Elemente mit dem aktuellen Tagesdatum werden ausgegeben.

**USER-DATE = <date 8..10 with-compl>**

Es werden die Bibliothekseinträge aller Elemente mit dem hier angegebenen Datum in der Form [JJ]JJ-MM-TT ausgegeben.

**USER-DATE = \*INTERVAL(...)**

Es werden die Bibliothekseinträge aller Elemente, die in dem angegebenen Zeitraum liegen, ausgegeben.

**FROM = 1900-01-01 / <date 8..10 with-compl>**

Beginn des Zeitraums.

**TO = \*TODAY / <date 8..10 with-compl>**

Ende des Zeitraums.

**CREATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Erzeugungsdatum des Elementes. Beschreibung der Operanden analog zu USER-DATE.

**MODIFICATION-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum der letzten Änderung für das Element. Beschreibung der Operanden analog zu USER-DATE.

**ACCESS-DATE = \*ANY / \*TODAY / <date 8..10 with-compl> / \*INTERVAL(...)**

Datum des letzten Zugriffs für das Element. Beschreibung der Operanden analog zu USER-DATE.

**USER-TIME = \*ANY / <time 1..8> / \*INTERVAL(...)**

Uhrzeit, die vom Benutzer vergeben wurde.

**USER-TIME = \*ANY**

Es werden die Bibliothekseinträge aller Elemente unabhängig von der Uhrzeit ausgegeben.

**USER-TIME = <time 1..8>**

Es werden die Bibliothekseinträge aller Elemente mit der hier angegebenen Uhrzeit in der Form HH:MM:SS ausgegeben.

**USER-TIME = \*INTERVAL(...)**

Es werden die Bibliothekseinträge aller Elemente, die in dem angegebenen Zeitraum liegen, ausgegeben.

**FROM = 00:00:00 / <time 1..8>**

Beginn des Zeitraums.

**TO = 23:59:59 / <time 1..8>**

Ende des Zeitraums.

**CREATION-TIME = \*ANY / <time 1..8> / \*INTERVAL(...)**

Zeitpunkt der Erzeugung des Elementes. Beschreibung der Operanden siehe USER-TIME, [Seite 421](#).

**MODIFICATION-TIME = \*ANY / <time 1..8> / \*INTERVAL(...)**

Zeitpunkt der letzten Änderung für das Element. Beschreibung der Operanden siehe USER-TIME, [Seite 421](#).

**ACCESS-TIME = \*ANY / <time 1..8> / \*INTERVAL(...)**

Zeitpunkt des letzten Zugriffs für das Element. Beschreibung der Operanden siehe USER-TIME, [Seite 421](#).

**CODED-CHARACTER-SET = \*ANY / \*NONE / <name 1..8 with-wild(20)>**

Zeichensatz, der dem Element zugeordnet ist.

**CODED-CHARACTER-SET = \*ANY**

Es werden die Bibliothekseinträge aller Elemente unabhängig von dem zugeordneten Zeichensatz ausgewählt.

**CODED-CHARACTER-SET = \*NONE**

Es werden die Bibliothekseinträge aller Elemente, denen kein Zeichensatz zugeordnet ist, ausgewählt.

**CODED-CHARACTER-SET = <name 1..8 with-wild(20)>**

Es werden die Bibliothekseinträge aller Elemente, denen der hier angegebene Zeichensatz zugeordnet ist, ausgewählt.

**STATE = \*ANY / \*FREE / \*IN-HOLD(...)**

Zustand, der dem Element zugeordnet ist.

**STATE = \*ANY**

Die Elemente werden, unabhängig von STATE, ausgewählt.

**STATE = \*FREE**

Nur Elemente mit STATE=FREE werden ausgewählt.

**STATE = \*IN-HOLD(...)**

Nur Elemente mit STATE=IN-HOLD werden ausgewählt.

**HOLDER = \*ANY / <name 1..8 with-wild(20)>**

HOLDER, der dem Element zugeordnet ist.

**HOLDER = \*ANY**

Die Elemente werden, unabhängig vom HOLDER, ausgewählt.

**HOLDER = <name 1..8 with-wild(20)>**

Nur Elemente, denen ein zum Muster passender HOLDER zugeordnet ist, werden ausgewählt.

**STORAGE-FORM = \*ANY / \*FULL / \*DELTA**

Speicherformat für das anzuzeigende Element.

**STORAGE-FORM = \*ANY**

LMS wählt das Element anhand der Speicherform aus. Das Element kann als Voll- oder Deltaelement gespeichert sein.

**STORAGE-FORM = \*FULL**

Das Element muss als Vollelement gespeichert sein.

**STORAGE-FORM = \*DELTA**

Das Element muss als Delta-Element gespeichert sein.

**SECONDARY-NAME = \*ANY / <alphanum-name 1..32 with-wild(68)>**

Sekundärname. Wird hier eine Angabe ungleich \*ANY gemacht, erfolgt die Auswahl über das Sekundärinhaltsverzeichnis der Bibliothek.

Bei der Angabe von Wildcards werden nur die ersten 32 Zeichen der Sekundärnamen zur Bestimmung der Auswahl herangezogen

**SECONDARY-ATTRIBUTE = \*ANY / \*CSECT / \*ENTRY**

Sekundärattribut. Wird hier eine Angabe ungleich \*ANY gemacht, erfolgt die Auswahl über das Sekundärinhaltsverzeichnis der Bibliothek.

**ELEMENT-SIZE = \*ANY / <integer 0..2147483647> / \*INTERVAL(...)**

Elementgröße, nach der die Elemente ausgewählt werden sollen.

**ELEMENT-SIZE = \*ANY**

Die Elementgröße dient nicht als Auswahlkriterium.

**ELEMENT-SIZE = <integer 0..2147483647>**

Es werden Elemente ausgewählt, deren Speicherplatzbelegung der angegebenen Anzahl der PAM-Seiten (2K-Einheit) entsprechen.

**ELEMENT-SIZE = \*INTERVAL(..)**

Es werden Elemente ausgewählt, deren Speicherplatzbelegung im angegebenen Bereich liegen.

**FROM = 0 / <integer 1..2147483647>**

Es werden Elemente ausgewählt, für die mindestens so viele PAM-Seiten (2K-Einheit) für die Speicherplatzbelegung angelegt wurden, wie angegeben.

**TO = 2147483647 / <integer 1..2147483647>**

Es werden Elemente ausgewählt, für die höchstens so viele PAM-Seiten (2K-Einheit) für die Speicherplatzbelegung angelegt wurden, wie angegeben.

**PROTECTION = \*ANY / \*NONE / \*PARAMETERS(...)**

Elementschutz für die ausgewählten Elemente.

**PROTECTION = \*ANY**

Die Elemente werden unabhängig vom Elementschutz ausgewählt.

**PROTECTION = \*NONE**

Die Elemente haben keinen zusätzlichen Elementschutz.

**PROTECTION = \*PARAMETERS(...)**

Angabe, mit welchem Schutz die auszuwählenden Elemente versehen sind.

**READ = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Schutzeinstellung bezüglich Lesen, die dem Element zugeordnet ist.

**READ = \*ANY**

Elemente werden unabhängig vom Leseschutz ausgewählt.

**READ = \*NONE**

Es werden nur Elemente ausgewählt, für die kein Leseschutz eingestellt ist.

**READ = \*BY-GUARD(...)**

Nur Elemente, für die Leseschutz durch GUARD eingestellt ist, werden ausgewählt.

**GUARD-NAME = <filename 1..40 without-cat-gen-vers with-wild>**

Nur Elemente, für die Leseschutz durch einen zum Muster passenden GUARD-NAMEN eingestellt ist, werden ausgewählt.

**READ = \*PARAMETERS(...)**

Nur Elemente, für die Leseschutz durch BACL und/oder Kennwort eingestellt ist, werden ausgewählt.

**USER = \*ANY / \*NONE / \*ALL / list-poss(3): \*OWNER / \*GROUP / \*OTHERS**

Der Kreis der Leseberechtigten, der dem Element zugeordnet ist.

**USER = \*ANY**

Es werden Elemente unabhängig vom Kreis der Leseberechtigten ausgewählt.

**USER = \*NONE**

Nur Elemente, für die an niemanden Leserecht vergeben ist, werden ausgewählt.

**USER = \*OWNER**

Nur Elemente, für die der Eigentümer der Bibliotheksdatei Leserecht hat, werden ausgewählt.



**USER = \*GROUP**

Nur Elemente, für die die Gruppe des Eigentümers der Bibliotheksdatei Leserecht hat, werden ausgewählt.

**USER = \*OTHERS**

Nur Elemente, für die Leserecht für OTHERS vergeben ist, werden ausgewählt.

**PASSWORD = \*ANY / \*YES / \*NO**

Kennwort bzgl. Lesen, das dem Element zugeordnet ist.

**PASSWORD = \*ANY**

Es werden Elemente unabhängig vom Kennwort ausgewählt.

**PASSWORD = \*YES**

Nur Elemente, die durch ein Lesekennwort geschützt sind, werden ausgewählt.

**PASSWORD = \*NO**

Nur Elemente, die durch kein Lesekennwort geschützt sind, werden ausgewählt.

**WRITE = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Schreibrecht. Nur Elemente, für die dieses Recht in der angegebenen Art vergeben ist, werden ausgewählt.

Beschreibung der Operanden analog zu READ.

**EXEC = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ausführungsrecht. Nur Elemente, für die dieses Recht in der angegebenen Art vergeben ist, werden ausgewählt.

Beschreibung der Operanden analog zu READ.

**HOLD = \*ANY / \*NONE / \*BY-GUARD(...) / \*PARAMETERS(...)**

Ausleihrecht. Nur Elemente, für die dieses Recht in der angegebenen Art vergeben ist, werden ausgewählt.

Beschreibung der Operanden analog zu READ.

**EXCEPT-ELEMENT = \*NONE / \*ELEMENT(...)**

Angabe der Elemente, die von der oben getroffenen Auswahl ausgeschlossen werden.

**EXCEPT-ELEMENT = \*NONE**

Es wird kein Element ausgeschlossen.

**EXCEPT-ELEMENT = \*ELEMENT(...)**

Angabe der Elemente, die von der Auswahl ausgeschlossen werden. Ein Element wird ausgeschlossen, wenn alle Felder der EXCEPT-ELEMENT-Struktur, die nicht auf \*ANY gesetzt sind, das Element als Treffer ausweisen. Sind alle Felder der EXCEPT-ELEMENT-Struktur auf \*ANY gesetzt, so werden alle Elemente ausgeschlossen.

Zur Beschreibung der Operanden siehe \*LIBRARY-ELEMENT auf [Seite 420](#).

**INFORMATION = \*LMS-DEFAULT / \*MEDIUM / \*MINIMUM / \*MAXIMUM / \*SUMMARY / \*DELTA-STRUCTURE / \*PARAMETERS (...)**

Dieser Parameter bestimmt den Umfang des auszugebenden Inhaltsverzeichnis. Er bestimmt auch den Umfang der strukturierten Ausgabe (siehe Parameterabhängigkeiten auf [Seite 430 f](#)).

**INFORMATION = \*MEDIUM**

Es wird Typ, Name, Version, Variantennummer und abhängig vom Operanden SORT das Datum bzw. die Elementgröße der ausgewählten Elemente ausgegeben.

**INFORMATION = \*MINIMUM**

Es wird Typ, Name und Version der ausgewählten Elemente ausgegeben.

**INFORMATION = \*MAXIMUM**

Es wird die gesamte Information der ausgewählten Elemente ausgegeben.

**INFORMATION = \*SUMMARY**

Es wird die Anzahl der ausgewählten Elemente pro Typ ausgegeben.

**INFORMATION = \*DELTA-STRUCTURE**

Es wird nur die Beziehung „Vorgänger - Nachfolger“ bei Delta-Elementen ausgegeben. Zusätzlich zur Elementbezeichnung werden die interne Delta-Nummer (DELTA#, spiegelt die chronologische Reihenfolge wieder) und die entsprechende Nummer der Basis (BASE#) ausgegeben. Diese internen Delta-Nummern sind innerhalb eines Baums eindeutig und beschreiben die Verkettung der Elemente im Baum (unabhängig von der externen benutzereigenen Versionsbezeichnung). Die Ausgabe eines Baums ist immer nach DELTA# sortiert, d.h. der Operand SORT wirkt nicht innerhalb eines Baums. Verschiedene Bäume sind durch einen durchgehenden Strich voneinander getrennt. Bei Voll-Elementen sind die Ausgabefelder DELTA# und BASE# leer.

**INFORMATION = \*PARAMETERS(...)**

Zusätzliche Ausgaben.

**GENERAL = \*LMS-DEFAULT / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein allgemeiner Informationsblock bestehend aus Speicherformat, Zustand, Elementgröße und ggf. Zeichensatz und HOLDER der ausgewählten Elemente ausgegeben werden soll.

**HISTORY = \*LMS-DEFAULT / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein HISTORY-Block bestehend aus Benutzerdatum/-zeit, Erzeugungsdatum/-zeit, Änderungsdatum/-zeit und ggf. Zugriffsdatum/zeit der ausgewählten Elemente ausgegeben werden soll.

**SECURITY = \*LMS-DEFAULT / \*NO / \*YES**

Gibt an, ob zusätzlich zu Typ, Name, Version und Variantennummer ein SECURITY-Block der ausgewählten Elemente ausgegeben werden soll, sofern für ein Zugriffsrecht bzw. Ausleihrecht ein zusätzlicher Elementschutz vereinbart ist.

**LAYOUT = \*LMS-DEFAULT / \*VARIABLE / \*FIXED**

Dieser Parameter bestimmt das Format des auszugebenden Inhaltsverzeichnisses.

**LAYOUT = \*VARIABLE**

Die Anzahl der Druckspalten ist abhängig von der längsten Elementbezeichnung innerhalb eines Elementtyps. Bei der Ausgabe auf den Bildschirm und keiner speziellen Sortierung richtet sich das Layout nach der längsten Elementbezeichnung innerhalb des Ausgabepuffers, wobei nachfolgende Ausgaben innerhalb eines Elementtyps sich nur verändern, falls längere Elementbezeichnungen vorkommen.

**LAYOUT = \*FIXED**

Das Inhaltsverzeichnis wird einspaltig im festen Format gedruckt. Eine Spalte bedeutet die untereinanderstehenden Einträge im Inhaltsverzeichnis.

**SORT = \*LMS-DEFAULT / \*BY-NAME / \*BY-VERSION / \*BY-USER-DATE / \*BY-CREATION-DATE / \*BY-MODIFICATION-DATE / \*BY-ACCESS-DATE / \*BY-ELEMENT-SIZE / \*BY-SECONDARY-NAME**

Sortierkriterium für die Inhaltsverzeichniseinträge der ausgewählten Elemente. Als erstes Sortierkriterium wird immer der Typ verwendet.

**SORT = \*BY-NAME**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Name und Version.

**SORT = \*BY-VERSION**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Version und Name.

**SORT = \*BY-USER-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Benutzerdatum, Name und Version.

**SORT = \*BY-CREATION-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Erzeugungsdatum, Name und Version.

**SORT = \*BY-MODIFICATION-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Änderungsdatum, Name und Version.

**SORT = \*BY-ACCESS-DATE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Zugriffsdatum, Name und Version.

**SORT = \*BY-ELEMENT-SIZE**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Elementgröße, Name und Version.

**SORT = \*BY-SECONDARY-NAME**

Die Inhaltsverzeichniseinträge der ausgewählten Elemente werden nach folgender Reihenfolge sortiert: Typ, Sekundärname, Sekundärattribut, Name und Version.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(…)**

Strukturierte Ausgabe.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe [12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variable die strukturierte Ausgabe erfolgen soll. Diese Variable muss als dynamische Listenvariable deklariert sein.

(Kommando: DECLARE-VARIABLE NAME=... (TYPE=\*STRUCTURE), MULTIPLE-ELEMENTS=\*LIST)

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0302	Element nicht gefunden
	64	LMS0303	Element nicht im Bereich der Referenz-Bedingung
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Hinweise*

- Wird der Operand SECONDARY-NAME bzw. SECONDARY-ATTRIBUTE angegeben und ist dieser Operand ungleich \*ANY, wird zusätzlich im Inhaltsverzeichnis eine Überschrift ausgegeben, die über den Sekundärnamen und das Sekundärattribut informiert. Die Sekundärnamen werden aber nicht in der maximalen Länge angezeigt.
- Bei Delta-Elementen bezieht sich der ausgegebene Wert für ELEM-SIZE bzw. die Auswahl der Elementgröße auf die komplette Speicherplatzbelegung des Delta-Baumes. Andere Maße über Größen einzelner Elementversionen sind über SHOW-ELEMENT ...,TEXT-INFO=\*STATISTICS zu erhalten.
- Bei Angabe einer Bibliotheksliste ist INFORMATION=\*DELTA-STRUCTURE nicht erlaubt. Für SECONDARY-NAME und SECONDARY-ATTRIBUTES ist nur der Wert \*ANY erlaubt.

**Notwendige Zugriffsrechte**

Leserecht für LIBRARY.

**Parameterabhängigkeiten**

Zwischen den Operanden INFORMATION, SORT und LAYOUT gibt es folgende Abhängigkeiten:

- Das jeweilige Datum bzw. die Elementgröße beeinflusst nur dann die Sortierung, wenn es auch im Inhaltsverzeichnis ausgegeben werden soll. Dazu darf nicht der Operand INFORMATION =\*MINIMUM eingestellt sein.
- Der Operand LAYOUT wirkt nur, wenn INFORMATION=\*MEDIUM/\*MINIMUM angegeben ist. Bei allen anderen Einstellungen wird das Inhaltsverzeichnis immer im festen Format ausgegeben.
- Wird INFORMATION=\*MAXIMUM angegeben, ist die Information pro Element zu lang für eine Zeile. Sie wird dann in einem Format unabhängig von der Layout-Steuerung ausgegeben.
- Die Ausgabe des Inhaltsverzeichnisses wird beschleunigt, wenn INFORMATION = \*MINIMUM gesetzt wird und die Auswahl sich auf die Operanden ELEMENT, VERSION und TYPE beschränkt. Alle anderen Operanden sollten mit dem Wert \*ANY eingestellt sein.

- Die Operanden INFORMATION (außer INFORMATION=\*SUMMARY) und SORT beeinflussen die Sortierreihenfolge des Inhaltsverzeichnisses. Die folgende Tabelle zeigt diese Abhängigkeiten:

SORT	INFORMATION	
	*MINIMUM	*MEDIUM/*MAXIMUM/*PAR
*BY-NAME	1. Typ 2. Name 3. Version	1. Typ 2. Name 3. Version
*BY-VERSION	1. Typ 2. Version 3. Name	1. Typ 2. Version 3. Name
*BY-USER-DATE	wie SORT = BY-NAME	1. Typ 2. Benutzerdatum 3. Name 4. Version
*BY-CREATION-DATE	wie SORT = BY-NAME	1. Typ 2. Erzeugungsdatum 3. Name 4. Version
*BY-MODIFICATION-DATE	wie SORT = BY-NAME	1. Typ 2. Änderungsdatum 3. Name 4. Version
*BY-ACCESS-DATE	wie SORT = BY-NAME	1. Typ 2. Zugriffsdatum 3. Name 4. Version
*BY-ELEMENT-SIZE	wie SORT = BY-NAME	1. Typ 2. Elementgröße 3. Name 4. Version

SORT	INFORMATION	
	*MINIMUM	*MEDIUM/*MAXIMUM/*PAR
*BY-SECONDARY-NAME - mit Referenz <sup>1</sup>	1. Typ 2. Sekundärname 3. Sekundärattribut 4. Name 5. Version	1. Typ 2. Sekundärname 3. Sekundärattribut 4. Name 5. Version
- ohne Referenz <sup>2</sup>	1. Typ 2. Name 3. Version	1. Typ 2. Name 3. Version

<sup>1</sup> mit Referenz bedeutet, dass bei der Elementauswahl entweder der Sekundärname oder das Sekundärattribut ungleich dem Default-Wert \*ANY angegeben wurde, d.h.:  
SECONDARY-NAME = <alphanum-name...> und/oder SECONDARY-ATTRIBUTE = \*CSECT oder \*ENTRY

<sup>2</sup> ohne Referenz bedeutet, dass bei der Elementauswahl sowohl kein Sekundärname als auch kein Sekundärattribut angegeben wurde.

Bei strukturierter Ausgabe (STRUCTURE-OUTPUT =<composed-name 1..255>) werden die Operanden SORT und INFORMATION beachtet. Es gibt folgende Abhängigkeiten:

- Die Sortierreihenfolge ist identisch mit der bei Textausgabe. Der Operand INFORMATION beeinflusst die Sortierreihenfolge wie dort.
- Pro auszugebenden Inhaltsverzeichniseintrag wird ein Listenelement erzeugt.
- Der Operand INFORMATION beeinflusst den Umfang der in die S-Variablen ausgegebenen Information.
- Die einzelnen Variablen-Elemente sind im Kapitel „Format der LMS-Ausgaben in S-Variable“ beschrieben (siehe [Seite 457](#)).
- Wird INFORMATION=\*DELTA-STRUCTURE angegeben, so erfolgt keine strukturierte Ausgabe.



*Beispiel*

Ausgeben des Inhaltsverzeichnisses der Bibliothek USER.BSPLIB. Die Bibliothek enthält genau ein Element, welches mit all seinen Attributen angezeigt wird.

```
//SHOW-ELEMENT-ATTRIBUTES -
  (LIBRARY=USER.BSPLIB,ELEMENT=*(VERSION=*)),INFORMATION=*MAXIMUM
INPUT  LIBRARY= :N:$USER.USER.BSPLIB
TYPE   = D
NAME   = TEST
VERSION = @                                VARIANT = 0001
-----GENERAL-----
ELEM-SIZE = 12
STORAGEW = *FULL
STATE     = *IN-HOLD    HOLDER     = MUBF
-----HISTORY-----
USER-DATE = 2012-08-12  CRE-DATE  = 2012-09-12  MOD-DATE  = 2012-08-12
USER-TIME = 11:55:36   CRE-TIME  = 11:55:36   MOD-TIME  = 11:55:36
                                         ACC-DATE  = 2012-08-12
                                         ACC-TIME  = 11:55:36
-----SECURITY-----
READ-PASS = *NONE          READ-USER = *OWNER *GROUP -
WR-PASS   = *NONE          WR-PASS   = *OWNER   -   -
```

## SHOW-LIBRARY-ATTRIBUTES

### Bibliotheksattribute anzeigen

Diese Anweisung gibt alle für die Bibliothek eingestellten Attribute aus. Dies sind:

- die durch MODIFY-LIBRARY-ATTRIBUTES eingestellten Attribute
- Bibliotheksgröße in 2K-Einheiten
- Anzahl der freien 2K-Einheiten (können durch Umkopieren beseitigt werden)
- Bibliotheksformat (NK2/NK4)
- UPAM-geschützt (Y/N)

#### SHOW-LIBRARY-ATTRIBUTES

```

LIBRARY = *STD / <filename 1..54 without-vers> / *LINK(...)
  *LINK(...)
    | LINK-NAME = <structured-name 1..8>
,TEXT-OUTPUT = *LOGGING-PARAMETERS / *NONE / *SYSOUT / *SYSLST(...) / *EDT(...)
  *SYSLST(...)
    | SYSLST-NUMBER = *STD / <INTEGER 1..99>
  *EDT(...)
    | WRITE-MODE = *EXTEND / *REPLACE
,STRUCTURE-OUTPUT = *SYSINE / *NONE / <composed-name 1..255>(…)
  <composed-name 1..255>(…)
    | WRITE-MODE = *REPLACE / *EXTEND

```

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Bibliothek, deren Attribute angezeigt werden sollen.

**LIBRARY = \*STD**

Es wird die globale, durch OPEN-LIBRARY eröffnete Bibliothek angezeigt.

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

Name der Bibliothek, deren Zustand angezeigt werden soll.

**LIBRARY = \*LINK(...)**

Der Zustand der über einen Dateikettungsname zugewiesenen Bibliothek wird angezeigt.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(...**

Strukturierte Ausgabe.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe [12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variable die strukturierte Ausgabe erfolgen soll. Diese Variable muss als dynamische Listenvariable deklariert sein. (Kommando:

```
DECLARE-VARIABLE NAME =...(TYPE=*STRUCTURE), MULTIPLE-ELEMENTS=*LIST)
```

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Hinweis*

Die einzelnen Variablen-Elemente sind im [Kapitel „Format der LMS-Ausgaben in S-Variable“](#) beschrieben (siehe [Seite 460](#)).

**Notwendige Zugriffsrechte**

Leserecht für LIBRARY.

*Beispiel*

```
//SHOW-LIBRARY-ATTRIBUTES LIB=BSPLIB
INPUT LIBRARY= :10SQ:$USER.BSPLIB
READ-PASS = *NONE      READ-USER = *OWNER  -   -
WR-PASS   = *YES       WR-USER   = *OWNER  -   -
EXEC-PASS = *NONE      EXEC-USER = *OWNER  -   -
HOLD-PASS = *NONE      HOLD-USER = *OWNER  -   -
ADMIN-PASS = *NONE     ADMIN-USER = *OWNER  -   -
FILE-SIZE = 291        FREE-SIZE  = 62      FORMAT = NK2  UPAM-PROT = N
ACCESS-DATE= *KEEP     WR-CONTRPL = *NONE    STORAGE= *STD
```

## SHOW-LIBRARY-STATUS

### Zustand der Bibliotheken anzeigen

Diese Anweisung zeigt den Zustand der benutzten Bibliotheken an.

LMS gibt nach der Ausführung der Anweisung folgende Informationen aus:

- Name der Bibliothek(en)
- Zustand der Bibliothek(en) (geöffnet oder geschlossen)
- evtl. zugeordneter Dateikettungsname
- evtl. zugeordneter bibliotheksspezifischer Defaulttyp

#### SHOW-LIBRARY-STATUS

**LIBRARY = \*ALL / \*STD / <filename 1..54 without-vers> / \*LINK(...)**

**\*LINK(...)**

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

**LIBRARY = \*ALL / \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Bibliothek, deren Zustand angezeigt werden soll.

**LIBRARY = \*ALL**

Es werden alle benutzten Bibliotheken angezeigt.

**LIBRARY = \*STD**

Es wird die globale, durch OPEN-LIBRARY eröffnete Bibliothek angezeigt.

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

Name der Bibliothek, deren Zustand angezeigt werden soll.

**LIBRARY = \*LINK(...)**

Der Zustand der über einen Dateikettungsname zugewiesenen Bibliothek wird angezeigt.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

*Beispiel*

Während des LMS-Laufs wurde mit fünf verschiedenen Bibliotheken gearbeitet. Eine Bibliothek wurde über den Dateikettungsamen LIB1 angesprochen:

```
//SHOW-LIBRARY-STATUS
STATUS FILENAME                                MODE    LINK  DEF-TYPE
OPEN   :N:$USER.LMSPL.LIB                      UPDATE
CLOSED :N:$USER.MODUL.LIB                       LIB1
CLOSED :N:$USER.MACRO.LIB                        M
CLOSED :N:$USER.QUELL.LIB
CLOSED :N:$USER.TEST.LIB
```

## SHOW-LMS-DEFAULTS

### Anzeigen der LMS-Defaultwerte

Diese Anweisung gibt die aktuelle Belegung der LMS-Defaultwerte aus. Sie können mit der Anweisung MODIFY-LMS-DEFAULTS geändert werden.

#### SHOW-LMS-DEFAULTS

```
DEFAULTS = *STD / *ALL / list-poss(2000): *ELEMENT-ATTRIBUTES / *FILE-ATTRIBUTES /
  *DESTROY-DATA / *WRITE-MODE / *DIALOG-CONTROL / *INFORMATION / *LAYOUT /
  *SORT / *OUTPUT-FORM / *DELETE-SOURCE / *PROTECTION / *MAX-ERROR-WEIGHT /
  *EDT-MODE / *NEXT-ATTEMPT / *COMPARE-PARAMETERS / *TEXT-INFORMATION /
  *MODULE-INFORMATION / *PHASE-INFORMATION / *LLM-INFORMATION
```

#### DEFAULTS = \*STD

Es werden die Default-Werte Element-Attributes, File-Attributes, Destroy-Data, Write-Mode, Dialog-Control, Information, Layout, Sort, Output-Form, Delete-Source und Protection mit ihrer aktuellen Belegung ausgegeben.

#### DEFAULTS = \*ALL

Es werden alle Default-Werte mit ihrer aktuellen Belegung ausgegeben.

#### DEFAULTS = \*ELEMENT-ATTRIBUTES

Es werden die aktuelle Belegung für Elementtyp, Quell- und Zielversion, Speicherform und die Dateiattribute ausgegeben.

#### DEFAULTS = \*FILE-ATTRIBUTES

Es wird die aktuelle Belegung der Dateizugriffsmethode ausgegeben.

#### DEFAULTS = \*DESTROY-DATA

Es wird ausgegeben, ob die Daten überschrieben werden oder nicht.

#### DEFAULTS = \*WRITE-MODE

Es wird die aktuelle Belegung des Schreibmodus ausgegeben.

#### DEFAULTS = \*DIALOG-CONTROL

Es wird die aktuelle Belegung der Dialogführung ausgegeben.

#### DEFAULTS = \*INFORMATION

Es wird die aktuelle Belegung für den Umfang des auszugebenden Inhaltsverzeichnis angezeigt.

#### DEFAULTS = \*LAYOUT

Es wird die aktuelle Belegung für das Format des auszugebenden Inhaltsverzeichnis angezeigt.

**DEFAULTS = \*SORT**

Es wird die aktuelle Belegung für das Sortierkriterium des auszugebenden Inhaltsverzeichnisses angezeigt.

**DEFAULTS = \*OUTPUT-FORM**

Es wird die aktuelle Belegung für das Darstellungsformat der Ausgabe angezeigt.

**DEFAULTS = \*DELETE-SOURCE**

Es wird ausgegeben, ob die ursprüngliche Datei erhalten oder gelöscht werden soll.

**DEFAULTS = \*PROTECION**

Es wird die aktuelle Belegung der Schutzübernahme ausgegeben.

**DEFAULTS = \*MAX-ERROR-WEIGHT**

Es wird die aktuelle Belegung der Spin-off Steuerung ausgegeben.

**DEFAULTS = \*EDT-MODE**

Es wird die aktuelle Belegung des Betriebsmodus für den EDT ausgegeben.

**DEFAULTS = \*NEXT-ATTEMPT**

Es wird die aktuelle Belegung der Steuerung der Öffnungsversuche ausgegeben.

**DEFAULTS = \*COMPARE-PARAMETERS**

Es wird die aktuelle Belegung der Vergleichsparameter ausgegeben.

**DEFAULTS = \*TEXT-INFORMATION**

Es wird die aktuelle Belegung der Informationsmenge für textartige Elemente ausgegeben.

**DEFAULTS = \*MODULE-INFORMATION**

Es wird die aktuelle Belegung der Informationsmenge für Bindemodule ausgegeben.

**DEFAULTS = \*PHASE-INFORMATION**

Es wird die aktuelle Belegung der Informationsmenge für Phasen ausgegeben.

**DEFAULTS = \*LLM-INFORMATION**

Es wird die aktuelle Belegung der Informationsmenge für Bindelademodule ausgegeben.



**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**Notwendige Zugriffsrechte**

Es sind keine Zugriffsrechte erforderlich.

*Beispiel*

```
//SHOW-LMS-DEFAULTS *ALL
ELEMENT-ATTRIBUTES
  TYPE = *NONE
  ELEMENT-VERSION = *ALL
  TO-ELEM-VERSION = *BY-SOURCE
  STORAGE-FORM = *STD
  SOURCE-ATTRIBUTES = *STD
FILE-ATTRIBUTES
  ACCESS-METHOD = *ISAM
DESTROY-DATA = *NO
WRITE-MODE = *CREATE
DIALOG-CONTROL = *NO
INFORMATION = *MEDIUM
LAYOUT = *VARIABLE
SORT = *BY-NAME
OUTPUT-FORM = *STD
DELETE-SOURCE = *NO
PROTECTION = *STD
MAX-ERROR-WEIGHT = *SERIOUS
EDT-MODE = *COMPATIBLE
NEXT-ATTEMPT = *NO
COMPARE-PARAMETERS
  RECORD-PART = *ALL
  SPACES = *STD
  INFORMATION = *MEDIUM
  LAYOUT = *COMPATIBLE
  JOIN-ELEMENT-SETS = *NO
TEXT-INFORMATION = *ALL
MODULE-INFORMATION = *ALL
PHASE-INFORMATION = *ALL
LLM-INFORMATION = *ALL
```

Es werden alle LMS-Defaultwerte ausgegeben. Diese Einstellungen sind direkt nach Start von LMS gültig.

## SHOW-LOGGING-PARAMETERS LMS-Protokollparameter anzeigen

Diese Anweisung gibt die globalen LMS-Einstellungen aus, die momentan gültig sind. Die Einstellungen werden mit der Anweisung MODIFY-LOGGING-PARAMETERS geändert.

Wird diese Anweisung ohne Operanden angegeben, werden alle Voreinstellungen ausgegeben (siehe Beispiel).

```
SHOW-LOGGING-PARAMETERS
```

```
PARAMETERS = *ALL / list-poss(2000): *LOGGING / *TEXT-OUTPUT / *OUTPUT-LAYOUT
```

**PARAMETERS = \*ALL / list-poss(2000): \*LOGGING / \*TEXT-OUTPUT / \*OUTPUT-LAYOUT**

Angabe der Parameter.

**PARAMETERS = \*ALL**

Es werden alle Voreinstellungen mit ihrer aktuellen Belegung ausgegeben.

**PARAMETERS = \*LOGGING**

Es wird der voreingestellte Umfang der LMS-Protokollmenge ausgegeben.

**PARAMETERS = \*TEXT-OUTPUT**

Es wird das voreingestellte Ausgabemedium ausgegeben.

**PARAMETERS = \*OUTPUT-LAYOUT**

Es werden die voreingestellten Parameter für das LMS-Protokollformat ausgegeben.

### Notwendige Zugriffsrechte

Es sind keine Zugriffsrechte erforderlich.

#### *Beispiel*

```
//SHOW-LOGGING-PARAMETERS
LOGGING                = *MINIMUM
TEXT-OUTPUT            = *SYSOUT
OUTPUT-LAYOUT
  LINES-PER-PAGE       = 60
  LINE-SIZE            = 132
  EXTRA-FORM-FEED     = *NO
  HEADER-LINES         = *YES
```

Es werden alle globalen LMS-Einstellungen ausgegeben. Diese Einstellungen sind direkt nach Start von LMS gültig.

## SHOW-STATISTICS

### Ausgeben von Statistiken

Beim Vergleichen von Elementen mit COMPARE-ELEMENT werden Vergleichsstatistiken erzeugt. Diese Vergleichsstatistiken speichert COMPARE-ELEMENT in einen internen Bereich C0 ab. Nach Ausführung der COMPARE-ELEMENT-Anweisung wird C0 auf den Bereich C1 addiert. C0 wird vor jeder weiteren Ausführung von COMPARE-ELEMENT neu initialisiert.

Die Anweisung SHOW-STATISTICS gibt diese Vergleichsstatistiken aus. Dazu muss die Nummer des Bereichs angegeben werden.

#### SHOW-STATISTICS

**NUMBER = \*C0 / \*C1**

**,HEADER-LINE = \*NONE / <c-string 1..132>**

**,TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

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

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

**\*EDT(...)**

**| WRITE-MODE = \*EXTEND / \*REPLACE**

**,STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(...**

**<composed-name 1..255>(...**

**| WRITE-MODE = \*REPLACE / \*EXTEND**

**NUMBER = \*C0 / \*C1**

Nummer der Bereichs, der ausgegeben werden soll:

\*C0 : Bereich, der die Statistik des aktuellen Vergleichs enthält.

\*C1 : Bereich, der die Gesamtstatistik aller bisher in diesem LMS-Lauf durchgeführten Vergleiche enthält.

**HEADER-LINE = \*NONE / <c-string 1..132>**

Es kann gesteuert werden, ob eine benutzereigene Überschrift (<c-string>) ausgegeben wird oder nicht. Standardmäßig wird keine benutzereigene Überschrift ausgegeben.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(...)**

Strukturierte Ausgabe.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe [12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variable die strukturierte Ausgabe erfolgen soll. Diese Variable muss als dynamische Strukturvariable deklariert sein.

(Kommando: DECLARE-VARIABLE NAME =...(TYPE=\*STRUCTURE), MULTIPLE-ELEMENTS=\*LIST)

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

Die einzelnen Variablen-Elemente sind im [Kapitel „Format der LMS-Ausgaben in S-Variablen“](#) beschrieben (siehe [Seite 462](#)).

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
2	0	LMS0313	Überlauf eines Statistikzählers
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**Notwendige Zugriffsrechte**

Es sind keine Zugriffsrechte erforderlich.

*Hinweis*

Der Maximalwert für Elementzähler ist 99 999, für Zeilenzähler ist er 999 999 999.

Bei Ausgabe auf SYSOUT zeigt LMS bis zu 8 Ziffern an, falls der Wert kleiner als 100 000 000 ist. Andernfalls wird der Wert in Exponentialnotation dargestellt, z.B. 12345E+4. In diesem Fall können die exakten Werte der strukturierten Ausgabe in eine S-Variable oder der Textausgabe nach SYSLST, zum EDT oder in ein Bibliothekselement entnommen werden.

Falls die Grenze von 9 Ziffern überschritten wird, wird die Meldung LMS0313 ausgegeben und die betroffenen Zähler zählen Modulo  $10^9$  weiter.

*Beispiel*

Die S-Variable C1 wurde deklariert:

```
/DECL-VARIABLE C1(TYPE=*STRUCTURE),MULTIPLE-ELEMENTS=*LIST
```

Es wurden zwei Vergleiche durchgeführt. Im Speicher C0 ist das Ergebnis des letzten Vergleichs enthalten, im Speicher C1 die Summe der bisher durchgeführten Vergleiche.

```
//SHOW-STATISTICS NUMBER=*C0
```

```
AREA C0
```

STATISTIC	PRIM. ELEM.	PRIM. LINES	INS. LINES	SAME LINES	DEL. LINES	INS+DEL LINES	SEC. LINES	SEC ELEM.
S (SAME)	0	0	-	0	-	-	0	0
C (CHANGED)	1	5	5	0	9	14	9	1
I (INSERTED)	0	0	0	-	-	0	-	-
D (DELETED)	-	-	-	-	0	0	0	0

```
-----
```

TOTAL	1	5	5	0	9	14	9	1
-------	---	---	---	---	---	----	---	---

```
//SHOW-STATISTICS NUMBER=*C1, STRUCTURE-OUTPUT=C1, TEXT-OUTPUT=*NONE
```

Die Struktur C1 kann mit /SHOW-VARIABLE C1#1 angesehen werden:

```
C1(*LIST).SAME.NUM-OF-PRIMARY = 5
C1(*LIST).SAME.NUM-OF-SECONDARY = 5
C1(*LIST).SAME.LINE.PRIMARY = 37
C1(*LIST).SAME.LINE.INS = 0
C1(*LIST).SAME.LINE.SAME = 37
C1(*LIST).SAME.LINE.DEL = 0
C1(*LIST).SAME.LINE.SECONDARY = 37
C1(*LIST).CHA.NUM-OF-PRIMARY = 0
C1(*LIST).CHA. ....
.....
C1(*LIST).DEL.LINE.SECONDARY = 0
C1(*LIST).TOTAL.NUM-OF.PRIMARY = 7
C1(*LIST).TOTAL.NUM-OF.SECONDARY = 5
C1(*LIST).TOTAL.LINE.PRIMARY = 47
C1(*LIST).TOTAL.LINE.INS = 10
C1(*LIST).TOTAL.LINE.SAME = 37
C1(*LIST).TOTAL.LINE.DEL = 0
C1(*LIST).TOTAL.LINE.SECONDARY = 37
```

## SHOW-TYPE-ATTRIBUTES

### Typattribute anzeigen

Diese Anweisung gibt alle für einen Elementtyp eingestellten Attribute aus.

SHOW-TYPE-ATTRIBUTES
<pre> <b>LIBRARY</b> = <b>*STD</b> / &lt;filename 1..54 without-vers&gt; / <b>*LINK(...)</b>   <b>*LINK(...)</b>       <b>LINK-NAME</b> = &lt;structured-name 1..8&gt; <b>,TYPE</b> = <b>*LMS-DEFAULT</b> / <b>*ALL</b> / &lt;alphanum-name 1..8 with-wild(20)&gt; <b>,TEXT-OUTPUT</b> = <b>*LOGGING-PARAMETERS</b> / <b>*NONE</b> / <b>*SYSOUT</b> / <b>*SYSLST(...)</b> / <b>*EDT(...)</b>   <b>*SYSLST(...)</b>       <b>SYSLST-NUMBER</b> = <b>*STD</b> / &lt;INTEGER 1..99&gt;   <b>*EDT(...)</b>       <b>WRITE-MODE</b> = <b>*EXTEND</b> / <b>*REPLACE</b> <b>,STRUCTURE-OUTPUT</b> = <b>*SYSINE</b> / <b>*NONE</b> / &lt;composed-name 1..255&gt;(...)   &lt;composed-name 1..255&gt;(...)       <b>WRITE-MODE</b> = <b>*REPLACE</b> / <b>*EXTEND</b> </pre>

**LIBRARY = \*STD / <filename 1..54 without-vers> / \*LINK(...)**

Angabe der Bibliothek, in dem sich der anzuzeigende Typ befindet.

**LIBRARY = \*STD**

Es wird die global eröffneten Bibliothek verwendet.

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

Name der Bibliothek, deren Typattribute angezeigt werden soll.

**LIBRARY = \*LINK(...)**

Die Typattribute der über einen Dateikettungsname zugewiesenen Bibliothek werden angezeigt.

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

Dateikettungsname der Bibliothek, der vor dem Aufruf von LMS mit einem /ADD-FILE-LINK-Kommando vereinbart wurde.

**TYPE = \*LMS-DEFAULT / \*ALL / <alphanum-name 1..8 with-wild(20)>**

Elementtyp, dessen Attribute ausgegeben werden sollen. Ist der LMS Default-Wert \*UNDEFINED, so wirkt \*LMS-DEFAULT wie \*ALL.



**TEXT-OUTPUT = \*LOGGING-PARAMETERS / \*NONE / \*SYSOUT / \*SYSLST(...) / \*EDT(...)**

Steuerung der Protokollausgabe.

**TEXT-OUTPUT = \*LOGGING-PARAMETERS**

Die Protokollausgabe erfolgt in das mit //MODIFY-LOGGING-PARAMETERS, TEXT-OUTPUT= festgelegte Ausgabemedium.

**TEXT-OUTPUT = \*NONE**

Die Protokollausgabe wird bis auf Fehlermeldungen unterdrückt.

**TEXT-OUTPUT = \*SYSOUT**

Die Ausgabe wird nach SYSOUT geschrieben.

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

Die Ausgabe wird nach SYSLST geschrieben.

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

Bezeichnet die SYSLST-Datei, in die die Ausgabe erfolgen soll.

**SYSLST-NUMBER = \*STD**

Es gilt die Systemdatei SYSLST.

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

Es gilt die Systemdatei aus der Menge SYSLST01 bis SYSLST99, deren Nummer hier angegeben wird.

**TEXT-OUTPUT = \*EDT(...)**

Die Ausgabe erfolgt in die Arbeitsdatei 9 des EDT. Tritt bei der Protokollausgabe ein Fehler auf, so wird auf den Standardprotokollstrom (SYSOUT) umgeschaltet.

**WRITE-MODE = \*EXTEND / \*REPLACE**

Schreibmodus der Ausgabe bezogen auf den Inhalt der Arbeitsdatei 9.

**WRITE-MODE = \*EXTEND**

Existieren Daten in der Arbeitsdatei 9, werden diese durch die Ausgabe erweitert. Ansonsten wird die Ausgabe an den Anfang der Arbeitsdatei geschrieben.

**WRITE-MODE = \*REPLACE**

Die Ausgabe erfolgt an den Anfang der Arbeitsdatei 9. Existierende Daten in der Arbeitsdatei werden dabei ersetzt.

**STRUCTURE-OUTPUT = \*SYSINF / \*NONE / <composed-name 1..255>(…)**

Strukturierte Ausgabe.

**STRUCTURE-OUTPUT = \*SYSINF**

Die strukturierte Ausgabe erfolgt in den durch ASSIGN-STREAM zugewiesenen SYSINF-Strom (siehe[12]).

**STRUCTURE-OUTPUT = \*NONE**

Es erfolgt keine strukturierte Ausgabe.

**STRUCTURE-OUTPUT = <composed-name 1..255>(…)**

Gibt an, in welche S-Variablen die strukturierte Ausgabe erfolgen soll. Diese Variablen müssen als dynamische Listenvariable deklariert sein.

(Kommando: DECLARE-VARIABLE NAME =...(TYPE=\*STRUCTURE), MULTIPLE-ELEMENTS=\*LIST)

**WRITE-MODE = \*REPLACE / \*EXTEND**

Gibt an, ob die Listenvariable überschrieben oder erweitert wird.

**WRITE-MODE = \*REPLACE**

Der alte Inhalt der Listenvariable wird überschrieben.

**WRITE-MODE = \*EXTEND**

Die neuen Listenelemente werden an die bestehende Liste angehängt.

Die einzelnen Variablen-Elemente sind im [Kapitel „Format der LMS-Ausgaben in S-Variablen“](#) beschrieben (siehe [Seite 464](#)).

**Anweisungs-Returncode**

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS0304	Typ nicht gefunden
	64	LMS1003	Fehler während Mengenverarbeitung (Wildcard) bei mindestens einem Element oder einer Datei
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

**Notwendige Zugriffsrechte**

Leserecht für LIBRARY.

*Hinweise*

- Wird ein bestimmter Typ angegeben, werden die explizit über MODIFY-TYPE-ATTR eingestellten Werte ausgegeben, sofern vorhanden, ansonsten die implizit immer vorhandenen Standardwerte.
- Für TYPE=\* werden die Attribute für alle explizit vereinbarten Typen und für Typen, unter denen Elemente existieren, ausgegeben.

*Beispiel*

Es werden die Typattribute für den Elementtyp S eingestellt und danach ausgegeben. Die verwendete Bibliothek ist die global eingestellte Bibliothek.

```
//MODIFY-TYPE-ATTRIBUTES TYPE=S,CONVENTION=*STD-SEQUENCE(EXAMPLE=V001)
//SHOW-TYPE-ATTRIBUTES TYPE=S
INPUT  LIBRARY= :10SQ:$USER.LIB.MODTYATT
TYPE   = S
SUPER-TYPE = *NONE
CONVENTION = *STD-SEQUENCE
EXAMPLE = V001
INIT-ELEM-P= *NONE
ADMINISTRAT= *NONE
STORAGE  = *STD           WR-CONTROL = *NONE
```

## SHOW-USER-EXITS

### Benutzerausgänge anzeigen

Diese Anweisung zeigt die aktiven Benutzerausgänge.

SHOW-USER-EXITS

Die Anweisung SHOW-USER-EXITS besitzt keine Operanden.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

### Beispiel

```
//SHOW-USER-EXITS
FUNCTION  ENTRY      LIBRARY      INT
SHOW     USELST     TEST.LIB     V1
```

## WRITE-COMMENT

### Kommentar schreiben

Mit dieser Anweisung werden Kommentare in das durch die Anweisung //MODIFY-LOGGING-PARAMETERS TEXT-OUTPUT=... festgelegte Ausgabemedium geschrieben, im Gegensatz zur Anweisung WRITE-TEXT, die immer nach SYSOUT oder SYSLST ausgibt.

<b>WRITE-COMMENT</b>
<b>COMMENT = ' ' / &lt;c-string 1..1024 with-low&gt;</b>

### **COMMENT = ' ' / <c-string 1..1024 with-low>**

Kommentartext. Wird nichts angegeben, wird als Standardwert ein Leerzeichen genommen, d.h. eine Leerzeile erzeugt.

### Anweisungs-Returncode

(SC2)	SC1	Maincode	Bedeutung
	0	CMD0001	Ohne Fehler
2	0	LMS0129	Anweisung abgebrochen durch Benutzer
	1	CMD0230	Syntaxfehler
	32	LMS1002	Interner Fehler
	64	LMS1004	Sonstige Fehler
	130	LMS0041	Systemadressraum erschöpft

### *Beispiel*

Der Text, der in Hochkommata eingeschlossen ist, wird als Kommentar interpretiert. Er kann auch über mehrere Zeilen gehen.

```
//write-comment 'Dies ist die letzte LMS-Anweisung, die beschrieben wird.-
//                Auf der nächsten Seite beginnt ein neues Kapitel.'
```



## 8 Format der LMS-Ausgaben in S-Variable

Sind Strukturelemente einer Variablen wegen Randbedingungen (siehe unten) nicht gültig, werden sie angelegt und mit Leerstring bzw. 0 versorgt.

Bedeutung der Spalten: **T** ist Datentyp (String oder Integer), **W** ist die Informationsauswahl (nur beim Format für //SHOW-ELEMENT-ATTRIBUTES; siehe [Seite 457](#)) und **Bed.** ist Bedingung.

### 8.1 Anweisung COMPARE-ELEMENT

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Bibliotheksname (erstes Element)	varname#.PRIMARY.LIB	S	<filename 1..54>	
Elementname (erstes Element)	varname#.PRIMARY.ELEM	S	<comp-name 1..64>	
Elementversion (erstes Element)	varname#.PRIMARY.VERSION	S	<comp-name 1..24> / `*UP-LIM`	
Elementtyp (erstes Element)	varname#.PRIMARY.TYPE	S	<alphanumeric 1..8>	
Bibliotheksname (zweites Element)	varname#.SECONDARY.LIB	S	<filename 1..54>	
Elementname (zweites Element)	varname#.SECONDARY.ELEM	S	<comp-name 1..64>	
Elementversion (zweites Element)	varname#.SECONDARY.VERSION	S	<comp-name 1..24> / `*UP-LIM`	
Elementtyp (zweites Element)	varname#.SECONDARY.TYPE	S	<alphanumeric 1..8>	
Ergebnis des Vergleichs	varname#.COMPARE-RESULT	S	`*SAME` / `*CHA` / `*DEL` / `*INS`	
Anzahl Sätze im ersten Element	varname#.LINE.PRIMARY	I	<integer>	
Anzahl eingefügter Sätze	varname#.LINE.INS	I	<integer>	
Anzahl gleicher Sätze	varname#.LINE.SAME	I	<integer>	
Anzahl gelöschter Sätze	varname#.LINE.DEL	I	<integer>	
Anzahl Sätze im zweiten Element	varname#.LINE.SECONDARY	I	<integer>	

## 8.2 Anweisung FIND-ELEMENT

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Bibliotheksname	varname#.LIB	S	<filename 1..54>	
Elementname	varname#.ELEM	S	<comp-name 1..64>	
Elementversion	varname#.VERSION	S	<comp-name 1..24> / *'UP-LIM'	
Elementtyp	varname#.TYPE	S	<alphanum 1..8>	
Anzahl der Sätze	varname#.LINE.ALL	I	<integer>	
Anzahl der Treffer-Sätze	varname#.LINE.MATCH	I	<integer>	



### 8.3 Anweisung SHOW-ELEMENT-ATTRIBUTES

Informationsauswahl	Wahl#
INFORMATION = *MINIMUM / *MEDIUM	1
INFORMATION = *MAXIMUM	2
INFORMATION = *PARAMETERS (GENERAL=*YES)	3
INFORMATION = *PARAMETERS (HISTORY=*YES)	4
INFORMATION = *PARAMETERS (SECURITY=*YES)	5
INFORMATION = *SUMMARY	6

Ausgabeinformation	Name der S-Variable	T	Inhalt	W	Bed.
Bibliotheksname	varname#.LIB	S	<filename 1..54>	1-6	
Elementname	varname#.ELEM	S	<comp-name 1..64>	1-5	
Elementversion	varname#.VERSION	S	<comp-name 1..24> / *'UP-LIM'	1-5	
Elementtyp	varname#.TYPE	S	<alphanum 1..8>	1-6	
Sekundärname	varname#.SECONDARY-NAME	S	<alphanum 1..32>	1,2	*1
Sekundärattribut	varname#.SECONDARY-ATTR	S	'*CSECT' / '*ENTRY'	1,2	*1
Benutzer-Datum	varname#.USER-DATE	S	<yyyy-mm-dd>	2,4	
Benutzer-Zeit	varname#.USER-TIME	S	<hh:mm:ss>	2,4	
Datum der Elementerzeugung	varname#.CRE-DATE	S	<yyyy-mm-dd>	2,4	
Zeitpunkt der Elementerzeugung	varname#.CRE-TIME	S	<hh:mm:ss>	2,4	
Datum der letzten Änderung des Elements	varname#.MOD-DATE	S	<yyyy-mm-dd>	2,4	
Zeitpunkt der letzten Änderung des Elements	varname#.MOD-TIME	S	<hh:mm:ss>	2,4	
Datum des letzten Zugriffs auf das Element	varname#.ACCESS-DATE	S	'*NONE' / <yyyy-mm-dd>	2,4	
Zeitpunkt des letzten Zugriffs auf das Element	varname#.ACCESS-TIME	S	'*NONE' / <hh:mm:ss>	2,4	
Speicherungsform (Voll- / Deltaspeicherung)	varname#.STOR-FORM	S	'*FULL' / '*DELTA'	2,3	
Zeichensatz, der dem Element zugeordnet ist	varname#.CODED-CHAR-SET	S	' / <name 1..8>	2,3	
Zustand des Elements	varname#.STA	S	'*FREE' / '*IN-HOLD'	2,3	
Holder, der dem Element zugeordnet ist	varname#.HOLDER	S	' / <name 1..8>	2,3	

Ausgabeinformation	Name der S-Variable	T	Inhalt	W	Bed.
Indikator, ob Elementschutz vereinbart ist	varname#.PROT.DEFI	S	'*NONE' / '*PAR'	2,5	
Indikator für Leseschutz	varname#.READ.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	2,5	*2
Name des Lese-Guards	varname#.READ.GUARD-NAME	S	<filename 1..18>	2,5	*3
Indikator für Leseberechtigung (BACL)	varname#.READ.USER-DEFI	S	'*NONE' / '*LIST'	2,5	*4
Indikator für Leseberechtigung des Eigentümers	varname#.READ.OWNER	S	'/' / '*OWNER'	2,5	*5
Indikator für Leseberechtigung der Gruppe des Eigentümers	varname#.READ.GROUP	S	'/' / '*GROUP'	2,5	*5
Indikator für Leseberechtigung für OTHERS	varname#.READ.OTHERS	S	'/' / '*OTHERS'	2,5	*5
Indikator, ob für das Element ein Lese-Kennwort vereinbart ist	varname#.READ.PASS	S	'*NO' / '*YES'	2,5	*4
Indikator für Schreibschutz	varname#.WRITE.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	2,5	*2
Name des Schreib-Guards	varname#.WRITE.GUARD-NAME	S	<filename 1..18>	2,5	*3
Indikator für Schreibberechtigung (BACL)	varname#.WRITE.USER-DEFI	S	'*NONE' / '*LIST'	2,5	*4
Indikator für Schreibberechtigung des Eigentümers	varname#.WRITE.OWNER	S	'/' / '*OWNER'	2,5	*5
Indikator für Schreibberechtigung der Gruppe des Eigentümers	varname#.WRITE.GROUP	S	'/' / '*GROUP'	2,5	*5
Indikator für Schreibberechtigung für OTHERS	varname#.WRITE.OTHERS	S	'/' / '*OTHERS'	2,5	*5
Indikator, ob für das Element ein Schreib-Kennwort vereinbart ist	varname#.WRITE.PASS	S	'*NO' / '*YES'	2,5	*4
Indikator für Ausführungsschutz	varname#.EXEC.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	2,5	*2
Name des Ausführungs-Guards	varname#.EXEC.GUARD-NAME	S	<filename 1..18>	2,5	*3
Indikator für Ausführungsberechtigung (BACL)	varname#.EXEC.USER-DEFI	S	'*NONE' / '*LIST'	2,5	*4
Indikator für Ausführungsberechtigung des Eigentümers	varname#.EXEC.OWNER	S	'/' / '*OWNER'	2,5	*5
Indikator für Ausführungsberechtigung der Gruppe des Eigentümers	varname#.EXEC.GROUP	S	'/' / '*GROUP'	2,5	*5
Indikator für Ausführungsberechtigung für OTHERS	varname#.EXEC.OTHERS	S	'/' / '*OTHERS'	2,5	*5
Indikator, ob für das Element ein Ausführungs-Kennwort vereinbart ist	varname#.EXEC.PASS	S	'*NO' / '*YES'	2,5	*4
Indikator für Ausleihschutz	varname#.HOLD.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	2,5	*2

Ausgabeinformation	Name der S-Variable	T	Inhalt	W	Bed.
Name des Ausleih-Guards	varname#.HOLD.GUARD-NAME	S	<filename 1..18>	2,5	*3
Indikator für Ausleihberechtigung (BACL)	varname#.HOLD.USER-DEFI	S	'*'NONE' / '*'LIST'	2,5	*4
Indikator für Ausleihberechtigung des Eigentümers	varname#.HOLD.OWNER	S	'*' / '*'OWNER'	2,5	*5
Indikator für Ausleihberechtigung der Gruppe des Eigentümers	varname#.HOLD.GROUP	S	'*' / '*'GROUP'	2,5	*5
Indikator für Ausleihberechtigung für OTHERS	varname#.HOLD.OTHERS	S	'*' / '*'OTHERS'	2,5	*5
Indikator, ob für das Element ein Ausleih-Kennwort vereinbart ist	varname#.HOLD.PASS	S	'*NO' / '*'YES'	2,5	*4
Elementgröße in 2K-Seiten	varname#.ELEM-SIZE	I	<integer>	2,3	
Anzahl Elemente für den Elementtyp	varname#.NUM-OF-ELEM	I	<integer>	6	

\*1 : SORT = \*BY-SECONDARY-NAME und SECONDARY-NAME ≠ \*ANY

\*2 : varname#.PROT-DEFI = '\*PAR'

\*3 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*BY-GUARD'

\*4 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*PAR'

\*5 : \*4 und varname#.xxxx.USER-DEFI = '\*LIST'

## 8.4 Anweisung SHOW-LIBRARY-ATTRIBUTES

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Bibliothekensname	varname#.LIB	S	<filename 1..54>	
zulässige Speicherungsform	varname#.STOR-FORM	S	'*STD' / '*NONE' / '*FULL' / '*DELTA'	
Steuerung zusätzlicher Prüfungen	varname#.WRITE-CONTR	S	'*NONE' / '*ACTIVATE' '*DEACTIVATE'	
Indikator für das Führen des Zugriffsdatums	varname#.ACCESS-DATE	S	'*NONE' / '*KEEP'	
Indikator für Verwaltungsschutz	varname#.ADM.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	
Name des Verwaltungs-Guards	varname#.ADM.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Verwaltungsberechtigung (BACL)	varname#.ADM.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Verwaltungsberechtigung des Eigentümers	varname#.ADM.OWNER	S	' / '*OWNER'	*5
Indikator für Verwaltungsberechtigung der Gruppe des Eigentümers	varname#.ADM.GROUP	S	' / '*GROUP'	*5
Indikator für Verwaltungsberechtigung für OTHERS	varname#.ADM.OTHERS	S	' / '*OTHERS'	*5
Indikator, ob für das Element ein Verwaltungs-Kennwort vereinbart ist	varname#.ADM.PASS	S	'*NO' / '*YES'	*4
Indikator, ob Elementschutz vereinbart ist	varname#.PROT-DEFI	S	'*NONE' / '*PAR'	
Indikator für Leseschutz	varname#.PROT.READ-DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Lese-Guards	varname#.PROT.READ.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Leseberechtigung (BACL)	varname#.PROT.READ.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Leseberechtigung des Eigentümers	varname#.PROT.READ.OWNER	S	' / '*OWNER'	*5
Indikator für Leseberechtigung der Gruppe des Eigentümers	varname#.PROT.READ.GROUP	S	' / '*GROUP'	*5
Indikator für Leseberechtigung für OTHERS	varname#.PROT.READ.OTHERS	S	' / '*OTHERS'	*5
Indikator, ob für das Element ein Lese-Kennwort vereinbart ist	varname#.PROT.READ.PASS	S	'*NO' / '*YES'	*4
Indikator für Schreibschutz	varname#.PROT.WRITE.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Schreib-Guards	varname#.PROT.WRITE.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Schreibberechtigung (BACL)	varname#.PROT.WRITE.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Schreibberechtigung des Eigentümers	varname#.PROT.WRITE.OWNER	S	' / '*OWNER'	*5

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Indikator für Schreibberechtigung der Gruppe des Eigentümers	varname#.PROT.WRITE.GROUP	S	'' / '*GROUP'	*5
Indikator für Schreibberechtigung für OTHERS	varname#.PROT.WRITE.OTHERS	S	'' / '*OTHERS'	*5
Indikator, ob für das Element ein Schreib-Kennwort vereinbart ist	varname#.PROT.WRITE.PASS	S	'*NO' / '*YES'	*4
Indikator für Ausführungsschutz	varname#.PROT.EXEC.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Ausführungs-Guards	varname#.PROT.EXEC.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Ausführungsberechtigung (BACL)	varname#.PROT.EXEC.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Ausführungsberechtigung des Eigentümers	varname#.PROT.EXEC.OWNER	S	'' / '*OWNER'	*5
Indikator für Ausführungsberechtigung der Gruppe des Eigentümers	varname#.PROT.EXEC.GROUP	S	'' / '*GROUP'	*5
Indikator für Ausführungsberechtigung für OTHERS	varname#.PROT.EXEC.OTHERS	S	'' / '*OTHERS'	*5
Indikator, ob für das Element ein Ausführungs-Kennwort vereinbart ist	varname#.PROT.EXEC.PASS	S	'*NO' / '*YES'	*4
Indikator für Ausleihschutz	varname#.PROT.HOLD.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Ausleih-Guards	varname#.PROT.HOLD.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Ausleihberechtigung (BACL)	varname#.PROT.HOLD.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Ausleihberechtigung des Eigentümers	varname#.PROT.HOLD.OWNER	S	'' / '*OWNER'	*5
Indikator für Ausleihberechtigung der Gruppe des Eigentümers	varname#.PROT.HOLD.GROUP	S	'' / '*GROUP'	*5
Indikator für Ausleihberechtigung für OTHERS	varname#.PROT.HOLD.OTHERS	S	'' / '*OTHERS'	*5
Indikator, ob für das Element ein Ausleih-Kennwort vereinbart ist	varname#.PROT.HOLD.PASS	S	'*NO' / '*YES'	*4
Format der Bibliothek	varname#.LIB-FORM	S	'*NK2' / '*NK4'	
Indikator, ob die Bibliothek (UPAM-)geschützt ist	varname#.PROT-LIB	S	'*YES' / '*NO'	
Bibliotheksgröße in 2K-Seiten	varname#.F-SIZE	I	<integer>	
Anzahl freier 2K-Seiten	varname#.FREE-SIZE	I	<integer>	

\*2 : varname#.PROT-DEFI = '\*PAR'

\*3 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*BY-GUARD'

\*4 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*PAR'

\*5 : \*4 und varname#.xxxx.USER-DEFI = '\*LIST'

## 8.5 Anweisung SHOW-STATISTICS

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Anzahl Primär-Elemente (Ergebnis „gleich“)	varname#.SAME.NUM-OF-PRIMARY	I	<integer>	
Anzahl Sekundär-Elemente (Ergebnis „gleich“)	varname#.SAME.NUM-OF-SECONDARY	I	<integer>	
Anzahl Primär-Element-Sätze (Ergebnis „gleich“)	varname#.SAME.LINE.PRIMARY	I	<integer>	
	varname#.SAME.LINE.INS	I	0	
Anzahl gleicher Sätze (Ergebnis „gleich“)	varname#.SAME.LINE.SAME	I	<integer>	
	varname#.SAME.LINE.DEL	I	0	
Anzahl Sekundär-Element-Sätze (Ergebnis „gleich“)	varname#.SAME.LINE.SECONDARY	I	<integer>	
Anzahl Primär-Elemente (Ergebnis „geändert“)	varname#.CHA.NUM-OF-PRIMARY	I	<integer>	
Anzahl Sekundär-Elemente (Ergebnis „geändert“)	varname#.CHA.NUM-OF-SECONDARY	I	<integer>	
Anzahl Primär-Element-Sätze (Ergebnis „geändert“)	varname#.CHA.LINE.PRIMARY	I	<integer>	
Anzahl eingefügter Sätze (Ergebnis „geändert“)	varname#.CHA.LINE.INS	I	<integer>	
Anzahl gleicher Sätze (Ergebnis „geändert“)	varname#.CHA.LINE.SAME	I	<integer>	
Anzahl gelöschter Sätze (Ergebnis „geändert“)	varname#.CHA.LINE.DEL	I	<integer>	
Anzahl Sekundär-Element-Sätze (Ergebnis „geändert“)	varname#.CHA.LINE.SECONDARY	I	<integer>	
Anzahl Primär-Elemente (Ergebnis „eingefügt“)	varname#.INS.NUM-OF-PRIMARY	I	<integer>	
	varname#.INS.NUM-OF-SECONDARY	I	0	
Anzahl Primär-Element-Sätze (Ergebnis „eingefügt“)	varname#.INS.LINE.PRIMARY	I	<integer>	
Anzahl eingefügter Sätze (Ergebnis „eingefügt“)	varname#.INS.LINE.INS	I	<integer>	
	varname#.INS.LINE.SAME	I	0	
	varname#.INS.LINE.DEL	I	0	
Anzahl Sekundär-Element-Sätze (Ergebnis „eingefügt“)	varname#.INS.LINE.SECONDARY	I	0	
	varname#.DEL.NUM-OF-PRIMARY	I	0	
Anzahl Sekundär-Elemente (Ergebnis „gelöscht“)	varname#.DEL.NUM-OF-SECONDARY	I	<integer>	
	varname#.DEL.LINE.PRIMARY	I	0	
	varname#.DEL.LINE.INS	I	0	
	varname#.DEL.LINE.SAME	I	0	
Anzahl gelöschter Sätze (Ergebnis „gelöscht“)	varname#.DEL.LINE.DEL	I	<integer>	
Anzahl Sekundär-Element-Sätze (Ergebnis „gelöscht“)	varname#.DEL.LINE.SECONDARY	I	<integer>	

<b>Ausgabeinformation</b>	<b>Name der S-Variable</b>	<b>T</b>	<b>Inhalt</b>	<b>Bed.</b>
Anzahl Primär-Elemente (Summe)	varname#.TOTAL.NUM-OF-PRIMARY	I	<integer>	
Anzahl Sekundär-Elemente (Summe)	varname#.TOTAL.NUM-OF-SECONDARY	I	<integer>	
Anzahl Primär-Element-Sätze (Summe)	varname#.TOTAL.LINE.PRIMARY	I	<integer>	
Anzahl eingefügter Sätze (Summe)	varname#.TOTAL.LINE.INS	I	<integer>	
Anzahl gleicher Sätze (Summe)	varname#.TOTAL.LINE.SAME	I	<integer>	
Anzahl gelöschter Sätze (Summe)	varname#.TOTAL.LINE.DEL	I	<integer>	
Anzahl Sekundär-Element-Sätze (Summe)	varname#.TOTAL.LINE.SECONDARY	I	<integer>	

Felder mit Wert 0 werden angelegt, um die Teil-Strukturen der Variablen einheitlich zu halten.

## 8.6 Anweisung SHOW-TYPE-ATTRIBUTES

Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Bibliotheksname	varname#.LIB	S	<filename 1..54>	
Elementtyp	varname#.TYPE	S	<alphanum 1..8>	
Name des übergeordneten Typs	varname#.SUPER-TYPE	S	'*NONE' / <alphanum 1..8>	
Name des Basis-Typs	varname#.BASE-TYPE	S	<alphanum 1..8>	
eingestellte Versionskonvention	varname#.CONVENTION	S	'*NONE' / '*STD-TREE' / '*STD-SEQ' / ' *MUL-SEQ'	
eingestelltes Versions-Beispiel (bei *STD-SEQ und *MUL-SEQ)	varname#.CONVENTION-EXAMPLE	S	'' / <comp-name 1..24>	
zulässige Speicherungsform	varname#.STOR-FORM	S	'*STD' / '*NONE' / '*FULL' / '*DELTA'	
Steuerung zusätzlicher Prüfungen	varname#.WRITE-CONTR	S	'*NONE' / '*ACTIVATE' / '*DEACTIVATE'	
Indikator für Verwaltungsschutz	varname#.ADM.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	
Name des Verwaltungs-Guards	varname#.ADM.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Verwaltungsberechtigung (BACL)	varname#.ADM.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Verwaltungsberechtigung des Eigentümers	varname#.ADM.OWNER	S	'' / '*OWNER'	*5
Indikator für Verwaltungsberechtigung der Gruppe des Eigentümers	varname#.ADM.GROUP	S	'' / '*GROUP'	*5
Indikator für Verwaltungsberechtigung für OTHERS	varname#.ADM.OTHERS	S	'' / '*OTHERS'	*5
Indikator, ob für das Element ein Verwaltungs-Kennwort vereinbart ist	varname#.ADM.PASS	S	'*NO' / '*YES'	*4
Indikator, ob Elementschutz vereinbart ist	varname#.PROT-DEFI	S	'*NONE' / '*PAR'	
Indikator für Leseschutz	varname#.PROT.READ-DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Lese-Guards	varname#.PROT.READ.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Leseberechtigung (BACL)	varname#.PROT.READ.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Leseberechtigung des Eigentümers	varname#.PROT.READ.OWNER	S	'' / '*OWNER'	*5
Indikator für Leseberechtigung der Gruppe des Eigentümers	varname#.PROT.READ.GROUP	S	'' / '*GROUP'	*5
Indikator für Leseberechtigung für OTHERS	varname#.PROT.READ.OTHERS	S	'' / '*OTHERS'	*5
Indikator, ob für das Element ein Lese-Kennwort vereinbart ist	varname#.PROT.READ.PASS	S	'*NO' / '*YES'	*4



Ausgabeinformation	Name der S-Variable	T	Inhalt	Bed.
Indikator für Schreibschutz	varname#.PROT.WRITE.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Schreib-Guards	varname#.PROT.WRITE.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Schreibberechtigung (BACL)	varname#.PROT.WRITE.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Schreibberechtigung des Eigentümers	varname#.PROT.WRITE.OWNER	S	' / '*OWNER'	*5
Indikator für Schreibberechtigung der Gruppe des Eigentümers	varname#.PROT.WRITE.GROUP	S	' / '*GROUP'	*5
Indikator für Schreibberechtigung für OTHERS	varname#.PROT.WRITE.OTHERS	S	' / '*OTHERS'	*5
Indikator, ob für das Element ein Schreib-Kennwort vereinbart ist	varname#.PROT.WRITE.PASS	S	'*NO' / '*YES'	*4
Indikator für Ausführungsschutz	varname#.PROT.EXEC.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Ausführungs-Guards	varname#.PROT.EXEC.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Ausführungsberechtigung (BACL)	varname#.PROT.EXEC.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Ausführungsberechtigung des Eigentümers	varname#.PROT.EXEC.OWNER	S	' / '*OWNER'	*5
Indikator für Ausführungsberechtigung der Gruppe des Eigentümers	varname#.PROT.EXEC.GROUP	S	' / '*GROUP'	*5
Indikator für Ausführungsberechtigung für OTHERS	varname#.PROT.EXEC.OTHERS	S	' / '*OTHERS'	*5
Indikator, ob für das Element ein Ausführungs-Kennwort vereinbart ist	varname#.PROT.EXEC.PASS	S	'*NO' / '*YES'	*4
Indikator für Ausleihschutz	varname#.PROT.HOLD.DEFI	S	'*NONE' / '*BY-GUARD' / '*PAR'	*2
Name des Ausleih-Guards	varname#.PROT.HOLD.GUARD-NAME	S	<filename 1..18>	*3
Indikator für Ausleihberechtigung (BACL)	varname#.PROT.HOLD.USER-DEFI	S	'*NONE' / '*LIST'	*4
Indikator für Ausleihberechtigung des Eigentümers	varname#.PROT.HOLD.OWNER	S	' / '*OWNER'	*5
Indikator für Ausleihberechtigung der Gruppe des Eigentümers	varname#.PROT.HOLD.GROUP	S	' / '*GROUP'	*5
Indikator für Ausleihberechtigung für OTHERS	varname#.PROT.HOLD.OTHERS	S	' / '*OTHERS'	*5
Indikator, ob für das Element ein Ausleih-Kennwort vereinbart ist	varname#.PROT.HOLD.PASS	S	'*NO' / '*YES'	*4

\*2 : varname#.PROT-DEFI = '\*PAR'

\*3 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*BY-GUARD'

\*4 : varname#.PROT-DEFI = '\*PAR' und varname#.xxxx.DEFI = '\*PAR'

\*5 : \*4 und varname#.xxxx.USER-DEFI = '\*LIST'



---

## 9 Beispiele

Das nachfolgende Kapitel zeigt anhand von Ablaufbeispielen einige typische Anwendungen von LMS.

In den Beispielen sind die Benutzereingaben durch Kleinbuchstaben und Fettdruck gekennzeichnet.

### 9.1 Aufnehmen, Korrigieren und Übersetzen von Quellprogrammen in Bibliotheken

Ein Quellprogramm wird als Element vom Typ S in einer Bibliothek abgelegt und übersetzt. Da beim Übersetzen Fehler gefunden wurden, wird das Element mit dem EDT korrigiert und anschließend wieder übersetzt. Der Modul aus dem EAM-Bereich wird in dieselbe Bibliothek mit dem Elementtyp R aufgenommen.

```
/start-lms _____ (1)
//modify-logging-parameters logging=*maximum _____ (2)
//modify-lms-defaults version=*increment _____ (3)
//open-library library=bsp1.bib,mode=*update _____ (4)
LIBRARY IS CLEARED AND PREPARED _____ (5)
```

- (1) LMS wird aufgerufen.
- (2) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (3) Die Zielversion wird bei jedem Schreiben um 1 erhöht.
- (4) Die Bibliothek BSP1.BIB wird als Bibliothek neu eingerichtet und als Ein- und Ausgabebibliothek zugewiesen.
- (5) Die Bibliothek BSP1.BIB ist eingerichtet.

```

//add-element from-file=quell.erfass,to-elem=(elem=erfass,type=s) ----- (6)
INPUT FILE
OUTPUT LIBRARY= :10SN:$USER.BSP1.BIB
      ADD :10SN:$USER.QUELL.ERFASS AS (S)ERFASS/001(0001)/2013-03-01 ----- (7)
//show-element-attributes ----- (8)
INPUT LIBRARY= :10SN:$USER.BSP1.BIB
TYP NAME VER (VAR#) DATE
(S) ERFASS 001 (0001) 2013-03-01
      1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS ----- (9)
//show-element (element=erfass, type=s) ----- (10)
INPUT LIBRARY= :10SN:$USER.BSP1.BIB
INPUT ELEMENT= (S)ERFASS/001(0001)/2013-03-01
      TITLE 'ERFASSEN VON DATEN'
      PRINT NOGEN
ERFAS START
      BALR 5,0
      USING *,5
      OPEN DATEI,OUTPUT
LESEN RDATA SATZ,ENDPGM
      PUT DATEI,SATZ
      B LESEN
ENDPGM TERM
*
DATEI FXB FCBTYPE=SAM, LINK=DATEN
SATZ DS CL84
      END
NUMBER OF PROCESSED RECORDS IS 14 ----- (11)
//end ----- (12)

```

- (6) Die Datei QUELL.ERFASS wird als Element ERFASS vom Typ S in die Bibliothek aufgenommen.
- (7) Erfolgsmeldung: Das Element ERFASS wurde mit der Versionsbezeichnung 001 und der Variantenummer 0001 in die Bibliothek geschrieben.
- (8) Das Inhaltsverzeichnis der Bibliothek BSP1.BIB soll aufgelistet werden.
- (9) Inhaltsverzeichniseintrag der Bibliothek BSP1.BIB.
- (10) Das Element ERFASS soll aufgelistet werden.
- (11) Inhalt des Elementes ERFASS.
- (12) LMS wird beendet.

```
/start-assembh _____ (13)
//compile source=*library-element(lib=bsp1.bib,element=erfass) _____ (14)
% ASS6011 ASSEMBLY TIME: 820 MSEC _____ (15)
% ASS6018 1 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: SIGNIFICANT ERROR
% ASS6006 LISTING GENERATOR TIME: 439 MSEC
//end
% ASS6012 END OF ASSEMBH
/start-lms _____ (16)
//modify-logging-parameters logging=*maximum _____ (17)
//modify-lms-defaults version=*increment _____ (18)
//open-library library=bsp1.bib,mode=*update _____ (19)
//edit-element (element=erfass,type=s) _____ (20)
```

- (13) Der ASSEMBLER wird aufgerufen.
- (14) Das Quellprogramm im Element ERFASS der Bibliothek BSP1.BIB soll übersetzt werden.
- (15) Das Programm ist fehlerhaft.
- (16) LMS wird wieder aufgerufen.
- (17) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (18) Die Zielversion wird bei jedem Schreiben um 1 erhöht.
- (19) Die Bibliothek BSP1.BIB wird zum Schreiben und Lesen eröffnet.
- (20) Das Element ERFASS soll mit dem EDT bearbeitet werden.

```

0.10      TITLE 'ERFASSEN VON DATEN'
0.20      PRINT NOGEN
0.30 ERFAS START
0.40      BALR 5,0
0.50      USING *,5
0.60      OPEN DATEI,OUTPUT
0.70 LESEN RDATA SATZ,ENDPGM
0.80      PUT DATEI,SATZ
0.90      B LESEN
1.00 ENDPGM TERM
1.10 *
x1.20 DATEI FcB FCBTYPE=SAM, LINK=DATEN
1.30 SATZ DS CL84
1.40 END
2.40
3.40
4.40
5.40
6.40
7.40
8.40
9.40

```

OUTPUT ELEMENT= (S)ERFASS/002(0001)/2013-03-01

halt

0000.10:001(0)

(21)

```
% LMS0420 EDITED ELEMENT (S)ERFASS/002(0002)/2013-03-01 TO BE ADDED? REPLY
(Y=YES; N=NO OR R=RETURN TO EDITOR)
```

y

```

INPUT LIBRARY= :10SN:$USER.BSP1.BIB
OUTPUT LIBRARY= :10SN:$USER.BSP1.BIB
INPUT ELEMENT= (S)ERFASS/001(0001)/2013-03-01
OUTPUT ELEMENT= (S)ERFASS/002(0002)/2013-03-01
CORRECT (S)ERFASS/001(0001)/2013-03-01 AS (S)ERFASS/002(0001)/2013-03-01

```

(22)

(23)

//end

(21) Der Fehler wird behoben:

- mit „x“ in der Anweisungsspalte Zeile 1.20 wird die Zeile auf überschreibbar gestellt;
- in der Zeile 1.20 wird „FcB“ in „FCB“ korrigiert und der EDT durch HALT in der Anweisungszeile beendet.

(22) Mit „Y“ wird das korrigierte Element in die Ausgabebibliothek mit Version 002 aufgenommen. Das alte Element (Version 001) bleibt erhalten.

(23) Erfolgsmeldung: Das Eingabeelement ERFASS wurde korrigiert. Das Ausgabeelement erhält denselben Namen und die Version 002.

```
/delete-system-file system-file=*omf
/start-assembh
//compile source=*library(lib=bsp1.bib,element=erfass)
% ASS6011 ASSEMBLY TIME: 1037 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 406 MSEC
//end
% ASS6012 END OF ASSEMBH _____ (24)
/start-lms _____ (25)
//modify-logging-parameters logging=*maximum _____ (26)
//open-library library=BSP1.BIB,mode=*update _____ (27)
//add-element from-file=*omf,to-elem=(type=r) _____ (28)
INPUT OMF
OUTPUT LIBRARY= :10SN:$USER.BSP1.BIB
      ADD ERFAS AS (R)ERFAS/@(0001)/2013-03-01
//end _____ (29)
```

- (24) Der Übersetzungslauf wird fehlerfrei ausgeführt.
- (25) LMS wird aufgerufen.
- (26) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (27) Die Bibliothek BSP1.BIB wird wiederum zum Lesen und Schreiben eröffnet.
- (28) Der Modul ERFAS wird aus dem EAM-Bereich als Element ERFAS abgelegt.
- (29) LMS wird beendet.

## 9.2 Kopieren von Elementen

Elemente aus unterschiedlichen Bibliotheken werden in eine Bibliothek kopiert.

```

/add-file-link link-name=lib1,file-name=modul.lib _____ (1)
/start-lms _____ (2)
//modify-logging-parameters logging=*maximum _____ (3)
//open-library library=bsp2.bib,mode=*update _____ (4)
LIBRARY IS CLEARED AND PREPARED
//show-element-attributes (library=*link(link-name=lib1),type=r) _____ (5)
INPUT LIBRARY= :10SN:$USER.MODUL.LIB,LINK=LIB1
TYP NAME VER (VAR#) DATE
(R) MODERF @ (0001) 2013-03-01
      1 (R)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//copy-element elem=(library=*link(link-name=lib1),elem=moderf,type=r), -
//      to-element=(library=*std,element=mod.erf) _____ (6)
INPUT LIBRARY= :10SN:$USER.MODUL.LIB,LINK=LIB1
OUTPUT LIBRARY= :10SN:$USER.BSP2.BIB
      COPY (R)MODERF/@(0001)/2011-02-19 AS (R)MOD.ERF/@(0001)/2011-02-19

```

- (1) Dadurch kann während des LMS-Laufs die Bibliothek MODUL.LIB über den Dateikettungsnamen LIB1 zugewiesen werden.
- (2) LMS wird aufgerufen.
- (3) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (4) Die Bibliothek BSP2.BIB wird als globale Bibliothek neu eingerichtet und zum Schreiben und Lesen eröffnet.
- (5) Das Inhaltsverzeichnis der Bibliothek MODUL.LIB, die über den Dateikettungsnamen LIB1 angegeben wird, soll aufgelistet werden.
- (6) Der Modul MODERF aus der Bibliothek MODUL.LIB wird in die Bibliothek BSP2.BIB mit dem Elementnamen MOD.ERF kopiert.



```

//show-element-attributes (library=macro.lib,type=m) ----- (7)
INPUT LIBRARY= :10SN:$USER.MACRO.LIB
TYP NAME VER (VAR#) DATE NAME VER (VAR#) DATE
(M) MAC1 @ (0001) 2011-02-19 MAC2 @ (0001) 2011-02-19
2 (M)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//copy-element elem=(library=macro.lib,elem=mac*,type=m), -
// to-element=(library=*std,element=mu*) ----- (8)
INPUT LIBRARY= :10SN:$USER.MACRO.LIB
OUTPUT LIBRARY= :10SN:$USER.BSP2.BIB
COPY (M)MAC1/@(0001)/2011-02-19 AS (M)MU1/@(0001)/2011-02-19
COPY (M)MAC2/@(0001)/2011-02-19 AS (M)MU2/@(0001)/2011-02-19 ----- (9)
//show-element-attributes (library=quell.lib,type=s) ----- (10)
INPUT LIBRARY= :10SN:$USER.QUELL.LIB
TYP NAME VER (VAR#) DATE NAME VER (VAR#) DATE
(S) EDTB @ (0001) 2011-02-19 PROT @ (0001) 2011-02-19
(S) SEINAUS @ (0001) 2011-02-19 SERFAS @ (0001) 2011-02-19
4 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//copy-element elem=(library=quell.lib,elem=*,type=s, -
// except-element=(element=edtb)) ----- (11)
INPUT LIBRARY= :10SN:$USER.QUELL.LIB
OUTPUT LIBRARY= :10SN:$USER.BSP2.BIB
COPY (S)PROT/@(0001)/2011-02-19 AS (S)PROT/@(0001)/2011-02-19
COPY (S)SEINAUS/@(0001)/2011-02-19 AS (S)SEINAUS/@(0001)/2011-02-19
COPY (S)SERFAS/@(0001)/2011-02-19 AS (S)SERFAS/@(0001)/2011-02-19

```

- (7) Das Inhaltsverzeichnis der Bibliothek MACRO.LIB, die als lokale Bibliothek zugewiesen wird, soll aufgelistet werden.
- (8) Diejenigen Elemente der Bibliothek MACRO.LIB werden in die Bibliothek BSP2.BIB kopiert, deren Elementbezeichnung mit "MAC" beginnt. Die neuen Elementbezeichnungen beginnen mit "MU". Ab der dritten Stelle werden die Elementbezeichnungen der Eingabeelemente übernommen.
- (9) Erfolgsmeldung: Die ausgewählten Elemente wurden mit den neuen Elementbezeichnungen kopiert.
- (10) Das Inhaltsverzeichnis der Bibliothek QUELL.LIB, die als lokale Bibliothek zugewiesen wird, soll aufgelistet werden.
- (11) Alle Elemente der Bibliothek QUELL.LIB mit Ausnahme des Elementes EDTB werden in die Ausgabebibliothek kopiert.

```
//show-element-attributes (library=test.lib) ----- (12)
INPUT  LIBRARY= :10SN:$USER.TEST.LIB
TYP NAME  VER (VAR#) DATE
(S) SERFAS @ (0001) 2011-02-19
      1 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//copy-element elem=(library=test.lib,elem=*,type=*), -
//            to-elm=(library=*std,element=*(version=007)) ----- (13)
INPUT  LIBRARY= :10SN:$USER.TEST.LIB
OUTPUT LIBRARY= :10SN:$USER.BSP2.BIB
      COPY (S)SERFAS/@(0001)/2011-02-19 AS (S)SERFAS/007(0001)/2011-02-19
```

- (12) Das Inhaltsverzeichnis der Programmbibliothek TEST.LIB, die als lokale Bibliothek zugewiesen wird, soll aufgelistet werden.
- (13) Alle Elemente der Programmbibliothek TEST.LIB werden in die Ausgabebibliothek kopiert und dort unter ihrem Namen und der Versionsnummer 007 abgelegt.

```

//show-element-attributes ----- (14)
INPUT LIBRARY= :10SN:$USER.BSP2.BIB
TYP NAME VER (VAR#) DATE          NAME VER (VAR#) DATE
(M) MU1 @ (0001) 2011-02-19  MU2 @ (0001) 2011-02-19
      2 (M)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYP NAME      VER (VAR#) DATE
(R) MOD.ERF @ (0001) 2011-02-19
      1 (R)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYP NAME      VER (VAR#) DATE          NAME      VER (VAR#) DATE
(S) PROT      @ (0001) 2011-02-19  SEINAUS @ (0001) 2011-02-19
(S) SERFAS 007 (0001) 2011-02-19  SERFAS @ (0001) 2011-02-19
      4 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
-----
      7 ELEMENT(S) IN THIS TABLE OF CONTENTS
//show-library-status ----- (15)
STATUS FILENAME                                MODE    LINK    DEF-TYPE
CLOSED :10SN:$USER.TEST.LIB
CLOSED :10SN:$USER.QUELL.LIB
CLOSED :10SN:$USER.MACRO.LIB
CLOSED :10SN:$USER.MODUL.LIB                                LIB1
OPEN   :10SN:$USER.BSP2.BIB                                UPDATE
//close-library ----- (16)
//show-library-status ----- (17)
STATUS FILENAME                                MODE    LINK    DEF-TYPE
CLOSED :10SN:$USER.TEST.LIB
CLOSED :10SN:$USER.QUELL.LIB
CLOSED :10SN:$USER.MACRO.LIB
CLOSED :10SN:$USER.MODUL.LIB                                LIB1
CLOSED :10SN:$USER.BSP2.BIB
//end ----- (18)

```

- (14) Das Inhaltsverzeichnis der aktuellen Eingabebibliothek BSP2.BIB soll mit allen Elementen aufgelistet werden. Es existieren 2 Elemente mit dem Elementnamen SERFAS mit unterschiedlicher Versionsnummer.
- (15) Der Zustand der im LMS-Lauf verwendeten Bibliotheken wird abgefragt.
- (16) Die Bibliothek BSP2.BIB wird geschlossen.
- (17) Der Zustand der im LMS-Lauf verwendeten Bibliotheken wird abgefragt.
- (18) Der LMS-Lauf wird beendet.

### 9.3 Vergleichen von Elementen

Das Element ERFASS (aufgelistet im Beispiel auf [Seite 468](#)) und das Element EINAUS werden miteinander verglichen. Ein Vergleichsprotokoll wird erstellt.

```
/start-lms _____ (1)
//modify-logging-parameters logging=*maximum _____ (2)
//open-library library=bsp1.bib,mode=*update _____ (3)
//add-element from-file=quell.einaus, to-elem=(elem=einaus,type=s) _____ (4)
INPUT FILE
OUTPUT LIBRARY= :10SN:$USER.BSP1.BIB
          ADD :10SN:$USER.QUELL.EINAUS AS (S)EINAUS/@(0001)/2013-03-01
```

- (1) LMS wird aufgerufen.
- (2) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (3) Die Bibliothek BSP1.BIB wird zum Lesen und Schreiben eröffnet.
- (4) Die Datei QUELL.EINAUS wird als Element EINAUS vom Typ S in die Bibliothek aufgenommen.

```
//show-element (element=einaus,type=s) ----- (5)
INPUT  LIBRARY= :10SN:$USER.BSP1.BIB
INPUT  ELEMENT= (S)ES@(0001)/2013-03-01
        TITLE 'ERFASSEN VON DATEN'
        PRINT NOGEN
ERFAS  START
        BALR 5,0
        USING *,5
        OPEN DATEI,OUTPUT
LESEN  RDATA SATZ,ENDPGM
        CLC  TEXT(4),=C'/EOF'
        BE  ENDPGM
        MVC  ATEXT,TEXT
        LH  9,SL
        AH  9,=H'1'
        STH 9,ASL
        WROUT ASATZ,ENDPGM
        PUT  DATEI,SATZ
        B   LESEN
ENDPGM TERM
*
DATEI  FCB  FCBTYPE=SAM,LINK=DATEN
        DS  0H
SATZ   DS  CL84
SL     DS  CL2
        DS  CL2
TEXT   DS  CL80
ASATZ  DS  OCL85
ASL    DS  CL2
        DC  X'000001'
ATEXT  DS  CL80
        END
NUMBER OF PROCESSED RECORDS IS      29
```

(5) Das Element EINAUS wird aufgelistet.

```

//show-element (element=erfass,type=s) ----- (6)
INPUT  LIBRARY= :10SQ:$USER.BSP1.BIB
INPUT  ELEMENT= (S)ERFASS/002(0001)/2013-03-01
        TITLE 'ERFASSEN VON DATEN'
        PRINT NOGEN
ERFAS  START
        BALR 5,0
        USING *,5
        OPEN DATEI,OUTPUT
LESEN  RDATA SATZ,ENDPGM
        PUT DATEI,SATZ
        B LESEN
ENDPGM TERM
*
DATEI  FCB FCBTYPE=SAM, LINK=DATEN
SATZ   DS CL84
        END
NUMBER OF PROCESSED RECORDS IS 14

//compare-element primary-elim=(elem=einaus,type=s), -
//               secondary-elim=(elem=erfass), -
//               compare-parameters = *par(information=*maximum) ----- (7)
FUNCTION = C O M P A R E
PAR      COMPARE= 00001/32764/L/MAX
PRIMARY  LIBRARY= :10SN:$USER.BSP1.BIB
PRIMARY  ELEMENT= (S)EINAUS/@(0001)/2013-03-01
SECONDARY LIBRARY= :10SN:$USER.BSP1.BIB
SECONDARY ELEMENT= (S)ERFASS/002(0001)/2013-03-01 ----- (8)
-----
SAME FROM #1 TO #7 AS FROM #1 TO #7 ----- (9)
#1 > TITLE 'ERFASSEN VON DATEN'<
#2 > PRINT NOGEN<
#3 >ERFAS START<
#4 > BALR 5,0<
#5 > USING *,5<
#6 > OPEN DATEI,OUTPUT
#7 >LESEN RDATA SATZ,ENDPGM<

```

- (6) Das Element ERFASS wird aufgelistet
- (7) Die Elemente EINAUS und ERFASS werden miteinander verglichen. Das Vergleichsprotokoll soll in vollem Umfang ausgegeben werden.
- (8) Beginn des Vergleichsprotokolls:  
Protokolliert werden die eingestellten Werte von COMPARE, die Namen von Primär- und Sekundärbibliothek und von Primär- und Sekundärelement.
- (9) Die Sätze mit der Satzkenennung #1 bis #7 sind in beiden Elementen gleich.

---

INS. FROM	#8 TO	#14	_____	(10)
	#8 >	CLC	TEXT(4),=C'/EOF'<	
	#9 >	BE	ENDPGM<	
	#10 >	MVC	ATEXT,TEXT<	
	#11 >	LH	9,SL<	
	#12 >	AH	9,=H'1'<	
	#13 >	STH	9,ASL<	
	#14 >	WROUT	ASATZ,ENDPGM< _____	(11)

---

SAME FROM	#15 TO	#19 AS FROM	#8 TO	#12	_____	(12)
	#15 >	PUT	DATEI,SATZ <			
	#16 >	B	LESEN<			
	#17 >	ENDPGM	TERM<			
	#18 >	*<				
	#19 >	DATEI	FCB FCBTYPE=SAM, LINK=DATEN< _____			(13)

---

INS.	#20	_____	(14)
	#20 >	DS OH< _____	(15)

---

## (08) –(21) Vergleichsprotokoll

- (10) Die Sätze #8 bis #14 sind nur im Primärelement vorhanden und werden als INS(erted) dargestellt.
- (11) Die eingefügten Sätze werden ausgegeben.
- (12) Die Sätze #15 bis #19 des Primärelementes sind gleich den Sätzen #8 bis #12 des Sekundärelementes.
- (13) Die gleichen Sätze werden ausgegeben.
- (14) Der Satz #20 ist nur im Primärelement vorhanden und wird als INS(erted) dargestellt.
- (15) Der eingefügte Satz wird ausgegeben.

```

SAME          #21          AS          #13 _____ (16)
              #21 >SATZ      DS      CL84< _____ (17)
-----
INS. FROM     #22 TO      #28 _____ (18)
              #22 >SL        DS      CL2<
              #23 >          DS      CL2<
              #24 >TEXT      DS      CL80<
              #25 >ASATZ     DS      OCL85<
              #26 >ASL       DS      CL2<
              #27 >          DC      X'000001'<
              #28 >ATEXT     DS      CL80< _____ (19)
-----
SAME          #29          AS          #14 _____ (20)
              #29 >          END<    _____ (21)
-----
PRIMARY      ELEMENT= (S)EINAUS/@(0001)/2013-03-01
SECONDARY    ELEMENT= (S)ERFASS/002(0001)/2013-03-01
RESULT: C    PRIMARY=     29 INSERTED=    15 (   3) DELETED=     0 (   0)
              SECONDARY=    14   SAME=     14 (   4) _____ (22)
//end _____ (23)

```

- (16) Der Satz #21 des Primärelementes ist gleich dem Satz #13 des Sekundärelementes.
- (17) Der gleiche Satz wird ausgegeben.
- (18) Die Sätze #20 bis #28 sind nur im Primärelement vorhanden und werden als INS(erted) dargestellt.
- (19) Die eingefügten Sätze werden ausgegeben.
- (20) Der Satz #29 des Primärelementes ist gleich dem Satz #14 des Sekundärelementes.
- (21) Der gleiche Satz wird ausgegeben.
- (22) Ergebnis des Vergleichs; ausgegeben wird die Anzahl der Sätze des Primärelementes, des Sekundärelementes, der eingefügten, gleichen und gelöschten Sätze. Die Zahl in Klammern gibt an, wieviel zusammenhängende Teile eingefügt, gleich oder gelöscht sind.
- (23) LMS wird beendet.



## 9.4 Arbeiten mit Delta-Elementen

```

/start-lms _____ (1)
//modify-logging-parameters logging=*maximum _____ (1)
//open-library library=bsp4.bib,mode=*update _____ (2)
LIBRARY IS CLEARED AND PREPARED
//add-element from-file=workelem,to-elm=(elm=delta(v=v00),type=s, -
//      storage-form=*delta) _____ (3)
INPUT FILE
OUTPUT LIBRARY= :10SN:$USER.BSP4.BIB
      ADD :10SN:$USER.WORKELEM AS (S)DELTA/V00(0001)/2013-03-01
      , FIRST DELTA VERSION
//edit-element (,element=delta(version=v00),type=s), -
//      to-elm=(,elm=delta(v=v01)) _____ (4)

```

```

0.10 MINI      START
0.20          BALR  3,0
0.30          USING *.3
0.40          OPEN  SAMFCB,OUTPUT
0.50          PUT   SAMFCB,EINGABE
0.60          TERM
0.70 SAMFCB   FCB   FCBTYPE=SAM,LINK=MINI
0.80 EINGABE DC   C'DA IST ER JA
0.90          END
1.90
2.90
3.90
4.90
5.90
6.90
7.90
8.90
9.90
10.90
11.90
12.90
13.90
          OUTPUT ELEMENT= (S)DELTA/V01(0002)/2013-03-01
halt _____ 0000.10:001(0)

```

- (1) LMS wird aufgerufen.
- (2) Die Bibliothek BSP4.BIB wird neu eingerichtet.
- (3) Die Datei WORKELEM wird als neues Delta-Element DELTA/V00 vom Typ S in die Bibliothek aufgenommen.
- (4) Das Delta-Element DELTA/V00 soll mit dem EDT bearbeitet und das Ergebnis als neues Element DELTA/V01 an DELTA/V00 angehängt werden.

```

% LMS0420 EDITED ELEMENT (S)DELTA/V01(0002)/2013-03-01 TO BE ADDED OR RETURN TO EDITOR ?
REPLY (Y=YES; N=NO OR R=RETURN)?

```

```

y
INPUT  LIBRARY= :10SN:$USER.BSP4.BIB
OUTPUT LIBRARY= :10SN:$USER.BSP4.BIB
INPUT  ELEMENT= (S)DELTA/V00(0001)/2013-03-01
OUTPUT ELEMENT= (S)DELTA/V01(0002)/2013-03-01
          CORRECT (S)DELTA/V00(0001)/2013-03-01 AS (S)DELTA/V01(0002)/2011-10-01
          ON BASE (S)DELTA/V00(0002)/2013-03-01
//copy-element (library=lib.arbeit,elem=input,type=s), -
//            to-elm=(elem=delta(version=v02,base=v00)) ----- (5)
INPUT  LIBRARY= :10SN:$USER.LIB.ARBEIT
OUTPUT LIBRARY= :10SN:$USER.BSP4.BIB
          COPY (S)INPUT/@(0001)/2011-02-19 AS (S)DELTA/V02(0003)/2011-02-19
          ON BASE (S)DELTA/V00(0003)/2011-02-19
//show-element-attributes information=*delta-structure ----- (6)
INPUT  LIBRARY= :10SN:$USER.BSP4.BIB
TYP     NAME                               VERSION   (VAR#) DATE       DLT#  BASE#
(S      ) DELTA . . . . . V00 . . . (0000)       00001 00000
(S      ) DELTA . . . . . V01 . . . (0000)       00002 00001
(S      ) DELTA . . . . . V02 . . . (0000)       00003 00001

```

3 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS

- (5) Das Element INPUT wird aus der Bibliothek LIB.ARBEIT als Delta-Element DELTA/V02 vom Typ S zur Basis V00 in die globale Bibliothek kopiert.
- (6) Das Inhaltsverzeichnis der Bibliothek BSP4.BIB als kompletter Delta-Baum soll aufgelistet werden.

```
//add-element workelem,(,delta(v11,v01),s,,delta) ----- (7)
```

```
INPUT FILE
```

```
OUTPUT LIBRARY= :10SN:$USER.BSP4.BIB
```

```
ADD :10SN:$USER.WORKELEM AS (S)DELTA/V11(0004)/2013-03-01 ON BASE
```

```
(S)DELTA/V01(0004)/2011-10-01
```

```
//show-element-attributes information=*delta-structure ----- (8)
```

```
INPUT LIBRARY= :10SN:$USER.BSP4.BIB
```

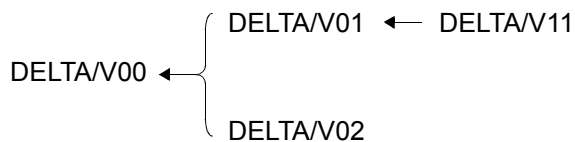
TYP	NAME	VERSION	(VAR#)	DATE	DLT#	BASE#
(S	) DELTA . . . . .	V00 . . . . .	(0000)		00001	00000
(S	) DELTA . . . . .	V01 . . . . .	(0000)		00002	00001
(S	) DELTA . . . . .	V02 . . . . .	(0000)		00003	00001
(S	) DELTA . . . . .	V11 . . . . .	(0000)		00004	00002

```
4 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
```

```
//end ----- (9)
```

- (7) Die Datei WORKELEM wird als Delta-Element DELTA/V11 vom Typ S zum Delta-Element DELTA/V01 aufgenommen. Diese Anweisung ist ein Beispiel für eine Eingabe nur mit Stellungsoperanden.
- (8) Das Inhaltsverzeichnis der Bibliothek BSP4.BIB als Delta-Baum soll aufgelistet werden.
- (9) LMS wird beendet.

Die Delta-Elemente haben nun folgende Zuordnung:



## 9.5 Ändern eines Bindemoduls

Das Element USELST wird als Bindemodul in die Bibliothek aufgenommen und dort mit einer Subanweisung geändert.

```

/start-assembh _____ (1)
//compile source=*library-element(lib=bsp5.bib,element=uselst), -
//module-lib=bsp5.bib(element=uselst) _____ (2)
% ASS6011 ASSEMBLY TIME: 240 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 453 MSEC
//end
% ASS6012 END OF ASSEMBH
/start-lms _____ (3)
//modify-logging-parameters logging=*maximum _____ (4)
//open-library library=bsp5.bib,mode=*update _____ (5)
//show-element-attributes (type=r) _____ (6)
INPUT LIBRARY= :10SN:$USER.BSP5.BIB
TYP NAME VER (VAR#) DATE
(R) USELST @ (0001) 2013-03-01
      1 (R)-ELEMENT(S) IN THIS TABLE OF CONTENTS

```

- (1) Aufruf des ASSEMBLERS für die Erzeugung eines Bindemoduls.
- (2) Das Quellprogramm USELST soll übersetzt werden. Bei fehlerfreier Übersetzung wird der so erzeugte Bindemodul in die Bibliothek BSP5.BIB geschrieben. Der Bindemodul erhält automatisch den Elementtyp R.
- (3) LMS wird aufgerufen.
- (4) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (5) Die Bibliothek BSP5.BIB wird zugewiesen. Sie muss zum Lesen und Schreiben geöffnet werden, andernfalls ist keine Änderung des Bindemoduls möglich.
- (6) Anzeige des Inhaltsverzeichnis für alle Elemente des Typs R. Die Bibliothek enthält jetzt den aus dem ASSEMBLER-Lauf erzeugten Bindemodul USELST.

```

//modify-element (element=uselst,type=r) _____ (7)
//add-text-modification address=x'c0',new-contents='aa'(old='ER') _____ (8)
//end-modify _____ (9)
INPUT LIBRARY= :10SN:$USER.BSP5.BIB
OUTPUT LIBRARY= :10SN:$USER.BSP5.BIB
INPUT ELEMENT= (R)USELST/@(0001)/2013-03-01
OUTPUT ELEMENT= (R)USELST/@(0002)/2013-03-01
TEXT-ADR:          00000C0
TEXT BEFORE CHANGE: E R
                  C5D9
TEXT AFTER CHANGE: a a
                  8181 _____ (10)
                  CORRECT (R)USELST/@(0001)/2013-03-01 AS (R)USELST/@(0002)/2013-03-01
                  , OUTPUT REPLACED _____ (11)
//show-element-attributes (type=r) _____ (12)
INPUT LIBRARY= :10SN:$USER.BSP5.BIB
TYP NAME VER (VAR#) DATE
(R) USELST @ (0002) 2013-03-01
      1 (R)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//end _____ (13)

```

- (7) Der Modul USELST soll geändert werden.
- (8) Diese Subanweisung ersetzt an der Adresse c0 den Text 'ER' durch 'aa'.
- (9) Die Eingabe der Subanweisungen wird beendet.
- (10) Der Text der Adresse 0000C0 wird ersetzt.
- (11) Meldung, dass die Änderung ausgeführt ist und das ursprüngliche Element durch das geänderte Element ersetzt wird.
- (12) Anzeige des Inhaltsverzeichnis für alle Elemente des Typs R. Die Bibliothek enthält jetzt den geänderten Bindemodul.
- (13) LMS wird beendet.

## 9.6 Erzeugen von SAM-/ISAM-Dateien

Mit dem EDT wird eine SAM-Datei erzeugt. Diese wird als Element aufgenommen und auf zwei verschiedene Arten als Datei gespeichert:

- als SAM-Datei, auf Grund der im Element gespeicherten Dateieigenschaften;
- als ISAM-Datei mit standardmäßig erzeugten ISAM-Schlüsseln.

```
/start-lms _____ (1)
```

```
//modify-logging-parameters logging=*maximum _____ (2)
```

```
//open-library library=bsp6.bib,mode=*update _____ (3)
```

```
LIBRARY IS CLEARED AND PREPARED
```

```
//modify-lms-defaults type=d _____ (4)
```

- (1) LMS wird aufgerufen.
- (2) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (3) Die Bibliothek BSP6.BIB wird neu eingerichtet.
- (4) Der Defaultwert für den Elementtyp wird eingestellt werden.

```
//call-edt _____ (5)
```

1.00	BACH	SEBASTIAN	MUENCHEN	AUF DER HOEHE 7	AB 3
2.00	BERGMANN	NORBERT	MUENCHEN	TORWEG 10	AB 5
3.00	FINK	SUSANNE	NUERNBERG	RINGSTR. 23	AB 1
4.00	MEYER	FRANZ	NUERNBERG	WASSERMUNGENWEG	AB 1
5.00	GRUNDLER	WOLFGANG	BASEL	SONNENSTR. 11	AB 2
6.00	KNOLL	MONIKA	FRANKFURT	BAUMALLEE 12	AB 3
7.00	LIEDL	ERIKA	MUENCHEN	IN DER BREITE 1	AB 5
8.00	WAGNER	JOHANN	AUGSBURG	AM SEE 45	AB 4
9.00					
10.00					
11.00					
12.00					
13.00					
14.00					
15.00					
16.00					
17.00					
18.00					
19.00					
20.00					
21.00					
22.00					
23.00					

```
write 'pers.dat';halt
```

```
0000.10:001(0)
```

```
//modify-lms-defaults (source-attributes=*keep) _____ (6)
//add-element from-file=pers.dat, to-elm=(,perdat) _____ (7)
INPUT FILE
OUTPUT LIBRARY= :10SN:$USER.BSP6.BIB
        ADD :10SN:$USER.PERS.DAT AS (D)PERDAT/@(0001)/2013-03-01
```

- (5) Es wird der EDT aufgerufen, um eine Datei zu erstellen oder zu bearbeiten.  
Anschließend werden die Daten eingegeben und mit WRITE als SAM-Datei PERS.DAT gespeichert. Mit HALT wird der EDT beendet und zu LMS zurückgekehrt.
- (6) Die Dateieigenschaften der EDT-Datei werden beibehalten.
- (7) Die Datei PERS.DAT wird als Element PERDAT vom Typ D in die Bibliothek aufgenommen.

```
//show-element-attributes (type=d)
INPUT LIBRARY= :10SN:$USER.BSP6.BIB
TYP NAME VER (VAR#) DATE
(D) PERDAT @ (0001) 2013-03-01
    1 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//extract-element (element=perdat) _____ (8)
INPUT LIBRARY= :10SN:$USER.BSP6.BIB
OUTPUT FILE
        EXTRACT (D)PERDAT/@(0001)/2013-03-01 AS :10SN:$USER.PERDAT
//exec-sys-cmd (show-file-attributes perdat,information=*all-attr) _____ (9)
```

```

00000003 :10SN:$USER.PERDAT
----- HISTORY -----
CRE-DATE   = 2013-03-01  ACC-DATE   = 2013-03-01  CHANG-DATE = 2013-03-01
CRE-TIME   = 10:32:56   ACC-TIME   = 10:32:56   CHANG-TIME = 10:32:56
ACC-COUNT  = 1          S-ALLO-NUM = 0
----- SECURITY -----
READ-PASS  = NONE       WRITE-PASS  = NONE       EXEC-PASS  = NONE
USER-ACC   = OWNER-ONLY ACCESS        = WRITE        ACL         = NO
AUDIT      = NONE       FREE-DEL-D  = *NONE       EXPIR-DATE = 2013-03-01
DESTROY    = NO         FREE-DEL-T  = *NONE       EXPIR-TIME = 00:00:00
SP-REL-LOCK= NO         ENCRYPTION  = *NONE
----- BACKUP -----
BACK-CLASS = A          SAVED-PAG  = COMPL-FILE  VERSION    = 1
MIGRATE    = ALLOWED
----- ORGANIZATION -----
FILE-STRUC = SAM        BUF-LEN    = STD(1)      BLK-CONTR  = PAMKEY
IO(USAGE)  = READ-WRITE IO(PERF)   = STD              DISK-WRITE = IMMEDIATE
REC-FORM   = (V,N)     REC-SIZE   = 0
AVAIL      = *STD
WORK-FILE  = *NO       F-PREFORM  = *K          SO-MIGR    = *ALLOWED
----- ALLOCATION -----
SUPPORT    = PUB        S-ALLOC    = 34          HIGH-US-PA = 1
EXTENTS    VOLUME      DEVICE-TYPE EXTENTS    VOLUME     DEVICE-TYPE
  1         10SN.2      D34211-2
NUM-OF-EXT = 1
:10SN: PUBLIC:      1 FILE RES=      3 FREE=      2 REL=      0 PAGES

```

- (8) Das Inhaltsverzeichnis der Bibliothek BSP6.BIB für den Elementtyp D soll aufgelistet werden.
- (9) Das Element PERDAT wird als Datei PERDAT ausgegeben. Da für diese Datei keine Dateieigenschaften spezifiziert wurden, erzeugt LMS entsprechend den gespeicherten Dateieigenschaften eine SAM-Datei.
- (10) Die Dateieigenschaften der erzeugten Datei werden aufgelistet.

```

//extract-element (element=perdat), to-file=persdat, -
//          file-attributes=*parameters(access-method=*isam)
INPUT LIBRARY= :10SN:$USER.BSP6.BIB
OUTPUT FILE
          EXTRACT (D)PERDAT/@(0001)/2013-03-01 AS :10SN:$USER.PERSDAT
//exec-sys-cmd (show-file-attributes persdat,information=all) ----- (10)

```



```

00000003 :10SN:$USER.PERSDAT
----- HISTORY -----
CRE-DATE   = 2013-03-01  ACC-DATE   = 2013-03-01  CHANG-DATE = 2013-03-01
CRE-TIME   = 10:32:56   ACC-TIME   = 10:32:56   CHANG-TIME = 10:32:56
ACC-COUNT  = 1          S-ALLO-NUM = 0
----- SECURITY -----
READ-PASS  = NONE       WRITE-PASS  = NONE       EXEC-PASS  = NONE
USER-ACC   = OWNER-ONLY ACCESS       = WRITE       ACL        = NO
AUDIT      = NONE       FREE-DEL-D  = *NONE      EXPIR-DATE = 2013-03-01
DESTROY    = NO         FREE-DEL-T  = *NONE      EXPIR-TIME = 00:00:00
SP-REL-LOCK= NO        ENCRYPTION  = *NONE
----- BACKUP -----
BACK-CLASS = A          SAVED-PAG  = COMPL-FILE  VERSION    = 1
MIGRATE    = ALLOWED
----- ORGANIZATION -----
FILE-STRUC = ISAM      BUF-LEN    = STD(1)     BLK-CONTR  = PAMKEY
IO(USAGE)  = READ-WRITE IO(PERF)   = STD           DISK-WRITE = IMMEDIATE
REC-FORM   = (V,N)    REC-SIZE   = 0
KEY-LEN    = 8        KEY-POS    = 5
AVAIL      = *STD
WORK-FILE  = *NO      F-PREFORM  = *K        SO-MIGR    = *ALLOWED
----- ALLOCATION -----
SUPPORT    = PUB       S-ALLOC    = 9          HIGH-US-PA = 3
EXTENTS    VOLUME     DEVICE-TYPE EXTENTS    VOLUME     DEVICE-TYPE
1          10SN.1     D34211-2
NUM-OF-EXT = 1
:10SN: PUBLIC:      1 FILE RES=      3 FREE=      1 REL=      0 PAGES
//end ----- (11)

```

- (11) Das Element PERDAT wird als Datei mit dem Namen PERSDAT angelegt. Durch den Operanden ACCESS-METHOD wird nun festgelegt, dass die Datei PERSDAT eine ISAM-Datei ist.
- (12) Die Dateieigenschaften der erzeugten Datei werden aufgelistet.
- (13) LMS wird beendet.

## 9.7 Ausgeben von Vergleichsstatistiken

Alle Elemente mehrerer Bibliotheken werden miteinander verglichen. Es wird anschließend an den Vergleich die jeweilige Vergleichsstatistik und am Ende die Gesamtsumme der im LMS-Lauf durchgeführten Vergleiche ausgegeben.

Dieser LMS-Lauf wird ohne globale Bibliothek durchgeführt, d.h. es wird kein OPEN-LIBRARY angegeben. Alle benötigten Bibliotheken werden lokal in Anweisungen vereinbart.

```

/start-lms _____ (1)
//modify-logging-parameters logging=*maximum _____ (2)
//compare-element primary-elem=(lib=lib.all.v2,element=*,type=s), -
//      secondary-elem=(lib=lib.sou.v1,element=*,type=s), -
//      compare-parameters = (information=*summary) _____ (3)
FUNCTION          = C O M P A R E
PAR              COMPARE= 00001/32764/L/SUM
PRIMARY          LIBRARY= :10SQ:$USER.LIB.ALL.V2
SECONDARY        LIBRARY= :10SQ:$USER.LIB.SOU.V1
-----
PRIMARY          ELEMENT= (S)EINAUS/@(0001)/2011-02-19
SECONDARY        ELEMENT= (S)EINAUS/@(0001)/2011-02-19
RESULT: S        PRIMARY=      8  INSERTED=      - (    -)  DELETED=      - (    -)
                  SECONDARY=      8      SAME=      8 (    1)
-----
PRIMARY          ELEMENT= (S)ERFASS/@(0002)/2011-02-19
SECONDARY        ELEMENT= (S)ERFASS/@(0001)/2011-02-19
RESULT: S        PRIMARY=     14  INSERTED=      - (    -)  DELETED=      - (    -)
                  SECONDARY=     14      SAME=     14 (    1)
-----
PRIMARY          ELEMENT= (S)PROT/@(0001)/2011-02-19
SECONDARY        ELEMENT= (S)PROT/@(0001)/2011-02-19
RESULT: S        PRIMARY=      5  INSERTED=      - (    -)  DELETED=      - (    -)
                  SECONDARY=      5      SAME=      5 (    1) _____ (4)

```

- (1) LMS wird aufgerufen.
- (2) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (3) Alle Elemente vom Typ S der Bibliothek LIB.ALL.V2 werden mit den Elementen der Bibliothek LIB.SOU.V1 verglichen.
- (4) Das Ergebnis des Vergleichs wird ausgegeben.

```
//show-statistics number=*c0
```

STATISTIC	PRIM. ELEM.	PRIM. LINES	INS. LINES	SAME LINES	DEL. LINES	INS+DEL LINES	SEC. LINES	SEC. ELEM.
S (SAME)	3	27	-	27	-	-	27	3
C (CHANGED)	0	0	0	0	0	0	0	0
I (INSERTED)	0	0	0	-	-	0	-	-
D (DELETED)	-	-	-	-	0	0	0	0
TOTAL	3	27	0	27	0	0	27	3

```
//compare-element primary-elem=(lib=lib.all.v2,element=*,type=m), -
// secondary-elem=(lib=lib.mac.v1,element=*,type=m), -
// compare-param =(information=*summary)
```

```
FUNCTION = C O M P A R E
PAR COMPARE= 00001/32764/L/SUM
PRIMARY LIBRARY= :10SQ:$USER.LIB.ALL.V2
SECONDARY LIBRARY= :10SQ:$USER.LIB.MAC.V1
```

```
PRIMARY ELEMENT= (M)MAC1/@(0001)/2011-02-19
SECONDARY ELEMENT= (M)MAC1/@(0001)/2011-02-19
RESULT: S PRIMARY= 5 INSERTED= - ( -) DELETED= - ( -)
SECONDARY= 5 SAME= 5 ( 1)
```

```
PRIMARY ELEMENT= (M)MAC2/@(0001)/2011-02-19
SECONDARY ELEMENT= (M)MAC2/@(0001)/2011-02-19
RESULT: S PRIMARY= 5 INSERTED= - ( -) DELETED= - ( -)
SECONDARY= 5 SAME= 5 ( 1)
```

```
PRIMARY ELEMENT= (M)MUC1/@(0001)/2011-02-19
RESULT: I PRIMARY= 5 INSERTED= 5 ( 1) DELETED= - ( -)
SECONDARY= - SAME= - ( -)
```

```
PRIMARY ELEMENT= (M)MUC2/@(0001)/2011-02-19
RESULT: I PRIMARY= 5 INSERTED= 5 ( 1) DELETED= - ( -)
SECONDARY= - SAME= - ( -)
```

- (5) Die aktuelle Vergleichsstatistik, die im Bereich C0 steht, wird ausgegeben. Danach führt LMS intern folgende Aktionen durch:
  - Der Bereich C0 wird auf den Bereich C1 aufaddiert.
  - Der Bereich C0 wird wieder gelöscht, um das Ergebnis des nächsten Vergleichs aufzunehmen.
- (6) Alle Elemente vom Typ M der Bibliothek LIB.ALL.V2 werden mit den Elementen der Makrobibliothek LIB.MAC.V1 verglichen.
- (7) Das Ergebnis des Vergleichs wird ausgegeben.

```
//show-statistics number=*c0 ----- (8)
```

```
AREA C0
```

STATISTIC	PRIM. ELEM.	PRIM. LINES	INS. LINES	SAME LINES	DEL. LINES	INS+DEL LINES	SEC. LINES	SEC. ELEM.
S (SAME)	2	10	-	10	-	-	10	2
C (CHANGED)	0	0	0	0	0	0	0	0
I (INSERTED)	2	10	10	-	-	10	-	-
D (DELETED)	-	-	-	-	0	0	0	0
TOTAL	4	20	10	10	0	10	10	2

```
//show-statistics number=*c1 ----- (9)
```

```
AREA C1
```

STATISTIC	PRIM. ELEM.	PRIM. LINES	INS. LINES	SAME LINES	DEL. LINES	INS+DEL LINES	SEC. LINES	SEC. ELEM.
S (SAME)	5	37	-	37	-	-	37	5
C (CHANGED)	0	0	0	0	0	0	0	0
I (INSERTED)	2	10	10	-	-	10	-	-
D (DELETED)	-	-	-	-	0	0	0	0
TOTAL	7	47	10	37	0	10	37	5

```
//end ----- (10)
```

- (8) Die aktuelle Vergleichsstatistik, die wieder im Bereich C0 steht, wird ausgegeben.
- (9) Der Bereich C1 wird ausgegeben. Er enthält die Gesamtstatistik aller Vergleiche, die in diesem LMS-Lauf bisher durchgeführt wurden.
- (10) LMS wird beendet.

## 9.8 Verzweigen in ein Benutzerprogramm beim Auflisten eines Elementes

Das Benutzerprogramm listet nur die ersten 10 Eingabesätze eines Elementes auf. Besteht ein Element aus weniger als 10 Sätzen, füllt das Programm mit eigenen Sätzen auf 10 Sätze auf.

```

/start-lms _____ (1)
//modify-logging-parameters logging=*maximum _____ (2)
//open-library library=use.lib,mode=*update _____ (3)
//show-element (element=uselst,type=s) _____ (4)
INPUT LIBRARY= :10SQ:$USER.USE.LIB
INPUT ELEMENT= (S)USELST/@(0001)/2011-02-19
*      TITLE 'USEREXIT FUER DIE FUNKTION: LST'
*
*      1.) DURCH ANSCHLUSS DIESES UNTERPROGRAMMS WERDEN NUR NOCH
*      DIE ERSTEN 10 SAETZE JE ELEMENT AUFGELISTET.
*
*      2.) HAT DAS ELEMENT WENIGER ALS 10 SAETZE, SO WERDEN
*      WEITERE SAETZE EINGEFUEGT.
* INPUT FROM LMS: R1=A(PARAMETERLISTE)
*      R13=A(SAVEAREA), 18 WORTE
*      R14=RUECKSPRUNGADRESSE
*      R15=A(BENUTZERPROGRAMM)
*
* PARDSEC  DSECT
AUFTRAG  DS    A          A(AUFTRAG VOM LMS)
*      - 'BOE':START OF ELEMENT
*      - 'REC':RECORD ANGEBOTEN
*      - 'EOE':END OF ELEMENT
ANTWORT  DS    A          A(ANTWORT VOM BENUTZERPROGRAMM)
*      - 'CON':CONTINUE
*      - 'DEL':DELETE RECORD
*      - 'INS':INSERT NEW RECORD
SATZ     DS    A          A(SATZ, INKL. 4 BYTE HEADER)
*      -

```

- (1) LMS wird aufgerufen.
- (2) Alle Meldungen und Anweisungen werden protokolliert.
- (3) Die Bibliothek USE.LIB wird zugewiesen.
- (4) Das Benutzer-Quellprogramm USELST wird aufgelistet.

```

PARDSECL EQU *-PARDSEC      L'DSECT
USELST    CSECT PAGE
          STM 0,15,0(13)    SAVE REGISTERS
          LR  10,15         BASE
          USING PARDSEC,1   LMS-PARAMETERLISTE
          USING USELST,10
          L   6,AUFTRAG      A(AUFTRAG)
          L   7,ANTWORT     A(ANTWORT)
          L   8,SATZ        A(SATZ)
          CLC 0(3,6),REC    SATZ ANGEBOTEN ?
          BE  DOSATZ        JA ——?
          CLC 0(3,6),BOE    START OF ELEMENT
          BE  DOBOE        JA ——?
          CLC 0(3,6),EOE    END OF ELEMENT
          BE  DOEOE        JA ——?
          B   RETURN

*
DOBOE     EQU *
          ZAP ANZAHL,PO     ZAEHLER := 0
          B   RETURN

*
DOSATZ    EQU *
          CP  ANZAHL,P10    SCHON 10 SAETZE AUSGEGEBEN ?
          BNL DODEL        JA (REST UEBERGEHEN) ——?
          AP  ANZAHL,P1     ZAEHLER := ZAEHLER +1
          B   DOCON

*
DOEOE     EQU *
          CP  ANZAHL,P10    SCHON 10 SAETZE AUSGEGEBEN ?
          BNL DOCON        JA (KEIN EINFUEGEN) ——?
          AP  ANZAHL,P1     ZAEHLER := ZAEHLER +1
          B   DOINS

*
DOINS     EQU *
          MVC 0(3,7),INS    INSERT RECORD
          LA  9,INSSATZ
          ST  9,SATZ        A(RECORD TO BE INSERTED)
          B   RETURN

*
DODEL     EQU *
          MVC 0(3,7),DEL    DELETE RECORD
          B   RETURN

*
DOCON     EQU *
          MVC 0(3,7),CON    CONTINUE

*

```

```

RETURN    EQU    *
          LM     0,15,0(13)      RESTORE REGISTERS
          BR     14
          TITLE  'KONSTANTEN UND VARIABLE'
BOE       DC     'BOE'          START OF ELEMENT
REC       DC     'REC'          SATZ ANGEBOTEN
EOE       DC     'EOE'          END OF ELEMENT
CON       DC     'CON'          CONTINUE
DEL       DC     'DEL'          DELETE RECORD
INS       DC     'INS'          INSERT NEW RECORD
ANZAHL   DC     PL2'0'
PO        DC     PL2'0'
P1        DC     PL2'1'
P10       DC     PL2'10'
INSSATZ  DC     Y(INSSATZE-INSSATZ)
          DC     XL2'4040'
          DC     '***** INSERT BY USER-PROGRAM *****'
INSSATZE EQU    *
          LTORG
          END

NUMBER OF PROCESSED RECORDS IS      89
//activate-user-exit function=*show-elm,entry=uselst,lib=use.lib _____ (5)
//show-element (element=einaus,type=s) _____ (6)
INPUT LIBRARY= :IOSQ:$USER.USE.LIB
INPUT ELEMENT= (S)EINAUS@(0001)/2011-02-19
USER EXIT USELST IN USE.LIB IS ACTIVE
#286 >          TITLE 'ERFASSEN VON DATEN'
#287 >          PRINT NOGEN
#288 > ERFAS     START
#289 >          BALR 5,0
#290 >          USING *,5
#291 >          OPEN DATEI,OUTPUT
#292 > LESEN     RDATA SATZ,ENDPGM
#293 >          CLC TEXT(4),=C'/EOF'
#294 >          BE  ENDPGM
#295 >          MVC  ATEXT,TEXT
NUMBER OF PROCESSED RECORDS IS      10

```

- (5) LMS verzweigt vor dem Auflisten eines Eingabesatzes in das Benutzerprogramm USELST, das in der Bibliothek USE.LIB steht.
- (6) Die ersten 10 Sätze des Elementes EINAUS der zugewiesenen Bibliothek USE.LIB werden aufgelistet.

```

//show-user-exits ----- (7)
FUNCTION  ENTRY    LIBRARY                INT
SHOW      USELST   USE.LIB                    V1
//show-element (element=persdat,type=s) ----- (8)
INPUT  LIBRARY= :10SQ:$USER.USE.LIB
INPUT  ELEMENT= (S)PERSDAT@(0001)/2011-02-19
USER EXIT USELST IN USE.LIB IS ACTIVE
#1113 > BACH          SEBASTIAN      MUENCHEN      AUF DER HOEHE 7      AB 3
#1114 > BERGMANN     NORBERT       MUENCHEN      TORWEG 10            AB 5
#1115 > FINK         SUSANNE       NUERNBERG     RINGSTR. 23         AB 1
#1116 > MEYER        FRANZ         NUERNBERG     WASSERMUNGENWEG    AB 1
#1117 > GRUNDLER     WOLFGANG     BASEL         SONNENSTR. 11       AB 2
#1118 > KNOLL        MONIKA        FRANKFURT     BAUMALLEE 12        AB 3
#1119 > LIEDL        ERIKA         MUENCHEN      IN DER BREITE 1     AB 5
#1120 > WAGNER       JOHANN        AUGSBURG      AM SEE 45            AB 4
***** INSERT BY USER-PROGRAM *****
***** INSERT BY USER-PROGRAM *****
NUMBER OF PROCESSED RECORDS IS      10
//end ----- (9)

```

- (7) Die aktiven Benutzerausgänge werden angezeigt.
- (8) Das Element PERSDAT wird aufgelistet. Da es kürzer ist als 10 Sätze, werden vom Benutzerprogramm Sätze angefügt.
- (9) LMS wird beendet.



## 9.9 Schutzattribute vergeben und anzeigen

Für eine Bibliothek und bestimmten Elementen in dieser Bibliothek sollen spezifische Schutzrechte vergeben werden.

```

/start-lms
//modify-logging-parameters logging=*maximum _____ (1)
//open-library library=bsp9.bib,mode=*update _____ (2)
LIBRARY IS CLEARED AND PREPARED
//modify-library-attributes administration=(user=*owner), -
//      init-elem-protection=(read=(user=*owner), -
//                               write=(user=*owner,password='P'),-
//                               exec=(user=*owner)) _____ (3)
//show-library-attributes _____ (4)
INPUT LIBRARY= :10SQ:$USER.BSP9.BIB
READ-PASS = *NONE      READ-USER = *OWNER  -   -
WR-PASS   = *YES       WR-USER   = *OWNER  -   -
EXEC-PASS = *NONE      EXEC-USER = *OWNER  -   -
ADMIN-PASS = *NONE     ADMIN-USER = *OWNER  -   -
FILE-SIZE = 12         FREE-SIZE = 4        FORMAT = NK2  UPAM-PROT = N
ACCESS-DATE= *NONE     WR-CONTROL = *NONE    STORAGE=*NONE
//modify-type-attributes type=d, -
//      init-elem-protection=(read=(user>(*owner,*group)), -
//                               write=(user=*owner)) _____ (5)

```

- (1) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (2) Die Bibliothek BSP9.BIB wird zum Lesen und Schreiben geöffnet.
- (3) Es wird das Administrationsrecht für den Bibliothekseigentümer 'USER' vergeben. Nur dieser darf Elemente anlegen, löschen oder umbenennen. Es wird ein initialer Elementschutz eingestellt. Für alle Elementtypen, für die nichts spezielleres eingestellt ist, werden Elemente mit diesem Schutz angelegt.
- (4) Ausgabe der Bibliotheksattribute.
- (5) Für Typ D wird ein spezieller initialer Elementschutz eingestellt: das Leserecht erhält der Eigentümer und die Gruppe, das Schreibrecht der Eigentümer der Bibliothek.

```

//add-element to-element=(element=test,type=s) ----- (6)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP9.BIB
OUTPUT ELEMENT= (S)TEST/@(0001)/2013-03-01
          ADD (S)TEST/@(0001)/2013-03-01

//add-element to-element=(element=test,type=d) ----- (7)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP9.BIB
OUTPUT ELEMENT= (D)TEST/@(0001)/2013-03-01
          ADD (D)TEST/@(0001)/2013-03-01

//copy-element element=(element=test,type=s), -
//          to-element=(element=test2,type=s) ----- (8)
INPUT  LIBRARY= :10SQ:$USER.BSP9.BIB
OUTPUT LIBRARY= :10SQ:$USER.BSP9.BIB
          COPY (S)TEST/@(0001)/2013-03-01 AS (S)TEST2/@(0001)/2013-03-01

//modify-element-protection element=(element=test2,type=s), -
//          new-protection=(read=(user=*all)) ----- (9)
OUTPUT LIBRARY= :10SQ:$USER.BSP9.BIB
          MODIFY (S)TEST2/@(0001)/2013-03-01

```

- (6) Es wird ein Element mit dem Namen TEST vom Typ S angelegt. Die Eingabe erfolgt über \*SYSDTA. Das Element erhält automatisch den für die Bibliothek geltenden Schutz (siehe (3)).
- (7) Es wird ein weiteres Element unter dem gleichen Namen, aber mit Typ D angelegt. Die Eingabe für dieses Element erfolgt ebenfalls über \*SYSDTA. Das Element erhält den Schutz, der für alle Elemente des Typs D vereinbart ist (siehe (5)).
- (8) Das Element TEST vom Typ S wird kopiert. Das neue Element soll TEST2 heißen. Der Typ wird beibehalten.
- (9) Das Element TEST2 erhält einen neuen Schutz; das Leserecht wird auf alle Anwender ausgeweitet.

```

//show-element-attributes element=(element=*(version=*)), -
//                               information=*maximum _____ (10)
INPUT  LIBRARY= :10SQ:$USER.BSP9.BIB
TYPE   = D
NAME   = TEST
VERSION = @                               VARIANT = 0001
USER-DATE = 2013-03-01  CRE-DATE = 2013-03-01  MOD-DATE = 2013-03-01
USER-TIME = 10:33:06   CRE-TIME = 10:33:06   MOD-TIME = 10:33:06
STORAGE = *FULL
STATE   = *FREE
ELEM-SIZE = 1
READ-PASS = *NONE      READ-USER = *OWNER *GROUP -
WR-PASS   = *NONE      WR-USER   = *OWNER - -

      1 (D)-ELEMENT(S) IN THIS TABLE OF CONTENTS
TYPE   = S
NAME   = TEST
VERSION = @                               VARIANT = 0001
USER-DATE = 2013-03-01  CRE-DATE = 2013-03-01  MOD-DATE = 2013-03-01
USER-TIME = 10:33:06   CRE-TIME = 10:33:06   MOD-TIME = 10:33:06
STORAGE = *FULL
STATE   = *FREE
ELEM-SIZE = 1
READ-PASS = *NONE      READ-USER = *OWNER - -
WR-PASS   = *YES       WR-USER   = *OWNER - -
EXEC-PASS = *NONE      EXEC-USER = *OWNER - -

TYPE   = S
NAME   = TEST2
VERSION = @                               VARIANT = 0001
USER-DATE = 2013-03-01  CRE-DATE = 2013-03-01  MOD-DATE = 2013-03-01
USER-TIME = 10:33:06   CRE-TIME = 10:33:06   MOD-TIME = 10:33:06
STORAGE = *FULL
STATE   = *FREE
ELEM-SIZE = 1
READ-PASS = *NONE      READ-USER = *OWNER *GROUP *OTHERS
WR-PASS   = *YES       WR-USER   = *OWNER - -
EXEC-PASS = *NONE      EXEC-USER = *OWNER - -

      2 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
_____
      3 ELEMENT(S) IN THIS TABLE OF CONTENTS
//end _____ (11)

```

(10) Alle Elemente werden mit ihren Schutzattributen angezeigt.

(11) Der LMS-Lauf wird beendet.

## 9.10 Automatische Versionserhöhung bei Konvention NONE

Das Element test wird immer unter gleichem Namen, aber mit unterschiedlichem Aufnahmeverfahren, in die Bibliothek aufgenommen.

```

/start-lms
//modify-logging-parameters logging=*maximum _____ (1)
//open-library library=bsp10.bib,mode=*update _____ (2)
LIBRARY IS CLEARED AND PREPARED
//show-type-attributes type=s _____ (3)
INPUT LIBRARY= :10SQ:$USER.BSP10.BIB
TYPE = S
SUPER-TYPE = *NONE
BASE-TYPE = S
CONVENTION = *NONE
INIT-ELEM-P= *NONE
ADMINISTRAT= *NONE
STORAGE = *NONE WR-CONTROL = *NONE
//add-element to-element=(element=test(version=*increment),type=s) _____ (4)
input1
*END
INPUT SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP10.BIB
OUTPUT ELEMENT= (S)TEST/001(0001)/2013-03-01
ADD (S)TEST/001(0001)/2013-03-01

```

- (1) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (2) Die Bibliothek BSP10.BIB wird zum Lesen und Schreiben geöffnet; es ist kein Element vom Typ S vorhanden.
- (3) Anzeigen der Typattribute für den Elementtyp S zur Kontrolle. Der Elementtyp S hat die Konvention NONE.
- (4) Erzeugen des ersten Elements über \*SYSDTA. Das Element wird unter Typ S mit Namen test über die automatische Versionserhöhung erzeugt.

```

//add-element to-element=(element=test(version=*increment),type=s) ----- (5)
input2
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/002(0001)/2013-03-01
        ADD (S)TEST/002(0001)/2013-03-01

//add-element to-element=(element=test(version=a001),type=s) ----- (6)
input3
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/A001(0001)/2013-03-01
        ADD (S)TEST/A001(0001)/2013-03-01

//add-element to-element=(element=test(version=*increment,base=a*),type=s) ----- (7)
input4
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/A002(0001)/2013-03-01
        ADD (S)TEST/A002(0001)/2013-03-01

//add-element to-elm=(elem=test(version=*highest-existing),type=s), -
//      write-mode=*any ----- (8)
input5
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/002(0002)/2013-03-01
        ADD (S)TEST/002(0002)/2013-03-01 , OUTPUT REPLACED

//show-element-attributes ----- (9)
INPUT  LIBRARY= :10SQ:$USER.BSP10.BIB
TYP NAME VER (VAR#) DATE      NAME VER (VAR#) DATE
(S) TEST A001 (0001) 2013-03-01  TEST A002 (0001) 2013-03-01
(S) TEST 001 (0001) 2013-03-01  TEST 002 (0002) 2013-03-01
    4 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS

//end ----- (10)

```

- (5) Weiterführen der Sequenz mit \*INCREMENT, d.h. es wird ein zweites Element mit gleichem Namen und Typ, aber um eins erhöhter Version erzeugt.
- (6) Explizites Einrichten der Version A001.
- (7) Erzeugen eines weiteren Elementes über die automatische Versionserhöhung; das Element setzt auf Version mit Präfix A\* auf.
- (8) Überschreiben der höchsten Version mit Typ S und Namen test.
- (9) Inhaltsverzeichnis der Bibliothek BSP10.BIB.
- (10) Beenden des LMS-Laufs.

## 9.11 Automatische Versionserhöhung bei Konvention STD-SEQUENCE

Es sollen Elemente in einem fest vorgegebenen Versionsformat in eine Bibliothek aufgenommen werden, wobei die Versionserhöhung automatisch erfolgen soll.

```

/start-lms
//modify-logging-parameters logging=*maximum _____ (1)
//open-library library=bsp11.bib,mode=*update _____ (2)
LIBRARY IS CLEARED AND PREPARED
//modify-type-attributes type=s,convention=*std-sequence(example=v001) _____ (3)
//show-type-attributes type=s _____ (4)
INPUT LIBRARY= :10SQ:$USER.BSP11.BIB
TYPE = S
SUPER-TYPE = *NONE
BASE-TYPE = S
CONVENTION = *STD-SEQUENCE
EXAMPLE = V001
INIT-ELEM-P= *NONE
ADMINISTRAT= *NONE
STORAGE = *NONE WR-CONTROL = *NONE
//add to-element=library-element(element=test(version=*increment),type=s) _____ (5)
input1
*END
INPUT SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP11.BIB
OUTPUT ELEMENT= (S)TEST/V001(0001)/2013-03-01
ADD (S)TEST/V001(0001)/2013-03-01

```

- (1) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (2) Die Bibliothek BSP11.BIB wird zum Lesen und Schreiben geöffnet; es ist kein Element vom Typ S vorhanden.
- (3) Einrichten der Konvention STD-SEQUENCE für Typ S mit dem in EXAMPLE vorgegebenen Beispiel für das Versionsformat.
- (4) Anzeigen der Typattribute für den Elementtyp S zur Kontrolle.
- (5) Erzeugen des ersten Elements über \*SYSDTA. Das Element wird unter Typ S mit Namen test über die automatische Versionserhöhung erzeugt.

```

//add-element to-element=(element=test(version=*increment),type=s) ----- (6)
input1
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/V002(0001)/2013-03-01
        ADD (S)TEST/V002(0001)/2013-03-01

//add to-element=library-element(element=test(version=w001),type=s) ----- (7)
input1
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/W001(0001)/2013-03-01
        ADD (S)TEST/W001(0001)/2013-03-01

//add to-element=library-element(element=test(version=999),type=s) ----- (8)
input1
*END
OUTPUT ELEMENT= (S)TEST/999/2013-03-01
% LMS0024 ERROR ON LIBRARY ':10SQ:$USER.BSP11.BIB', *** PLAM ERROR CODE '0476'.
% PLA0476 VERSION OR PREFIX NOT ACCORDING TO EFFECTIVE CONVENTION
        NO ADD (S)TEST/999/2013-03-01 , ERROR OCCURRED
SKIPPED:input1

//add to-element=(element=test(version=*highest-existing),type=s), -
//  write-mode=*any ----- (9)
input1
*END
INPUT  SYSDTA
OUTPUT ELEMENT= (S)TEST/W001(0002)/2013-03-01
        ADD (S)TEST/W001(0002)/2013-03-01 , OUTPUT REPLACED

```

- (6) Weiterführen der Sequenz mit \*INCREMENT, d.h. es wird ein zweites Element mit gleichem Namen und Typ, aber um eins erhöhter Version erzeugt.
- (7) Weiterführen der Sequenz mit Versionswechsel, d.h. es wird ein drittes Element mit gleichem Namen und Typ, aber durch die explizite Angabe der Version mit der Version W001 erzeugt.
- (8) Es wird versucht, ein weiteres Element aufzunehmen, das nicht ein der Konvention STD-SEQUENCE entsprechendes Versionsformat besitzt. Diese Anweisung wird von LMS zurückgewiesen.
- (9) Überschreiben der höchsten Version mit Typ S und Namen test.

```
//show-element-attributes ----- (10)
INPUT LIBRARY= :10SQ:$USER.BSP11.BIB
TYP NAME VER (VAR#) DATE          NAME VER (VAR#) DATE
(S) TEST V001 (0001) 2013-03-01    TEST V002 (0001) 2013-03-01
(S) TEST W001 (0002) 2013-03-01
      3 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//end ----- (11)
```

(10) Übersicht über die vorhandenen Elemente.

(11) Beenden des LMS-Laufs.



## 9.12 Automatische Versionserhöhung bei Konvention STD-TREE

```

/start-lms
//modify-logging-parameters logging=*maximum _____ (1)
//open-library library=bsp12.bib,mode=*update _____ (2)
LIBRARY IS CLEARED AND PREPARED
//modify-type-attributes type=s,convention=*std-tree _____ (3)
//show-type-attributes type=s _____ (4)
INPUT LIBRARY= :10SQ:$USER.BSP12.BIB
TYPE = S
SUPER-TYPE = *NONE
BASE-TYPE = S
CONVENTION = *STD-TREE
INIT-ELEM-P= *NONE
ADMINISTRAT= *NONE
STORAGE = *NONE WR-CONTROL = *NONE
//add to-element=library-element(element=test(version=*increment),type=s) _____ (5)
input1
*END
INPUT SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/001.001(0001)/2013-03-01
ADD (S)TEST/001.001(0001)/2013-03-01

```

- (1) Zusätzlich zu Fehlermeldungen werden auch Erfolgsmeldungen protokolliert.
- (2) Die Bibliothek BSP12.BIB wird zum Lesen und Schreiben geöffnet; es ist kein Element vom Typ S vorhanden.
- (3) Einrichten der Konvention STD-TREE für den Elementtyp S.
- (4) Anzeigen der Typattribute für den Elementtyp S zur Kontrolle.
- (5) Erzeugen des ersten Elements über \*SYSDTA. Das Element wird unter Typ S mit Namen test über die automatische Versionserhöhung erzeugt.

```

//add to-element=library-element(element=test(version=*increment),type=s) ----- (6)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/001.002(0001)/2013-03-01
          ADD (S)TEST/001.002(0001)/2013-03-01

//add to-element=(element=test(version=*increment,base=1.1),type=s) ----- (7)
*END

INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/001.001.001.001(0001)/2013-03-01
% LMS0095 INPUT DATA RECORDS MISSING
          ADD (S)TEST/001.001.001.001(0001)/2013-03-01

//add-element to-element=library-element(element=test(version=2.1),type=s) ----- (8)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/002.001(0001)/2013-03-01
          ADD (S)TEST/002.001(0001)/2013-03-01

//add to-element=(element=test(version=*increment,base=1.1.1.*),type=s) ----- (9)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/001.001.001.002(0001)/2013-03-01
          ADD (S)TEST/001.001.001.002(0001)/2013-03-01

//add to-element=(element=test(version=*highest-existing),type=s), -
// write-mode=*any ----- (10)
input1
*END
INPUT  SYSDTA
OUTPUT LIBRARY= :10SQ:$USER.BSP12.BIB
OUTPUT ELEMENT= (S)TEST/002.001(0002)/2013-03-01
          ADD (S)TEST/002.001(0002)/2013-03-01 , OUTPUT REPLACED

```

- (6) Weiterführen der Sequenz mit \*INCREMENT, d.h. es wird ein zweites Element mit gleichem Namen und Typ, aber um eins erhöhter Version erzeugt.
- (7) Eröffnen eines Nebenzweigs an der Version 1.1
- (8) Erzeugen eines Elementes mit gleichem Typ und Namen, aber mit einer explizit angegebenen Version.
- (9) Weiterführung des Nebenzweigs mit Präfix 1.1.1.
- (10) Überschreiben der höchsten Version mit Typ S und Namen TEST.

```
//show-element-attributes ----- (11)
INPUT LIBRARY= :10SQ:$USER.BSP12.BIB
TYP NAME VERSION      (VAR#) DATE
(S) TEST 001.001 . . . (0001) 2013-03-01
(S) TEST 001.001.001.001 (0001) 2013-03-01
(S) TEST 001.001.001.002 (0001) 2013-03-01
(S) TEST 001.002 . . . (0001) 2013-03-01
(S) TEST 002.001 . . . (0002) 2013-03-01
      5 (S)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//end ----- (12)
```

- (11) Ausgeben des Inhaltsverzeichnis der Bibliothek BSP12.BIB. Die Bibliothek enthält fünf Elemente mit gleichem Namen und Typ, aber unterschiedlicher Versions- und Variantenummer.
- (12) Beenden des LMS-Laufs.

## 9.13 make-Lauf

Das folgende Beispiel für die make-Funktionalität besteht aus mehrere Prozeduren.

Die Prozedur MAINPROCEDURE hat folgende Parameter:

TARGET:	Zielkomponente des make-Laufs
SUCC-PROC:	Folgeverarbeitung
SELECT:	Berücksichtigen der Zeitstempel
PROCEDURE:	erzeugte Prozedur
MAKEFILE:	Beschreibung des eigenen Programmsystems
MAKEDEFAULTS:	Beschreibung globaler Voreinstellungen

Die Parameter TARGET, SUCC-PROC, SELECT und PROCEDURE sind in der gleichen Weise anzugeben wie bei BEGIN-MAKE. Bei Aufruf mit den voreingestellten Parametern ist die Zielkomponente des make-Laufs das erste Ziel in MAKEDEFAULTS bzw. MAKEFILE; es wird eine Prozedur erzeugt, die anschließend mit INCLUDE-PROCEDURE aufgerufen wird.

Die Datei MAKEDEFAULTS enthält Defaulteinstellungen und Standardaktionen, also die Teile, die von mehreren Programmsystemen gemeinsam benutzt werden können.

Die Datei MAKEFILE enthält die eigentlichen Definitionen des Programmsystems, die Abhängigkeiten.

Das Programmsystem enthält eine Phasendatei, die aus zwei Objektmodulen gebunden wird. Die Objektmodule werden aus gleichnamigen Typ-S-Elementen unter Zuhilfenahme von zwei Typ-M-Elementen übersetzt. Die Pseudo-Ziele CLEAR und PRINT ermöglichen es, spezielle Aktionen für das Programmsystem zu veranlassen (Aufräumen und Drucken).

Prozedur MAINPROCEDURE:

```

/DECLARE-PARAMETER (TARGET(INI-VALUE='*FIRST-TARGET'),-
/
/          SUCC-PROC(INI-VALUE='*CREATE-PROC'),-
/          SELECT(INI-VALUE='*MODIFIED'),-
/          PROCEDURE(INI-VALUE='#P'),-
/          MAKEFILE(INI-VALUE='MAKEFILE'),-
/          MAKEDEFAULTS(INI-VALUE='MAKEDEFAULTS'))
/SET-VAR SUBO=SUBSTR('&(PROCEDURE)',1,2)  „DATEI ODER BIB.-ELEMENT“ _____ (1)
/BEGIN-BLOCK DATA-INSERTION=YES
/START-LMS

```

(1) Zulassen der /INCLUDE-PROCEDURE-Kommandos.

```

/IF (SUBO='*L')
// MODIFY-LMS-DEFAULTS TYPE=J
// DELETE-ELEMENT &(PROCEDURE) „BIB.-ELEMENT LOESCHEN“
// STEP
/ELSE
// EXEC-SYSTEM-CMD DELETE-FILE &(PROCEDURE) „DATEI LOESCHEN“
// STEP
/END-IF
//BEGIN-MAKE TARGET=&(TARGET), - - - - - (2)
//      SELECT=&(SELECT), -
//      SUCCESS-PROCESSING=&(SUCC-PROC), -
//      PROCEDURE=&(PROCEDURE)
//INCLUDE-PROCEDURE NAME=&MAKEDEFAULTS - - - - - (3)
//INCLUDE-PROCEDURE NAME=&MAKEFILE
//END-BLOCK
//END-MAKE - - - - - (4)
//END
/IF (SUBSTR('&(SUCC-PROC)',2,2) <> 'CR') „NICHT CREATE-PROC“
/  EXIT-PROC „FOLGEVERARBEITUNG BEREITS AUSGEFUEHRT“
/END-IF
/INC-PROC &(PROCEDURE) „AUSFUEHREN, EVTL. NICHT VORHANDEN“
/SET-JOB-STEP
/EXIT-PROCEDURE

```

- (2) Der make-Lauf wird gestartet.
- (3) Aufrufen von MAKEDEFAULTS und MAKEFILE.
- (4) Die make-Subanweisungen und damit der make-Lauf werden beendet.  
Die make-Folgeverarbeitung, wie in BEGIN-MAKE (hier der Default INCLUDE-PROCEDURE) wird angestoßen.

## Datei MAKEDEFAULTS:

```
/RESUME-PROGRAM
//MODIFY-MAKE-DEFAULTS LIBRARY=BSPLIB,-
//  CURRENT-TARGET-VAR=CURT, FROM-OBJECTS-VAR=ALLOBJ _____ (5)
//SET-STD-ACTION - _____ (6)
//  TARGET-TYPE=R, FROM-TYPE=S,-
//  ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,COMPILE),(&&(CURT.ELEM))'
/HOLD-PROGRAM _____ (7)
/EXIT-PROC
```

- (5) Default-Bibliothek und make-Variablen werden eingestellt. Die Werte gelten für die nachfolgenden Anweisungen.
- (6) Es wird eine Standard-Aktion (Übersetzen) für den Übergang vom Elementtyp S nach Elementtyp R vereinbart.
- (7) Vorbereiten des Rücksprungs aus der Prozedur.

## Datei MAKEFILE:

```

/RESUME-PROGRAM
//SET-DEPENDENCY - _____ (8)
// TARGET-OBJECT=PROG, -
// FROM-OBJECT=(*LIB(,TEIL1,R),*LIB(,TEIL2,R)), -
// ACTION=' /CALL-PROCEDURE LINK,(OBJVAR=ALLOBJ) '
//SET-DEPENDENCY - _____ (9)
// TARGET-OBJECT=*LIB(*,R),-
// FROM-OBJECT=(*LIB(*,S), *LIB(,GLOBALDATA,M), *LIB(,HEADERS,M)) _____ (10)
//SET-DEPENDENCY - _____ (11)
// TARGET-OBJECT=CLEAR,-
// ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,DELETE.TEMPS) '
//SET-DEPENDENCY - _____ (12)
// TARGET-OBJECT=PRINT,-
// ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,PRINT.LISTFILES) '
//SET-PREPROCESSING - _____ (13)
// ACTION=( ' /CALL-PROCEDURE *LIB(BSPLIB,INIT) ' , -
//          ' /DECL-VAR ALLOBJ(TYPE=STRUC),MULT-ELEM=LIST,SCOPE=TASK' )
//SET-POSTPROCESSING - _____ (14)
// ACTION=' /CALL-PROCEDURE *LIB(BSPLIB,STOP) '
/HOLD-PROGRAM
/EXIT-PROC

```

- (8) Darstellung der Abhängigkeit „Phase wird aus Objektmodulen gebunden“. Die Phasendatei PROG ist das erste Ziel des make-Laufs (\*FIRST-TARGET).
- (9) Darstellung der Abhängigkeit „Objektmodul wird aus gleichnamigem Source erzeugt, weitere Komponenten (Typ M) werden benötigt.“ Die Standard-Aktion unter (3) wird verwendet.
- (10) „Gleichnamig“ wird durch die Verwendung von make-Auswahl und -Konstruktion formuliert. Unter TARGET-OBJECT steht die Auswahl, unter FROM-OBJECT die Konstruktion. Die LMS-Einschränkung, dass die Konstruktion mindestens ein Wildcard-Symbol enthalten muss, gilt nicht für make-Konstruktionsangaben. \*LIB(,HEADERS,M) enthält kein Wildcard-Symbol.
- (11) Die Zieldatei CLEAR ist von keinem FROM-OBJECT abhängig, ist also nie aktuell. Wird CLEAR als Ziel des make-Laufs angegeben, werden Aufräumaktionen für das Programmsystem angestoßen. Die Aktionen für CLEAR sollen nicht jedesmal durchgeführt werden, wenn das Programmsystem aktualisiert wird.
- (12) Wie (11). Als Aktion für PRINT ist das Drucken von Ergebnissen vorgesehen.

- (13) Wenn eine Prozedur erzeugt wird, soll die Aktion `/CALL INIT` an deren Anfang gestellt werden. Ebenfalls soll die Variable `ALLOBJ` taskglobal deklariert werden, da sie später in der Prozedur `LINK` importiert werden soll. Die Deklaration wird im Ablauf der erzeugten Prozedur nicht überschrieben.
- (14) Wenn eine Prozedur erzeugt wird, soll die Aktion `/CALL STOP` an deren Ende gestellt werden.

Die Datei `MAKEFILE` soll noch in einer, dem UNIX-make ähnlichen Schreibweise vorgestellt werden:

Datei `MAKEFILE`:

```
/RESUME-PROGRAM
//SET-DEPENDENCY PROG, (*LIB(,TEIL1,R),*LIB(,TEIL2,R)), -
//  '/CALL-PROCEDURE *LIB(BSPLIB,LINK),(OBJVAR=ALLOBJ)'
//SET-DEPENDENCY *LIB(*,R),(*LIB(*,S),*LIB(,GLOBALDATA,M),*LIB(,HEADERS,M))
//SET-DEPENDENCY CLEAR, *NONE,-
//  '/CALL-PROCEDURE *LIB(BSPLIB,DELETE.TEMPS)''
//SET-DEPENDENCY PRINT, *NONE,-
//  '/CALL-PROCEDURE *LIB(BSPLIB,PRINT.LISTFILES)''
//SET-PREPROCESSING ('/CALL-PROCEDURE *LIB(BSPLIB,INIT)',-
//  '/DECL-VAR ALLOBJ(TYPE=STRUC),MULT-ELEM=LIST,SCOPE=TASK')
//SET-POSTPROCESSING '/CALL-PROCEDURE *LIB(BSPLIB,STOP)''
/HOLD-PROGRAM
/EXIT-PROC
```



Wenn das Programmsystem vollständig erzeugt werden muss, sieht die resultierende Prozedur im Prinzip so aus:

```

/DECL-VAR SYSLMSMAKE(TYPE=STRUC(DEF=*DYN)) _____ (1)
/CALL-PROCEDURE *LIB(BSPLIB,INIT)
/DECL-VAR ALLOBJ(TYPE=STRUC),MULT-ELEM=*LIST,SCOPE=*TASK
/IF-BLOCK-ERROR _____ (2)
/EXIT-PROCEDURE
/END-IF
/DECLARE-VARIABLE CURT(TYPE=*STRUC) _____ (3)
/IF-BLOCK-ERROR
/END-IF
...
/SET-VARIABLE CURT.ELEM = 'TEIL1' _____ (4)
...
/CALL-PROCEDURE *LIB(BSPLIB,COMPILE),(&(CURT.ELEM))
...
/SET-VARIABLE CURT.ELEM = 'TEIL2'
...
/CALL-PROCEDURE *LIB(BSPLIB,COMPILE),(&(CURT.ELEM))
...
/DECLARE-VARIABLE ALLOBJ(TYPE=*STRUC),MULT-ELEM=*LIST _____ (5)
/IF-BLOCK-ERROR
/END-IF
/FREE-VARIABLE ALLOBJ
...
/SET-VARIABLE SYSLMSMAKE.ELEM = 'TEIL1'
...
/SET-VARIABLE ALLOBJ = SYSLMSMAKE,MODE=APPEND _____ (6)
/SET-VARIABLE SYSLMSMAKE.ELEM = 'TEIL2'
...
/SET-VARIABLE ALLOBJ = SYSLMSMAKE,MODE=APPEND
/CALL-PROCEDURE *LIB(BSPLIB,LINK),(OBJVAR=ALLOBJ)
...
/CALL-PROCEDURE *LIB(BSPLIB,STOP)

```

- (1) Deklaration der make-internen Hilfsvariablen zum Listenaufbau.
- (2) Diese Anweisungsfolge wird nach jeder Aktion (außer POSTPROCESSING) eingefügt, um einen sauberen Fehlerrückzug zu gewährleisten.
- (3) Deklarieren von CURT (Fehler werden toleriert, da CURT z.B. global deklariert sein kann).

- (4) Die anderen Komponenten der S-Variablen sowie der S-Listenvariable ALLOBJ werden ebenfalls gesetzt.
- (5) Deklarieren einer leeren S-Listenvariablen für ALLOBJ. Die Behandlung von ALLOBJ findet ebenfalls vor den anderen Prozeduraufrufen statt. Das Ergebnis wird dort jedoch nicht verwendet.
- (6) Anhängen der FROM-OBJECTS an die S-Listenvariable ALLOBJ.

Die in MAKEFILE gerufenen Prozeduren haben folgende Schnittstellen:

1. INIT und STOP haben keine Parameter.
2. COMPILE hat nur einen Elementnamen als Parameter, Bibliothek und andere Informationen werden der Einfachheit halber nicht eingegeben.
3. LINK hat den Namen einer S-Listenvariablen als Eingabe, die alle Objekte enthält, die zusammengebunden werden sollen. Die Listenvariable wird wie im folgenden Muster abgearbeitet:

```

/DECL-PAR NAME=INPUTVAR(TYPE=STRING)
/DECL-VAR I(TYPE=STRUC(DEF=*DYN))
/IMPORT-VAR &(INPUTVAR) _____ (1)
...
/FOR I=*LIST(&INPUTVAR) _____ (2)
/   DO-SOMETHING ... &(I) _____ (3)
/END-FOR

```

- (1) Importieren der angegebenen Listenvariablen (sie muss taskglobal sein).
- (2) Aufzählen der Listenelemente in der Variablen I.
- (3) Verwenden von I entweder direkt in SDF-Eingaben, oder in /SEND-DATA oder /SEND-STATEMENT.

## 9.14 Nutzung der Ausgabe in S-Variable

Diese Prozedur führt einen Abgleich zwischen zwei Bibliotheken durch:

Alle Elemente, die in der neueren Bibliothek (Prozedurparameter NEW-LIB) enthalten sind, werden in der älteren Bibliothek (Prozedurparameter OLD-LIB) gelöscht.

```

/BEGIN-PARAMETER-DECLARATION
/DECLARE-PARAMETER NAME=OLD-LIB(INITIAL-VALUE=*PROMPT)
/DECLARE-PARAMETER NAME=NEW-LIB(INITIAL-VALUE=*PROMPT)
/END-PARAMETER-DECLARATION
/DECLARE-VARIABLE NAME=NEW-ELEMENTS(TYPE=*STRUC), -
/
      MULTIPLE-ELEMENTS=*LIST _____ (1)
/BEGIN-BLOCK DATA-INSERTION=*YES
/START-LMS
//SHOW-ELEM-ATTR -
//      ELEM=*LIB-ELEM(LIB=&NEW-LIB,ELEM=*ALL,TYPE=*ALL),-
//      INF=*MIN,STRUCTURE-OUTPUT=NEW-ELEMENTS _____ (2)
/DECLARE-VARIABLE NAME=LOOP(TYPE=*STRUC) _____ (3)
/FOR LOOP=*LIST(NEW-ELEMENTS)
//DELETE-ELEM *LIB-ELEM(LIB=&OLD-LIB, -
//      ELEM=&(LOOP.ELEM)(VERS=&(LOOP.VERSION)), -
//      TYPE=&(LOOP.TYPE)) _____ (4)
//STEP "falls Element in &OLD-LIB nicht existiert"
/END-FOR
//END
/END-BLOCK

```

- (1) Deklarieren der Variablen für die Ausgabe von //SHOW-ELEMENT-ATTRIBUTES.
- (2) Ausgabe von //SHOW-ELEMENT-ATTRIBUTES in die Variable.
- (3) Deklarieren der Schleifen-Variablen.
- (4) Löschen der Elemente in einer Schleife; für Element, Version und Typ werden die Werte der Schleifen-Variablen benutzt.

## 9.15 Bibliothekslisten

Das folgende Beispiel zeigt die Einbettung von Bibliothekslisten in eine Entwicklungsumgebung.

Wenn mehrere Entwickler auf einer gemeinsamen Datenbasis arbeiten, ist es üblich, neben dem zentralen Datenpool lokale Daten zu halten. Mit Hilfe des Ausleihmechanismus werden Daten hin und her übertragen.

Es ist zweckmäßig, beim Übersetzen einer lokalen Variante zentrale und lokale Daten zu einer Bibliotheksliste zusammenzufassen, da dann nicht für jedes Bibliothekselement der genaue Ablageort bekannt sein muss.

Die folgende Abbildung zeigt die zentrale und eine lokale Bibliothek, die zur Entwicklungsumgebung gehören. Quell-Elemente haben Elementtyp S, Include-Elemente M. Das Ergebnis ist PROGRAM mit Typ L. Mit Hilfe von Bibliothekslisten kann PROGRAM sowohl zentral als auch lokal mit dem gleichen Makefile aktualisiert werden. Die Bibliotheksliste, mit der PROGRAM aktualisiert wird, soll den Namen SYSPLAMALT-PROGRAM erhalten (Bibliothekslisten müssen mit "SYSPLAMALT-" beginnen). Bei lokaler Verwendung muss SYSPLAMALT-PROGRAM aus zwei Bibliotheken bestehen, bei zentraler nur aus einer.

- lokale Sicht: SYSPLAMALT-PROGRAM = '(LIB,\$ZENTRAL.LIB)'  
(erst lokal, dann unter \$ZENTRAL suchen)
- zentrale Sicht: SYSPLAMALT-PROGRAM = '(\$ZENTRAL.LIB)'  
(nur unter \$ZENTRAL suchen)

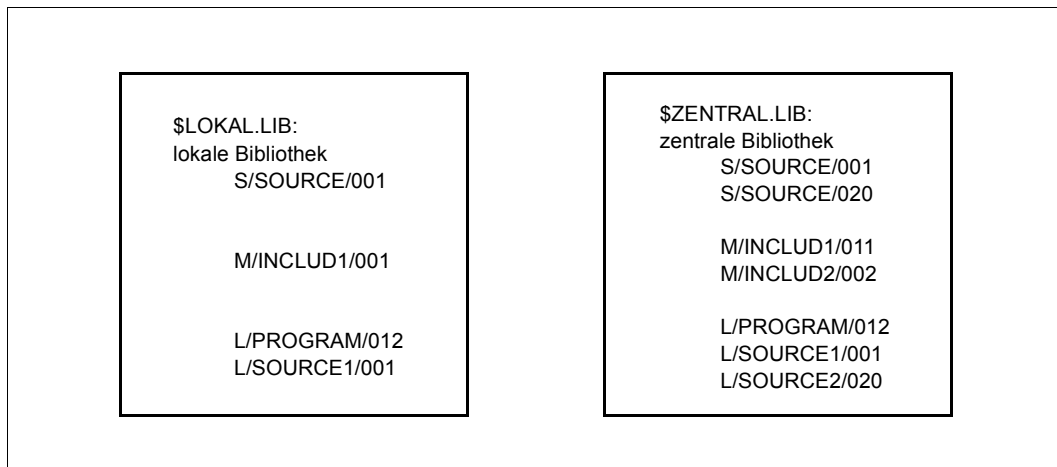


Bild 11: Zusammenfassung von Source-Bibliotheken

```

//begin-make
//modify-make-defaults library=sysplamalt-program,-
//      from-objects-var=from
//set-dependency *l(lib,program,l), "$LOKAL.LIB od. $ZENTRAL.LIB"-
//      (*l(,source1,l),-
//      *l(,source2,l)),'/call bind'
//set-dependency *l(,source1,l),-
//      (*l(,source1,s),-
//      *l(,includ1,m),-
//      *l(,includ2,m)),-
//      '/call compile lib=&&(from#1.lib),elem=source1'
//set-dependency *l(,source2,l),-
//      (*l(,source2,s),-
//      *l(,includ2,m)),-
//      '/call compile lib=&&(from#1.lib),elem=source2'
//end-make

```

In den Prozeduren BIND und COMPILE wird die Bibliotheksliste auf zwei verschiedene Weisen verwendet. BIND verwendet direkt die Bibliotheksliste auf der Eingabeseite:

```

...
//include-module element=source1,library=sysplamalt-program
//include-module element=source2,library=sysplamalt-program
...

```

Bei lokaler Verwendung hat dann die lokale Bibliothek Vorrang vor der zentralen.

Die Prozedur COMPILE hat einen LIB-Parameter, der über die make-S-Variable FROM mit der Trefferbibliothek versorgt wird; der Übersetzer arbeitet dann direkt mit diesem Wert. Die Include-Verarbeitung läuft mit der Bibliotheksliste SYSPLAMALT-PROGRAM. Dann haben lokale Includes Vorrang vor zentralen.



---

# 10 Anhang

## 10.1 Zusatzinformationen in LMS-Meldungen

Zu den eigentlichen Texten der LMS-Meldungen erscheinen in verschiedenen Kombinationen die folgenden Zusatztexte:

<b>Zusatzinformation</b>	<b>Bedeutung</b>
FUNCTION TERMINATED	Funktion wird abgebrochen
(LIBRARY INPUT)	Bibliothekseingabe
(LIBRARY OUTPUT)	Bibliotheksausgabe
(LISTING-MEMBER-OUTPUT)	Protokollausgabe in Element
STATEMENT(S) IS (ARE) SKIPPED	Die ganze Anweisung wird ignoriert
SYSLST (LISTING)	Protokollausgabe auf SYSLST
OUTPUT-LIBRARY LOCKED	Die Ausgabebibliothek ist gesperrt
OPENEROR ON LIBRARY / ELEMENT	Beim Öffnen der Bibliothek/des Elementes trat ein Fehler auf.
OUTPUT-LIBRARY MISSING	Die Ausgabebibliothek existiert nicht.
RECFORM=F	Das Bibliothekselement hat Sätze fester Länge.
KEYPOS>5	Die KEYPOS ist größer als 5.
KEYLEN>8	Die KEYLEN ist größer als 8. Dies gilt nur für den Aufruf des EDT.
KEYLEN>16	Die KEYLEN ist größer als 16. Dies gilt nur für den Aufruf des EDOR.
KEYS DO EXIST IN ELEMENT	Das zu erweiternde Bibliothekselement ist mit SOURCE-ATTRIBUTES=KEEP aufgenommen worden und enthält die ISAM- Schlüssel.
DIFFERENT FILETYPE/ VALUE PROPAGATION (MIN/MAX)	Der Dateityp bzw. die "VALUE PROPAGATION" der aufzunehmenden Datei stimmen nicht mit denen des zu erweiternden Bibliothekselement überein.

Zusatzinformation	Bedeutung
DIFFERENT RECORD FORMAT	Das Satzformat der aufzunehmenden Datei stimmt nicht mit dem Satzformat des zu erweiternden Bibliothekselementes überein.
DIFFERENT RECORD SIZE	Die Satzlänge der aufzunehmenden Datei stimmt nicht mit der Satzlänge des zu erweiternden Bibliothekselementes überein.
DIFFERENT KEYPOSITION	Die Position der Schlüssel der aufzunehmenden Datei stimmt nicht mit der Position der Schlüssel des zu erweiternden Bibliothekselementes überein.
DIFFERENT KEYLENGTH	Die Länge der Schlüssel der aufzunehmenden Datei stimmt nicht mit der Länge der Schlüssel des zu erweiternden Bibliothekselementes überein.
DIFFERENT LOGLENGTH	Die LOGLENGTH der aufzunehmenden Datei stimmt nicht mit der LOGLENGTH des zu erweiternden Bibliothekselementes überein.
DIFFERENT VALUE LENGTH	Die VALLEN der aufzunehmenden Datei stimmt nicht mit der VALLEN des zu erweiternden Bibliothekselementes überein.
OUTPUT-LIBRARY IS NOT A PLAM-LIBRARY	Die Ausgabebibliothek ist keine Programmbibliothek.
FIXED RECORD FORMAT ON INPUT-FILE	Das zu erweiternde Bibliothekselement enthält keine Dateiattribut- und die Eingabe-Datei hat Sätze fester Länge.
KEYPOSITION≠5 ON INPUT-FILE	Das zu erweiternde Bibliothekselement enthält keine Dateiattribut- und die Eingabe-Datei hat eine Schlüsselposition, die ungleich '5' ist.
KEYLENGTH>16 ON INPUT-FILE	Das zu erweiternde Bibliothekselement enthält keine Dateiattribut- und die Eingabe-Datei hat eine Schlüssellänge, die größer als '16' ist.
RECORD SIZE>2032 ON INPUT-FILE	Die aufzunehmende Datei hat eine Satzlänge von mehr als 2032 Byte.



## 10.2 Meldungen der Zugriffsroutine AMCB

Die Meldungen der internen LMS-Zugriffsroutine AMCB haben folgenden Aufbau:

{AMCB} : xxxx

Bedeutung

xxxx AMCB-Fehlerschlüssel

xxxx	Bedeutung der AMCB-Fehlerschlüssel
0000	Kein Fehler
0001	DMS-Fehler
0002	Ungültiger OP-CODE
0003	Dateiname im Kontrollblock fehlt
0004	Kein / geänderte FCB-ADR oder FCB-ADR für FOP mit DMS-OPEN zeigt auf aktiven FCB
0005	Falsche OP-CODE-Reihenfolge
0006	Ungültiger Bibliothekstyp
0007	Widersprüchliche LIB-Typen im Kontrollblock
0008	Bibliothek wurde repariert
0009	Bibliothek muss repariert werden
0010	Adresse nicht innerhalb der Elementgrenzen
0011	Beim Schreiben: Bibliotheksgrenze erreicht
0012	Widersprüchliche Angaben im Kontrollblock und FCB
0013	Zusatzinformation fehlt
0014	Ungültige Zusatzinformation
0015	Satz zu lang
0017	Letztes Element in Bibliothek wurde gelöscht
0019	Es handelt sich um keine LMS-Bibliothek
0022	Modul nicht komplett (z.B. kein END-Satz)
0023	Falscher Satztyp im Modul
0050	OVERWRITE-Fehler
0051	Speicher zu klein
0052	Element wird überschrieben
0053	Eingabedatei ist leer
0054	Leere Datei wird ersetzt

xxxx	Bedeutung der AMCB-Fehlerschlüssel
0062	Funktion nicht implementiert
0063	Keine Dateien vorhanden
0066	1. Satz kein ESD-Satz
0100	Illegale Programmdatei
0101	DVS-Fehler
0102	Unbekannter Dateityp
0107	Versorgung weder LINK-Name noch FILE-Name
0108	USER Fehler
0109	OPEN Fehler
0111	Kein freier Platz in FILE-Table
0112	FSTAT Fehler
0118	Kein leeres FILE bei CREATE
0119	Open auf leeres FILE
0120	Dateiname ungültig
0121	FILE-ID hat keinen Eintrag in FILE-Table
0122	Geforderter OPEN-Status ungleich IST-STATUS
0123	Kein Platz mehr für FCB
0124	Kein Platz mehr für Zugriffsvermerk in FT
0125	2. Zugriff auf Ausgabebibliothek
0126	Kein Platz mehr für LINK-Table-Eintrag
0127	LINK-Table-Eintrag fehlt
0131	Bibliothek ist noch für CTL oder PRT geöffnet
0136	Zugriffsfehler z.B. gesperrte Datei
0150	Unbekannte Zugriffsart

## 10.3 Alte LMS-Unterprogramm-Schnittstelle

Aus Kompatibilitätsgründen wird noch die hier beschriebene alte Unterprogramm-Schnittstelle unterstützt.

Die neue Unterprogramm-Schnittstelle ist in [1] beschrieben.

Wird LMS als Unterprogramm mit eigenständiger Dialogführung aufgerufen, kehrt es nach der Verarbeitung von END zum aufrufenden Programm zurück. LMS bleibt nach der Verarbeitung von END geladen. Ansonsten ist der Ablauf wie nach dem Laden mit dem Kommando /START-EXECUTABLE-PROGRAM. Bei jedem Aufruf von LMS meldet LMS eine eigene STXIT-Routine an.

Beim Rücksprung in das Hauptprogramm wird die LMS-STXIT-Routine abgemeldet; das Hauptprogramm muss seine eigene STXIT-Routine wieder anmelden.

Beim Aufruf der Unterprogramm-Schnittstelle sind folgende Registerkonventionen zu beachten:

Register 1      muss Null sein.

Register 13    enthält die Adresse eines Sicherstellungsbereichs von 18 Worten, der vom aufrufenden Programm zur Verfügung gestellt werden muss. Dieser Bereich wird von LMS für die Speicherung der Register des rufenden Programms verwendet.

Register 14    enthält die Rücksprungadresse.

Register 15    enthält die Einsprungadresse LMSUPSDF.

Vor dem Rücksprung in das rufende Programm hinterlegt LMS im Register 15 folgende Rücksprungcodes:

X'00'          LMS wurde normal beendet

X'04'          LMS wurde abnormal beendet

*Beispiel*

Aus dem Programm UPROG wird LMS als Unterprogramm aufgerufen. In LMS wird ein Element aus einer Programmbibliothek in eine Datei ausgegeben. Nach Beenden des LMS-Laufs wird ins Benutzerprogramm zurückgesprungen.

```

/START-LMS
//OPEN UEB.BIB,U _____ (1)
//SHOW-ELEM *LIB(,LMSCALL,S) _____ (2)
INPUT LIBRARY= :N:$USER.UEB.BIB,DEV=DISK
INPUT ELEMENT= (S)LMSCALL/(0001)/1995-08-12
UPROG START
      BALR 3,0
      USING *,3
      MVC OUTPUT,ANMELD
AUFRUF WROUT OUT,TERM
      LA 14,RUECK _____ (3)
      LA 1,0 _____ (4)
      LA 13,SAVE _____ (5)
      L 15,=V(LMSUPSDF) _____ (6)
      BALR 14,15
RUECK MVC OUTPUT,ABMELD
      WROUT OUT,TERM
*
*
TERM TERM
*
*
SAVE DS 18F
*
ANMELD DC '***** *LMS WIRD JETZT AUFGERUFEN *****'
ABMELD DC '***** LMS WURDE BEENDET - WEITER IM PROGRAMM *****'
OUT DC Y(ENDE-OUT)
      DS CL2
      DC X'01'
OUTPUT DS CL50
ENDE EQU *
*
*
      END UPROG
NUMBER OF PROCESSED RECORDS IS 29
//END
/
.
.
.
/START-EXECUTABLE-PROGRAM FROM-FILE=(LIB=UEB.BIB,ELEM=LMSCALL)
% BLS0500 PROGRAM 'LMSCALL', VERSION '007' OF '95-08-12' LOADED

***** *LMS WIRD JETZT AUFGERUFEN *****

```

```
% LMS0310 LMS VERSION '03.4B00' STARTED _____ (7)
//OPEN UEB.BIB,U
//SHOW-E-ATTR
INPUT LIBRARY= :N:$USER.UEB.BIB,DEV=DISK
TYP NAME VER (VAR#) DATE
(C) LMSCALL 007 (0001) 1995-08-12
    1 (C)-ELEMENT(S) IN THIS TABLE OF CONTENTS
//EXTRACT-ELEM *LIB(,LMSCALL,S),LMSCALL1
INPUT LIBRARY= :N:$USER.UEB.BIB,DUF2EV=DISK
OUTPUT FILE
    SEL (S)LMSCALL/(0001)/1995-08-12 AS LMSCALL1
//END _____ (8)
***** LMS WURDE BEENDET - WEITER IM PROGRAMM *****
```

- (1) Die Programmbibliothek UEB.BIB wird als Ein- und Ausgabebibliothek zugewiesen.
- (2) Das Quellprogramm LMSCALL wird aufgelistet.
- (3) Die Rücksprungadresse wird nach Register 14 geladen.
- (4) Register 1 wird auf 0 gesetzt.
- (5) Die Adresse des Sicherstellungsbereiches wird nach Register 13 geladen.
- (6) Die Einsprungadresse LMSUPSDF wird nach Register 15 geladen.
- (7) Aus dem Benutzerprogramm heraus wird LMS aufgerufen.
- (8) Nach Beenden des LMS-Laufes wird ins Benutzerprogramm zurückgesprungen.

## 10.4 Migration alter Bibliotheksformate

LMSSDF bearbeitet ausschließlich das PLAM-Bibliotheksformat. Die folgenden alten Bibliotheksformate können wie folgt ins PLAM-Bibliotheksformat konvertiert werden. Scheitern diese Konvertierungen, so liegt ein unbekanntes Bibliotheksformat vor.

### 1. Konvertieren von LMR-, MLU- und COBLUR-Bibliotheken

```
/EXEC $LMSCONV      oder /EXEC $LMS
$LIB libold
$LIB libnew,NEW
$DUP* *
$END
```

**libold** ist die zu konvertierende Bibliothek im LMR-, MLU- oder COBLUR-Format.  
**libnew** ist eine neuerstellte Bibliothek im PLAM-Format.

### 2. Konvertieren von FMS-Bibliotheken

```
/EXEC $LMS
$LIB libnew,NEW
$ADDS FMS=libold(*)
$END
```

**libold** ist die zu konvertierende Bibliothek im FMS-Format.

**libnew** ist eine neuerstellte Bibliothek im PLAM-Format.

**ADDx FMS=libold(...)** übernimmt das Originaldatum für das Zielelement.

**ADDx FMS=libold(...)>...** vergibt das Tagesdatum für das Zielelement.

## 10.5 Produktbestandteile

logische ID	Dateiname	Funktion
SYSPRG.ISP	LMS	Starterphase LMS (ISP)
SYSPRG.SDF	LMSSDF	Starterphase LMSSDF
SYSLNK	SYSLNK.LMS.034	LMS-Modulbibliothek
SYSLIB	SYSLIB.LMS.034	Benutzerschnittstelle für LMS-UP
SYSTEMES	SYSTEMES.LMS.034	Meldungsdatei
SYSSSC	SYSSSC.LMS.034	Subsystem-Deklarationen
SYSSSC	SYSSSC.LMS.034.PRELOAD	Subsystem-Deklarationen zum Vorladen des privaten Anteils von LMS (siehe Freigabemitteilung)
SYSSII	SYSSII.LMS.034	Strukturinformation für IMON
SYSSDF	SYSSDF.LMS.034	Syntaxdatei SDF
SYSFGM.D	SYSFGM.LMS.034.D	Freigabemitteilung Deutsch
SYSFGM.E	SYSFGM.LMS.034.E	Freigabemitteilung Englisch
SYSREP	SYSREP.LMS.034	Korrekturdatei
SYSRMS	SYSRMS.LMS.034	Korrekturdepot für RMS
SYSNRF	SYSNRF.LMS.034	Noref-Datei
SYSACF	SYSACF.LMS.034	Aliaskatalog für LMSLIB

Tabelle 6: Produktbestandteile





---

# Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie in auch gedruckter Form bestellen.

- [1] **LMS** (BS2000/OSD)  
Unterprogrammchnittstelle  
Benutzerhandbuch
- [2] **ARCHIVE** (BS2000/OSD)  
Benutzerhandbuch
- [3] **SDF** (BS2000/OSD)  
Dialogschnittstelle SDF  
Benutzerhandbuch
- [4] **BS2000/OSD-BC**  
Kommandos  
Benutzerhandbuch
- [5] **BINDER** (BS2000/OSD)  
Benutzerhandbuch
- [6] **SECOS** (BS2000/OSD)  
**Security Control System - Zugangs- und Zugriffskontrolle**  
Benutzerhandbuch
- [7] **SECOS** (BS2000/OSD)  
**Security Control System - Beweissicherung**  
Benutzerhandbuch
- [8] **JV**  
(BS2000/OSD)  
Jobvariablen  
Benutzerhandbuch
- [9] **BS2000/OSD-BC**  
Einführung in das DVS  
Benutzerhandbuch

- [10] **EDT (BS2000/OSD)**  
**Anweisungen**  
Benutzerhandbuch
- [11] **EDT (BS2000/OSD)**  
**Unicode-Modus Anweisungen**  
Benutzerhandbuch
- [12] **SDF-P (BS2000/OSD)**  
**Programmieren in der Kommandosprache**  
Benutzerhandbuch
- [13] **AID (BS2000)**  
Advanced Interactive Debugger  
**Basishandbuch**  
Benutzerhandbuch
- [14] **openFT (BS2000)**  
**Enterprise File Transfer in der offenen Welt**  
Benutzerhandbuch
- [15] **XHCS (BS2000/OSD)**  
8-bit-Code-Verarbeitung im BS2000/OSD  
Benutzerhandbuch
- [16] **C-Bibliotheksfunktionen (BS2000/OSD)**  
**für POSIX-Anwendungen**  
Referenzhandbuch

---

# Stichwörter

\$LMSPAR 75

## A

Abbrechen LMS-Ausgabe 104  
abgeleitete Komponente 131  
Abhängigkeit 131  
Abhängigkeiten (make)  
    Auswertung 140  
    Struktur 140  
Abhängigkeiten vom Elementtyp 50  
Abkürzungsmöglichkeiten der LMS-  
    Anweisungen 162  
Ablageeinheit 37  
Ablauf des Editieraufrufs 246  
Ablaufbeispiele  
    für typische LMS-Anwendungen 467  
Ablaufsteuerung für make 139  
ACS (Alias Catalogue System) 119  
ACTIVATE-USER-EXIT (Anweisung) 166  
ADD-ELEMENT (Anweisung) 173  
ADD-RECORD (MODIFY-ELEMENT-  
    Subanweisung) 310  
ADD-REP-RECORD (MODIFY-ELEMENT-  
    Subanweisung) 296  
ADD-TEXT-MODIFICATION (MODIFY-ELE-  
    MENT-Subanweisung) 297  
AID (advanced interactive debugger) 307  
Aktionen (actions) 137  
Aktivieren Benutzerprogramm 166  
Aktualität (Definition) 131  
Aktualität Programmsystem 131  
Aliaskatalog 119  
Aliasname 119, 148  
alphanum-name (Datentyp) 149

alte LMS-Unterprogramm-Schnittstelle 523  
AMCB-Fehlerschlüssel 521  
AMODE-Attribut 29  
Ändern  
    Bibliotheksattribute 333  
    Elementschutz 262, 323  
    Protokollparameter 360  
    von Bindeladmodulen 285  
    von Bindemodulen 285, 484  
    von Phasen 285  
Änderungen  
    gegenüber Vorgänger-Version 16  
Änderungsdatum und -zeit 47  
Anweisungen, Übersicht 164  
anweisungs-Returncode Mechanismus 104  
Anweisungskurznamen 163  
Anzeigen  
    Benutzerausgänge 452  
    Typattribute 448  
Arbeiten mit Delta-Elementen 481  
Archivieren mit Delta-Technik 96  
Attribute  
    inhaltsbeschreibende 48  
    organisatorische 48  
    vom Benutzer vergeben 48  
    von Zugriffsmethode vergeben 47  
Attributsatz 116  
Aufbau einer Bibliothek 32  
Aufbauen Delta-Baum 99  
Auflisten  
    Inhaltsverzeichnis 414  
    von Elementen 91

- Aufnahmen
  - von Dateien 173
  - von Delta-Elementen 98
  - von Elementen 88
  - von Quellprogrammen 467
- Auftragsschalter 109
  - 4 gesetzt 102
- Ausfügen von Sätzen 312
- Ausgabe
  - in Arbeitsdatei 9 (EDT) 362
  - in Bibliothekselement 362
  - nach SYSLST 361
  - nach SYSOUT 361
  - umlenken 22
  - unterdrücken 361
- Ausgabe umlenken 22
- Ausgabebibliothek 33
- Ausgabemedium bestimmen 361
- Ausgangs-Objekte 137
- Ausgangskomponente 202
- Ausgeben
  - Inhaltsverzeichnis 95
  - Vergleichsstatistik 444
  - von Elementen 91
  - von Elementen als Dateien 265
  - von Vergleichsstatistiken 490
- Ausleihmechanismus 27
- Ausleihverfahren 132
- Auswahlangabe
  - bei make 139
  - Beispiele 43
  - für Elementbezeichnung 43
  - Negativauswahl 43
- B**
- BASE# 347
- Basistyp 40
- Basisversion 55
- Bearbeiten von Elementen 88
- Beenden
  - LMS-Lauf 78, 264
  - MODIFY-ELEMENT-Subanweisungen 300, 311
- BEGIN-BLOCK (SDF-P-Kommando) 73
- BEGIN-MAKE (Anweisung) 184
- Behälter 96
- Beispiele
  - für typische LMS-Anwendungen 467
- Benutzeranschlüsse 108
- Benutzerausgang
  - anzeigen 452
  - lösen 236
- Benutzerausgang (Schnittstelle) 168
- Benutzerprogramm
  - aktivieren 166
  - nachladen 168
  - verzweigen in 166, 493
- Benutzertyp 40
- Beweissicherung 70
- Bibliothek 32
  - arbeiten mit mehreren 84
  - Definition 19
  - Eigenschaften 19
  - Element aufnehmen 88
  - eröffnen (Lesen und Schreiben) 80
  - eröffnen (Lesen) 80
  - globale 80
  - globale und lokale (Gültigkeit) 84
  - Inhaltsverzeichnis ausgeben 95
  - interne Dateioorganisation 110
  - komplett kopieren 233
  - kopieren auf NK4-Pubset 235
  - lokale 81
  - neu einrichten 80
  - reorganisieren 386
  - schließen 206
  - Zustand anzeigen 437
  - zuweisen 79
- Bibliotheksattribute
  - ändern 333
  - anzeigen 434
- Bibliotheksformat 116
  - konvertieren 118
- Bibliotheksliste 79, 82, 83, 516, 517
- Bildschirmwechsel steuern 104
- Bindelademodule
  - ändern 285
  - korrigieren 94

- Bindelademodule (LLM) 38
- Bindemodule 39
  - ändern 285, 484
  - korrigieren 94
- Blätter (Delta-Baum) 53
- BLKCTRL 110
- BLKCTRL-Wert
  - PAM-Dateien 112
  - Phasen 112
  - SAM/ISAM-Dateien 112
- Blockkontrollfeld (CF) 117
  
- C**
- c-string (Datentyp) 149
- CALL-EDT (Anweisung) 204
- case-sensitive (Zusatz zu Datentypen) 161
- cat (Zusatz zu Datentypen) 160
- cat-id (Datentyp) 149
- CCS (coded character set) 121
- CCSN (coded character set name) 121
  - auswerten 125
  - explizit setzen 124
  - implizit setzen 123
  - protokollieren 124
- CLASS2-Option
  - NONKEY 113
  - PAMKEY 113
- CLOSE-LIBRARY (Anweisung) 206
- Code (codierter Zeichensatz) 121
- codierter Zeichensatz 121
- command-rest (Datentyp) 149
- COMPARE-ELEMENT (Anweisung) 208
- Compiler-Ergebnisinformationen 38
- compl (Zusatz zu Datentypen) 155
- composed-name (Datentyp) 149
- COPY-ELEMENT (Anweisung) 220
- COPY-LIBRARY (Anweisung) 233
- corr (Zusatz zu Datentypen) 160, 161
- CPU-LIMIT-Operand 72
- CSECT-Name 307
  
- D**
- Darstellungsmittel
  - Handbuch 14
- date (Datentyp) 149
- Datei aufnehmen 173
- Dateiattribut BLKCTRL 110
- Dateikettungsname 81
- Daten
  - beliebigen Formates 39
  - druckaufbereitet 38
- Datenschutz durch Überschreiben 69
- Datentransfer Datei-Bibliothekselement 110
- Datentypen SDF 146, 149
  - Zusätze 146
- Definition
  - Satzidentifikation 308
- DELETE-ELEMENT (Anweisung) 238
- DELETE-RECORD-TYPE (MODIFY-ELEMENT-Subanweisung) 299
- DELTA 347
- Delta
  - als Ordnungsmittel 97
  - als Speicherungsform 97
- Delta-Baum 53, 97
  - aufbauen 99
  - Blätter 53
  - Elemente 53
  - Speicherung 53
- Delta-Bildung 97
- Delta-Element 96
  - aufnehmen 98
  - bearbeiten 481
  - löschen 91
  - ordnen 97
  - speichern 97
  - überschreiben 100
  - umbenennen 100
- Delta-Menge 98
- Delta-Nummer, interne 99, 347
- Delta-Sequenz 97, 98
- Delta-Speicherung 53, 96
- Delta-Technik 20, 35, 96
- DELTA# 347
- DESTROY-DATA 69
  - als Elementattribut 69
  - als Verarbeitungsparameter 69
- device (Datentyp) 149



END (Anweisung) 78, 264  
END-MAKE (make-Subanweisung) 191  
END-MODIFY (MODIFY-ELEMENT-Subanweisung) 300, 311  
Erfolgsmeldung 103  
Erstellen von Textelementen 245  
Erstellungsdatum 47  
Erzeugen  
  von ISAM-Dateien 266  
  von S-Variablen 24  
Extra-Seitenvorschub (Protokollparameter) 363  
EXTRACT-ELEMENT (Anweisung) 265

## F

Fehlerbehandlung 104  
Festlegen von globalen Parametern 304  
filename (Datentyp) 150  
fixed (Datentyp) 149  
Folgezeilen 162  
formaler Vergleich 92  
Format  
  der LMS-Ausgaben in S-Variable 455  
Fragemechanismus im Dialog 28  
from-object 140  
full-filename siehe Datentyp filename 150  
Funktionen von LMS 71

## G

Geltungsbereich (Scope) 55  
gen (Zusatz zu Datentypen) 160  
globale Bibliothek 80  
  zuweisen 81  
Graph 140  
  Komponente 141

## H

Handbuch  
  Darstellungsmittel 14  
Hilfsdatei  
  für EDT 245  
  Name 245

## I

IFG-Benutzer-Profile 39  
IFG-Formatmasken 38  
INCLUDE-Satz 299  
Index  
  -Schreibweise 159  
  global 158  
  Konstruktionszeichenfolge 158  
  platzhalter-spezifisch 158  
Inhalt eines Elementes 37  
Inhaltsverzeichnis  
  auflisten 414  
  ausgeben 95  
  einer Bibliothek 32  
  sortieren 414  
  Sortierreihenfolge 431  
initialer Elementschutz 65  
integer (Datentyp) 151  
interne Delta-Nummer 99, 347  
interne Nummer (Vorgänger-Element) 99  
interner Speicher C0 444  
ISAM-Datei 113  
  erzeugen 266, 486  
ISAM-Schlüssel, Schrittweite berechnen 266  
ISD-Sätze 299

## K

Kommentar schreiben 453  
Komponente 131  
  Definition 131  
  Source- 131  
Konstruktionsangabe 159  
  bei make 139  
  für Elementbezeichnung 45  
Konstruktionszeichenfolge 158

### Konvention

- Auswahl Standardbasis 57
- Defaultversion 57
- einstellen 57
- Erhöhungsverfahren 57
- Format 57
- MULTI-SEQUENCE 60
- NONE 58
- Prüfungen 57
- STD-SEQ 132
- STD-SEQUENCE 58
- STD-TREE 59, 132

Konvertieren Bibliotheksformat 118

### Kopieren

- strukturertend 220
- von Elementen 220, 472

Korrekturanweisung, siehe MODIFY-ELEMENT-Subanweisung

Korrekturen rückgängigmachen 306

### Korrigieren

- Bindelademodule 94
- Bindemodule 94
- mit EDIT-ELEMENT 94
- mit MODIFY-ELEMENT 94
- Phasen 94
- Textelemente 94
- von Elementen 94
- von Quellprogrammen 467
- von Textelementen 245
- von Textsätzen 297

Kurzname 148

## L

Listenelemente 38

Listenvariable dynamische 436

LLM, siehe Bindelademodule

### LMS

- beenden 78, 264
- im Dialogbetrieb 28
- im Stapelbetrieb 28
- starten 71

LMS und EDT 126

LMS-Anweisungen, Übersicht 164

LMS-Ausgabe abbrechen 104

### LMS-Lauf

- Abbruch durch Fehler 109
- beenden 78
- starten 71
- steuern 101
- unterbrechen 108

LMS-Protokoll 102

LMS-Protokollparameter 101

logischer Vergleich 92

logisches Löschen 91, 100

lokale Bibliothek 81

- zuweisen 81

### Löschen

- Delta-Element 91
- von Elementen 91, 238
- von Satzarten 299

Lösen des Benutzerausgangs 236

low (Zusatz zu Datentypen) 155

LSD-Sätze 299, 307

## M

make-Funktionalität 27, 133

- Beispiel 508

Einsatzgebiet 133

make-S-Variable definieren 192

make-Subanweisung

- END-MAKE 191
- MODIFY-MAKE-DEFAULTS 192
- SET-POSTPROCESSING 200
- SET-PREPROCESSING 201
- SET-STD-ACTION 137, 202

make-Subanweisungen 134

- beenden 184

einleiten 184

Makros 38

man (Zusatz zu Datentypen) 160, 161

mandatory (Zusatz zu Datentypen) 161

### Mehrfachzugriff

- auf Bibliotheken 33
- auf Delta-Elemente 36
- einschränken 34

Mehrfachzugriff auf Elemente 36

Meldungen der Zugriffsroutine AMCB 521

Metasyntax SDF 146, 147



Mißerfolgsmeldung 103  
 MODIFY-CSECT-ATTRIBUTES (MODIFY-ELEMENT-Subanweisung) 301  
 MODIFY-ELEMENT (Anweisung) 285  
 MODIFY-ELEMENT-ATTRIBUTES (Anweisung) 313  
 MODIFY-ELEMENT-PROTECTION (Anweisung) 262, 323  
 MODIFY-ELEMENT-Subanweisung  
   ADD-RECORD 310  
   ADD-REP-RECORD 296  
   ADD-TEXT-MODIFICATION 297  
   DELETE-RECORD-TYPE 299  
   END-MODIFY 300, 311  
   MODIFY-CSECT-ATTRIBUTES 301  
   MODIFY-MODIFICATION-DEFAULTS 304  
   REMOVE-MODIFICATION 306  
   REMOVE-RECORD 312  
   RENAME-SYMBOLS 307  
 MODIFY-ELEMENT-Subanweisungen  
   ausführen 295  
   beenden 300, 311  
   für Elementtypen R, C, L 295  
   für Textelemente 308  
 MODIFY-LIBRARY-ATTRIBUTES (Anweisung) 333  
 MODIFY-LMS-DEFAULTS (Anweisung) 339  
 MODIFY-LOGGING-PARAMETERS (Anweisung) 360  
 MODIFY-MAKE-DEFAULTS (make-Subanweisung) 192  
 MODIFY-MODIFICATION-DEFAULTS (MODIFY-ELEMENT-Subanweisung) 304  
 MODIFY-TYPE-ATTRIBUTES (Anweisung) 365  
 MONJV-Operand 71

**N**

Nachladen des Benutzerprogramms 168  
 name (Datentyp) 151  
 Namenskonvention 49  
 Negativauswahl 43  
 NK4-Platten 116

**O**

odd (Zusatz zu Datentypen) 160  
 OPEN-LIBRARY (Anweisung) 373

**P**

PAM-Datei 112, 114  
 PAM-Elemente 39  
 Pamkey-Eliminierung 110  
 partial-filename (Datentyp) 152  
 path-compl (Zusatz zu Datentypen) 155  
 Phasen 38  
   ändern 285  
   korrigieren 94  
 Physikalisches Löschen 91, 100, 238  
 PLAM 11  
 Plattenplatzbedarf vermindern 386  
 Platzhalter 43  
 POSIX  
   Platzhalter 156  
   posix-filename (Datentyp) 152  
   posix-pathname (Datentyp) 152  
   Primärelement 92  
   product-version (Datentyp) 153  
   Produktbestandteile 527  
   Programmabschnittsmerkmale verändern 301  
   Programmbibliothek 11  
   Programmbibliotheken 19  
   Programmsystem 131  
     aktuelles 131  
     Definition 131  
     Sourcen 141  
   Protokollausgabe steuern 102  
   Protokollformat steuern 102  
   Protokollparameter  
     ändern 360  
     anzeigen 443  
     bestimmen 363  
   Protokollumfang steuern 102  
 PROVIDE-ELEMENT (Anweisung) 377  
 Prozedur 38  
   speichern 95

### Q

- Quellprogramm 39
  - aufnehmen 467
  - korrigieren 467
  - übersetzen 467
- Quellversion 54
- quotes (Zusatz zu Datentypen) 161

### R

- Readme-Datei 15
- Referenzeintrag 49
- Referenzjahr 48
- Referenzsatz 49
- Remote File Access 33, 34, 47, 69
- REMOVE-MODIFICATION (MODIFY-ELEMENT-Subanweisung) 306
- REMOVE-RECORD (MODIFY-ELEMENT-Subanweisung) 312
- RENAME-SYMBOLS (MODIFY-ELEMENT-Subanweisung) 307
- Reorganisieren Bibliothek 386
- REORGANIZE-LIBRARY (Anweisung) 386
- REP-Satz 299
  - einfügen 296
- Reservieren Element 377
- RETURN-ELEMENT (Anweisung) 391
- RFA 33, 34, 47, 69
- Rückgängigmachen von Korrekturen 306

### S

- S-Variable
  - erzeugen 24
  - Format der LMS-Ausgaben 455
  - SYSLMSPAR 75
- SAM-Datei 114
  - erzeugen 486
- SAM/ISAM-Datei 112
- SAT (security audit trail) 70
- Satz oder Satzbereich im Element löschen 312
- Satzarten löschen 299
- Satzidentifikation
  - definieren 308
  - für Textelemente 308
- Satzkennung 308

- Satznummer 308
- Schließen Bibliothek 206
- Schlüsselwort
  - \*UNCHANGED 101
- Schnittstelle des Benutzerausgangs 168
- Schutzattribute, Übersicht 66
- Scope (Geltungsbereich) 55
- SDF-Standardanweisungen 145
- Seitennumerierung 361
- Sekundärattribut 49
- Sekundärelement 92
- Sekundärinhaltsverzeichnis 49
- Sekundärname 47, 49
- SEND-MESSAGE-Kommando 109
- sep (Zusatz zu Datentypen) 160
- SET-POSTPROCESSING (make-Subanweisung) 200
- SET-PREPROCESSING (make-Subanweisung) 201
- SET-STD-ACTION (make-Subanweisung) 137, 202
- SHOW-ELEMENT (Anweisung) 400
- SHOW-ELEMENT-ATTRIBUTES (Anweisung) 414
- SHOW-LIBRARY-STATUS (Anweisung) 437
- SHOW-LMS-DEFAULTS (Anweisung) 439
- SHOW-LOGGING-PARAMETERS (Anweisung) 443
- SHOW-STATISTICS (Anweisung) 444
- SHOW-TYPE-ATTRIBUTES (Anweisung) 448
- SHOW-USER-EXITS (Anweisung) 452
- SoftBooks 16, 103
- Software-Entwicklungsprozeß 131
  - Ausleihverfahren 132
  - make-Funktionalität 133
- Softwareprodukt XHCS 122
- Sortieren Inhaltsverzeichnis 414
- Sortierreihenfolge für Inhaltsverzeichnis 431
- Source im Programmsystem 141
- Source(-Komponente) 131
- spec (Zusatz zu Datentypen) 160
- Speichern
  - von Delta-Elementen 97
  - von textartigen Elementen 96

Speicherungsform 35, 47  
 Spin-off Mechanismus 104  
 Spin-off-Mechanismus 139  
 Standard-Aktionen angeben 137, 202  
 Standardwerte  
     ändern 339  
     ausgeben 439  
 Stapelbetrieb 28  
 START-LMS-Kommando 71  
 Startdatei von LMS 75  
 Starten  
     LMS-Lauf 71  
 Steuern LMS-Lauf 101  
 structured-name (Datentyp) 153  
 Subanweisungen 94  
 Subsystem  
     ACS 119  
     XHCS-SYS 122  
 SUPER-TYPE 367  
 Symbole umbenennen 307  
 SYSLMSPAR 75  
 SYSPAR.LMS 75  
 Systemkatalog, Eintrag 32

**T**

target-object 140  
 Task-File-Table (TFT) 116  
 temp-file (Zusatz zu Datentypen) 161  
 text (Datentyp) 153  
 textartige Elementtypen 39  
 Textdaten 38  
 Textelement 39  
     erstellen 245  
     korrigieren 94, 245  
 Textsätze korrigieren 297  
 time (Datentyp) 153  
 TOUCH 140  
 TXTP-Sätze 299  
 Typ, übergeordneter 367  
 Typattribute anzeigen 448  
 Typbäume 40  
 Typumlenkung 79  
 Typverträglichkeit 50

**U**

Überschrift ausgeben (Protokollparameter) 364  
 Übersetzen von Quellprogrammen 467  
 Übersicht LMS-Anweisungen 164  
 Umbenennen  
     von Delta-Elementen 100  
     von Elementen 95, 313  
     von Symbolen 307  
     von Versionen 313  
 Umlenken Ausgabe 22  
 under (Zusatz zu Datentypen) 155  
 Unicode-Zeichensätze  
     UTF16 121  
     UTF8 121  
     UTFE 121  
 Unterbrechen  
     LMS-Lauf 108  
 Unterstützung von MIP-Variablen 24  
 UPAM-Datei 117  
 user (Zusatz zu Datentypen) 161

**V**

Variantennummer 42  
 Verändern von  
     Programmabschnittsmerkmalen 301  
 Vergleich  
     formal 92  
     logisch 92  
 Vergleichen  
     von Elementen 92, 476  
     von textartigen Elementen 208  
 Vergleichsbasis 97  
 Vergleichsergebnis 93  
 Vergleichsprotokoll 93, 208  
 Vergleichsstatistik 93, 208  
     ausgeben 444, 490  
 vers (Zusatz zu Datentypen) 161  
 Version umbenennen 313  
 VERSION-Operand 71  
 Versionsangabe  
     'XFF' 54  
     \*BY-SOURCE 54  
     Syntax 42  
 Versionsbezeichnung 54

Versionshaltung [52](#)  
Versionskonvention [57](#)  
Versionsspeicherung [52](#)  
Versionsverwaltung [52](#)  
Verwaltungsrecht  
    für Bibliothek [61](#)  
    für einen Typ [61](#)  
Verzweigen in Benutzerprogramm [166, 493](#)  
Voll-Element [96](#)  
Voll-Speicherung [96](#)  
Voraussetzungen zur Software-Konfiguration [18](#)  
Voreinstellungen von LMS [77](#)  
Vorgängerelement [97](#)  
vsn (Datentyp) [153](#)  
VTSU (virtual terminal support) [122](#)

### W

wild(n) (Zusatz zu Datentypen) [156](#)  
wildcard-Angaben [43](#)  
wildcard-Syntax [43](#)  
with (Zusatz zu Datentypen) [155](#)  
without (Zusatz zu Datentypen) [160](#)  
WRITE-COMMENT (Anweisung) [453](#)

### X

x-string (Datentyp) [154](#)  
x-text (Datentyp) [154](#)  
XHCS [122](#)

### Z

Zeichensatz  
    codierter [121](#)  
    erweiterter [121](#)  
Zeilenlänge (Protokollparameter) [363](#)  
Ziel-Objekte [137](#)  
Zielkomponente [202](#)  
Zielversion [54](#)  
Zugriffsdatum und -zeit [47](#)  
Zugriffsrecht für Elemente [61](#)  
    Ausführungsrecht [61](#)  
    Ausleihrecht [61](#)  
    Leserecht [61](#)  
    Schreibrecht [61](#)  
Zusätze zu Datentypen [146, 155](#)  
Zusatzinformationen  
    zu LMS-Meldungen [519](#)  
Zuweisen von Bibliotheken [79](#)